



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0038245
(43) 공개일자 2015년04월08일

- | | |
|--|--|
| <p>(51) 국제특허분류(Int. Cl.)
H04L 12/801 (2013.01) G06F 13/00 (2006.01)
G06F 3/0481 (2013.01)</p> <p>(52) CPC특허분류
H04L 47/10 (2013.01)
G06F 13/00 (2013.01)</p> <p>(21) 출원번호 10-2015-7004603</p> <p>(22) 출원일자(국제) 2013년09월12일
심사청구일자 2015년02월24일</p> <p>(85) 번역문제출일자 2015년02월24일</p> <p>(86) 국제출원번호 PCT/JP2013/005427</p> <p>(87) 국제공개번호 WO 2014/041813
국제공개일자 2014년03월20일</p> <p>(30) 우선권주장
JP-P-2012-202112 2012년09월13일 일본(JP)</p> | <p>(71) 출원인
닛본 덴끼 가부시끼가이샤
일본국 도쿄도 미나토구 시바 5쵸메 7방 1코</p> <p>(72) 발명자
고이테 도시오
일본국 도쿄도 미나토구 시바 5쵸메 7-1 닛본 덴끼 가부시끼가이샤 내</p> <p>(74) 대리인
문두현, 문기상</p> |
|--|--|

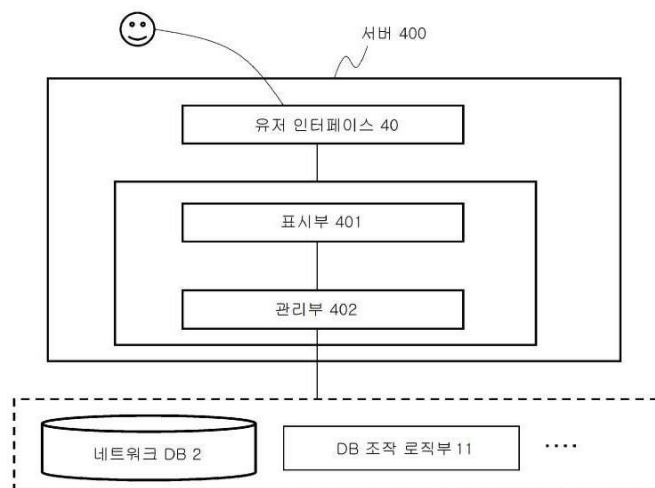
전체 청구항 수 : 총 18 항

(54) 발명의 명칭 정보 처리 장치, 구성 방법, 통신 시스템, 및 프로그램

(57) 요약

제어의 유연성을 손상시키지 않고 중앙 제어형 네트워크의 설정의 용이성(구성 용이성)을 향상시킬 수 있다. 네트워크를 제어하는 제어 장치를 구성하는 정보 처리 장치는, 네트워크를 제어하기 위한 제어 정보를 저장하는 데이터베이스에 대응하는 제 1 아이콘, 및 미리 정해진 알고리즘에 의거하여 제어 정보를 각각 조작하는 복수의 모듈 각각에 대응하는 복수의 제 2 아이콘을 디스플레이에 표시하는 표시부, 및 제 1 아이콘과 어느 하나의 제 2 아이콘의 접속에 따라, 제 2 아이콘에 대응하는 모듈을 데이터베이스에 접속하고, 모듈에 의해 발행되는 데이터베이스 조작 커맨드에 응답하여 제어 정보를 조작하는 기능을 갖는 제어 장치를 구성하는 관리부를 포함하는 정보 처리 장치.

대표도



(52) CPC특허분류

G06F 3/04817 (2013.01)

명세서

청구범위

청구항 1

네트워크를 제어하는 제어 장치를 구성하는 정보 처리 장치로서,

상기 네트워크를 제어하기 위한 제어 정보를 저장하는 데이터베이스에 대응하는 제 1 아이콘, 및 미리 정해진 알고리즘에 의거하여 상기 제어 정보를 각각 조작하는 복수의 모듈 각각에 대응하는 복수의 제 2 아이콘을 디스플레이에 표시하는 표시부, 및

상기 제 1 아이콘과 어느 하나의 상기 제 2 아이콘과의 접속에 따라, 상기 제 2 아이콘에 대응하는 모듈을 상기 데이터베이스에 접속하고, 상기 모듈에 의해 발행되는 데이터베이스 조작 커맨드에 응답하여 상기 제어 정보를 조작하는 기능을 갖는 제어 장치를 구성하는 관리부를 포함하는 정보 처리 장치.

청구항 2

제 1 항에 있어서,

상기 제 1 아이콘에 대한, 상기 데이터베이스의 상태 표시의 요청에 따라, 상기 표시부는 상기 데이터베이스에 저장된 제어 정보를 표시하는 정보 처리 장치.

청구항 3

제 1 항 또는 제 2 항에 있어서,

상기 제 1 아이콘과 어느 하나의 상기 제 2 아이콘과의 접속에 따라, 상기 표시부는, 상기 제 2 아이콘에 대응하는 모듈에 의해 발행되는 데이터베이스 조작 커맨드에 응답하여 제어 정보를 조작해서 얻어지는 결과를 표시하는 정보 처리 장치.

청구항 4

제 1 항 내지 제 3 항 중 어느 한 항에 있어서,

상기 제 1 아이콘과 상기 네트워크의 상태의 표시를 실행하는 모듈에 대응하는 상기 제 2 아이콘과의 접속에 따라, 상기 표시부는 상기 네트워크의 상태를 표시하는 정보 처리 장치.

청구항 5

제 1 항 내지 제 4 항 중 어느 한 항에 있어서,

상기 제어 정보에 대한 조작은, 상기 복수의 모듈에 의해 사용되는 공통의 데이터베이스 조작 커맨드에 응답하여 실행되는 정보 처리 장치.

청구항 6

제 1 항 내지 제 5 항 중 어느 한 항에 있어서,

상기 데이터베이스는 상기 네트워크의 토폴로지에 대한 제 1 정보 및 상기 네트워크에서의 패킷 처리 방법에 대한 제 2 정보를 포함하는 정보 처리 장치.

청구항 7

제 1 항 내지 제 6 항 중 어느 한 항에 있어서,

상기 데이터베이스는, 상기 제어 장치가 상기 네트워크로부터 수신하는 패킷을 관리하기 위한 제 3 정보, 및 상기 제어 장치가 상기 네트워크에 전송하는 패킷을 관리하기 위한 제 4 정보를 포함하는 정보 처리 장치.

청구항 8

제 1 항 내지 제 7 항 중 어느 한 항에 있어서,

상기 제 2 아이콘은, 통신 단말들 사이의 경로 계산, 네트워크 토폴로지의 관리, 네트워크 토폴로지의 애그리게이션(aggregation), 네트워크의 분할, 및 네트워크의 결합으로 구성되는 그룹에서 선택되는 적어도 하나를 실행하는 모듈에 대응하는 정보 처리 장치.

청구항 9

네트워크를 제어하는 제어 장치의 구성 방법으로서,

상기 네트워크를 제어하기 위한 제어 정보를 저장하는 데이터베이스에 대응하는 제 1 아이콘, 및 미리 정해진 알고리즘에 의거하여 상기 제어 정보를 각각 조작하는 복수의 모듈 각각에 대응하는 복수의 제 2 아이콘을 디스플레이에 표시하는 단계, 및

상기 제 1 아이콘과 어느 하나의 상기 제 2 아이콘과의 접속에 따라, 상기 제 2 아이콘에 대응하는 모듈을 상기 데이터베이스에 접속하고, 상기 모듈에 의해 발행되는 데이터베이스 조작 커맨드에 응답하여 상기 제어 정보를 조작하는 기능을 갖는 제어 장치를 구성하는 단계를 포함하는 구성 방법.

청구항 10

제 9 항에 있어서,

상기 제 1 아이콘에 대한, 상기 데이터베이스의 상태 표시의 요청에 따라, 상기 데이터베이스에 저장된 제어 정보를 상기 디스플레이에 표시하는 단계를 포함하는 구성 방법.

청구항 11

제 9 항 또는 제 10 항에 있어서,

상기 제 1 아이콘과 어느 하나의 상기 제 2 아이콘과의 접속에 따라, 상기 제 2 아이콘에 대응하는 모듈에 의해 발행되는 데이터베이스 조작 커맨드에 응답하여 제어 정보를 조작해서 얻어지는 결과를 상기 디스플레이에 표시하는 단계를 포함하는 구성 방법.

청구항 12

제 9 항 내지 제 11 항 중 어느 한 항에 있어서,

상기 제 1 아이콘과 상기 네트워크의 상태의 표시를 실행하는 모듈에 대응하는 상기 제 2 아이콘과의 접속에 따라, 상기 네트워크의 상태를 상기 디스플레이에 표시하는 단계를 포함하는 구성 방법.

청구항 13

제 9 항 내지 제 12 항 중 어느 한 항에 있어서,

상기 복수의 모듈에 의해 사용되는 공통의 데이터베이스 조작 커맨드에 응답하여, 상기 제어 정보에 대한 조작을 실행하는 단계를 포함하는 구성 방법.

청구항 14

제 9 항 내지 제 13 항 중 어느 한 항에 있어서,

상기 데이터베이스는 상기 네트워크의 토폴로지에 대한 제 1 정보 및 상기 네트워크에서의 패킷 처리 방법에 대한 제 2 정보를 포함하는 구성 방법.

청구항 15

제 9 항 내지 제 14 항 중 어느 한 항에 있어서,

상기 데이터베이스는, 상기 제어 장치가 상기 네트워크로부터 수신하는 패킷을 관리하기 위한 제 3 정보, 및 상기 제어 장치가 상기 네트워크에 전송하는 패킷을 관리하기 위한 제 4 정보를 포함하는 구성 방법.

청구항 16

제 9 항 내지 제 15 항 중 어느 한 항에 있어서,

상기 제 2 아이콘은, 통신 단말들 사이의 경로 계산, 네트워크 토폴로지의 관리, 네트워크 토폴로지의 애그리게이션, 네트워크의 분할, 및 네트워크의 결합으로 구성되는 그룹에서 선택되는 적어도 하나를 실행하는 모듈에 대응하는 구성 방법.

청구항 17

네트워크를 제어하는 제어 장치를 구성하는 정보 처리 장치를 포함하는 통신 시스템으로서,

상기 정보 처리 장치는,

상기 네트워크를 제어하기 위한 제어 정보를 저장하는 데이터베이스에 대응하는 제 1 아이콘, 및 미리 정해진 알고리즘에 의거하여 상기 제어 정보를 각각 조작하는 복수의 모듈 각각에 대응하는 복수의 제 2 아이콘을 디스플레이에 표시하는 표시부, 및

상기 제 1 아이콘과 어느 하나의 상기 제 2 아이콘과의 접속에 따라, 상기 제 2 아이콘에 대응하는 모듈을 상기 데이터베이스에 접속하고, 상기 모듈에 의해 발행되는 데이터베이스 조작 커맨드에 응답하여 상기 제어 정보를 조작하는 기능을 갖는 제어 장치를 구성하는 관리부를 포함하는 통신 시스템.

청구항 18

네트워크를 제어하는 제어 장치를 구성하는 정보 처리 장치에,

상기 네트워크를 제어하기 위한 제어 정보를 저장하는 데이터베이스에 대응하는 제 1 아이콘, 및 미리 정해진 알고리즘에 의거하여 상기 제어 정보를 각각 조작하는 복수의 모듈 각각에 대응하는 복수의 제 2 아이콘을 디스플레이에 표시하는 처리, 및

상기 제 1 아이콘과 어느 하나의 상기 제 2 아이콘과의 접속에 따라, 상기 제 2 아이콘에 대응하는 모듈을 상기 데이터베이스에 접속하고, 상기 모듈에 의해 발행되는 데이터베이스 조작 커맨드에 응답하여 상기 제어 정보를 조작하는 기능을 갖는 제어 장치를 구성하는 처리를 실행시키는 프로그램.

발명의 설명

기술 분야

[0001]

(관련 출원의 상호 참조)

[0002]

본 발명은 2012 년 9 월 13 일에 출원된 일본 특허 출원 제2012-202112에 의거하여 우선권을 주장하고 그 공개물은 전체가 참조에 의해 본원에 통합된다.

[0003]

본 발명은 정보 처리 장치, 구성 방법, 통신 시스템, 및 프로그램에 관한 것이다. 특히, 본 발명은 중앙식으로 관리되는 스위치들의 그룹을 제어하는 제어 장치를 구성하는 데 사용되는 정보 처리 장치, 구성 방법, 통신 시스템, 및 프로그램에 관한 것이다.

배경 기술

[0004]

최근, 오픈플로우(OpenFlow)라고 하는 기술이 제안되었다(비특허문헌 1 및 2 참조). 오픈플로우는 엔드-투-엔드(end-to-end) 플로우로서의 통신이라 이해되고 플로우마다 경로 제어, 장애 복구, 로드 밸런싱, 및 최적화를 행한다. 비특허문헌 2에 따른 오픈플로우 스위치는 오픈플로우 제어부와 통신하기 위한 보안 채널을 갖고 오픈플로우 제어부에 의해 적절히 추가 또는 재가입되는 플로우 테이블에 따라 동작한다. 플로우 테이블에는, 플로우마다, 세 가지로 이루어지는 세트, 즉 패킷 헤더가 매칭되는 매칭 조건(Match Fields), 플로우 통계 정보(Counters), 및 처리 내용을 정의하는 인스트럭션이 정의된다(비특허문헌 2의 "4.1 Flow Table" 섹션을 참조).

[0005]

예를 들면, 오픈플로우 스위치는 패킷의 수신 시 들어오는 패킷의 헤더 정보와 매칭되는 매칭 조건(비특허문헌 2의 "4.3 Match Fields" 참조)을 갖는 엔트리에 대해서 플로우 테이블을 검색한다. 검색 결과로서 오픈플로우 스위치가 들어오는 패킷과 매칭되는 엔트리를 발견하면, 오픈플로우 스위치는 플로우 통계 정보(Counters)를 업데이트하고 엔트리의 Instructions 필드에 기입된 처리 내용(지정된 포트로부터 패킷 전송, 플러딩(flooding), 드롭(drop) 등)에 의거하여 들어오는 패킷을 처리한다. 검색 결과로서 오픈플로우 스위치가 들어

오는 패킷과 매칭되는 엔트리를 발견하지 못할 경우, 오픈플로우 스위치는 엔트리 설정 요청(Packet-In 메시지)을 보안 채널을 통해 오픈플로우 제어부에 전송한다. 즉, 오픈플로우 스위치는 들어오는 패킷을 처리하기 위한 제어 정보의 전송을 오픈플로우 제어부에 요청한다. 오픈플로우 스위치는 처리 내용을 정의하는 플로우 엔트리를 수신하고 플로우 테이블을 업데이트한다. 이렇게, 제어 정보로서 플로우 테이블에 저장된 엔트리를 사용함으로써 오픈플로우 스위치는 패킷 포워딩을 실시한다.

선행기술문헌

비특허문헌

[0006] (비특허문헌 0001) Nick McKeown 외 7 명, "OpenFlow: Enabling Innovation in Campus Networks, "[online], [2012 년 7 월 13 일 검색], Internet <URL:http://www.openflow.org/documents/openflow-wp-latest.pdf>

(비특허문헌 0002) "OpenFlow Switch Specification" Version 1.0.0 Implemented(Wire Protocol 0x01), [online], [2012 년 7 월 13 일 검색], Internet <URL:http://www.openflow.org/documents/openflow-spec-v1.0.0.pdf>

발명의 내용

해결하려는 과제

[0007] 본 발명에 의해 다음 분석이 이루어졌다. 비특허문헌 1 및 2에서 오픈플로우로 대표되는 중앙 제어형 네트워크에서 세부 제어가 행해질 수 있다. 그러나, 비특허문헌 1 및 2의 오픈플로우 제어부에 대응하는 제어 장치의 의도한 동작을 실행하게 하기 위해서는, 오퍼레이터가 제어부의 제어 동작을 프로그래밍할 필요가 있다. 제어부의 프로그래밍은 비특허문헌 2의 기본 사양 외에도, 네트워크 구성의 관리, 프로그래밍, 및 테스트 환경의 구성 등을 위한 고도의 지식과 기술을 필요로 한다.

[0008] 예를 들면, 비특허문헌 2에서 오픈플로우 제어부를 구성하기 위해서는, 각 스위치와 제어 채널 사이의 연관, LLDP(Link Layer Discovery Protocol)에 의한 토폴로지 검색, 경로 계산, 경로 상의 복수의 스위치에서의 플로우 엔트리들의 설정, 및 스위치마다의 입출력 패킷들의 관리를 포함하여, 다양한 아이템이 설정 및 프로그래밍될 필요가 있다. 예를 들면, 프로그래밍을 실시하기 위해서는 토폴로지 관리 및 경로 계산과 관련된 API(Application Programming Interface), API에 관련된 많은 함수, 및 이들 함수에 부여될 필요가 있는 인수들 등과 같은 상당한 양의 지식 및 스킬이 필요로 된다.

[0009] 따라서 중앙 제어형 네트워크 디바이스들을 설치할 경우에도, 사용자는 제어의 유연성을 충분히 활용할 수 없을 수 있다.

[0010] 본 발명의 목적은, 제어의 유연성을 손상시키지 않고 중앙 제어형 네트워크의 설정 용이성(구성 용이성)을 향상시킬 수 있는 정보 처리 장치, 구성 방법, 통신 시스템, 및 프로그램을 제공하는 것이다.

과제의 해결 수단

[0011] 제 1 양태에 따르면, 네트워크를 제어하는 제어 장치를 구성하는 정보 처리 장치가 제공된다. 상기 정보 처리 장치는, 상기 네트워크를 제어하기 위한 제어 정보를 저장하는 데이터베이스에 대응하는 제 1 아이콘, 및 미리 정해진 알고리즘에 의거하여 상기 제어 정보를 각각 조작하는 복수의 모듈 각각에 대응하는 복수의 제 2 아이콘을 디스플레이에 표시하는 표시부, 및 상기 제 1 아이콘과 어느 하나의 상기 제 2 아이콘과의 접속에 따라, 상기 제 2 아이콘에 대응하는 모듈을 상기 데이터베이스에 접속하고, 상기 모듈에 의해 발행되는 데이터베이스 조작 커맨드에 응답하여 상기 제어 정보를 조작하는 기능을 갖는 제어 장치를 구성하는 관리부를 포함한다.

[0012] 제 2 양태에 따르면, 네트워크를 제어하는 제어 장치의 구성 방법이 제공된다. 상기 구성 방법은, 상기 네트워크를 제어하기 위한 제어 정보를 저장하는 데이터베이스에 대응하는 제 1 아이콘, 및 미리 정해진 알고리즘에 의거하여 상기 제어 정보를 각각 조작하는 복수의 모듈 각각에 대응하는 복수의 제 2 아이콘을 디스플레이에 표시하는 것, 및 상기 제 1 아이콘과 어느 하나의 상기 제 2 아이콘과의 접속에 따라, 상기 제 2 아이콘에 대응하는 모듈을 상기 데이터베이스에 접속하고, 상기 모듈에 의해 발행되는 데이터베이스 조작 커맨드에 응답하여 상

기 제어 정보를 조작하는 기능을 갖는 제어 장치를 구성하는 것을 포함한다. 이 방법은 특정 기기, 즉 데이터베이스 및 모듈을 이용하여 네트워크를 제어하는 제어 장치와 연관된다.

[0013]

제 3 양태에 따르면, 네트워크를 제어하는 제어 장치를 구성하는 정보 처리 장치를 포함하는 통신 시스템이 제공된다. 상기 정보 처리 장치는, 상기 네트워크를 제어하기 위한 제어 정보를 저장하는 데이터베이스에 대응하는 제 1 아이콘, 및 미리 정해진 알고리즘에 의거하여 상기 제어 정보를 각각 조작하는 복수의 모듈 각각에 대응하는 복수의 제 2 아이콘을 디스플레이에 표시하는 표시부, 및 상기 제 1 아이콘과 어느 하나의 상기 제 2 아이콘과의 접속에 따라, 상기 제 2 아이콘에 대응하는 모듈을 상기 데이터베이스에 접속하고, 상기 모듈에 의해 발행되는 데이터베이스 조작 커맨드에 응답하여 상기 제어 정보를 조작하는 기능을 갖는 제어 장치를 구성하는 관리부를 포함한다.

[0014]

제 4 양태에 따르면, 네트워크를 제어하는 제어 장치를 구성하는 정보 처리 장치에, 상기 네트워크를 제어하기 위한 제어 정보를 저장하는 데이터베이스에 대응하는 제 1 아이콘, 및 미리 정해진 알고리즘에 의거하여 상기 제어 정보를 각각 조작하는 복수의 모듈 각각에 대응하는 복수의 제 2 아이콘을 디스플레이에 표시하는 처리, 및 상기 제 1 아이콘과 어느 하나의 상기 제 2 아이콘과의 접속에 따라, 상기 제 2 아이콘에 대응하는 모듈을 상기 데이터베이스에 접속하고, 상기 모듈에 의해 발행되는 데이터베이스 조작 커맨드에 응답하여 상기 제어 정보를 조작하는 기능을 갖는 제어 장치를 구성하는 처리를 실행시키는 프로그램이 제공된다. 이 프로그램은 컴퓨터 판독 가능한 (비일시적인) 저장 매체에 기록될 수 있다. 즉, 본 발명은 컴퓨터 프로그램 제품으로 구현될 수 있다.

