

(12) 특허협력조약에 의하여 공개된 국제출원

(19) 세계지식재산권기구
국제사무국

(43) 국제공개일
2015년 12월 30일 (30.12.2015)



(10) 국제공개번호
WO 2015/199366 A1

- (51) 국제특허분류:
G06F 9/48 (2006.01) G06F 9/50 (2006.01)
- (21) 국제출원번호: PCT/KR2015/005914
- (22) 국제출원일: 2015년 6월 12일 (12.06.2015)
- (25) 출원언어: 한국어
- (26) 공개언어: 한국어
- (30) 우선권정보:
10-2014-0079129 2014년 6월 26일 (26.06.2014) KR
10-2014-0104319 2014년 8월 12일 (12.08.2014) KR
10-2014-0112071 2014년 8월 27일 (27.08.2014) KR
10-2015-0046740 2015년 4월 2일 (02.04.2015) KR
- (71) 출원인: (주) 구버넷 (GUBERNET INC.) [KR/KR]; 305-700 대전시 유성구 가정로 218, 8-204, Daejeon (KR).
- (72) 발명자; 겸
- (71) 출원인 : 정기웅 (JUNG, Kiung) [KR/KR]; 305-707 대전시 유성구 가정로 43, 111-1401, Daejeon (KR).
- (74) 대리인: 리앤목 특허법인 (Y.P.LEE, MOCK & PARTNERS); 135-971 서울시 강남구 언주로 30길 13 대림아크로텔 12층, Seoul (KR).

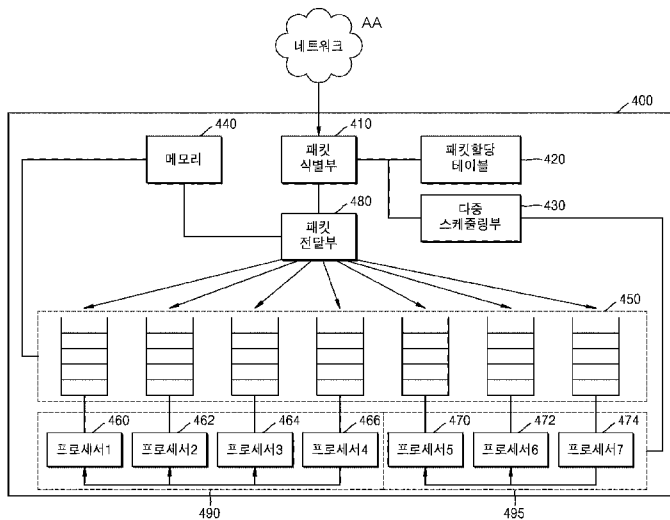
- (81) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 국내 권리의 보호를 위하여): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) 지정국 (별도의 표시가 없는 한, 가능한 모든 종류의 역내 권리의 보호를 위하여): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 유라시아 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 유럽 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

공개:

— 국제조사보고서와 함께 (조약 제 21 조(3))

(54) Title: METHOD FOR SCHEDULING IN MULTIPROCESSING ENVIRONMENT AND DEVICE THEREFOR

(54) 발명의 명칭 : 다중 프로세싱 환경에서의 스케줄링 방법 및 그 장치



410 ... Packet identification unit
 420 ... Packet allocation table
 430 ... Multi-scheduling unit
 440 ... Memory
 460, 462, 464, 466, 470, 472, 474 ... Processor
 480 ... Packet transfer unit
 AA ... Network

(57) Abstract: Disclosed are a network interface device for processing a packet, and a method therefor. The network interface device comprises multiple queues. Upon receiving a packet through a physical network, the network interface device identifies the flow of the packet, stores the packet to the multiple queues in units of flow, and then executes parallel processing of the packet through a multiprocessor.

(57) 요약서: 패킷을 처리하는 네트워크 인터페이스 장치 및 그 방법이 개시된다. 네트워크 인터페이스 장치는, 복수 개의 큐를 포함하고, 물리적 망을 통해 패킷을 수신하면, 패킷의 플로우를 식별하여 플로우 단위로 복수 개의 큐에 패킷을 저장한 후 다중 프로세서를 통해 병렬처리한다.

WO 2015/199366 A1

명세서

발명의 명칭: 다중 프로세싱 환경에서의 스케줄링 방법 및 그 장치 기술분야

- [1] 본 발명은 다중 스케줄링 방법 및 그 장치에 관한 것으로서, 보다 상세하게는 패킷을 병렬처리하는 다중 프로세싱 환경에서 스케줄링하는 방법 및 그 장치에 관한 것이다.

배경기술

- [2] 다중 프로세싱 시스템은 복수 개의 CPU(Central Processing Unit) 코어(core)를 이용하여 복수 개의 프로세스를 병렬처리한다. 그러나 다중 프로세싱의 경우, CPU 코어간에 부하를 균등하게 분포시키는 문제, CPU 코어들간 공유하는 자원의 충돌, 캐시 효율성 저하 등의 문제점이 있다.
- [3] 특히 패킷을 병렬처리하는 다중 프로세싱 시스템의 경우, 패킷 처리 효율을 높이기 위해서는 동일한 플로우에 속한 패킷을 동일한 CPU에서 처리하도록 하며 CPU에 대한 플로우 친화도(flow affinity)를 유지하는 것이 바람직하나, 이 경우 특정 CPU에 부하가 과중 되어 전체 CPU간에 부하 분포의 불균형이 발생할 수 있으며, 이로 인하여 다중 프로세싱 시스템의 전체 처리 효율이 낮아질 수 있는 문제점이 있다.
- [4] 이를 해결하기 위해서는 CPU 사이의 부하분산(load balancing)을 주기적으로 수행하여야 하나, 이 경우 플로우를 처리하는 CPU가 부하분산 과정에서 바뀌어 플로우 친화도가 낮아지며, 패킷의 처리 순서가 뒤바뀌어 패킷 순위 재처리(Packet re-ordering)의 과정이 필요한 이유 등으로 다중 프로세싱 시스템의 패킷 처리 효율이 낮아지는 문제점이 있다.
- [5] 이와 같이, 다중 프로세싱 시스템의 처리 효율을 높이기 위하여 플로우 친화도를 높이고 적절한 부하분산이 필요하나 양자는 서로 상충하므로 이를 적절히 보완할 필요가 있다.
- [6] 또한 최근 인터넷을 통하는 통신량이 급증하고, 이에 따라 서버의 대용량화, 고속화가 빠르게 진행되고 있다. 서버의 대용량화에 따르는 물리적 부피 증가를 해소하고 비용 절감 등을 꾀하기 위하여 서버의 가상화가 가속되고 있다. 서버의 대용량화, 고속화, 가상화에 따라 물리적 네트워크로부터 수신되는 가상화 환경에서 생성된 데이터 패킷을 포함하는 대용량 데이터에 대한 병렬처리와 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.
- [7] 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 높이는데 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.

발명의 상세한 설명

기술적 과제

- [8] 본 발명이 이루고자 하는 기술적 과제는, 패킷 병렬 처리를 수행하는 다중 프로세싱 환경에서 플로우 친화도(flow affinity)와 부하분산(load balancing)의 상충성을 완화시키고 전체 프로세서들의 사용 효율을 증가시키는 스케줄링 방법 및 그 장치를 제공하는 데 있다.

과제 해결 수단

- [9] 상기의 기술적 과제를 달성하기 위한, 본 발명에 따른 스케줄링 방법의 일 예는, 다중 프로세싱 장치에서의 스케줄링 방법에 있어서, 복수의 프로세서들 전체 또는 그 일부를 적어도 하나 이상의 프로세서 그룹으로 그룹핑하는 단계; 및 수신한 패킷의 플로우에 대해 기 지정된 프로세서 그룹 또는 프로세서가 존재하면, 상기 플로우를 기 지정된 프로세서 그룹 또는 프로세서에 할당하는 단계; 및 수신한 패킷의 플로우에 대해 기 지정된 프로세서 그룹 또는 프로세서가 존재하지 않으면, 상기 플로우를 위한 새로운 프로세서 그룹을 생성하여 할당하거나 어느 프로세서 그룹에도 속하지 않은 프로세서에 상기 플로우를 할당하는 단계;를 포함한다.
- [10] 상기의 기술적 과제를 달성하기 위한, 본 발명에 따른 스케줄링 방법의 일 예는, 다중 프로세싱 장치에서의 스케줄링 방법에 있어서, 복수의 프로세서들의 부하상태나 처리용량 또는 수신 패킷의 속성을 기초로 단일 스케줄링 또는 다중 스케줄링을 결정하는 단계; 상기 단일 스케줄링의 경우에, 복수의 프로세서들 중 하나를 스케줄러로 지정하는 단계; 및 상기 다중 스케줄링의 경우에, 복수의 프로세서들을 적어도 둘 이상의 프로세서 그룹으로 그룹핑하고, 각각의 프로세서 그룹 내 프로세서들 중 하나를 각각의 프로세서 그룹의 스케줄러로 지정하는 단계;를 포함한다.
- [11] 상기의 기술적 과제를 달성하기 위한, 본 발명에 따른 스케줄링 방법의 일 예는, 물리적 망을 통해 수신된 패킷으로부터 물리적 네트워크 프레임에 인캡슐레이션된, 가상화 환경 네트워크 계층 정보를 포함하는 덩패킷을 획득하는 단계; 상기 덩패킷에 포함된 상기 가상화 환경 네트워크 계층 정보를 기초로 상기 덩패킷을 덩플로우로 식별하는 단계; 및 상기 덩패킷을 상기 식별된

딥플로우 단위로 구분하여 해당하는 큐에 할당하는 단계;를 포함한다.

- [12] 상기의 기술적 과제를 달성하기 위한, 본 발명에 따른 네트워크 인터페이스 장치의 일 예는, 물리적 망을 통해 수신된 패킷으로부터 물리적 네트워크 프레임에 인캡슐레이션된, 가상화 환경 네트워크 계층 정보를 포함하는 딥패킷을 획득하는 패킷수신부; 상기 딥패킷에 포함된 상기 가상화 환경 네트워크 계층 정보를 기초로 상기 딥패킷을 딥플로우로 식별하는 패킷분석부; 및 상기 딥패킷을 상기 식별된 딥플로우 단위로 구분하여 해당하는 큐에 할당하는 스케줄러;를 포함한다.

발명의 효과

- [13] 본 발명에 따르면, 플로우 친화도와 부하 분산의 상충성을 완화하여 병렬 처리의 성능을 향상시킬 수 있다. 또한 동적으로 지정되는 복수 개의 스케줄러를 이용함으로써 패킷 스케줄링과 큐잉에 따른 대기시간(latency)을 감소할 수 있다. 또한 복수 개의 스케줄러를 통해 트래픽 속성에 따른 다양한 스케줄링 알고리즘을 용이하게 적용할 수 있다. 또한 복수의 가상 머신을 포함하는 가상화 환경을 구비한 서버의 부하가 감소 된다. 딥플로우 단위로 패킷을 처리함으로써 딥패킷과 프로세서 사이의 친화도를 높여 병렬 프로세싱의 효율을 높인다. 또한 가상 스위치의 부하를 네트워크 인터페이스 카드로 분산시켜 가상 네트워크 프로세싱의 효율을 높일 수 있다. 또한 딥플로우 단위로 큐잉하여 처리함으로써 가상 머신 종단간 딥플로우 단위의 QoS가 보장되는 스케일러블한 통신 처리를 구현할 수 있다.

도면의 간단한 설명

- [14] 도 1은 본 발명에 따른, 단일 스케줄링을 수행하는 다중 프로세싱 장치의 일 예를 도시한 도면,
 [15] 도 2는 본 발명에 따른, 단일 스케줄링을 수행하는 다중 프로세싱 방법의 일 예를 도시한 도면,
 [16] 도 3은 본 발명에 따른 다중 스케줄링을 위한 분류 정책의 일 예를 도시한 도면,
 [17] 도 4는 본 발명에 따른 다중 스케줄링 방법이 적용된 다중 프로세싱 장치의 일 예의 구성을 도시한 도면,
 [18] 도 5는 본 발명에 따른 다중스케줄링부의 상세 구성의 일 예를 도시한 도면,
 [19] 도 6은 본 발명에 따른 다중 프로세싱 환경에서의 다중 스케줄링 방법의 일 예를 도시한 흐름도,
 [20] 도 7은 본 발명에 따른 다중 프로세싱 환경에서의 다중 스케줄링 방법의 다른 일 예를 도시한 흐름도,
 [21] 도 8은 본 발명에 따른 다중 프로세싱 환경에서 프로세서 그룹 내 스케줄러로 지정된 프로세서의 스케줄링 방법의 일 예를 도시한 흐름도,
 [22] 도 9는 본 발명에 따른 다중 스케줄링을 위한 프로세서 그룹핑의 일 예를 도시한 도면,

- [23] 도 10은 본 발명에 따른 다중 스케줄링을 위하여 프로세서를 동적으로 그룹핑하는 방법의 일 예를 도시한 도면,
 [24] 도 11은 본 발명에 따른 다중 프로세싱을 위한 네트워크 인터페이스 장치를 포함한 시스템의 개략적인 구조의 일 예를 도시한 도면,
 [25] 도 12는 본 발명에 따른 NIC의 자원을 동적으로 설정하는 방법의 일 예를 도시한 도면,
 [26] 도 13은 본 발명에 따른 NIC의 일 실시예의 구성을 도시한 도면,
 [27] 도 14는 본 발명에 따른 NIC의 덱플로우 기반 큐 할당의 일 예를 도시한 도면,
 [28] 도 15는 본 발명에 따른 NIC의 덱플로우 기반 큐 할당의 다른 일 예를 도시한 도면,
 [29] 도 16은 본 발명에 사용되는 덱패킷의 일 예를 도시한 도면, 그리고,
 [30] 도 17은 본 발명에 따른 가상 머신 환경을 위한 패킷 처리 방법의 일 예를 도시한 흐름도이다.

