(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2009년02월11일
(11) 등록번호 10-0883405
(24) 등록일자 2009년02월05일

(51) Int. Cl.
G06F 15/16 (2006.01)
(21) 출원번호 10-2003-7016335
(22) 출원일자 2003년12월12일
심사청구일자 2007년01월26일
번역문제출일자 2003년12월12일
(65) 공개번호 10-2004-0010707
(43) 공개일자 2004년01월31일
(86) 국제출원번호 PCT/US2002/002541
국제출원일자 2002년01월29일
(87) 국제공개번호 WO 2002/103579
국제공개일자 2002년12월27일
(30) 우선권주장
09/881,848 2001년06월18일 미국(US)
(56) 선행기술조사문헌
US6006232
WO09923585
전체 청구항 수 : 총 10 항

(73) 특허권자
어드밴스드 마이크로 디바이시즈, 인코포레이티드
미국 캘리포니아 94088-3453 서니베일 원 에이엠디 플레이스 메일 스톱68
(72) 발명자
에림리바하디르
미국캘리포니아95008캠벨렌웨이2249
아카리아야틴
미국캘리포니아94089
서니베일와일드우드애비뉴#911235
(74) 대리인
박장원

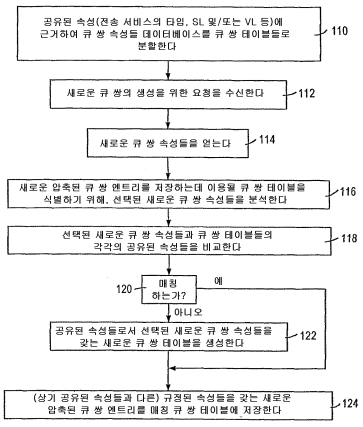
심사관 : 김성훈

(54) 공유된 속성들에 근거하여 압축된 큐 쌍으로부터 다중의가상 큐 쌍들을 생성하는 장치

(57) 요 약

호스트 채널 어댑터는 유사한 특성들을 갖는 큐 쌍들을, 공유된 속성들을 갖는 압축된 큐 쌍 엔트리들을 저장하도록 구성된 큐 쌍 테이블들로 압축함으로써 다중의 큐 쌍들을 효율적으로 관리하도록 구성된다. 따라서, 다중의 가상 큐 쌍들은 큐 쌍 속성들 데이터베이스내에 저장된 소수의 물리적 큐 쌍들로부터 생성될 수 있다.

대 표 도 - 도3



특허청구의 범위

청구항 1

새로운 큐 쌍의 생성을 위한 요청을 수신하는 단계와;

상기 새로운 큐 쌍에 대한 적어도 하나의 선택된 속성과 복수의 큐 쌍 테이블들의 각각의 공유된 속성들과의 사이에 제1의 매칭을 판단함으로써 상기 복수의 큐 쌍 테이블들 중 하나를 식별하는 단계와, 여기서 각각의 큐 쌍 테이블은 대응하는 공유된 속성을 갖는 큐 쌍들을 표시하는 압축된 큐 쌍 엔트리들을 가지며; 그리고

상기 대응하는 공유된 속성과는 다른, 상기 새로운 큐 쌍의 규정된 속성들을 갖는 새로운 압축된 큐 쌍 엔트리를, 상기 적어도 하나의 선택된 속성에 매칭하는 상기 대응하는 공유된 속성을 갖는 상기 하나의 공유된 큐 쌍 테이블에 저장하는 단계를 포함하는 것을 특징으로 하는 호스트 채널 어댑터에서의 큐 쌍 관리 방법.

청구항 2

제 1 항에 있어서,

상기 적어도 하나의 선택된 속성은 전송 서비스의 타입을 특정하고, 상기 제1의 매칭을 판단하는 것은 상기 특정된 전송 서비스의 타입과, 상기 큐 쌍 테이블들의 상기 각각의 대응하는 공유된 속성들로서 할당된 각각의 전송 서비스의 타입 간의 매칭을 판단하는 것을 포함하는 것을 특징으로 하는 호스트 채널 어댑터에서의 큐 쌍 관리 방법.

청구항 3

제 2 항에 있어서,

각각의 큐 쌍 테이블의 상기 대응하는 공유된 속성들은 대응하는 전송 서비스의 타입 및 대응하는 서비스 레벨을 포함하고, 상기 제1의 매칭을 판단하는 것은 상기 특정된 전송 서비스의 타입과 상기 특정된 전송 서비스의 타입에 관련된 상기 새로운 큐 쌍에 대한 할당된 서비스 레벨과의 매칭 및 상기 특정된 전송 서비스의 타입과 상기 큐 쌍 테이블들의 각각에 할당된 규정된 서비스 레벨과의 매칭을 판단하는 것을 더 포함하는 것을 특징으로 하는 호스트 채널 어댑터에서의 큐 쌍 관리 방법.

청구항 4

제 3 항에 있어서,

상기 특정된 전송 서비스의 타입은 신뢰성 있는 접속(reliable connection), 신뢰성 있는 데이터그램(reliable datagram), 신뢰성 없는 접속(unreliable connection), 신뢰성 없는 데이터그램(unreliable datagram) 또는 원시 데이터그램(raw datagram) 중 하나를 특정하는 것을 특징으로 하는 호스트 채널 어댑터에서의 큐 쌍 관리 방법.

청구항 5

제 2 항에 있어서,

상기 새로운 큐 쌍을 할당된 가상 레인에 할당하는 단계를 더 포함하고, 각각의 큐 쌍 테이블의 상기 대응하는 공유된 속성들은 대응하는 전송 서비스의 타입 및 대응하는 규정된 가상 레인을 포함하고, 상기 제1의 매칭을 판단하는 것은 상기 특정된 전송 서비스의 타입과 상기 특정된 전송 서비스의 타입에 관련된 상기 새로운 큐 쌍에 대한 상기 할당된 가상 레인과의 매칭 및 상기 특정된 전송 서비스의 타입과 상기 큐 쌍 테이블들의 각각에 대한 상기 규정된 가상 레인과의 매칭을 판단하는 것을 더 포함하는 것을 특징으로 하는 호스트 채널 어댑터에서의 큐 쌍 관리 방법.

청구항 6

제 1 항에 있어서,

상기 적어도 하나의 선택된 속성은 할당된 가상 레인을 특정하고, 상기 제1의 매칭을 판단하는 것은 상기 할당된 가상 레인과 상기 큐 쌍 테이블들의 상기 각각의 대응하는 공유된 속성들로서 할당된 각각의 규정된 가상 레

인들 간의 매칭을 판단하는 것을 포함하는 것을 특징으로 하는 호스트 채널 어댑터에서의 큐 쌍 관리 방법.