발명의 효과

[0015]

본 발명에 따르면, 제어의 유연성을 손상시키지 않고 중앙 제어형 네트워크의 설정의 용이성(구성 용이성)을 향상시킬 수 있다.

도면의 간단한 설명

[0016]

- 도 1은 본 발명의 제 1 실시예에 따른 시스템의 구성을 나타내는 도면.
- 도 2는 본 발명의 제 1 실시예에 따른 제어 장치의 구성을 나타내는 도면.
- 도 3은 본 발명의 제 1 실시예에 따른 네트워크 DB의 구성을 나타내는 도면.
- 도 4는 본 발명의 제 1 실시예에 따른 동작을 나타내는 순서도.
- 도 5는 본 발명의 제 1 실시예에 따른 동작을 나타내는 순서도.
- 도 6은 본 발명의 제 1 실시예에 따른 동작을 나타내는 순서도.
- 도 7은 본 발명의 제 2 실시예에 따른 네트워크 DB의 구성을 나타내는 도면.
- 도 8은 본 발명의 제 2 실시예에 따른 네트워크 DB의 구성을 나타내는 도면.
- 도 9는 본 발명의 제 2 실시예에 따른 네트워크 DB의 구성을 나타내는 도면.
- 도 10은 본 발명의 제 2 실시예에서 사용되는 데이터베이스 조작 커맨드들(DB 조작 커맨드들)을 나타내는 도면.
- 도 11은 본 발명의 제 2 실시예에서 사용되는 데이터베이스 조작 커맨드들(DB 조작 커맨드들)을 나타내는 도면.
- 도 12는 본 발명의 제 3 실시예에 따른 시스템의 구성을 나타내는 도면.
- 도 13은 본 발명의 제 3 실시예에 따른 오픈플로우 제어부의 구성을 나타내는 도면.
- 도 14는 본 발명의 제 3 실시예에 따른 토폴로지 처리부에 의한 네트워크 DB의 조작 플로우를 나타내는 도면.
- 도 15는 본 발명의 제 3 실시예에 따른 플로우 처리부에 의한 네트워크 DB의 조작 플로우를 나타내는 도면.
- 도 16은 본 발명의 제 3 실시예에 따른 패킷 처리부에 의한 네트워크 DB의 조작 플로우를 나타내는 도면.
- 도 17은 본 발명의 제 3 실시예에 따른 동작을 나타내는 순서도.
- 도 18은 본 발명의 제 3 실시예에 따른 동작을 나타내는 순서도.
- 도 19는 본 발명의 제 3 실시예에 따른 동작을 나타내는 순서도.

- 도 20은 본 발명의 제 3 실시예에 따른 동작을 나타내는 순서도.
- 도 21은 본 발명의 제 4 실시예에 따른 시스템의 구성을 나타내는 도면.
- 도 22는 본 발명의 제 4 실시예에 따른 네트워크 토폴로지의 그래픽 표시를 나타내는 도면.
- 도 23은 본 발명의 제 4 실시예에 따른 패킷 처리 정보의 그래픽 표시를 나타내는 도면.
- 도 24는 본 발명의 제 4 실시예에 따른 패킷 입출력 정보의 그래픽 표시를 나타내는 도면.
- 도 25는 본 발명의 제 5 실시예에 따른 시스템의 구성을 나타내는 도면.
- 도 26은 본 발명의 제 6 실시예에 따른 시스템의 구성을 나타내는 도면.
- 도 27은 본 발명의 제 7 실시예에 따른 시스템의 구성을 나타내는 도면.
- 도 28은 본 발명의 제 7 실시예에 따른 애그리게이션 로직부에 의한 네트워크 DB의 조작을 나타내는 도면.
- 도 29는 본 발명의 제 7 실시예에 따른 애그리게이션 로직부에 의한 네트워크 DB의 조작을 나타내는 도면.
- 도 30은 본 발명의 제 7 실시예에 따른 애그리게이션 로직부에 의해 네트워크 DB의 조작을 나타내는 도면.
- 도 31은 본 발명의 제 8 실시예에 따른 시스템의 구성을 나타내는 도면.
- 도 32는 본 발명의 제 8 실시예에 따른 결합 로직부에 의한 네트워크 DB의 조작을 나타내는 도면.
- 도 33은 본 발명의 제 8 실시예에 따른 결합 로직부에 의한 네트워크 DB의 조작을 나타내는 도면.
- 도 34는 본 발명의 제 9 실시예에 따른 시스템의 구성을 나타내는 도면.
- 도 35는 본 발명의 제 9 실시예에 따른 제어 장치의 컴포넌트 매니저의 구성을 나타내는 도면.
- 도 36은 본 발명의 제 10 실시예에 따른 시스템의 구성을 나타내는 도면.
- 도 37은 본 발명의 제 10 실시예에 따른 표시부에 표시되는 화면을 나타내는 도면.
- 도 38은 본 발명의 제 10 실시예에 따른 사용자 작업 내용과 제어 장치 구성 사이의 대응 관계를 나타내는 도면.
- 도 39는 본 발명의 제 10 실시예에 따른 사용자 작업 내용과 제어 장치 구성 사이의 대응 관계를 나타내는 도면.
- 도 40은 본 발명의 제 10 실시예에 따른 사용자 작업 내용과 제어 장치 구성 사이의 대응 관계를 나타내는 도면.
- 도 41은 본 발명의 제 10 실시예에 따른 사용자 작업 내용과 제어 장치 구성 사이의 대응 관계를 나타내는 도면.
- 도 42는 본 발명의 제 10 실시예에 따른 사용자 작업 내용과 제어 장치 구성 사이의 대응 관계를 나타내는 도면.

발명을 실시하기 위한 구체적인 내용

<제 1 실시예>

본 발명의 일 실시예에 따르면, 제어 장치는, 예를 들면 네트워크의 구성 및 네트워크에서의 패킷 처리 방법에 대한 제어 정보를 저장하는 DB(데이터베이스)를 조작함으로써 중앙식으로 네트워크를 제어할 수 있다. 예를 들면, DB는 Key-Value(키-값) 스토어에 의해 구성되며 미리 정해진 DB 조작 커맨드를 통해 조작된다. DB는 DB에 대하여 정보를 입출력하기 위한 커맨드 및 DB의 정보를 삭제하는 커맨드 등의 간단한 조작 커맨드를 통해 조작될 수 있다. 제 1 실시예에서, 제어 장치는 조작 커맨드를 통해 DB를 조작하는 모듈을 조합함으로써 구성될 수 있다. 따라서, 제 1 실시예에 따른 제어 장치는, API 또는 그와 관련된 많은 함수를 사용하지 않고 간단한 DB 조작 커맨드들을 통해 프로그래밍될 수 있다.

다음으로, 제 1 실시예를 도면을 참조하여 상세하게 설명한다. 도 1은 본 발명의 제 1 실시예에 따른 통신 시스템의 구성을 나타낸다. 도 1에서, 통신 시스템은 제어 장치(1), 네트워크 DB(2), 및 복수의 노드(3)를 포함

하는 네트워크(4)로 구성된다.

- [0020] 제어 장치(1)는 네트워크 DB(2)를 조작함으로써 중앙식으로 네트워크(4)를 제어한다.
- [0021] 네트워크(4)의 각 노드(3)는 제어 장치(1)에 따라 패킷을 처리한다.
- [0022] 예를 들면, 네트워크 DB(2)는 Key-Value 스토어에 의해 구성되는 데이터베이스이다. 예를 들면, 네트워크 DB(2)는 식별 정보(Key) 및 데이터(Value)를 쌍을 지어 저장한다. 네트워크 DB(2)에서, 식별 정보(Key)는 데이터(Value)마다 설정된다.
- [0023] 도 2는 제어 장치(1)의 구성을 나타내는 도면이다. 도 2에서, 제어 장치(1)는 DB 조작 로직부(11), 네트워크 제어부(12), 및 네트워크 DB(2)에 접속된 DB 인터페이스부(10)를 포함한다.
- [0024] DB 인터페이스부(10)는 DB 조작 로직부(11)에 의해 발행된 미리 정해진 DB 조작 커맨드를 사용함으로써 네트워크 DB(2)를 조작하는 기능을 갖는다. 또한, 제어 장치(1)는 복수의 DB 조작 로직부(11)를 조합함으로써 구성될 수 있다. 따라서, DB 인터페이스부(10)가 복수의 DB 조작 로직부(11)에 접속될 수 있다(도 2에서 파선으로 나타낸 부분). DB 인터페이스부(10)는 복수의 DB 조작 로직부(11)로부터 발행된 커맨드들을 이용함으로써, 네트워크 DB(2)를 조작할 수 있다. DB 인터페이스부(10)는 네트워크 DB(2) 내의 복수의 DB 조작 로직부(11)의 조합에 의거한 정보를 구성할 수 있다. 즉, 오퍼레이터는 DB 인터페이스부(10)에 접속된 DB 조작 로직부(11)들의 조합을 변경함으로써 네트워크 DB(2)의 구성을 용이하게 변경할 수 있다. 따라서, 오퍼레이터는 네트워크(4)의 타입 또는 네트워크(4)에서 사용되는 통신 프로토콜에 적합한 제어 장치(1)를 용이하게 구성할 수 있다.
- [0025] 네트워크 DB(2)를 조작하는 커맨드들의 예를 이하에 열거한다. 그러나, 다음의 커맨드들은 예이므로, 본 발명에 따른 DB 조작 커맨드들은 다음 예에 한정되는 것은 아니다.
- [0026] 정보 입출력 커맨드
- [0027] 정보 삭제 커맨드
- [0028] 통지 예약 커맨드
- [0029] 정보 입출력 커맨드는 네트워크 DB(2)에 정보를 입력하는 데 사용된다. 또한, 정보 입출력 커맨드는 네트워크 DB(2)로부터 정보를 취득하는 데 사용된다. 예를 들면, DB 인터페이스부(10)는 커맨드 "get(key)"을 사용하여 네트워크 DB(2)로부터 식별 정보(key)에 대응하는 데이터(Value)를 취득한다. 또한, DB 인터페이스부(10)는 커맨드 "put(key, value)"을 사용하여 네트워크 DB(2)에 식별 정보(Key)에 대응하는 데이터(Value)를 입력한다. 데이터(Value)가 리스팅되면, DB 인터페이스부(10)는 커맨드 "push(key, value)" 또는 커맨드 "pop(key)"을 사용하여, 데이터(Value)의 입력 또는 출력을 실행할 수 있다. 즉, 데이터를 큐 또는 스택과 같이 해서 사용할 수 있다.
- [0030] 정보 삭제 커맨드는 네트워크 DB(2)로부터 정보를 삭제하는 데 사용된다. 예를 들면, DB 인터페이스부(10)는 커맨드 "delete(key)"를 사용하여 네트워크 DB(2)로부터 식별 정보(key)에 대응하는 데이터(value)를 제거한다.
- [0031] 예를 들면, 통지 예약 커맨드는 예약을 위해 사용된다. 이 커맨드에 의해, 지정된 식별 정보(key)에 대응하는 조작이 네트워크 DB(2) 상에서 실행될 경우, 네트워크 DB(2)는 식별 정보(key)에 대응하는 데이터(value)를 DB 인터페이스부(10)에 통지한다. 예를 들면, DB 인터페이스부(10)는 네트워크 DB(2)에 커맨드 "subscribe(key)"를 전송하고, 식별 정보(key)에 대응하는 조작이 실행될 경우, DB 인터페이스부(10)는 네트워크 DB(2)로부터 식별 정보(key)에 대응하는 데이터(value)를 수신한다. 식별 정보에는, 와일드카드 또는 정규 표현 등의 key 패턴이 사용될 수 있다. 예를 들면, 네트워크 DB(2)는 커맨드 "publish(key, value, op)"를 사용하여, 식별 정보에 대응하는 데이터(value)를 DB 인터페이스부(10)에 통지한다. 키에 대응하는 조작(put/get/delete를 포함)이 "op"에 포함될 수 있다. 통지받은 후, DB 조작 로직부(11)는 식별 정보에 대응하는 키의 추가, 취득, 또는 삭제를 인식할 수 있다.
- [0032] 복수의 DB 조작 로직부(11)는 공통의 DB 조작 커맨드들을 공유할 수 있다. 복수의 DB 조작 로직부(11)가 공통의 커맨드들로 네트워크 DB(2)를 조작할 수 있게 함으로써, 네트워크 오퍼레이터는 DB 조작 로직부(11)들을 용이하게 확장할 수 있다. 즉, 새로운 기능을 갖는 DB 조작 로직부(11)를 제어 장치(1)에 추가할 경우, 오퍼레이터는 DB 조작 로직부(11)와 네트워크 DB(2) 사이의 인터페이스를 새롭게 설계할 필요가 없다.
- [0033] DB 조작 로직부(11)는 DB 조작 로직부(11)의 타입에 의거하여 정의된 미리 정해진 알고리즘에 의거하여 네트워크 DB(2)에 저장된 정보를 조작하는 기능을 갖는다. 예를 들면, DB 조작 로직부(11)는 네트워크(4)로부터 취득

된 정보를 DB 인터페이스부(10)를 통해 네트워크 DB(2)에 입력하는 기능을 갖는다.

- [0034] 네트워크 제어부(12)는 DB 조작 로직부(11)의 조작에 의해 구성된 네트워크 DB(2)에 의거하여 네트워크(4)를 제어하는 기능을 갖는다. 예를 들면, 네트워크 제어부(12)는, 네트워크(4)에서 사용되는 통신 프로토콜에 의존하여, 네트워크(4)를 제어하는 방법을 변경할 수 있다.
- [0035] 도 3은 네트워크 DB(2)의 구성을 나타낸다. 이 도 3의 구성은 일례이며, 이에 따라 네트워크 DB(2)의 구성은 도 3의 구성에 한정되지 않는다.
- [0036] 예를 들면, 네트워크 DB(2)는 네트워크(4)의 토폴로지에 대한 정보(토폴로지 정보; 제 1 정보), 패킷 처리 방법에 대한 정보(패킷 처리 정보; 제 2 정보), 및 패킷에 대한 정보(패킷 정보; 제 3, 제 4 정보)를 포함한다.
- [0037] 예를 들면, 토폴로지 정보는 네트워크(4)에 존재하는 각 노드(3)에 대한 정보(노드 정보) 및 각 노드 쌍 사이의 링크에 대한 정보(링크 정보)를 포함한다. 예를 들면, 노드 정보는 노드(3)의 식별자 및 통신 포트에 대한 정보를 포함한다. 예를 들면, 링크 정보는 링크의 식별자, 링크에 접속된 노드(3)의 식별자, 및 링크에 접속된 노드(3)의 통신 포트의 식별자를 포함한다.
- [0038] 패킷 처리 정보는 플로우 정보를 포함한다. 플로우 정보는 패킷이 속하는 플로우에 대한 정보이다. 플로우는 미리 정해진 속성(통신 수신처, 송신원 등에 의거하여 식별될 수 있는 속성)을 갖는 통신 패킷들의 그룹을 의미한다. 예를 들면, 플로우 정보는 매칭 조건, 경로 정보, 및 처리 정보를 포함한다. 예를 들면, 매칭 조건은 플로우 정보에 대응하는 패킷들을 식별하기 위한 조건을 나타낸다. 예를 들면, 경로 정보는 매칭 조건과 매칭되는 플로우에 속하는 패킷들이 이동하는 경로를 나타낸다. 예를 들면, 처리 정보는 경로 정보에 대응하는 경로를 통해 이동하는 패킷들에 대해 실행될 처리(예를 들면, 헤더 정보의 재기입)를 나타낸다.
- [0039] 패킷 정보는 입력 패킷 정보 및 출력 패킷 정보를 포함한다. 입력 패킷 정보는 네트워크(4)로부터 수신되는 패킷 중 임의의 플로우 정보의 매칭 조건과 매칭되지 않는 패킷이다. 출력 패킷 정보는 네트워크(4)로부터 출력될 필요가 있는 패킷이다.
- [0040] 예를 들면, 네트워크 DB(2)에 저장된 토폴로지 정보, 패킷 처리 정보, 및 입력 패킷 정보는 미리 정해진 포맷으로 표준화되어 있을 수 있다. 네트워크 DB(2)에 저장되는 정보를 표준화함으로써, 네트워크 DB(2)는 다양한 프로토콜을 따르는 네트워크들과의 호환성을 가질 수 있다.
- [0041] 도 4 내지 도 6은 본 실시예에 따른 조작들을 나타낸다. 도 4 내지 도 6의 조작들은 예이므로, 본 발명에 따른 조작들은 이 도면들에 도시된 조작들에 한정되지 않는다.
- [0042] 도 4 내지 도 6은 네트워크 DB(2)가 복수의 DB 조작 로직부(11)("DB 조작 로직부들(11-1, 11-2)")에 의해 조작되는 예를 나타낸다. DB 조작 로직부(11-1)는 네트워크 DB(2)를 조작하고 토폴로지에 대한 정보를 입출력하는 기능을 갖는다. DB 조작 로직부(11-2)는 네트워크 DB(2)에 저장된 정보에 의거하여 패킷 처리 정보를 생성하는 기능을 갖는다.
- [0043] 도 4는 DB 조작 로직부(11-1)가 네트워크 DB(2)를 조작함으로써 네트워크(4)의 토폴로지를 관리하는 동작을 나타낸다.
- [0044] 제어 장치(1)의 DB 조작 로직부(11-1)는 네트워크 제어부(12)를 통해 네트워크(4)로부터 토폴로지에 대한 정보를 수집한다(S10). 토폴로지에 대한 정보는, 예를 들면 LLDP(Link Layer Discovery Protocol)를 사용하여 수집된다. LLDP는, 네트워크 상의 장치들이 이웃 장치들과 정보를 교환하고 장치들의 통신 포트들에 연결된 이웃 장치들에 대한 정보를 수집하는 프로토콜이다. 예를 들면, 네트워크 제어부(12)는 LLDP를 통해 네트워크(4)의 각 노드(3)에 의해 수집된 정보를 각 노드(3)로부터 수집한다.
- [0045] 예를 들면, DB 조작 로직부(11-1)는, 네트워크 제어부(12)를 통해 수집된 정보에 의거하여, 네트워크 DB(2)에 대응하는 미리 정해진 포맷의 토폴로지 정보를 생성한다(S11).
- [0046] DB 조작 커맨드(정보 입출력 커맨드)를 이용함으로써, DB 조작 로직부(11-1)는 생성된 토폴로지 정보를 DB 인터페이스부(10)를 통해 네트워크 DB(2)에 저장한다(S12).
- [0047] 도 5는, DB 조작 로직부(11-2)가 네트워크 DB(2)를 조작함으로써 네트워크(4)에서 흐르는 패킷들을 처리하는 방법에 대한 정보를 생성하는 동작을 나타낸다.
- [0048] DB 조작 커맨드(정보 입출력 커맨드)를 사용함으로써, 제어 장치(1)의 DB 조작 로직부(11-2)는 네트워크 DB(2)

로부터 DB 인터페이스부(10)를 통해 토폴로지 정보를 취득한다(S13).

