



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년01월07일
 (11) 등록번호 10-1583325
 (24) 등록일자 2015년12월31일

(51) 국제특허분류(Int. Cl.)
H04L 12/861 (2013.01)
 (21) 출원번호 10-2014-0112071
 (22) 출원일자 2014년08월27일
 심사청구일자 2014년08월27일
 (30) 우선권주장
 1020140104319 2014년08월12일 대한민국(KR)
 (56) 선행기술조사문헌
 KR100329911 B1*
 KR1020110048828 A*
 JP2013531299 A*
 JP2013543611 A*
 *는 심사관에 의하여 인용된 문헌

(73) 특허권자
주식회사 구버넷
 대전광역시 유성구 가정로 218 ,8-204(가정동,
 한국전자통신연구원)
정기용
 대전광역시 유성구 가정로 43, 111동 1401호 (신
 성동, 삼성한울아파트)
 (72) 발명자
정기용
 대전광역시 유성구 가정로 43, 111동 1401호 (신
 성동, 삼성한울아파트)
 (74) 대리인
리앤목특허법인

전체 청구항 수 : 총 12 항

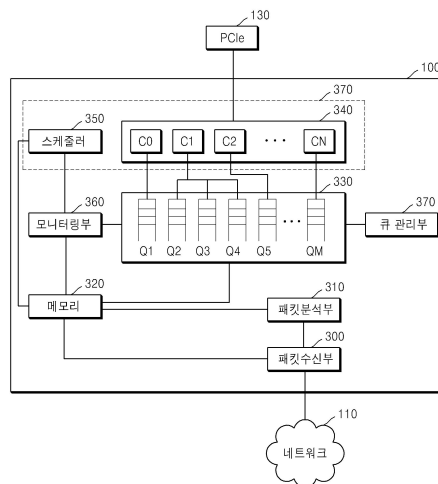
심사관 : 윤태섭

(54) 발명의 명칭 **가상 패킷을 처리하는 네트워크 인터페이스 장치 및 그 방법**

(57) 요약

가상 패킷을 처리하는 네트워크 인터페이스 장치 및 그 방법이 개시된다. 복수 개의 가상 머신이 구현된 서버와 연결된 네트워크 인터페이스 장치는, 복수 개의 큐를 포함하고, 물리적 망을 통해 가상 머신으로 전송할 가상 패킷을 수신하면, 가상 패킷의 가상 플로우를 식별하여 가상 플로우 단위로 복수 개의 큐에 가상 패킷을 저장한 후 다중 프로세서를 통해 병렬처리한다.

대표도 - 도3



명세서

청구범위

청구항 1

복수 개의 가상 머신이 구현된 서버와 연결된 네트워크 인터페이스 장치에 있어서,

각 가상머신에 할당되는 복수 개의 큐;

물리적 망을 통해 가상 머신으로 전송할 가상 패킷을 수신하는 패킷수신부;

가상 패킷이 전달되어야 할 목적지 가상 머신과 가상 패킷의 가상 플로우를 식별하는 패킷분석부; 및

상기 식별한 목적지 가상 머신에 할당된 복수 개의 큐에 상기 식별한 가상 플로우 단위로 가상 패킷을 저장하는 스케줄러;를 포함하고

상기 패킷분석부는, 가상 패킷에 포함된 가상 네트워크 환경의 가상 네트워크계층을 포함한 각 계층의 정보를 획득하고, 상기 가상 네트워크계층을 포함한 적어도 하나 이상이 계층을 분석하여 가상 플로우를 식별하는 것을 특징으로 하는 네트워크 인터페이스 장치.

청구항 2

제 1항에 있어서,

가상 플로우와 큐의 맵핑관계를 저장하는 플로우 테이블;를 더 포함하고,

상기 스케줄러는 상기 플로우 테이블을 참조하여 상기 패킷 분석부에 의해 식별된 가상 플로우와 맵핑된 큐에 상기 가상 패킷을 저장하되, 상기 플로우 테이블에 상기 패킷 분석부에 의해 식별된 가상 플로우 맵핑된 큐에 대한 정보가 없으면 목적지 가상 머신에 할당된 큐 중 하나를 선택하고, 상기 플로우 테이블을 갱신하는 것을 특징으로 하는 네트워크 인터페이스 장치.

청구항 3

제 1항에 있어서, 상기 복수 개의 큐는,

가상 머신별 적어도 하나 이상의 가상 플로우와 맵핑된 적어도 하나 이상의 큐로 구성되는 것을 특징으로 하는 네트워크 인터페이스 장치.

청구항 4

제 1항에 있어서,

적어도 하나 이상의 큐를 포함하는 가상 머신별로 할당된 복수 개의 큐 그룹을 포함하고,

상기 스케줄러는 가상 패킷에 대해, 가상 패킷의 목적지 가상 머신을 기초로 상기 복수 개의 큐 그룹 중 하나를 할당하고, 상기 가상 플로우를 기초로 선택된 큐 그룹 내 큐를 할당하는 것을 특징으로 하는 네트워크 인터페이스 장치.

