



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년08월28일
(11) 등록번호 10-1982230
(24) 등록일자 2019년05월20일

(51) 국제특허분류(Int. Cl.)
H04L 12/931 (2013.01) G06F 9/455 (2018.01)
H04L 12/935 (2013.01)
(52) CPC특허분류
H04L 49/70 (2013.01)
G06F 9/45533 (2013.01)
(21) 출원번호 10-2018-0078872
(22) 출원일자 2018년07월06일
심사청구일자 2018년07월06일
(56) 선행기술조사문헌
KR101477012 B1
KR101658824 B1

(73) 특허권자
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
양경식
서울특별시 동대문구 무학로45길 40, 401호 (용두동)
유봉열
서울특별시 송파구 풍성로24길 24-1, 501호
유혁
서울특별시 용산구 이촌로71길 10, 213동1904호(이촌동, 한가람아파트)
(74) 대리인
김홍석

전체 청구항 수 : 총 11 항

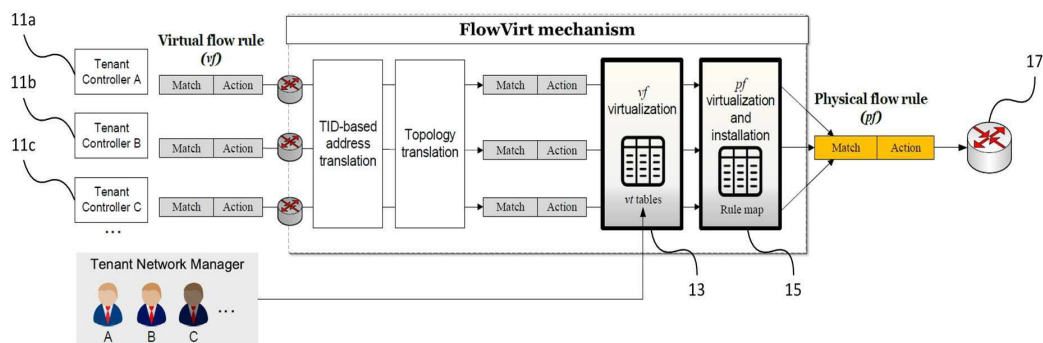
심사관 : 김진권

(54) 발명의 명칭 프로그래밍 가능한 네트워크 가상화에서의 플로우 룰 가상화 장치 및 그 방법

(57) 요약

플로우 룰 가상화 방법이 개시된다. 상기 플로우 룰 가상화 방법은 (a) 가상화 타입 테이블을 생성하는 단계, (b) 상기 가상화 타입 테이블을 이용하여 테넌트 컨트롤러로부터 생성된 가상의 플로우 룰에 대하여 상기 가상의 플로우 룰과 물리적 네트워크에서 실행되는 물리적 플로우 룰 간에 맵핑 정도를 나타내는 가상화 타입을 선택하는 단계, (c) 상기 선택된 가상화 타입에 따라 상기 가상의 플로우 룰을 상기 물리적 플로우 룰로 가상화하는 단계, (d) 룰 맵(rule map)을 이용하여 상기 물리적 플로우 룰을 가상화하는 단계, 및 (e) 상기 가상화된 물리적 플로우 룰을 상기 물리적 스위치에 설치하는 단계를 포함한다.

대표도 - 도2



(52) CPC특허분류

H04L 49/3009 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 20150002880041001

부처명 과학기술정보통신부

연구관리전문기관 정보통신기술진흥센터

연구사업명 정보통신방송연구개발사업

연구과제명 SDN 2.0 실현을 위한 네트워크 가상화 플랫폼 핵심 기술 및 서비스 연구

기 여 율 1/2

주관기관 고려대학교 산학협력단

연구기간 2018.01.01 ~ 2018.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호 20150002800041001

부처명 과학기술정보통신부

연구관리전문기관 정보통신기술진흥센터

연구사업명 정보통신방송연구개발사업

연구과제명 (SW 스타랩) 성능 및 보안 SLA 보장이 가능한 차세대 클라우드 인프라SW 개발

기 여 율 1/2

주관기관 고려대학교 산학협력단

연구기간 2018.01.01 ~ 2018.12.31

공지예외적용 : 있음

명세서

청구범위

청구항 1

- (a) 가상화 타입 테이블을 생성하는 단계;
- (b) 상기 가상화 타입 테이블을 이용하여 테넌트 컨트롤러로부터 생성된 가상의 플로우 룰에 대하여 상기 가상의 플로우 룰과 물리적 네트워크에서 실행되는 물리적 플로우 룰 간에 맵핑 정도를 나타내는 가상화 타입을 선택하는 단계;
- (c) 상기 선택된 가상화 타입에 따라 상기 가상의 플로우 룰을 상기 물리적 플로우 룰로 가상화하는 단계;
- (d) 룰 맵(rule map)을 이용하여 상기 물리적 플로우 룰을 가상화하는 단계; 및
- (e) 상기 가상화된 물리적 플로우 룰을 물리적 스위치에 설치하는 단계를 포함하는 플로우 룰 가상화 방법.

청구항 2

제1항에 있어서,

상기 가상화 타입은 병합의 단위에 따라 홉 기반의 홉 타입(hop vt), 최종 호스트 기반의 종단 타입(end vt), 애플리케이션 기반의 애플리케이션 타입(application vt) 및 테넌트 내의 병합이 없을 때의 임의 타입(arbitrary vt)을 포함하고,

상기 홉 타입은 테넌트 내 병합일 때의 홉 격리 타입(hop-isolated vt) 및 테넌트 간 병합일 때의 홉 병합 타입(hop-merged vt)을 포함하는 플로우 룰 가상화 방법.

청구항 3

제1항에 있어서,

상기 가상화 타입 테이블은 테넌트로부터 얻은 적어도 하나 이상의 가상화 타입 정책으로 구성되고,

상기 가상화 타입 테이블의 각 항목은 스위치 ID, 플로우 정보, 가상화 타입 및 우선 순위를 포함하는 플로우 룰 가상화 방법.

청구항 4

제3항에 있어서,

상기 우선 순위는 상기 적어도 하나 이상의 가상화 타입 정책이 상기 가상의 플로우 룰에 적용되는 순위를 의미하는 플로우 룰 가상화 방법.

청구항 5

제4항에 있어서,

상기 (b) 단계는 가상 스위치 ID와 상기 가상화 타입 테이블의 스위치 ID를 비교하고, 상기 가상의 플로우 룰의 매칭 필드와 상기 플로우 정보를 비교하여 상기 적어도 하나 이상의 가상화 타입 정책 중에서 일치하는 가상화 타입 정책을 조회하고, 일치하는 가상화 타입 정책들 중에서 가장 높은 우선 순위를 갖는 가상화 타입 정책의 가상화 타입을 선택하는 플로우 룰 가상화 방법.

청구항 6

제2항에 있어서,

상기 (c) 단계는

상기 선택된 가상화 타입이 애플리케이션 타입이고, 상기 물리적 스위치에 상기 애플리케이션 타입보다 거친 (coarser-grained) 가상화 타입의 물리적 플로우 룰이 설치되어 있는 경우, 상기 물리적 스위치에 표시 규칙 (indicator rule)을 설치하는 단계를 더 포함하고,

상기 표시 규칙은 상기 거친 가상화 타입의 물리적 플로우 룰보다 높은 우선 순위를 갖고, 상기 물리적 스위치의 첫 번째 애플리케이션 패킷을 네트워크 하이퍼바이저에 전달하도록하는 플로우 룰 가상화 방법.