발명의 실시를 위한 형태

- [31] 이하에서, 첨부된 도면들을 참조하여 본 발명에 따른 다중 프로세싱 환경에서의 다중 스케줄링 방법 및 그 장치에 대해 상세히 살펴본다.
 [32] 도 1은 본 발명에 따른, 단일 스케줄링을 수행하는 다중 프로세싱 장치의 일 예를 도시한 도면이다.
 [33] 도 1을 참조하면, 다중 프로세서 장치(100)는 패킷 식별부(105), 패킷전달부(140), 패킷할당테이블(130), 메모리(104), 복수 개의 큐(110,112,114), 복수 개의 프로세서(120,122,124) 및 제어부(150)를 포함한다.
 [34] 패킷식별부(105)는 유무선 네트워크 또는 다른 기기로부터 패킷을 수신하고, 수신한 패킷의 플로우(flow)를 식별한다. 또한 패킷식별부(105)는 수신한 패킷 플로우에 할당된 프로세서가 존재하는지 패킷할당테이블(130)을 참조하여 파악한다.
 [35] 패킷할당테이블(130)은 패킷 플로우별로 각각 할당된 프로세서의 정보를 포함한다. 예를 들어, 패킷할당테이블(130)은 제1플로우와 제2플로우를 처리하기 위한 프로세서로 제1프로세서가 할당되어 있고, 제3플로우를 처리하기 위한 프로세서로 제2프로세서가 할당되어 있다는 정보를 포함한다. 패킷할당테이블(130)에 저장되는 정보는 이후 살필 스케줄러에 의해 생성되고 갱신된다.
 [36] 메모리(104)는 패킷식별부(105)가 수신한 패킷을 저장한다. 이때 메모리(104)는 패킷식별부(105)에 의해 파악된 패킷의 플로우 정보, 패킷할당테이블(130)을 참조하여 파악한 프로세서 정보 등을 함께 저장할 수 있다.
 [37] 패킷전달부(140)는 메모리(104)에 저장된 패킷들을 해당 프로세서의 큐에 전달한다. 패킷전달부(140)는 메모리에 저장된 패킷을 순서대로 프로세서 큐에 전달하거나, QoS(Quality of Service), 우선순위(priority) 등 다양한 조건 등을

- 고려하여 비순차적으로 패킷을 프로세서의 큐에 전달할 수도 있다.
- [38] 큐(110,112,114)는 각 프로세서가 처리할 패킷을 메모리(104)로부터 전달받아 저장한다. 본 실시 예에서 큐(110,112,114)는 프로세서(120,122,124)마다 각각 하나씩 존재하는 것으로 도시하였으나, 반드시 이에 한정되는 것은 아니며 하나의 프로세서에 두 개 이상의 큐가 존재하거나, 또는 하나의 큐를 두 개 이상의 프로세서가 공유할 수 있다. 또는 큐(110,112,114)는 도 11 내지 도 17에 개시된 방법을 통해 그룹핑될 수 있다.
- [39] 또한 큐(110,112,114)는 선입선출(FIFO) 구조이나, 반드시 이에 한정되는 것은 아니며, 후입선출(LIFO) 또는 우선순위(priority) 기반 출력 등 다양한 형태의 구조로 구현될 수 있는 등 각 프로세서에서 처리할 패킷을 저장할 수 있는 형태이면 된다.
- [40] 패킷할당테이블(130)에 패킷 플로우 정보가 존재하지 아니한 경우에, 제어부(150)는 복수 개의 프로세서들 중 하나를 스케줄러로 지정한 후 인터럽트 요청 신호를 전송한다. 인터럽트 요청 신호를 수신한 프로세서는 패킷을 처리할 프로세서를 선택하고 패킷할당테이블에 관련 정보를 저장한다.
- [41] 복수 개의 프로세서(120,122,124)는 각각 패킷을 처리한다. 또한 다중 프로세싱 시스템의 효율과 제조비용의 절감 등을 고려하여 패킷 스케줄링을 위한 별도의 스케줄러를 구비하지 아니하고, 복수 개의 프로세서 중 하나(예를 들어, 프로세서1(120))를 스케줄러로 활용할 수 있다. 복수 개의 프로세서 중 하나를 스케줄러로 활용하는 방법에 대해서는 도 2를 참조하여 살펴본다. 물론 본 실시 예에서 복수 개의 프로세서 외에 별도의 스케줄러를 구비하여 구현할 수도 있다.
- [42] 도 2는 본 발명에 따른, 단일 스케줄링을 수행하는 다중 프로세싱 방법의 일 예를 도시한 도면이다.
- [43] 도 1 및 도 2를 함께 참조하면, 패킷식별부(105)는 수신한 패킷을 분석하여 패킷 플로우를 식별한다(S200, S210). 여기서 플로우 식별 방법은 수신한 패킷의 네트워크 전 계층의 트래픽 속성을 분석하여 기 설정된 네트워크 통신 정책에 따라 분류한다. 일 예로, 수신한 패킷의 송신 노드 주소, 목적지 주소, 세션 및 응용 계층 등의 속성을 요소로 기 설정된 통신 정책에 의하여 하나의 플로우를 구분할 수 있다. 패킷식별부(105)는 패킷할당테이블(130)을 참조하여, 플로우를 처리할 프로세서의 정보가 존재하는지 파악한다(S220). 패킷과 해당 패킷의 플로우 정보, 프로세서 정보 등은 메모리(104)에 저장된다.
- [44] 패킷할당테이블(130)에 플로우를 처리할 프로세서의 정보가 존재하는 경우(S230), 패킷전달부(140)는 패킷을 해당 프로세서의 큐에 전달한다(S260). 예를 들어, 패킷식별부(105)에 의해 수신 패킷이 제1플로우로 식별되고, 패킷할당테이블(130)에 제1플로우를 처리할 프로세서로 제2프로세서가 할당되어 있다면, 패킷전달부(140)는 패킷을 제2프로세서(122)의 큐(112)로 전달한다.
- [45] 반면, 패킷할당테이블(130)에 플로우를 처리할 프로세서의 정보가 존재하지

아니하면(즉, 새로운 플로우인 경우)(S230), 제어부(150)는 복수 개의 프로세서 중 스케줄러로 지정된 프로세서에게 인터럽트 요청신호를 전달한다(S240). 제어부(150)는 복수 개의 프로세서들 중 현재 부하가 가장 적은 프로세서를 스케줄러로 지정하거나, 기 설정된 스케줄러 결정 알고리즘을 통하여 스케줄러를 지정하거나, 기 설정된 프로세서를 스케줄러로 지정할 수 있다. 본 실시 예의 경우 프로세서1(120)이 스케줄러로 지정되어 있다.

- [46] 인터럽트 요청신호를 수신한 프로세서(120)는 이전 수행하던 작업을 중단하고, 스케줄링 동작을 수행한다(S250). 예를 들어, 스케줄러로 지정된 프로세서(120)는 새로운 플로우를 처리할 프로세서를 선택하고(S250), 선택한 프로세서에 대한 정보를 패킷할당데이터블(130)에 저장한다(S260). 스케줄러로 지정된 프로세서에 인터럽트 요청이 전달되어 있는 경우, 그 인터럽트가 해제될 때까지 새로 입력되는 패킷 처리를 위한 인터럽트 요청은 불허된다.
- [47] 또한, 스케줄러로 지정된 프로세서(120)는 프로세서들의 부하 불균형이 일정 수준 이상인 경우 등의 특정 이벤트가 발생하는 경우 또는 주기적으로 종래의 여러 부하분산(load balancing) 알고리즘을 적용하여 각 프로세서간의 부하재분산(re-balancing)을 시도할 수 있다.
- [48] 도 1,2의 경우, 하나의 스케줄러(120)가 선택되어 해당 작업을 수행하는 동안 시스템으로부터 새로운 인터럽트가 허용되지 않으며, 요청되어 있는 인터럽트가 해제될 때까지 다른 새로운 플로우에 대한 처리가 지연된다. 또한 부하 불균형을 해소하기 위한 부하재분산을 복수 개의 프로세서들 전체에 대해 수행하므로, 플로우 친화도와 부하분포간의 상충성이 더 심화되는 문제점이 있다. 이는 도 4의 다중 스케줄링을 통해 완화될 수 있다.
- [49] 도 3은 본 발명에 따른 다중 스케줄링을 위한 분류 정책의 일 예를 도시한 도면이다.
- [50] 도 3을 참조하면, 분류 정책은 복수 개의 프로세서들을 그룹으로 분할하기 위한 정책을 포함한다. 도 4에서 살펴보겠지만, 다중 스케줄링의 경우 복수 개의 프로세서들을 적어도 둘 이상의 그룹으로 나누어 각 그룹별로 스케줄링을 수행한다. 이를 위해서는 복수 개의 프로세서들을 그룹으로 나누기 위한 정책이 필요하다.
- [51] 분류 정책의 일 예로, 도 3에 도시된 바와 같이 패킷 플로우 기반의 정책이 있다. 패킷 플로우를 계층적으로 나눌 수 있는 속성을 기반으로 플로우를 두 그룹 A,B로 분할할 수 있다. 이 경우, 복수 개의 프로세서들은 현재 처리하고 있는 플로우가 어떤 그룹에 속하는지에 따라 두 그룹으로 분할될 수 있다.
- [52] 또 다른 예로, 각 프로세서들의 부하를 기반으로 하는 분류 정책이 있다. 각 그룹별 부하 분포가 고르게 될 수 있도록 프로세서들을 기 설정된 그룹 수에 따라 분할할 수 있다.
- [53] 복수 개의 프로세서들을 그룹으로 분할하기 위한 분류 정책은 복수일 수 있다. 예를 들어, 제1 정책은 복수 개의 프로세서들을 플로우 기반으로 두 개의

그룹으로 분할하는 정책이고, 제2 정책은 복수 개의 프로세서들을 플로우 기반으로 세 개의 그룹으로 분할하는 정책이고, 제3 정책은 복수 개의 프로세서들을 부하정도에 따라 적어도 둘 이상의 그룹으로 분할하는 정책일 수 있다.

- [54] 본 발명은 도 3의 실시 예에 한정되는 것은 아니며, 이 외 각 프로세서들을 분할하기 위한 다양한 분류 정책을 적용할 수 있다. 분류 정책은 미리 설정되어 있으며, 사용자가 별도의 입출력 인터페이스를 통해 분류 정책을 갱신할 수 있다. 도 4에서는 다중 스케줄링을 위하여 복수 개의 프로세서들을 나누는 기준, 즉 분류 정책이 기 설정되어 있다고 가정한다.
- [55] 도 4는 본 발명에 따른 다중 스케줄링 방법이 적용된 다중 프로세싱 장치의 일 예의 구성을 도시한 도면이다.
- [56] 도 4를 참조하면, 다중 프로세싱 장치(400)는 패킷식별부(410), 패킷전달부(480), 패킷할당테이블(420), 다중스케줄링부(430), 메모리(440), 복수 개의 큐(450) 및 복수 개의 프로세서(460,462,464,466,470,472,474)를 포함한다.
- [57] 패킷식별부(410), 패킷전달부(480), 패킷할당테이블(420), 메모리(440), 복수 개의 큐(450)와 프로세서(460,462,464,466,470,472,474) 등은 도 1에서 설명한 구성과 기능을 모두 포함한다. 따라서 본 실시 예는 도 1과 동일한 구성과 기능에 대한 중복 설명은 생략하고, 본 실시 예에 따른 다중스케줄링을 위하여 필요한 구성과 기능 위주로 설명한다.
- [58] 다중스케줄링부(430)는 부하분포 상태, 트래픽 속성, 트래픽 처리 용량 등 다중 프로세싱 시스템의 상태 정보를 기초로 단일 스케줄링을 수행할지 다중 스케줄링을 수행할지를 결정한다. 예를 들어, 다중스케줄링부(430)는 단일 스케줄링을 수행하다가 다중 스케줄링으로 변경하거나, 그 반대로 다중 스케줄링을 수행하다가 단일 스케줄링으로 변경할 수 있다.
- [59] 다중스케줄링부(430)는 도 3과 같은 분류 정책이 다수 개 존재하는 경우, 상태 정보를 기초로 어떤 분류 정책을 적용할지 결정할 수 있다. 다중스케줄링부(430)는 다중 스케줄링을 수행할 것으로 결정하면, 분류 정책에 따라 복수 개의 프로세서들을 적어도 둘 이상의 그룹으로 분류하고, 각 그룹별로 스케줄링을 수행할 스케줄러를 지정한다. 다중스케줄링부의 상세 구성은 도 5에 도시되어 있다.
- [60] 예를 들어, 도 4의 예와 같이, 다중스케줄링부(430)는 7개의 프로세서들을 두 개의 프로세서 그룹(제1그룹:프로세서1~4, 제2그룹:프로세서5~7)으로 분할하고, 각 그룹 내 하나의 프로세서(466,474)를 기 설정된 스케줄러 결정 알고리즘에 의하여 스케줄러로 지정한다. 다중스케줄링부(430)에서 프로세서들을 그룹별로 분할하게 되면 패킷할당테이블(420)에 프로세서 그룹에 대한 정보를 저장할 수 있다.
- [61] 예를 들어, 도 3과 같이 플로우 기반으로 프로세서들을 그룹핑하는 경우에, 다중스케줄링부(430)는 플로우가 어느 그룹에 속하는지에 대한 정보를

패킷할당테이블(420)에 저장한다. 새로 수신한 패킷 플로우에 대한 정보가 패킷할당테이블(420)에 존재하지 아니한 경우, 패킷식별부(410)는 새로운 플로우가 어떤 그룹에 속하는지 파악하고 해당 패킷과 해당 패킷에 대한 식별정보를 메모리(440)에 저장한다. 다중스케줄링부(430)는 부하 정도 또는 기 설정된 스케줄러 결정 알고리즘에 의하여 해당 그룹 내에 해당 패킷을 처리할 스케줄러를 지정하고, 지정된 스케줄러에게 해당 패킷을 처리하도록 인터럽트 요청신호를 전송한다. 스케줄러로 지정된 프로세서는, 도 1에서 설명한 바와 같이, 해당 플로우를 처리할 그룹 내 프로세서를 선택하는 등의 스케줄링 동작을 수행한다.