청구항 7

큐 쌍들의 규정된 속성들을 저장하도록 구성된 큐 쌍 속성들 데이터베이스와, 여기서 상기 큐 쌍 속성들 데이터베이스는 각각의 공유된 속성들을 갖는 복수의 큐 쌍 테이블들을 포함하고, 각각의 큐 쌍 테이블은 대응하는 공유된 속성을 갖는 큐 쌍들을 표시하는 압축된 큐 쌍 엔트리들을 저장하도록 구성되며; 그리고

새로운 큐 쌍의 적어도 하나의 선택된 속성에 매칭하는 상기 대응하는 공유된 속성을 갖는 상기 큐 쌍 테이블들 중 식별된 하나에, 새로운 압축된 큐 쌍 엔트리로서 상기 새로운 큐 쌍을 저장하도록 구성된 큐 쌍 속성들 관리 모듈을 포함하여 구성되며, 여기서 상기 새로운 압축된 큐 쌍 엔트리는 상기 대응하는 공유된 속성과는 다른, 상기 새로운 큐 쌍의 규정된 속성들을 갖는 것을 특징으로 하는 호스트 채널 어댑터.

청구항 8

제 7 항에 있어서,

상기 대응하는 공유된 속성은 전송 서비스의 타입 및 서비스 레벨 중 적어도 하나를 포함하는 것을 특징으로 하는 호스트 채널 어댑터.

청구항 9

제 7 항에 있어서,

상기 대응하는 공유된 속성은 전송 서비스의 타입 및 가상 레인 중 적어도 하나를 포함하는 것을 특징으로 하는 호스트 채널 어댑터.

청구항 10

제 9 항에 있어서,

상기 새로운 큐 쌍의 상기 적어도 하나의 선택된 속성은 상기 전송 서비스의 타입으로서, 신뢰성 있는 접속(reliable connection), 신뢰성 있는 데이터그램(reliable datagram), 신뢰성 없는 접속(unreliable connection), 신뢰성 없는 데이터그램(unreliable datagram) 또는 원시 데이터그램(raw datagram) 중 하나를 특정하는 것을 특징으로 하는 호스트 채널 어댑터.

명 세 서

기 술 분 야

<1> 본 발명은 타겟 채널 어댑터들과 통신을 행하고, InfiniBand™ 서버 시스템에서 데이터 패킷들의 전송에 이용되는 큐 쌍들(queue pairs)을 관리하도록 구성되는 호스트 채널 어댑터에 관한 것이다.

배 경 기 술

<2> 네트워킹 기술은 임무 결정적 네트워킹 응용들(mission critical networking applications)에서 더 강하고 신뢰성 있는 서버들을 제공해야 하는 목표하에서 서버 아키텍처 및 설계에 있어 여러가지 개선들이 요구되고 있다. 특히, 클라이언트 요청들에 응답하는데 있어 서버들의 이용은, 이 서버들이 네트워크가 동작가능한 상태를 유지하도록 극도로 높은 신뢰성을 가져야 함을 요한다. 이에 따라, 서버 신뢰성(reliability), 접근성(accessibility) 및 서비스가능성(serviceability)에 대해 상당한 관심을 갖게 되었다.

<3> 또한, 서버들에 이용되는 프로세서들에 대해서도 상당한 개선들이 요구되는바, 마이크로프로세서 속도 및 대역폭이 접속된 입/출력(I/O) 버스들의 용량을 초과하게 되는 경우, 서버 처리량이 상기 버스 용량에 한정되었었다. 이에 따라, 어드레싱, 프로세서 클러스터링(clustering) 및 고속 I/O에 의하여 서버 성능을 개선시키고자 하는 여러가지 서버 표준들이 제안되어 왔다.

<4> 이들 제안된 서버 표준들에 의해 InfiniBand™ 무역협회에 의해 채택된 InfiniBand™ 아키텍처 명세서(Architecture Specification)(릴리즈 1.0)가 개발되게 되었다. 상기 InfiniBand™ 아키텍처 명세서는 중앙 처

리 장치들, 주변장치들 및 서버 시스템 내부의 스위치들 간의 고속 네트워킹 접속을 기술한다. 따라서, 용어 "InfiniBand™ 네트워크"는 서버 시스템 내의 네트워크를 말한다. 상기 InfiniBand™ 아키텍처 명세서는 I/O 동작들과 프로세스간 통신들(IPC: interprocessor communications)을 둘다 기술한다.

<5> InfiniBand™ 아키텍처 명세서의 특징은 TCP/IP 기반의 프로토콜들과 같은 기존의 네트워킹 프로토콜들에 존재하는 전송 계층 서비스들을 하드웨어로 구현한다는 것이다. 전송 계층 서비스들을 하드웨어 기반으로 구현하면, 중앙 처리 장치의 처리 요건들을 감소시키는 장점(즉, "무부하(offloading)")을 제공함으로써, 상기 서버 시스템의 운영 체제에 부하가 걸리지 않게 한다.

<6> 그러나, 임의의 하드웨어 구현들은 실질적으로 비용이 많이 드는 하드웨어 설계들을 필요로 할 수 있다. 호스트 채널 어댑터(HCA: host channel adapter)는 다중의 큐 쌍들(QPs: queue pairs)을 관리하는데, 이 다중의 큐 쌍들은 데이터 통신을 행하기 위해 InfiniBand™ 네트워크 노드들에서 소비자 응용들에 의해 이용된다. 불행하게도, 상당히 많은 수의 큐 쌍들이 발생될 수 있는바, 이것은 상기 HCA가 상기 많은 수의 큐 쌍들을 관리하기 위해서는 상당히 많은 자원들을 확장해야 함을 요구한다.

발명의 상세한 설명

<7> 호스트 채널 어댑터가 효율적이고 경제적인 방식으로 구현될 수 있게 해주는 장치에 대한 필요성이 존재한다.

<8> 호스트 채널 어댑터가 자원들의 실질적인 확장이 없이도 다중의 큐 쌍들을 관리할 수 있게 해주는 장치에 대한 필요성이 또한 존재한다.

<9> 이들 및 기타 다른 필요성들은 본 발명에 의해 달성되고, 여기서 유사한 특성들을 갖는 큐 쌍들을, 공유된 속성들을 갖는 압축된 큐 쌍 엔트리들을 저장하도록 구성된 큐 쌍 테이블들로 압축함으로써 다중의 큐 쌍들을 효율적으로 관리하도록 호스트 채널 어댑터가 구성된다.

<10> 본 발명의 일 양상은 호스트 채널 어댑터에 있어 하나의 방법을 제공한다. 상기 방법은 새로운 큐 쌍의 생성을 위한 요청을 수신하는 단계를 포함한다. 또한, 상기 방법은 상기 새로운 큐 쌍에 대한 적어도 하나의 선택된 속성과 상기 큐 쌍 테이블들의 각각의 공유된 속성들 간의 매칭(match)을 판단함으로써 다수의 큐 쌍 테이블들 중 하나를 식별하는 단계를 포함하고, 각각의 큐 쌍 테이블은 대응하는 공유된 속성을 갖는 큐 쌍들을 표시하는 압축된 큐 쌍 엔트리들을 갖는다. 상기 방법은 상기 대응하는 공유된 속성과는 다른, 상기 새로운 큐 쌍의 규정된 속성들을 갖는 새로운 압축된 큐 쌍 엔트리를, 상기 적어도 하나의 선택된 속성을 매칭하는 상기 대응하는 공유된 속성을 갖는 하나의 공유된 큐 쌍 테이블에 저장하는 단계를 또한 포함한다.