청구항 7

제2항에 있어서,

상기 (d) 단계는

상기 물리적 플로우 룰이 상기 물리적 스위치에 기 설치된 물리적 플로우 룰과 충돌하는지 여부를 검사하는 단계를 더 포함하고,

상기 물리적 플로우 룰과 상기 물리적 스위치에 기 설치된 물리적 플로우 룰이 충돌하는 것으로 판단되면 상기 룰 맵을 이용하여 상기 물리적 플로우 룰을 중단 타입으로 확장시켜 가상화하는 플로우 룰 가상화 방법.

청구항 8

가상화 타입 테이블을 생성하는 테이블 생성부;

상기 가상화 타입 테이블을 이용하여 테넌트 컨트롤러로부터 생성된 가상의 플로우 룰에 대하여 상기 가상의 플로우 룰과 물리적 네트워크에서 실행되는 물리적 플로우 룰 간에 맵핑 정도를 나타내는 가상화 타입을 선택하고, 상기 선택된 가상화 타입에 따라 상기 가상의 플로우 룰을 상기 물리적 플로우 룰로 가상화하는 vf 가상화부; 및

룰 맵(rule map)을 이용하여 상기 물리적 플로우 룰을 가상화하는 pf 가상화부를 포함하는 플로우 룰 가상화 장치.

청구항 9

제8항에 있어서,

상기 가상화 타입은 병합의 단위에 따라 홉 기반의 홉 타입(hop vt), 최종 호스트 기반의 중단 타입(end vt), 애플리케이션 기반의 애플리케이션 타입(application vt) 및 테넌트 내의 병합이 없을 때의 임의 타입(arbitrary vt)을 포함하고,

상기 홉 타입은 테넌트 내 병합일 때의 홉 격리 타입(hop-isolated vt) 및 테넌트 간 병합일 때의 홉 병합 타입(hop-merged vt)을 포함하는 플로우 룰 가상화 장치.

청구항 10

제9항에 있어서,

상기 선택된 가상화 타입이 애플리케이션 타입이고, 물리적 스위치에 상기 애플리케이션 타입보다 거친 (coarser-grained) 가상화 타입의 물리적 플로우 룰이 설치되어 있는 경우, 상기 물리적 스위치에 표시 규칙

(indicator rule)을 설치하는 표시 규칙 설치부를 더 포함하고,

상기 표시 규칙은 상기 거친 가상화 타입의 물리적 플로우 룰보다 높은 우선 순위를 갖고, 상기 물리적 스위치의 첫 번째 애플리케이션 패킷을 네트워크 하이퍼바이저에 전달하도록하는 플로우 룰 가상화 장치.

청구항 11

제9항에 있어서,

상기 pf 가상화부는 상기 물리적 플로우 룰이 물리적 스위치에 기 설치된 물리적 플로우 룰과 충돌하는지 여부를 검사하고, 상기 물리적 플로우 룰과 물리적 스위치에 기 설치된 물리적 플로우 룰이 충돌하면 상기 룰 맵을 이용하여 상기 물리적 플로우 룰을 중단 타입으로 확장시켜 가상화하는 플로우 룰 가상화 장치.

발명의 설명

기술 분야

[0001] 본 발명은 플로우 룰 가상화 장치 및 그 방법에 관한 것으로서, 보다 구체적으로 프로그래밍 가능한 네트워크 가상화(P-NV)의 동적 확장성을 위한 플로우 룰 가상화 장치 및 그 방법에 관한 것이다.

배경 기술

[0002] 최근 몇 년 동안 클라우드 컴퓨팅 기술은 다양한 분야에서 널리 채택되어왔다. 클라우드 컴퓨팅은 주어진 자원에 대하여 고급 시스템 관리가 필요한 서비스를 가능하게 하는 에지(edge)나 포그(fog) 컴퓨팅의 형태로서 사용된다. 클라우드 센터의 네트워크 자원의 다양한 사용 사례는 네트워크 가상화와 함께 실현 가능하며, 유연한 시스템의 배포로 인해 클라우드 인프라 내에서 자신들만의 가상 네트워크 관리 기능을 요구하는 테넌트들이 증가하고 있다.

[0003] 그러나, 대부분의 클라우드 솔루션은 호스트 노드에만 IaaS(Infrastructure as a Service)를 제공하며, 이러한 솔루션들은 가상 머신 모니터들 간에 오버레이 네트워킹을 사용하는 호스트 기반 네트워크 가상화(H-NV, Host-based Network Virtualization)에 의존한다. H-NV는 임의의 가상 네트워크를 구성하는데, 이는 가상 스위치와 물리적 토폴로지에 맵핑된 링크로 구성되고, 각 테넌트들을 위해 가상 네트워크 관리 및 최적화가 제한된다.

[0004] 반면에, 네트워크 하이퍼바이저(network hypervisor)가 전체 네트워크 스위치를 중앙 집중식으로 가상화하는 프로그래밍 가능한 네트워크 가상화(P-NV, Programmable Network Virtualization)는 완전히 프로그래밍 가능한 가상 네트워크를 제공하고, 각 테넌트를 위해 유연한 네트워크 관리를 가능하게 한다. 그러나, P-NV는 지금까지 널리 채택되지 않았다.

[0005] 클라우드 네트워킹에서의 핵심 요구 사항은 확장성(scalability)이다. H-NV와 P-NV의 확장성을 비교한 그래프를 도시한 도 1을 참조하면, 최종 호스트가 동일한 네트워킹 작업을 수행할 때의 H-NV와 P-NV가 각각 소비하는 자원 사용량을 살펴보면, P-NV의 컨트롤 채널 사용 및 CPU 주기가 H-NV와 비교할 때 각각 400[%] 및 170[%]만큼 높다. 이러한 P-NV의 확장성 문제는 기존의 네트워크 하이퍼바이저가 단지 모든 플로우 룰(flow rule)을 일대일 방식으로 물리적 네트워크에 설치하기 때문에 발생한다.

선행기술문헌

특허문헌

[0006] (특허문헌 0001) 대한민국 등록특허 제1794719호

(특허문헌 0002) 대한민국 등록특허 제1658824호

발명의 내용

해결하려는 과제

- [0007] 플로우 룰의 수가 증가하면 하이퍼바이저의 작업과 그에 따른 자원 소비가 증가한다. 즉, P-NV의 확장성을 향상시키기 위해 플로우 룰의 수를 줄여야 한다. 플로우 룰의 저장에 사용되는 메모리를 줄이는 방법으로 압축(compression)이 있다. 그러나, 압축 기술은 지연을 발생시킬 우려가 있으며, 모든 호스트 간의 플로우 룰을 먼저 설정하는 것이 필요하므로 자원 소모를 오히려 증가시킬 수도 있다. 또한, 압축 기술은 테넌트의 네트워크 관리 정책을 고려하지 않은채 무분별하게 플로우 룰을 압축함으로써 테넌트 시맨틱(tenant semantic)을 위반할 위험성이 있다.
- [0008] 이에 따라, 본 발명의 목적은 반응적 포워딩 스키마(reactive forwarding scheme)를 통해 플로우 룰을 가상화하여 P-NV 확장성을 향상시킬 수 있는 플로우 룰 가상화 장치 및 그 방법을 제공하는데 있다.