- [62] 예를 들어, 도 3의 그룹핑 정책에 의해 플로우 그룹 A(300)에 대해 프로세서1~4를 포함하는 프로세서 제1그룹(490)이 할당되어 있고, 플로우 그룹 B(310)에 대해 프로세서5~7을 포함하는 프로세서 제2그룹(495)이 할당되어 있다고 하자. 수신한 패킷의 플로우가 플로우 그룹 A(300)에 속하면, 다중스케줄링부(430)는 프로세서 제1그룹(490)에 속한 프로세서들 중 하나를 스케줄러로 지정하여 해당 그룹(490) 내 스케줄링 동작이 수행될 수 있도록 한다. 이때 프로세서 제2그룹(495)은 제1그룹(490)의 스케줄링 동작 여부와 무관하게 새로운 패킷의 처리 또는 스케줄링 동작을 수행할 수 있어 전체 프로세서의 처리 효율이 향상된다. 다시 말해, 프로세서 제1그룹(490)과 프로세서 제2그룹(495)의 스케줄링 동작이 병렬로 수행될 수 있다.
- [63] 또 다른 예로, 다중 스케줄링에 있어서, 프로세서들의 그룹핑은 프로세서의 부하나 여러 정책 등에 따라 수시로 변경될 수 있으며, 그룹핑되지 아니한 프로세서들을 새로운 수신 패킷의 플로우를 위한 프로세서 그룹으로 새롭게 그룹핑할 수 있다. 이에 대해서는 도 9 및 도 10에서 다시 설명한다.
- [64] 도 5는 본 발명에 따른 다중스케줄링부의 상세 구성의 일 예를 도시한 도면이다.
- [65] 도 5를 참조하면, 다중스케줄링부(430)는 정책판단부(500), 그룹분할부(510), 스케줄러지정부(520)를 포함한다.
- [66] 정책판단부(500)는 다중 프로세싱 환경의 여러 상태 정보, 예를 들어, 부하분포상태, 트래픽 속성, 트래픽 처리 용량 등의 정보를 이용하여 단일 스케줄링 또는 다중 스케줄링을 결정한다. 또한 정책판단부(500)는 다중 스케줄링을 수행하는 경우 복수 개의 프로세서들을 어떻게 분할할지, 또는 분할된 그룹별 어떤 정책을 적용할지를 결정한다.
- [67] 예를 들어, 정책 판단부(500)는 다중 프로세싱 장치의 전체 트래픽 처리 용량이 일정 수준 이하인 경우에 다중 스케줄링을 수행할 것으로 결정하고, 분류 정책으로도 3과 같은 플로우 기반의 정책을 선택할 수 있다.
- [68] 그룹분할부(510)는 정책판단부(500)에 의해 결정된 분류 정책에 따라 복수 개의 프로세서들을 적어도 둘 이상의 그룹으로 분할한다.
- [69] 스케줄러지정부(520)는 부하 정도 또는 기 설정된 스케줄러 선택 알고리즘에

의하여 각 그룹 내 프로세서들 중 하나를 스케줄러로 지정한다. 예를 들어, 스케줄러지정부(520)는 각 그룹별로 부하가 가장 적은 프로세서를 스케줄러로 지정할 수 있다. 이외 특정 프로세서를 스케줄러로 고정하거나 다른 여러 선택 방법을 적용하여 동적으로 스케줄러를 지정할 수 있다.

[70] 도 6은 본 발명에 따른 다중 프로세싱 환경에서의 다중 스케줄링 방법의 일 예를 도시한 흐름도이다.

[71] 도 4 및 도 6을 함께 참조하면, 다중스케줄링부(430)는 트래픽 용량이나, 플로우 속성, 부하 분포 상태 등 여러 상태 정보를 파악한다(S600). 이러한 상태 정보를 기초로 다중스케줄링부(430)는 단일 스케줄링을 수행할지 아니면 다중스케줄링을 수행할지 여부를 판단한다(S610). 단일 스케줄링을 수행하는 경우, 다중 스케줄링부(430)는 복수 개의 프로세서들 중 하나를 스케줄러로 지정한다. 다중 스케줄링을 수행하는 경우, 다중스케줄링부(430)는 분류 정책에 따라 복수 개의 프로세서들을 적어도 둘 이상의 그룹으로 분할하고(S620), 각 그룹별로 스케줄러로 동작할 프로세서를 지정한다(S630). 각 그룹별로 지정된 프로세서는 도 1과 같이, 인터럽트 요청 신호에 따라 해당 패킷에 대한 스케줄링 작업을 하되, 전체 프로세서들을 대상으로 스케줄링을 수행하는 것이 아니라 자신이 속한 그룹 내 프로세서들에 대해서만 스케줄링을 수행한다. 따라서, 둘 이상으로 분할된 각 그룹은 그룹별로 지정된 스케줄러에 의하여 독립적으로 동시에 스케줄링 작업이 수행 가능하다. 다시 말해, 각 그룹별 스케줄러는 다중스케줄링부(430)로부터 각각 인터럽트 신호를 수신하며, 다른 그룹의 스케줄러에 대한 인터럽트의 해제 여부와 관계없이 패킷 처리를 위한 프로세서의 선택 등과 같은 스케줄링 작업을 수행할 수 있다. 그리고 분할된 각 그룹은 필요에 따라 독립적으로 서로 다른 정책 또는 알고리즘이 적용될 수 있다.

[72] 도 7은 본 발명에 따른 다중 프로세싱 환경에서의 다중 스케줄링 방법의 다른 일 예를 도시한 흐름도이다. 도 7은 앞서 살핀 다중스케줄링부에 의해 복수 개의 프로세서들이 그룹핑되어 있는 경우를 가정한다.

[73] 도 4 및 도 7을 참조하면, 패킷식별부(410)는 패킷을 수신하면(S700), 패킷을 분석하여 플로우를 식별한다(S710). 패킷할당테이블(420)에 해당 플로우에 대한 정보가 존재하는 경우에(S730), 패킷식별부(410)는 새로운 플로우가 어떤 그룹에 속하는지 등을 파악하고 해당 패킷과 해당 패킷에 대한 식별정보를 메모리(440)에 저장한다. 다중스케줄링부(430)는 해당 스케줄러에게 해당 패킷을 처리하도록 인터럽트 요청신호를 전송한다.

[74] 패킷할당테이블(420)에 해당 플로우에 대한 정보가 존재하지 아니한 경우에(S730), 다중스케줄링부(430)는 분류 정책을 참조하여 플로우가 속한 그룹을 파악한다(S740). 예를 들어, 프로세서 그룹이 도 3과 같은 플로우 기반으로 나뉘어지는 경우, 다중스케줄링부(430)는 플로우를 계층적으로 구분하는 상위 속성을 기준으로 새롭게 인식한 플로우가 어느 그룹에 속하는지

파악한다. 또 다른 예로서, 프로세서 그룹이 부하 분포에 따라 나뉘어지는 경우, 다중스케줄링부(430)는 부하가 상대적으로 낮은 그룹을 새롭게 인식한 플로우가 속하는 그룹으로 선택할 수 있다.

- [75] 다중스케줄링부(430)는 새로운 플로우가 어떤 그룹에 속하는지 파악한 후, 부하 정도 또는 기 설정된 스케줄러 결정 알고리즘을 이용하여 해당 그룹 내에 해당 패킷을 처리할 스케줄러를 지정하고, 지정된 스케줄러에게 해당 패킷을 처리하도록 인터럽트 요청신호를 전송한다.
- [76] 인터럽트 신호를 수신한 프로세서는 스케줄러로 동작하며, 새로운 플로우를 처리할 프로세서를 선택하고, 관련 정보를 패킷할당테이블(420)에 저장한다(S750, S760).
- [77] 예를 들어, 도 4를 다시 참조하면, 새로운 플로우가 프로세서1~4로 구성된 그룹에 할당되는 경우, 다중스케줄링부(430)는 스케줄러로 지정된 프로세서4(466)에게 인터럽트 신호를 전송한다. 프로세서4(466)는 기 지정된 프로세서 결정 알고리즘에 의하여 해당 패킷을 처리할 프로세서로 프로세서1(460)을 선택하고 해당 정보를 패킷할당테이블(420)에 저장한다. 이후 동일 플로우의 패킷이 수신된 경우에 패킷전달부(480)는 해당 패킷을 프로세서1(460)에 할당한다.
- [78] 도 8은 본 발명에 따른 다중 프로세싱 환경에서 프로세서 그룹 내 스케줄러로 지정된 프로세서의 스케줄링 방법의 일 예를 도시한 흐름도이다.
- [79] 도 8을 참조하면, 복수의 프로세서는 인터럽트 신호를 수신하기 이전에는 일반적인 플로우 처리 과정 등을 수행한다(S800). 새롭게 수신한 패킷의 플로우에 할당된 프로세서 그룹 또는 프로세서가 존재하면, 다중스케줄링부(430)는 기 지정된 프로세서 그룹 또는 프로세서에 플로우를 할당한다. 새롭게 수신한 패킷의 플로우에 할당된 프로세서 그룹 또는 프로세서가 존재하지 아니하면, 다중스케줄링부(430)는 새로운 패킷의 플로우를 위한 프로세서 그룹을 생성한 후 새롭게 생성한 프로세서 그룹 내 어느 하나의 프로세서에게 인터럽트 신호를 전송한다(S810).
- [80] 인터럽트 신호를 수신한 프로세서는 이전에 수행하던 동작을 멈추고(S820), 새로운 패킷 플로우를 어떤 프로세서에게 할당할지를 결정하는 스케줄링 동작을 수행한 후(S830), 인터럽트 신호 수신 전에 수행하던 동작을 다시 재개한다(S840). 다중스케줄링부(430)는 인터럽트 신호를 프로세서 그룹별로 각각 전송할 수 있으며, 따라서 각 프로세서 그룹은 동시에 각각 스케줄링 동작을 수행할 수 있다.
- [81] 도 9는 본 발명에 따른 다중 스케줄링을 위한 프로세서 그룹핑의 일 예를 도시한 도면이다.
- [82] 도 9를 참조하면, 다중스케줄링부(430)는 단일 스케줄링의 경우에는 복수의 프로세서들(900) 중 어느 하나를 스케줄러로 지정하며, 다중 스케줄링의 경우에는 프로세서들 전체 또는 그 일부를 적어도 하나 이상의

- 그룹(910,920)으로 그룹핑하고 각 그룹별 스케줄러를 지정한다.
- [83] 다중스케줄링부(430)는 프로세서들의 부하 상태나 기 설정된 여러 가지 정책에 따라 새로운 프로세서 그룹을 생성하거나 기존 프로세서 그룹을 갱신할 수 있다.
- [84] 예를 들어, 다중 스케줄링의 경우, 다중스케줄링부(430)는 복수의 프로세서들을 프로세서1~3의 제1그룹(910), 프로세서4~5의 제2그룹(920)의 두 그룹으로 그룹핑되고 나머지 프로세서들을 그룹핑하지 아니할 수 있다.
- [85] 다중 스케줄링 동작 중에, 새로운 프로세서 그룹의 생성이 필요한 경우에, 다중스케줄링부(430)는 현재 그룹핑되지 아니한 프로세서들 전체 또는 그 일부를 그룹핑하여 새로운 프로세서 그룹을 생성하거나 기존 프로세서 그룹들을 변경하여 새로운 프로세서 그룹을 생성할 수 있다.
- [86] 또 다른 예로, 다중스케줄링부(430)는 다중 스케줄링 동작 중에 이미 그룹핑되어 있는 그룹, 예를 들어, 프로세서1~3의 제1그룹(910)과 프로세서4~5의 제2그룹(920)에 새로운 프로세서를 추가하거나 그룹 내 프로세서 중 일부를 그룹에서 제외하는 등의 갱신 동작을 수행할 수 있다.
- [87] 도 10은 본 발명에 따른 다중 스케줄링을 위하여 프로세서를 동적으로 그룹핑하는 방법의 일 예를 도시한 도면이다.
- [88] 도 10을 참조하면, 다중스케줄링부(430)는 수신한 패킷의 플로우를 위한 새로운 프로세서 그룹이 필요하거나, 이전 프로세서 그룹의 변경이 필요하다면, 프로세서 그룹을 새롭게 생성하거나 기존 프로세서 그룹을 갱신한다(S1000). 예를 들어, 새롭게 수신한 패킷의 플로우에 할당된 프로세서 그룹이나 프로세서가 존재하지 아니한 경우에, 다중스케줄링부(430)는 프로세서들 중 그룹핑되지 아니한 프로세서들 전체나 그 일부를 새로운 프로세서 그룹으로 생성한다. 또 다른 예로, 프로세서 그룹 내 부하 정도 또는 전체 프로세서 부하 정도가 일정 수준에 이르거나 기 설정된 정책에 따라, 다중스케줄링부(430)는 프로세서 그룹 전체를 다시 새롭게 구성하거나, 특정 프로세서 그룹에 새로운 프로세서를 추가하거나 그룹 내 기존 프로세서 중 적어도 하나를 그룹에서 제외할 수 있다.
- [89] 다중스케줄링부(430)는 생성 또는 갱신된 프로세서 그룹 내 어느 하나의 프로세서를 스케줄러로 지정한다(S1010). 각 그룹 내 어느 하나의 프로세서는 항상 스케줄러로 동작하는 것이 아니라 도 8에서 설명한 바와 같이 인터럽트 신호를 수신한 경우에만 스케줄러로 동작하고, 스케줄링 동작이 완료되면 다른 일반적인 프로세서와 동일한 동작을 수행한다.
- [90] 지금까지는 주로 복수 개의 프로세서들을 그룹핑하여 스케줄링하는 방법에 대해 살펴보았으나, 이하에서는 주로 복수 개의 프로세서와 연결된 큐들을 그룹핑하여 스케줄링하는 방법에 대해 살펴본다. 도 1~10의 기술적 구성이 도 11~17의 기술적 구성에 부가되거나 반대로 도 11~17의 구성이 도 1~10의 구성에 부가될 수 있다. 다시 말해, 도 1~10에서 설명한 복수 개의 프로세서의 그룹핑과 도 11~17에서 설명할 복수 개의 큐의 그룹핑이 동시에 수행될 수도 있다. 다중