- [0049] DB 조작 로직부(11-2)는 통지 예약 커맨드를 사용하여 네트워크 DB(2)를 미리 설정해서, 토폴로지 정보가 변경될 경우 네트워크 DB(2)가 토폴로지 정보의 변경을 DB 조작 로직부(11-2)에 통지할 수 있다. 토폴로지 정보가 변경될 경우, 네트워크 DB(2)는 토폴로지 정보의 변경을 DB 인터페이스부(10)를 통해 DB 조작 로직부(11-2)에 통지한다.
- [0050] 네트워크 DB(2)로부터 취득한 토폴로지 정보에 의거하여, DB 조작 로직부(11-2)는 네트워크 DB(2)에 대응하는 미리 정해진 포맷의 패킷 처리 정보를 생성한다(S14). 예를 들면, DB 조작 로직부(11-2)는 패킷 포워딩 경로(도 3의 "경로 정보")를 계산하고 계산된 포워딩 경로를 패킷 처리 정보에 저장한다. 예를 들면, DB 조작 로직부(11-2)는 미리 정해진 계산 로직을 사용함으로써 포워딩 경로를 계산한다. 예를 들면, DB 조작 로직부(11-2)는 송신원으로부터 수신처까지의 최단 경로를 패킷 포워딩 경로에 설정한다.
- [0051] 또한, DB 조작 로직부(11-2)는 패킷 매칭 조건 및 처리 정보(예를 들면, 헤더의 재기입)를 패킷 처리 정보에 저장한다.
- [0052] DB 조작 커맨드(정보 입출력 커맨드)를 이용함으로써, DB 조작 로직부(11-2)는 생성된 패킷 처리 정보를 DB 인터페이스부(10)를 통해 네트워크 DB(2)에 저장한다(S15).
- [0053] 도 6은 제어 장치(1)가 네트워크 DB(2)를 조작함으로써 네트워크(4)의 노드(3)를 제어하는 동작을 나타낸다.
- [0054] DB 조작 커맨드(정보 입출력 커맨드)를 이용함으로써, 제어 장치(1)의 DB 조작 로직부(11-1)는 네트워크 DB(2)로부터 DB 인터페이스부(10)를 통해 패킷 처리 정보를 취득한다(S16).
- [0055] DB 조작 로직부(11-1)가 통지 예약 커맨드를 이용함으로써 네트워크 DB(2)를 미리 설정해서, 패킷 처리 정보가 변경되는 경우, 네트워크 DB(2)는 패킷 처리 정보의 변경을 DB 조작 로직부(11-1)에 통지한다. 예를 들면, DB 조작 로직부(11-1)는 통지 예약 커맨드를 이용하여 네트워크 DB(2)의 업데이트의 통지를 위한 통지 조건을 네트워크 DB(2)에 설정한다. 이러한 미리 설정된 통지 조건과 매칭되는 업데이트가 실행될 경우, 네트워크 DB(2)는 네트워크 DB(2)의 업데이트된 내용을 DB 조작 로직부(11-1)에 통지한다. 예를 들면, 패킷 처리 정보가 변경될 경우, 네트워크 DB(2)는 패킷 처리 정보의 변경을 DB 인터페이스부(10)를 통해 DB 조작 로직부(11-1)에 통지한다.
- [0056] 패킷 처리 정보에 의거하여, DB 조작 로직부(11-1)는 네트워크(4)에 설정될 처리 명령(노드(3)에 의해 실행되는 패킷 처리 방법을 정의하는 정보)을 생성하고 네트워크 제어부(12)를 통해 처리 명령을 노드(3)에 설정한다(S17).
- [0057] 예를 들면, DB 조작 로직부(11-1)는 패킷 처리 정보의 경로 정보를 참조하고 경로 정보에 대응하는 포워딩 경로상의 각 노드(3)에 처리 명령을 설정한다. 예를 들면, DB 조작 로직부(11-1)는 각 노드(3)에, 포워딩 경로에 대응하는 통신 포트로부터 패킷의 포워딩을 정의하는 처리 명령을 설정해서, 경로 정보에 대응하는 포워딩 경로를 따라 패킷을 포워딩한다.
- [0058] 도 4 내지 도 6에 나타난 상기 조작들에서와 같이, 네트워크 DB(2)를 조작함으로써, 제어 장치(1)는 중앙식으로 네트워크(4)에서의 패킷 처리를 제어할 수 있다.
- [0059] 상술한 바와 같이, DB 조작 커맨드들로 네트워크 DB(2)를 조작함으로써, 제어 장치(1)는 네트워크(4)를 제어할 수 있다. 따라서, 네트워크(4)의 오퍼레이터는 네트워크 DB(2)의 DB 조작 커맨드들을 조합함으로써 제어 장치(1)를 프로그래밍할 수 있다. DB 조작 커맨드들의 수는 API 또는 그에 관련된 함수들에 필요한 커맨드들의 수보다 훨씬 적다. 이에 따라, 네트워크(4)의 오퍼레이터는 고도의 지식 및 스킬을 습득할 필요없이 제어 장치(1)를 용이하게 프로그래밍할 수 있다.
- [0060] <제 2 실시예>
- [0061] 다음으로, 본 발명의 제 2 실시예에 대해 설명한다. 제 2 실시예에서, 네트워크 DB(2)의 구성에 및 DB 조작 커맨드들의 예를 상세하게 설명한다.
- [0062] 도 7 내지 도 9는 네트워크 DB(2)의 구성을 나타낸다. 도 7 내지 도 9의 구성은 일례이며, 따라서 네트워크 DB(2)의 구성은 이에 한정되지 않는다.
- [0063] 또한, 제 2 실시예에서, 네트워크 DB(2)는 Key-Value 스토어에 의해 구성된다. 그러나, 네트워크 DB(2)는 Key-

Value 스토어 이외의 다른 방법에 의해 구성될 수 있다.

- [0064] 도 7은 네트워크 DB(2)에 저장되는 토폴로지 정보를 나타낸다. 토폴로지 정보는 네트워크(4)의 노드(3)들에 대한 식별 정보(Key) 및 노드(3)들 간의 링크들에 대한 식별 정보(Key)에 의해 구성된다. 각 식별 정보(Key)는 데이터(Value)와 연관지어진다.
- [0065] 예를 들면, 식별 정보 `"/nodes"`에 대응하는 데이터(Value)는 네트워크(4)에 존재하는 노드들(3)의 식별자(노드 ID)들을 나타낸다. 도 7의 `{1, 2, 3, 4, ...}`는 노드 식별자들의 리스트를 나타낸다.
- [0066] 식별 정보 `"/nodes/[node_id]"`의 `"[node_id]"`는 네트워크(4)에 존재하는 노드(3)들 중 하나의 식별자를 나타낸다. 예를 들면, 식별 정보 `"/nodes/[node_id]"`에 대응하는 데이터(Value)는, 제어 장치(1)가 네트워크(4)의 노드(3)를 제어할 때 사용되는 식별자에 대한 정보 또는 노드(3)의 제조 벤더에 대한 정보를 나타낸다. 제어 장치(1)가 네트워크(4)의 노드(3)를 제어할 때 사용되는 식별자는 상기 식별 정보 `"/nodes"`와 연관된 식별자와는 상이하다. 오픈플로우 프로토콜에서, 예를 들면 제어 장치(1)가 네트워크(4)의 노드(3)를 제어할 때 사용하는 식별자는 DPID(DataPath ID)라고 한다.
- [0067] 예를 들면, 식별 정보 `"/nodes/[node_id]/ports"`에 대응하는 데이터(Value)는 `"[node_id]"`에 대응하는 노드(3)의 통신 포트의 식별자의 리스트를 나타낸다. 도 7에서의 `{1, 2, 3, 4, ...}`는 통신 포트의 식별자의 리스트를 나타낸다.
- [0068] 식별 정보 `"/nodes/[node_id]/ports/[port_id]"`의 `"[port_id]"`는 `"[node_id]"`에 대응하는 노드(3)의 통신 포트들 중 하나를 나타내는 식별자를 나타낸다. 예를 들면, 식별 정보 `"/nodes/[node_id]/ports/[port_id]"`의 데이터(Value)는 포트 상태(예를 들면, 링크-업 상태 또는 링크-다운 상태), 포트의 통신 대역폭 등을 포함한다.
- [0069] 예를 들면, 식별 정보 `"/links"`에 대응하는 데이터(Value)는 네트워크(4)에 존재하는 노드(3)들 사이의 링크들의 식별자를 나타낸다. 도 7의 `{1, 2, 3, 4, ...}`는 링크 식별자들의 리스트를 나타낸다.
- [0070] 식별 정보 `"/link/[link_id]"`의 `"[link_id]"`는 네트워크(4)에 존재하는 노드(3)들 사이의 링크들 중 하나의 식별자를 나타낸다. 예를 들면, 식별 정보 `"/link/[link_id]"`에 대응하는 데이터(Value)는 링크 양단의 노드들에 대한 정보 및 이들 노드들의 통신 포트들에 대한 정보를 포함한다. 예를 들면, 도 7에서, 데이터(Value)는 링크의 패킷 소스 노드의 식별자(`"src_node_id"`) 및 링크의 패킷 수신처 노드의 식별자(`"dst_node_id"`)를 포함한다. 도 7에서, `"src_port_id"`는 `"src_node_id"`에 대응하는 노드의 통신 포트들 중 링크에 대응하는 포트의 식별자를 나타낸다. 도 7에서, `"dst_port_id"`는 `"dst_node_id"`에 대응하는 노드의 통신 포트들 중 링크에 대응하는 포트의 식별자를 나타낸다.
- [0071] 네트워크 DB(2)는 도 7에 나타난 이들 데이터베이스 엔트리들을 사용함으로써 네트워크(4)의 토폴로지를 관리한다. DB 조작 커맨드들을 사용하고 도 7에 나타난 엔트리들을 조작함으로써, DB 조작 로직부(11)는 네트워크(4)의 토폴로지를 관리하고 조작할 수 있다.
- [0072] 도 8은 네트워크 DB(2)에 저장된 패킷 처리 정보를 나타낸다. 예를 들면, 패킷 처리 정보는 패킷 처리 정보에 대응하는 패킷을 식별하기 위한 매칭 조건(Filter); 매칭 조건과 매칭되는 패킷에 대한 포워딩 경로(Path); 및 매칭 조건과 매칭되는 패킷에 적용될 미리 정해진 처리(Action)를 포함한다.
- [0073] 식별 정보(`"/flows"`)에 대응하는 데이터(Value)는 패킷 처리 정보의 식별자를 나타낸다. 도 8의 `{1, 2, 3, 4, ...}`는 패킷 처리 정보의 식별자들의 리스트를 나타낸다.
- [0074] 식별 정보(`"/flows/[flow_id]"`)의 `"[flow_id]"`는 패킷 처리 정보의 하나의 식별자를 나타낸다. 예를 들면, 식별 정보(`"/flows/[flow_id]"`)의 데이터(Value)는 `"[flow_id]"`에 대응하는 패킷 처리 정보에 대응하는 패킷들을 식별하기 위한 매칭 조건을 포함한다. 예를 들면, 처리 방법이 알려져 있지 않은 패킷을 수신하면, 제어 장치(1)의 DB 조작 로직부(11)는 DB 조작 커맨드를 사용하여, 패킷에 대응하는 매칭 조건을 갖는 패킷 처리 정보에 대해 네트워크 DB(2)를 검색한다. 네트워크 DB(2)는 패킷에 대한 정보(예를 들면, 헤더 정보)를 참조하고 패킷에 대응하는 매칭 조건을 갖는 패킷 처리 정보를 DB 조작 로직부(11)에 통지한다. 제공된 패킷 처리 정보에 의거하여, DB 조작 로직부(11)는 네트워크(4)의 노드(3)에 패킷을 처리하도록 지시한다.
- [0075] 네트워크 DB(2)가 패킷에 대한 정보(예를 들면, 헤더 정보)를 참조하여 패킷에 대응하는 매칭 조건을 갖는 패킷 처리 정보를 검색하기 위한 DB 조작 커맨드를 갖지 않을 경우가 있다. 이러한 경우에는, 제어 장치(1)의 DB 조작 로직부(11)는 미리 DB 조작 커맨드를 사용하여 네트워크 DB(2)로부터 필요한 패킷 처리 정보를 판독하고 그 정보를 저장할 수 있다. 이렇게, DB 조작 로직부(11)는, 단순히 패킷에 대한 정보를 참조함으로써 대응하는 패

킷 처리 정보를 결정할 수 있다. 결정된 패킷 처리 정보에 의거하여, DB 조작 로직부(11)는 네트워크(4)의 노드(3)에 패킷 처리에 대한 지시를 한다.

[0076] 식별 정보 "/flows/[flow_id]"에 대응하는 데이터(Value)는 매칭 조건과 매칭되는 패킷에 대한 포워딩 경로("Path") 및 매칭 조건과 매칭되는 패킷에 대해 실행되는 처리("Action")를 포함한다. DB 조작 로직부(11)는 패킷 포워딩에 대한 지시를 네트워크(4)의 노드(3)들에 통지해서, 패킷은 패킷 처리 정보에 정의된 포워딩 경로를 따라 포워딩된다. 또한, 패킷에 대해 실시되는 처리("Action")가 존재할 경우, DB 조작 로직부(11)는 네트워크(4)의 노드(3)에 처리를 실행하도록 지시한다.

[0077] 도 9는 처리 방법이 결정되지 않은 패킷들로서 제어 장치(1)에 포워딩되는 패킷들에 대한 데이터베이스 엔트리들을 나타낸다. 네트워크 DB(2)는 처리 방법이 결정되지 않은 패킷에 대한 정보를 데이터베이스 엔트리들로서 관리할 수 있다.

[0078] 식별 정보 "/packets/in"에 대응하는 데이터(Value)는 패킷 처리 방법이 결정되지 않은 패킷들로서 제어 장치(1)에 전송되는 패킷들의 식별자의 리스트를 나타낸다. 예를 들면, 식별자 리스트는, 리스트의 말미에 값을 추가하기 위한 DB 조작 커맨드 "push_tail" 및 리스트의 선두로부터의 값을 추출하기 위한 DB 조작 커맨드 "pop_head"를 이용함으로써 FIFO(First In First Out)의 큐로서 사용된다.

[0079] 예를 들면, 식별 정보 "/packets/in[packet_id]"의 데이터(Value)는 패킷의 식별자, 패킷의 수신 시각, 패킷을 수신한 노드(3)의 식별자, 및 패킷 수신 노드(3)의 통신 포트의 식별자를 포함한다.

[0080] 예를 들면, 패킷이 제어 장치(1)에 포워딩될 경우, DB 조작 로직부(11)는 DB 조작 커맨드를 사용하여, 패킷에 대한 정보를 네트워크 DB(2)에 저장한다.

[0081] 식별 정보 "/packets/out"에 대응하는 데이터(Value)는 처리 방법이 결정되어 있고 제어 장치(1)로부터 네트워크(4)로 반환될 필요가 있는 패킷들의 식별자들의 리스트를 나타낸다. 예를 들면, "/packets/in"에 대응하는 큐에 저장된 패킷들 중에서, 처리 방법이 결정된 패킷들은 "/packets/out"에 대응하는 큐에 저장된다. 예를 들면, "/packets/out"에 대응하는 데이터(Value)는 리스트이며 FIFO형 큐로서 사용된다.

[0082] 예를 들면, 식별 정보 "/packets/out/[packet_id]"에 대응하는 데이터(Value)는 제어 장치(1)로부터 패킷이 반환되는 노드의 식별자, 노드의 통신 포트들 중에서 패킷이 반환되는 포트의 식별자, 및 패킷에 저장된 데이터(즉, 패킷 자체)를 포함한다.

[0083] 예를 들면, DB 조작 로직부(11)는 통지 예약 커맨드를 사용하여, 네트워크 DB(2)를 조작해서, "/packets/out"에 대응하는 큐에 패킷이 저장되어 있을 경우 및 키 "/packets/out"이 조작될 경우, DB 조작 로직부(11)는 통지를 받는다. 패킷이 큐에 저장되어 있을 경우(즉, "/packets/out" 및 "/packets/out/[packet_id]"에 대응하는 엔트리가 업데이트될 경우), 네트워크 DB(2)는 "/packets/out/[packet_id]"에 대응하는 데이터의 업데이트를 DB 조작 로직부(11)에 통지한다.

[0084] 식별 정보 "/packets/out" 및 "/packets/out/[packet_id]"에 저장된 패킷은 "/packets/in"에 대응하는 큐에 저장된 패킷에 대응하지 않을 수 있다. 이렇게, 제어 장치(1)는 패킷들을 능동적으로 출력할 수 있다.

[0085] 도 10 및 도 11은, DB 조작 로직부(11)가 네트워크 DB(2)를 조작할 때 사용되는 DB 조작 커맨드들을 나타낸다. 도 10 및 도 11에 나타난 바와 같이, push 커맨드들 및 pop 커맨드들을 이용함으로써, 리스트의 임의의 위치로부터 데이터 삽입 및 취득 등의 조작들이 실행될 수 있다. 또한, 예를 들면, 제어 장치(1)는 예약 커맨드를 사용하여, 데이터베이스에서 특정 키를 갖는 엔트리의 변경을 통지받는다. 또한, 예를 들면, get, put, 및 delete 등의 커맨드들을 사용하여, 노드, 링크, 플로우, 및 패킷에 대한 상세 정보 또는 값을 취득, 설정, 및 삭제한다.

[0086] 도 10 및 도 11에 나타난 이들 DB 조작 커맨드들을 이용하고 도 7 내지 도 9에 나타난 바와 같이 구성된 네트워크 DB(2)를 조작함으로써, DB 조작 로직부(11)는 중앙식으로 네트워크(4)를 제어할 수 있다. 따라서, DB 조작 커맨드들을 사용하고 네트워크 DB(2)에 대한 조작 지시들을 조합함으로써, 오퍼레이터는 제어 장치(1)를 프로그래밍할 수 있다. DB 조작 커맨드들의 수는 일반적인 프로그래밍에 사용되는 API 또는 그에 관련되는 함수들에 필요한 커맨드들의 수보다 훨씬 적다. 이에 따라, 오퍼레이터는 일반적인 프로그래밍과 비교하여 제어 장치(1)를 매우 용이하게 프로그래밍할 수 있다.

[0087] <제 3 실시예>

- [0088] 다음으로, 본 발명의 제 3 실시예에 대해 설명한다. DB 조작 로직부(11)는 다양한 기능을 갖는 모듈에 의해 실시될 수 있다. 제 3 실시예에서는, 오픈플로우 제어부(11A)가 DB 조작 로직부(11)로서 배치되어 있다.
- [0089] 도 12는 제 3 실시예에 따른 시스템의 구성을 나타낸다. 오픈플로우 제어부(11A)는 오픈플로우 프로토콜에 의거하여 네트워크(4)를 제어하는 기능을 갖는다. 제 3 실시예를, 네트워크(4)가 오픈플로우 프로토콜과 호환 가능한 노드(3)에 의해 구성되는 것으로 상정하여 설명한다. 제 3 실시예에서는, 필요에 따라, 네트워크(4)를 오픈플로우 네트워크라고 한다.
- [0090] 도 13은 오픈플로우 제어부(11A)의 구성을 나타낸다. 오픈플로우 제어부(11A)는 토폴로지 처리부(110), 플로우 처리부(111), 및 패킷 처리부(112)를 포함한다.
- [0091] 토폴로지 처리부(110)는 DB 조작 커맨드를 사용하여, DB 인터페이스부(10)를 통해 네트워크 DB(2)를 조작한다. 예를 들면, 토폴로지 처리부(110)는 네트워크 DB(2)를 조작하여, 네트워크 DB(2)에 토폴로지 정보를 설정한다. 또한, 예를 들면, 토폴로지 처리부(110)는 네트워크 DB(2)를 조작하여, 네트워크 DB(2)로부터 토폴로지 정보를 취득한다.
- [0092] 도 14는 토폴로지 처리부(110)가 네트워크 DB(2)를 조작하는 조작 예를 나타낸다. 토폴로지 처리부(110)는 커맨드 "push_tail(/nodes)"를 사용하여, 엔트리 리스트 "/nodes"의 말미에 오픈플로우 네트워크에 추가된 노드의 식별자를 부가한다. 또한, 토폴로지 처리부(110)는 커맨드 "put(/nodes/[node_id], <node_info>)"를 사용하여, 커맨드의 인수인 "<node_info>"에 대응하는 정보를 엔트리 "/nodes/[node_id]"에 추가한다([node_id]는 엔트리 "/nodes"에 추가된 노드의 식별자임). 예를 들면, 토폴로지 처리부(110)는 제 1 실시예에서 설명한 정보를 엔트리 "/nodes/[node_id]"에 추가한다.
- [0093] 상기 커맨드 "push_tail(/nodes)" 대신에, 커맨드 "push_tail(/nodes, <node_info>)"를 사용하여, 상기 일련의 조작들을 일괄적으로 행할 수 있다. 이러한 경우에, 커맨드 "put(/nodes/[node_id], <node_info>)"의 실행을 생략할 수 있다.
- [0094] 토폴로지 처리부(110)는 커맨드 "push_tail(/links)"을 사용하여, 오픈플로우 네트워크에 추가된 노드 사이의 링크의 식별자를 엔트리 리스트 "/links"의 말미에 추가한다. 또한, 토폴로지 처리부(110)는 커맨드 "put(/links/[link_id], <link_info>)"를 사용하여, 커맨드의 인수인 "<link_info>"에 대응하는 정보를 엔트리 "/links/[link_id]"에 추가한다([link_id]는 엔트리 "/links"에 추가된 링크의 식별자임). 예를 들면, 토폴로지 처리부(110)는 제 1 실시예에서 설명한 정보를 엔트리 "/links/[link_id]"에 추가한다.
- [0095] 상기 커맨드 "push_tail(/links)" 대신에 커맨드 "push_tail(<link_info>)"를 이용함으로써, 상기 일련의 조작들을 일괄적으로 행할 수 있다. 이 경우에, 커맨드 "put(/links/[link_id], <link_info>)"의 실행을 생략할 수 있다.
- [0096] 노드 또는 링크가 오픈플로우 네트워크에서 삭제될 경우, 토폴로지 처리부(110)는 커맨드 "pop_at"를 사용하여, 노드 또는 링크 리스트로부터 삭제된 노드 또는 링크의 식별자를 삭제한다. 노드 또는 링크가 오픈플로우 네트워크에서 삭제될 경우, 토폴로지 처리부(110)는 커맨드 "삭제"를 사용하여, 네트워크 DB(2)로부터 삭제된 노드 또는 링크에 대응하는 엔트리를 삭제한다.
- [0097] 커맨드 "pop_at"을 이용함으로써, 상기 일련의 조작들을 일괄적으로 행할 수 있다. 따라서, 커맨드 "삭제"의 실행을 생략할 수 있다.
- [0098] 커맨드 "get"을 이용함으로써, 토폴로지 처리부(110)는 지정된 식별 정보(key)에 대응하는 데이터(value)을 취득한다.
- [0099] 플로우 처리부(111)는 DB 조작 커맨드를 사용하여, DB 인터페이스부(10)를 통해 네트워크 DB(2)를 조작한다. 예를 들면, 플로우 처리부(111)는 네트워크 DB(2)를 조작하여, 네트워크 DB(2)에 패킷 처리 정보를 설정한다. 또한, 예를 들면, 플로우 처리부(111)는 네트워크 DB(2)를 조작하여, 처리 방법이 알려지지 않은 패킷에 대응하는 패킷 처리 정보에 대해 네트워크 DB(2)를 검색한다.
- [0100] 도 15는 플로우 처리부(111)가 네트워크 DB(2)를 조작하는 조작 예를 나타낸다. 제어 장치(1)가 오픈플로우 네트워크로부터 처리 방법이 알려져 있지 않은 패킷을 수신할 경우에, 플로우 처리부(111)는 패킷 처리부(112)로부터 패킷의 처리 방법에 대한 검색 요청을 수신한다. 예를 들면, 패킷 처리부(112)는 커맨드 "get_matching_flow(<packet_info>)"를 사용하여, 플로우 처리부(111)에, 패킷의 처리 방법에 대한 검색을 요청한다. 커맨드의 인수인 "<packet_info>"는 처리 방법이 알려져 있지 않은 패킷에 대한 정보이다(예를 들면,

헤더 정보 또는 패킷 자체).