과제의 해결 수단

- [0009] 본 발명의 일 실시 예에 따른 플로우 룰 가상화 방법은 (a) 가상화 타입 테이블을 생성하는 단계, (b) 상기 가상화 타입 테이블을 이용하여 테넌트 컨트롤러로부터 생성된 가상의 플로우 룰에 대하여 상기 가상의 플로우 룰과 물리적 네트워크에서 실행되는 물리적 플로우 룰 간에 맵핑 정도를 나타내는 가상화 타입을 선택하는 단계, (c) 상기 선택된 가상화 타입에 따라 상기 가상의 플로우 룰을 상기 물리적 플로우 룰로 가상화하는 단계, (d) 룰 맵(rule map)을 이용하여 상기 물리적 플로우 룰을 가상화하는 단계, 및 (e) 상기 가상화된 물리적 플로우 룰을 상기 물리적 스위치에 설치하는 단계를 포함할 수 있다.
- [0010] 본 발명의 일 실시 예에 따른 플로우 룰 가상화 장치는 가상화 타입 테이블을 생성하는 테이블 생성부, 상기 가상화 타입 테이블을 이용하여 테넌트 컨트롤러로부터 생성된 가상의 플로우 룰에 대하여 상기 가상의 플로우 룰과 물리적 네트워크에서 실행되는 물리적 플로우 룰 간에 맵핑 정도를 나타내는 가상화 타입을 선택하고, 상기 선택된 가상화 타입에 따라 상기 가상의 플로우 룰을 상기 물리적 플로우 룰로 가상화하는 vf 가상화부, 및 룰 맵(rule map)을 이용하여 상기 물리적 플로우 룰을 가상화하는 pf 가상화부를 포함할 수 있다.

발명의 효과

- [0011] 본 발명의 일 실시 예에 따르면, 플로우 룰을 가상화하여 플로우 룰의 수를 감소시키고, 컨트롤 채널의 대역폭과 CPU 사용량을 감소시킴으로써 P-NV의 확장성을 향상시킬 수 있다.
- [0012] 또한, 테넌트들을 위한 다양한 플로우 룰 운영 정책들을 만족시킬 수 있다.

도면의 간단한 설명

- [0013] 도 1은 H-NV와 P-NV의 확장성을 비교한 그래프를 도시한 것이다.
- 도 2는 본 발명의 일 실시 예에 따른 플로우 룰 가상화 시스템을 개략적으로 도시한 것이다.
- 도 3은 본 발명의 일 실시 예에 따른 플로우 룰 가상화 장치의 블록도이다.
- 도 4는 가상화 타입 테이블의 일 예를 도시한 것이다.
- 도 5는 가상의 플로우 룰 병합의 일 예를 도시한 것이다.
- 도 6은 스위치의 플로우 테이블에 대하여 설치된 표시 규칙을 도시한 것이다.
- 도 7은 pf 가상화의 일 예를 도시한 것이다.
- 도 8은 본 발명의 일 실시 예에 따른 플로우 룰 가상화 방법의 순서도이다.
- 도 9a는 토폴로지의 형태의 일 예를 도시한 것이다.
- 도 9b는 도 9a에 도시된 토폴로지의 형태에 따른 플로우 룰의 수의 비율을 비교한 것이다.
- 도 10은 네트워크 하이퍼바이저에서의 CPU의 사용량을 도시한 것이다.

발명을 실시하기 위한 구체적인 내용

- [0014] 본 명세서에 개시되어 있는 본 발명의 개념에 따른 실시예들에 대해서 특정한 구조적 또는 기능적 설명들은 단지 본 발명의 개념에 따른 실시예들을 설명하기 위한 목적으로 예시된 것으로서, 본 발명의 개념에 따른 실시예

들은 다양한 형태로 실시될 수 있으며 본 명세서에 설명된 실시예들에 한정되지 않는다.

- [0015] 본 발명의 개념에 따른 실시예들은 다양한 변경들을 가할 수 있고 여러 가지 형태들을 가질 수 있으므로 실시예들을 도면에 예시하고 본 명세서에 상세하게 설명하고자 한다. 그러나 이는 본 발명의 개념에 따른 실시예들을 특정한 개시형태들에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 변경, 균등물, 또는 대체물을 포함한다.
- [0016] 제1 또는 제2 등의 용어를 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만, 예를 들어 본 발명의 개념에 따른 권리 범위로부터 이탈되지 않은 채, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소는 제1 구성요소로도 명명될 수 있다.
- [0017] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다. 구성요소들 간의 관계를 설명하는 표현들, 예를 들어 "~사이에"와 "바로~사이에" 또는 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.
- [0018] 본 명세서에서 사용한 용어는 단지 특정한 실시예들을 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수개의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 실시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함으로 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0019] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다. 이하, 실시예들을 첨부된 도면을 참조하여 상세하게 설명한다.
- [0021] 기존의 P-NV에서, 새로운 패킷 연결에 대한 진입 스위치는 새로운 연결이 있으면 플로우 룰 요청 메시지를 네트워크 하이퍼바이저에 전송한다. 네트워크 하이퍼바이저가 플로우 룰 요청 메시지를 수신하면, 테넌트 컨트롤러(tenant controller, TC)에 메시지를 전달한다. 테넌트 컨트롤러는 메시지를 수신하면 연결에 대한 패킷 전달 경로를 계산하고, 각 가상 스위치에 대하여 구성된 플로우 룰을 네트워크 하이퍼바이저에 전송한다. 한편, 기존의 P-NV는 주소 및 토폴로지 변환만 고려하여 테넌트 컨트롤러로부터 생성된 플로우 룰을 전송한다.
- [0022] 이에 대하여, 본 발명은 P-NV가 갖는 확장성 문제를 해결하기 위해 플로우 룰을 가상화하여 플로우 룰의 수를 줄이고, P-NV 하이퍼바이저의 컨트롤 메시지나 CPU 사이클과 같은 컴퓨팅 자원의 후속 프로비저닝을 감소시키고자 한다.
- [0023] 이를 위하여, 가상의 플로우 룰과 물리적 플로우 룰의 맵핑을 관리하는 구조체인 룰 맵(rule map)을 이용한다. 특히, 세밀한(fine-grained) 가상화를 제공하기 위해 플로우 룰을 가상(virtual) 및 물리적(physical)으로 분리한다. 다음으로, 가상의 플로우 룰에서 물리적 플로우 룰로의 맵핑을 유지한다. 이러한 맵핑을 사용하여, 본 발명은 가상의 플로우 룰을 병합(merge)하여 더 적은 수의 물리적 플로우 룰로 만든다.
- [0025] 도 2는 본 발명의 일 실시 예에 따른 플로우 룰 가상화 시스템을 개략적으로 도시한 것이다.
- [0026] 도 2를 참조하면, FlowVirt라고도 명명될 수 있는 본 발명에 따른 플로우 룰 가상화 장치 및 방법이 포함되는 플로우 룰 가상화 시스템(1)은 테넌트 컨트롤러(11a 내지 11c)로부터 생성된 플로우 룰을 가상의 플로우 룰(virtual flow rule, vf)로 정의하고, 가상의 플로우 룰이 물리적 네트워크에서 실행되는 플로우 룰을 물리적 플로우 룰(physical flow rule, pf)로 정의한다.
- [0027] 여기서, 가상의 플로우 룰과 물리적 플로우 룰은 튜플 (M, A)로 정의될 수 있다. 이때, M은 패킷의 특정 필드에 대한 매칭 정보를 포함하고, A는 매칭된 패킷에 대하여 수행하는 동작 정보를 포함할 수 있다.
- [0028] 일 실시 예에 따라, 가상의 플로우 룰 또는 물리적 플로우 룰의 매칭 정보는 vf.M 또는 pf.M으로 나타낼 수 있

고, 동작 정보는 vf.A 또는 pf.A로 나타낼 수 있다.