청구항 5

제 1항에 있어서,

복수 개의 프로세서;를 더 포함하고,

각 프로세서는 상기 복수 개의 큐 중 적어도 하나 이상의 큐와 연결되는 것을 특징으로 하는 네트워크 인터페이스 장치.

청구항 6

제 5항에 있어서,

상기 스케줄러는 상기 복수 개의 프로세서 중 하나가 동적으로 선택되는 것을 특징으로 하는 네트워크 인터페이스 장치.

청구항 7

제 1항에 있어서,

상기 서버로부터 가상 머신에 관한 정보를 포함하는 가상화 환경 정보를 수신하고, 상기 가상화 환경 정보를 기초로 상기 복수 개의 큐를 동적으로 생성하는 큐관리부;를 더 포함하는 것을 특징으로 하는 네트워크 인터페이스 장치.

청구항 8

복수 개의 가상 머신을 위한 패킷 처리 방법에 있어서,

물리적 망을 통해 가상 패킷을 수신하는 단계;

가상 패킷이 전달되어야 할 목적지 가상 머신과 상기 가상 패킷의 가상 플로우를 식별하는 단계;

상기 목적지 가상 패킷에 할당된 복수 개의 큐에 상기 가상 패킷을 상기 가상 플로우 단위로 구분하여 저장하는 단계; 및

각 큐에 저장된 패킷을 목적지 가상 머신으로 전송하는 단계;를 포함하고,

상기 가상 플로우를 식별하는 단계는, 가상 패킷에 포함된 가상 네트워크 환경의 가상 네트워크계층을 포함한 각 계층의 정보를 획득하고, 상기 가상 네트워크계층을 포함한 적어도 하나 이상의 계층을 분석하여 플로우를 식별하는 단계;를 포함하고,

각 가상 머신별로 적어도 하나 이상의 큐가 할당되는 것을 특징으로 하는 패킷 처리 방법.

청구항 9

제 8항에 있어서, 상기 저장하는 단계는,

적어도 하나 이상의 큐를 포함하는 복수 개의 큐 그룹 중 상기 목적지 가상 머신을 기초로 큐 그룹을 선택하는 단계; 및

상기 가상 플로우를 기초로 상기 선택된 큐 그룹 내 패킷을 저장할 큐를 선택하는 단계;를 포함하는 것을 특징으로 하는 패킷 처리 방법.

청구항 10

제 9항에 있어서,

상기 복수 개의 큐 그룹은 가상 머신의 환경 정보에 따라 동적으로 생성되는 것을 특징으로 하는 패킷 처리 방법.

청구항 11

제 1항에 있어서,

가상 머신에 관한 정보를 포함하는 가상화 환경 정보 또는 부하분포에 따라 각 가상 머신에 할당되는 큐의 개수가 동적으로 결정되는 것을 특징으로 하는 네트워크 인터페이스 장치.

청구항 12

제 8항에 있어서,

각 가상 머신에 할당되는 큐의 개수는 가상 머신에 관한 정보를 포함하는 가상화 환경 정보 또는 부하분포를 기초로 동적으로 결정되는 것을 특징으로 하는 패킷 처리 방법.

발명의 설명

기술 분야

[0001] 본 발명은 네트워크 인터페이스 장치 및 패킷 처리 방법에 관한 것으로서, 보다 상세하게는 가상 패킷을 처리할 수 있는 네트워크 인터페이스 카드와 이를 이용한 가상 패킷 처리 방법에 관한 것이다.

배경 기술

[0002] 최근 인터넷을 통하는 통신량이 급증하고, 이에 따라 서버의 대용량화, 고속화가 빠르게 진행되고 있다. 한편, 서버의 대용량화에 따르는 물리적 부피 증가를 해소하고 비용 절감 등을 피하기 위하여 서버의 가상화가 가속되고 있다. 서버의 대용량화, 고속화, 가상화에 따라 물리적 네트워크로부터 수신되는 가상화 환경에서 생성된 데이터 패킷을 포함하는 대용량 데이터에 대한 병렬처리의 고효율화가 필수적으로 요구되고 있으며, 가상화 서버에서 가상스위치 기능이 수행될 때 서버의 부하 증가에 기인하는 성능 저하가 야기됨에 따라 가상스위치 기능에 따르는 서버의 부하를 물리적 네트워크 인터페이스 장치로 이전하는 기술 개념 실현이 요구되고 있다.