- [0101] 패킷 처리부(112)로부터의 요청에 의거하여, 플로우 처리부(111)는 "<packet_info>"에 대응하는 매칭 조건을 갖는 패킷 처리 정보에 대해 네트워크 DB(2)를 검색한다. 플로우 처리부(111)는 네트워크 DB(2)로부터 검색된 패킷 처리 정보의 식별자("flow_id")를 수신한다. 예를 들면, 수신된 식별자에 의거하여 커맨드 "get(/flows/[flow_id])"을 사용하여, 플로우 처리부(111)는 패킷 처리 정보를 취득한다. 플로우 처리부(111)는 패킷 처리 정보를 패킷 처리부(112)에 제공한다. 검색 조건과 매칭되는 그러한 패킷 처리 정보가 존재하지 않을 경우에는, 플로우 처리부(111)는 그 결과를 패킷 처리부(112)에 통지한다.
- [0102] 네트워크 DB(2)가 "<packet_info>"에 대응하는 매칭 조건을 갖는 패킷 처리 정보에 대해 검색하기 위한 커맨드를 갖고 있지 않을 경우에는, 미리 플로우 처리부(111)는 DB 조작 커맨드를 사용하여, 네트워크 DB(2)로부터 필요한 패킷 처리 정보를 판독하고 정보를 저장한다. 이렇게, 플로우 처리부(111)는 패킷에 대한 정보를 참조함으로써 대응하는 패킷 처리 정보를 결정할 수 있다.
- [0103] 대응하는 패킷 처리 정보가 존재하지 않을 경우에는, 패킷 처리부(112)는 커맨드 "push_tail(/packets/in,<packet_info>)"을 사용하여 수신된 패킷을 네트워크 DB(2)에 추가한다.
- [0104] 대응하는 패킷 처리 정보가 존재할 경우에는, 패킷 처리부(112)는 플로우 처리(111)에 패킷을 처리도록 요청한다.
- [0105] 취득된 패킷 처리 정보에 의거하여, 플로우 처리부(111)는 오픈플로우 네트워크의 노드(3)에 패킷 취급 조작(즉, 패킷 처리 규칙)을 설정함과 동시에 노드(3)가, 패킷 처리 정보에 의거하여 노드(3)의 적절한 노드 통신 포트로부터 수신된 패킷을 출력하게 한다.
- [0106] 예를 들면, 패킷 취급 조작은, 오픈플로우 프로토콜에 의거하여 정의된 포맷의 정보이며, 노드(3)가 패킷을 처리하기 위해 사용된다. 패킷 취급 조작은 노드(3)에 의해 수신된 패킷을 식별하기 위한 식별 조건 및 식별 조건과 매칭되는 패킷을 처리하기 위한 방법을 포함한다. 플로우 처리부(111)는 노드(3)에 패킷 취급 조작을 설정하고, 노드(3)는 설정된 패킷 취급 조작을 메모리에 저장한다. 노드(3)는 수신된 패킷과 매칭되는 식별 조건을 갖는 패킷 취급 조작에 대해 메모리를 검색한다. 노드(3)가 수신된 패킷과 매칭되는 식별 조건을 갖는 패킷 취급 조작을 발견하면, 검색된 패킷 취급 조작에 의거하여, 노드(3)는 수신된 패킷을 처리한다(예를 들면, 패킷 취급 조작에 의해 지정된 통신 포트를 통해 패킷을 포워딩함).
- [0107] 예를 들면, 플로우 처리부(111)는 패킷 처리 정보에 포함되는 매칭 조건에 의거하여 패킷 취급 조작의 식별 조건을 설정한다.
- [0108] 패킷 처리 정보에 포함된 경로 정보(Path)에 의거하여, 플로우 처리부(111)는 오픈플로우 네트워크에서의 패킷 포워딩 경로를 인식한다. 플로우 처리부(111)는 패킷 취급 조작을 패킷 포워딩 경로 상의 노드(3)에 통지한다. 패킷 취급 조작에 의거하여, 플로우 처리부(111)는 포워딩 경로 상의 각 노드(3)에, 패킷 처리 정보에 대응하는 포워딩 경로를 따라 패킷을 전송하도록 지시할 수 있다. 즉, 플로우 처리부(111)는 노드(3)들에 패킷을 포워딩하도록 지시할 경우, 각 노드(3)에 설정된 패킷 취급 조작은 포워딩 경로에 대응하는 통신 포트를 통해 패킷을 포워딩하기 위한 처리 방법을 포함한다.
- [0109] 패킷 처리 정보가 패킷에 대해 실행될 필요가 있는 미리 정해진 처리(Action)를 포함할 경우, 예를 들면 플로우 처리부(111)는 미리 정해진 처리(Action)를 실행할 필요가 있는 노드(3)를 결정한다. 플로우 처리부(111)는, 미리 정해진 처리로서 결정된 처리(예를 들면, 헤더 재가입 또는 캡슐화/디캡슐화(decapsulation))가 정의되는 패킷 취급 조작을 결정된 노드(3)에 설정한다.
- [0110] 예를 들면, 플로우 처리부(111)는 커맨드 "subscribe"를 사용하여, 네트워크 DB(2)에서 미리 정해진 업데이트가 실행될 경우(패킷 처리 정보 또는 링크 정보의 업데이트), 네트워크 DB(2)는 업데이트에 대한 정보를 플로우 처리부(111)에 통지한다. 커맨드 "subscribe"에 의해 지정된 정보가 업데이트되어 있을 경우, 네트워크 DB(2)는 커맨드 "publish"를 사용하여, 정보를 플로우 처리부(111)에 통지한다. 예를 들면, 커맨드 "publish"를 사용하여 제공된 정보에 의거하여, 플로우 처리부(111)는 오픈플로우 네트워크의 노드(3)에 이미 설정된 패킷 취급 조작을 변경할지의 여부를 판정한다. 패킷 취급 조작의 변경이 필요할 경우에는, 플로우 처리부(111)는 노드(3)에 새로운 패킷 취급 조작을 생성하여 설정한다.
- [0111] 오픈플로우 네트워크의 링크 또는 노드가 삭제될 경우, 예를 들면 플로우 처리부(111)는 삭제된 링크 또는 노드에 대응하는 패킷 처리 정보를 네트워크 DB(2)로부터 삭제한다. 예를 들면, 플로우 처리부(111)는 삭제된 링크

또는 노드가 배치되는 포워딩 경로를 포함하는 패킷 처리 정보를 네트워크 DB(2)로부터 제거한다.

- [0112] 플로우 처리부(111)는 커맨드 "pop_at"을 사용하여, 패킷 처리 정보의 리스트로부터 삭제될 필요가 있는 패킷 처리 정보에 대응하는 식별자를 삭제한다. 플로우 처리부(111)는 커맨드 "delete"를 사용하여, 패킷 처리 정보에 대응하는 엔트리("/flows/[flow_id]")를 삭제한다. 예를 들면, 플로우 처리부(111)는 삭제된 패킷 처리 정보를 대체하는 새로운 패킷 처리 정보를 생성하고, 커맨드 "push_at"을 사용하여, 생성된 패킷 처리 정보를 네트워크 DB(2)에 저장한다.
- [0113] 커맨드 "pop_at"을 사용함으로써, 상기 일련의 조작들을 일괄적으로 실행할 수 있다. 따라서, 커맨드 "delete"의 실행을 생략할 수 있다.
- [0114] 패킷 처리부(112)는 DB 조작 커맨드를 사용하여, DB 인터페이스부(10)를 통해 네트워크 DB(2)를 조작한다. 예를 들면, 패킷 처리부(112)는 네트워크 DB(2)를 조작하여, 네트워크 DB(2)에 패킷에 대한 정보를 설정한다.
- [0115] 도 16은 패킷 처리부(112)가 네트워크 DB(2)를 조작하는 조작 예를 나타낸다. 예를 들면, 오픈플로우 네트워크로부터 처리 방법이 알려져 있지 않은 패킷을 수신할 경우, 패킷 처리부(112)는 플로우 처리부(111)에 패킷에 대응하는 처리 방법에 대해 요청한다. 상술한 바와 같이, 예를 들면, 패킷 처리부(112)는 커맨드 "get_matching_flow(<packet_info>)"를 사용하여, 플로우 처리부(111)에 패킷 처리 방법에 대한 검색을 요청한다.
- [0116] 패킷 처리부(112)가, 플로우 처리부(111)에 의해 패킷에 대응하는 패킷 처리 정보가 존재하지 않음을 통지받을 경우, 패킷 처리부(112)는 패킷에 대한 정보를 네트워크 DB(2)에 등록한다. 예를 들면, 패킷 처리부(112)는 커맨드 "push_tail(/packets/in)"를 사용하여, 엔트리 리스트 "/packets/in"의 말미에 패킷의 식별자를 등록한다. 또한, 패킷 처리부(112)는 커맨드 "put(/packets/in/[packet_id], <packet_info>)"를 사용하여, 커맨드의 인수인 "<packet_info>"에 대응하는 정보를, 엔트리 "/packets/in/[packet_id]"에 추가한다 ([packet_id]는 엔트리 /packets/in에 추가되는 패킷의 식별자임).
- [0117] 상기 "push_tail(/packets/in)" 대신에, 커맨드 "push_tail(/packets/in, <node_info>)"를 사용함으로써, 상기 일련의 조작들을 일괄적으로 실행할 수 있다. 이러한 경우에, 커맨드 "put(/packets/in/[node_id], <packet_info>)"의 실행을 생략할 수 있다.
- [0118] 엔트리 "/packets/in"에 추가된 패킷에 대응하는 패킷 처리 정보가 생성될 경우, 패킷은 엔트리 "/packets/out"에 추가된다. 예를 들면, 패킷 처리부(112)가 네트워크 DB(2)를 미리 설정해서, 커맨드 "subscribe"가 사용될 경우 및 패킷이 엔트리 "/packets/out"에 추가될 경우, 네트워크 DB(2)는 엔트리 "/packets/out"의 업데이트에 대한 정보를 패킷 처리부(112)에 통지한다. 예를 들면, 패킷 처리부(112)가 커맨드 "publish"를 사용하여, 엔트리 "/packets/out"이 업데이트될 경우 패킷 처리부(112)는 네트워크 DB(2)로부터 변경된 키를 수신한다. 이렇게, 패킷 처리부(112)는 엔트리 "/packets/out"에 등록된 패킷의 식별자를 인식할 수 있다. 예를 들면, 플로우 처리부(111)가 오픈플로우 네트워크의 노드에 패킷 처리 조작을 설정할 경우, 패킷 처리부(112)는 수신된 식별자에 대응하는 패킷을 오픈플로우 네트워크에 전송한다. 또한, 패킷 처리부(112)는 커맨드 "pop_at(/packets/out, <packet_id>)"를 사용하여 엔트리 "/packets/out"으로부터 수신된 식별자를 삭제하고, 커맨드 "delete"를 사용하여 엔트리 "/packets/out/[packet_id]"를 제거한다.
- [0119] 상기 커맨드 "pop_at"을 대신하여 커맨드 "pop_head(/packets/out)"이 사용될 경우, 패킷 정보를 "/packets_out"의 선두로부터 수신할 수 있고, 엔트리 "/packets/out/[packet_id]"를 삭제할 수 있다. 따라서, 상기 커맨드 "delete"의 실행을 생략할 수 있다.
- [0120] 도 17 내지 도 20은 제 3 실시예에 따른 동작을 나타내는 순서도이다.
- [0121] 도 17은 노드가 네트워크(4)에 추가될 경우 실행되는 동작을 나타낸다. 토폴로지 처리부(110)는 오픈플로우 네트워크에의 노드 추가를 검출하고, 추가된 노드에 대한 정보를 수집한다.
- [0122] 예를 들면, 토폴로지 처리부(110)는 커맨드 "push_tail(/nodes)"을 사용하여, 노드를 네트워크 DB(2)의 엔트리 리스트 "/nodes"에 추가한다. 리스트에 엔트리 리스트를 추가한 후, 토폴로지 처리부(110)는 네트워크 DB(2)로부터 네트워크 DB(2)에서의 노드의 식별자인 "node_id"를 수신한다. 예를 들면, 토폴로지 처리부(110)는 커맨드 "put"을 사용하여, 노드에 대한 정보를 네트워크 DB(2)의 엔트리들에 추가한다. 예를 들면, 추가된 노드에 대한 정보는 노드에 대한 정보 및 노드의 통신 포트에 대한 정보이다.
- [0123] 도 18은 노드가 네트워크(4)로부터 삭제될 경우 실행되는 동작을 나타낸다. 토폴로지 처리부(110)는 오픈플로

우 네트워크로부터 노드의 삭제를 검출하고 삭제된 노드에 대한 정보를 수집한다.

[0124] 예를 들면, 토폴로지 처리부(110)는 커맨드 "pop_at"을 사용하여, 엔트리 리스트 "/nodes"로부터 삭제된 노드에 대응하는 식별자를 제거한다. 또한, 토폴로지 처리부(110)는 커맨드 "pop_at"을 사용하여, 엔트리 리스트 "/links"로부터 삭제된 노드에 관련된 링크의 식별자를 삭제한다.

[0125] 예를 들면, 토폴로지 처리부(110)는 커맨드 "delete"를 사용하여, 오픈플로우 네트워크로부터 삭제된 노드에 대한 정보 및 삭제된 노드에 관련된 링크에 대한 정보를 삭제한다.

[0126] 도 19는, 네트워크(4)에 링크가 추가될 경우에 실행되는 조작 및 네트워크(4)로부터 링크가 삭제될 경우에 실행되는 조작을 나타낸다. 토폴로지 처리부(110)는 오픈플로우 네트워크에의 링크의 추가를 검출하고 추가된 링크에 대한 정보를 수집한다.

[0127] 예를 들면, 토폴로지 처리부(110)는 커맨드 "push_tail"을 사용하여, 엔트리 "/links"에 링크를 추가한다. 엔트리 리스트 "/links"에 링크를 추가한 후, 토폴로지 처리부(110)는 네트워크 DB(2)로부터 리스트에 추가된 링크의 식별자를 수신한다. 수신된 식별자에 의거하여, 토폴로지 처리부(110)는 커맨드 "put"을 사용하여, 오픈플로우 네트워크에 추가되는 링크에 대한 정보를 엔트리 "/links/[link_id]"에 등록한다.

[0128] 토폴로지 처리부(110)는 오픈플로우 네트워크로부터 링크의 삭제를 검출하고 삭제된 링크에 대한 정보를 수집한다. 예를 들면, 토폴로지 처리부(110)는 커맨드 "delete"를 사용하여, 네트워크 DB(2)로부터 삭제된 링크의 엔트리 "/links/[link_id]"를 삭제한다. 또한, 예를 들면, 토폴로지 처리부(110)는 커맨드 "pop_at"을 사용하여, 엔트리 리스트 "/links"로부터 삭제된 링크의 식별자를 삭제한다.

[0129] 도 20은, 오픈플로우 제어부(11A)가 네트워크 DB(2)에 포함되는 패킷 처리 정보에 의거하여 오픈플로우 네트워크를 제어하는 동작을 나타낸다. 패킷 처리부(112)는 오픈플로우 네트워크로부터 패킷 처리 방법이 알려져 있지 않은 패킷을 수신한다(도 20의 "packet_in"). 예를 들면, 패킷 처리부(112)는 커맨드 "get_matching_flow(<packet_info>)"를 사용하여, 수신된 패킷에 대응하는 패킷 처리 정보에 대해 플로우 처리부(111)에 질의한다.

[0130] 예를 들면, 플로우 처리부(111)는, "get_matching_flow(<packet_info>)"에 포함된 패킷 정보("packet_info")와 매칭되는 매칭 조건을 갖는 엔트리 ("/flows/[flow_id]")에 대해 네트워크 DB(2)를 검색한다.

[0131] 플로우 처리부(111)가 패킷 처리부(112)에 의해 수신된 패킷과 매칭되는 패킷 처리 정보를 발견한 경우, 예를 들면 플로우 처리부(111)는 네트워크(4)의 노드(3)에 설정될 패킷 취급 조작(처리 규칙)을 생성하고 패킷 취급 조작을 패킷 처리부(112)를 통해 노드(3)에 통지한다. 노드(3)에 패킷 취급 조작을 통지하는 것에 이외에도, 패킷 처리부(112)는 네트워크(4)로부터 수신된 패킷을 네트워크(4)에 반환한다(도 20의 "packet_out").

[0132] 플로우 처리부(111)가 패킷 처리부(112)에 의해 수신된 패킷과 매칭되는 패킷 처리 정보를 발견하지 못할 경우, 플로우 처리부(111)는 커맨드 "push_tail"을 사용하여, 패킷 처리부(112)에 의해 수신된 패킷을 엔트리 "/packets/in"에 추가한다. 엔트리 "/packets/in"에 패킷을 추가한 후에, 플로우 처리부(111)는 네트워크 DB(2)로부터 엔트리에 추가된 패킷의 식별자를 수신한다(도 20의 "packet_id"). 수신된 식별자에 의거하여, 플로우 처리부(111)는 커맨드 "put"을 사용하여, 패킷 처리부(112)에 의해 수신된 패킷에 대한 정보를 엔트리 "/packets/in/[packet_id]"에 추가한다.

[0133] <제 4 실시예>

[0134] 다음으로, 본 발명의 제 4 실시예를 개시한다. 제 4 실시예에서, 제어 장치(1)가 가시화 DB 조작 로직부(11B)를 사용하여 네트워크 DB(2)를 조작하는 예를 설명한다.

[0135] 제어 장치(1)로 네트워크(4)를 관리하는 오퍼레이터를 지원하기 위해, 가시화 DB 조작 로직부(11B)는, 정보를 시각적으로 파악할 수 있도록 네트워크 DB(2)에 저장된 정보의 적어도 일부를 그래픽적으로 표시하는 기능을 갖는다.

[0136] 도 21은 제 4 실시예에 따른 구성을 나타낸다. 도 21의 가시화 DB 조작 로직부(11B)는 네트워크 DB(2)로부터 토폴로지 정보를 취득하고, 디스플레이 디바이스 상에 노드들 및 링크들의 접속 관계를 표시한다. 예를 들면, 도 22에 나타난 바와 같이, 가시화 DB 조작 로직부(11B)는 네트워크 토폴로지를 그래픽적으로 표시한다. 각 노드에 할당된 번호는 노드의 식별자를 나타낸다. 또한, 각 링크에 할당된 번호는 링크의 식별자를 나타낸다. 네트워크 DB(2)가 가시화 DB 조작 로직부(11B)에 토폴로지 정보의 변화를 통지할 경우, 이 통지에 의거하여, 가

시화 DB 조작 로직부(11B)는 네트워크 DB(2)로부터 노드 및 링크 정보를 취득하고 노드들 및 링크들의 표시된 접속 관계를 변경한다.

[0137] 이렇게, 도 21의 가시화 DB 조작 로직부(11B)를 네트워크 DB(2)에 접속함으로써, 오퍼레이터는 관리 대상 네트워크의 구성을 시각 정보로서 취득할 수 있다. 또한, 가시화 DB 조작 로직부(11B)는 네트워크 DB(2)의 패킷 처리 정보를 그래픽적으로 표시할 수 있다. 예를 들면, 가시화 DB 조작 로직부(11B)는 엔트리 `"/flows/[flow_id]"`에 포함된 경로 정보(Path)를 그래픽적으로 표시되는 토폴로지에 겹쳐 놓는다. 도 23은 그래픽적으로 표시된 패킷 처리 정보를 나타낸다. 도 23에서, 식별자 "10"을 갖는 엔트리 `"/flows/10"`가 그래픽으로 표시되어 있다. 패킷들이 이동하는 노드들 및 링크들에 대한 정보 및 패킷들이 포워딩되는 방향에 대한 정보가 그래픽적으로 표시된다.

[0138] 도 24에 나타난 바와 같이, 가시화 DB 조작 로직부(11B)는 엔트리들 `"/packets/in/[packet_id]"` 및 `"/packets/out/[packet_id]"`을 그래픽적으로 표시할 수 있다.

[0139] <제 5 실시예>

[0140] 다음으로, 본 발명의 제 5 실시예에 대해 설명한다. 제 5 실시예에서, 제어 장치(1)가 슬라이스(slice) 로직부(11C)를 사용하여 네트워크 DB(2)를 조작하는 예를 나타낸다.

[0141] 도 25는 제 5 실시예에 따른 구성을 나타낸다. 도 25의 슬라이스 로직부(11C)는 네트워크 DB(2)를 복제하여 네트워크 슬라이스(가상 분할)를 행하는 기능을 갖는다. 슬라이스 로직부(11C)는 단일 네트워크 DB(2)의 상태 또는 변화를 복수의 네트워크 DB(2A 및 2B)로 확장하고 다시 복수의 네트워크 DB(2A 및 2B)의 상태 또는 변화를 원래의 네트워크 DB(2)로 피드백한다. 도 25에서, 제어 장치(1)는 2 개의 네트워크 DB 인터페이스부들(10-1, 10-2)을 포함한다. 그러나, 네트워크 DB 인터페이스부는 각 네트워크 DB마다 배치될 수 있다. 또는, 단일 네트워크 DB 인터페이스부가 복수의 네트워크 DB(2)에 접속될 수 있다.