<11> 본 발명의 다른 양상은 호스트 채널 어댑터를 제공한다. 상기 호스트 채널 어댑터는 큐 쌍 속성들 데이터베이스 및 큐 쌍 속성들 관리 모듈을 포함한다. 상기 큐 쌍 속성들 데이터베이스는 큐 쌍들의 규정된 속성들을 저장하도록 구성되고, 상기 큐 쌍 속성들 데이터베이스는 각각의 공유된 속성들을 갖는 다수의 큐 쌍 테이블들을 포함하고, 각각의 큐 쌍 테이블은 상기 대응하는 공유된 속성을 갖는 큐 쌍들을 표시하는 압축된 큐 쌍 엔트리들을 저장하도록 구성된다. 상기 큐 쌍 속성들 관리 모듈은 새로운 압축된 큐 쌍 엔트리로서 새로운 큐 쌍을, 상기 새로운 큐 쌍의 적어도 하나의 선택된 속성을 매칭하는 상기 대응하는 공유된 속성을 갖는 상기 큐 쌍 테이블들 중 식별된 하나의 테이블에 저장하도록 구성되고, 상기 새로운 압축된 큐 쌍 엔트리는 상기 대응하는 공유된 속성과는 다른, 상기 새로운 큐 쌍의 규정된 속성들을 갖는다.

<12> 본 발명의 추가적인 장점들 및 독창적인 특징들은 다음의 상세한 설명에서 부분적으로 설명될 것이고, 다음의 설명을 숙지한 당업자이면 부분적으로 알 수 있게 되거나 또는 본 발명의 실시에 의해 알 수 있게 된다. 본 발명의 장점들은 첨부된 청구항들에서 특별히 기재된 수단들 및 결합들에 의하여 알 수 있게 되고 달성될 수 있다.

실 시 예

<17> 도 1은 본 발명의 일 실시예에 따라 패킷들을 발생하고 전송하도록 구성된 호스트 채널 어댑터(HCA)(12)를 예시하는 블록도이다. InfiniBand™ 아키텍처 명세서에 따른 상기 HCA(12)는, 우선순위(priority-based ordering)에 따라 전송 패킷들을 발생시킴으로써 하드웨어 자원들이 효율적으로 이용되도록 하는 방식으로 구현된다. 또한, 상기 HCA(12)는 트래픽 흐름을 깨지 않고 내장형 프로세스들을 추가시킬 수 있게 함으로써 유연성을 제공한다. 따라서, 상기 HCA(12)는 통상적인 구현 기술들에 비해 복잡성이 매우 적은 경제적인 방식으로 구현될 수 있다.

<18> 상기 InfiniBand™ 아키텍처 명세서에 따른 상기 HCA(12)를 구현하는 통상의 장치들이 갖는 하나의 문제점은, 전송 계층 서비스가 먼저 예를 들면, 전송 계층 헤더를 구성하고, 패킷 시퀀스 번호를 발생시키고, 서비스 타입(예를 들면, 신뢰성 있는 접속, 신뢰성 있는 데이터그램, 신뢰성 없는 접속, 신뢰성 없는 데이터그램 등) 및 다른 전송 계층 동작들을 유효화함으로써 수행된다는 점이다. 일단 상기 전송 계층 동작들이 완료되면, 상기 패킷은 서비스 계층과 가상 레인의 매핑, 링크 계층 흐름 제어 패킷 발생, 링크 계층 전송 크레디트 검사 및 다른 동작들을 포함하는 링크 계층 동작들을 위해 링크 계층 서비스에 전송된다. 비록 이러한 통상적인 구현 타입이 InfiniBand™ 아키텍처 명세서에 기술된 네트워크 계층들을 정확하게 따른다는 장점이 있지만은, 이러한 장치는 상당히 많은 하드웨어를 필요로 한다. 특히, 전송 계층은 더 복잡한 동작들을 포함하기 때문에 일반적으로 링크 계층보다 더 많은 프로세싱 전력을 필요로 한다. 따라서, 상기 전송 계층의 하드웨어로의 구현시에도 하드웨어 시스템이 복잡하지 않게 할 필요성이 존재한다. 또한, 낮은 우선순위 동작들에서 전송 계층 자원들이 불필요하게 낭비될 우려가 있다.

<19> 개시된 실시예에 따르면, 링크 계층 동작들은 전송될 데이터 패킷들의 우선순위들의 결정의 바람직성(desirability)에 근거하여 분할된다. 특히, 도 1의 상기 HCA(12)는 수신된 WQE들의 우선순위를 결정하도록 구성되는 사전-링크 모듈(pre-link module) 및 네트워크에서 전송을 위해 데이터 패킷을 준비하도록 구성되는 사후-링크 모듈(post-link module)을 포함한다. 상기 사전-링크 모듈(40)은 상기 사전-링크 모듈에 의해 결정된 우선순위들에 따라 상기 WQE들의 순서를 정하고, 관련된 큐 쌍 속성들에 근거하여 상기 WQE들에 대한 적절한 전송 계층 헤더들을 발생시키도록 구성되는 전송 서비스 모듈(42)에 상기 결정된 순서로 상기 WQE들을 출력한다. 다시 말하면, 상기 사전-링크 모듈(40)은 상기 전송 서비스 모듈(42)이 상기 전송 계층 프로세스내에서 낮은 우선순위 WQE들에 대한 자원들을 낭비하지 못하게 하거나 또는 높은 우선순위 WQE들을 차단하지 못하게 한다. 따라서, 더 높은 우선순위 접속들은 상기 HCA를 통해 상기 전송 계층에서 개선된 서비스를 얻을 수 있다.

<20> 예를 들면, 주문형 집적회로로서 구현되는 상기 HCA(12)는 사전-링크 모듈(40), 전송 서비스 모듈(42), 사후-링크 모듈(44) 및 미디어 액세스 제어(MAC: media access control) 모듈(46)을 포함한다. 상기 HCA(12)는 또한 하기에 설명되는, 전송 데이터를 저장하도록 구성되는 메모리(48)로의 로컬 액세스 및 오버플로우 버퍼들을 갖는다.

<21> 상기 사전-링크 모듈(40)은 작업 큐 요소 FIFO(work queue element FIFO)(50), 가상 레인 FIFO들(virtual lane FIFOs)(52), 사전-링크 프로세스 모듈(54), 서비스 계층-가상 레인(SL-VL) 매핑 테이블(56), 가상 레인(VL) 중재 테이블(58) 및 가상 레인(VL) 중재 모듈(60)을 포함한다.

<22> 상기 HCA(12)는 작업 큐 요소(WQE)들의 형태로 중앙 처리 장치(CPU)로부터 데이터를 수신하도록 구성되고, 상기 WQE FIFO(50)에 저장된다. 각각의 WQE는 대응하는 규정된 동작이 목적지 InfiniBand™ 네트워크 노드(즉, "응답자") 예를 들면, 타겟에 의해 수행되도록 하는, 상기 CPU(즉, "요청자")에 의해 실행되는 소비자 응용으로부터의 대응 요청을 특정한다. 요청자와 응답자 간의 상호동작은 큐 쌍(QP: queue pair)을 통해 특정되고, 여기서 큐 쌍은 전송 작업 큐 및 수신 작업 큐를 포함한다.