- 프로세싱 장치는 네트워크 인터페이스 장치로 구현될 수 있다.
- [91] 도 11은 본 발명에 따른 다중 프로세싱을 위한 네트워크 인터페이스 장치를 포함한 시스템의 개략적인 구조의 일 예를 도시한 도면이다.
- [92] 도 11을 참조하면, 네트워크 인터페이스 장치는 네트워크 인터페이스 카드(Network Interface Card, NIC)(1100)로 구현된다. 다만 네트워크 인터페이스 장치는 네트워크 인터페이스 카드(1100)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.
- [93] 서버(1120)는 복수 개의 가상 머신(1150,1152,1154)과, 가상 스위치(1140) 및 연결슬롯(1130)을 포함한다. 가상 스위치(1140)는 NIC(1100)를 통해 수신한 패킷을 목적지 가상 머신으로 전달한다. 연결슬롯(1130)은 NIC(1100)와 서버(1120)를 연결하는 인터페이스이며, 일 예로 PCIe(Peripheral Component Interconnect Express)로 구현될 수 있다. 이 경우, NIC(1100)는 PCIe 슬롯에 탈부착될 수 있다.
- [94] NIC(1100)는 네트워크(1110)로부터 수신한 패킷의 상위 계층의 트래픽 특성을 분석하여 플로우를 식별하고, 식별한 플로우를 다중 프로세서를 통해 병렬 처리한다. 여기서 패킷은 복수의 가상 머신에 전달될 수 있도록 종래의 다양한 터널링과 같은 기법 등을 이용하여 가상화 환경 네트워크 계층(Layer) 정보를 인캡슐레이션(encapsulation)한 패킷을 의미한다. 가상화 환경 네트워크 계층이란 가상머신으로 형성된 네트워크 계층을 의미하며, 가상화 환경 네트워크 계층 정보는 가상머신으로 형성된 네트워크 계층 내 패킷 전송을 위하여 물리적 네트워크 프레임에 인캡슐레이션된, 가상머신으로 형성된 네트워크 계층 정보를 의미한다. 이하에서는 본 실시 예에서 사용되는 가상화 환경 네트워크 계층 정보에 기초하여 식별된 패킷을 딥패킷(deep packet)이라고 한다. 딥패킷은 물리적 네트워크에서 일반 통신 프로토콜에 의하여 인식되어 원활한 전송이 이루어질 수 있도록 물리적 네트워크 프레임에 인캡슐레이션된다. 또한 딥패킷의 가상화 환경 네트워크 계층정보를 이용하여 분류한 플로우를 딥플로우(deep flow)라고 명칭한다. 딥플로우는 통신 서비스 구조 상 가상머신에서 생성된 서비스 종단의 플로우로 설명된다.
- [95] 딥플로우는 딥패킷의 물리적 네트워크 프레임을 제거한 가상화 환경 네트워크 프레임에서의 상위 계층(vL3 이상) 트래픽 속성에 따라 분류된 가상화 환경 네트워크에서의 특정 트래픽으로 정의할 수 있다. 딥플로우는 기 설정된 여러 정책에 따라 분류되고 식별될 수 있다. 예를 들어, 패킷분석부(1310)는 가상 머신의 TCP 플로우를 딥플로우로 식별할 수 있다. 딥패킷의 구조에 대해서는 도 16을 참조하여 설명한다.
- [96] NIC(1100)는 수신한 딥패킷을 병렬 처리하기 위한 복수 개의 큐와 복수 개의 프로세서들을 포함하며, 큐의 크기와 개수 등은 고정되거나, 딥플로우에 관한 정보, 서버의 가상화 환경 정보, 프로세서들의 부하 등에 따라 동적으로 변경될

수 있다.

[97] 도 12는 본 발명에 따른 NIC의 자원을 동적으로 설정하는 방법의 일 예를 도시한 도면이다.

[98] 도 11 및 도 12를 함께 참조하면, NIC(1100)가 서버(1120)의 연결슬롯(1130)에 부착되어 서버(1120)와 연결되는 경우(S1200), NIC(1100)는 서버(1120)로부터 가상 머신의 수를 포함한 가상화 환경 정보를 수신한다(S1210). NIC(1100)는 디플로우에 관한 정보, 수신한 가상화 환경 정보 또는 프로세서의 부하분포 등에 따라 큐의 크기와 개수, 큐 그룹의 생성 등 자원을 동적으로 설정한다(S1220).

[99] 예를 들어, NIC(1100)가 서버(1120)로부터 가상 머신이 4개라는 가상화 환경 정보를 수신하는 경우, NIC(1100)는 12개의 큐를 각 가상 머신별로 3개씩 할당할 수 있다. 또 다른 예로, NIC(1100)는 디플로우에 관한 정보를 기초로 디플로우를 두 개의 그룹으로 분할하고 각 그룹에 6개의 큐를 할당할 수 있다. 가상 머신별 할당되는 큐의 개수나 각 큐의 크기 등은 기 설정된 규칙에 따라 다양하게 설정될 수 있다.

[100] 도 13은 본 발명에 따른 NIC의 일 실시예의 구성을 도시한 도면이다.

[101] 도 13을 참조하면, NIC(1100)는 패킷수신부(1300), 패킷분석부(1310), 메모리(1320), 복수 개의 큐(1330), 복수 개의 프로세서(1340), 스케줄러(1350), 모니터링부(1360) 및 큐 관리부(1370)를 포함한다. 패킷수신부(1300) 등을 포함한 각 구성요소 사이의 연결선은 본 발명의 이해를 돕기 위한 하나의 예일 뿐이며, 큐 관리부(1370)와 모니터링부(1360) 사이의 연결, 스케줄러(1350)와 복수 개의 큐(1330) 사이의 연결 등 다양한 연결 관계가 설정될 수 있음은 물론이다.

[102] 패킷수신부(1300)는 외부 네트워크에서 일반적인 이더넷 프레임으로 인식되도록 종래의 다양한 터널링과 같은 방법 등에 의해 디캡시팅 인캡슐레이션된 패킷을 수신하면 이를 디캡슐레이션하여 물리적 네트워크에 해당하는 헤더 부분을 제거하고 가상화 환경에서의 데이터 패킷 프레임을 복원한다.

[103] 패킷분석부(1310)는 복원된 디캡시팅의 디플로우를 식별한다. 디플로우를 식별하기 위해서는 가상화 환경에서의 데이터링크계층(vL2 레이어)만이 아니라 네트워크계층(vL3 레이어) 이상의 상위 계층까지 해석되어야 한다. 이를 위해, 패킷분석부(1310)는 DPI(Deep Packet Insepection) 과정을 통해 디캡슐레이션된 디캡시팅의 가상 데이터링크계층(vL2 레이어)부터 가상 애플리케이션계층(vL7 레이어)까지 분석하여, 디플로우를 식별한다. 디플로우 식별을 위한 디캡시팅에 대한 분석은 가상 데이터링크계층부터 가상 애플리케이션계층 모두를 분석하는 것으로 한정되는 것은 아니며, 디플로우 식별 정책에 따라 분석의 범위는 달라질 수 있다.

[104] 메모리(1320)는 디캡시팅과 패킷분석부(1310)에 의해 식별된 디플로우 정보 등을 저장하고, 또한 디플로우와 큐 사이의 맵핑관계를 나타내는 플로우 테이블을

저장하고 관리한다.

- [105] 일 실시 예로, 패킷수신부(1300)은 디캡슐레이션된 덤패킷을 메모리(1320)에 저장하고, 덤패킷의 저장 사실을 패킷분석부(1310)에 통보한다. 그러면 패킷분석부(1310)는 메모리(1320)에 저장된 해당 덤패킷에 대한 덤프로우 식별을 수행한다. 즉, 새로운 덤패킷의 수신 사실을 알게 된 패킷분석부(1310)는 기 설정된 정책에 따라 해당 덤패킷의 덤프로우 특성을 식별하여 그 정보를 저장하고 저장 사실을 스케줄러(1350)에 알려준다.
- [106] 스케줄러(1350)는 식별된 덤프로우들을 각각 해당하는 큐에 할당하며, 각 큐를 다중 프로세서(1340)에 병렬 할당한다. 보다 구체적으로, 스케줄러(1350)는 메모리(1320)에 저장된 플로우 테이블을 참조하여 덤패킷의 덤프로우가 맵핑된 큐를 검색하고, 메모리(1320)에 저장된 덤패킷을 검색된 큐에 전달한다. 만약 테이블에 수신한 덤패킷의 덤프로우에 대한 맵핑정보가 존재하지 않는다면, 스케줄러(1350)는 종래의 다양한 방법을 통해 덤프로우를 특정 큐에 할당하고, 덤프로우와 큐사이의 맵핑관계를 플로우 테이블에 저장한다.
- [107] 스케줄러(1350)는 가상 머신별 덤패킷을 덤프로우 단위로 큐잉할 수 있다. 예를 들어, 덤프로우와 큐 사이의 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, 덤프로우를 가상 머신별로 서로 다른 그룹의 큐에 할당하는 것이 바람직하다. 다시 말해, 스케줄러(1350)는 도 14와 같은 가상머신별로 큐들을 그룹핑하는 경우에, 제1 가상머신에 대한 제1 플로는 제1 그룹(1400)의 큐에 덤프로우 단위로 할당하고, 제2 가상머신에 대한 제2 플로는 제2 그룹(1410)의 큐에 덤프로우 단위로 할당한다.
- [108] 예를 들어, 새로운 덤패킷이 메모리(1320)에 적재되었다는 사실과 해당 덤패킷의 덤프로우 정보를 수신하면, 스케줄러(1350)는 플로우 테이블을 참조하여 덤프로우가 어떤 큐에 할당되어 있는지를 검색하고, 메모리(1320)에 적재되어 있는 덤패킷을 검색된 해당 큐에 적재할 수 있도록 한다. 만약 식별된 덤프로우에 대한 정보를 플로우 테이블에서 찾을 수 없는 경우, 스케줄러(1350)는 해당 덤패킷을 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 큐를 선택하는 정책, 덤패킷이 전송될 가상 머신 내 큐 중 가장 부하가 적은 큐를 선택하는 정책, 사용율이 가장 낮은 프로세서에 할당되어 있는 큐를 선택하는 정책 등이 있다.
- [109] 복수 개의 큐(1330)는 각각 적어도 하나 이상의 덤프로우와 맵핑된다. 덤프로우 단위로 큐잉하는 경우 프로세서 친화도가 높아지며, 이에 따라 병렬 처리의 효율도 증가한다. 복수 개의 큐(1330)는 가상 머신별로 적어도 하나 이상의 큐를 포함하는 그룹으로 분할될 수 있다. 또한 복수 개의 큐(1330)는 도 15와 같이

- 적어도 두 개 이상의 파티션으로 분할될 수 있다.
- [110] 스케줄러(1350)는 복수 개의 프로세서 중에서 선택된 프로세서일 수 있다. 예를 들어, 전체 프로세서들(1380) 중 특정 프로세서(1350)를 스케줄러로 지정하거나, 각 프로세서들의 부하 정도를 모니터링부(1360)를 통해 파악한 후 부하가 가장 적은 프로세서를 스케줄러(1350)로 선택할 수 있다. 이 외에도 스케줄러를 선택하기 위한 다양한 방법이 적용될 수 있다. 프로세서 중 스케줄러가 지정되는 경우, 제어부(미도시)는 도 8에 도시된 방법과 같이, 스케줄링이 필요할 때마다 인터럽트 신호를 발생하여 스케줄러로 지정된 프로세서에 전송하고, 인터럽트 신호를 수신한 프로세서는 하던 작업을 중지하고 스케줄러로서 동작을 완료한 후 다시 이전 작업을 수행한다.
- [111] 복수 개의 프로세서들(1340)은 각 큐에 저장된 덱패킷을 병렬 처리하여 서버의 가상 머신으로 전송한다. 복수 개의 프로세서들에 대해서도 1 내지 도 10에서 설명된 단일 스케줄링 또는 다중 스케줄링 방법이 적용될 수 있다. 즉 복수 개의 프로세서들은 그룹핑되어 각 그룹마다 스케줄링될 수 있다. 복수 개의 프로세서들(1340)은 적어도 하나 이상의 큐와 연결된다.
- [112] 예를 들어, 복수 개의 프로세서들(1340)은 플로우 친화도를 고려하여 큐와 연결된다. 다시 말해, 동일하거나 유사한 덱플로우 속성을 갖는 덱패킷을 저장하는 큐들을 묶어 프로세서와 연결한다.
- [113] 또 다른 예로서, 복수 개의 프로세서들(1340)은 가상 머신별로 큐와 연결될 수 있다. 도 14를 참조하면, 제1 프로세서는 제1 가상머신에 할당된 제1 내지 제3 큐(1400)와 연결되고, 제2 프로세서는 제2 가상머신에 할당된 제4 내지 제6 큐(1410)와 연결되고, 제3 프로세서는 제3 가상머신에 할당된 제7 및 제8 큐(1420)에 연결될 수 있다.
- [114] 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 제6 큐와 연결될 수 있다. 즉, 프로세서는 적어도 둘 이상의 가상머신에 할당된 큐의 전부 또는 일부와 연결될 수도 있다.
- [115] 모니터링부(1360)는 프로세서(1340)와 큐(1330)의 부하 등을 포함한 각종 상태를 모니터링한다.
- [116] 큐 관리부(1370)는 모니터링 결과에 따라 큐들을 도 15와 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 설명된 도 12의 과정을 통해 파악된 서버의 가상화 환경 등에 따라서도 가상머신별 큐의 개수나 크기 등을 동적으로 설정할 수 있다.
- [117] 도 14는 본 발명에 따른 NIC의 덱플로우 기반 큐 할당의 일 예를 도시한 도면이다.
- [118] 도 14를 참조하면, 큐들(1330)은 가상 머신별로 구분된다. 예를 들어, 제1 내지