[0003] 종래의 가상화 환경을 지원하는 NIC의 경우, 물리적 네트워크 인터페이스 장치에서 가상화 환경을 지원하는 방법으로 가상 머신 단위로 큐를 생성 관리하여 네트워크 인터페이스 장치와 서버의 가상 스위치 사이의 병목 현상을 감소시키는 시도가 있다. 그러나 종래의 경우에는 수신된 데이터 패킷의 병렬처리를 위한 프로세서 할당 및 큐의 재분배 시 가상 머신 단위로만 이루어진다. 즉 가상화 환경의 물리적 계층만이 고려된 프로세서 할당이 이루어진다. 따라서 병렬처리에서 프로세싱 효율을 높이는 데 아주 중요한 요소 중 하나인 프로세서 친화도가 고려될 수 없고, 프로세서의 사용 부하만을 고려하여 프로세서 할당 및 큐의 재분배가 일어난다. 이는 병렬 처리의 효율을 감소시키는 요인으로 작용될 수 있다.

선행기술문헌

특허문헌

[0004] (특허문헌 0001) 미국특허공개공보 제2013-0239119호

발명의 내용

해결하려는 과제

[0005] 본 발명이 해결하고자 하는 기술적 과제는, 패킷을 가상 플로우 단위로 처리하여 병렬 프로세싱의 효율을 높이고, 가상 플로우 단위로 QoS를 보장하며, 가상 네트워크 환경에서 서버의 부하를 분산시키는 네트워크 인터페이스 장치 및 패킷 처리 방법을 제공하는 데 있다.

과제의 해결 수단

[0006] 상기의 기술적 과제를 달성하기 위한, 본 발명에 따른 네트워크 인터페이스 장치의 일 예는, 복수 개의 가상 머신이 구현된 서버와 연결된 네트워크 인터페이스 장치에 있어서, 복수 개의 큐; 물리적 망을 통해 가상 머신으로 전송할 가상 패킷을 수신하는 패킷수신부; 가상 패킷의 가상 플로우를 식별하는 패킷분석부; 및 상기 식별한 가상 플로우 단위로 상기 복수 개의 큐에 가상 패킷을 저장하는 스케줄러;를 포함한다.

[0007] 상기의 기술적 과제를 달성하기 위한, 본 발명에 따른 패킷 처리 방법의 일 예는, 복수 개의 가상 머신을 위한 패킷 처리 방법에 있어서, 물리적 망을 통해 가상 패킷을 수신하는 단계; 가상 패킷의 가상 플로우를 식별하는 단계; 상기 가상 패킷을 상기 가상 플로우 단위로 구분하여 큐에 저장하는 단계; 및 각 큐에 저장된 패킷을 목적지 가상 머신으로 전송하는 단계;를 포함한다.

발명의 효과

[0008] 본 발명에 따르면, 복수의 가상 머신을 포함하는 가상화 환경을 구비한 서버의 부하가 감소 된다. 또한 가상 플로우 단위로 패킷을 처리함으로써 가상 패킷과 프로세서 사이의 친화도를 높여 병렬 프로세싱의 효율을 높인다. 또한 가상 스위치의 부하를 네트워크 인터페이스 카드로 분산시켜 가상 네트워크 프로세싱의 효율을 높일 수 있

다. 또한 가상 플로우 단위로 큐잉하여 처리함으로써 가상 머신 종단간 가상플로우 단위의 QoS가 보장되는 스케일러블한 통신 처리를 구현할 수 있다.