[0142] 상술한 바와 같이, 임의의 네트워크 DB(2)들에 슬라이스 로직부(11C)를 접속함으로써, 복수의 네트워크 오퍼레이터(네트워크 사용자) 각각은 슬라이스된 네트워크를 전용 네트워크로서 관리할 수 있다. 예를 들면, 상위 네트워크 오퍼레이터는 네트워크 DB(2)를 관리할 수 있고 상위 네트워크 오퍼레이터에 의해 허용된 하위 네트워크 오퍼레이터는 네트워크 DB들(2A 및 2B)을 관리할 수 있다.

[0143] <제 6 실시예>

[0144] 다음으로, 본 발명의 제 6 실시예에 대해 설명한다. 제 6 실시예에서, 제어 장치(1)가 최단 경로 로직부(11D)를 이용하여 네트워크 DB(2)를 조작하는 예를 설명한다. 최단 경로 로직부(11D)는 네트워크(4)의 패킷 포워딩 송신처로부터 수신처에의 경로 중 최단 경로를 계산하는 기능을 갖는다.

[0145] 도 26은 제 6 실시예에 따른 구성을 나타내는 도면이다. 네트워크 DB(2)로부터 취득한 토폴로지 정보 및 호스트 정보 기억부(21)로부터 판독된 호스트 정보에 의거하여, 도 26의 최단 경로 로직부(11D)는 임의의 호스트들 사이의 최단 포워딩 경로를 계산하고 최단 포워딩 경로를 최단 경로 기억부(20)에 저장한다.

[0146] 네트워크 DB(2)가 최단 경로 로직부(11D)에 토폴로지 관련 엔트리의 업데이트를 통지할 경우, 업데이트의 내용에 의거하여, 최단 경로 로직부(11D)는 최단 경로를 재계산하고 최단 경로 기억부(20)를 업데이트한다. 예를 들면, 최단 경로 로직부(11D)는 커맨드 "subscribe"를 사용하여, 엔트리 `"/nodes"` 또는 `"/links"`가 업데이트될 경우, 네트워크 DB(2)는 최단 경로 로직부(11D)에 업데이트된 정보를 통지한다.

[0147] 커맨드 "subscribe"에 의해 지정된 엔트리가 업데이트될 경우, 네트워크 DB(2)는 커맨드 "publish"를 사용하여, 최단 경로 로직부(11D)에 엔트리에 대한 정보를 통지한다. 네트워크 DB(2)로부터 토폴로지 정보의 업데이트에 대한 통지를 수신할 경우, 최단 경로 로직부(11D)는 새로운 토폴로지 정보에 대응하는 최단 경로를 재계산하고 최단 경로를 최단 경로 기억부(20)에 저장한다. 최단 경로 로직부(11D)는, 최단 경로의 재계산 시 패킷 처리 정보를 업데이트할 수 있다. 예를 들면, 패킷 처리 정보에 포함된 패킷 정보("Path information")는 최단 경로의 재계산 시에 업데이트될 수 있다. 예를 들면, 최단 경로 로직부(11D)는 커맨드 "put"을 사용하여, 업데이트된 패킷 처리 정보를 네트워크 DB(2)의 엔트리 `"/flows/[flow_id]"`에 저장한다.

[0148] 상술한 바와 같이, 도 26의 최단 경로 로직부(11D)를 갖는 제어 장치(1)를 네트워크 DB(2)에 접속함으로써, 최단 경로를 계산하는 기능 및 이 최단 경로에 대응하는 패킷 처리 정보를 업데이트하는 기능이 제어 장치(1)에 추가된다.

- [0149] <제 7 실시예>
- [0150] 다음으로, 본 발명의 제 7 실시예에 대해 설명한다. 제 7 실시예에서, 제어 장치(1)가 애그리게이션 로직부(11E)를 사용하여 네트워크 DB(2)를 조작하는 예를 설명한다. 애그리게이션 로직부(11E)는 복수의 노드들 및 복수의 링크들을 집성하여 네트워크 토폴로지를 추상화하는 기능을 갖는다.
- [0151] 도 27은 제 7 실시예에 따른 구성을 나타낸다. 예를 들면, 도 27의 애그리게이션 로직부(11E)는 네트워크 DB(2)에 저장된 토폴로지 정보를 추상화하고 추상화된 토폴로지 정보를 네트워크 DB(2C)에 저장한다.
- [0152] 예를 들면, 애그리게이션 로직부(11E)는 DB 인터페이스부들(10-3 및 10-4)을 통해 네트워크 DB들(2 및 2C)을 각각 조작한다.
- [0153] 도 28은, 애그리게이션 로직부(11E)가 네트워크 DB(2)의 토폴로지 정보를 추상화하고 추상화된 토폴로지 정보를 네트워크 DB(2C)에 저장하는 동작을 설명한다. 예를 들면, 애그리게이션 로직부(11E)는 오퍼레이터에 의한 조작에 따라 토폴로지 정보를 추상화한다. 도 28에서, 네트워크 DB(2)에 저장된 토폴로지 정보 중에서, 애그리게이션 로직부(11E)는 노드들(1 및 2)을 애그리게이트하고 추상화하고, 노드들(3 및 4)을 애그리게이트하고 추상화한다. 노드들(1 및 2)을 애그리게이트함으로써 생성되는 새로운 노드는 네트워크 DB(2C)에 저장되는 노드(1)에 대응한다. 노드들(3 및 4)을 애그리게이트함으로써 생성되는 새로운 노드는 네트워크 DB(2C)에 저장되는 노드(2)에 대응한다.
- [0154] 도 29 및 도 30은 각각 네트워크 DB들(2 및 2C)의 엔트리들을 나타낸다. 애그리게이션 로직부(11E)는 도 29의 엔트리들 "/nodes/1" 및 "/nodes/2"를 애그리게이트한다. 또한, 애그리게이션 로직부(11E)는 도 29의 엔트리들 "/nodes/3" 및 "/nodes/4"를 애그리게이트한다. 이러한 애그리게이트 작업들에 의해, 네트워크 DB(2C)에 저장되는 노드들의 수는 "3"으로 추상화된다.
- [0155] 예를 들면, 애그리게이션 로직부(11E)는 커맨드 "push"를 사용하여, 네트워크 DB(2C)의 엔트리 "/nodes"에 3개의 추상화된 노드들을 추가한다. 도 30에 나타난 바와 같이, 애그리게이션 동작들에 의해, 네트워크 DB(2C)의 엔트리 "/nodes"의 데이터(Value)는 "{1, 2, 3}"으로 업데이트된다. 예를 들면, 애그리게이션 로직부(11E)는 커맨드 "put"을 사용하여, 추상화된 노드들 각각에 대한 정보를 네트워크 DB(2C)의 엔트리 "/nodes/[node_id]"에 저장한다. 예를 들면, 애그리게이션 로직부(11E)는, 제어 장치(1)가 노드들을 제어할 때 사용되는 ID들(도 29 및 30의 "node ID")을 엔트리 "/nodes/[node_id]"에 등록한다. 또한, 엔트리 "/nodes/[node_id]"에 등록되어 있는 노드가 추상화된 노드일 경우, 애그리게이션 로직부(11E)는, 노드가 추상화되기 전의(도 30에서 "원래의") ID를 엔트리 "/nodes/[node_id]"에 등록한다.
- [0156] 도 30에서, "노드 ID(a)" 및 "노드 ID(b)"가 추상화에 의해 생성된 2개의 노드들의 ID들로서 할당된다. 또한, 네트워크 DB(2)의 엔트리 "/nodes/5"에 대응하는 노드가 네트워크 DB(2C)의 엔트리 "/nodes/3"에 등록된다. "node ID(c)"는 네트워크 DB(2C)에 엔트리 "/nodes/3"의 노드 ID로서 할당된다.
- [0157] 예를 들면, 애그리게이션 로직부(11E)는 노드들을 통합할 때 대응하는 링크들도 애그리게이트한다. 예를 들면, 도 29 및 도 30에서, 애그리게이션 로직부(11E)는 도 29의 "/links/2" 및 "/links/4"에 대응하는 링크들을 도 30의 "/links/1"로 통합한다. 예를 들면, 애그리게이션 로직부(11E)는 커맨드 "put"을 사용하여, 도 29의 "/links/2" 및 "/links/4"에 대응하는 데이터(Value)를 네트워크 DB(2C)의 엔트리 "links/1"에 저장한다. 또한, 애그리게이션 로직부(11E)는 도 29의 "/nodes/1" 및 "/nodes/2"의 애그리게이션에 의해 더 이상 필요하지 않은 링크(도 29의 "/links/1")를 네트워크 DB(2C)에 등록하지 않는다. 예를 들면, 도 29에서, "{src_node : A, dst_node : B}"는 패킷이 노드 ID(A)를 갖는 노드로부터 노드 ID(B)를 갖는 노드에 포워딩되는 링크를 나타낸다.
- [0158] 상술한 바와 같이, 애그리게이션 로직부(11E)는 토폴로지를 추상화할 수 있다. 즉, 오퍼레이터는 추상화된 토폴로지에 의거하여 관리 대상 네트워크를 제어할 수 있고 용이하게 네트워크를 관리할 수 있다.
- [0159] <제 8 실시예>
- [0160] 다음으로, 본 발명의 제 8 실시예에 대해 설명한다. 제 8 실시예에서, 제어 장치(1)가 결합 로직부(11F)를 사용하여 네트워크 DB(2)를 조작하는 예를 설명한다. 결합 로직부(11F)는 복수의 네트워크 DB에 포함되는 토폴로지 정보를 결합하는 기능을 갖는다.
- [0161] 도 31은 제 8 실시예에 따른 구성을 나타낸다. 도 31의 결합 로직부(11F)는 네트워크 DB들(2D 및 2E)에 포함되

는 토폴로지 정보를 결합하고 결합된 토폴로지 정보를 네트워크 DB(2F)에 저장한다.

[0162] 도 32는 네트워크 DB(2D)에 저장된 토폴로지 정보 및 네트워크 DB(2E)에 저장된 정보를 나타낸다. 네트워크 DB(2D)는 3 개의 노드들(노드 1 내지 3) 및 3 개의 링크들(링크 1 내지 3)로 구성된 토폴로지 정보를 포함한다. 네트워크 DB(2E)는 3 개의 노드들(노드 1 내지 3) 및 3 개의 링크들(링크 1 내지 3)로 구성된 토폴로지 정보를 포함한다.

[0163] 도 33은 네트워크 DB들(2D 및 2E)의 토폴로지 정보를 결합하여 생성된 새로운 토폴로지 정보를 나타낸다.

[0164] 도 32 및 도 33에 나타난 바와 같이, 2 개의 토폴로지들을 결합함으로써, 6 개의 노드들이 네트워크 DB(2F)에 저장된다. 예를 들면, 결합 로직부(11F)는 커맨드 "push"를 사용하여, 엔트리 "/nodes"에 노드 리스트를 저장한다. 결합 로직부(11F)가 커맨드 "push"로 네트워크 DB(2F)를 조작하도록 함으로써, 결합 노드들의 리스트 ("{1, 2, 3, 4, 5, 6}")가 엔트리 "/nodes"에 저장된다. 또한, 예를 들면, 결합 로직부(11F)는 커맨드 "put"을 사용하여, 노드들에 대한 정보를 엔트리 "/nodes/[node_id]"에 저장한다.

[0165] 도 32 및 도 33에 나타난 바와 같이, 2 개의 토폴로지를 결합함으로써, 7 개의 링크들이 네트워크 DB(2F)에 저장된다. 예를 들면, 결합 로직부(11F)는 커맨드 "push"를 사용하여, 링크 리스트를 엔트리 "/links"에 저장한다. 결합 로직부(11F)가 커맨드 "push"로 네트워크 DB(2F)를 조작하게 함으로써, 결합 노드들의 리스트 ("{1, 2, 3, 4, 5, 6, 7}")가 엔트리 "/links"에 저장된다. 또한, 예를 들면, 결합 로직부(11F)는 커맨드 "put"을 사용하여 링크들에 대한 정보를 엔트리 "/links/[link_id]"에 저장한다.

[0166] <제 9 실시예>

[0167] 다음으로, 본 발명의 제 9 실시예에 대해 설명한다. 오퍼레이터는 컴포넌트 매니저(30)의 기능들을 이용함으로써, 네트워크 DB(2)들 또는 DB 조작 로직부(11)들(또는 제 3 실시예 내지 제 8 실시예에서 설명된 컴포넌트들(11A 내지 11F)을 관리할 수 있다. 제 9 실시예에서는, 컴포넌트 매니저(30)를 설명한다.

[0168] 도 34는 제 9 실시예에 따른 시스템의 구성을 나타낸다. 오퍼레이터는 컴포넌트 매니저(30)의 기능들을 사용하여 네트워크 DB(2)들 및 DB 조작 로직부(11)들을 관리한다. 또한, 도 34에 나타난 바와 같이, 네트워크 DB(2)들 및 DB 조작 로직부(11)들이 복수의 서버들(서버들(200 내지 400))에 배치될 경우에도, 컴포넌트 매니저(30)는 네트워크 DB(2)들 및 DB 조작 로직부(11)들을 통합 관리할 수 있다. 도 34에서, 제어 장치(1)는 복수의 서버에 배치된 네트워크 DB(2)들 및 DB 조작 로직부(11)들에 의해 구성된다.

[0169] 도 35는 컴포넌트 매니저(30)의 구성을 나타낸다. 도 35에서, 컴포넌트 매니저(30)는 컴포넌트 정보 기억부(31), 컴포넌트 관리부(32), 외부 제어 인터페이스부(33), 및 컴포넌트 제어 인터페이스부(34)를 포함한다.

[0170] 컴포넌트 정보 기억부(31)는 상기의 실시예에서 설명한 네트워크 DB(2)들 및 DB 조작 로직부(11)들(예를 들면, 제 3 내지 제 8 실시예에 기재된 컴포넌트들(11A-11F))의 구성 및 접속 관계를 저장한다.

[0171] 컴포넌트 제어 인터페이스부(34)는 네트워크의 DB(2)들, DB 조작 로직부(11)들 등에 접속하기 위한 인터페이스로서의 기능을 갖는다. 컴포넌트 매니저(30)는 컴포넌트 제어 인터페이스부(34)를 통해 네트워크 DB(2)들 및 DB 조작 로직부(11)들에 액세스한다.

[0172] 컴포넌트 관리부(32)는 오퍼레이터로부터의 조작에 응답하여, 네트워크 DB(2)를 생성 및 삭제하고 DB 조작 로직부(11)를 개시 및 종료하기 위한 개시 및 종료 기능(320)을 갖는다. 컴포넌트 관리부(32)는 오퍼레이터로부터의 조작에 응답하여, 컴포넌트 제어 인터페이스부(34)를 통해 컴포넌트들을 접속하기 위한 접속 기능(321)을 갖는다. 컴포넌트 관리부(32)는 DB 조작 로직부(11)들이 적절하게 조작되고 있는지의 여부를 감시하는 사활(死活) 감시 기능(322)을 포함한다. 또한, 컴포넌트 관리부(32)는 오퍼레이터로부터의 조작에 의거하여 컴포넌트 정보 기억부(31)의 내용을 업데이트한다.

[0173] 도 34의 프로세스 관리부(9)들은 컴포넌트 제어 인터페이스부(34)에 접속된다. 예를 들면, 도 34에 나타난 바와 같이, DB 조작 로직부(11)가 복수의 서버 상에서 구성될 경우, 프로세스 관리부(9)들은, 컴포넌트 관리부(32)로부터의 지시들에 의거하여, 서버들 상에서 동작하는 DB 조작 로직부(11)를 개시, 종료, 및 감시하는 것을 포함하는 관리 조작들을 실행한다. 또한, 도 34에 나타난 바와 같이, 복수의 DB 조작 로직부(11)는 서로 다른 네트워크들(예를 들면, 오픈플로우 네트워크 및 MPLS(Multi-Protocol Label Switching) 네트워크)에 접속될 수 있다.

[0174] <제 10 실시예>

- [0175] 다음으로, 본 발명의 제 10 실시예에 대해 설명한다. 도 36은 본 발명의 제 10 실시예에 따른 시스템의 구성을 나타낸다. 다음으로, 도 36의 사용자 인터페이스(40)를 도면을 참조하여 설명한다.
- [0176] 도 36에 나타난 서버(정보 처리 장치)(400)는, 오퍼레이터에 의해 사용되는 사용자 인터페이스(40)의 조작에 의거하여 제어 장치(1)를 구성하는 기능을 갖는다.
- [0177] 서버(400)는 표시부(401) 및 관리부(402)를 포함한다. 표시부(401)는 사용자 인터페이스(40)(예를 들면, 디스플레이 상에 표시되는 화면)에 네트워크 DB(2) 및 DB 조작 로직부(11)에 대응하는 아이콘들(아이콘들은 네트워크 DB(2)에 대응하는 제 1 아이콘 및 DB 조작 로직부(11)들에 대응하는 제 2 아이콘들)을 표시하는 기능을 갖는다. 관리부(402)는 오퍼레이터에 의해 조작되는 아이콘들의 접속 관계에 의거하여 제어 장치(1)를 구성하는 기능을 갖는다.
- [0178] 도 37은 표시부(401)에 의해 표시되는 화면을 나타낸다. 도 37의 화면은 상기 네트워크 DB(2) 및 DB 조작 로직부(11)들(예를 들면, 제 3 내지 제 8 실시예에 나타난 컴포넌트들(11A-11F))에 대응하는 템플릿 아이콘들을 표시하는 템플릿 표시 영역 및 작업 영역을 포함하는 레이아웃을 갖는다.
- [0179] 예를 들면, 오퍼레이터가 도 37에서 템플릿 표시 영역으로부터 필요한 네트워크 DB(2)들, 템플릿 DB 조작 로직부(11)들, 및 최단 경로 로직부(11D)에 대응하는 템플릿들을 드래그 앤 드롭할 경우, 대응하는 아이콘들이 작업 영역에 표시된다. 오퍼레이터가 작업 영역에 표시되는 아이콘들을 접속할 경우, 작업 영역의 아이콘들 사이에 접속 선이 표시된다. 예를 들면, 표시부(401)가 오퍼레이터로부터 표시 요청을 접수할 경우, 팝업 윈도우 등을 이용한 오브젝트 표시 영역에 네트워크 DB(2) 또는 DB 조작 로직부(11)에 대한 세부 정보나 가시화 DB 조작 로직부(11B)에 의해 생성되는 네트워크 토폴로지가 표시된다. 예를 들면, 표시부(401)가 작업 영역의 아이콘의 클릭(표시 요청) 등을 검출했을 경우, 팝업 윈도우 등을 이용한 오브젝트 표시 영역에는 네트워크 DB(2) 또는 DB 조작 로직부(11)에 대한 세부 정보나 가시화 DB 조작 로직부(11B)에 의해 생성되는 네트워크 토폴로지가 표시된다.
- [0180] 도 38은 사용자 인터페이스(40)를 통해 실행되는 조작 및 관리부(402)에 의해 구성되는 제어 장치(1)의 구성을 서로 연관시킨 도면이다. 관리부(402)는 컴포넌트 매니저(30)에 의해 구성될 수 있다.
- [0181] 예를 들면, 표시부(401)가 DB 조작 로직부(11)에 의해 네트워크 DB(2)의 조작 결과들을 표시할 수 있다. 예를 들면, DB 조작 로직부(11)에 대응하는 아이콘 및 네트워크 DB(2)에 대응하는 아이콘이 서로 접속될 경우, 표시부(401)는 DB 조작 로직부(11)에 의한 네트워크 DB(2)의 조작 결과를 표시한다.
- [0182] 도 38은 아이콘(NW-DB; 네트워크 DB(2)에 대응함) TP1, 아이콘(OF-NW; 오픈플로우 제어부(11A)에 대응함) TP2, 및 아이콘(최단 경로; 최단 경로 로직부(11D)에 대응함) TP3이 템플릿 표시 영역으로부터 선택된 상태를 나타낸다. 이 아이콘들이 작업 영역에 배치되어 서로 접속된다. 이렇게, 네트워크 DB(2), 오픈플로우 제어부(11A), 및 최단 경로의 로직부(11D)를 제어 장치(1)에 추가하고 접속한다. 네트워크 DB(2)에 대응하는 네트워크(4)는 오픈플로우 프로토콜에 의해 제어되고, 네트워크(4)에서 최단 경로 제어가 실행된다.
- [0183] 도 39는 도 38의 아이콘(가시화; 가시화 DB 조작 로직부(11B)에 대응함) TP4가 아이콘(NW-DB; 네트워크 DB(2)에 대응함) TP1에 접속된다. 이렇게, 그래프 등을 이용한 네트워크 토폴로지 표시 기능이 제어 장치(1)에 추가된다.
- [0184] 도 40은, 아이콘(OF-NW) TP2가 접속되는 아이콘(NW-DB) TP1에 아이콘(애그리게이션; 애그리게이션 로직부(11E)) TP5가 추가적으로 접속되는 그래프이다. 아이콘(애그리게이션) TP5에 대응하는 애그리게이션 로직부(11E)는 네트워크 DB의 토폴로지 정보를 애그리게이션해서 추상화하고 그 정보를 다른 네트워크 DB(2)에 저장한다. 아이콘(가시화) TP5는 각 네트워크 DB(2)에 접속된다. 도 40에 나타난 바와 같이, 복잡한 네트워크 토폴로지 및 추상화된 네트워크 토폴로지 모두를 그래프로서 표시할 수 있다.
- [0185] 도 41은, 아이콘(OF-NW) TP2가 접속된 아이콘(NW-DB) TP1에 아이콘(슬라이스; 슬라이스 로직부(11C)에 대응함) TP6가 추가적으로 접속되고, 논리적으로 분할된 네트워크 DB들이 생성되는 그래프이다. 이렇게, 단일 네트워크가 복수의 가상 네트워크들로 슬라이스될 수 있다. 또한, 이 슬라이스된 네트워크 DB들을 나타내는 아이콘들(NW-DB; 네트워크 DB들)에 템플릿 표시 영역의 다양한 아이콘들을 접속함으로써, 슬라이스된 네트워크들이 개별적으로 제어 및 사용될 수 있다.
- [0186] 도 42는, 아이콘(OF-NW) TP2가 접속되는 복수의 아이콘들(NW-DB) TP1에 아이콘(결합; 결합 로직부(11F)에 대응함) TP7이 접속되고 이 네트워크들을 결합하는 네트워크 DB가 생성되는 그래프이다. 이렇게, 복수의 네트워크

들이 단일 네트워크로서 제어 및 사용될 수 있다.