- 제3 큐(1400)는 제1 가상머신에 할당되고, 제4 내지 제6 큐(1410)는 제2 가상머신에 할당되고, 제7 및 제8 큐(1420)는 제3 가상머신에 할당된다. 스케줄러는 가상 머신별로 딥플로우를 참조하여 큐잉을 수행한다.
- [119] 예를 들어, 제1 가상머신으로 향하는 딥플로우를 우선순위에 따라 식별하는 경우에, 스케줄러(1350)는 제1 가상머신에 할당된 제1 내지 제3 큐(1400)에 우선순위를 기반으로 딥패킷을 분류하여 저장한다. 즉 제1 가상머신으로 향하는 딥플로우 중에서 우선순위가 가장 높은 딥플로우는 제1 큐에 저장하고, 그 다음 우선순위의 딥플로우는 제2 큐에 저장하고, 나머지 우선순위의 딥플로우는 제3 큐에 저장한다.
- [120] 도 15는 본 발명에 따른 NIC의 딥플로우 기반 큐 할당의 다른 일 예를 도시한 도면이다.
- [121] 도 15를 참조하면, 큐들(1330)은 적어도 둘 이상의 파티션(1520,1530)으로 구분된다. 각 파티션(1520,1530)마다 스케줄러(1500,1510)가 할당된다. 예를 들어, 제1 파티션(1520)에 제1 스케줄러(1500)가 할당되고, 제2 파티션(1530)에 제2 스케줄러(1510)가 할당된다. 각 스케줄러(1500,1510)는 할당된 파티션에 대하여 독립적으로 스케줄링 작업을 병렬 수행한다. 본 실시 예는 큐들(1330)을 두 개의 그룹(1520,1530)으로 그룹핑하는 예를 도시하고 있으나, 실시 예에 따라 세 개 이상의 그룹으로 그룹핑될 수 있으며, 각 그룹의 스케줄링은 도 1 내지 도 10에서 설명한 다중 스케줄링 방법이 적용될 수 있다. 스케줄러는 앞서 설명한 바와 같이 복수 개의 프로세서들(1380) 중 소정의 방법에 의해 선택된 프로세서일 수 있다.
- [122] 예를 들어, 도 13과 같이 하나의 스케줄러에 의한 스케줄링 수행 중에, 모니터링부에서 측정한 큐의 부하분포가 기 설정된 임계치 이하로 떨어지면, 큐의 재분배 또는 프로세서 재할당이 결정될 수 있다. 또는 네트워크로부터 수신되는 딥패킷의 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 15와 같이 큐가 복수 개의 파티션으로 구분되고 추가적인 스케줄러의 지정이 필요한 경우, 가장 부하가 적은 프로세서를 추가 스케줄러로 지정할 수 있다.
- [123] 각 파티션에 속한 큐들은 가상머신 기반으로 그룹핑(1540)될 수 있으며, 그룹(1540) 내 큐들은 딥플로우 기반으로 분류될 수 있다. 이 경우, 파티션 - 가상머신별 그룹 - 각 그룹별 플로우 단위 큐의 계층적 구조가 생성된다.
- [124] 도 16은 본 발명에 사용되는 딥패킷의 일 예를 도시한 도면이다.
- [125] 도 16을 참조하면, 딥패킷은 물리적 네트워크 프레임(1610), 터널링 필드(1620), 가상머신간 네트워크 프레임(1630), 데이터 필드(1600)를 포함한다.
- [126] 물리적 네트워크 프레임(1610)은 L2, IP, TCP 등 종래의 물리적 네트워크의 계층을 나타내는 정보를 포함한다. 터널링 필드(1620)는 터널링 정보 등을

나타낸다. 가상머신간 네트워크 프레임(1630)은 가상머신간 네트워크 환경에서의 각 계층(vL2 ~ vL7 등)에 대한 정보를 포함한다. 데이터 필드(1600)는 데이터를 포함한다.

- [127] 도 16의 덤패킷의 구조는 본 발명의 이해를 돕기 위한 하나의 예일 뿐, 본 발명이 이에 한정되는 것은 아니며 가상 머신 환경을 위한 다양한 형태의 덤패킷의 구조를 정의하여 사용할 수 있다.
- [128] 또한 메모리에 저장된 덤패킷의 구조와 큐에 저장된 덤패킷의 구조는 실시 예에 따라 동일하거나 다를 수 있다. 예를 들어, 네트워크로부터 수신한 도 16의 패킷을 디캡슐레이션하여 복원한 덤패킷을 가상 머신 환경에서 처리 가능한 최적의 구조로 변경하거나 덤패킷의 필드 중 가상 머신 환경에서 불필요한 필드의 일부 또는 전부를 삭제하는 등 다양하게 설계 변경하여 큐에 저장할 수 있다.
- [129] 도 17은 본 발명에 따른 가상 머신 환경을 위한 패킷 처리 방법의 일 예를 도시한 흐름도이다.
- [130] 도 17을 참조하면, 네트워크 인터페이스 장치는 덤패킷을 수신하면(S1700), DPI 과정 등을 통해 덤패킷을 분석하여 덤패킷이 전달되어야 할 목적지 가상 머신과 덤프로우를 식별한다(S1710). 네트워크 인터페이스 장치는 가상 머신별로 할당된 적어도 한 개 이상의 큐에 대해 덤프로우 단위로 덤패킷을 저장한다(S1720). 그리고 네트워크 인터페이스 장치는 복수 개의 프로세서들을 통해 각 큐에 저장된 덤패킷을 처리하여 가상 머신으로 전송한다(S1730).
- [131] 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.
- [132] 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.

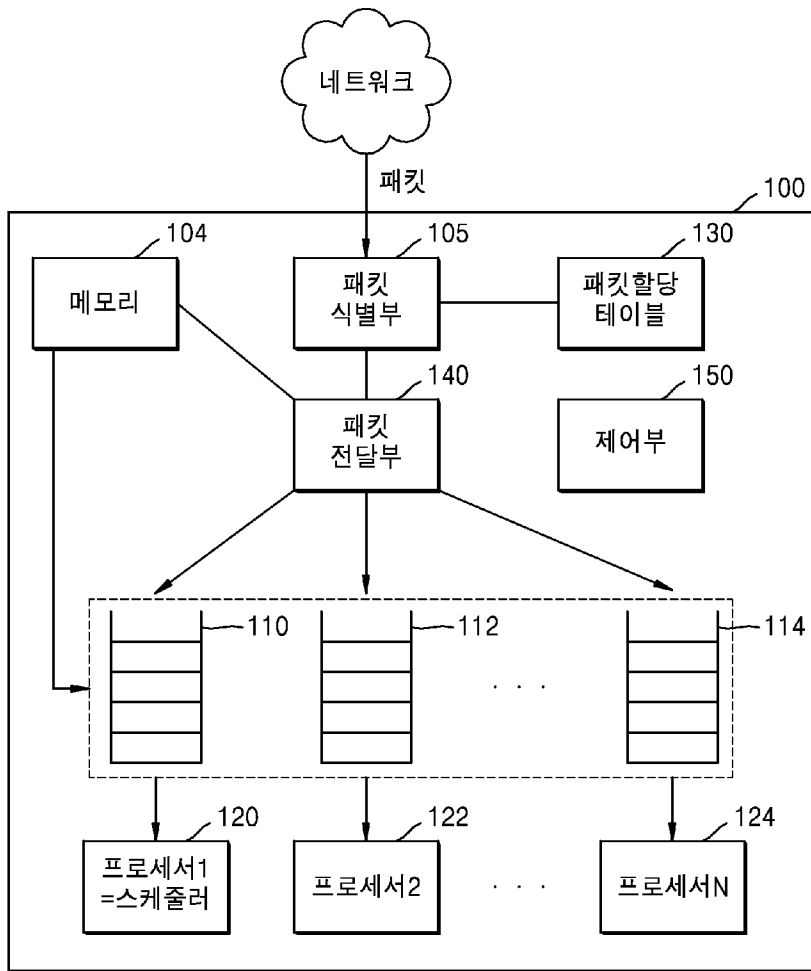
청구범위

- [청구항 1] 다중 프로세싱 장치에서의 스케줄링 방법에 있어서,
 복수의 프로세서들 전체 또는 그 일부를 적어도 하나 이상의 프로세서 그룹으로 그룹핑하는 단계;
 수신한 패킷의 플로우에 대해 기 지정된 프로세서 그룹 또는 프로세서가 존재하면, 상기 플로우를 기 지정된 프로세서 그룹 또는 프로세서에 할당하는 단계; 및
 수신한 패킷의 플로우에 대해 기 지정된 프로세서 그룹 또는 프로세서가 존재하지 않으면, 상기 플로우를 위한 새로운 프로세서 그룹을 생성하여 할당하거나 어느 프로세서 그룹에도 속하지 않은 프로세서에 상기 플로우를 할당하는 단계;를 포함하는 것을 특징으로 하는 스케줄링 방법.
- [청구항 2] 제 1항에 있어서, 상기 새로운 프로세서 그룹을 생성하는 할당하는 단계는,
 복수의 프로세서들 중 그룹핑되지 아니한 프로세서들 전체 또는 그 일부를 새로운 프로세서 그룹으로 그룹핑하는 단계; 및
 새로운 프로세서 그룹에 수신한 패킷의 플로우를 할당하는 단계;를 포함하는 것을 특징으로 하는 스케줄링 방법.
- [청구항 3] 제 1항에 있어서,
 프로세서 그룹별로 프로세서 그룹 내 프로세서들 중 하나에 인터럽트 신호를 전송하는 단계; 및
 인터럽트 신호를 수신한 프로세서는 이전 수행하던 작업을 중단하고, 스케줄링 동작을 수행한 후 다시 이전 수행하던 작업을 재개하는 단계;를 더 포함하는 것을 특징으로 하는 스케줄링 방법.
- [청구항 4] 다중 프로세싱 장치에서의 스케줄링 방법에 있어서,
 복수의 프로세서들의 부하상태나 처리용량 또는 수신 패킷의 속성을 기초로 단일 스케줄링 또는 다중 스케줄링을 결정하는 단계;
 상기 단일 스케줄링의 경우에, 복수의 프로세서들 중 하나를 스케줄러로 지정하는 단계; 및
 상기 다중 스케줄링의 경우에, 복수의 프로세서들을 적어도 둘 이상의 프로세서 그룹으로 그룹핑하고, 각각의 프로세서 그룹 내 프로세서들 중 하나를 각각의 프로세서 그룹의 스케줄러로 지정하는 단계;를 포함하는 것을 특징으로 하는 스케줄링 방법.
- [청구항 5] 제 4항에 있어서,
 상기 다중 스케줄링의 경우에, 수신한 패킷의 플로우에 대해 기 지정된 프로세서 그룹 또는 프로세서가 존재하지 아니하면, 상기 플로우를 위한 새로운 프로세서 그룹을 생성하는 단계;를 더 포함하는 것을 특징으로 하는 스케줄링 방법.