도면의 간단한 설명

- [0009] 도 1은 본 발명에 따른 네트워크 인터페이스 장치를 포함한 전체 시스템의 개략적인 구조를 도시한 도면,
- 도 2는 본 발명에 따른 네트워크 인터페이스 장치의 자원을 동적으로 설정하는 방법의 일 예를 도시한 도면,
- 도 3은 본 발명에 따른 네트워크 인터페이스 장치의 일 실시예의 구성을 도시한 도면,
- 도 4는 본 발명에 따른 네트워크 인터페이스 장치의 가상 플로우 기반 큐 할당의 일 예를 도시한 도면,
- 도 5는 본 발명에 따른 네트워크 인터페이스 장치의 가상 플로우 기반 큐 할당의 다른 일 예를 도시한 도면,
- 도 6은 본 발명에 사용되는 가상 패킷의 일 예를 도시한 도면, 그리고,
- 도 7은 본 발명에 따른 가상 네트워크 환경을 위한 패킷 처리 방법의 일 예를 도시한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0010] 이하에서, 첨부된 도면들을 참조하여 본 발명에 따른 네트워크 인터페이스 장치 및 패킷 처리 방법에 대해 상세히 설명한다.
- [0011] 도 1은 본 발명에 따른 네트워크 인터페이스 장치를 포함한 전체 시스템의 개략적인 구조를 도시한 도면이다.
- [0012] 도 1을 참조하면, 네트워크 인터페이스 장치는 네트워크 인터페이스 카드(Network Interface Card, NIC)(100)로 구현된다. 다만 네트워크 인터페이스 장치는 네트워크 인터페이스 카드(100)에 반드시 한정되는 것은 아니며, 서버 내외에서 하드웨어 또는 소프트웨어 등의 다양한 형태로 구현될 수 있다. 설명의 편의를 위하여 이하, 네트워크 인터페이스 장치를 NIC로 표현한다.
- [0013] 서버(120)는 복수 개의 가상 머신(150,152,154)과, 가상 스위치(140) 및 연결슬롯(130)을 포함한다. 가상 스위치(140)는 NIC(100)를 통해 수신한 가상 패킷을 목적지 가상 머신으로 전달한다. 연결슬롯(130)은 NIC(100)와 서버(120)를 연결하는 인터페이스이며, 일 예로 PCIe(Peripheral Component Interconnect Express)로 구현될 수 있다. 이 경우, NIC(100)는 PCIe 슬롯에 탈부착될 수 있다.
- [0014] NIC(100)는 네트워크(110)로부터 수신한 가상 패킷에 대해 가상화 환경에서의 상위 계층의 트래픽 특성을 분석하여 가상 플로우를 식별하고, 식별한 가상 플로우를 다중 프로세서를 통해 병렬 처리한다. 여기서 가상 패킷이라 함은 종래의 다양한 터널링과 같은 기법 등을 이용하여 일반 네트워크 패킷에 가상 네트워크 환경의 계층(Layer) 정보를 인캡슐레이션(encapsulation)한 패킷을 의미하며, 가상 플로우는 인캡슐레이션된 가상 패킷의 물리적 네트워크 프레임이 제거한 가상 네트워크 프레임에서의 상위 계층(vL3 이상) 트래픽 속성에 따라 분류된 가상 환경에서의 특정 트래픽으로 정의할 수 있다. 가상 플로우는 기 설정된 여러 정책에 따라 분류되고 식별될 수 있다. 예를 들어, 패킷분석부(310)는 가상 머신의 TCP 플로우를 가상 플로우로 식별할 수 있다. 가상 패킷의 구조에 대해서는 도 6을 참조하여 설명한다.
- [0015] NIC(100)는 수신한 가상 패킷을 병렬 처리하기 위한 복수 개의 큐와 복수 개의 프로세서들을 포함하며, 큐의 크기와 개수 등은 고정되거나, 서버의 가상화 환경이나 프로세서들의 부하 등에 따라 동적으로 변경될 수 있다.
- [0016] 도 2는 본 발명에 따른 NIC의 자원을 동적으로 설정하는 방법의 일 예를 도시한 도면이다.
- [0017] 도 1 및 도 2를 함께 참조하면, NIC(100)가 서버(120)의 연결슬롯(130)에 부착되어 서버(120)와 연결되는 경우(S200), NIC(100)는 서버(120)로부터 가상 머신의 수를 포함한 가상화 환경 정보를 수신한다(S210). NIC(100)는 수신한 가상화 환경 정보에 따라 큐의 크기와 개수, 큐 그룹의 생성 등 자원을 동적으로 설정한다(S220).
- [0018] 예를 들어, NIC(100)가 서버(120)로부터 가상 머신이 4개라는 가상화 환경 정보를 수신하는 경우, NIC는 12개의 큐를 각 가상 머신별로 3개씩 할당할 수 있다. 가상 머신별 할당되는 큐의 개수나 각 큐의 크기 등은 기 설정된 규칙에 따라 다양하게 설정될 수 있다.
- [0019] 도 3은 본 발명에 따른 NIC의 일 실시예의 구성을 도시한 도면이다.
- [0020] 도 3을 참조하면, NIC(100)는 패킷수신부(300), 패킷분석부(310), 메모리(320), 복수 개의 큐(330), 복수 개의 프로세서(340), 스케줄러(350), 모니터링부(360) 및 큐 관리부(370)를 포함한다. 패킷수신부(300) 등을 포함한

각 구성요소 사이의 연결선은 본 발명의 이해를 돕기 위한 하나의 예일 뿐이며, 큐 관리부(370)와 모니터링부(360) 사이의 연결, 스케줄러(350)와 복수 개의 큐(330) 사이의 연결 등 다양한 연결 관계가 설정될 수 있음은 물론이다.

