



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2008년02월27일  
 (11) 등록번호 10-0807817  
 (24) 등록일자 2008년02월20일

(51) Int. Cl.

*H04L 12/40* (2006.01)

(21) 출원번호 10-2004-0107498  
 (22) 출원일자 2004년12월17일  
 심사청구일자 2004년12월17일  
 (65) 공개번호 10-2006-0069646  
 (43) 공개일자 2006년06월22일

(56) 선행기술조사문헌  
 1020040107498 - 691960  
 US2003-101265A1

전체 청구항 수 : 총 17 항

(73) 특허권자

**엔에이치엔(주)**

경기도 성남시 분당구 정자동 25-1 분당벤처타운

(72) 발명자

**천경욱**

서울 양천구 목1동 하이페리온 102동 4502호

(74) 대리인

**특허법인무한**

심사관 : 김병균

**(54) 버스형 네트워크 구조의 통신 네트워크 시스템에서서브시스템 사이의 로드를 조절하는 방법**

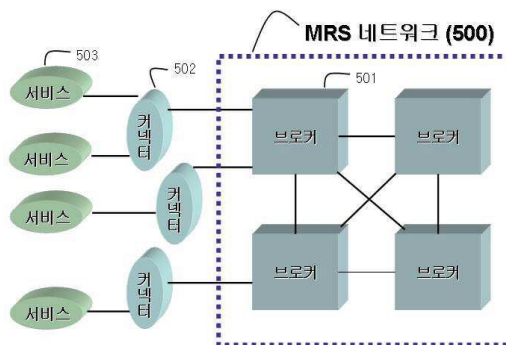
**(57) 요약**

본 발명은 버스형 네트워크 구조의 통신 네트워크 시스템에서 서브시스템(subsystem) 사이의 로드(load)를 조절하는 방법에 관한 것으로, 더욱 상세하게는 어느 한 프로세스 내에 존재하는 하나 이상의 서비스(service)를 하나의 커넥터(connector)를 통해 메시지 라우팅을 처리하는 브로커(broker)와 연결하고 네트워크상의 모든 브로커를 완전 메시(full mesh) 형태로 상호 연결하는 버스형 네트워크 구조에 있어서, 각 브로커 간 로드 밸런스(load balance)를 적절히 조절하기 위한 방법에 관한 것이다.

본 발명에 따른 버스형 네트워크 구조의 통신 네트워크 시스템에서 서브시스템 사이의 로드를 조절하는 방법은, 상기 통신 네트워크 시스템은 메시지 라우팅을 처리하는 브로커와, 커넥터와, 상기 커넥터를 통해 상기 브로커에 연결되는 통신 가능한 터미널 노드(terminal node)인 다수의 서비스를 포함하고, 상기 통신 네트워크 시스템상의 각 브로커를 완전 메시(full mesh) 형태로 상호 연결하는 단계(상기 브로커는 메시지 라우팅을 처리하기 위해 라우팅 경로를 설정하거나 상기 커넥터와의 연결 관계를 설정하는 모듈임), 상기 커넥터에서 상기 각 브로커의 네트워크 주소가 기록된 브로커 리스트를 유지하는 단계, 상기 커넥터가 상기 서비스로부터 상기 서비스 등록 메시지를 수신하고, 상기 브로커 리스트 중 하나의 브로커를 임의로 선택하여 연결 요청 메시지를 전송하는 단계, 상기 선택된 브로커가 상기 연결 요청 메시지를 수신하고, 자신의 현재 로드와 로드 한계를 고려하여 상기 커넥터와의 연결 여부를 판단하는 단계를 포함하는 것을 특징으로 한다.

본 발명에 따르면, 버스형 네트워크 구조를 통해 서버 간 연결 구조가 단순화되고, 각 브로커의 현재 로드와 로드 한계에 기초하여 커넥터를 통해 서비스가 등록된 브로커를 선택하고, 브로커의 정상 및 비정상 종료 시에 각 커넥터에 연결된 서비스들이 다른 브로커로 동적으로 이동하여 브로커 간 로드 밸런스가 자동적으로 조절되는 로드 조절 방법이 제공된다.

**대표도** - 도5



## 특허청구의 범위

### 청구항 1

버스형 네트워크 구조의 통신 네트워크 시스템에서 서브시스템(subsystem) 사이의 로드(load)를 조절하는 방법에 있어서,

상기 통신 네트워크 시스템은 메시지 라우팅을 처리하는 브로커(broker)와, 커넥터(connector)와, 상기 커넥터를 통해 상기 브로커에 연결되는 통신 가능한 터미널 노드(terminal node)인 다수의 서비스(service)를 포함하고,

상기 통신 네트워크 시스템상의 각 브로커를 완전 메시(full mesh) 형태로 상호 연결하는 단계 - 상기 브로커는 메시지 라우팅을 처리하기 위해 라우팅 경로를 설정하거나 상기 커넥터와의 연결 관계를 설정하는 모듈임 -;

상기 커넥터에서 상기 각 브로커의 네트워크 주소가 기록된 브로커 리스트를 유지하는 단계;

상기 커넥터가 상기 서비스로부터 상기 서비스 등록 메시지를 수신하고, 상기 브로커 리스트 중 하나의 브로커를 임의로 선택하여 연결 요청 메시지를 전송하는 단계; 및

상기 선택된 브로커가 상기 연결 요청 메시지를 수신하고, 자신의 현재 로드와 로드 한계를 고려하여 상기 커넥터와의 연결 여부를 판단하는 단계

를 포함하고,

상기 연결 여부를 판단하는 상기 단계는,

상기 커넥터와 연결이 설정될 시 상기 선택된 브로커에 연결되는 커넥터의 총 개수가 상기 로드 한계를 초과할 경우 상기 연결 요청을 거부하는 것을 특징으로 하는 로드 조절 방법.

### 청구항 2

삭제

### 청구항 3

제1항에 있어서,

상기 연결 요청이 거부된 경우, 상기 커넥터가 상기 브로커 리스트 중 상기 선택된 브로커를 제외한 다른 브로커를 임의로 선택하여 상기 연결 요청 메시지를 전송하는 단계를 더 포함하는 것을 특징으로 하는 로드 조절 방법.

### 청구항 4

버스형 네트워크 구조의 통신 네트워크 시스템에서 서브시스템(subsystem) 사이의 로드(load)를 조절하는 방법에 있어서,

상기 통신 네트워크 시스템은 메시지 라우팅을 처리하는 브로커(broker)와, 커넥터(connector)와, 상기 커넥터를 통해 상기 브로커에 연결되는 통신 가능한 터미널 노드(terminal node)인 다수의 서비스(service)를 포함하고,

상기 통신 네트워크 시스템상의 각 브로커를 완전 메시(full mesh) 형태로 상호 연결하는 단계 - 상기 브로커는 메시지 라우팅을 처리하기 위해 라우팅 경로를 설정하거나 상기 커넥터와의 연결 관계를 설정하는 모듈임 -;

상기 통신 네트워크 시스템으로부터 정상적으로 종료되는 제1 브로커가 상기 각 브로커 중 하나인 제2 브로커를 선택하는 단계;

상기 제1 브로커에 연결된 커넥터에게 상기 제2 브로커로 연결을 재설정할 것을 요청하는 연결 재설정 메시지를 전송하는 단계; 및

상기 커넥터가 상기 연결 재설정 메시지를 수신하고, 상기 제2 브로커와 연결을 설정하여 상기 커넥터에 연결된 서비스를 등록하기 위한 서비스 등록 메시지를 상기 제2 브로커로 전송하는 단계

를 포함하고,

상기 커넥터와 상기 서비스는 동일한 프로세스 내에 포함되는 것을 특징으로 하는 로드 조절 방법.

#### 청구항 5

제4항에 있어서,

상기 제1 브로커가 상기 각 브로커로 종료 예정 정보를 전송하는 단계; 및

상기 각 브로커가 상기 종료 예정 정보를 수신하고, 상기 각 브로커에 연결된 커넥터에게 상기 종료 예정 정보를 전송하는 단계

를 더 포함하는 것을 특징으로 하는 로드 조절 방법.

#### 청구항 6

제4항에 있어서,

상기 제2 브로커가 자신의 라우팅 정보를 갱신하고, 상기 제1 브로커 및 상기 각 브로커로 상기 갱신된 라우팅 정보를 전송하는 단계

를 더 포함하는 것을 특징으로 하는 로드 조절 방법.

#### 청구항 7

제6항에 있어서,

상기 제1 브로커가 상기 갱신된 라우팅 정보를 수신하고, 상기 갱신된 라우팅 정보를 이용하여 자신의 라우팅 테이블을 갱신하는 단계;

상기 갱신된 라우팅 테이블에 상기 커넥터와 연관된 라우팅 정보가 존재하는지 여부를 판단하는 단계; 및

상기 판단 결과 상기 커넥터와 연관된 라우팅 정보가 존재하지 않는 경우, 상기 커넥터와의 연결을 최종 종료하는 단계

를 더 포함하는 것을 특징으로 하는 로드 조절 방법.

#### 청구항 8

제4항에 있어서,

상기 연결 재설정 메시지의 전송 시로부터 소정의 시간이 경과한 후에도 상기 커넥터가 상기 제1 브로커와 연결을 유지하는 경우, 상기 제1 브로커가 상기 커넥터와의 연결을 종료하고 자신을 종료시키는 단계

를 더 포함하는 것을 특징으로 하는 로드 조절 방법.

#### 청구항 9

제4항에 있어서,

상기 제1 브로커와 상기 각 브로커 상호 간에는 브로커별 현재 로드와 로드 한계에 대한 정보를 포함하는 브로커 상태 메시지가 교환되고,

상기 제2 브로커를 선택하는 상기 단계에서, 상기 제1 브로커는 상기 브로커 상태 메시지에 기초하여 상기 제2 브로커를 선택하는 것을 특징으로 하는 로드 조절 방법.

#### 청구항 10

버스형 네트워크 구조의 통신 네트워크 시스템에서 서브시스템(subsystem) 사이의 로드(load)를 조절하는 방법에 있어서,

상기 통신 네트워크 시스템은 메시지 라우팅을 처리하는 브로커(broker)와, 커넥터(connector)와, 상기 커넥터를 통해 상기 브로커에 연결되는 통신 가능한 터미널 노드(terminal node)인 다수의 서비스(service)를

포함하고,

상기 통신 네트워크 시스템상의 각 브로커를 완전 메시(full mesh) 형태로 상호 연결하는 단계 - 상기 브로커는 메시지 라우팅을 처리하기 위해 라우팅 경로를 설정하거나 상기 커넥터와의 연결 관계를 설정하는 모듈임 -;

상기 통신 네트워크 시스템으로부터 비정상적으로 종료되는 제1 브로커를 상기 각 브로커 중 하나인 제2 브로커가 감지하는 단계;

상기 제2 브로커가 자신의 로드 한계를 상향 조정하고, 상기 제1 브로커에 연결된 커넥터 중 적어도 일부 커넥터와 연결을 설정하는 단계; 및

상기 제2 브로커가 상기 연결 설정에 따라 자신의 라우팅 정보를 갱신하고, 상기 각 브로커로 상기 갱신된 라우팅 정보를 전송하는 단계

를 포함하고,

상기 커넥터와 상기 서비스는 동일한 프로세스 내에 포함되는 것을 특징으로 하는 로드 조절 방법.

### 청구항 11

제10항에 있어서,

상기 일부 커넥터와 연결을 설정하는 상기 단계는,

상기 제1 브로커와 연결된 커넥터의 수를 상기 제1 브로커를 제외한 상기 각 브로커의 총 개수로 나눈 값에 따라 상기 일부 커넥터의 수를 결정하는 단계

를 포함하는 것을 특징으로 하는 로드 조절 방법.

### 청구항 12

제10항에 있어서,

상기 각 브로커에 연결된 커넥터에게 상기 제1 브로커의 종료 정보를 전송하는 단계; 및

상기 커넥터가 상기 종료 정보를 수신하여 자신의 브로커 리스트를 갱신하는 단계

를 더 포함하는 것을 특징으로 하는 로드 조절 방법.

### 청구항 13

제10항에 있어서,

상기 제1 브로커와 상기 각 브로커 상호 간에는 브로커별 현재 상태에 대한 정보를 포함하는 브로커 상태 메시지가 교환되고,

상기 통신 네트워크 시스템으로부터 비정상적으로 종료되는 제1 브로커를 상기 각 브로커 중 하나인 제2 브로커가 감지하는 상기 단계에서, 상기 제2 브로커는 상기 브로커 상태 메시지가 정상적으로 수신되지 않는 제1 브로커를 감지하는 것을 특징으로 하는 로드 조절 방법.