<23> 상기 WQE는 서비스 레벨(SL) 정보 및 상기 시스템 메모리(48)에서의 실제 메시지의 위치에 대한 포인터를 포함한다. 상기 InfiniBand™ 아키텍처 명세서는 상기 InfiniBand™ 네트워크(10)를 통과하는 패킷이 16가지의 이용가능한 서비스 레벨들 중 하나의 레벨에서 동작하도록 하는 서비스 레벨(SL) 속성을 정의한다. 따라서, 상기 요청자는 상기 WQE의 선택된 우선순위에 근거하여 (예를 들면, 서비스 품질, 우선순위 등에 근거하는) 이용가능한 서비스 레벨을 선택할 수 있다.

<24> 상기 사전-링크 모듈(40)은 서비스 레벨-가상 레인 매핑(SL-VL 매핑)과 가상 레인 중재를 둘다 제공한다. 특히, 상기 InfiniBand™ 아키텍처 명세서에 정의되어 있는 가상 레인들은 다중 논리 흐름들이 단일 물리적 링크를 통해 구현될 수 있게 해주는바, 링크 레벨 흐름 제어는 다른 가상 레인들에 영향을 미치지 않고 하나의 가상 레인에 적용될 수 있다. 상기 사전-링크 프로세스 모듈(54)은 상기 서비스 계층-가상 계층 매핑 테이블(56)을 관리 및 유지하도록 구성된다. 특히, 상기 사전-링크 프로세스 모듈(54)은 상기 WQE FIFO(50)로부터 WQE를 검색하고, 상기 WQE 내에서 특정된 서비스 계층에 근거하여 대응하는 가상 레인을 결정한다. 상기 검색된 WQE에 대한 적절한 가상 레인을 식별하면, 상기 사전-링크 프로세스 모듈(54)은 대응하는 가상 레인 FIFO(52)에 상기 WQE를 포워딩한다.

<25> 상기 사전-링크 모듈(40)은 상기 사전-링크 프로세스 모듈(54)에 의한 할당에 근거하여 WQE들의 저장을 위한 가

상 레인 FIFO들(52a, 52b, 52c, 52d, 52e 및 52f)을 포함한다. 예를 들면, 상기 가상 레인 FIFO(52a)는 내장형 프로세서 동작들 예를 들면, 링크 계층 제어 패킷들 및 에러 상태들의 처리와 관련된 WQE들을 저장하는데 이용된다. 다시 말하면, 규정된 동작이 하드웨어로 구현되지 않을 때, 상기 요청은 하기에 설명되는, 내장형 프로세서(80)에 의한 추가의 프로세싱을 위해 내장형 프로세서 큐(78)에 전송되며, 따라서, 상기 내장형 프로세서(80)는 패킷들을 출력 데이터 트래픽의 흐름으로 출력하기 위해 그 자신의 할당된 큐(52a)를 갖는다. 상기 가상 레인 FIFO(52b)는 관리 트래픽과 관련된 WQE들을 저장하는데 이용된다. 상기 가상 레인 FIFO들(52c, 52d, 52e 및 52f)은 각각의 할당된 가상 레인들과 관련된 WQE들을 저장하는데 이용된다. 비록 상기 개시된 실시예가 4개의 할당된 가상 레인들의 이용을 기술하지만은, 추가적으로 할당된 가상 레인들을 위해 추가적인 가상 레인 FIFO들이 추가될 수도 있다.

<26> 상기 VL 중재 모듈(60)은 레지스트들을 갖는 상태 머신으로서 구현되고, 설정, 관리 및 해제를 포함하는 상기 가상 레인들에 대한 서비스를 위해 상기 VL 중재 테이블(58)을 관리하도록 구성된다. 상기 VL 중재 모듈(60)은 또한 어느 가상 레인을 서비스할 것인지를 결정하고, 상기 가상 레인들의 결정된 우선순위에 근거하여 상기 가상 레인 FIFO들(52)로부터 상기 WQE들을 출력한다. 예를 들면, 상기 가상 레인 FIFO(52b)는 전형적으로 관리(높은 우선순위) 트래픽을 저장하고, 따라서 상기 VL 중재 모듈(60)은 전형적으로 상기 다른 가상 레인 FIFO들(52c, 52d, 52e 또는 52f)을 서비스하기에 앞서 상기 가상 레인 FIFO(52b)를 비운다. 그 다음, 상기 VL 중재 모듈(60)은 상기 VL 중재 테이블(58) 내의 각각의 가중치(weight) 테이블들에 저장된 가중치 우선순위들에 근거하여 상기 가상 레인 FIFO들(52c, 52d, 52e 또는 52f)로부터 상기 WQE들을 선택적으로 출력한다.

<27> 따라서, 상기 사전-링크 모듈(40)은 상기 WQE들의 결정된 우선순위에 근거하여 예를 들면, 할당된 가상 레인들에 근거하거나 또는 상기 WQE가 내장형 프로세스, 관리 트래픽 또는 흐름 제어 트래픽을 위한 것인지에 근거하여 규정된 순서로 상기 WQE들을 출력한다.

<28> 상기 전송 서비스 모듈(42)은 큐 쌍들의 설정, 관리 및 해제를 포함하는 전송 서비스들을 관리하도록 구성된다. 특히, 상기 HCA(12)는 통신 관리 에이전트로부터 수신된 큐 쌍 명령들을 저장하도록 구성되는 큐 쌍 설정 FIFO(62)를 포함한다. 상기 통신 관리 에이전트는 전송 접속들의 설정 및 해제를 담당하고, 상기 통신 관리 에이전트는 서브넷 관리자와 통신을 행하여 상기 HCA(12)에 대한 전송 접속들을 설정한다. 또한, 접속 설정 동안 각각의 끝에서 상기 통신 관리 에이전트들은 통상의 전송 계층 서비스와 달리, (바이패스 서비스 서브모듈(bypass service submodule)(68a)과 관련하여 하기에 설명된) 바이패스 서비스를 이용하여, 상기 전송 접속들을 설정한다.

<29> 상기 전송 서비스 모듈(42)은 큐 쌍 속성들 데이터베이스(64) 및 큐 쌍 속성들 관리 모듈(66)을 포함한다. 상기 큐 쌍 속성들 관리 모듈(66)은 상기 큐 쌍 설정 FIFO(62) 내의 큐 쌍 명령들을 프로세싱하고, 상기 수신된 큐 쌍 명령들에 근거하여 상기 큐 쌍 속성들 데이터베이스(64)를 갱신하도록 구성된다. 예를 들면, 상기 큐 쌍 속성들 데이터베이스(64)는 소스 큐 쌍 번호, 목적지 큐 쌍 번호 및 가능하게는 소스 에이전트 및 목적지 에이전트에 관한 정보를 저장한다. 따라서, 상기 큐 쌍 속성들 데이터베이스(64)는 신뢰성 있는 접속 서비스, 신뢰성 있는 데이터그램 서비스, 신뢰성 없는 접속 서비스, 신뢰성 없는 데이터그램 서비스 및 원시 데이터그램 서비스(raw datagram service)를 포함하는 서로 다른 전송 서비스들을 지원하는데 필요한 모든 정보를 포함할 것이다. 상기 큐 쌍 속성들 데이터베이스(64) 내의 큐 쌍 속성들의 저장에 관한 추가적인 세부사항들은 도 2 및 도 3을 참조하여 하기에 설명된다.