- [0021] 패킷수신부(300)는 외부 네트워크에서 일반적인 이더넷 프레임으로 인식되도록 종래의 다양한 터널링과 같은 방법 등을 통해 인캡슐레이션된 가상 패킷을 수신하면 이를 디캡슐레이션하여 물리적 네트워크에 해당하는 헤더 부분을 제거하고 가상화 환경에서의 데이터 패킷 프레임으로 복원한다.
- [0022] 패킷분석부(310)는 디캡슐레이션된 가상 패킷의 가상 플로우를 식별한다. 가상 플로우를 식별하기 위해서는 가상화 환경에서의 데이터링크계층(vL2 레이어)만이 아니라 네트워크계층(vL3 레이어) 이상의 상위 계층까지 해석되어야 한다. 이를 위해, 패킷분석부(310)는 DPI(Deep Packet Inspection) 과정을 통해 디캡슐레이션된 가상 패킷의 가상 데이터링크계층(vL2 레이어)부터 가상 애플리케이션계층(vL7 레이어)까지 분석하여, 가상 플로우를 식별한다. 가상 플로우 식별을 위한 가상 패킷에 대한 분석은 가상 데이터링크계층부터 가상 애플리케이션계층 모두를 분석하는 것으로 한정되는 것은 아니며, 가상 플로우 식별 정책에 따라 분석의 범위는 달라질 수 있다.
- [0023] 메모리(320)는 가상 패킷과 패킷분석부(310)에 의해 식별된 가상 플로우 정보 등을 저장하고, 또한 가상 플로우와 큐 사이의 맵핑관계를 나타내는 플로우 테이블을 저장하고 관리한다.
- [0024] 일 실시 예로, 패킷수신부(300)는 디캡슐레이션된 가상 패킷을 메모리(320)에 저장하고, 가상 패킷의 저장 사실을 패킷분석부(310)에 통보한다. 그러면 패킷분석부(310)는 메모리에 저장된 해당 가상 패킷에 대한 가상 플로우 식별을 수행한다. 즉, 새로운 가상 패킷의 수신 사실을 알게 된 패킷분석부(310)는 기 설정된 정책에 따라 해당 가상 패킷의 가상 플로우 특성을 식별하여 그 정보를 저장하고 저장 사실을 스케줄러(350)에 알려준다.
- [0025] 스케줄러(350)는 식별된 가상 플로우를 해당 가상 플로우 큐에 할당하며, 가상 플로우 큐를 다중 프로세서(340)에 병렬 할당한다. 보다 구체적으로, 스케줄러(350)는 메모리(320)에 저장된 플로우 테이블을 참조하여 가상 패킷의 가상 플로우가 맵핑된 큐를 검색하고, 메모리(320)에 저장된 가상 패킷을 검색된 큐에 전달한다. 만약 테이블에 해당하는 가상 플로우가 존재하지 않는다면, 스케줄러(350)는 종래의 다양한 방법을 통해 가상 플로우를 특정 큐에 할당하고, 가상 플로우와 큐사이의 맵핑관계를 플로우 테이블에 저장한다.
- [0026] 스케줄러(350)는 가상 머신별 가상 패킷을 가상 플로우 단위로 큐잉할 수 있다. 예를 들어, 가상 플로우와 큐사이의 맵핑 관계를 설정할 때, 제1 가상머신과 제2 가상머신을 향하는 동일한 성격(예를 들어, 동일한 QoS 우선순위)의 제1 플로우와 제2 플로우가 동일한 큐에 할당될 수 있다. 본 발명이 이러한 경우를 배제하는 것은 아니나 병렬 처리의 효율을 보다 높이기 위하여, 가상 플로우를 가상 머신별로 서로 다른 그룹의 큐에 할당하는 것이 바람직하다. 다시 말해, 스케줄러(350)는 도 4와 같은 가상머신별로 큐들을 그룹핑하는 경우에, 제1 가상머신에 대한 제1 플로우는 제1 그룹(400)의 큐에 가상 플로우 단위로 할당하고, 제2 가상머신에 대한 제2 플로우는 제2 그룹(410)의 큐에 가상 플로우 단위로 할당한다.
- [0027] 예를 들어, 새로운 가상 패킷이 메모리(320)에 적재되었다는 사실과 해당 가상 패킷의 가상 플로우 정보를 수신하면, 스케줄러(350)는 플로우 테이블을 참조하여 가상 플로우가 어떤 가상 플로우 큐에 할당되어 있는지를 검색하고, 메모리(320)에 적재되어 있는 가상 패킷을 검색된 해당 큐에 적재할 수 있도록 한다. 만약 식별된 가상 플로우에 대한 정보를 플로우 테이블에서 찾을 수 없는 경우, 스케줄러(350)는 해당 가상 패킷을 기 설정된 정책에 따라 해당 가상 머신에 속한 큐 중 하나에 할당할 수 있다. 여기서 기 설정된 정책은 실시 예에 따라 다양할 수 있으며, 일 예로 플로우 친화도를 고려하여 가상 플로우 큐를 선택하는 정책, 가상 패킷에 해당하는 가상 머신 내 큐 중 가장 부하가 적은 큐를 선택하는 정책, 사용율이 가장 낮은 프로세서에 할당되어 있는 큐를 선택하는 정책 등이 있다.
- [0028] 복수 개의 큐(330)는 각각 적어도 하나 이상의 가상 플로우와 맵핑된다. 가상 플로우 단위로 큐잉하는 경우 프로세서 친화도가 높아지며, 이에 따라 병렬 처리의 효율도 증가한다. 복수 개의 큐(330)는 가상 머신별로 적어도 하나 이상의 큐를 포함하는 그룹으로 분할될 수 있다. 또한 복수 개의 큐(330)는 도 5와 같이 적어도 두 개 이상의 파티션으로 분할될 수 있다.
- [0029] 스케줄러(350)는 복수 개의 프로세서 중에서 선택된 프로세서일 수 있다. 예를 들어, 전체 프로세서들(370) 중 특정 프로세서(350)를 스케줄러로 지정하거나, 각 프로세서들의 부하 정도를 모니터링부(360)를 통해 파악한 후 부하가 가장 적은 프로세서를 스케줄러(350)로 선택할 수 있다. 이 외에도 스케줄러를 선택하기 위한 다양한 방법이 적용될 수 있다. 프로세서 중 스케줄러가 지정되는 경우, 제어부(미도시)는 스케줄링이 필요할 때마다 인터럽트 신호를 발생하여 스케줄러로 지정된 프로세서에 전송하고, 인터럽트 신호를 수신한 프로세서는 하던 작