### 청구항 14

버스형 네트워크 구조의 통신 네트워크 시스템에서 서브시스템(subsystem) 사이의 로드(load)를 조절하는 방법에 있어서,

상기 통신 네트워크 시스템은 메시지 라우팅을 처리하는 브로커(broker)와, 커넥터(connector)와, 상기 커넥터를 통해 상기 브로커에 연결되는 통신 가능한 터미널 노드(terminal node)인 다수의 서비스(service)를 포함하고,

상기 통신 네트워크 시스템상의 각 브로커를 완전 메시(full mesh) 형태로 상호 연결하는 단계 - 상기 브로커는 메시지 라우팅을 처리하기 위해 라우팅 경로를 설정하거나 상기 커넥터와의 연결 관계를 설정하는 모듈임 -;

상기 커넥터에서 상기 각 브로커의 네트워크 주소가 기록된 브로커 리스트를 유지하는 단계;

상기 통신 네트워크 시스템으로부터 비정상적으로 종료되는 제1 브로커를 상기 커넥터가 감지하는 단계;  
 상기 브로커 리스트로부터 상기 제1 브로커의 네트워크 주소를 삭제하여 상기 브로커 리스트를 갱신하는 단계;  
 상기 갱신된 브로커 리스트 중 임의의 제2 브로커를 선택하고, 상기 제2 브로커와 연결을 설정하여 상기 커넥터에 연결된 서비스를 등록하기 위한 서비스 등록 메시지를 상기 제2 브로커로 전송하는 단계를 포함하고,  
 상기 커넥터와 상기 서비스는 동일한 프로세스 내에 포함되는 것을 특징으로 하는 로드 조절 방법.

**청구항 15**

제14항에 있어서,  
 상기 제2 브로커가 자신의 라우팅 정보를 갱신하고, 상기 각 브로커로 상기 갱신된 라우팅 정보를 전송하는 단계를 더 포함하는 것을 특징으로 하는 로드 조절 방법.

**청구항 16**

제1항, 제4항, 제10항 및 제14항 중 어느 하나의 항에 있어서,  
 각 커넥터에는 하나의 브로커만이 연결되고,  
 각 서비스는 하나의 커넥터를 통해 상기 브로커와 연결되고,  
 상기 커넥터와 상기 서비스는 동일한 프로세스 내에 포함되는 것을 특징으로 하는 로드 조절 방법.

**청구항 17**

제16항에 있어서,  
 상기 커넥터는 하나의 프로세스 내에 하나만 존재하고, 상기 서비스는 하나의 프로세스 내에 적어도 하나 이상 존재하는 것을 특징으로 하는 로드 조절 방법.

**청구항 18**

제1항 및 제3항 내지 제15항 중 어느 한 항의 방법을 실행하기 위한 프로그램이 기록되어 있는 것을 특징으로 하는 컴퓨터에서 판독 가능한 기록 매체.

**명세서**

**발명의 상세한 설명**

**발명의 목적**

**발명이 속하는 기술 및 그 분야의 종래기술**

- <28> 본 발명은 버스형 네트워크 구조의 통신 네트워크 시스템에서 서브시스템(subsystem) 사이의 로드(load)를 조절하는 방법에 관한 것으로, 더욱 상세하게는 어느 한 프로세스 내에 존재하는 하나 이상의 서비스(service)를 하나의 커넥터(connector)를 통해 메시지 라우팅을 처리하는 브로커(broker)와 연결하고 네트워크상의 모든 브로커를 완전 메시(full mesh) 형태로 상호 연결하는 버스형 네트워크 구조에 있어서, 각 브로커 간 로드 밸런스(load balance)를 적절히 조절하기 위한 방법에 관한 것이다.
- <29> 종래기술에서는 게임 서비스를 제공하는 게임 서버들이 메시(mesh)형 구조로 모두 연결되는데, 도 1은 이러한 종래기술에 따른 게임 서버간 네트워크 연결 모습을 도시한 도면이다.
- <30> 도 1과 같은 모습의 종래기술에 따르면, 게임 서버들이 추가로 연결될 때마다 전체 네트워크 관점에서의 연결 수가 기하급수적으로 증가할 수 밖에 없는데, 도 2는 종래기술에 있어서 게임 서버의 수가 계속적으로 증가함에 따라 발생할 수 있는 게임 서버간 네트워크 연결 모습을 도시한 도면이다.

- <31> 이와 같이 게임 서버들이 메시형 구조로 상호 연결되는 종래기술에 따르면, 게임 서버의 수가 증가함에 따라 그 연결 구조가 매우 복잡해지며, 그 결과 게임 사용자의 증가에 따른 게임 서버의 확장이 어려워지는 문제점이 발생하였다. 특히, 게임 서버간 연결을 글로벌로 확장할 경우에는 특히 그 관리가 어려워질 수 밖에 없었다.
- <32> 더욱이, 하나의 서버가 연결되는 상대 서버의 수가 많을 수록 전체 연결 수는 기하급수적으로 증가하게 되는데, 종래기술에서 하나의 게임 서버는 통상적으로 로그인 서버, 랭킹 서버, 데이터베이스 서버와 연결되며, 여기에 추가적으로 채널리스트 서버 및 공지 서버와 하나의 멀티캐스트 그룹으로 묶여 있으므로, 실제적으로 전체 네트워크에 존재하는 연결 수는 단순한 게임 서버간 연결 수를 크게 초과하여 그 관리가 매우 어려운 실정이다.
- <33> 이에, 모든 게임 서버들이 메시 형태로 연결되는 종래기술에 따른 네트워크 구조를 과감히 탈피하고, 서버간 연결 구조를 단순화하여 그 관리가 용이하고 효율적인 서비스 확장이 가능한 새로운 통신 네트워크 구조의 출현이 요청되어 왔으며, 이러한 새로운 통신 네트워크 구조에서 각 서브시스템 간 로드 밸런스를 효율적으로 조절할 수 있는 방법이 요구된다.

**발명이 이루고자 하는 기술적 과제**

- <34> 본 발명은 상술한 바와 같은 종래기술의 문제점을 해결하기 위해 안출된 것으로서, 버스형 네트워크 구조를 통해 서버 간 연결 구조가 단순화된 새로운 통신 네트워크 시스템에서, 각 브로커의 현재 로드와 로드 한계에 기초하여 커넥터를 통해 서비스가 등록될 브로커를 선택하고, 브로커의 정상 및 비정상 종료 시에 각 커넥터에 연결된 서비스들이 다른 브로커로 동적으로 이동하여 브로커 간 로드 밸런스가 자동적으로 조절되는 로드 조절 방법을 제공하는 것을 그 목적으로 한다.
- <35> 본 발명은 버스형 네트워크 구조를 통해 서버간 연결 구조가 단순화된 새로운 통신 네트워크 시스템에서 각 서브시스템 간 로드 분산을 효율적으로 조절함으로써, 서비스 확장성 및 성능 확장성을 지원하면서도 유지 및 관리가 용이한 통신 네트워크 시스템을 구축하는 것을 그 목적으로 한다.
- <36> 본 발명은 모든 게임 서버들이 메시 형태로 연결되는 종래기술에 따른 네트워크 구조를 과감히 탈피하고, 서버간 연결 구조를 단순화하여 그 관리가 용이하고 효율적인 서비스 확장이 가능한 새로운 통신 네트워크 구조를 제시하는 것을 그 목적으로 한다.

**발명의 구성 및 작용**

- <37> 상기의 목적을 달성하고, 상술한 종래기술의 문제점을 해결하기 위하여, 본 발명의 일실시예에 따른 버스형 네트워크 구조의 통신 네트워크 시스템에서 서브시스템 사이의 로드를 조절하는 방법은, 상기 통신 네트워크 시스템은 메시지 라우팅을 처리하는 브로커(broker)와, 커넥터(connector)와, 상기 커넥터를 통해 상기 브로커에 연결되는 통신 가능한 터미널 노드(terminal node)인 다수의 서비스(service)를 포함하고, 상기 통신 네트워크 시스템상의 각 브로커를 완전 메시(full mesh) 형태로 상호 연결하는 단계(상기 브로커는 메시지 라우팅을 처리하기 위해 라우팅 경로를 설정하거나 상기 커넥터와의 연결 관계를 설정하는 모듈임), 상기 커넥터에서 상기 각 브로커의 네트워크 주소가 기록된 브로커 리스트를 유지하는 단계, 상기 커넥터가 상기 서비스로부터 상기 서비스 등록 메시지를 수신하고, 상기 브로커 리스트 중 하나의 브로커를 임의로 선택하여 연결 요청 메시지를 전송하는 단계, 상기 선택된 브로커가 상기 연결 요청 메시지를 수신하고, 자신의 현재 로드와 로드 한계를 고려하여 상기 커넥터와의 연결 여부를 판단하는 단계를 포함하는 것을 특징으로 한다.
- <38> 본 발명의 다른 실시예에 따른 로드 조절 방법은 상기 통신 네트워크 시스템상의 각 브로커를 완전 메시 형태로 상호 연결하는 단계(상기 브로커는 메시지 라우팅을 처리하기 위해 라우팅 경로를 설정하거나 상기 커넥터와의 연결 관계를 설정하는 모듈임), 상기 통신 네트워크 시스템으로부터 정상적으로 종료되는 제1 브로커가 상기 각 브로커 중 하나인 제2 브로커를 선택하는 단계, 상기 제1 브로커에 연결된 커넥터에게 상기 제2 브로커로 연결을 재설정할 것을 요청하는 연결 재설정 메시지를 전송하는 단계, 상기 커넥터가 상기 연결 재설정 메시지를 수신하고, 상기 제2 브로커와 연결을 설정하여 상기 커넥터에 연결된 서비스를 등록하기 위한 서비스 등록 메시지를 상기 제2 브로커로 전송하는 단계를 포함하는 것을 특징으로 한다.
- <39> 본 발명의 또 다른 실시예에 따른 로드 조절 방법은 상기 통신 네트워크 시스템상의 각 브로커를 완전 메시 형태로 상호 연결하는 단계(상기 브로커는 메시지 라우팅을 처리하기 위해 라우팅 경로를 설정하거나 상기 커넥터와의 연결 관계를 설정하는 모듈임), 상기 통신 네트워크 시스템으로부터 비정상적으로 종료되는 제1 브로커를 상기 각 브로커 중 하나인 제2 브로커가 감지하는 단계, 상기 제2 브로커가 자신의 로드 한계를 상향 조정하고, 상기 제1 브로커에 연결된 커넥터 중 적어도 일부 커넥터와 연결을 설정하는 단계, 상기 제2 브로커가 상기 연



결 설정에 따라 자신의 라우팅 정보를 갱신하고, 상기 각 브로커로 상기 갱신된 라우팅 정보를 전송하는 단계를 포함하는 것을 특징으로 한다.

- <40> 본 발명의 또 다른 실시예에 따른 로드 조절 방법은 상기 통신 네트워크 시스템상의 각 브로커를 완전 메시 형태로 상호 연결하는 단계(상기 브로커는 메시지 라우팅을 처리하기 위해 라우팅 경로를 설정하거나 상기 커넥터와의 연결 관계를 설정하는 모듈임), 상기 커넥터에서 상기 각 브로커의 네트워크 주소가 기록된 브로커 리스트를 유지하는 단계, 상기 통신 네트워크 시스템으로부터 비정상적으로 종료되는 제1 브로커를 상기 커넥터가 감지하는 단계, 상기 브로커 리스트로부터 상기 제1 브로커의 네트워크 주소를 삭제하여 상기 브로커 리스트를 갱신하는 단계, 상기 갱신된 브로커 리스트 중 임의의 제2 브로커를 선택하고, 상기 제2 브로커와 연결을 설정하여 상기 커넥터에 연결된 서비스를 등록하기 위한 서비스 등록 메시지를 상기 제2 브로커로 전송하는 단계를 포함하는 것을 특징으로 한다.
- <41> 이하, 첨부된 도면을 참조하여 본 발명에 따른 통신 네트워크 시스템 및 이러한 통신 네트워크 시스템에서 각 서브시스템 사이의 로드를 조절하는 방법에 대하여 상세히 설명한다.
- <42> 도 3은 본 발명에 따른 버스형 네트워크 구조의 개략적인 모습을 도시한 도면이다.
- <43> 종래기술에서는 도 1과 같이 게임 서버를 비롯한 모든 게임 관련 서버들이 메시(mesh)형 구조로 연결되지만, 본 발명에 따른 버스형 네트워크 구조에서는 도 3과 같이 모든 서버들이 버스(BUS) 구조를 통해 연결되므로 그 연결 구조가 매우 단순화된다.
- <44> 도 4는 본 발명에 따른 버스형 네트워크 구조의 통신 네트워크 시스템의 연결 모습을 개략적으로 도시한 도면이다.
- <45> 도 1과 같은 모습의 종래기술에 따르면, 도 2와 같이 게임 서버들이 추가로 연결될 때마다 전체 네트워크 관점에서의 연결 수가 기하급수적으로 증가할 수 밖에 없다. 반면, 본 발명에 따른 버스형 네트워크 구조에서는 도 4와 같이 각 서버들은 하나의 브로커와의 연결만을 유지하면 되고, 네트워크가 확장되더라도 중간 연결은 브로커들을 통해서 이루어지므로, 서버가 추가로 연결되더라도 전체 네트워크 관점에서의 연결 수가 크게 증가하지 않는다. 따라서, 도 3과 도 4와 같은 본 발명의 버스형 네트워크 구조를 채용한 통신 네트워크 시스템은 새로운 서비스 연계가 용이하며, 그 유지 및 관리 또한 매우 편리할 수 밖에 없다.
- <46> 도 5는 본 발명에 따른 통신 네트워크 시스템에서 브로커, 커넥터 및 서비스의 연결 모습을 도시한 도면이다.
- <47> 도 5에 도시한 것과 같이, 본 발명에 따른 통신 네트워크 시스템은 메시지 라우팅을 처리하는 브로커(501), 커넥터(502), 커넥터(502)를 통해 브로커(501)에 연결되는 다수의 서비스(503)를 포함하여 구성된다.
- <48> 서비스(503)는 통신 가능한 터미널 노드(terminal node)로서, 각 서비스(503)는 하나의 커넥터(502)를 통해 브로커(501)와 연결된다.
- <49> 커넥터(502)는 서비스(503)와 브로커(501)간 연결을 매개하는 모듈로서, 각 커넥터(502)에는 하나의 브로커(501)만이 연결된다.
- <50> 본 발명의 일 실시예에 따르면, 커넥터(502)와 서비스(503)는 동일한 프로세스 내에 포함되며, 커넥터(502)는 하나의 프로세스 내에 하나만 존재하고, 서비스(503)는 하나의 프로세스 내에 적어도 하나 이상 존재할 수 있다. 즉, 프로세스 별로 하나의 커넥터(502)가 존재하고, 커넥터(502)는 자신이 속하는 프로세스 내에 존재하는 모든 서비스(503)와 브로커(501)간 연결을 매개할 수 있다.
- <51> 본 실시예에 의하면, 커넥터와 프로세스가 1:1로 매칭되어 통신 네트워크 시스템의 통일성을 기할 수 있고, 서로 속하는 프로세스가 상이한 서비스(503)들이 동일한 하나의 커넥터(502)를 통해 동일한 브로커(501)에 연결됨으로써 야기될 수 있는 비효율적 라우팅을 예방할 수 있으며, 데이터 송수신상의 혼란을 피할 수 있다.
- <52> 브로커(501)는 효율적인 메시지 라우팅을 처리하기 위해 라우팅 경로를 설정하거나 커넥터(502)와의 연결 관계를 설정하는 모듈로서, 각 브로커(501)는 완전 메시(full mesh) 형태로 상호 연결된다.
- <53> 본 명세서에서는 이와 같이 브로커(501)가 완전 메시 형태로 상호 연결되어 형성된 네트워크를 MRS(Message Routing Server) 네트워크(500)로 정의한다. 따라서, MRS 네트워크(500)란 통신을 하려는 다양한 시스템간의 효율적인 메시지 송수신을 위한 네트워크 서비스 플랫폼을 의미한다. 커넥터(502)는 이러한 MRS 네트워크(500)를 이용하여 메시지 송수신을 수행하기 위해 사용되는 프로그래밍 인터페이스를 제공하는 모듈로서, 서비스(503)는 커넥터(502)가 제공하는 상기 프로그래밍 인터페이스를 통해 MRS 네트워크(500)를 이용한다.