- [0029] 다음으로, 플로우 룰 가상화 장치 및 방법은 가상의 플로우 룰에 대하여 가상화 타입(virtual type, vt)을 선택하고, 가상화 타입 테이블(virtual type table)(13)을 이용하여 가상의 플로우 룰을 물리적 플로우 룰로 가상화한다. 여기서, 가상의 플로우 룰을 물리적 플로우 룰로 가상화하는 것은 적어도 하나 이상의 가상의 플로우 룰을 가상화 타입에 따라 서로 병합하는 것을 의미할 수 있다.
- [0030] 한편, 물리적 플로우 룰이 스위치에 설치되어 있어 설치하고자 하는 물리적 플로우 룰과 충돌위험이 있는 경우, 가상화된 물리적 플로우 룰을 룰 맵(15)을 이용하여 재가상화(re-virtualization)할 수 있다.
- [0031] 마지막으로, 플로우 룰 가상화 장치 및 방법은 물리적 스위치(17)를 포함하는 물리적 네트워크에 가상화된 물리적 플로우 룰을 설치한다.
- [0032] 이하에서는 도 3 내지 도 8을 참조하여 본 발명의 일 실시 예에 따른 플로우 룰 가상화 장치 및 방법에 대하여 상세히 설명하기로 한다.
- [0034] 도 3은 본 발명의 일 실시 예에 따른 플로우 룰 가상화 장치의 블록도이다.
- [0035] 도 3을 참조하면, 플로우 룰 가상화 장치(10)는 테이블 생성부(100), vf 가상화부(200), 표시 규칙 설치부(300) 및 pf 가상화부(400)를 포함한다.
- [0036] 테이블 생성부(100)는 가상화 타입 테이블을 생성한다. 가상화 타입 테이블은 각 테넌트마다 하나씩 할당될 수 있다.
- [0037] 가상화 타입 테이블의 일 예를 도시한 도 4를 참조하면, 가상화 타입 테이블의 각 엔트리(entry)는 각 테넌트가 가상 네트워크를 관리하기 위한 정책인 가상화 타입 정책(vt policy)으로 구성될 수 있다.
- [0038] 가상화 타입 테이블의 각 항목은 가상화 타입 정책의 구문(syntax)으로서, 예를 들면 스위치 ID(21a), 플로우 정보(flow information)(21b), 가상화 타입(21c) 및 우선 순위(21d) 등을 포함할 수 있다.
- [0039] 스위치 ID(21a)는 물리적 스위치의 식별자 정보를 의미할 수 있다.
- [0040] 플로우 정보(21b)는 패킷 전송과 관련된 정보로서 예를 들면 IP 정보나 포트 정보 등을 포함할 수 있다.
- [0041] 가상화 타입(21c)은 가상의 플로우 룰과 물리적 플로우 룰 간에 맵핑 정도(즉, 병합의 정도)를 나타내는 것으로서, 아래의 표 1과 같이 구분될 수 있다.

표 1

	Inter-merging Off	Inter-merging On
Hop	Hop-isolated	Hop-merged
End	End	Not available
Application	Application	Not available
Arbitrary	arbitrary	Not available

- [0042] 상기 표 1을 참조하면, 가상화 타입(21c)은 테넌트 간 병합 여부 및 병합의 단위에 따라 5가지 타입(hop-isolated, hop-merged, end, application 및 arbitrary)이 있다.
- [0044] 가상의 플로우 룰과 물리적 플로우 룰 간의 병합은 크게 테넌트 내의 병합(intra-merging)과 테넌트 간의 병합(inter-merging)이 있다.
- [0045] 테넌트 내의 병합은 테넌트의 격리성(isolation)을 보장한다. 테넌트 내의 병합에서의 가상화 타입은 테넌트 내의 병합의 단위에 따라 홉 기반의 홉 타입(hop vt), 최종 호스트 기반의 중단 타입(end vt), 애플리케이션 기반의 애플리케이션 타입(application vt) 및 테넌트 내의 병합이 없을 때의 임의 타입(arbitrary vt)을 포함할 수 있다.
- [0046] 여기서, 홉 타입은 가상의 플로우 룰의 홉 주소에만 일치하는 물리적 플로우 룰을 생성하도록 하며, 중단 타입은 가상의 플로우 룰의 홉 주소 및 호스트 주소와 일치하는 물리적 플로우 룰을 생성하도록 하며, 애플리케이션 타입은 홉 주소, 호스트 주소 및 애플리케이션 주소가 일치하는 물리적 플로우 룰을 생성하도록 하며, 임의 타입은 플로우 룰의 매칭 정보를 조정하지 않는다. 즉, 임의 타입은 기존의 P-NV와 동일하다.