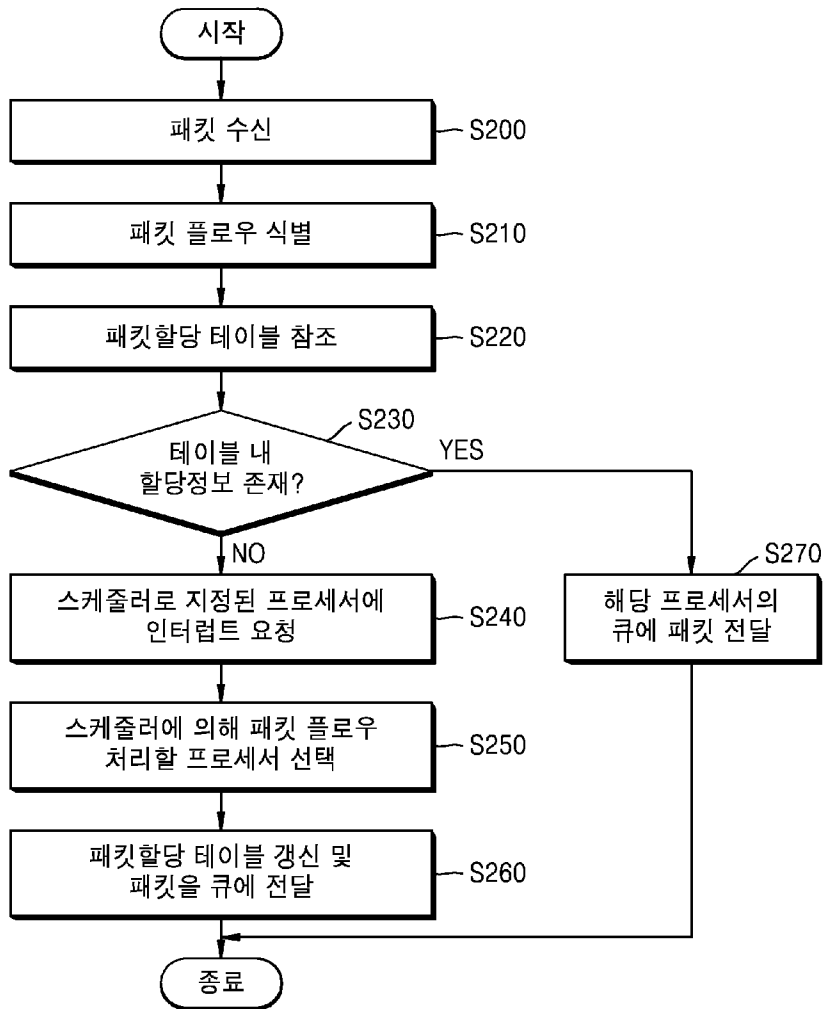
- [청구항 6] 제 5항에 있어서, 상기 새로운 프로세서 그룹을 생성하는 단계는, 복수의 프로세서들 중 그룹핑되지 아니한 프로세서들 전체 또는 그 일부를 새로운 프로세서 그룹으로 생성하는 단계;를 포함하는 것을 특징으로 하는 스케줄링 방법.
- [청구항 7] 물리적 망을 통해 수신된 패킷으로부터 물리적 네트워크 프레임에 인캡슐레이션된, 가상화 환경 네트워크 계층 정보를 포함하는 덩패킷을 획득하는 단계;
상기 덩패킷에 포함된 상기 가상화 환경 네트워크 계층 정보를 기초로 상기 덩패킷을 덩플로우로 식별하는 단계; 및
상기 덩패킷을 상기 식별된 덩플로우 단위로 구분하여 해당하는 큐에 할당하는 단계;를 포함하는 것을 특징으로 하는 스케줄링 방법.
- [청구항 8] 제 7항에 있어서,
하나 이상의 큐를 가상머신별로 그룹핑하는 단계;를 더 포함하는 것을 특징으로 하는 스케줄링 방법.
- [청구항 9] 제 7항에 있어서,
상기 식별하는 단계는, 상기 덩패킷의 목적지 가상머신을 식별하는 단계를 포함하고,
상기 할당하는 단계는, 상기 목적지 가상머신과 상기 덩플로우를 기초로 상기 덩패킷을 해당하는 큐에 할당하는 단계;를 포함하는 것을 특징으로 하는 스케줄링 방법.
- [청구항 10] 제 7항에 있어서,
상기 가상화 환경 네트워크 계층 정보는 가상머신으로 형성된 네트워크 계층 내 패킷 전송을 위하여 물리적 네트워크 프레임에 인캡슐레이션된, 가상머신으로 형성된 네트워크 계층 정보를 의미하고,
상기 덩패킷은 물리적 네트워크에서 일반 통신 프로토콜에 의하여 인식되어 원활한 전송이 이루어질 수 있도록 물리적 네트워크 프레임에 인캡슐레이션된 것을 특징으로 하는 스케줄링 방법.
- [청구항 11] 제 7항에 있어서, 상기 할당하는 단계는,
적어도 하나 이상의 큐를 포함하는 복수 개의 큐 그룹 중 목적지 가상머신을 기초로 큐 그룹을 선택하는 단계; 및
상기 덩플로우를 기초로 상기 선택된 큐 그룹 내 패킷을 저장할 큐를 선택하는 단계;를 포함하는 것을 특징으로 하는 스케줄링 방법.
- [청구항 12] 제 7항에 있어서,
각 가상 머신에 할당되는 큐의 개수는 가상 머신에 관한 정보를 포함하는 가상화 환경 정보 또는 부하분포를 기초로 동적으로 결정되는 것을 특징으로 하는 스케줄링 방법.
- [청구항 13] 제 7항에 있어서,
적어도 하나 이상의 큐를 포함하는 가상 머신별로 할당된 복수 개의 큐

- 그룹을 포함하고,
스케줄러는 상기 덤패킷에 대해, 덤패킷의 목적지 가상 머신을 기초로
상기 복수 개의 큐 그룹 중 하나를 할당하고, 상기 덤프로우를 기초로
선택된 큐 그룹 내 큐를 할당하는 것을 특징으로 하는 스케줄링 방법.
- [청구항 14] 물리적 망을 통해 수신된 패킷으로부터 물리적 네트워크 프레임에
인캡슐레이션된, 가상화 환경 네트워크 계층 정보를 포함하는 덤패킷을
획득하는 패킷수신부;
상기 덤패킷에 포함된 상기 가상화 환경 네트워크 계층 정보를 기초로
상기 덤패킷을 덤프로우로 식별하는 패킷분석부; 및
상기 덤패킷을 상기 식별된 덤프로우 단위로 구분하여 해당하는 큐에
할당하는 스케줄러;를 포함하는 것을 특징으로 하는 네트워크
인터페이스 장치.
- [청구항 15] 제 14항에 있어서, 상기 스케줄러는,
적어도 하나 이상의 큐를 포함하는 복수 개의 큐 그룹 중 목적지 가상
머신을 기초로 큐 그룹을 선택하고, 상기 덤프로우를 기초로 상기 선택된
큐 그룹 내 패킷을 저장할 큐를 선택하는 것을 특징으로 하는 네트워크
인터페이스 장치.

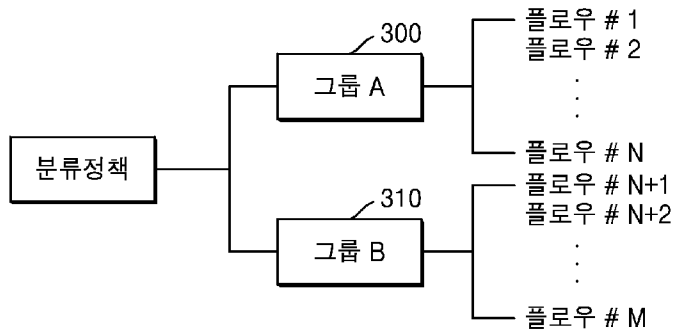
[도1]



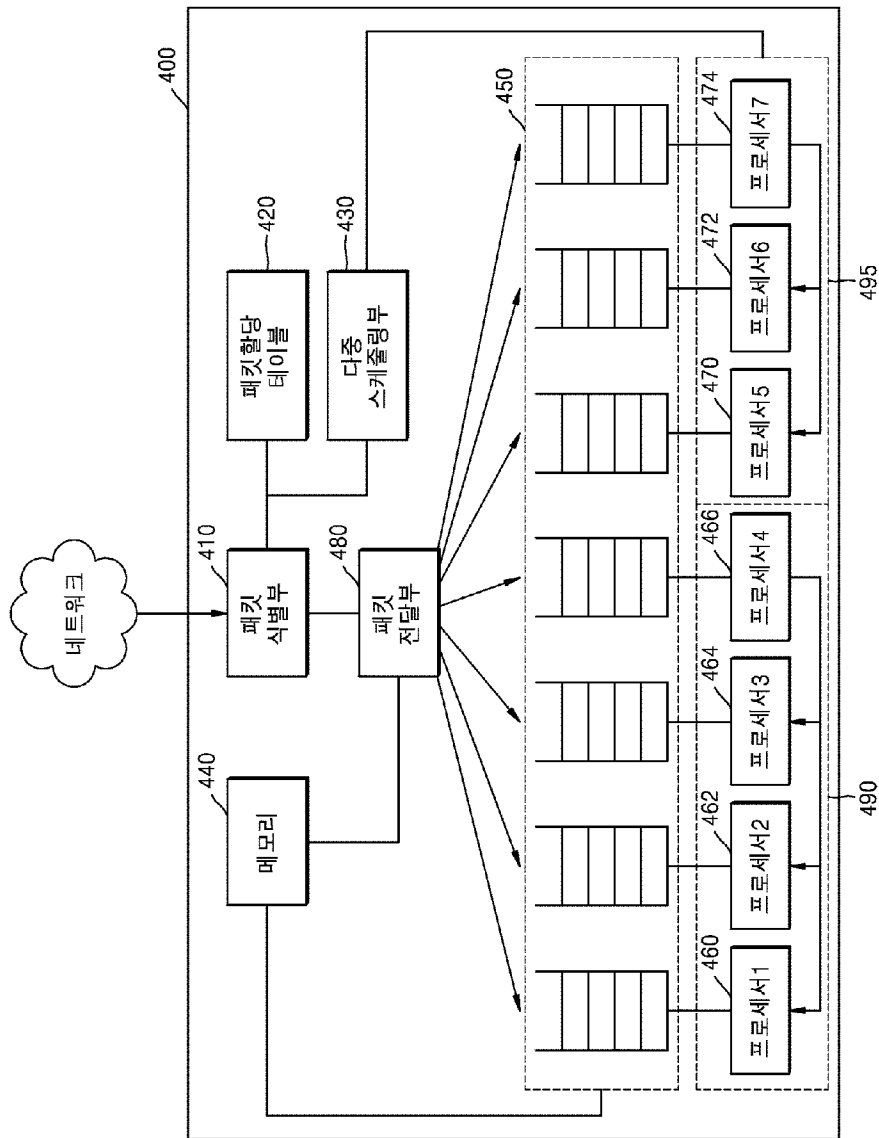
[도2]



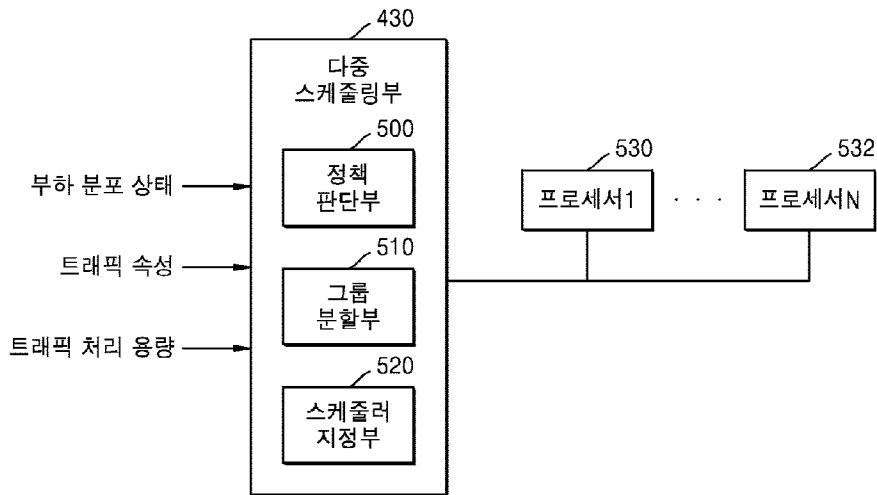
[도3]



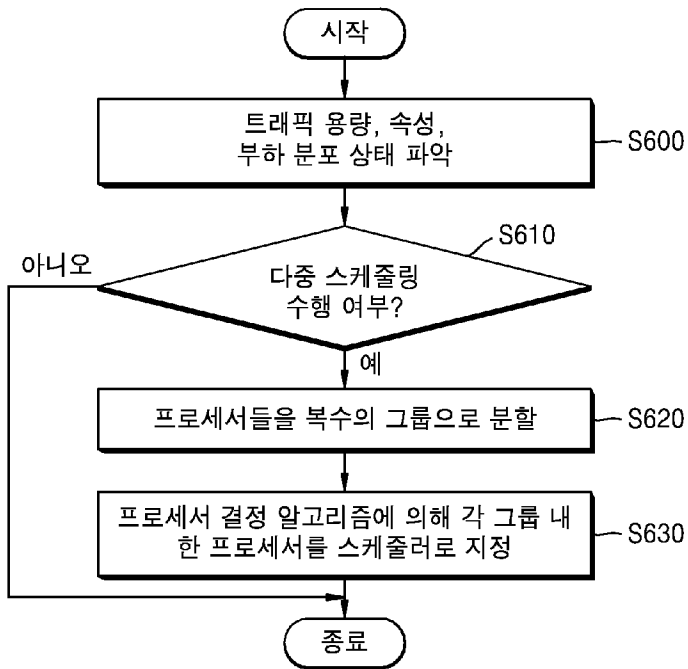
[도4]