- [0187] 상술한 바와 같이, 본 실시예에 따른 사용자 인터페이스를 사용함으로써, 오퍼레이터는 고도의 지식 및 기술 없이 초기 설정들 또는 네트워크 제어 장치의 기능들을 커스터마이징할 수 있다. 또한, 오퍼레이터는 사용자 인터페이스(40)를 통해 각 네트워크를 설정하고, 네트워크 DB의 엔트리를 참조하고, 엔트리를 조작하는 것이 가능해질 수 있다. 예를 들면, 도 37의 오브젝트 표시 영역의 필드에 커서를 움직임으로써 오퍼레이터는 설정 값을 수정하거나, 제어(플로우) 조작에 관련된 엔트리를 추가, 변경, 또는 삭제 가능한 것이 바람직하다.
- [0188] 이와 같이 본 발명의 예시적인 실시예를 설명했지만, 본 발명은 이에 한정되는 것은 아니다. 추가적인 수정, 치환, 또는 조정이 본 발명의 기본적인 기술적 사상에서 벗어나지 않고 이루어질 수 있다. 예를 들면, 각 도면에서의 네트워크 구성 또는 컴포넌트 구성은 단순히 본 발명의 이해를 용이하게 하기 위해 사용된다. 즉, 본 발명은 도면에서의 이 구성들에 한정되는 것은 아니다.
- [0189] 또한, 예를 들면, 상기 실시예들에서 설명한 DB 조작 로직부(11)의 변형 이외에도 네트워크 추출 로직을 적용할 수 있다. 이러한 로직으로, 단일 네트워크 DB의 토폴로지를 부분적으로 추출하고 추출된 토폴로지를 다른 네트워크 DB로 확장함으로써, 네트워크 DB의 플로우가 원래 네트워크 DB의 플로우로 확장될 수 있다. 이렇게 2개의 네트워크 DB를 접속함으로써, 네트워크를 부분적으로 제어하는 기능이 네트워크 제어 장치에 추가된다. 예를 들면, 다른 네트워크 오퍼레이터가 네트워크의 부분만을 조작할 수 있게 하거나 그 부분만의 라우팅 알고리즘을 변경할 수 있게 할 수 있다.
- [0190] 또한, 상기 실시예들은, 실제 네트워크를 제어하는 것을 상정하여 설명했지만, NS3(Network Simulator version3) 등의 네트워크 시뮬레이터와 링크되는 네트워크 제어부가 배치될 수 있다. 이렇게, 네트워크 제어 장치는 실제 물리적 네트워크뿐만 아니라 시뮬레이터 상의 네트워크를 제어할 수 있다. 따라서, 네트워크 오퍼레이터는 제어 장치를 실제 물리적 네트워크에 적용하기 전에 시뮬레이터 상에서 생성되는 네트워크 제어 장치의 동작 또는 성능을 테스트할 수 있다.
- [0191] 마지막으로, 본 발명의 바람직한 형태를 요약한다.
- [0192] <제 1 형태>
- [0193] (상술한 제 1 양태의 정보 처리 장치를 참조)
- [0194] <제 2 형태>
- [0195] 제 1 형태의 정보 처리 장치에 있어서, 상기 제 1 아이콘에 대한, 상기 데이터베이스의 상태 표시의 요청에 따라, 상기 표시부는 상기 데이터베이스에 저장된 제어 정보를 표시한다.
- [0196] <제 3 형태>
- [0197] 제 1 형태 또는 제 2 형태의 정보 처리 장치에 있어서, 상기 제 1 아이콘과 어느 하나의 상기 제 2 아이콘과의 접속에 따라, 상기 표시부는, 상기 제 2 아이콘에 대응하는 모듈에 의해 발행되는 데이터베이스 조작 커맨드에 응답하여 제어 정보를 조작해서 얻어지는 결과를 표시한다.
- [0198] <제 4 형태>
- [0199] 제 1 형태 내지 제 3 형태 중 어느 하나의 정보 처리 장치에 있어서, 상기 제 1 아이콘과 상기 네트워크의 상태의 표시를 실행하는 모듈에 대응하는 상기 제 2 아이콘과의 접속에 따라, 상기 표시부는 상기 네트워크의 상태를 표시한다.
- [0200] <제 5 형태>
- [0201] 제 1 형태 내지 제 4 형태 중 어느 하나의 정보 처리 장치에 있어서, 상기 제어 정보에 대한 조작은, 상기 복수의 모듈에 의해 사용되는 공통의 데이터베이스 조작 커맨드에 응답하여 실행된다.
- [0202] <제 6 형태>
- [0203] 제 1 형태 내지 제 5 형태 중 어느 하나의 정보 처리 장치에 있어서, 상기 데이터베이스는 상기 네트워크의 토폴로지에 대한 제 1 정보 및 상기 네트워크에서의 패킷 처리 방법에 대한 제 2 정보를 포함한다.
- [0204] <제 7 형태>
- [0205] 제 1 형태 내지 제 6 형태 중 어느 하나의 정보 처리 장치에 있어서, 상기 데이터베이스는, 상기 제어 장치가

상기 네트워크로부터 수신하는 패킷을 관리하기 위한 제 3 정보, 및 상기 제어 장치가 상기 네트워크에 전송하는 패킷을 관리하기 위한 제 4 정보를 포함한다.

[0206] <제 8 형태>

[0207] 제 1 형태 내지 제 7 형태 중 어느 하나의 정보 처리 장치에 있어서, 상기 제 2 아이콘은, 통신 단말들 사이의 경로 계산, 네트워크 토폴로지의 관리, 네트워크 토폴로지의 애그리게이션(aggreat ion), 네트워크의 분할, 및 네트워크의 결합으로 구성되는 그룹에서 선택되는 적어도 하나를 실행하는 모듈에 대응한다.

[0208] <제 9 형태>

[0209] (상술한 제 2 양태의 구성 방법을 참조)

[0210] <제 10 형태>

[0211] 제 9 형태의 구성 방법에 있어서, 상기 제 1 아이콘에 대한, 상기 데이터베이스의 상태 표시의 요청에 따라, 상기 데이터베이스에 저장된 제어 정보가 상기 표시부에 표시된다.

[0212] <제 11 형태>

[0213] 제 9 또는 제 10 형태의 구성 방법에 있어서, 상기 제 1 아이콘과 어느 하나의 상기 제 2 아이콘과의 접속에 따라, 상기 제 2 아이콘에 대응하는 모듈에 의해 발행되는 데이터베이스 조작 커맨드에 응답하여 제어 정보를 조작해서 얻어지는 결과가 상기 표시부에 표시된다.

[0214] <제 12 형태>

[0215] 제 9 내지 제 11 형태 중 어느 하나의 구성 방법에 있어서, 상기 제 1 아이콘과 상기 네트워크의 상태의 표시를 실행하는 모듈에 대응하는 상기 제 2 아이콘과의 접속에 따라, 상기 네트워크의 상태가 상기 표시부에 표시된다.

[0216] <제 13 형태>

[0217] 제 9 내지 제 12 형태 중 어느 하나의 구성 방법에 있어서, 상기 복수의 모듈에 의해 사용되는 공통의 데이터베이스 조작 커맨드에 응답하여, 상기 제어 정보에 대한 조작이 실행된다.

[0218] <제 14 형태>

[0219] 제 9 내지 제 13 형태 중 어느 하나의 구성 방법에 있어서, 상기 데이터베이스는 상기 네트워크의 토폴로지에 대한 제 1 정보 및 상기 네트워크에서의 패킷 처리 방법에 대한 제 2 정보를 포함한다.

[0220] <제 15 형태>

[0221] 제 9 내지 제 14 형태 중 어느 하나의 구성 방법에 있어서, 상기 데이터베이스는, 상기 제어 장치가 상기 네트워크로부터 수신하는 패킷을 관리하기 위한 제 3 정보, 및 상기 제어 장치가 상기 네트워크에 전송하는 패킷을 관리하기 위한 제 4 정보를 포함한다.

[0222] <제 16 형태>

[0223] 제 9 내지 제 15 형태 중 어느 하나의 구성 방법에 있어서, 상기 제 2 아이콘은, 통신 단말들 사이의 경로 계산, 네트워크 토폴로지의 관리, 네트워크 토폴로지의 애그리게이션, 네트워크의 분할, 및 네트워크의 결합으로 구성되는 그룹에서 선택되는 적어도 하나를 실행하는 모듈에 대응한다.

[0224] <제 17 형태>

[0225] (상술한 제 3 양태의 통신 시스템을 참조)

[0226] <제 18 형태>

[0227] (상술한 제 4 양태의 프로그램을 참조)

[0228] 위 비특허문헌의 공개 전체는 참조에 의해 본원에 통합된다. 실시예 및 예의 수정 및 조정이 본 발명의 전체 공개(특허청구범위를 포함함)의 범주 내에서 본 발명의 기본적인 기술적 사상에 의거하여 가능하다. 다양한 개시 요소들(특허청구범위, 실시예, 예, 도면 등 각각에서의 요소들을 포함함)의 다양한 조합 및 선택이 본 발명의 특허청구범위의 범주 내에서 가능하다. 즉, 본 발명이 특허청구범위 및 기술적 사상을 포함하는 개시 전체

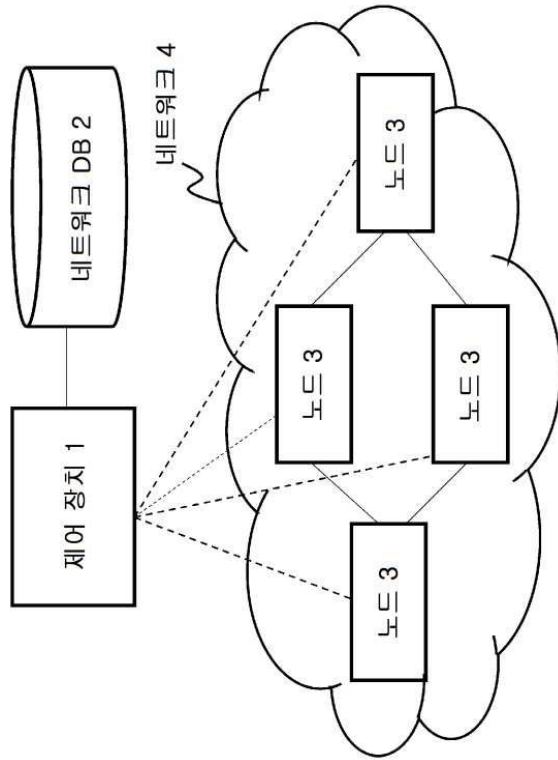
에 따라 통상의 기술자에 의해 이루어질 수 있는 다양한 변형 및 변경을 포함하는 것은 물론이다. 설명은 수치 값의 범위를 개시한다. 그러나, 설명이 임의의 수치 값 또는 그 범위에 포함되는 작은 범위를 특별히 개시하지 않을지라도, 이 값 및 범위는 구체적으로 기재되어 있는 것으로 간주되어야 한다.

부호의 설명

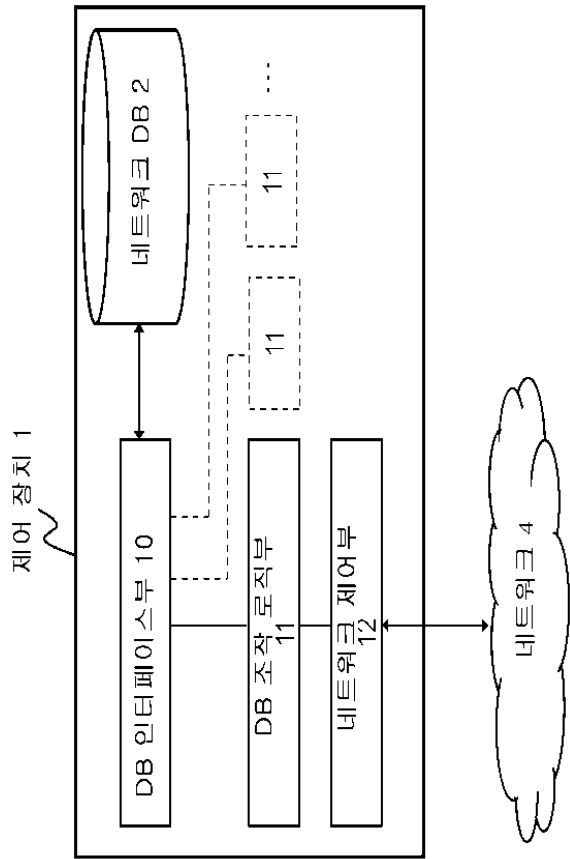
- [0229]
- 1 제어 장치
 - 2, 2A 내지 2F 네트워크 데이터베이스(네트워크 DB)
 - 3 노드
 - 4 네트워크
 - 4A 관리 네트워크
 - 9 프로세스 관리부
 - 10, 10-1 내지 10-6 DB 인터페이스부
 - 11, 11-1, 11-2 DB 조작 로직부
 - 12 네트워크 제어부
 - 11A 오픈플로우 제어부
 - 11B 가시화 DB 조작 로직부
 - 11C 슬라이스 로직부
 - 11D 최단 경로 로직부
 - 11E 애플리케이션 로직부
 - 11F 결합 로직부
 - 20 최단 경로 기억부
 - 21 호스트 정보 기억부
 - 30 컴포넌트 매니저(구성 관리부)
 - 31 컴포넌트 정보 기억부
 - 32 컴포넌트 관리부
 - 33 외부 제어 인터페이스부
 - 34 컴포넌트 제어 인터페이스부
 - 40 사용자 인터페이스
 - 100, 200, 300, 400 서버
 - 110 토폴로지 처리부
 - 111 플로우 처리부
 - 112 패킷 처리부
 - 320 개시 및 종료 기능
 - 321 접속 기능
 - 322 사활 감시 기능
 - 401 표시부
 - 402 관리부
 - TP1 내지 TP6 아이콘

도면

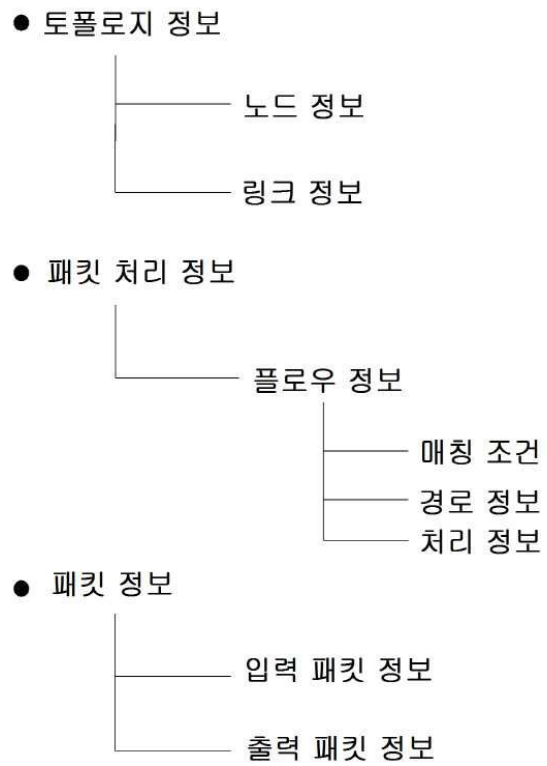
도면1



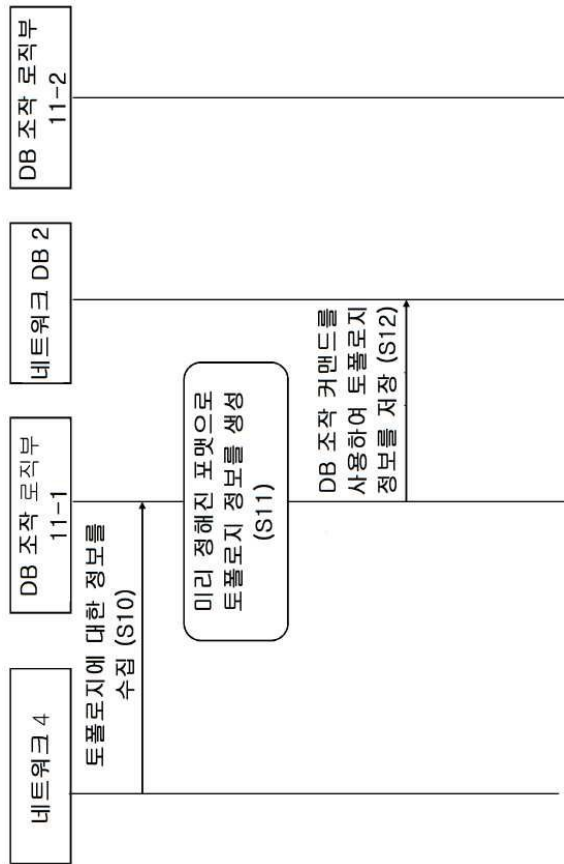
도면2



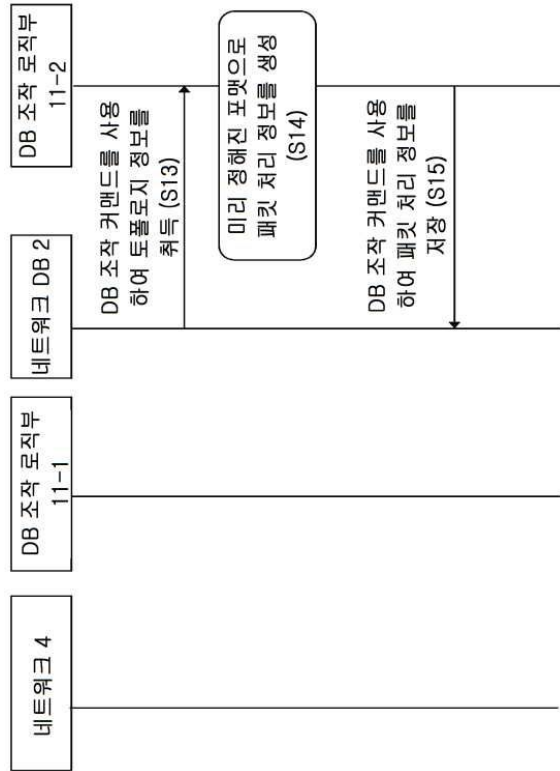
도면3



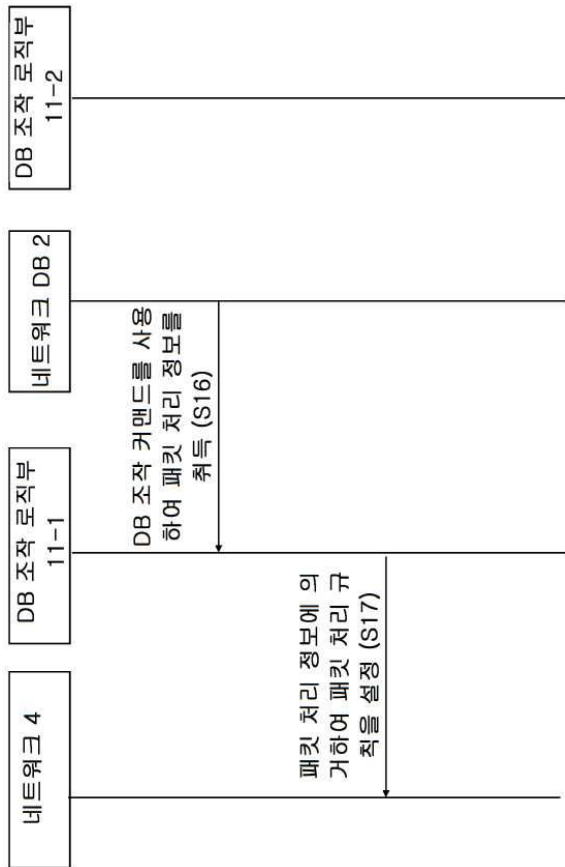
도면4



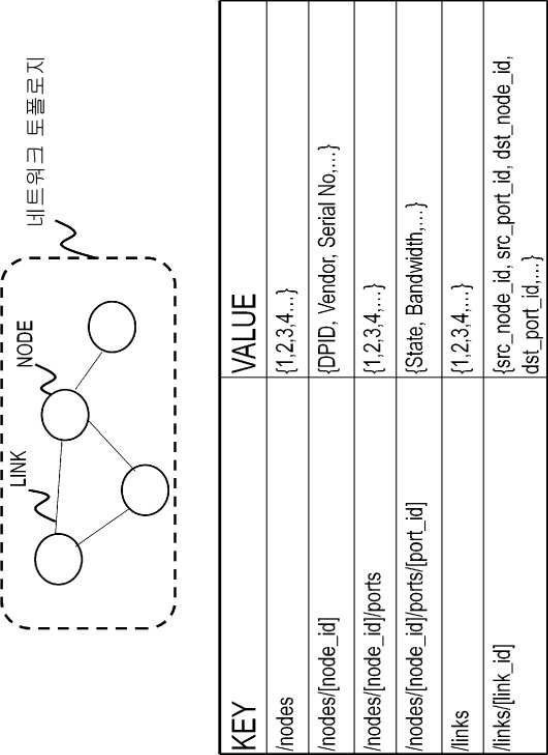
도면5



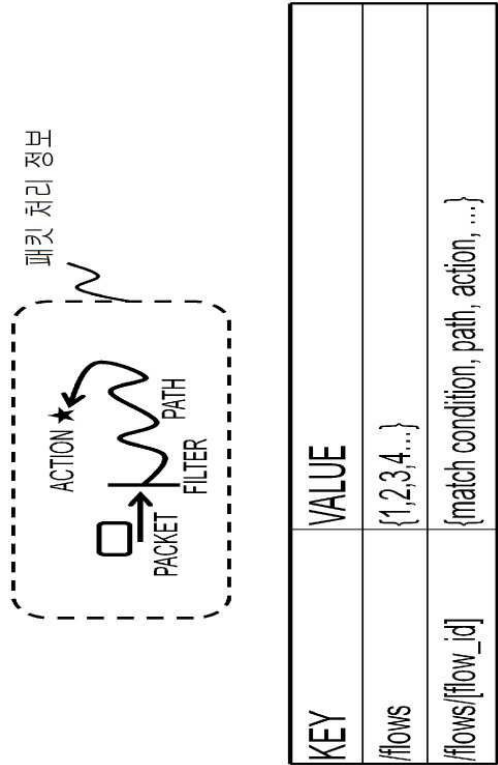
도면6



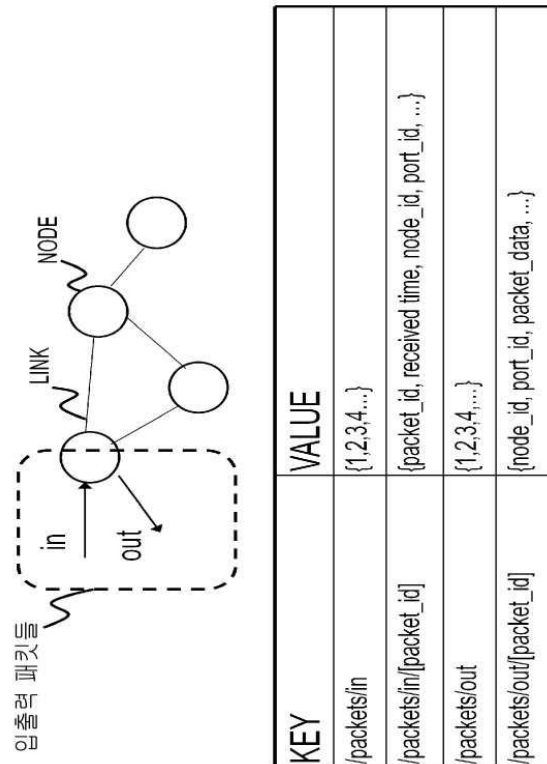
도면7



도면8



도면9



도면10

DB 조작 커맨드들의 예

정보 입출력 커맨드들:

- get(key)
 - 키와 연관된 값을 취득
- put(key, value)
 - 키와 연관된 값을 설정
- push_head(key, value)
 - 키와 연관된 리스트의 선두에 값을 부가
- push_tail(key, value)
 - 키와 연관된 리스트의 말미에 값을 부가
- push_at(key, position, value)
 - 키와 연관된 리스트의 지정된 위치에 값을 삽입
 - position은 위치뿐만 아니라 키 값에 의해 지정 가능
- pop_head(key)
 - 키와 연관된 리스트의 선두로부터 값을 취득 및 삭제
- pop_tail(key)
 - 키와 연관된 리스트의 말미로부터 값을 취득 및 삭제
- pop_at(key, position)
 - 키와 연관된 리스트의 지정된 위치로부터 값을 취득 및 삭제
 - position은 위치뿐만 아니라 키 값에 의해 지정 가능

도면11

DB 조작 커맨드들의 예

통지 예약 커맨드:

•subscribe(key_pattern)

- 지정된 패턴과 매칭되는 조작이 실행될 경우에 통지되도록 지시

•unsubscribe(key_pattern)

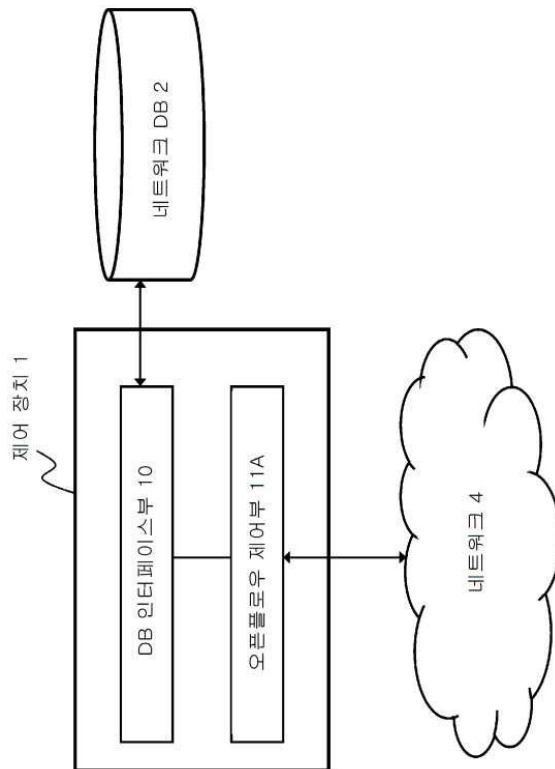
- 통지 예약 커맨드를 취소

정보 삭제 커맨드:

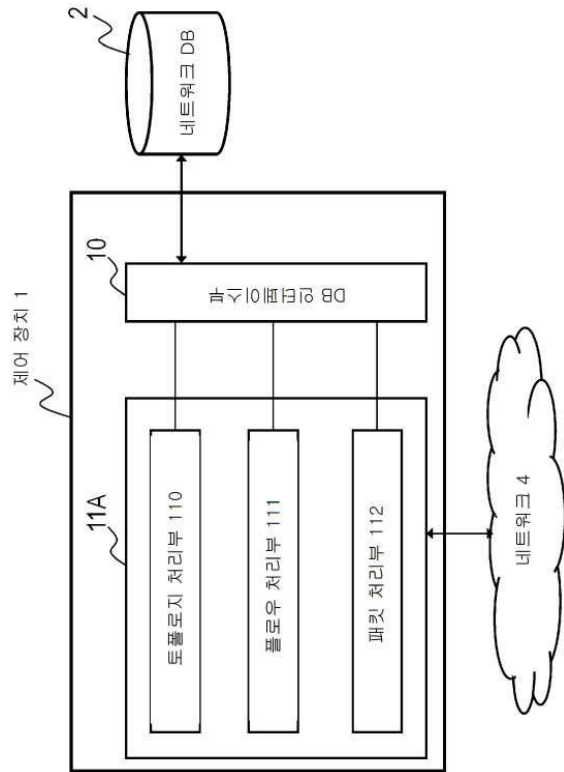
•delete(key)