- [0047] 테넌트 간의 병합은 테넌트의 격리성을 보장하지 않는다. 일 실시 예에 따라 패킷에 TID(tenant identifier)가 할당되는 경우, 물리적 홉 주소는 TID의 영향을 받지 않으므로 홉 타입은 병합과 함께 사용될 수 있다. 이에 따라, 홉 타입은 홉 격리 타입(hop-isolated vt)와 홉 병합 타입(hop-merged vt)을 가질 수 있다. 즉, 테넌트 내 병합인 경우 홉 격리 타입이고, 테넌트 간 병합인 경우 홉 병합 타입이다.
- [0048] 그러나, 종단 타입과 애플리케이션 타입에서는 각 호스트와 애플리케이션이 단일 가상 네트워크에 속하므로 테넌트 간에 가상 주소가 겹치는 문제가 있으며, 임의 타입에서는 테넌트 컨트롤러가 가상의 플로우 룰을 어떻게 구성하는지 확신할 수 없다는 문제가 있다. 이에 따라, 종단 타입, 애플리케이션 타입 및 임의 타입은 테넌트 간에 가상의 플로우 룰을 병합할 수 없으므로, 홉 타입과 달리 오로지 테넌트 내의 병합에서만 유효하다.
- [0049] 이러한 구문을 사용하여 테넌트는 전체 가상 네트워크, 특정 가상 스위치, 특정 호스트 또는 특정 애플리케이션에 대한 가상화 타입 정책을 지정할 수 있다.
- [0050] 이에 대하여 도 4의 각 엔트리를 참조하면, 첫 번째 엔트리는 모든 스위치, 호스트 및 애플리케이션과 매칭되므로 홉 병합 타입의 가상화 타입 정책이다. 두 번째 엔트리는 특정 가상 스위치를 지정하였으나 플로우 정보를 지정하지 않으므로, 홉 격리 타입의 가상화 타입 정책이다. 세 번째 엔트리는 플로우 정보에 호스트를 지정하였으므로 종단 타입의 가상화 타입 정책이고, 네 번째 엔트리는 플로우 정보에 애플리케이션을 지정하였으므로 애플리케이션 타입의 가상화 타입 정책이다.
- [0051] 한편, 가상화 타입에 따라 가상의 플로우 룰을 물리적 플로우 룰로 가상화하는 구성에 대한 상세한 설명은 후술하기로 한다.
- [0052] 우선 순위(21d)는 복수의 가상화 타입 정책이 가상의 플로우 룰에 적용가능할 때, 적용되는 순위를 의미한다. 각 가상화 타입 정책에 부여된 우선 순위(21d)에 따라, 가상의 플로우 룰의 가상화 타입(21c)을 선택할 때 상기 가상의 플로우 룰과 매칭되는 가상화 타입 정책들 중에서 가장 높은 우선 순위를 갖는 가상화 타입 정책의 가상화 타입(21c)을 선택하도록 할 수 있다.
- [0053] vf 가상화부(200)는 테이블 생성부(100)에서 생성된 가상화 타입 테이블을 이용하여 테넌트 컨트롤러로부터 생성된 가상의 플로우 룰에 대하여 가상화 타입을 선택하고, 선택된 가상화 타입에 따라 상기 가상의 플로우 룰을 상기 물리적 플로우 룰로 가상화한다.
- [0054] 가상의 플로우 룰에 대한 가상화 타입의 선택은 가상화 타입 테이블을 이용하여 수행될 수 있다. 보다 구체적으로, vf 가상화부(200)는 가상 스위치 ID와 가상화 타입 테이블의 스위치 ID를 비교하고, 병합하고자 하는 가상의 플로우 룰의 매칭 필드와 가상화 타입 테이블의 플로우 정보를 비교한다. 비교를 통해 가상화 타입 테이블에 포함된 적어도 하나 이상의 가상화 타입 정책 중에서 일치하는 가상화 타입 정책을 조회하고, 일치하는 가상화 타입 정책의 가상화 타입을 가상화 플로우 룰의 가상화 타입으로서 선택한다.
- [0055] 한편, 일치하는 가상화 타입 정책이 복수개가 있는 경우, vf 가상화부(200)는 각각의 가상화 타입 정책에 대한 우선 순위를 비교하고, 일치하는 가상화 타입 정책들 중에서 가장 높은 우선 순위를 갖는 가상화 타입 정책의 가상화 타입을 선택할 수 있다.
- [0056] 가상화 타입이 선택되면, vf 가상화부(200)는 선택된 가상화 타입에 따라 가상의 플로우 룰을 병합하여 물리적 플로우 룰로 가상화한다. 상술한 바와 같이, 가상의 플로우 룰을 물리적 플로우 룰로 가상화하는 것은 적어도 하나 이상의 가상의 플로우 룰을 가상화 타입에 따라 서로 병합하는 것을 의미할 수 있다.
- [0057] 가상의 플로우 룰 병합의 일 예를 도시한 도 5를 참조하면, 스위치 A, B, C, D를 포함하고, 스위치 A에서 B로 트래픽이 흐르는 네트워크에서, 스위치 B에 대하여 4개의 가상의 플로우 룰(31)이 있다. 이러한 가상의 플로우 룰(31)은 가상화 타입의 선택에 따라 병합된다.
- [0058] 예를 들어, 가상화 타입이 홉 타입으로 선택되면 호스트와 애플리케이션에 대한 플로우 정보가 와일드 카딩(wild carding)되고(33a), 모든 애플리케이션과 매칭되어 가상화 타입이 종단 타입으로 선택되면 애플리케이션에 대한 플로우 정보가 와일드 카딩되고(33b), 가상화 타입이 애플리케이션 타입으로 선택되면 특정 호스트 및 특정 애플리케이션을 지정한다(33c).
- [0059] 표시 규칙 설치부(300)는 물리적 스위치에 표시 규칙(indicator rule)을 설치한다. 표시 규칙 설치부(300)는 일 실시 예에 따라 가상화 타입 테이블에 애플리케이션 타입의 가상화 타입 정책이 있는 경우 표시 규칙을 설치할 수 있다.