업을 중지하고 스케줄러로서 동작을 완료한 후 다시 이전 작업을 수행한다.

- [0030] 복수 개의 프로세서들(340)은 각 큐에 저장된 가상 패킷을 병렬 처리하여 서버의 가상 머신으로 전송한다. 복수 개의 프로세서들(340)은 적어도 하나 이상의 큐와 연결된다.
- [0031] 예를 들어, 복수 개의 프로세서들(340)은 플로우 친화도를 고려하여 큐와 연결된다. 다시 말해, 동일하거나 유사한 가상 플로우 속성을 갖는 가상 패킷을 저장하는 큐들을 묶어 프로세서와 연결한다.
- [0032] 또 다른 예로서, 복수 개의 프로세서들은 가상 머신별로 큐와 연결될 수 있다. 도 4를 참조하면, 제1 프로세서는 제1 가상머신에 할당된 제1 내지 제3 큐와 연결되고, 제2 프로세서는 제2 가상머신에 할당된 제4 내지 제6 큐와 연결되고, 제3 프로세서는 제3 가상머신에 할당된 제7 및 제8 큐에 연결될 수 있다.
- [0033] 또 다른 예로서, 제1 프로세서는 제1 가상머신에 할당된 제1 내지 제3 큐와 함께 제2 가상머신에 할당된 제4 큐와 연결되고, 이 경우 제2 프로세서는 제2 가상머신에 할당된 제5 및 제6큐와 연결될 수 있다. 즉, 프로세서는 적어도 둘 이상의 가상머신에 할당된 큐의 전부 또는 일부와 연결될 수도 있다.
- [0034] 모니터링부(360)는 프로세서(340)와 큐(330)의 부하 등을 포함한 각종 상태를 모니터링한다.
- [0035] 큐 관리부(370)는 모니터링 결과에 따라 큐들을 도 5와 같이 복수의 파티션으로 분할하고 각 파티션마다 스케줄러를 두어 처리하거나, 복수의 큐를 하나로 합치거나 분할하고, 또는 가상머신에 할당된 큐의 수를 늘리거나 줄이는 등 큐의 크기와 개수 등을 조정한다. 큐 관리부는 앞서 살핀 도 2의 과정을 통해 파악된 서버의 가상화 환경에 따라서도 가상머신별 큐의 개수나 크기 등을 동적으로 설정할 수 있다.
- [0036] 도 4는 본 발명에 따른 NIC의 가상 플로우 기반 큐 할당의 일 예를 도시한 도면이다.
- [0037] 도 4를 참조하면, 큐들(330)은 가상 머신별로 구분된다. 예를 들어, 제1 내지 제3 큐(400)는 제1 가상머신에 할당되고, 제4 내지 제6 큐는 제2 가상머신에 할당되고, 제7 및 제8 큐는 제3 가상머신에 할당된다. 스케줄러는 가상 머신별로 가상 플로우를 참조하여 큐잉을 수행한다.
- [0038] 예를 들어, 제1 가상머신으로 향하는 가상 플로우를 우선순위에 따라 식별하는 경우에, 스케줄러는 제1 가상머신에 할당된 제1 내지 제3 큐(400)에 우선순위를 기반으로 가상 패킷을 분류하여 저장한다.
- [0039] 도 5는 본 발명에 따른 NIC의 가상 플로우 기반 큐 할당의 다른 일 예를 도시한 도면이다.
- [0040] 도 5를 참조하면, 큐들(330)은 적어도 둘 이상의 파티션(520,530)으로 구분된다. 각 파티션(520,530)마다 스케줄러(500,510)가 할당된다. 예를 들어, 제1 파티션(520)에 제1 스케줄러(500)가 할당되고, 제2 파티션(530)에 제2 스케줄러(510)가 할당된다. 각 스케줄러(500,510)는 할당된 파티션에 대하여 독립적으로 스케줄링 작업을 병렬 수행한다. 스케줄러는 앞서 설명한 바와 같이 복수 개의 프로세서들(370) 중 소정의 방법에 의해 선택된 프로세서일 수 있다.
- [0041] 예를 들어, 도 3과 같이 하나의 스케줄러에 의한 스케줄링 수행 중에, 모니터링부에서 측정된 큐의 부하분포가 기 설정된 임계치 이하로 떨어지면, 큐의 재분배 또는 프로세서 재할당이 결정될 수 있다. 또는 네트워크로부터 수신되는 가상 패킷의 통계적 양과 NIC 내의 총 프로세서에 의해 수행되는 프로세서 능력을 계산하여 프로세서의 부하가 어느 임계치 이하이면 큐의 재분배 또는 프로세서 재할당이 결정될 수 있다. 큐의 재분배 또는 프로세서 재할당시, 도 5와 같이 큐가 복수 개의 파티션으로 구분되고 추가적인 스케줄러의 지정이 필요한 경우, 가장 부하가 적은 프로세서를 추가 스케줄러로 지정할 수 있다.
- [0042] 각 파티션에 속한 큐들은 가상머신 기반으로 그룹핑(540)될 수 있으며, 그룹(540) 내 큐들은 가상 플로우 기반으로 분류될 수 있다. 이 경우, 파티션 - 가상머신별 그룹 - 각 그룹별 플로우 단위 큐의 계층적 구조가 생성된다.
- [0043] 도 6은 본 발명에 사용되는 가상 패킷의 일 예를 도시한 도면이다.
- [0044] 도 6을 참조하면, 가상 패킷은 물리적 네트워크 프레임(610), 터널링 필드(620), 가상 네트워크 프레임(630), 데이터 필드(600)를 포함한다.
- [0045] 물리적 네트워크 프레임(610)은 L2, IP, TCP 등 종래의 물리적 네트워크의 계층을 나타내는 정보를 포함한다. 터널링 필드(620)는 터널링 정보 등을 나타낸다. 가상 네트워크 프레임(630)은 가상 네트워크 환경에서의 각 계층(vL2 ~ vL7 등)에 대한 정보를 포함한다. 데이터 필드(600)는 데이터를 포함한다.
- [0046] 도 6의 가상 패킷의 구조는 본 발명의 이해를 돕기 위한 하나의 예일 뿐, 본 발명이 이에 한정되는 것은 아니며