<30> 상기 큐 쌍 속성들 관리 모듈(66)은 로컬 및 원격 통신 에이전트들 간의 통신 동안 예를 들면, 상기 로컬 및 원격 통신 에이전트들 간에 메시지들이 교환됨에 따라 패킷 시퀀스 번호들이 증가할 때, 상기 큐 쌍 속성들 데이터베이스(64)를 갱신함으로써 상기 전송 서비스들을 관리한다.

<31> 상기 큐 쌍 속성들 관리 모듈(66)은 또한 서비스 서브모듈들(68)을 포함하고, 각각의 서비스 서브모듈(68)은 상기 사전-링크 모듈(40)로부터의 대응하는 수신 WQE에 근거하여 대응하는 전송 서비스 타입을 관리하도록 구성된다. 예를 들면, 상기 바이패스 서비스 서브모듈(68a)은 접속 설정 동안 바이패스 서비스들을 관리하거나 또는, 예를 들면, 상기 원시 데이터그램 서비스를 이용하는 네트워크 관리자들에 의한 관리 동작들과 관련된 큐 쌍들을 관리하도록 구성된다. 상기 CPU 보조 서비스 서브모듈(CPU aided service submodule)(68b)은 내장형 가상 레인 FIFO(52a)를 이용하는 내장형 프로세서 동작들에 근거하여 큐 쌍들을 관리하도록 구성되고, 따라서, 상기 CPU 보조 서비스 서브모듈(68b)은 로컬 및 원격 내장형 프로세스들 간의 협력을 가능하게 하고, 또한, 상기 내장형 가상 레인 FIFO(52a)와 결합하여 상기 CPU 보조 서비스 서브모듈(68b)의 구현은 상기 원격 통신 에이전트로부터 재전송 요청이 수신되면 메시지들이 재전송될 수 있게 해준다. 신뢰성 있는 접속(RC) 서비스 서브모듈

(68c) 및 신뢰성 없는 접속(UC) 서비스 모듈(68d)은 신뢰성 있는 접속 및 신뢰성 없는 접속 전송 서비스들 각각과 관련된 큐 쌍들을 관리하도록 구성된다. 비록 도시되지는 않았지만, 상기 큐 쌍 속성들 관리 모듈(66)은 또한 신뢰성 있는 데이터그램 서비스와 신뢰성 없는 데이터그램 서비스 및 원시 데이터그램 서비스를 관리하기 위한 서브모듈들(68)을 포함한다.

<32> 따라서, 사전-링크 모듈(40)로부터 WQE를 수신하면 상기 전송 서비스 모듈(42)은 프로세싱을 위해 상기 WQE를 적절한 서브모듈(68)에 공급한다(예를 들면, 상기 RC 서비스 서브모듈(68c)에 의해 처리되는 RC 서비스에 대한 WQE). 상기 WQE는 서비스 레벨(SL) 정보 및 시스템 메모리(48)에서의 상기 실제 메시지의 위치에 대한 포인터를 포함한다. 상기 서브모듈(68)은 상기 적절한 WQE의 수신에 응답하여, 상기 WQE를 분석하고, 그리고 상기 전송 데이터에 대한 메모리 위치를 식별하는 상기 포인터(즉, 상기 전송 계층에 대한 페이로드)를 상기 WQE로부터 검색하고; 상기 서브모듈(68)은 상기 전송 데이터의 DMA 인출을 수행하고, 상기 큐 쌍 속성들 데이터베이스(64) 내의 적절한 큐 쌍 속성들을 갱신하고, 그리고 대응하는 전송 포맷으로 상기 WQE에 대한 전송 계층 헤더를 생성하여 상기 외부 메모리(48)에 저장하고; 예를 들면, 상기 서브모듈(68a)은 원시 전송 헤더를 발생시킬 수 있고, 한편 상기 모듈들(68c 또는 68d)은 신뢰성 있는 접속 서비스 또는 신뢰성 없는 접속 서비스 각각에 따른 전송 헤더를 발생시킬 수 있다.

<33> 상기 서브모듈(68)은 그 다음에 상기 전송 계층 헤더의 위치를 식별하는 헤더 포인터(P1)를 생성한다. 그 다음, 상기 서브모듈(68)은 상기 사후-링크 모듈(44)에 패킷 요청(90)으로서 페이로드 포인터(P2)와 헤더 포인터(P1)를 전송함으로써, 상기 사후-링크 모듈(44)이 상기 공급된 포인터들에 근거하여 전송하기 위해 전송 패킷을 조립할 수 있게 해준다. 대안적으로, 상기 서브모듈(68)은 전송 계층 프레임(전송 계층 헤더 및 전송 데이터를 포함함)을 저장하는 시스템 메모리 위치에 대한 프레임 포인터를 발생시킬 수 있다. 바람직한 경우, 상기 서브모듈(68)은 또한 상기 사후-링크 모듈에 상기 전송 계층 프레임(전송 계층 헤더 및 전송 데이터를 포함함)을 포워딩할 수 있다. 대안적으로, 상기 외부 메모리에 기입할 때, 상기 CPU는 데이터의 시작부에서 공백을 남겨두어, 상기 모듈들(68) 내에서 생성되는 실제 헤더 정보가 대응하는 빈 메모리 공간에 저장될 수 있게 한다. 상기 사후-링크 모듈(44)에 전해진 포인터는 상기 외부 메모리 내의 상기 프레임의 시작부를 지시하는 포인터일 수 있다.

<34> 상기 사후-링크 모듈(44)은 전송 계층 정보(예를 들면, 전송 계층 프레임, 패킷 요청 등)의 수신에 응답하여, 전송 패킷의 발생 및 전송 FIFO(70)에의 저장을 위해 상기 시스템 메모리(48)로부터 전송 계층 헤더 및 전송 계층 페이로드를 인출한다. 특히, 상기 사후-링크 모듈(44)은 또한 링크 계층 필드들(예를 들면, 로컬 및 글로벌 라우팅 헤더들, 순환 잉여 검사(CRC) 필드들 등)을 발생시킴으로써 상기 전송 패킷을 발생시키고, 상기 전송 FIFO(70)에 상기 전송 패킷을 저장하고, 그리고 상기 InfiniBand™ 아키텍처 명세서에 따라 링크 계층 제어 동작들을 처리하도록 구성된 링크 계층 제어 모듈(72)을 포함한다. 일단 상기 전송 패킷이 발생되면, 상기 포인터들은 하기에 설명된 프리 버퍼 관리자(free buffer manager)(76)에 포워딩된다.

<35> 상기 링크 계층 제어 모듈(72)은 크레디트 기반의(credit-based) 흐름 제어에 따라 상기 전송 패킷들을 출력한다. 특히, 상기 링크 계층 제어 모듈(72)은 할당 가상 레인에서 전송 패킷의 전송을 위해 이용가능한 크레디트들을 모니터한다. 특히, 크레디트들은 가상 레인당 방식으로(on a per virtual lane basis) 전송되고, 여기서 수신기는 유입 가상 레인 버퍼로부터 취해진 패킷들에 근거하여 크레디트를 발행하고, 상기 크레디트들은 전송기에 전송됨으로써, 상기 전송기가 흐름 제어를 관리할 수 있게 해준다. 따라서, 상기 링크 계층 제어 모듈(72)이 식별된 가상 레인이 불충분한 수의 크레디트들을 갖는다고 판단한 경우, 상기 링크 계층 제어 모듈(72)은 충분한 수의 크레디트들이 수신될 때까지 대응하는 전송 패킷의 전송을 연기한다. 상기 가상 레인이 충분한 수의 크레디트들을 갖는 경우, 상기 링크 계층 제어 모듈(72)은 전송을 위해 MAC 모듈(46)에 상기 전송 패킷을 포워딩한다.