- [0060] 보다 구체적으로, SDN(Software-Defined Networking)에서 진입 스위치에 새로운 패킷을 처리할 규칙이 설치되어 있지 않은 경우 진입 스위치는 네트워크 하이퍼바이저에 플로우 룰을 요청한다. 그러나, 상기 진입 스위치에 예를 들어 중단 타입, 흡-병합 타입 또는 흡 격리 타입과 같이 애플리케이션 타입보다 거친(coarser-grained) 가상화 타입의 물리적 플로우 룰이 설치되어 있으면 보다 세밀한 가상화 타입의 플로우 룰이 설치될 수 없다. 이에 따라, 진입 스위치는 네트워크 하이퍼바이저에 플로우 룰 생성에 대한 요청을 할 수 없다.
- [0061] 스위치의 플로우 테이블에 대하여 설치된 표시 규칙을 도시한 도 6을 참조하면, 우선 임의의 테넌트에 할당된 가상화 타입 테이블(미도시)에 각각 낮은 우선 순위의 중단 타입과 높은 우선 순위의 애플리케이션 타입의 가상화 타입 정책이 있다고 가정한다. 도 6과 같이 호스트 1과 호스트 2 사이의 경로에서 첫 번째 스위치인 스위치 A의 플로우 테이블(41)에는 이미 중단 타입으로 가상화된 물리적 플로우 룰이 있을 때, 포트 5001에서 포트 5002로의 첫 번째 패킷은 가상화 타입 테이블에 애플리케이션 타입의 가상화 타입 정책이 있음에도 불구하고 이미 설치된 중단 타입으로 가상화된 물리적 플로우 룰에 의해 포워딩(forwarding)된다. 따라서, 포트 5001에서 포트 5002로의 적절한 플로우 룰이 생성되지 않는다. 이는 상술한 바와 같이 물리적 스위치에 이미 거친 가상화 타입의 물리적 플로우 룰이 설치되어있기 때문에 발생한다.
- [0062] 이러한 문제점을 해결하기 위하여, 표시 규칙 설치부(300)는 가상화 타입 테이블에 애플리케이션 타입의 가상화 타입 정책이 있는 경우 표시 규칙(43)을 물리적 스위치에 생성한다. 여기서, 표시 규칙(43)은 첫 번째 애플리케이션 패킷을 네트워크 하이퍼바이저에 전달하는 규칙으로서 기 설치된 물리적 플로우 룰보다 우선 순위가 높게 설정된다. 이에 따라, 스위치 A는 포트 5001에서 포트 5002로의 첫 번째 패킷을 호스트 2가 아닌 네트워크 하이퍼바이저로 포워딩할 수 있다.
- [0063] 이러한 본 발명의 일 실시 예에 따른 표시 규칙 설치부(300)를 통해 물리적 스위치에 거친 가상화 타입의 물리적 플로우 룰이 설치되어 있더라도, 애플리케이션 타입의 가상화 타입 정책에 의한 물리적 플로우 룰을 설치할 수 있다.
- [0064] 한편, 진입 스위치의 이전 흡이 소스 호스트이므로, 진입 스위치의 플로우 룰은 호스트 주소를 매칭해야한다. 즉, 진입 스위치의 플로우 룰은 최소한 중단 타입의 가상화 타입을 가지므로, 표시 규칙 설치부(300)는 애플리케이션 타입에 대하여만 표시 규칙을 설치하면 된다.
- [0065] pf 가상화부(400)는 상기 생성된 물리적 플로우 룰이 물리적 스위치에 기 설치된 물리적 플로우 룰과 충돌하는지 여부에 대한 검사를 수행하고, 검사에 따라 기 설치된 물리적 플로우 룰과 충돌하면 상기 생성된 물리적 플로우 룰에 대하여 룰 맵을 이용하여 가상화를 수행한다.
- [0066] 여기서, 물리적 플로우 룰을 가상화하는 것은 룰 맵을 이용하여 충돌하는 물리적 플로우 룰의 가상화 타입을 보다 세밀한 가상화 타입으로 확장하는 것을 의미할 수 있다.
- [0067] 여기서, 룰 맵은 가상의 플로우 룰을 물리적 플로우 룰로 가상화할 때 필요한 가상의 플로우 룰과 물리적 플로우 룰의 맵핑을 관리하는 구조체이다.
- [0068] 충돌에 따른 물리적 플로우 룰의 가상화에 대하여 보다 상세히 설명하면, pf 가상화부(400)는 물리적 스위치에 기 설치된 물리적 플로우 룰이 존재하면, 설치하고자 하는 물리적 플로우 룰의 pf.A를 검사하여 기 설치된 물리적 플로우 룰과 충돌하는지 확인한다. 이러한 충돌은 가상의 플로우 룰이 모두 흡 타입(흡 병합 타입 또는 흡 격리 타입)으로 가상화될 때만 발생한다. 가상화는 플로우 룰 단위로 수행되는데, 가상의 플로우 룰 각각이 흡 타입의 물리적 플로우 룰로 가상화되어 물리적 스위치에 설치되는 경우 기 설치된 물리적 플로우 룰로 인하여 패킷의 포워딩이 불가능한 상황, 즉 충돌이 발생한다.
- [0069] pf 가상화부(400)는 상술한 바와 같이 충돌이 발생하는 경우 룰 맵을 이용하여 충돌이 발생하는 물리적 플로우 룰을 중단 타입으로 확장시켜서 가상화를 수행한다.
- [0070] pf 가상화의 일 예를 도시한 도 7을 참조하면, 도 7의 좌측에 도시된 테이블(51)은 테넌트 컨트롤러로부터 생성된 각 테넌트에 대한 가상의 플로우 룰이다. 가상의 플로우 룰에 대하여 흡 병합 타입의 가상화를 수행하면, 도 7의 우측 상단의 테이블(53)과 같이 흡 주소(이전의 흡 주소 및 현재의 흡 주소)가 일치하는 플로우 룰 간에 병합된다. 즉, 병합 이전의 상위 세 개의 플로우 룰은 흡 병합 타입으로 병합되고, 병합된 플로우 룰이 물리적 스위치에 설치되어 A로부터 B로 전송된 패킷을 C로 포워딩할 수 있다. 그러나, 나머지 플로우 룰이 흡 병합 타입으로 병합된 후에 물리적 스위치에 설치하고자 할 때, 기 설치된 물리적 플로우 룰의 수행하는 동작 정보(output action)와 충돌한다. 즉, 기 설치된 물리적 플로우 룰은 패킷을 C로 포워딩하는 반면, 설치하고자 하는

물리적 플로우 룰은 패킷을 D로 포워딩하므로 충돌이 발생한다.

- [0071] 이에 따라, pf 가상화부(400)는 룰 맵에 포함된 기존의 물리적 플로우 룰 정보를 이용하여 기 설치된 물리적 플로우 룰과 충돌하는 물리적 플로우 룰을 중단 타입으로 가상화하여 물리적 플로우 룰에 대한 가상화를 도 7의 우측 하단의 테이블(55)과 같이 수행한다.
- [0072] 한편, pf 가상화부(400)를 통해 가상화된 물리적 플로우 룰이 생성되면, 네트워크 하이퍼바이저는 가상화된 물리적 플로우 룰을 물리적 스위치에 설치하고 룰 맵에서 물리적 플로우 룰을 업데이트한다. 또한, 충돌이 발생하지 않는 것으로 판단될 경우 네트워크 하이퍼바이저는 물리적 플로우 룰을 물리적 스위치에 설치한다.
- [0074] 도 8은 본 발명의 일 실시 예에 따른 플로우 룰 가상화 방법의 순서도이다. 이하에서는 앞서 설명한 부분과 중복되는 부분에 대한 상세한 설명은 생략하기로 한다.
- [0075] 도 8을 참조하면, S510 단계는 테넌트로부터 가져온 가상화 타입 정책을 이용하여 가상화 타입 테이블을 생성한다. 여기서, 생성되는 가상화 타입 테이블의 각 엔트리 가상화 타입 정책으로 구성되고, 가상화 타입 테이블의 각 항목은 스위치 ID, 플로우 정보, 가상화 타입 및 우선 순위를 포함할 수 있다.
- [0076] S520 단계는 S510 단계에서 생성된 가상화 타입 테이블을 이용하여 테넌트 컨트롤러로부터 생성된 가상의 플로우 룰에 대하여 가상화 타입을 선택한다.
- [0077] 가상화 타입의 선택을 위하여, S520 단계는 가상 스위치 ID와 가상화 타입 테이블의 스위치 ID를 비교하고, 가상의 플로우 룰의 매칭 필드와 가상화 타입 테이블의 플로우 정보를 비교하여 일치하는 가상화 타입 정책을 조회하는 단계 및 일치하는 가상화 타입 정책들 중에서 가장 높은 우선 순위를 갖는 가상화 타입 정책의 가상화 타입을 선택하는 단계를 더 포함할 수 있다.
- [0078] S530 단계는 S520 단계에서 선택된 가상화 타입에 따라 가상의 플로우 룰을 물리적 플로우 룰로 가상화한다.
- [0079] S540 단계는 물리적 플로우 룰과 물리적 플로우 룰을 설치하고자 하는 물리적 스위치에 기 설치된 물리적 플로우 룰과 충돌하는지 여부를 검사한다.
- [0080] S540 단계의 검사 결과에 따라 물리적 플로우 룰과 물리적 스위치에 기 설치된 물리적 플로우 룰이 충돌하는 것으로 판단되면, S550 단계는 룰 맵을 이용하여 S530 단계를 통해 생성된 물리적 플로우 룰을 가상화한다. 여기서, S550 단계는 룰 맵을 이용하여 물리적 플로우 룰을 중단 타입으로 확장시켜 가상화를 수행할 수 있다.
- [0081] S560 단계는 충돌이 발생한 것으로 판단되어 물리적 플로우 룰에 대한 가상화가 수행되면 네트워크 하이퍼바이저는 가상화된 물리적 플로우 룰을 예를 들면 물리적 스위치와 같은 물리적 네트워크에 설치한다. 한편, 충돌이 발생하지 않는 것으로 판단되면 네트워크 하이퍼바이저는 물리적 플로우 룰을 물리적 네트워크에 설치한다.
- [0082] 이하에서는 본 발명의 일 실시 예에 따른 플로우 룰 가상화 장치(10) 및 방법과 기존의 네트워크 가상화 기법인 OVX(OpenVirteX)의 성능을 비교하기로 한다.
- [0084] **플로우 룰의 수**
- [0085] 도 9a는 토폴로지의 형태의 일 예를 도시한 것이며, 도 9b는 도 9a에 도시된 토폴로지의 형태에 따른 플로우 룰의 수의 비율을 비교한 것이다.
- [0086] 도 9a 및 도 9b를 참조하면, 각 토폴로지의 형태에 따른 플로우 룰의 수의 비율은 각 스위치에 대하여 총 플로우 룰의 수와 OVX 플로우 룰의 수의 비율을 의미한다. 이에 대하여, OVX는 총 플로우 룰의 수와 거의 동일한 수를 나타내는 반면에, 본 발명의 경우 특히 홉 병합 타입에서는 선형 토폴로지인 경우 플로우 룰의 수를 91[%]만큼 감소시켰으며, Internet2 토폴로지와 fat-tree 토폴로지에서도 각각 87[%]와 82[%]만큼 감소시켰다. 또한, 홉 격리 타입과 중단 타입에서도 홉 병합 타입과 유사한 수치만큼 플로우 룰의 수를 감소시켰다.
- [0088] **컴퓨팅 자원 사용량**
- [0089] 컴퓨팅 자원 사용량은 크게 네트워크 하이퍼바이저와 물리적 네트워크를 연결하는 제어 채널의 대역폭과 네트워크 하이퍼바이저에서의 CPU의 사용량 측면에서 성능을 비교할 수 있다.
- [0090] 아래의 표 2는 플로우 룰 수정 패킷(FlowMod)의 수와 대역폭을 나타낸 것이다.