- <54> 한편, 도 5에서와 같이 본 발명에 따른 통신 네트워크 시스템에서 실제 커넥터(502)와 브로커(501)간의 연결은 하나뿐이며, 각 서비스(503)가 브로커(501)와 통신할 때는 상기 하나의 연결을 통해 순차적으로 통신한다.
- <55> 도 6은 본 발명에 따른 MRS 네트워크를 통해 각종 서버가 연결된 모습의 일례를 도시한 도면이다.
- <56> 도 6에 도시한 것과 같이 게임 스트링 서버, 채널 리스트 서버, 게임 서버, 공지 서버, 관리 서버, 데이터베이스 서버를 비롯한 각종 게임 관련 서버들이 MRS 네트워크(600)를 통해 연결되어 있으며, 각 서버는 완전 메시 형태로 상호 연결된 복수의 브로커 중 어느 하나의 브로커와 하나의 연결만을 유지하게 된다.
- <57> 종래기술에서는 모든 게임 관련 서버들이 메시(mesh)형 구조로 연결되므로 하나의 서버가 연결되는 상대 서버의 수가 많을 수록 전체 연결 수는 기하급수적으로 증가하게 된다. 반면, 본 발명에 따르면, 도 6과 같이 각 서버들은 하나의 브로커와의 연결만을 유지하면 되고, 네트워크가 확장되더라도 중간 연결은 브로커들을 통해서 이루어지므로, 서버가 추가로 연결되더라도 전체 네트워크 관점에서의 연결 수가 크게 증가하지 않는 장점이 있다.
- <58> 도 7은 본 발명에 따른 통신 네트워크 시스템상에서 사용되는 메시지의 구조를 설명하기 위한 도면으로서, 도 7을 참조하여 본 발명에 따른 통신 네트워크 시스템의 서브 시스템인 커넥터와 브로커 및 브로커와 브로커간 송수신되는 메시지의 구조를 설명한다.
- <59> 본 명세서에서는 각 메시지의 필드를 정의하기 위한 자료형으로 옥텟(octet) 표기법을 고수하지 않고, 일반적으로 윈도우즈에서 널리 통용되는 자료형을 기준으로 메시지의 구조를 설명하도록 한다. 이 경우, 윈도우즈 자료형과 옥텟 표기는 "BYTE : octet(8)", "WORD : octet(16)", "DWORD : octet(32)"와 같이 매핑된다.
- <60> 본 발명에 따른 통신 네트워크 시스템상에서 사용되는 메시지는 크게 MRMSGHeader와 MRCMPHeader의 2가지로 구분되며, 각 메시지는 공통적 필드인 MRHeader와 각 메시지마다 특화된 필드로 구분된다.
- <61> MRHeader는 본 발명에 따른 통신 네트워크 시스템상에서 상호 교환되는 모든 메시지가 공통적으로 가져야 하는 메시지 헤더 정보이다. MRHeader 메시지는 단독으로 사용될 수 없고, 프로토콜 타입 필드에서 기술한 값과 추가적인 메시지 정보를 MRHeader 뒤에 추가하여 송수신해야 한다.

**표 1**

<62>

타입	명칭	설명
WORD	버전	메시지의 버전 정보를 표기한다. 상위 1byte에 메이저 버전(major version)을 표기하고, 하위 1byte에 마이너 버전(minor version)을 표기한다.
BBYTE	프로토콜 타입	이 값은 PT_MESSAGE, PT_MRCMP 2개의 값을 가질 수 있다. PT_MESSAGE는 MRHeader 뒤에 따라오는 메시지가 MRMSGHeader 구조임을 규정하며, PT_MRCMP는 MRHeader 뒤에 따라오는 메시지가 MRCMPHeader 구조임을 규정한다.
BBYTE	TTL	Time to live이다. 본 발명에 따른 통신 네트워크상에서 송수신되는 메시지는 MRS 네트워크의 브로커가 라우팅을 처리할 때 마다 TTL 값이 1씩 감소하게 되며, 0이 될 경우에는 더 이상 라우팅하지 않고 송신 에러를 발생시킨다.
BDWORD	길이	MRHeader 및 뒤에 따라오는 MRMSGHeader 또는 MRCMPHeader를 포함하는 전체 메시지의 길이를 의미한다.

- <63> MRMSGHeader의 구조로 전달되는 메시지는 MRS 네트워크에 연결된 다수의 서비스간에 서비스가 지정한 페이로드(payload) 값을 전달하기 위한 구조를 갖고 있다.
- <64> 여기에는 메시지를 송신하는 송신자의 주소인 소스 어드레스(Source address)와 메시지를 수신하는 수신자의 주소인 목적지 어드레스(Destination Address)가 기술되며, MRS 네트워크는 목적지 어드레스의 형태와 연결 정보를 기반으로 라우팅을 시도한다. 각 세부 필드에 대한 설명은 아래 표와 같다.

**표 2**

<65>

타입	명칭	설명
MRHeader(8bytes)	Not defined	앞서 기술한 MRHeader의 구조이다.



MRADDRESS(16bytes)	소스 어드레스	메시지 전송자의 주소이다. 한편, MRADDRESS는 MRMSGHeader의 소스 어드레스와 목적지 어드레스 필드를 정의하기 위한 자료형이다.
MRADDRESS(16bytes)	목적지 어드레스	메시지 수신자의 주소이다.
Not defined	Not defined	사용자 정의 메시지이다. 해당 메시지의 MRHeader 구조 내의 길이는 MRHeader로부터 사용자가 정의한 페이로드까지의 전체 길이를 포함해야 한다.

<66> MRMPHeader의 구조로 전달되는 메시지는 본 발명에 따른 통신 네트워크 시스템의 서버 시스템인 커넥터와 브로커, 브로커와 브로커간 신호를 송수신하기 위해 정의된 메시지이다. 각 세부 필드에 대한 설명은 아래 표와 같다.

**표 3**

타입	명칭	설명
MRHeader(8bytes)	Not defined	앞서 기술한 MRHeader 구조이다.
WORD	Control code	송수신하고자 하는 신호 명령(signal command)이다. 상위 1byte는 신호의 역할 분류를 수행하고, 하위 1byte는 해당 역할 분류 내에서의 구분자 역할을 수행한다.
Not defined	Not defined	Control code와 관련된 가변 길이 추가 필드이다. 이 필드는 Control code의 내용에 따라 서로 상이한 형태로 다양하게 변경될 수 있다.

<68> 이하에서는 본 발명에서 채용한 주소 체계에 대하여 설명한다.

<69> 본 발명에서는 MRS 네트워크를 이용하여 데이터를 송수신하기 위해서 소스 어드레스 및 목적지 어드레스를 지칭할 때 IP 어드레스가 아니라 본 발명에 따른 새로운 주소 체계를 사용한다. MRS 네트워크에서는 특정 서비스를 식별하기 위해 유니캐스트(unicast), 애니캐스트(anycast), 멀티캐스트(multicast)의 3가지 주소 유형을 지원하며, 이 주소는 Cast 타입과 주소 유형별 어드레스로 이루어지며 16bytes의 길이를 가진다.

<70> 온라인 게임의 경우 다수의 서비스 인스턴스(instance)들이 상호 연동하여 동작하고 서비스 인스턴스 간 통신도 많을 수 밖에 없는데, 종래에는 이러한 서비스 인스턴스에 대하여 네트워크 주소를 할당한 예가 없는 바, 동일한 네트워크 주소를 가지는 하나의 프로세스가 많은 서비스 인스턴스를 포함하여 많은 수의 서비스 인스턴스 통신을 처리해야 하는 문제점이 있었다.

<71> 이에, 본 발명에서는 새로운 주소 체계를 도입하여 개별 서비스 인스턴스마다 주소를 할당함으로써, 서비스 인스턴스 간 통신을 효율적으로 처리하도록 하였다. 이 경우, 기존처럼 게임 서버에서 일일이 많은 게임방 하나 하나로 메시지를 분배하지 않아도 되므로, 관리 비용이 현저히 줄어드는 효과가 발생한다.

<72> 또한, 본 발명에서는 유니캐스트 외에도 애니캐스트와 멀티캐스트의 3가지 주소 유형을 지원하며, 따라서 애니캐스트 또는 멀티캐스트를 적용하여 서로 다른 물리적 서버에서 동작하는 인스턴스들을 하나의 애니캐스트 또는 멀티캐스트 그룹으로 묶을 수 있다.

<73> 도 8은 본 발명에 따른 주소 체계를 설명하기 위한 도면이다.

<74> 도 8에서 Cast 타입은 주소의 유형을 나타내며, CT\_UNICAST, CT\_MULTICAST 및 CT\_ANYCAST 중 하나의 값을 가진다.

<75> 유니캐스트(Unicast) 어드레스는 MRS 네트워크를 이용하는 모든 서비스를 유일하게 구분할 수 있는 주소이다. 유니캐스트 어드레스는 전체 네트워크에서 특정 서비스가 구동되고 있는 서버를 구분하는 서버 명칭(server name)과 동일 서버 내에서 해당 서비스를 유일하게 구분하는 인스턴스 ID로 이루어진다.

<76> 서버 명칭은 MRS 네트워크를 이용하는 서비스가 구동되고 있는 컴퓨터 하드웨어인 서버에 대한 구분을 나타내는 11bytes의 유일한 값을 의미하며, 전체 네트워크에서 서비스가 구동되고 있는 서버를 구분하는 11bytes의 유일한 값으로 컴퓨터 이름을 사용할 수 있다.

<77> 인스턴스 ID는 동일한 서버 내에서 서비스를 유일하게 구분하는 식별자이다. 예를 들어, 1-65535 값은 고정된 유니캐스트 어드레스가 필요한 경우를 위해 예약되어 있고, 65536 이후 값에 대해서는 서버 내에서 유일하게 동

적으로 할당하여 사용할 수 있도록 할 수 있다.

<78> 이러한 유니캐스트 어드레스는 아래 표와 같이 정리할 수 있다.

표 4

<79>

타입	명칭	설명
BYTE	Cast 타입	유니캐스트 어드레스의 경우 항상 CT_UNICAST이다.
DWORD	인스턴스 ID	해당 컴퓨터에서 다수의 어드레스가 필요할 경우 인스턴스 넘버를 이용하여 각각을 구분한다. 인스턴스 ID는 0x0000~0xffff 까지는 예약된 번지로 사용될 수 있다.
BYTE(x11)	서버 명칭	해당 컴퓨터를 구분해 낼 수 있는 식별자이다. 서버 명칭에는 일반적으로 로컬 컴퓨터의 NetBIOS Name을 사용할 수 있다. 그러므로 MRS 네트워크를 사용하는 모든 컴퓨터는 다른 컴퓨터와 구분될 수 있는 컴퓨터 이름을 가져야 하며, 그 길이는 11 bytes 이내이다.

<80> 멀티캐스트(Multicast) 어드레스와 애니캐스트(Anycast) 어드레스는 단순히 15bytes 길이의 값을 사용하므로, 서비스 사이에서 자유롭게 설정하여 사용할 수 있다. 이 값은 전체 네트워크상에서 유일해야 하며, 사전에 공지가 되어야 한다.

<81> 멀티캐스트/애니캐스트 어드레스는 아래 표와 같이 정리할 수 있다.