<36> 상기 MAC 모듈(46)은 InfiniBand™ 아키텍처 명세서에 따라 전송 FIFO(70)에 저장된 상기 전송 패킷을 출력하도록 구성된다. 특히, 상기 MAC 모듈(46)은 전송 모듈(74), 프리 버퍼 관리자(76), 내장형 프로세서 입력 큐(78) 및 내장형 프로세서(80)를 포함하고, 상기 내장형 프로세서(80)는 링크 흐름 제어 패킷 구성 모듈(82)을 갖는다. 상기 전송 모듈(74)은 InfiniBand™ 네트워크(10)에의 상기 전송 패킷의 전송을 위해 미디어 액세스 제어 동작들, 그리고 선택적으로 물리적 계층 송수신기 동작들을 수행하도록 구성된다.

<37> 상기 프리 버퍼 관리자(76)는 일단 상기 전송 패킷이 응답자에 의해 성공적으로 수신되면 상기 외부 메모리(48)로부터 이용가능한 공간을 해제(release)하도록 구성된다. 특히, 전송 패킷에 대한 상기 메모리 포인터들은

일단 상기 전송 패킷이 발생되면 상기 사후-링크 모듈(44)로부터 전송되고; 만약 응답자가, 상기 전송 패킷이 신뢰성 있는 접속 서비스로 재전송되어야 한다는 메시지를 전송하면, 상기 전송 패킷은 상기 사후-링크 모듈(44)에 의해 재발생되어 상기 응답자에게 재전송될 수 있다. 일단 상기 전송 패킷이 성공적으로 수신되면, 상기 프레임 포인터들은 다른 에이전트에 의해 사용하기 위해 해제될 수 있다.

<38> 흐름 제어는 상기 내장형 프로세서 입력 큐(78)로부터의 정보의 수신에 근거하여 상기 내장형 프로세서(80)에 의해 처리되는데, 특히, InfiniBand™ 아키텍처 명세서에 따른 흐름 제어 프로토콜은 크레디트 기반의 흐름 제어를 이용한다. 상기 내장형 프로세서(80)는 상기 내장형 프로세서 입력 큐(78)에 저장된 메시지에 근거하여, 상기 링크 흐름 제어 패킷 구성 모듈(82)을 이용하여 링크 흐름 제어 패킷들을 발생시킨다. 상기 내장형 프로세서(80)는 외부 메모리(48)에 상기 링크 흐름 제어 패킷을 기입하고, 그 다음에 상기 내장형 프로세서(80)는 관련 동작 및 내장형 프로세서 가상 레인 FIFO(52a) 내의 흐름 제어 패킷의 위치를 특정하는 포인터를 포함하는 WQE를 발생시킨다. 그 다음, 상기 링크 흐름 제어 패킷은 출력될 수 있어, 다른 전송 노드에 대한 이용가능한 크레디트들의 수를 특정한다.

<39> 따라서, 상기 내장형 프로세서(80)는 흐름 제어 헤더를 포함하는 링크 흐름 제어 프레임을 발생시키고, 네트워크에 전송하기 위해 에러 프로세서 입력 큐(78)에 상기 링크 흐름 제어 프레임을 출력할 수 있다.

<40> 도 2는 본 발명의 일 실시예에 따라 큐 쌍 속성들 데이터베이스(64)를 더 상세하게 예시하는 도면이다. 상기 큐 쌍 속성들 데이터베이스(64)는 압축된 큐 쌍 엔트리들(102)을 저장하도록 구성되는 다중의 큐 쌍 테이블들(100)(상기 큐 쌍 속성들 관리 모듈(66)에 의해 설정 및 유지됨)을 포함한다. 특히, 각각의 큐 쌍 테이블(100)은 공유된 속성들 라벨(104)로 예시된, 공유된 속성들의 대응하는 할당된 그룹핑을 갖는다. 예를 들면, 각각의 큐 쌍 테이블(100)은 동일한 전송 서비스를 갖는 큐 쌍들의 저장을 위해 상기 큐 쌍 속성들 관리 모듈에 의해 구성될 수 있고, 여기서, 테이블들(100a, 100b, 100c, 100d 및 100e)은 신뢰성 있는 접속(RC), 신뢰성 있는 데이터그램(RD), 신뢰성 없는 접속(UC), 신뢰성 없는 데이터그램(UD) 및 원시 데이터그램(RWD) 각각에 대한 큐 쌍들을 저장하기 위해 상기 큐 쌍 속성들 관리 모듈(66)에 의해 이용되고, 이러한 예에서, 상기 테이블(100a)은 오직, 신뢰성 있는 접속 전송 서비스를 제공하기 위해 할당된 큐 쌍 엔트리들(102)을 저장한다.

<41> 대안적으로, 각각의 큐 쌍 테이블(100)은 상기 공유된 속성들 라벨(104)로 표시된, 관련된 공유된 속성으로서 동일한 서비스 레벨(SL) 또는 동일한 가상 레인(VL)을 갖는 큐 쌍들의 저장을 위해 구성될 수 있다. 또한, 각각의 큐 쌍 테이블(100)은 다중의 공유값들을 저장하도록 구성될 수 있고, 여기서, 상기 큐 쌍 테이블(100)은 오직, 상기 관련된 공유값들(예를 들면, 신뢰성 있는 접속 및 SL1의 값을 갖는 서비스 레벨, 또는 신뢰성 있는 접속 및 VL1의 값을 갖는 가상 레인)을 제공하기 위해 할당된 큐 쌍 엔트리들(102)을 저장하도록 구성될 수 있다. 가상 레인에 근거하는 큐 쌍들 및 선택된 큐 쌍 테이블들의 저장은 HCA 성능에 영향을 줄 수 있는 프로세싱 자원들을 희생시켜 더 높은 레벨의 압축을 제공하지만, 서비스 레벨에 근거하여 선택된 큐 쌍 테이블들에서 큐 쌍들의 저장은 가상 레인 기반의 압축보다 더 낮은 레벨의 압축을 제공하고, 프로세싱 자원들을 덜 필요로 하게 됨으로써, 상기 HCA가 더 높은 레벨의 성능을 제공할 수 있게 한다.

<42> 또한, 흐름 제어에 이용되는 크레디트들이 가상 레인 기반이기 때문에, 압축된 큐 쌍 엔트리들을 가상 레인에 근거한 큐 쌍 테이블(100)에 삽입하면, 트래픽 흐름의 우선순위를 유지할 수 있다.