표 2

Case	Linear	Fat-tree	Internet2
Hop-merged	1.8/321	4.0/689	2.8/485
Hop-isolated	2.6/460	4.5/761	2.9/512
End	5.3/902	10.4/1749	5.9/1003
Application	23.3/3939	37.4/6317	30.2/5131
Arbitrary	24.4/3650	34.1/5093	26.9/4037
Mixed	11.3/1839	20.0/3227	13.3/2152
OVX	22.0/4050	37.3/6914	28.6/5314

상기 표 2의 데이터는 플로우 룰 수정 패킷의 수/대역폭을 의미한다. 상기 표 2에서, 본 발명은 OVX와 비교하여 새로운 플로우에 대한 요청 수가 감소하였다. 또한, OVX가 최대 대역폭을 사용하는 반면에, 본 발명은 홉 기반의 패킷 전달을 위해 목적지 MAC 주소만 설정하는 작업을 추가하므로, 메시지의 길이와 대역폭이 OVX보다 짧다.

도 10은 네트워크 하이퍼바이저에서의 CPU의 사용량을 도시한 것이다.

도 10을 참조하면, OVX와 비교할 때 CPU의 사용량이 절반이상 감소한 것을 확인할 수 있다. 한편, 애플리케이션 타입과 임의 타입의 경우 플로우 룰의 수가 거의 감소하지 않았으므로, CPU 사용량에서도 큰 차이가 없다.

오버헤드

아래의 표 3은 플로우 룰을 가상화하기 위하여 추가로 지연되는 시간을 나타낸 것이다.

표 3

	Delay	Increased delay
OVX	5.05	-
Hop-merged	5.74	0.69
Hop-isolated	5.90	0.84
End	5.86	0.81
Application	5.86	0.80
Arbitrary	5.84	0.79
Mixed	5.98	0.92

상기 표 3을 참조하면, 추가로 지연되는 시간의 평균은 약 0.8[s]이다. 다만, 이러한 추가 지연은 플로우 룰 설치에 있어서 단 한 번만 발생하기 때문에, 전반적인 패킷 전달에서 큰 영향을 미치지 않음을 확인할 수 있다.

이상 살펴본 바와 같이 본 발명의 일 실시 예에 따른 플로우 룰 가상화 장치(10) 및 방법은 플로우 룰의 가상화를 통하여 플로우 룰의 수와 제어 채널 대역폭을 감소시킴으로써 P-NV에 존재하는 확장성의 문제를 비교적 적은 오버헤드로 해결할 수 있다.

이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로

(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

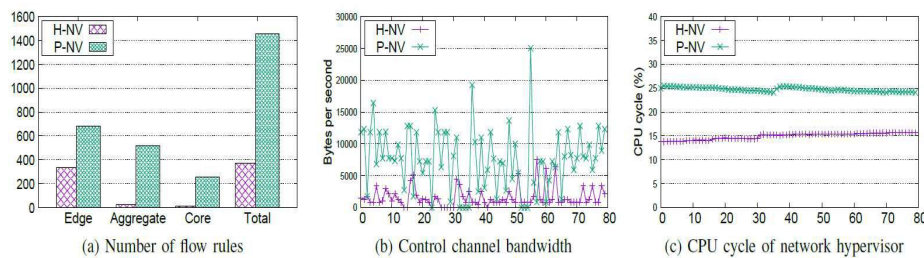
[0104] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic 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. 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 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0105] 이상과 같이 실시예들이 비록 한정된 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

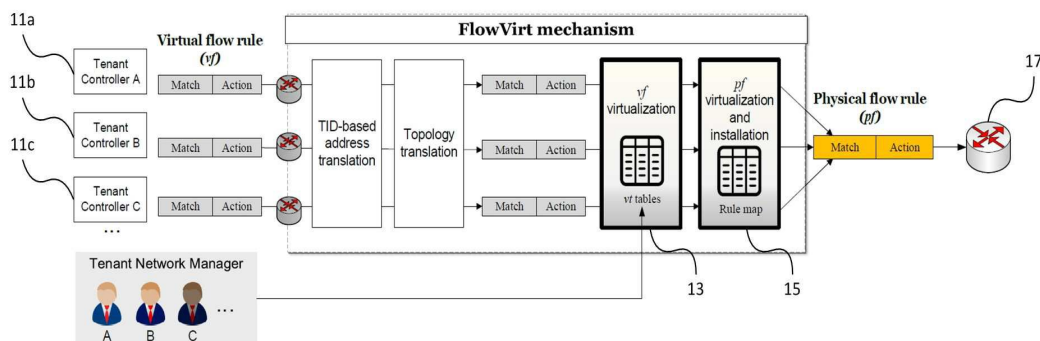
[0106] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