표 5

<82>

타입	명칭	설명
BYTE	Cast 타입	멀티캐스트 어드레스의 경우 CT_MULTICAST로 정의되고, 애니캐스트 어드레스의 경우 CT_ANICAST로 정의된다.
BYTE(x15)	서비스 명칭	해당 서비스를 구분할 수 있는 식별자이다. 멀티캐스트 및 애니캐스트 어드레스는 컴퓨터 또는 서비스를 지칭하는 것이 아닌 가상의 주소 체계이다. 또한, 이러한 서비스 명칭은 MRS 네트워크 전체를 통해서 유일하게 식별되는 값이다.

<83> 본 발명에 따른 MRS 네트워크를 사용하는 경우로서, (1) 서비스가 다른 서비스에서 제공되는 기능을 이용하기 위해 MRS 네트워크에 접속하는 경우와 (2) 서비스가 다른 서비스로부터 요청된 사항을 처리하는 기능을 제공하기 위해 MRS 네트워크에 접속하는 경우가 있을 수 있다.

<84> 서비스가 다른 서비스에서 제공되는 기능을 이용하기 위해 MRS 네트워크에 접속하는 상기 (1)의 경우에 대한 일례로서, 사용자 로그인 정보를 확인하기 위해 로그인 서버에 요청을 보내고 응답을 받는 서비스가 있을 수 있다. 이렇게 다른 서비스에서 제공되는 기능을 이용하기 위한 서비스는 MRS 네트워크와 서비스를 매개하는 모듈인 커넥터를 통해 다음과 같이 동작한다.

<85> 먼저, 프로세스의 시작 시 서비스와 커넥터간 데이터 송수신을 위해 프로그래밍 인터페이스를 초기화하고, MRS 네트워크에 서비스의 유니캐스트 어드레스를 등록한 후, 다른 서비스에 요청 메시지를 송신하고 응답 메시지를 수신한다. 그리고, 다른 서비스에서 제공되는 기능을 모두 이용한 경우에는 MRS 네트워크로부터 서비스의 유니캐스트 어드레스를 해제하고, 프로세스 종료 시 상기 프로그래밍 인터페이스를 종료한다.

<86> 서비스가 다른 서비스로부터 요청된 사항을 처리하는 기능을 제공하기 위해 MRS 네트워크에 접속하는 상기 (2)의 경우에 대한 일례로서, 데이터베이스 조회 기능을 제공하는 서비스, 사용자의 로그인 정보 또는 위치 정보를 제공하는 서비스가 있을 수 있다. 이렇게 다른 서비스에 특정 기능을 제공하기 위한 서비스는 MRS 네트워크와 서비스를 매개하는 모듈인 커넥터를 통해 다음과 같이 동작한다.

<87> 먼저, 프로세스의 시작 시 서비스와 커넥터간 데이터 송수신을 위해 프로그래밍 인터페이스를 초기화하고, MRS 네트워크에 서비스의 유니캐스트 어드레스를 등록한 후, 제공할 기능에 대한 애니캐스트 또는 멀티캐스트 어드레스에 가입(join)하며, 다른 서비스로부터 요청 메시지를 수신하고 응답 메시지를 송신한다. 그리고, 다른 서비스에 특정 기능을 모두 제공한 경우에는 가입된 애니캐스트 또는 멀티캐스트 어드레스로부터 탈퇴(leave)하고, MRS 네트워크로부터 서비스의 유니캐스트 어드레스를 해제하며, 프로세스 종료 시 상기 프로그래밍 인터페이스를 종료한다.

- <88> 이와 같이, 본 발명에서 커넥터는 서비스의 유니캐스트 어드레스를 MRS 네트워크에 등록하고, 상기 서비스를 제공할 기능에 따라 애니캐스트 또는 멀티캐스트 어드레스에 가입시키며, 서비스 종료 시에는 상기 가입된 애니캐스트 또는 멀티캐스트 어드레스로부터 탈퇴시킬 수 있다.
- <89> 이하에서는 본 발명에 따른 통신 네트워크 시스템의 서브시스템(subsystem)인 서비스, 커넥터 및 브로커의 동작 과정 및 각 브로커 간 로드를 조절하는 방법에 대하여 도 9 내지 도 15를 참조하여 설명한다.
- <90> 도 9는 본 발명의 일실시예에 따른 서비스 등록 과정을 도시한 흐름도로서, 본 발명에 따른 통신 네트워크 시스템에서 특정 서비스를 제공하거나 서비스를 사용하고자 하는 시스템은 자신을 커넥터를 통해 MRS 네트워크에 등록해야 한다.
- <91> 본 발명의 일실시예에 따르면, 각 브로커에는 자신과 연결 가능한 커넥터의 최대 개수를 지정하는 로드 한계가 설정되어 있으며, 커넥터로부터 연결 요청 메시지를 수신한 브로커는 자신에게 현재 연결된 커넥터의 수에 해당하는 현재 로드와 자신의 로드 한계를 고려하여 커넥터와의 연결 여부를 판단하도록 동작한다.
- <92> 커넥터는 완전 메시 형태로 상호 연결된 각 브로커의 네트워크 주소가 기록된 브로커 리스트를 유지하며, 상기 네트워크 주소는 각 브로커의 IP 주소일 수 있다.
- <93> 서비스는 자신을 MRS 네트워크에 등록하기 위해 커넥터에게 서비스 등록 메시지를 전송하고(S901), 커넥터는 상기 서비스 등록 메시지를 수신하고 이를 분석하여 이미 등록된 서비스인지 여부를 검사하고(S902), 이미 등록된 서비스가 아닌 경우에는 상기 서비스 등록 메시지의 전송을 위해 자신과 연결될 브로커를 선택하게 된다. 이때, 커넥터는 자신이 유지하고 있는 브로커 리스트 중 하나의 브로커를 임의로 선택하여 그 브로커로 연결 요청 메시지를 전송할 수 있다.
- <94> 커넥터로부터 선택된 브로커는 상기 연결 요청 메시지를 수신하고, 자신에게 현재 연결된 커넥터의 수(즉, 현재 로드)와 자신과 연결 가능한 커넥터의 최대 개수(즉, 로드 한계)를 고려하여 상기 커넥터와의 연결 여부를 판단한다. 예를 들어, 상기 선택된 브로커는 상기 커넥터와의 연결이 설정될 시 자신에게 연결되는 커넥터의 총 개수가 자신의 로드 한계를 초과할 경우 커넥터로부터의 연결 요청을 거부하도록 동작할 수 있다.
- <95> 한편, 연결 요청이 거부된 커넥터는 자신의 브로커 리스트 중 상기 선택된 브로커를 제외한 다른 브로커를 임의로 선택하여 그 브로커로 상기 연결 요청 메시지를 재차 전송할 수 있다.
- <96> 만일, 브로커가 자신의 현재 로드와 로드 한계를 고려하여 커넥터와의 연결이 가능하다고 판단하고, 그에 따라 커넥터의 연결 요청을 승인할 경우에는 양자 간 연결이 설정되며, 커넥터는 서비스로부터 수신한 서비스 등록 메시지를 위와 같이 연결이 설정된 브로커로 전송하게 된다(S903).
- <97> 본 실시예에 따르면, 하나의 브로커에 연결되는 커넥터의 수가 적절한 정도로 유지되고, 전체 네트워크에서 각 브로커에 연결된 커넥터의 수가 균형을 이루게 되어 각 브로커 간 로드 밸런스가 자연스럽게 조절된다.
- <98> 브로커는 상기 서비스 등록 메시지를 전송한 서비스의 라우팅 정보를 자신의 라우팅 테이블에 추가하여 라우팅 테이블을 갱신하고(S904), 상기 라우팅 정보를 다른 브로커에게 전송하며(S905), 이를 수신한 다른 브로커는 상기 라우팅 정보를 이용하여 자신의 라우팅 테이블을 갱신한다(S906).
- <99> 이와 같이, 본 발명에서 브로커는 자신의 라우팅 테이블이 갱신되었을 경우, 갱신된 정보를 MRS 네트워크를 구성하는 다른 브로커들에게 전송하여, 다른 브로커들이 항상 최신의 라우팅 정보를 유지할 수 있도록 동작한다.
- <100> 도 10은 본 발명의 일실시예에 따른 서비스 해제 과정을 도시한 흐름도로서, MRS 네트워크에 자신을 등록한 시스템은 자신을 MRS 네트워크로부터 제거하기 위해서 커넥터를 통해 서비스 해제 과정을 수행하며, 서비스 해제 과정을 수행한 이후에는 MRS 네트워크를 통한 모든 서비스 요청은 차단된다.
- <101> 서비스는 커넥터에게 서비스 해제 메시지를 전송하고(S1001), 커넥터는 상기 서비스 해제 메시지를 분석하여 최종 해제인지 여부를 검사하고(S1002), 최종 해제인 경우 상기 서비스 해제 메시지를 브로커에게 전달한다(S1003).
- <102> 브로커는 상기 서비스 해제 메시지를 전송한 서비스의 라우팅 정보를 자신의 라우팅 테이블에서 삭제하여 라우팅 테이블을 갱신하고(S1004), 상기 라우팅 정보의 삭제를 다른 브로커에게 통보하며(S1005), 이를 수신한 다른 브로커는 상기 라우팅 정보의 삭제를 감지하여 자신의 라우팅 테이블을 갱신한다(S1006).
- <103> 도 11은 본 발명의 일실시예에 따른 메시지 송수신 과정을 도시한 흐름도로서, 본 발명에서 서비스를 이용하기

위해 서비스로 메시지를 전송하거나, 또는 서비스 요청을 위해 메시지를 전달 받는 과정을 도시한 것이다.

- <104> 특정 서비스를 이용하고자 하는 서비스는 먼저 연결 메시지를 커넥터에게 전달하여 가상의 연결을 설정하며(S1101), 이 때 이용하고자 하는 서비스의 어드레스를 규정한다.
- <105> 메시지 송신 시에, 서비스는 커넥터의 프로그래밍 인터페이스를 이용하여 메시지 전송을 수행하고(S1102), 커넥터는 서비스로부터 전달된 메시지에 MRS 네트워크에서 사용되는 메시지 라우팅 프로토콜(Message Routing Protocol: MRP)에 따른 MRP Header를 추가한 후, 자신과 연결된 브로커에게 해당 메시지를 전송하며(S1103), 브로커는 커넥터로부터 수신한 메시지의 목적지를 검색하고(S1104), 해당 브로커 또는 커넥터에게 메시지를 전송한다(S1105).
- <106> 메시지 수신 시에, 브로커는 다른 브로커로부터 메시지를 수신하고(S1106), 수신한 메시지의 목적지를 검색하고(S1107), 해당 커넥터에게 메시지를 전송하며(S1108), 커넥터는 전송된 메시지의 MRP Header를 제거하고 브로커로부터 전송된 메시지의 목적지 어드레스와 인스턴스를 검색하며(S1109), 해당 서비스에게 메시지를 전송하여 해당 서비스가 메시지를 수신하게 된다(S1110).
- <107> 본 발명에 따른 통신 네트워크 시스템에서는 상기와 같은 과정을 통해 커넥터를 통하여 브로커로 전달되는 메시지들이 다른 브로커나 커넥터로 투명하게 송수신되고, 특정 서비스가 전달한 메시지는 라우팅 경로를 통해 경로 상에 존재하는 브로커들 사이에 투명하게 송수신되도록 할 수 있다.
- <108> 도 12는 본 발명의 일실시예에 있어서, 브로커가 MRS 네트워크에 새로 추가되는 경우 진행되는 과정을 도시한 흐름도이다.
- <109> 새로 추가되는 브로커는 우선 다른 브로커들의 연결을 기다리며 연결 대기 상태에 있다가(S1201), 이미 실행되고 있는 다른 브로커들에 연결한다(S1202). 새로 추가되는 브로커와 연결된 브로커들은 자신과 연결된 다른 브로커가 기록된 브로커 리스트(broker list)에 새로 추가된 브로커를 추가하여 자신의 브로커 리스트를 갱신한다(S1203). 이와 동일한 방식으로, 새로 추가되는 브로커는 이미 실행되고 있는 또 다른 브로커들에 연결하고(S1204), 그에 따라 새로 연결된 브로커들의 브로커 리스트도 갱신되며(S1205), 이와 같이 MRS 네트워크상에 실행되고 있는 모든 브로커들의 브로커 리스트가 갱신될 수 있다.
- <110> 새로 추가되는 브로커는 이미 실행된 브로커들에게 모두 연결을 한 후, 그 중에서 하나의 브로커를 임의로(randomly) 선택하고(S1206), 그 선택된 브로커에게 라우팅 테이블을 요청하고(S1207) 이를 수신하여(S1208) 자신의 라우팅 테이블에 반영하며(S1209), 자신이 커넥터와 연결될 준비가 되었음을 다른 브로커들에게 통보한다(S1210).
- <111> 상기 준비 사실을 통보 받은 브로커는 자신에게 연결된 커넥터들 중에서 새로운 브로커에게 옮겨갈 재설정 커넥터의 수를 계산하고, 그 수만큼의 커넥터들을 선택한다(S1211). 또한, 상기 준비 사실을 통보 받은 브로커는 자신에게 연결된 모든 커넥터들에게 새로운 브로커가 연결되었음을 통보하며(S1212), 이를 통보 받은 커넥터들은 자신과 연결된 브로커들이 기록된 브로커 리스트에 새로운 브로커를 추가한다(S1213).
- <112> 한편, 상기 선택된 재설정 커넥터들에는 새로운 브로커로 연결을 재설정하라는 메시지가 전달되며(S1214), 이를 전달 받은 커넥터들은 새로운 브로커에 연결하여 자신에게 연결된 서비스들에 대한 서비스 등록 메시지를 연결된 새로운 브로커로 전달한다(S1215).
- <113> 단계(S1210) 내지 단계(S1215)는 MRS 네트워크상에 실행되고 있는 모든 브로커들에 대해 수행될 수 있으며, 이러한 과정을 통해 이미 실행되고 있는 브로커와 연결되어 있던 커넥터들 및 이와 연결된 서비스들의 일정 부분이 MRS 네트워크에 새로 추가된 브로커로 옮겨가게 되어 브로커간 효율적인 로드(load) 분산이 이루어질 수 있다.
- <114> 도 13은 본 발명의 일실시예에 있어서, 브로커가 정상적으로 종료되는 경우에 진행되는 로드 조절 방법의 과정을 도시한 흐름도이다.
- <115> 본 발명에 따른 통신 네트워크 시스템으로부터 정상적으로 종료되는 제1 브로커는 자신에게 더 이상 커넥터가 연결되지 않도록 설정하고(S1301), 자신이 종료될 것이라는 종료 예정 정보를 다른 브로커들에게 전송하며(S1302), 상기 제1 브로커와 상기 종료 예정 정보를 수신한 다른 브로커들은 자신과 연결된 커넥터들에게 상기 종료 예정 정보를 전송하여 제1 브로커가 정상적으로 종료된다는 사실을 통보한다(S1303).
- <116> 상기 제1 브로커는 자신과 완전 메시 형태로 상호 연결된 다른 브로커들 중 하나인 제2 브로커를 선택하고, 상



기 제1 브로커에 연결된 커넥터에게 상기 제2 브로커로 연결을 재설정할 것을 요청하는 연결 재설정 메시지를 전송한다(S1304).