<43> 따라서, 도 2에 예시된 각각의 큐 쌍 테이블(100)은 동일한 전송 서비스; 동일한 전송 서비스 및/또는 동일한 서비스 레벨; 또는 동일한 전송 서비스 및/또는 동일한 가상 레인을 갖는 큐 쌍들에 대한 큐 쌍 엔트리들(102)을 저장하도록 구성될 수 있다. 따라서, 상기 큐 쌍 속성들 데이터베이스(64)는 분할된 큐 쌍 테이블들(100)을 이용하는 보다 소수의 물리적 큐 쌍들을 이용하여 다중의 가상 큐 쌍들을 저장할 수 있다. 다른 관련 속성들은 큐 쌍 엔트리(102)에 필요한 정보를 선택된 큐 쌍 테이블로 압축할 때 공유된 속성들에 대해 이용될 수 있음에 주목할 필요가 있다.

<44> 도 3은 본 발명의 일 실시예에 따른 상기 큐 쌍 속성들 데이터베이스(64)에 새로운 압축된 큐 쌍 엔트리를 생성 및 저장하는 방법을 예시하는 도면이다. 상기 방법은 단계(110)에서 시작하며, 단계(110)에서, 상기 큐 쌍 속성들 관리 모듈(66)은 상기 큐 쌍 속성들 데이터베이스(64)를 분할하여, 선택된 공유된 속성들 예를 들면, 전송 서비스의 타입, 서비스 레벨, 가상 레인 등에 근거하여 다중의 큐 쌍 테이블들(100)을 설정한다.

<45> 상기 큐 쌍 테이블들(100)이 설정된 후에, 상기 큐 쌍 속성들 데이터베이스(64)는 새로운 압축된 큐 쌍 엔트리의 저장을 시작할 수 있다. 특히, 상기 큐 쌍 속성들 관리 모듈(66)은 단계(112)에서, 예를 들면, 로컬 또는 원격 통신 관리 에이전트 즉, 서브넷 관리자로부터 수신된 큐 쌍 명령들에 근거하여 새로운 큐 쌍의 생성을 위한

요청을 수신한다.

<46> 상기 큐 쌍 속성들 관리 모듈(66)은 단계(114)에서, 상기 수신된 큐 쌍 명령들로부터 상기 새로운 큐 쌍 속성들을 얻고, 단계(116)에서, 상기 새로운 압축된 큐 쌍 엔트리를 저장하는데 이용될 상기 큐 쌍 테이블(100)을 식별하기 위해, 선택된 새로운 큐 쌍 속성들을 분석한다. 특히, 상기 큐 쌍 속성들 관리 모듈(66)은 단계(118)에서, 상기 새로운 큐 쌍으로부터의 상기 선택된 속성들과 상기 큐 쌍 테이블들(100)의 (각각의 라벨들(104)로 표시된) 각각의 공유된 속성들을 비교하여, 매칭 여부를 판단한다.

<47> 단계(120)에서, 매칭이 검출되지 않으면, 상기 큐 쌍 속성들 관리 모듈(66)은 단계(122)에서, 공유된 속성들로서 상기 선택된 큐 쌍 속성들을 갖는 새로운 큐 쌍 테이블(100)을 생성한다. 그러나, 단계(120)에서, 매칭이 검출되면, 상기 큐 쌍 속성들 관리 모듈(66)은 단계(124)에서, 상기 공유된 속성들을 제외한 상기 새로운 큐 쌍의 필요한 속성들을 포함하는 새로운 압축된 큐 쌍 엔트리(102)를, 상기 매칭 속성들(104)을 갖는 선택된 큐 쌍 테이블(100)에 저장한다. 따라서, 상기 새로운 엔트리(102)는 오직 상기 공유된 속성들(104)과 다른 정보를 저장해야 함으로써, 상기 데이터베이스(64)내의 소수의 물리적 큐 쌍들로부터 다중의 가상 큐 쌍들의 저장을 가능하게 한다.

<48> 본 발명은 현재 가장 실제적인 바람직한 실시예인 것으로 고려되는 것으로 설명되었지만은, 본 발명을 상기 개시된 실시예들로 한정하고자 하는 것이 아니라, 첨부된 청구항들의 정신 및 범위내에 포함된 다양한 수정들 및 등가의 장치들을 포함하고자 하는 것임을 이해해야 한다.

도면의 간단한 설명

<13> 첨부 도면들에 대해 참조가 이루어지고, 여기서, 동일한 참조 부호로 표기된 요소들은 도면 전반에서 동일한 요소들을 나타낸다.

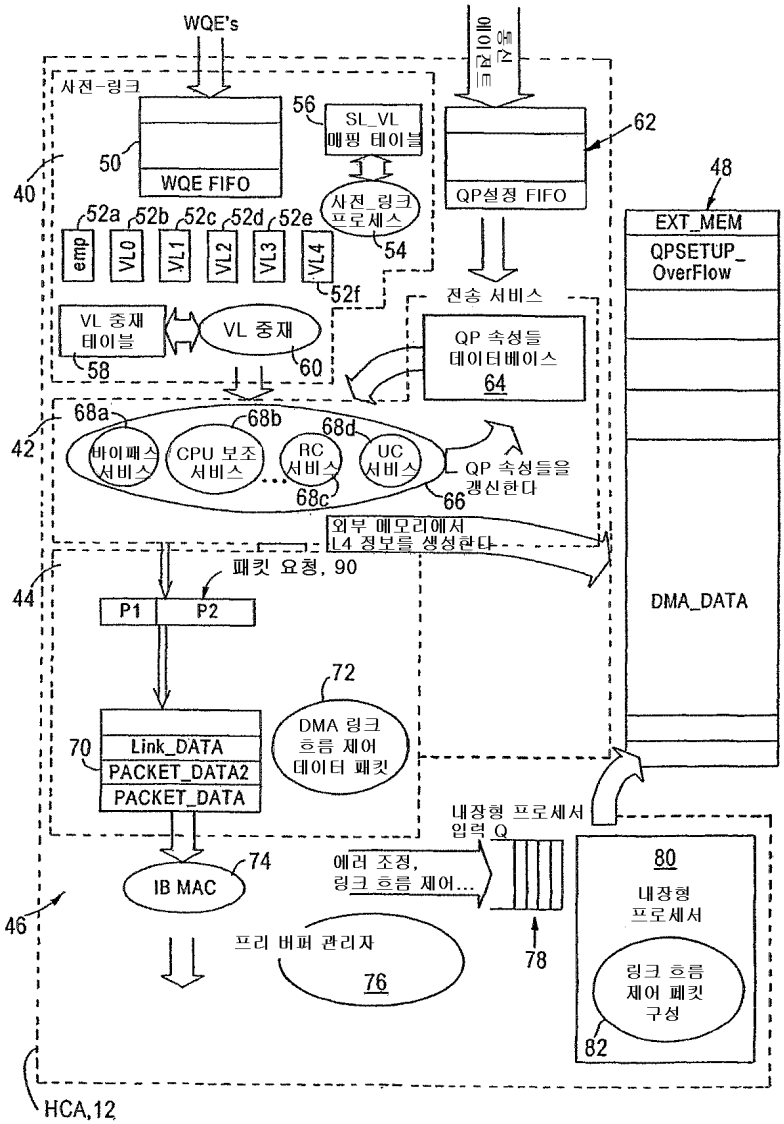
<14> 도 1은 본 발명의 일 실시예에 따라 전송 패킷들을 발생하도록 구성된 호스트 채널 어댑터를 예시하는 도면이고;