- 키 및 키와 연관된 값을 삭제

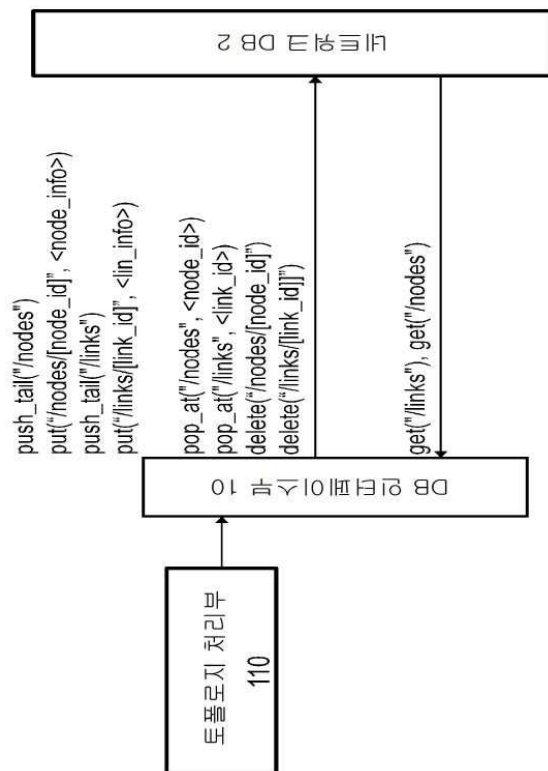
도면12



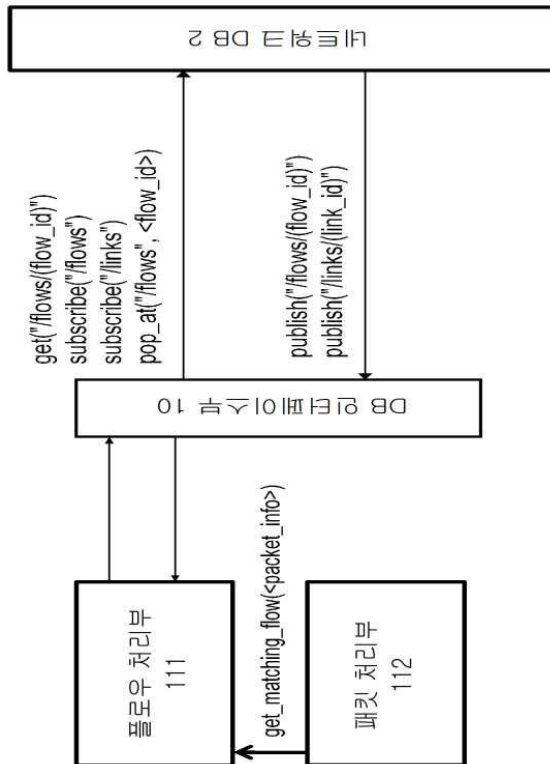
도면13



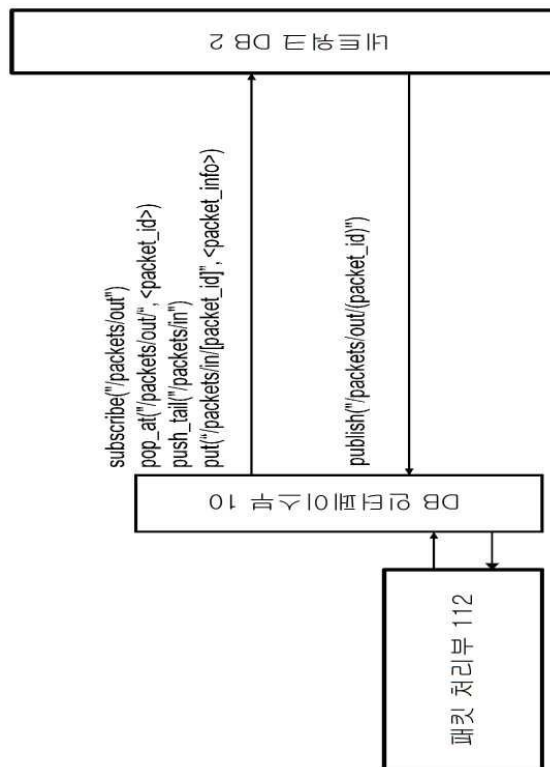
도면14



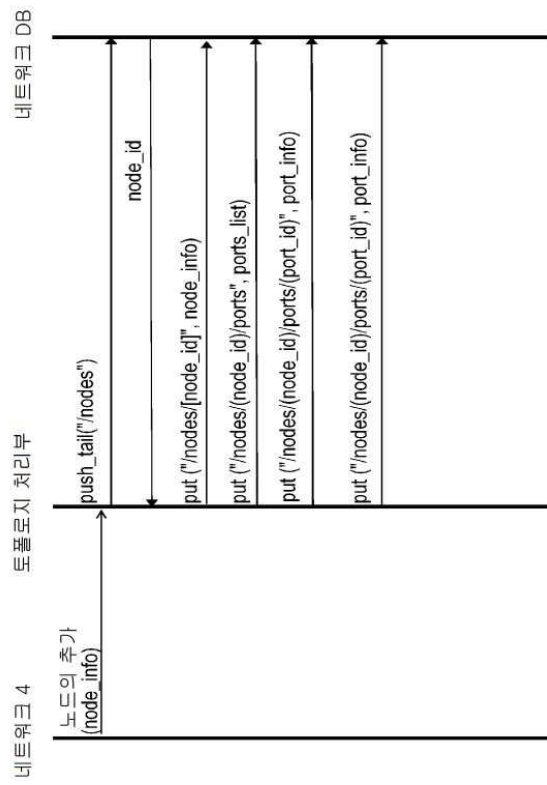
도면15



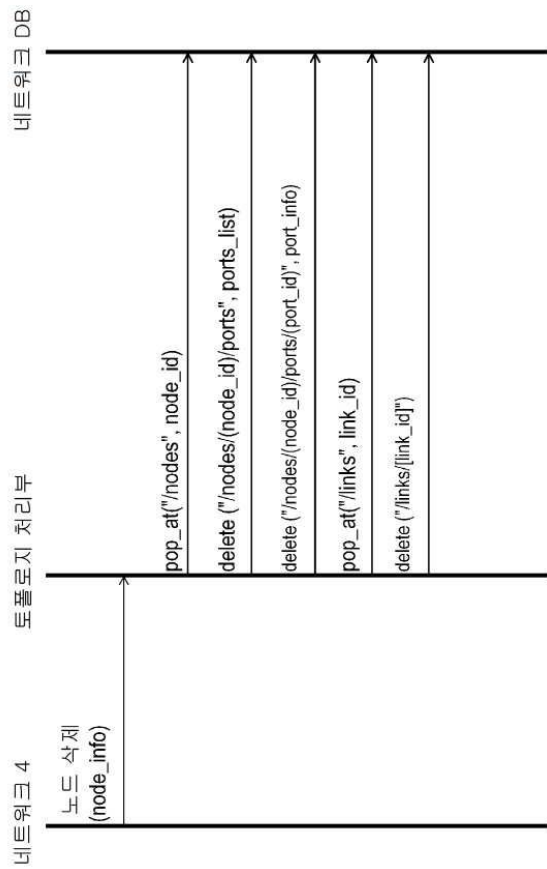
도면16



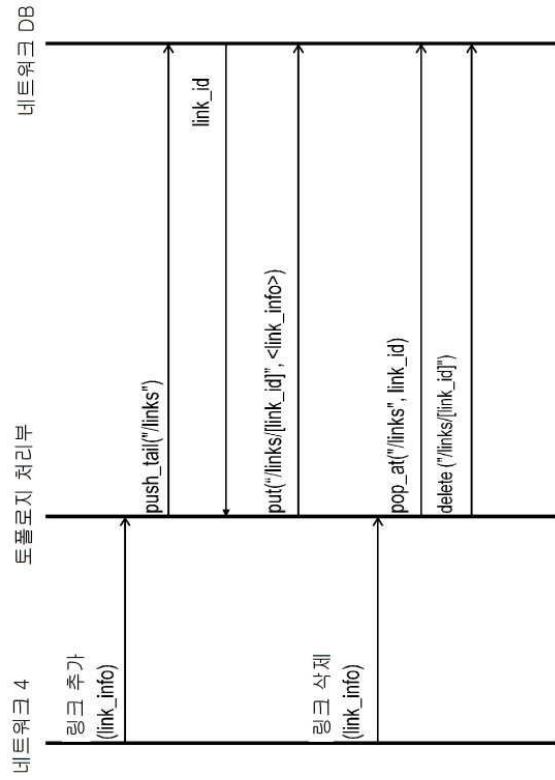
도면17



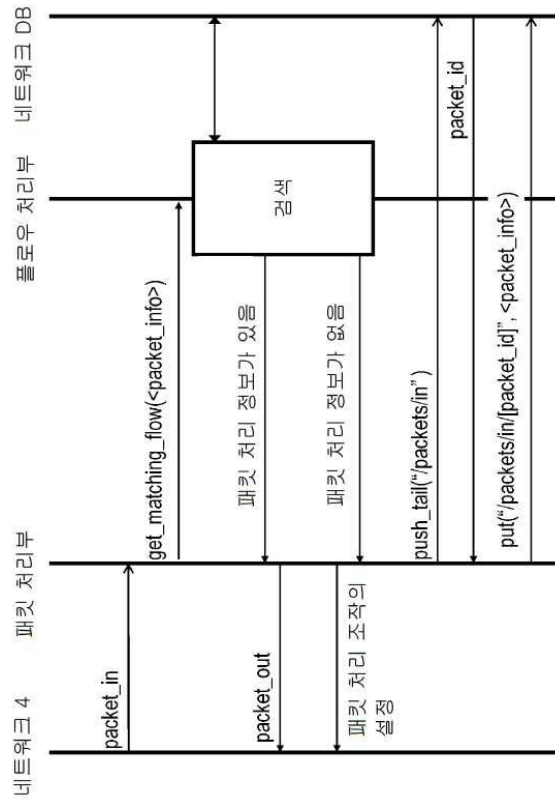
도면18



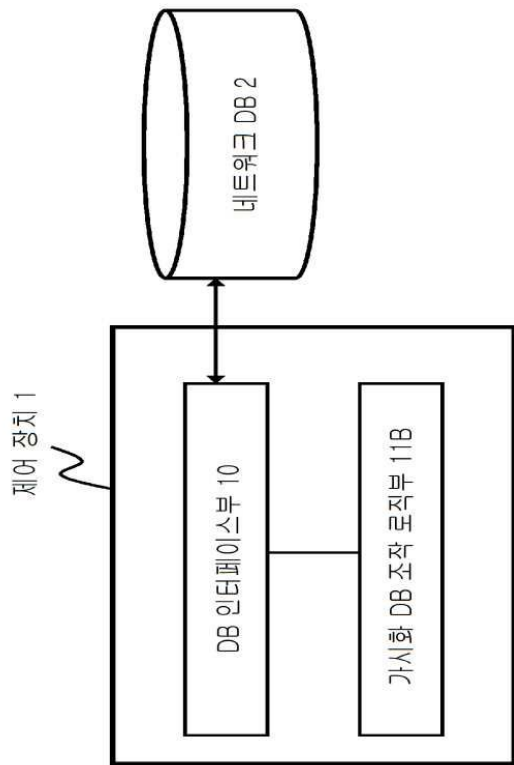
도면19



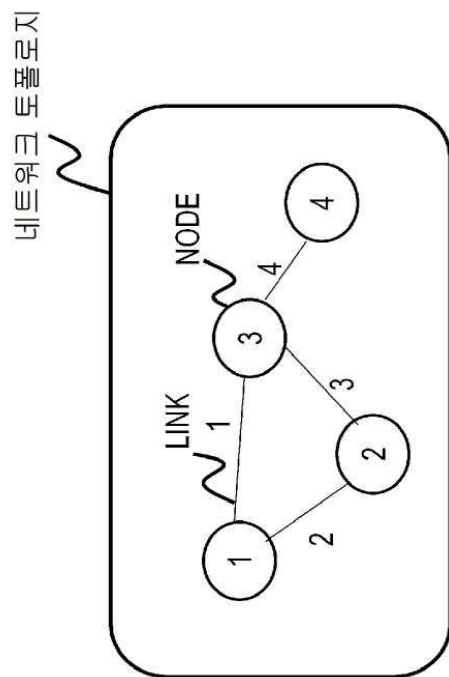
도면20



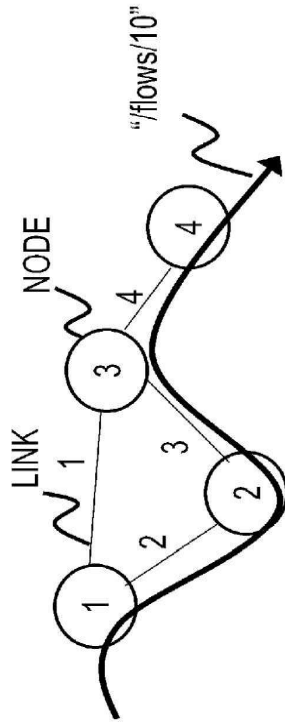
도면21



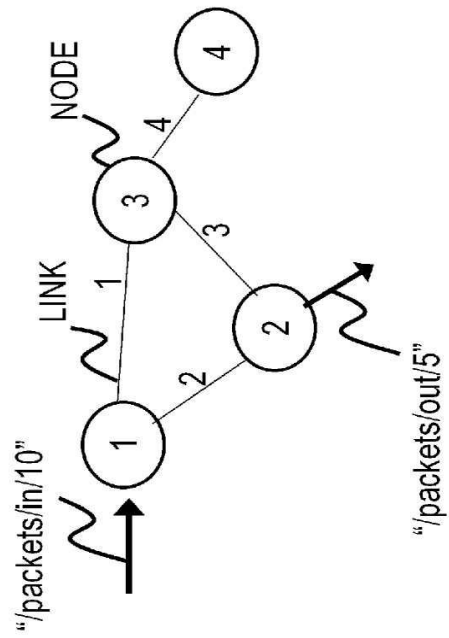
도면22



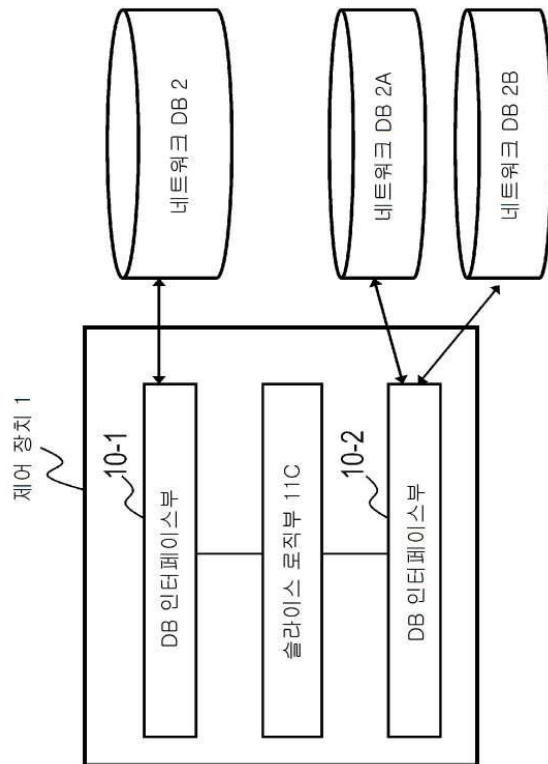
도면23



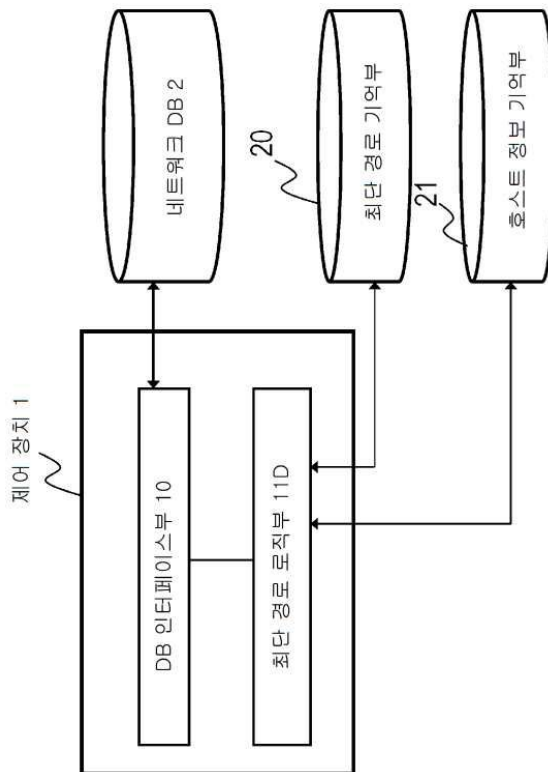
도면24



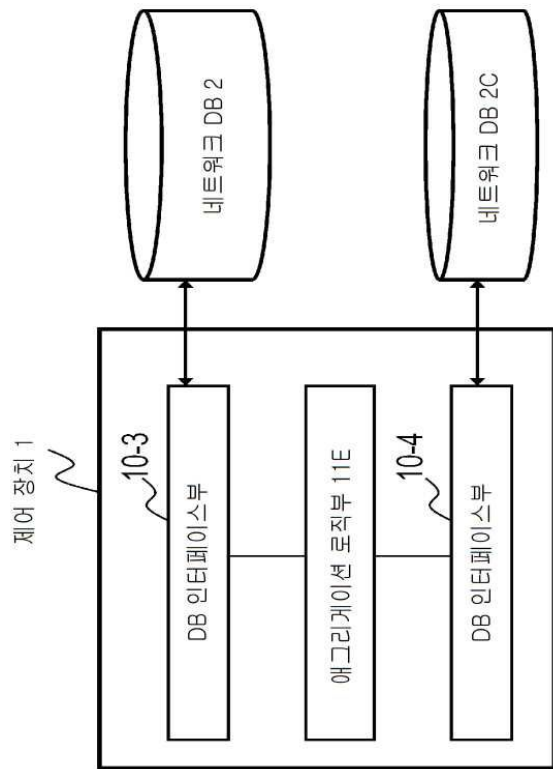
도면25



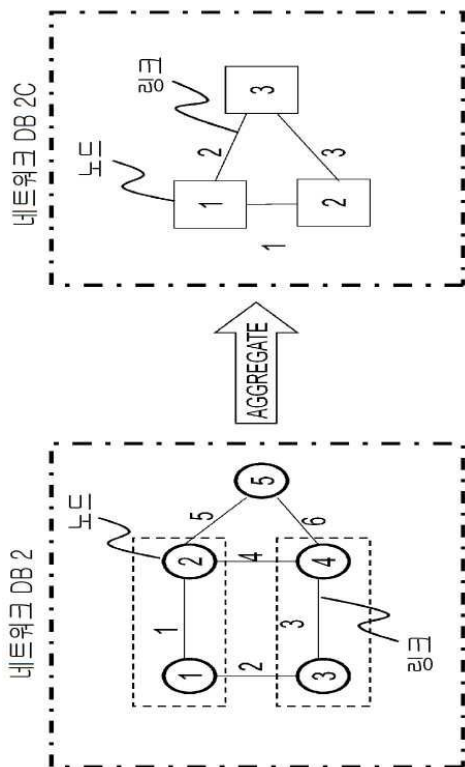
도면26



도면27



도면28



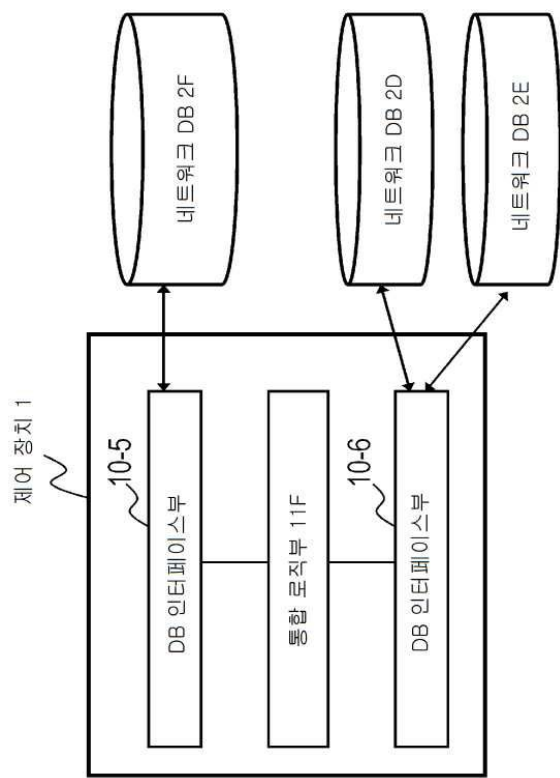
도면29

KEY	VALUE
/nodes	{1,2,3,4,5}
/nodes/1	{ 노드 ID(A)}
/nodes/2	{ 노드 ID(B)}
/nodes/3	{ 노드 ID(C)}
/nodes/4	{ 노드 ID(D)}
/nodes/5	{ 노드 ID(E)}
/links	{1,2,3,4,5,6,7}
/links/1	{src_node:A, dst_node: B, ... }
/links/2	{src_node:A, dst_node: C, ... }
/links/3	{src_node:C, dst_node: D, ... }
/links/4	{src_node:D, dst_node: B, ... }
/links/5	{src_node:B, dst_node: E, ... }
/links/6	{src_node:D, dst_node: E, ... }

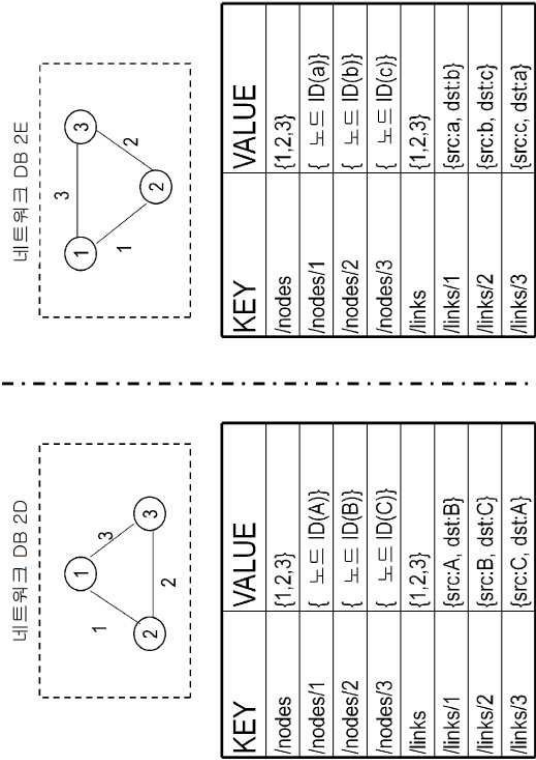
도면30

KEY	VALUE
/nodes	{1,2,3}
/nodes/1	{ 노드 ID(a), original(A, B)}
/nodes/2	{ 노드 ID(b), original(C, D)}
/nodes/3	{ 노드 ID(C), original(E)}
/links	{1,2,3}
/links/1	{src_node:A, dst_node: C, ... }, {src_node:D, dst_node: B, ... }
/links/2	{src_node:B, dst_node: E, ... }
/links/3	{src_node:D, dst_node: E, ... }

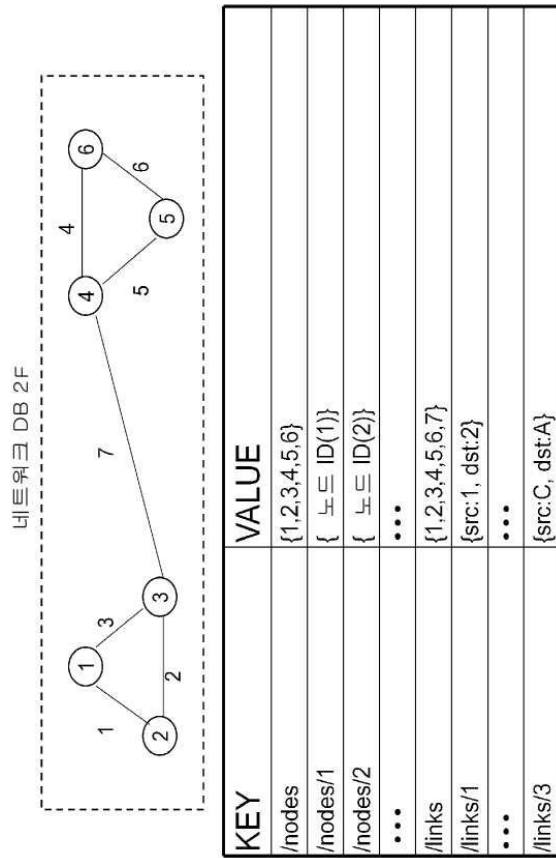
도면31



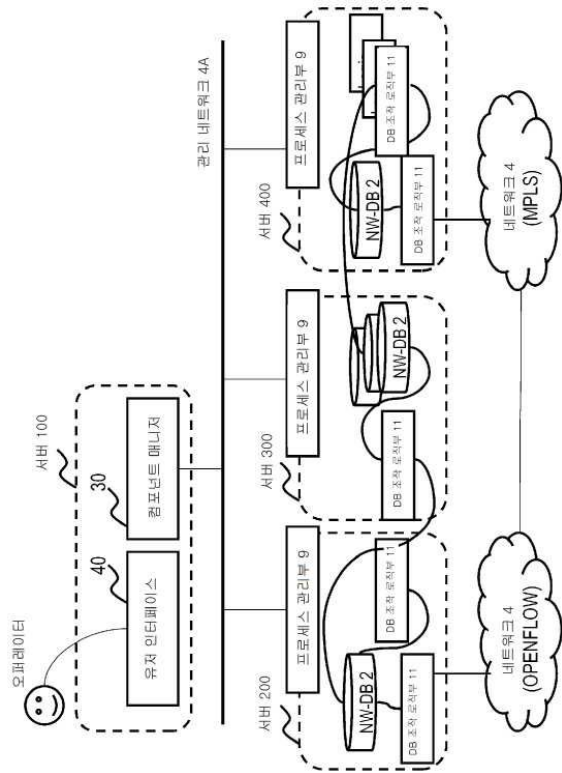
도면32



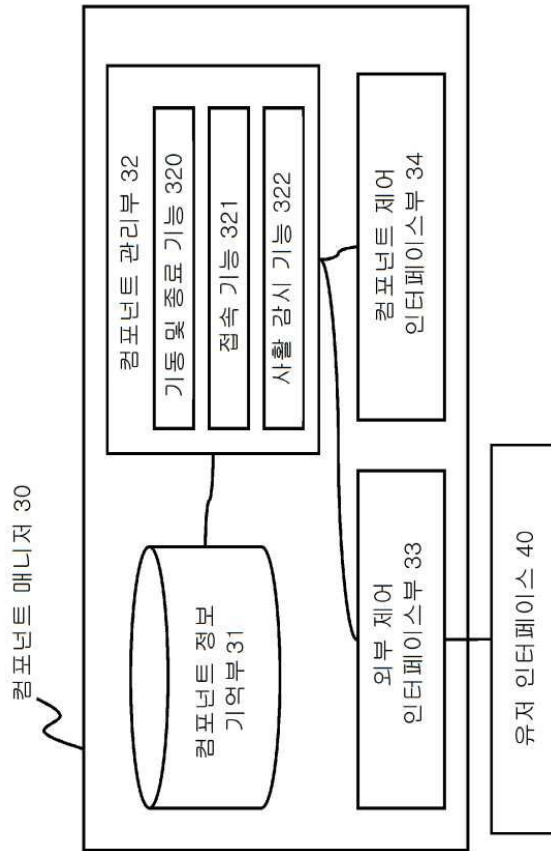
도면33



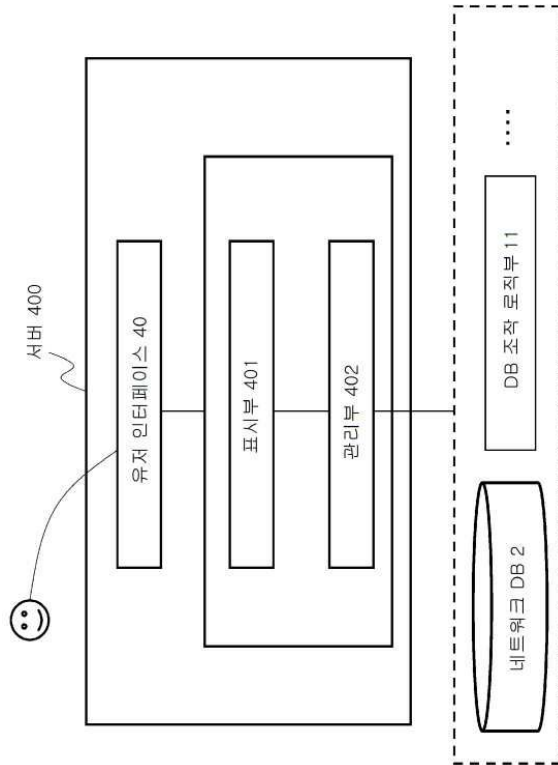
도면34



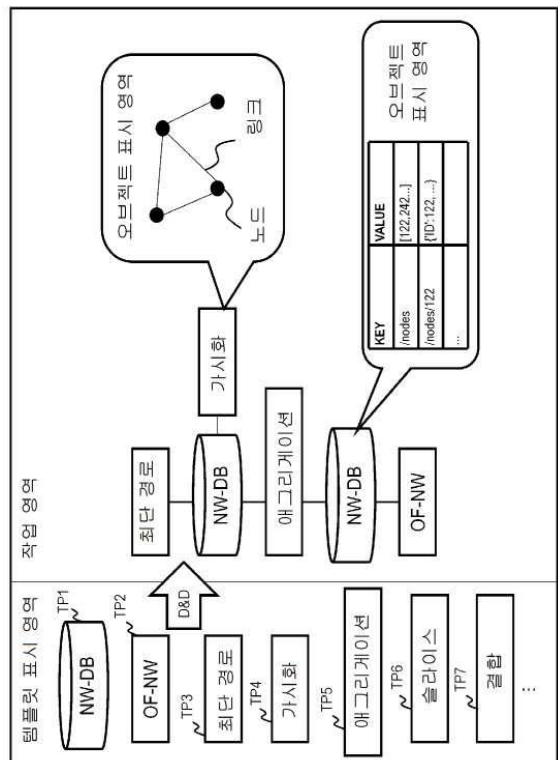
도면35



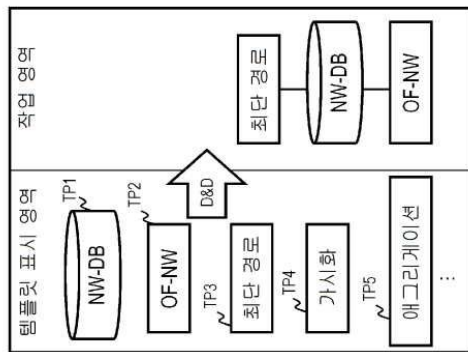
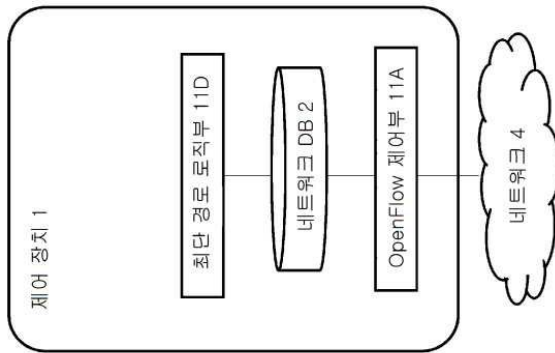
도면36



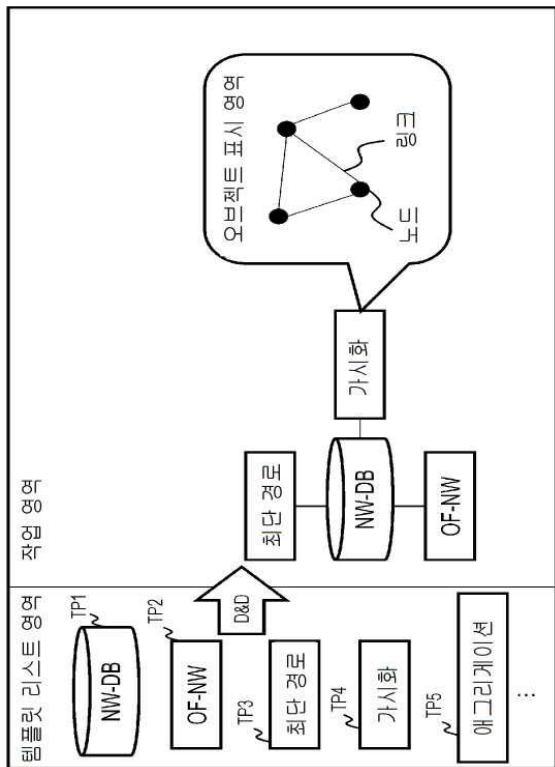
도면37



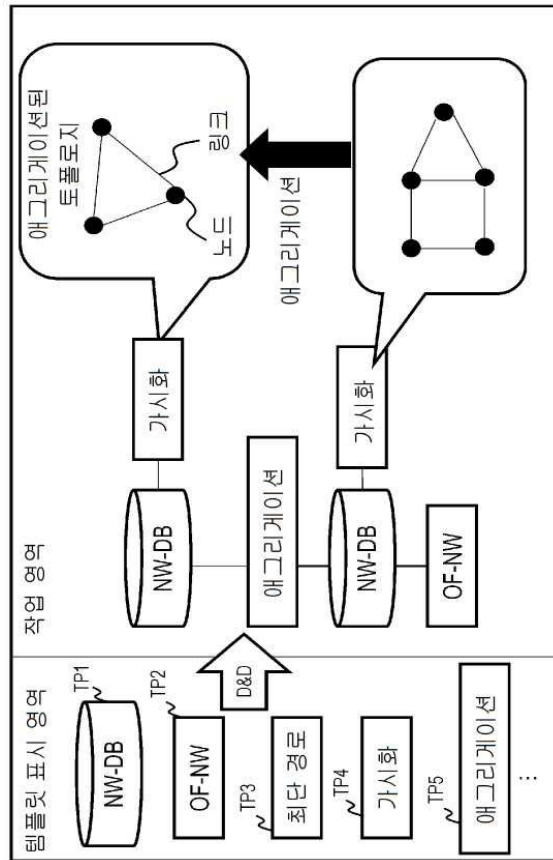
도면38



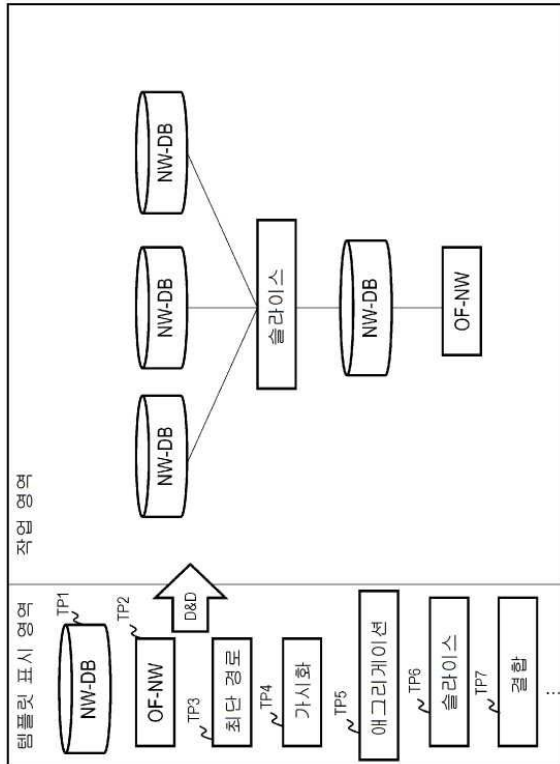
도면39



도면40



도면41



도면42