- <117> 상기 제1 브로커에 연결된 커넥터는 상기 연결 재설정 메시지를 수신하고, 상기 제2 브로커와 연결을 설정하여 상기 커넥터에 연결된 서비스를 등록하기 위한 서비스 등록 메시지를 상기 제2 브로커로 전송한다(S1305).
- <118> 한편, 본 발명의 일실시예에 따르면 MRS 네트워크에서 완전 메시 형태로 연결된 모든 브로커 상호 간에는 브로커별 현재 연결된 커넥터의 수(즉, 현재 로드)와 연결 가능한 커넥터의 최대 개수(즉, 로드 한계)에 대한 정보를 포함하는 브로커 상태 메시지가 교환된다.
- <119> 즉, 브로커 상태 메시지는 브로커 상호 간에 교환되는 각 브로커의 현재 연결 상태 및 최대 용량(Capacity)에 대한 정보를 포함하는 메시지로써, 상기 제2 브로커를 선택함에 있어서 상기 제1 브로커는 브로커 상태 메시지에 포함된 브로커별 현재 로드와 로드 한계에 기초하여 상기 제2 브로커를 선택할 수 있다.
- <120> 본 실시예에 따르면, 정상적으로 종료되는 브로커에 연결된 커넥터가 동적으로 다른 브로커에 연결되는 과정에 있어서, 커넥터가 로드 한계를 초과하는 브로커로 연결을 요청한 후 거절되어 다른 브로커로 재차 연결을 요청해야 하는 불필요한 절차가 생략될 수 있다.
- <121> 상기 제1 브로커의 정상 종료로 인해 커넥터가 새로이 연결되는 상기 제2 브로커는 자신의 라우팅 정보를 갱신하고, 상기 제1 브로커 및 다른 브로커들에게 상기 갱신된 라우팅 정보를 전송한다(S1306).
- <122> 이와 같이, 본 발명에서 브로커는 자신의 라우팅 정보가 갱신되었을 경우, 갱신된 정보를 MRS 네트워크를 구성하는 다른 브로커들에게 전송하여, 다른 브로커들이 항상 최신의 라우팅 정보를 유지할 수 있도록 동작한다.
- <123> 정상적으로 종료되는 상기 제1 브로커는 상기 제2 브로커로부터 상기 갱신된 라우팅 정보를 수신하고, 상기 갱신된 라우팅 정보를 이용하여 자신의 라우팅 테이블을 갱신하며, 상기 갱신된 대상(즉, 상기 제1 브로커에 연결되어 있던 커넥터들)과 연관된 라우팅 정보가 자신의 라우팅 테이블에 존재하는지 여부를 판단하고(S1307), 상기 판단 결과 상기 갱신된 대상과 연관된 라우팅 정보가 존재하지 않는 경우에는 그 연결을 최종적으로 종료한다(S1308).
- <124> 한편, 정상적으로 종료되는 상기 제1 브로커는 자신과 연결된 커넥터에게 상기 제2 브로커로 연결을 재설정할 것을 요청하는 연결 재설정 메시지의 전송 시점으로부터 소정의 시간이 경과하였는지 여부를 검사하고(S1309), 상기 소정의 시간이 경과한 후에도 자신에게 연결된 커넥터가 존재하는 경우, 모든 커넥터와의 연결을 종료하고, 자기 자신도 종료하도록 동작한다(S1310).
- <125> 본 발명에서는 위와 같은 과정을 통해 브로커의 정상 종료 시에 각 커넥터에 연결된 서비스들이 다른 브로커로 동적으로 이동하여 브로커 간 로드 밸런스가 자동적으로 조절될 수 있도록 한다.
- <126> 도 14는 본 발명의 일실시예에 있어서, 브로커의 비정상적인 종료를 다른 브로커가 감지하는 경우에 진행되는 로드 조절 방법의 과정을 도시한 흐름도이다.
- <127> 본 발명에 따른 통신 네트워크 시스템에서 시스템 장애 등으로 인해 제1 브로커가 비정상적으로 종료되면(S1401), MRS 네트워크에서 상기 제1 브로커와 완전 메시 형태로 상호 연결된 다른 브로커 중 하나인 제2 브로커는 상기 제1 브로커의 비정상적인 종료를 감지하고(S1402), 그에 따라 자신의 로드 한계, 즉 최대 용량을 상향 조정한다(S1403).
- <128> 예를 들어, 브로커 하나에 연결될 수 있는 커넥터의 최대 수가 50개가 원래의 로드 한계라면, 다른 브로커가 비정상적으로 종료하는 것이 감지될 경우에는 브로커 하나에 연결될 수 있는 커넥터의 최대 수를 60개로 재조정하여, 비정상적으로 종료되는 브로커에 연결된 커넥터들이 신속하게 나머지 브로커들로 옮겨서 연결되도록 동작할 수 있다. 이 경우, 서비스 연결이 중단되는 시간이 획기적으로 단축될 수 있다.
- <129> 한편, 본 발명의 일실시예에 따르면 MRS 네트워크에서 완전 메시 형태로 연결된 모든 브로커 상호 간에는 브로커별 현재 상태에 대한 정보를 포함하는 브로커 상태 메시지가 교환된다.
- <130> 본 실시예에서 이러한 브로커 상태 메시지는 각 브로커가 다른 브로커가 정상 상태에 있는지 또는 비정상 상태에 있는지 여부를 판단할 수 있는 기준으로서 활용될 수 있으며, 본 발명에 따른 통신 네트워크 시스템으로부터 비정상적으로 종료되는 상기 제1 브로커를 상기 제2 브로커가 감지함에 있어서, 상기 제2 브로커는 상기 브로커 상태 메시지가 정상적으로 수신되지 않는 제1 브로커를 비정상 종료되는 브로커로서 감지할 수 있다.

- <131> 또한, 상기 제1 브로커의 비정상적인 종료를 감지한 상기 제2 브로커는 상기 제1 브로커에 연결된 커넥터 중 적어도 일부 커넥터와 연결을 설정하고, 그에 따라 상기 일부 커넥터와 연결된 서비스들이 상기 제2 브로커에 등록되도록 동작한다.
- <132> 본 발명의 일실시예에 따르면, 상기 제2 브로커는 상기 일부 커넥터와 연결을 설정함에 있어서, 상기 제1 브로커와 연결된 커넥터의 수를 상기 제1 브로커를 제외한 MRS 네트워크상의 각 브로커의 총 개수로 나눈 값에 따라 상기 일부 커넥터의 수를 결정할 수 있다. 예를 들어, 비정상적으로 종료되는 상기 제1 브로커를 제외하고 MRS 네트워크상에 2개의 브로커가 존재하며 상기 제1 브로커와 연결된 커넥터의 수가 N개라고 하면, 상기 제2 브로커에는 "N/2"개의 커넥터가 연결될 수 있는 것이다.
- <133> 본 실시예에 의하면, 어느 브로커가 시스템 장애 등으로 인해 비정상적으로 종료하더라도 하나의 브로커에 연결되는 커넥터의 수가 적절한 정도로 유지되고, 전체 네트워크에서 각 브로커에 연결된 커넥터의 수가 균형을 이루게 되어 각 브로커 간 로드 밸런스가 자연스럽게 조절될 수 있다.
- <134> 한편, 상기 제2 브로커는 상기와 같은 연결 설정에 따라 자신의 라우팅 정보를 갱신하고, 상기 갱신된 라우팅 정보를 MRS 네트워크상의 다른 브로커들에게도 전송하여 모든 브로커들이 자신의 라우팅 테이블을 최신으로 갱신하도록 할 수 있다(S1404).
- <135> 또한, MRS 네트워크상의 모든 브로커에 연결된 커넥터들에게는 상기 제1 브로커의 종료 사실을 알리는 종료 정보가 전송되며, 이에 따라 커넥터들은 자신의 브로커 리스트를 갱신할 수 있게 된다(S1405).
- <136> 본 발명에서는 위와 같은 과정을 통해 브로커의 비정상 종료 시에도 각 커넥터에 연결된 서비스들이 다른 브로커로 동적으로 이동하여 브로커 간 로드 밸런스가 자동적으로 조절될 수 있도록 한다.
- <137> 도 15는 본 발명의 일실시예에 있어서, 브로커의 비정상적인 종료를 커넥터가 감지하는 경우에 진행되는 로드 조절 방법의 과정을 도시한 흐름도이다.
- <138> 커넥터는 완전 메시 형태로 상호 연결된 각 브로커의 네트워크 주소가 기록된 브로커 리스트를 유지하며, 상기 네트워크 주소는 각 브로커의 IP 주소일 수 있다.
- <139> 본 발명에 따른 통신 네트워크 시스템으로부터 제1 브로커가 비정상적으로 종료되면(S1501), 상기 제1 브로커와 연결된 커넥터는 자신과 연결된 상기 제1 브로커가 비정상적으로 종료된 사실을 감지하고(S1502), 자신의 브로커 리스트로부터 상기 제1 브로커의 네트워크 주소를 삭제하여 브로커 리스트를 갱신한다(S1503).
- <140> 비정상적으로 종료된 상기 제1 브로커를 브로커 리스트로부터 삭제한 커넥터는 자신의 갱신된 브로커 리스트 중 임의의 제2 브로커를 선택하고(S1504), 상기 제2 브로커와 연결을 설정하여(S1505), 자신에게 연결된 모든 서비스들을 등록하기 위한 서비스 등록 메시지를 상기 제2 브로커로 전송함으로써 서비스 등록을 수행할 수 있다(S1506).
- <141> 한편, 상기와 같이 서비스들이 추가로 등록된 제2 브로커는 자신의 라우팅 정보를 갱신하고, 상기 갱신된 라우팅 정보를 MRS 네트워크상의 다른 브로커들에게 전송하여 모든 브로커들이 최신의 라우팅 정보를 유지할 수 있도록 한다(S1507).
- <142> 본 발명에서는 위와 같은 과정을 통해 브로커의 비정상 종료 시에 브로커뿐만 아니라 비정상적으로 종료되는 브로커에 연결된 커넥터도 이를 감지하여, 커넥터에 연결된 서비스들이 다른 브로커로 동적으로 이동하여 브로커 간 로드 밸런스가 자동적으로 조절될 수 있도록 한다.
- <143> 이하에서는 본 발명에 따른 통신 네트워크 시스템의 서브 시스템인 커넥터의 구조 및 각 구성요소의 기능에 대하여 설명한다.
- <144> 도 16은 본 발명에 따른 커넥터의 구조를 도시한 블록도이다.
- <145> 도 16에 도시한 것과 같이, 커넥터(1600)는 API(1601), 메시지 큐 관리 모듈(1602), 서비스 풀 관리 모듈(1603), 서비스 관리 모듈(1604) 및 연결 관리 모듈(1605)를 포함하여 구성된다.
- <146> API(1601)는 커넥터(1600)가 제공하는 기능을 커넥터(1600)에 연결된 서비스에게 노출하는 어플리케이션 프로그래밍 인터페이스(Application Programming Interface)이며, 본 발명에 따른 MRS 네트워크를 이용하고자 하는 시스템은 데이터를 송수신하기 위해 API(1601)를 사용한다.
- <147> 도 17은 커넥터와 서비스의 프로세스에 대한 관계와 API(1601)를 설명하기 위한 도면이다.