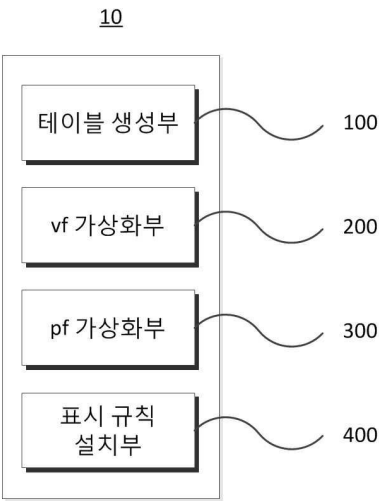
도면1



도면2



도면3



도면4

21a

21b

21c

21d

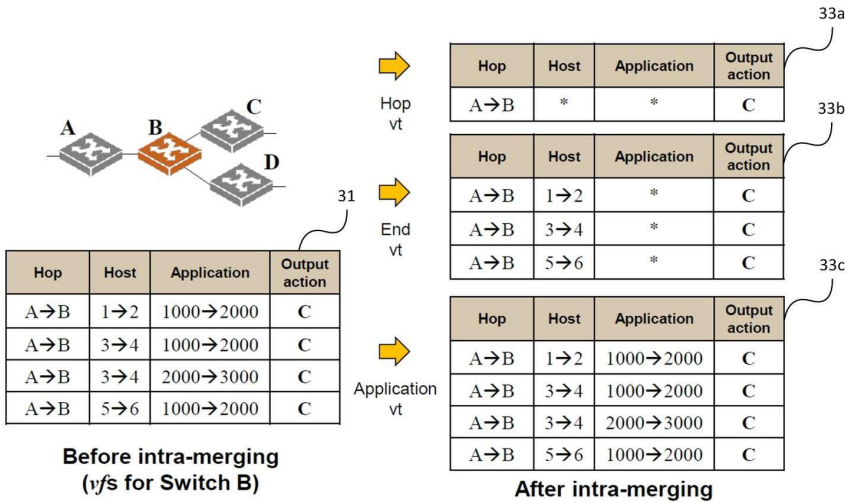
Flow information example

Switch ID	Ingress port	Ethernet source	Ethernet destination	Ethernet type	IP source	IP destination	IP protocol	Source port	Destination port	Virtualization type	Priority
*	*	*	*	*	*	*	*	*	*	Hop-merged	Low
4	*	*	*	*	*	*	*	*	*	Hop-isolated	High
*	*	*	*	*	10.0.0.1	10.0.0.2	*	*	*	End	Mid
*	*	*	*	*	*	*	*	*	12345	Application	Mid

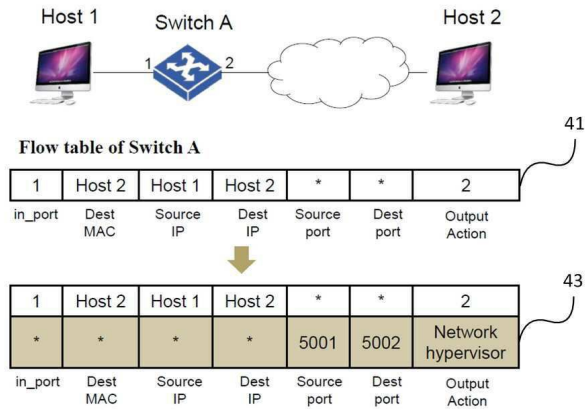
Range

- Entire virtual network
- Specific virtual switch
- Particular hosts
- Particular application

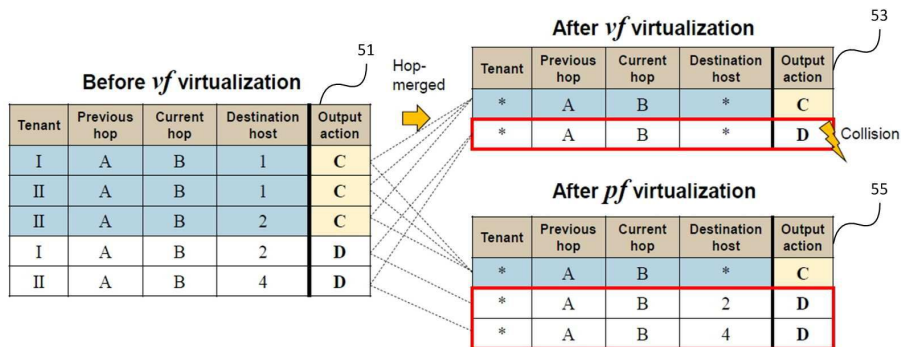
도면5



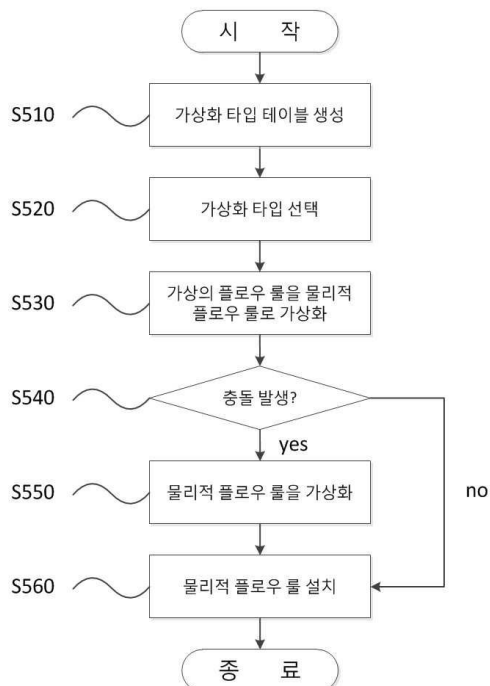
도면6



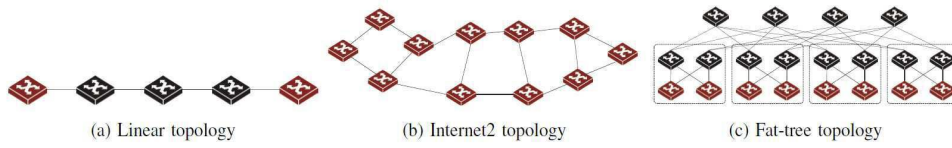
도면7



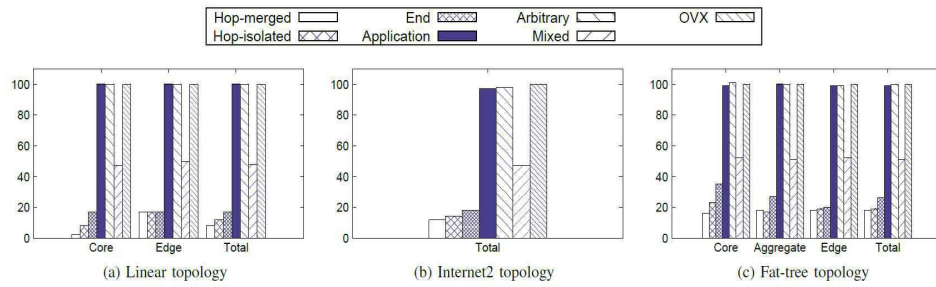
도면8



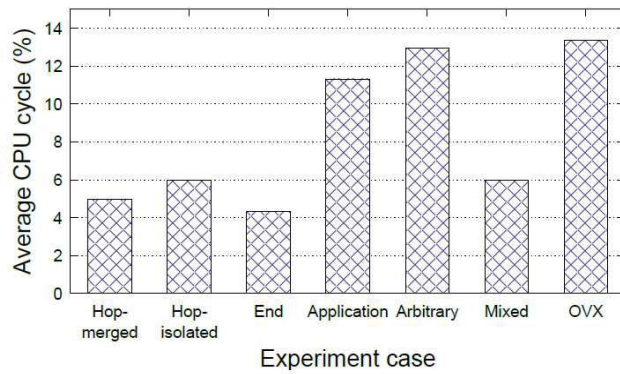
도면9a



도면9b



도면10



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 11항

【변경전】

상기 물리적 스위치

【변경후】

물리적 스위치