가상 네트워크 환경을 위한 다양한 형태의 가상 패킷의 구조를 정의하여 사용할 수 있다.

[0047] 또한 메모리에 저장된 가상 패킷의 구조와 큐에 저장된 가상 패킷의 구조는 실시 예에 따라 동일하거나 다를 수 있다. 예를 들어, 네트워크로부터 수신한 도 6의 가상 패킷을 가상화 환경에서 처리 가능한 최적의 구조로 변경하거나 가상 패킷의 필드 중 가상화 환경에서 불필요한 필드의 일부 또는 전부를 삭제하는 등 다양하게 설계 변경가능하여 큐에 저장할 수 있다.

[0048] 도 7은 본 발명에 따른 가상 네트워크 환경을 위한 패킷 처리 방법의 일 예를 도시한 흐름도이다.

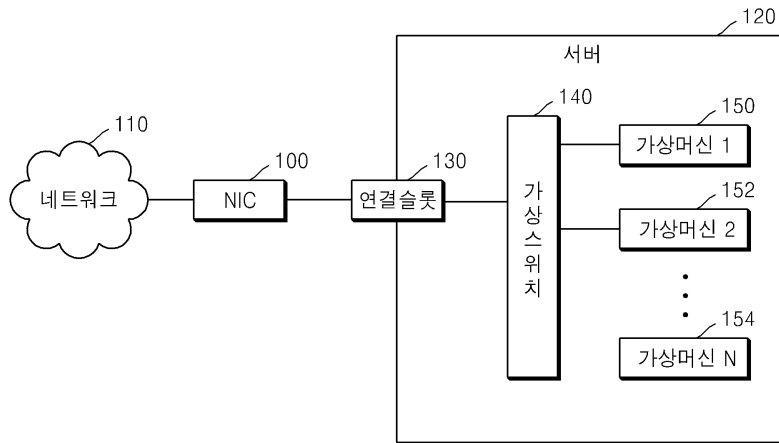
[0049] 도 7을 참조하면, 가상 네트워크 장치는 가상 패킷을 수신하면(S700), DPI 과정 등을 통해 가상 패킷을 분석하여 가상 패킷이 전달되어야 할 목적지 가상 머신과 가상 플로우를 식별한다(S710). 가상 네트워크 장치는 가상 머신별로 할당된 적어도 한 개 이상의 큐들에 대해 가상 플로우 단위로 해당하는 큐에 가상 패킷을 저장한다(S720). 그리고 가상 네트워크 장치는 복수 개의 프로세서들을 통해 각 큐에 저장된 가상 패킷을 처리하여 가상 머신으로 전송한다(S730).