- <148> 도 17에 도시한 것과 같이, 본 발명의 실시시에 따르면, 커넥터와 서비스는 동일한 프로세스 내에 포함되며, 커넥터는 하나의 프로세스 내에 하나만 존재하고, 서비스는 하나의 프로세스 내에 적어도 하나 이상 존재할 수 있다. 즉, 프로세스 별로 하나의 커넥터가 존재하고, 커넥터는 자신이 속하는 프로세스 내에 존재하는 모든 서비스와 MRS 네트워크를 구성하는 브로커간의 연결을 매개할 수 있다.
- <149> 본 실시예에 의하면, 커넥터와 프로세스가 1:1로 매칭되어 통신 네트워크 시스템의 통일성을 기할 수 있고, 서로 속하는 프로세스가 상이한 서비스들이 동일한 하나의 커넥터를 통해 동일한 브로커에 연결됨으로써 야기될 수 있는 비효율적 라우팅을 예방할 수 있으며, 데이터 송수신상의 혼란을 피할 수 있다.
- <150> 한편, 도 17에 도시한 것과 같이, 하나의 프로세스 내에 존재하는 하나 이상의 서비스들은 하나의 커넥터(1600)를 통해 MRS 네트워크에 연결될 수 있는데, 여기서 서비스와 커넥터(1600)간의 연결 및 데이터 송수신은 API(1601)를 통해서 이루어진다.
- <151> 즉, MRS 네트워크를 이용하여 데이터를 송수신하기 위해, 서비스는 직접 MRS 네트워크와 연결을 설정하여 데이터를 송수신할 수는 없으며, 커넥터(1600)에서 제공하는 API(1601)를 이용하여 데이터를 송수신하게 된다. 그 결과, API(1601)는 서비스로부터 서비스의 등록 또는 해제 요청, 및 브로커(1610)로의 데이터 전송 요청 등을 수신하게 된다.
- <152> 커넥터(1600)는 이러한 API(1601)를 제공하는 모듈로서, MRS 네트워크를 사용하는 각 프로세스에 적재되어 해당 프로세스에서 생성되는 모든 서비스에 대한 MRS 네트워크로의 메시지 송수신 기능을 담당한다.
- <153> 메시지 큐(queue) 관리 모듈(1602)은 송수신되는 본 발명에 따른 MRS 네트워크상에서 사용되는 메시지 라우팅 프로토콜(Message Routing Protocol: MRP) 패킷을 관리하는 역할을 하며, 여기서 MRP 패킷은 커넥터(1600)와 브로커(1610)간 송수신되는 데이터의 단위를 의미한다.
- <154> 도 18은 이러한 메시지 큐 관리 모듈(1602)의 구조를 도시한 블록도이다.
- <155> 도 18에 도시한 것과 같이, 메시지 큐 관리 모듈(1602)은 송신 큐(1801)와 수신 큐(1802)를 포함할 수 있다.
- <156> 여기서, 송신 큐(1801)는 브로커(1610)로 송신할 MRP 패킷을 관리하고, 수신 큐(1802)는 브로커(1610)로부터 수신된 MRP 패킷을 관리하는 역할을 담당한다.
- <157> 서비스 풀(pool) 관리 모듈(1603)은 서비스를 브로커(1610)에 등록 또는 제거하며 등록된 서비스를 관리하는 역할을 하며, 서비스 관리 모듈(1604)은 상기 등록된 서비스에 대한 정보를 관리하는 역할을 한다.
- <158> 도 19는 이러한 서비스 관리 모듈(1604)의 구조를 도시한 블록도이다.
- <159> 도 19에 도시한 것과 같이, 서비스 관리 모듈(1604)은 수신 메시지 큐(1901), 수신 버퍼 큐(1902) 및 완성 상태 큐(1903)를 포함할 수 있다.
- <160> 여기서, 수신 메시지 큐(1901)는 브로커(1610)로부터 수신된 메시지를 관리하고, 수신 버퍼 큐(1902)는 서비스가 등록한 수신 버퍼를 관리하며, 완성 상태 큐(1903)는 서비스가 요청한 입출력에 대한 완성된 결과를 관리하는 역할을 담당한다.
- <161> 연결 관리 모듈(1605)은 MRP 패킷을 송수신하여 브로커(1610)와의 소켓(socket) 연결을 관리하는 역할을 한다.
- <162> 도 20은 이러한 연결 관리 모듈(1605)의 구조를 도시한 블록도이다.
- <163> 도 20에 도시한 것과 같이, 연결 관리 모듈(1605)은 메시지 전송 모듈(2001)과 연결 제어 모듈(2002)를 포함할 수 있다.
- <164> 여기서, 메시지 전송 모듈(2001)은 메시지 큐 관리 모듈(1602)로부터 송신할 MRP 패킷을 로딩(loading)하여 브로커(1610)로 전송하며, 브로커(1610)로부터 수신된 MRP 패킷을 메시지 큐 관리 모듈(1602)로 전달하는 역할을 담당한다. 한편, 연결 제어 모듈(2002)은 커넥터(1600)와 브로커(1610) 사이의 제어 메시지를 처리하여 양자간 연결을 제어하는 역할을 담당한다.
- <165> 이하에서는 본 발명에 따른 통신 네트워크 시스템의 또 다른 서브 시스템인 브로커의 구조 및 각 구성요소의 기능에 대하여 설명한다.
- <166> 도 21은 브로커의 구조를 도시한 블록도이다.
- <167> 도 21에 도시한 것과 같이, 브로커(2100)는 링크 관리 모듈(2101), 라우팅 정보 관리 모듈(2102), 메시지 분류

모듈(2103), 메시지 라우터(2104), 메시지 디시리얼라이저(2105), 메시지 트랜잭터(2106), 메시지 시리얼라이저(2107) 및 자동 컨피규레이터(2108)를 포함하여 구성될 수 있다.

- <168> 링크 관리 모듈(2101)은 커넥터 또는 다른 브로커와의 연결을 유지 및 관리하고, 상기 커넥터 또는 다른 브로커와 데이터 송수신을 수행하는 역할을 한다.
- <169> 라우팅 정보 관리 모듈(2102)은 브로커(2100)에 등록된 서비스의 라우팅 정보를 유지 및 관리하는 역할을 하며, 브로커(2100)에 연결된 커넥터 또는 다른 브로커와의 연결 정보를 포함하는 연결 풀(pool)을 유지할 수 있다.
- <170> 메시지 분류 모듈(2103)은 링크 관리 모듈(2101)이 수신한 데이터의 형태를 파악하고, 소정의 기준에 따라 상기 수신한 데이터를 단순 메시지와 복잡 메시지로 분류하는 역할을 한다.
- <171> 메시지 라우터(2104)는 메시지 분류 모듈(2103)로부터 단순 메시지를 수신하고, 라우팅 정보 관리 모듈(2102)로부터 상기 단순 메시지와 연관된 목적지의 위치 정보를 획득하여 링크 관리 모듈(2101)에 전달하는 역할을 한다.
- <172> 메시지 디시리얼라이저(2105)(deserializer)는 메시지 분류 모듈(2103)로부터 복잡 메시지를 수신하고, 상기 복잡 메시지를 가공하여 객체(object)화하는 역할을 한다.
- <173> 메시지 트랜잭터(2106)(transactor)는 메시지 디시리얼라이저(2105)로부터 객체를 수신하고, 상기 객체를 이용하여 브로커(2100)를 제어하는 역할을 한다.
- <174> 메시지 시리얼라이저(2107)(serializer)는 메시지 트랜잭터(2106)로부터 상기 객체를 수신하고, 상기 객체를 전송 가능한 선형 데이터로 가공하여 링크 관리 모듈(2101)에 전달하는 역할을 한다.
- <175> 자동 컨피규레이터(2108)(configurator)는 브로커(2100)가 속한 네트워크의 상태를 추적하여 브로커(2100)의 상태를 자동으로 조정하는 역할을 한다.
- <176> 도 22는 링크 관리 모듈(2101)의 구조를 도시한 블록도이다.
- <177> 도 22에 도시한 것과 같이, 링크 관리 모듈(2101)은 OS 소켓 서브 시스템(2201), 링크 어셉터(2202), 데이터 수신 모듈(2203), 데이터 전송 모듈(2204) 및 링크 에이전트(2205)를 포함할 수 있다.
- <178> OS 소켓 서브 시스템(2201)은 브로커(2100)와 연결된 커넥터 또는 다른 브로커와의 데이터 송수신을 위한 인터페이스의 역할을 담당한다.
- <179> 링크 어셉터(2202)(accepter)는 브로커(2100)와 연결된 커넥터 또는 다른 브로커로부터 연결 요청을 OS 소켓 서브 시스템(2201)을 통해 수신하고, 상기 연결 요청에 따른 연결 정보를 라우팅 정보 관리 모듈(2102)이 유지하는 연결 풀에 기록하는 역할을 담당한다.
- <180> 데이터 수신 모듈(2203)은 브로커(2100)와 연결된 커넥터 또는 다른 브로커로부터 데이터를 OS 소켓 서브 시스템(2201)을 통해 수신하고 이를 메시지 분류 모듈(2103)에 전달하는 역할을 담당한다.
- <181> 데이터 전송 모듈(2204)은 메시지 시리얼라이저(2107)에서 가공된 데이터를 수신하여 OS 소켓 서브 시스템(2201)을 통해 브로커(2100)와 연결된 커넥터 또는 다른 브로커로 전송하는 역할을 담당한다.
- <182> 링크 에이전트(2205)(agent)는 데이터 전송 모듈(2204)의 요청에 따라 OS 소켓 서브 시스템(2201)을 통해 브로커(2100)와 연결된 커넥터 또는 다른 브로커로의 연결을 시도하고, 연결이 성공되면 그에 따라 생성된 연결 정보를 이용하여 라우팅 정보 관리 모듈(2102)가 유지하는 연결 풀을 갱신하는 역할을 담당한다.
- <183> 한편, 데이터 전송 모듈(2204)은 상기 연결 풀을 조회하여 전송할 데이터의 목적지와 연관된 커넥터 또는 다른 브로커와의 연결 여부를 확인하고, 연결 설정이 되어 있지 않은 경우에는 링크 에이전트(2205)에게 브로커(2100)와 연결된 커넥터 또는 다른 브로커로의 연결 설정을 요청한 후 데이터 전송을 시도하도록 동작할 수 있다.
- <184> 본 발명에 따른 로드 조절 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광

매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 상기 매체는 프로그램 명령, 데이터 구조 등을 지정하는 신호를 전송하는 반송파를 포함하는 광 또는 금속선, 도파관 등의 전송 매체일 수도 있다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

<185> 이상과 같이 본 발명은 비록 한정된 실시예와 도면에 의해 설명되었으나, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상의 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다.

<186> 그러므로, 본 발명의 범위는 설명된 실시예에 국한되어 정해져서는 아니되며, 후술하는 특허청구범위뿐만 아니라 이 특허청구범위와 균등한 것들에 의해 정해져야 한다.

**발명의 효과**

<187> 본 발명에 따르면, 버스형 네트워크 구조를 통해 서버 간 연결 구조가 단순화된 새로운 통신 네트워크 시스템에서 각 브로커의 현재 로드와 로드 한계에 기초하여 커넥터를 통해 서비스가 등록될 브로커를 선택하고, 브로커의 정상 및 비정상 종료 시에 각 커넥터에 연결된 서비스들이 다른 브로커로 동적으로 이동하여 브로커 간 로드 밸런스가 자동적으로 조절되는 로드 조절 방법이 제공된다.

<188> 본 발명에 따르면, 버스형 네트워크 구조를 통해 서버간 연결 구조가 단순화된 새로운 통신 네트워크 시스템에서 각 서브시스템 간 로드 분산을 효율적으로 조절함으로써, 서비스 확장성 및 성능 확장성을 지원하면서도 유지 및 관리가 용이한 통신 네트워크 시스템이 구축된다.

<189> 본 발명에 따르면, 모든 게임 서버들이 메시 형태로 연결되는 종래기술에 따른 네트워크 구조를 과감히 탈피하고, 서버간 연결 구조를 단순화하여 그 관리가 용이하고 효율적인 서비스 확장이 가능한 새로운 통신 네트워크 구조를 제시된다.

**도면의 간단한 설명**

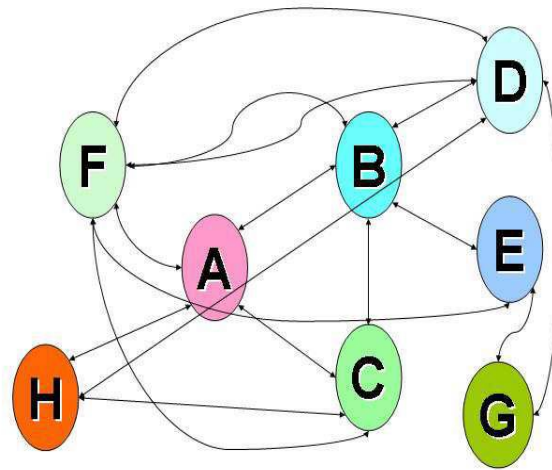
- <1> 도 1은 종래기술에 따른 게임 서버간 네트워크 연결 모습을 도시한 도면이다.
- <2> 도 2는 종래기술에 있어서 게임 서버의 수가 계속적으로 증가함에 따라 발생할 수 있는 게임 서버간 네트워크 연결 모습을 도시한 도면이다.
- <3> 도 3은 본 발명에 따른 버스형 네트워크 구조의 개략적인 모습을 도시한 도면이다.
- <4> 도 4는 본 발명에 따른 버스형 네트워크 구조의 통신 네트워크 시스템의 연결 모습을 개략적으로 도시한 도면이다.
- <5> 도 5는 본 발명에 따른 통신 네트워크 시스템에서 브로커, 커넥터 및 서비스의 연결 모습을 도시한 도면이다.
- <6> 도 6은 본 발명에 따른 MRS 네트워크를 통해 각종 서버가 연결된 모습의 일례를 도시한 도면이다.
- <7> 도 7은 본 발명에 따른 통신 네트워크 시스템상에서 사용되는 메시지의 구조를 설명하기 위한 도면이다.
- <8> 도 8은 본 발명에 따른 주소 체계를 설명하기 위한 도면이다.
- <9> 도 9는 본 발명의 일실시예에 따른 서비스 등록 과정을 도시한 흐름도이다.
- <10> 도 10은 본 발명의 일실시예에 따른 서비스 해제 과정을 도시한 흐름도이다.
- <11> 도 11은 본 발명의 일실시예에 따른 메시지 송수신 과정을 도시한 흐름도이다.
- <12> 도 12는 본 발명의 일실시예에 있어서, 브로커가 MRS 네트워크에 새로 추가되는 경우 진행되는 과정을 도시한 흐름도이다.
- <13> 도 13은 본 발명의 일실시예에 있어서, 브로커가 정상적으로 종료되는 경우에 진행되는 로드 조절 방법의 과정을 도시한 흐름도이다.
- <14> 도 14는 본 발명의 일실시예에 있어서, 브로커의 비정상적인 종료를 다른 브로커가 감지하는 경우에 진행되는

로드 조절 방법의 과정을 도시한 흐름도이다.