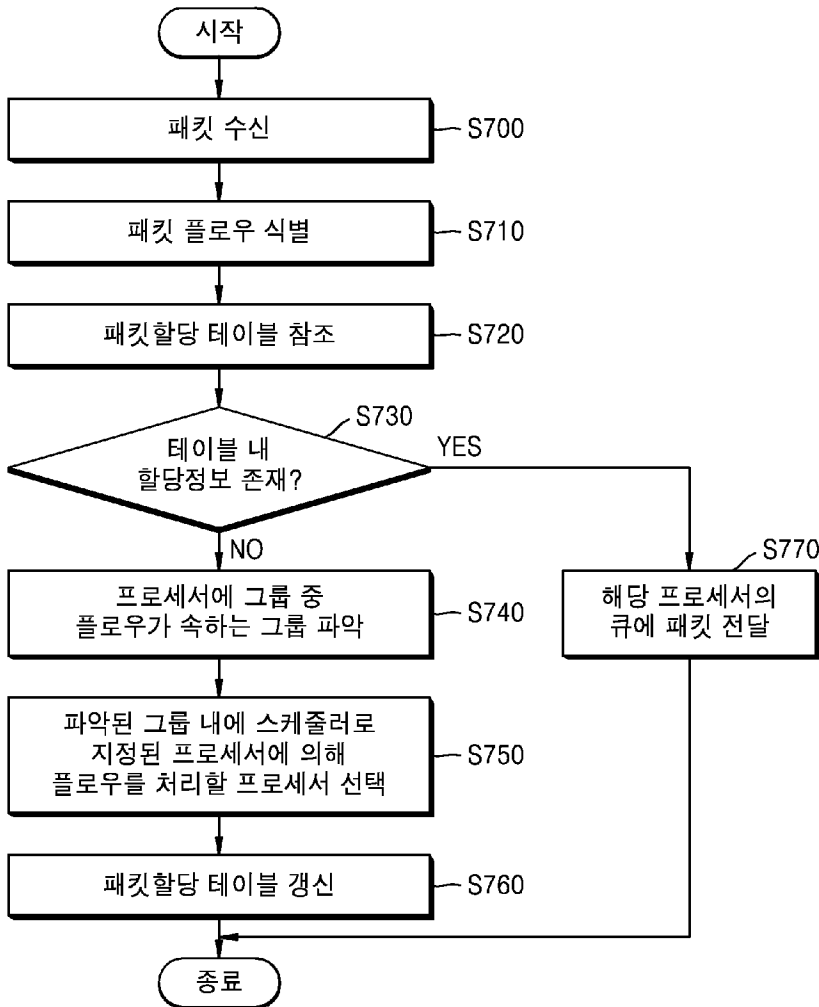
[도5]



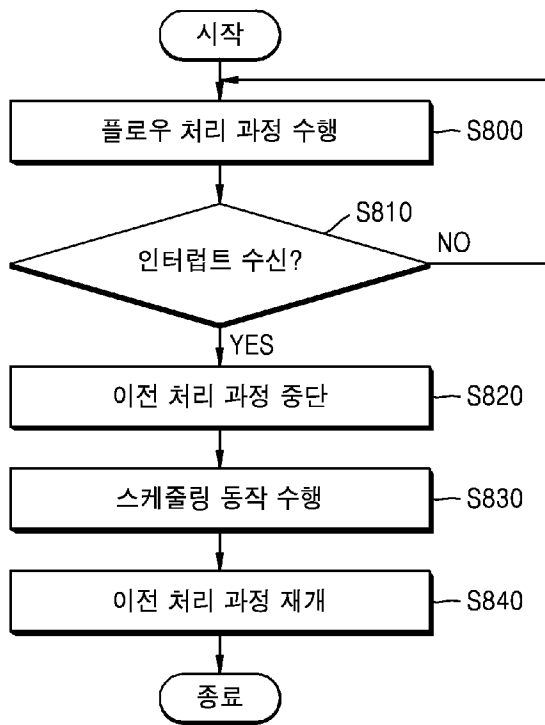
[도6]



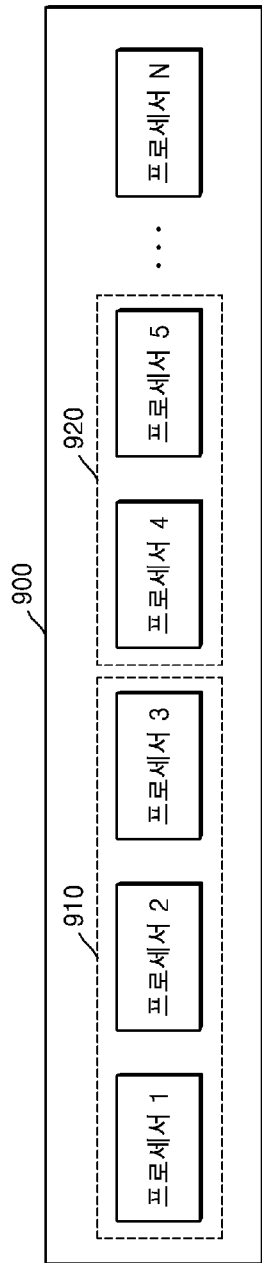
[도7]



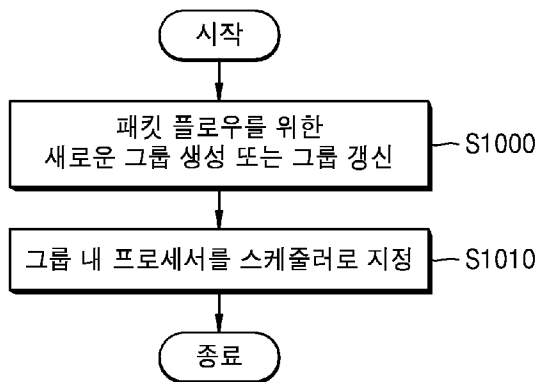
[도8]



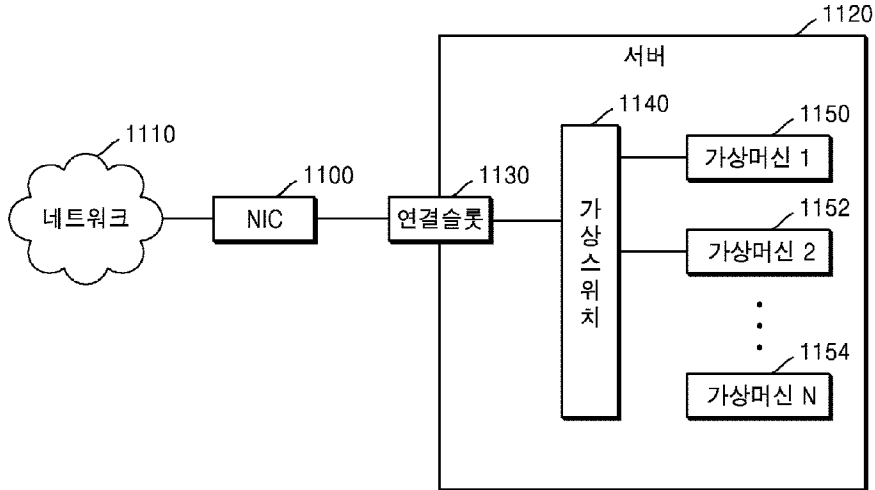
[도9]



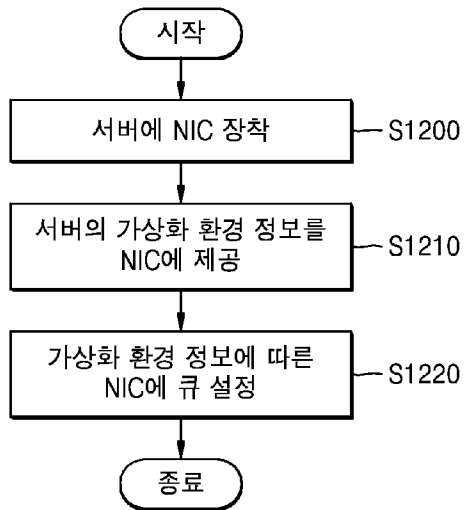
[도10]



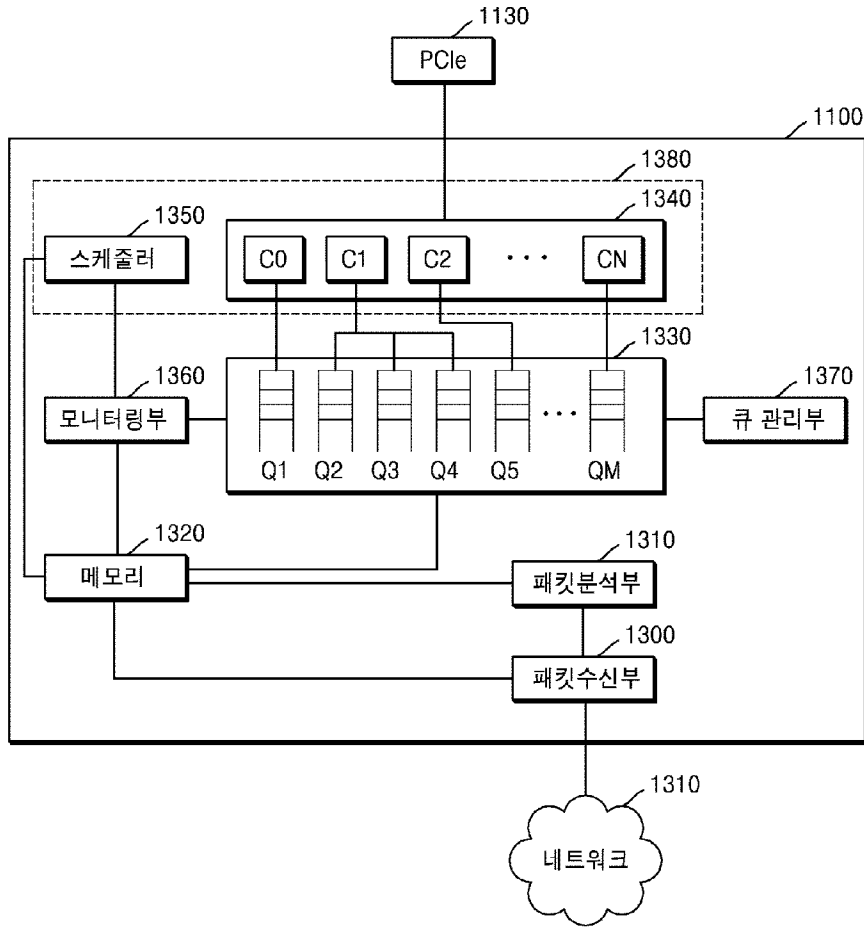
[도11]



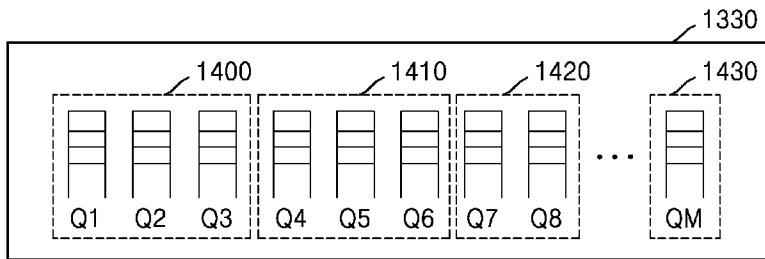
[도12]



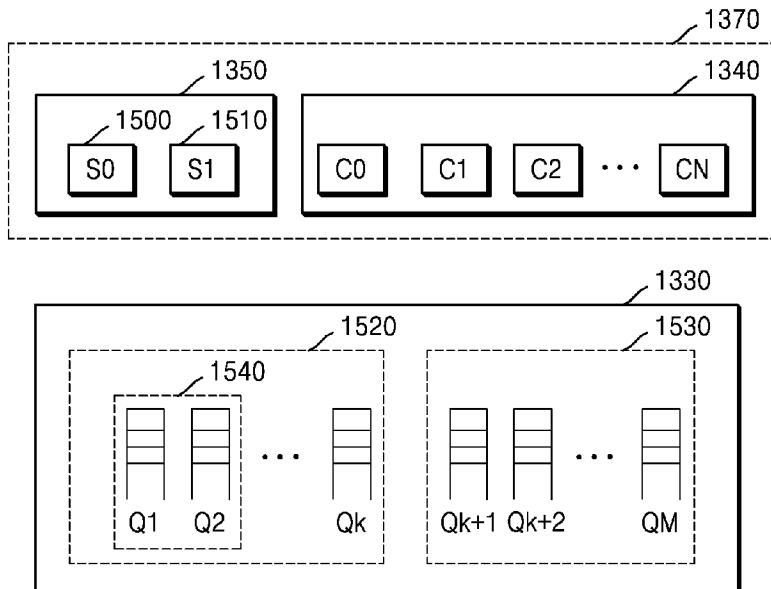
[도13]



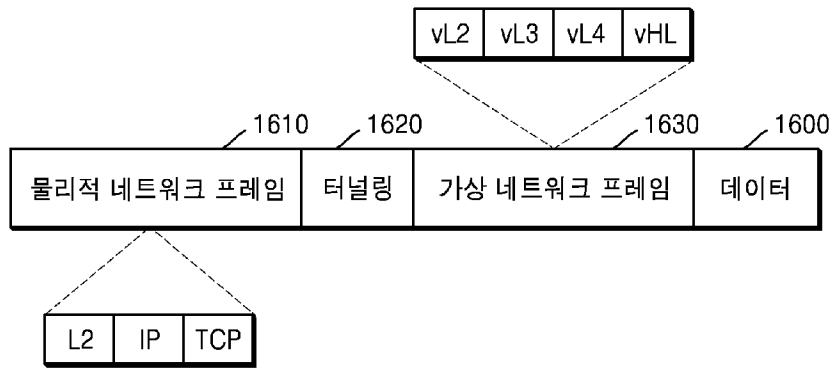
[도14]



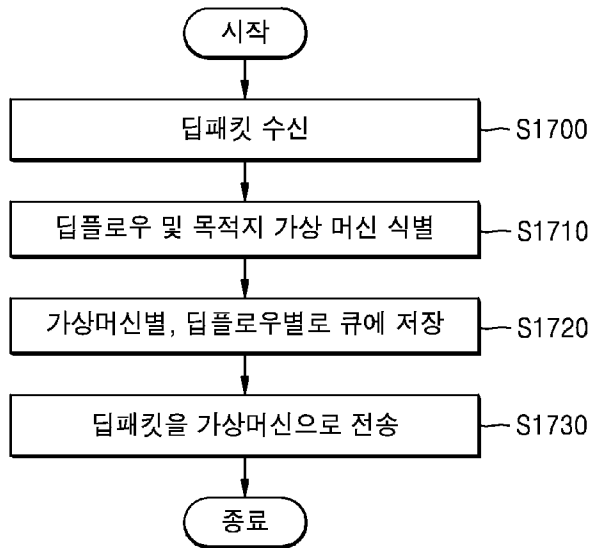
[도15]



[도16]



[도17]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2015/005914

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/48(2006.01)i, G06F 9/50(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 9/48; G06F 15/16; G05B 15/02; G06F 9/00; H04L 12/56; H04L 12/775; G06F 9/50

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean Utility models and applications for Utility models: IPC as above

Japanese Utility models and applications for Utility models: IPC as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS (KIPO internal) & Keywords: processor group, scheduling, packet flow, allocation

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 08-152903A (FUJITSU LTD.) 11 June 1996 See abstract, paragraphs [0010]-[0017] and figure 1.	1-6
A		7-15
Y	KR 10-2004-0005824 A (CROSSBEAM SYSTEMS, INC.) 16 January 2004 See abstract, claims 1 and 2.	1-6
A		7-15
Y	US 2008-0059555 A1 (ARCHER, C. J. et al.) 06 March 2008 See abstract, paragraph [0040], figures 3, 5, claims 1 and 2.	3-6
A		1-2,7-15
X	JP 2002-344503 A (FUJITSU LTD.) 29 November 2002 See abstract, paragraphs [0031]-[0037] and [0059]-[0066], claims 1-3.	7,9,10,12,14
A		1-6,8,11,13,15
A	KR 10-2011-0065942 A (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE et al.) 16 June 2011 See abstract, figure 3, claims 1 and 9.	1-15



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

02 SEPTEMBER 2015 (02.09.2015)

Date of mailing of the international search report

03 SEPTEMBER 2015 (03.09.2015)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 189 Seonsa-ro, Daejeon 302-701,
Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2015/005914

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

Claims 1-6 pertain to a scheduling method for allocating the flow of a received packet by grouping a processor into groups and designating one of the processors in a processor group as a scheduler.