<15> 도 2는 본 발명의 일 실시예에 따라, 도 1의 큐 쌍 속성들 데이터베이스를 더 상세하게 예시하는 도면이고; 그리고

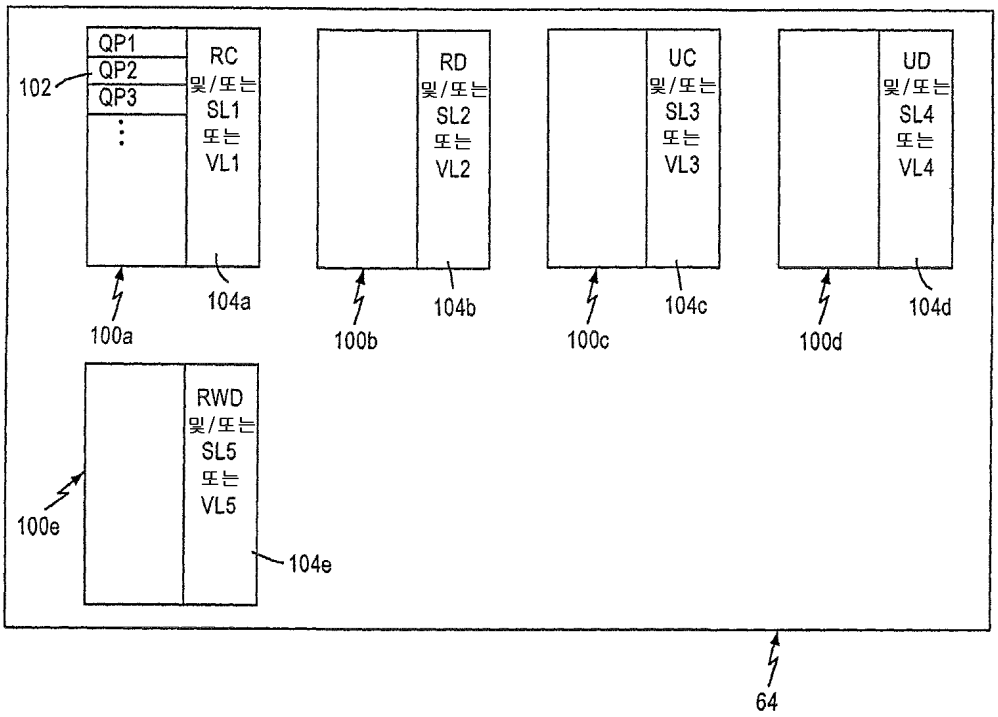
<16> 도 3은 본 발명의 일 실시예에 따라, 새로운 큐 쌍을 상기 큐 쌍 속성들 데이터베이스에 저장하는 방법을 예시하는 도면이다.

도면

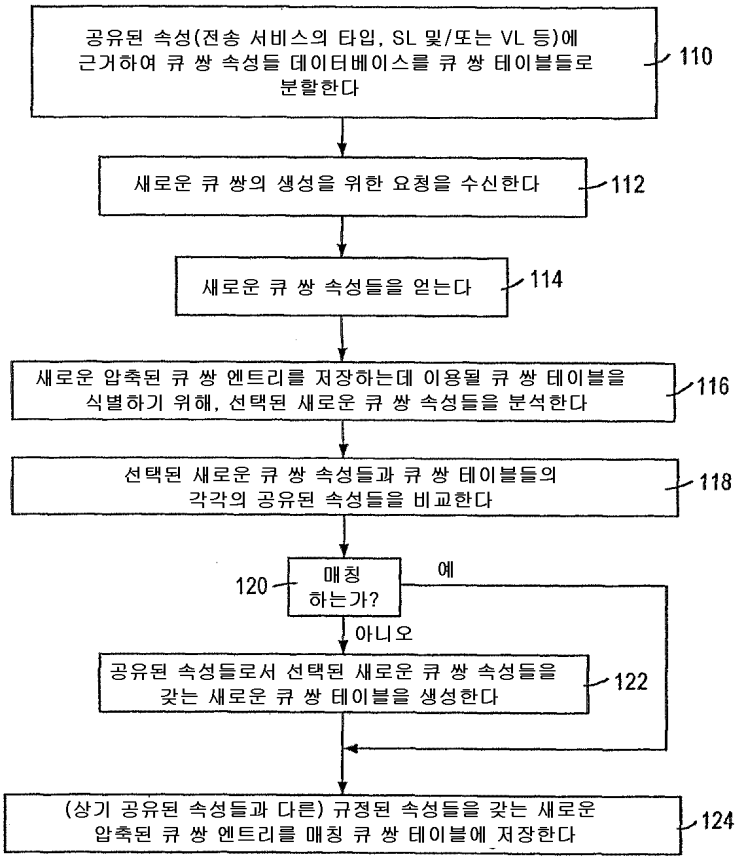
도면1


WQE's
사전-링크
50
WQE FIFO
56
SL_VL
매핑 테이블
40
52a
52b
52c
52d
52e
emp
VL0
VL1
VL2
VL3
VL4
사전 링크
프로세스
54
52f
VL 중재
테이블
58
VL 중재
60
전송 서비스
QP 속성들
데이터베이스
64
62
QP설정 FIFO
48
EXT_MEM
QPSETUP_
OverFlow
DMA_DATA
42
68a
바이패스
서비스
68b
CPU 보조
서비스
68d
RC
서비스
UC
서비스
68c
66
QP 속성들을
갱신한다
외부 메모리에서
L4 정보를 생성한다
44
패킷 요청, 90
P1
P2
72
DMA 링크
흐름 제어
데이터 패킷
70
Link_DATA
PACKET_DATA2
PACKET_DATA
내장형 프로세서
입력 Q
에러 조정,
링크 흐름 제어...
74
IB MAC
80
내장형
프로세서
링크 흐름
제어 패킷
구성
82
46
프리 버퍼 관리자
76
78
HCA,12

도면2


QP1
QP2
QP3
RC
및/또는
SL1
또는
VL1
102
104a
100a
RD
및/또는
SL2
또는
VL2
104b
100b
UC
및/또는
SL3
또는
VL3
104c
100c
UD
및/또는
SL4
또는
VL4
104d
100d
RWD
및/또는
SL5
또는
VL5
104e
100e
64

도면3


공유된 속성(전송 서비스의 타입, SL 및/또는 VL 등)에 근거하여 큐 쌍 속성들 데이터베이스를 큐 쌍 테이블들로 분할한다
110
새로운 큐 쌍의 생성을 위한 요청을 수신한다
112
새로운 큐 쌍 속성들을 얻는다
114
새로운 압축된 큐 쌍 엔트리를 저장하는데 이용될 큐 쌍 테이블을 식별하기 위해, 선택된 새로운 큐 쌍 속성들을 분석한다
116
선택된 새로운 큐 쌍 속성들과 큐 쌍 테이블들의 각각의 공유된 속성들을 비교한다
118
120
매칭 하는가?
예
아니오
공유된 속성들로서 선택된 새로운 큐 쌍 속성들을 갖는 새로운 큐 쌍 테이블을 생성한다
122
(상기 공유된 속성들과 다른) 규정된 속성들을 갖는 새로운 압축된 큐 쌍 엔트리를 매칭 큐 쌍 테이블에 저장한다
124