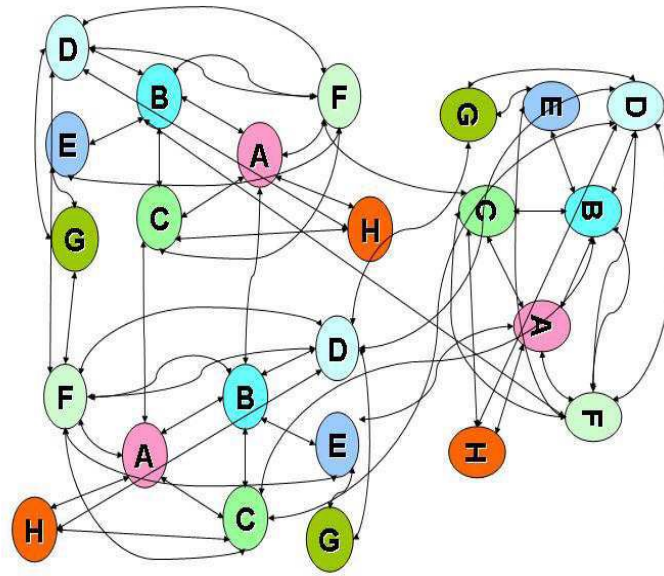
- <15> 도 15는 본 발명의 일실시예에 있어서, 브로커의 비정상적인 종료를 커넥터가 감지하는 경우에 진행되는 로드 조절 방법의 과정을 도시한 흐름도이다.
- <16> 도 16은 본 발명에 따른 커넥터의 구조를 도시한 블록도이다.
- <17> 도 17은 본 발명에 따른 커넥터에서 제공되는 프로그래밍 인터페이스인 API를 설명하기 위한 도면이다.
- <18> 도 18은 본 발명에 따른 커넥터의 일 구성요소인 메시지 큐 관리 모듈의 구조를 도시한 블록도이다.
- <19> 도 19는 본 발명에 따른 커넥터의 일 구성요소인 서비스 관리 모듈의 구조를 도시한 블록도이다.
- <20> 도 20은 본 발명에 따른 커넥터의 일 구성요소인 연결 관리 모듈의 구조를 도시한 블록도이다.
- <21> 도 21은 본 발명에 따른 브로커의 구조를 도시한 블록도이다.
- <22> 도 22는 본 발명에 따른 브로커의 일 구성요소인 링크 관리 모듈의 구조를 도시한 블록도이다.
- <23> <도면의 주요 부분에 대한 부호의 설명>
- <24> 500 : MRS 네트워크
- <25> 501 : 브로커
- <26> 502 : 커넥터
- <27> 503 : 서비스