[0050] 본 발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록매체의 예로는 다양한 형태의 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광데이터 저장장치 등이 있다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

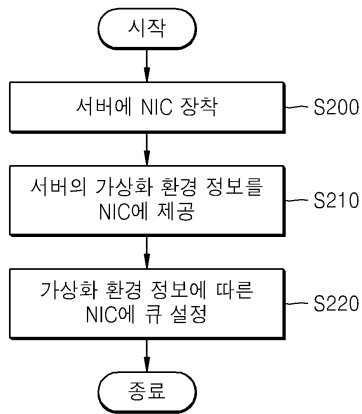
[0051] 이제까지 본 발명에 대하여 그 바람직한 실시예들을 중심으로 살펴보았다. 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

도면

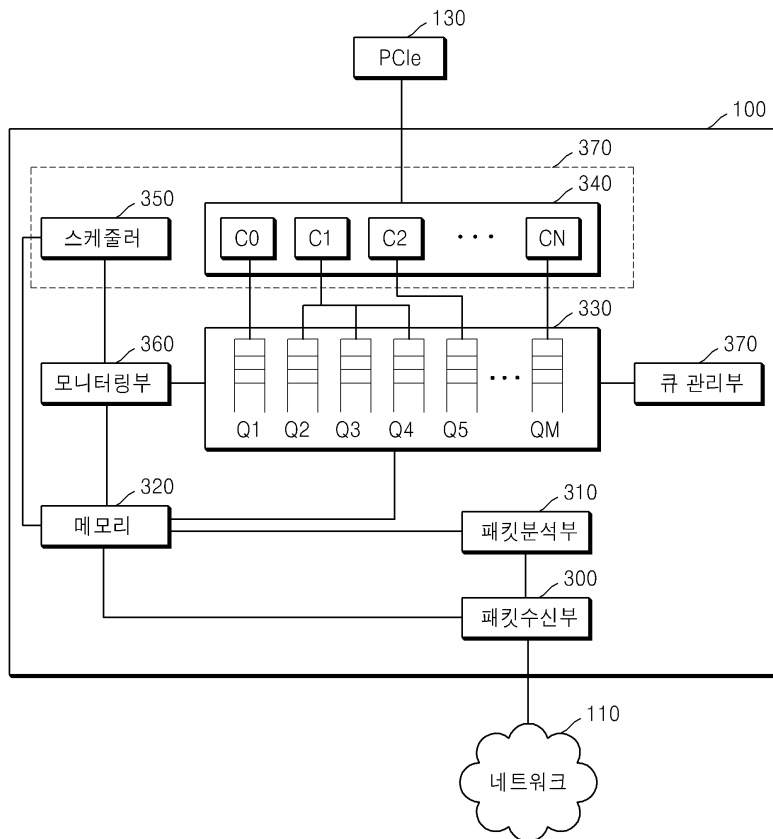
도면1



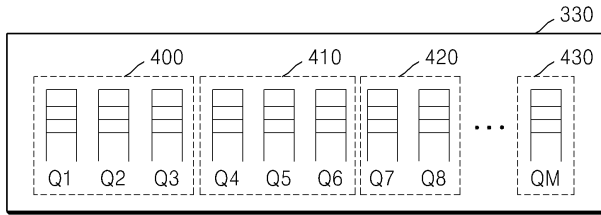
도면2



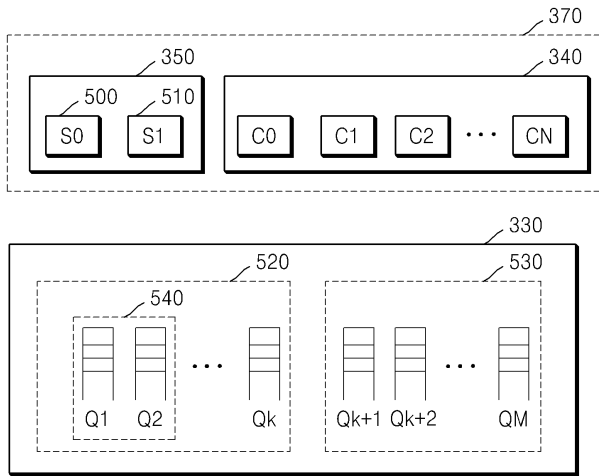
도면3



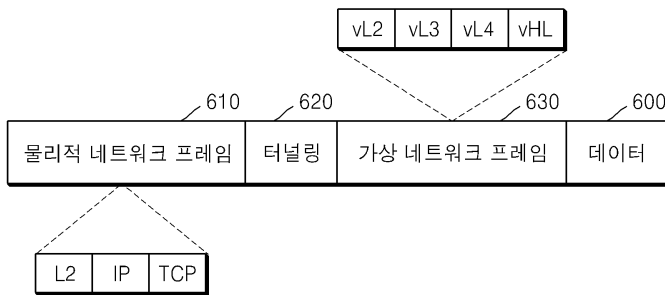
도면4



도면5



도면6



도면7