Claims 7-15 pertain to a scheduling method and a network interface device for acquiring and identifying a deep packet and classifying the deep packet by a deep flow unit so as to allocate the classified deep packet to a queue.

1. As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

- The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/KR2015/005914

Patent document cited in search report	Publication date	Patent family member	Publication date
JP 08-152903A	11/06/1996	JP 3429582 B2	22/07/2003
KR 10-2004-0005824 A	16/01/2004	AU 2001-91227 A1	08/04/2002
		AU 2006-265624 A1	11/01/2007
		AU 2006-265624 B2	17/11/2011
		CA 2423475 A1	04/04/2002
		CA 2614328 A1	11/01/2007
		CN 1518694 A	04/08/2004
		CN 1518694 C	16/05/2007
		EP 1381937 A2	21/01/2004
		EP 1898822 A2	19/03/2008
		EP 1960867 A2	27/08/2008
		EP 2432188 A1	21/03/2012
		EP 2442525 A1	18/04/2012
		JP 2004-524598 A	12/08/2004
		JP 2009-504201 A	05/02/2009
		US 2002-0059424 A1	16/05/2002
		US 2002-0165947 A1	07/11/2002
		US 2006-0010207 A1	12/01/2006
		US 2006-0143499 A1	29/06/2006
		US 2007-0006215 A1	04/01/2007
		US 2007-0016183 A1	18/01/2007
		US 2007-0192863 A1	16/08/2007
		US 2008-0133517 A1	05/06/2008
		US 2008-0133518 A1	05/06/2008
		US 2008-0134330 A1	05/06/2008
		US 2008-0162390 A1	03/07/2008
		US 2008-0229415 A1	18/09/2008
		US 2008-0262990 A1	23/10/2008
		US 2008-0262991 A1	23/10/2008
		US 2010-0042565 A1	18/02/2010
		US 7836443 B2	16/11/2010
		US 7979368 B2	12/07/2011
		US 8010469 B2	30/08/2011
		US 8046465 B2	25/10/2011
		US 8080009 B2	20/12/2011
		US 8402540 B2	19/03/2013
		US 8512333 B2	20/08/2013
		WO 02-27469 A2	04/04/2002
		WO 02-27469 A3	06/11/2003
		WO 2007-005830 A2	11/01/2007
		WO 2007-005830 A3	18/12/2008
		WO 2007-070838 A2	21/06/2007
		WO 2007-070838 A3	03/07/2008
US 2008-0059555 A1	06/03/2008	US 7647590 B2	12/01/2010
		WO 2008-025761 A2	06/03/2008
		WO 2008-025761 A3	17/04/2008

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/KR2015/005914

Patent document cited in search report	Publication date	Patent family member	Publication date
JP 2002-344503 A	29/11/2002	NONE	
KR 10-2011-0065942 A	16/06/2011	NONE	

A. 발명이 속하는 기술분류(국제특허분류(IPC))
G06F 9/48(2006.01)i, G06F 9/50(2006.01)i

B. 조사된 분야

조사된 최소문헌(국제특허분류를 기재)
G06F 9/48; G06F 15/16; G05B 15/02; G06F 9/00; H04L 12/56; H04L 12/775; G06F 9/50

조사된 기술분야에 속하는 최소문헌 이외의 문헌
한국등록실용신안공보 및 한국공개실용신안공보: 조사된 최소문헌란에 기재된 IPC
일본등록실용신안공보 및 일본공개실용신안공보: 조사된 최소문헌란에 기재된 IPC

국제조사에 이용된 전산 데이터베이스(데이터베이스의 명칭 및 검색어(해당하는 경우))
eKOMPASS(특허청 내부 검색시스템) & 키워드: 프로세서 그룹, 스케줄링, 패킷 플로우, 할당

C. 관련 문헌

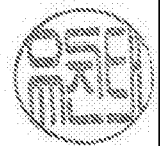
카테고리*	인용문헌명 및 관련 구절(해당하는 경우)의 기재	관련 청구항
Y	JP 08-152903A (FUJITSU LTD) 1996.06.11 요약, 문단번호 [0010]-[0017] 및 도면 1 참조.	1-6
A		7-15
Y	KR 10-2004-0005824 A (크로스빔 시스템즈, 인크.) 2004.01.16 요약, 청구항 1 및 2 참조.	1-6
A		7-15
Y	US 2008-0059555 A1 (ARCHER, C. J. 등) 2008.03.06 요약, 문단번호 [0040], 도면 3, 5, 청구항 1 및 2 참조.	3-6
A		1-2,7-15
X	JP 2002-344503 A (FUJITSU LTD) 2002.11.29 요약, 문단번호 [0031]-[0037] 및 [0059]-[0066], 청구항 1-3 참조.	7,9,10,12,14
A		1-6,8,11,13,15
A	KR 10-2011-0065942 A (한국전자통신연구원 등) 2011.06.16 요약, 도면 3, 청구항 1 및 9 참조.	1-15

추가 문헌이 C(계속)에 기재되어 있습니다. 대응특허에 관한 별지를 참조하십시오.

* 인용된 문헌의 특별 카테고리:
 “A” 특별히 관련이 없는 것으로 보이는 일반적인 기술수준을 정의한 문헌
 “E” 국제출원일보다 빠른 출원일 또는 우선일을 가지나 국제출원일 이후에 공개된 선출원 또는 특허 문헌
 “L” 우선권 주장에 의문을 제기하는 문헌 또는 다른 인용문헌의 공개일 또는 다른 특별한 이유(이유를 명시)를 밝히기 위하여 인용된 문헌
 “O” 구두 개시, 사용, 전시 또는 기타 수단을 언급하고 있는 문헌
 “P” 우선일 이후에 공개되었으나 국제출원일 이전에 공개된 문헌
 “T” 국제출원일 또는 우선일 후에 공개된 문헌으로, 출원과 상충하지 않으며 발명의 기초가 되는 원리나 이론을 이해하기 위해 인용된 문헌
 “X” 특별한 관련이 있는 문헌. 해당 문헌 하나만으로 청구된 발명의 신규성 또는 진보성이 없는 것으로 본다.
 “Y” 특별한 관련이 있는 문헌. 해당 문헌이 하나 이상의 다른 문헌과 조합하는 경우로 그 조합이 당업자에게 자명한 경우 청구된 발명은 진보성이 없는 것으로 본다.
 “&” 동일한 대응특허문헌에 속하는 문헌

국제조사의 실제 완료일 2015년 09월 02일 (02.09.2015)	국제조사보고서 발송일 2015년 09월 03일 (03.09.2015)
--	---

ISA/KR의 명칭 및 우편주소 대한민국 특허청 (35208) 대전광역시 서구 청사로 189, 4동 (둔산동, 정부대전청사) 팩스 번호 +82-42-472-7140	심사관 유진태 전화번호 +82-42-481-8530
---	------------------------------------



제2기재란 일부 청구항을 조사할 수 없는 경우의 의견(첫 번째 용지의 2의 계속)

PCT 제17조(2)(a)의 규정에 따라 다음과 같은 이유로 일부 청구항에 대하여 본 국제조사보고서가 작성되지 아니하였습니다.

- 1. 청구항:
이 청구항은 본 기관이 조사할 필요가 없는 대상에 관련됩니다. 즉,
- 2. 청구항:
이 청구항은 유효한 국제조사를 수행할 수 없을 정도로 소정의 요건을 충족하지 아니하는 국제출원의 부분과 관련됩니다. 구체적으로는,
- 3. 청구항:
이 청구항은 종속청구항이나 PCT규칙 6.4(a)의 두 번째 및 세 번째 문장의 규정에 따라 작성되어 있지 않습니다.

제3기재란 발명의 단일성이 결여된 경우의 의견(첫 번째 용지의 3의 계속)

본 국제조사기관은 본 국제출원에 다음과 같이 다수의 발명이 있다고 봅니다.

청구항 제1항 내지 제6항은 프로세서를 그룹으로 그룹핑하고 프로세서 그룹 내 프로세서들 중 하나를 스케줄러로 지정하여 수신한 패킷의 플로우를 할당하는 스케줄링 방법에 관한 것이고,

청구항 제7항 내지 제15항은 덤패킷을 획득하고 식별하며 덤패킷을 덤프로우 단위로 구분하여 큐에 할당하는 스케줄링 방법 및 네트워크 인터페이스 장치에 관한 것입니다.

- 1. 출원인이 모든 추가수수료를 기간 내에 납부하였으므로, 본 국제조사보고서는 모든 조사 가능한 청구항을 대상으로 합니다.
- 2. 추가수수료 납부를 요구하지 않고도 모든 조사 가능한 청구항을 조사할 수 있었으므로, 본 기관은 추가수수료 납부를 요구하지 아니하였습니다.
- 3. 출원인이 추가수수료의 일부만을 기간 내에 납부하였으므로, 본 국제조사보고서는 수수료가 납부된 청구항만을 대상으로 합니다. 구체적인 청구항은 아래와 같습니다.
- 4. 출원인이 기간 내에 추가수수료를 납부하지 아니하였습니다. 따라서 본 국제조사보고서는 청구범위에 처음 기재된 발명에 한정되어 있으며, 해당 청구항은 아래와 같습니다.

이의신청에
관한 기재

- 출원인의 이의신청 및 이의신청료 납부(해당하는 경우)와 함께 추가수수료가 납부되었습니다.
- 출원인의 이의신청과 함께 추가수수료가 납부되었으나 이의신청료가 보정요구서에 명시된 기간 내에 납부되지 아니하였습니다.
- 이의신청 없이 추가수수료가 납부되었습니다.

국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
JP 08-152903A	1996/06/11	JP 3429582 B2	2003/07/22
KR 10-2004-0005824 A	2004/01/16	AU 2001-91227 A1	2002/04/08
		AU 2006-265624 A1	2007/01/11
		AU 2006-265624 B2	2011/11/17
		CA 2423475 A1	2002/04/04
		CA 2614328 A1	2007/01/11
		CN 1518694 A	2004/08/04
		CN 1518694 C	2007/05/16
		EP 1381937 A2	2004/01/21
		EP 1898822 A2	2008/03/19
		EP 1960867 A2	2008/08/27
		EP 2432188 A1	2012/03/21
		EP 2442525 A1	2012/04/18
		JP 2004-524598 A	2004/08/12
		JP 2009-504201 A	2009/02/05
		US 2002-0059424 A1	2002/05/16
		US 2002-0165947 A1	2002/11/07
		US 2006-0010207 A1	2006/01/12
		US 2006-0143499 A1	2006/06/29
		US 2007-0006215 A1	2007/01/04
		US 2007-0016183 A1	2007/01/18
		US 2007-0192863 A1	2007/08/16
		US 2008-0133517 A1	2008/06/05
		US 2008-0133518 A1	2008/06/05
		US 2008-0134330 A1	2008/06/05
		US 2008-0162390 A1	2008/07/03
		US 2008-0229415 A1	2008/09/18
		US 2008-0262990 A1	2008/10/23
		US 2008-0262991 A1	2008/10/23
		US 2010-0042565 A1	2010/02/18
		US 7836443 B2	2010/11/16
		US 7979368 B2	2011/07/12
		US 8010469 B2	2011/08/30
		US 8046465 B2	2011/10/25
		US 8080009 B2	2011/12/20
		US 8402540 B2	2013/03/19
		US 8512333 B2	2013/08/20
		WO 02-27469 A2	2002/04/04
		WO 02-27469 A3	2003/11/06
		WO 2007-005830 A2	2007/01/11
		WO 2007-005830 A3	2008/12/18
		WO 2007-070838 A2	2007/06/21
WO 2007-070838 A3	2008/07/03		
US 2008-0059555 A1	2008/03/06	US 7647590 B2	2010/01/12
		WO 2008-025761 A2	2008/03/06
		WO 2008-025761 A3	2008/04/17

국제조사보고서에서 인용된 특허문헌	공개일	대응특허문헌	공개일
JP 2002-344503 A	2002/11/29	없음	
KR 10-2011-0065942 A	2011/06/16	없음	