**도면**

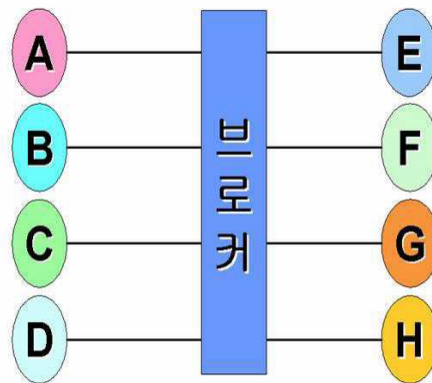
**도면1**



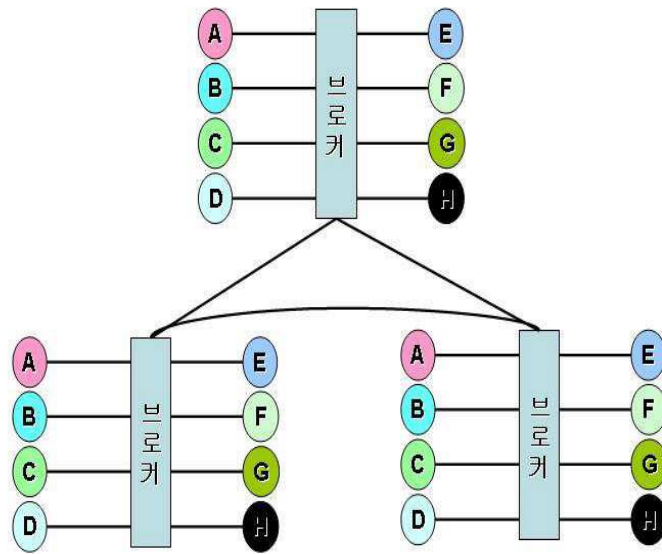
도면2



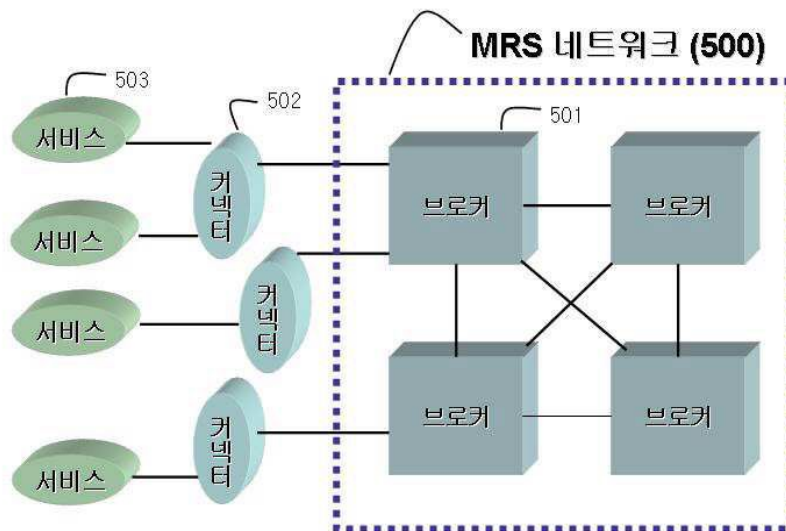
도면3



도면4

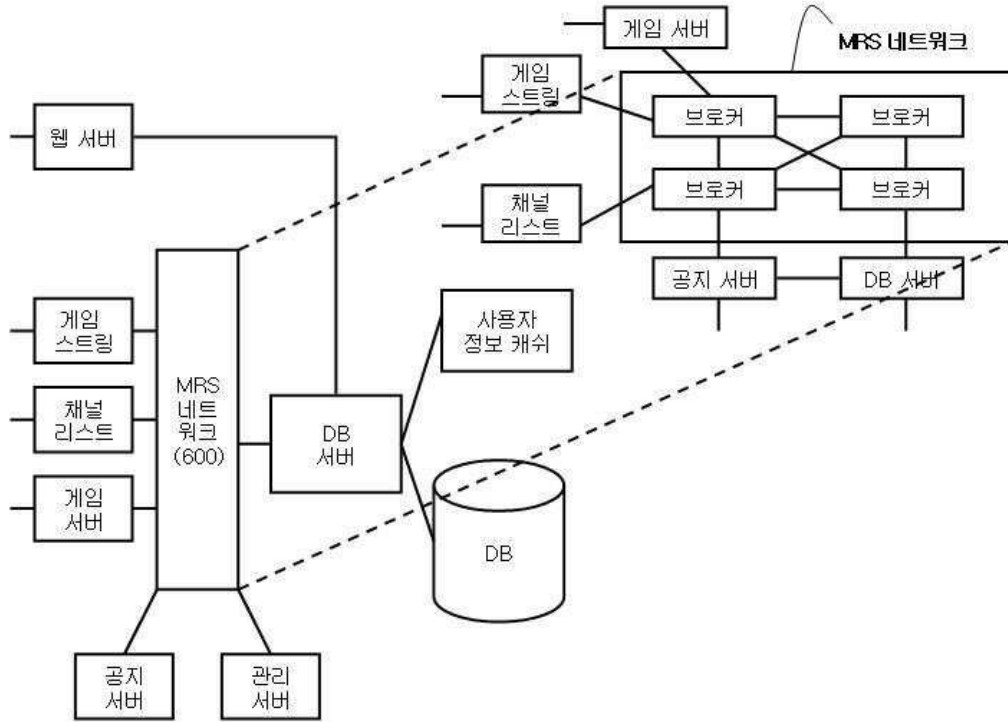


도면5





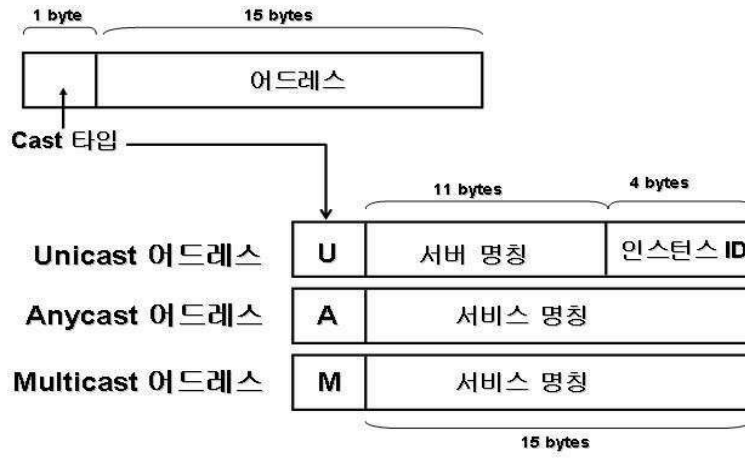
도면6



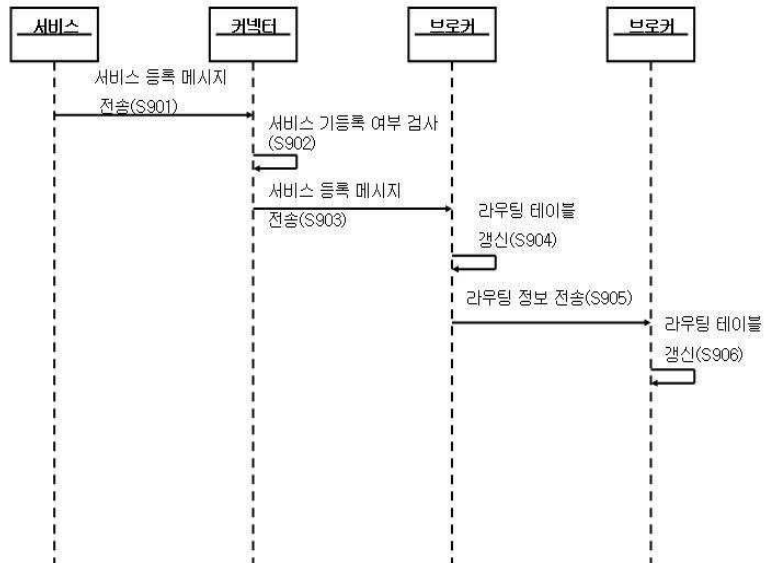
도면7



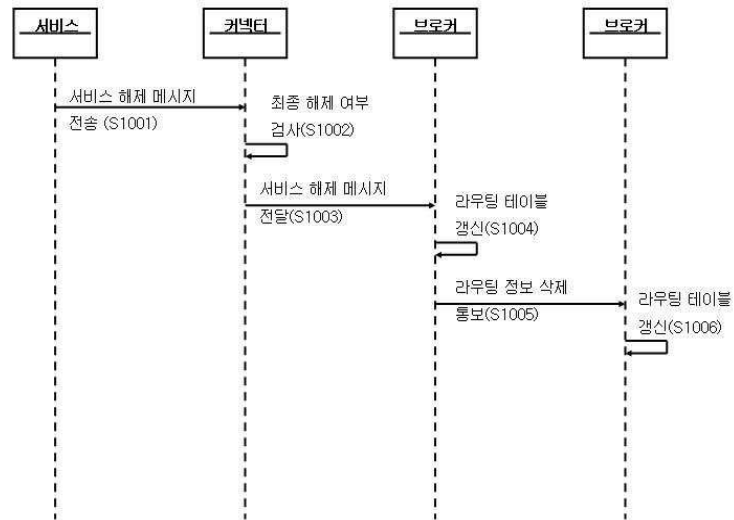
도면8



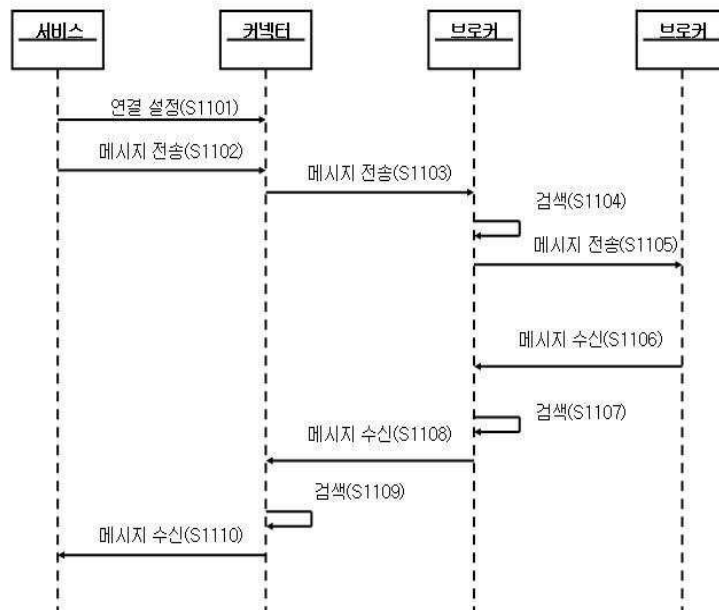
도면9



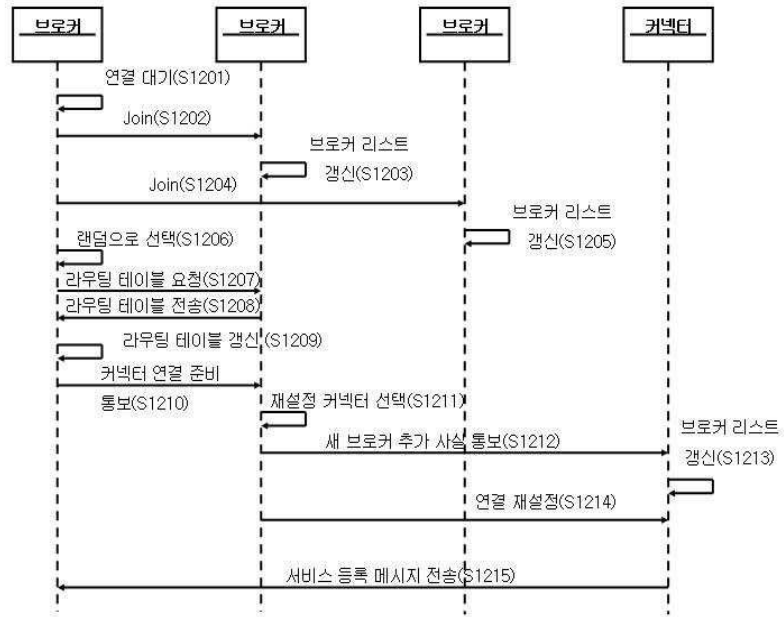
도면10



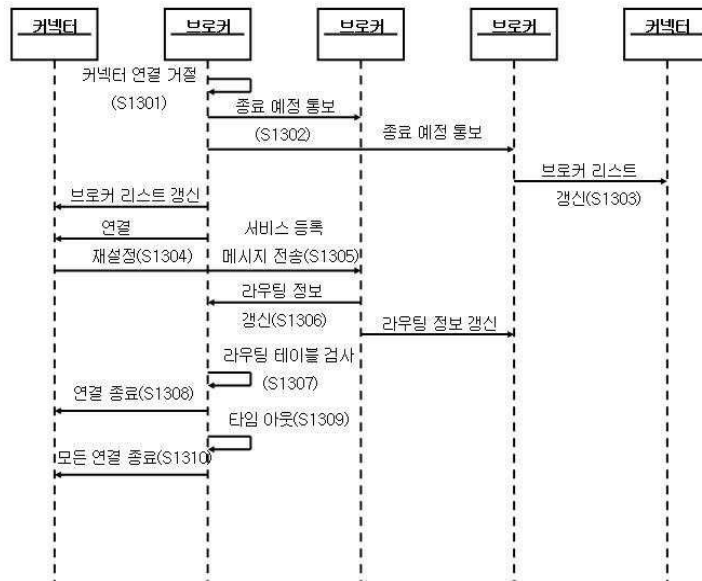
도면11



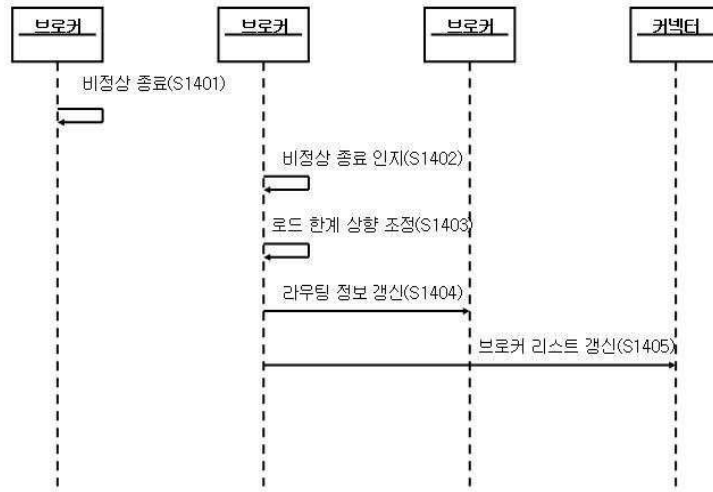
도면12



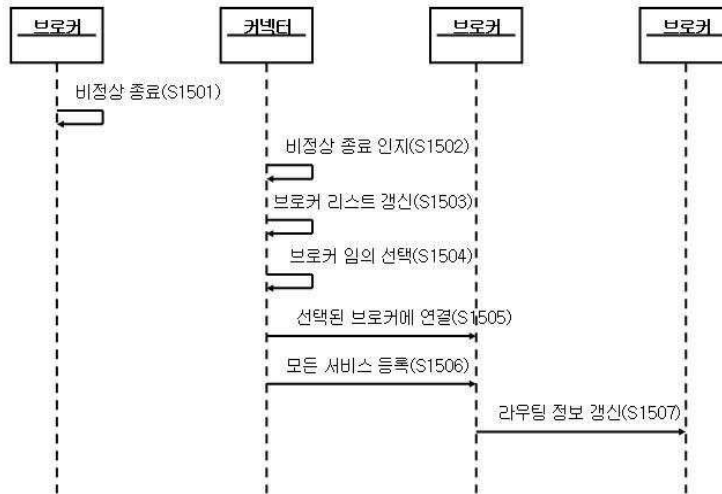
도면13



도면14

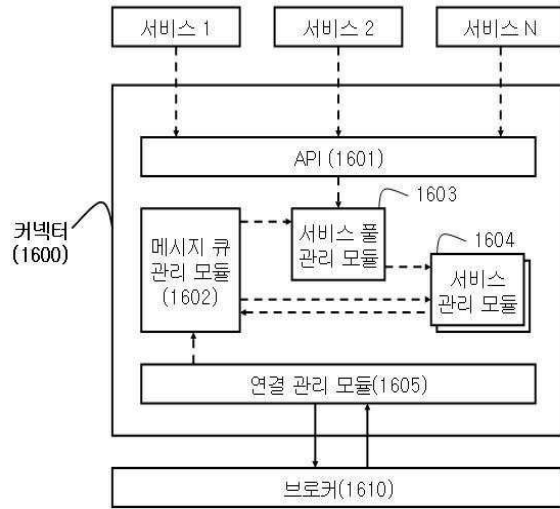


도면15

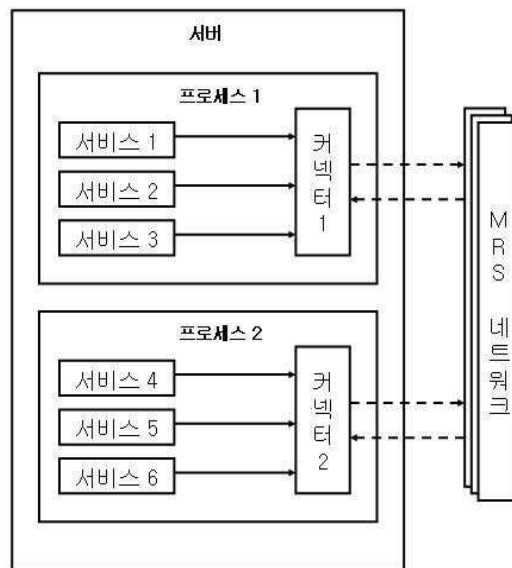




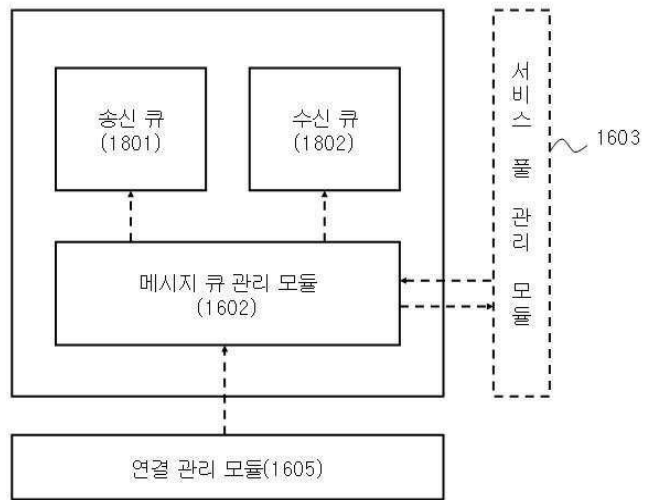
도면16



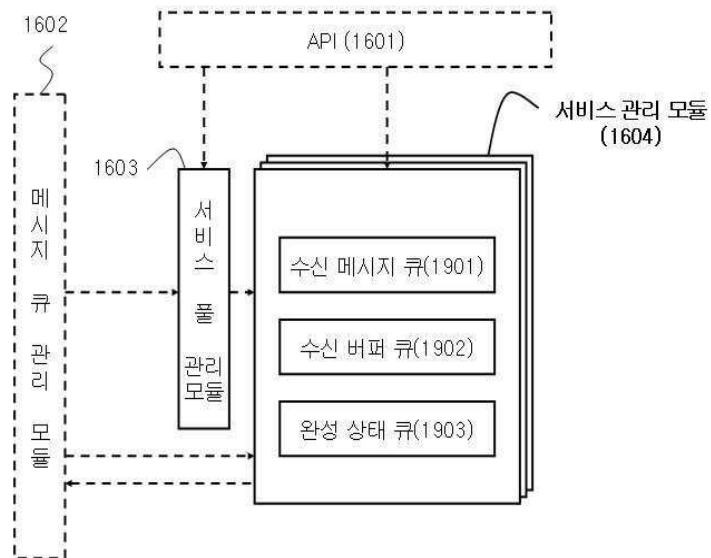
도면17



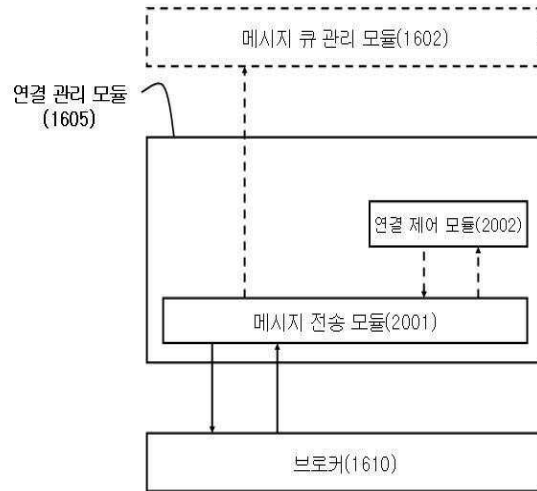
도면18



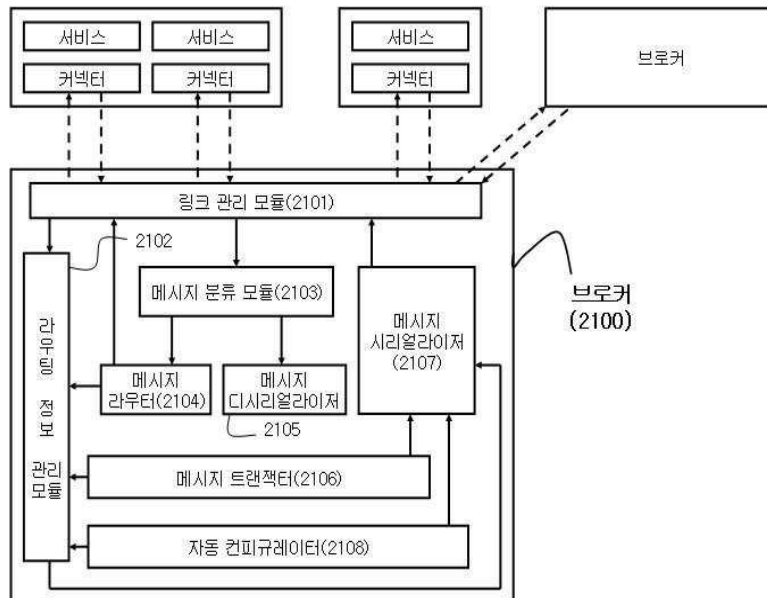
도면19



도면20



도면21



도면22

