(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl. [7]
H04L 12/56

(11) 공개번호 특2002-0091202
(43) 공개일자 2002년12월05일

(21) 출원번호 10-2002-7013738
(22) 출원일자 2002년10월12일
번역문 제출일자 2002년10월12일
(86) 국제출원번호 PCT/GB2001/01337
(86) 국제출원출원일자 2001년03월26일
(87) 국제공개번호 WO 2001/79992
(87) 국제공개일자 2001년10월25일

(81) 지정국 국내특허 : 알바니아, 아르메니아, 오스트리아, 오스트레일리아, 아제르바이잔, 보스니아-헤르체고비나, 바베이도스, 불가리아, 브라질, 벨라루스, 캐나다, 스위스, 중국, 쿠바, 체코, 독일, 덴마크, 에스토니아, 스페인, 핀랜드, 영국, 그루지야, 헝가리, 이스라엘, 아이슬란드, 일본, 케냐, 키르기즈, 북한, 대한민국, 카자흐스탄, 세인트루시아, 스리랑카, 라이베리아, 레소토, 리투아니아, 룩셈부르크, 라트비아, 몰도바, 마다가스카르, 마케도니아, 몽고, 말라위, 멕시코, 노르웨이, 뉴질랜드, 슬로베니아, 슬로바키아, 타지키스탄, 투르크메니스탄, 터어키, 트리니다드토바고, 우크라이나, 우간다, 우즈베키스탄, 베트남, 폴란드, 포르투칼, 루마니아, 러시아, 수단, 스웨덴, 싱가포르, 아랍에미리트, 안티구아바부다, 코스타리카, 도미니카연방, 알제리, 모로코, 탄자니아, 남아프리카, 벨리즈, 모잠비크, 가나, 시에라리온, 유고슬라비아, 짐바브웨, 그레나다, 감비아, 크로아티아, 인도네시아, 인도,
AP ARIPO특허: 케냐, 레소토, 말라위, 수단, 스와질랜드, 우간다, 시에라리온, 가나, 감비아, 짐바브웨, 모잠비크, 탄자니아,
EA 유라시아특허: 아르메니아, 아제르바이잔, 벨라루스, 키르기즈, 카자흐스탄, 몰도바, 러시아, 타지키스탄, 투르크메니스탄,
EP 유럽특허: 오스트리아, 벨기에, 스위스, 독일, 덴마크, 스페인, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투칼, 스웨덴, 핀랜드, 사이프러스, 터어키,
OA OAPI특허: 부르키나파소, 베넹, 중앙아프리카, 콩고, 코트디브와르, 카메룬, 가봉, 기네, 말리, 모리타니, 니제르, 세네갈, 차드, 토고, 기네비쏘,

(30) 우선권주장
09/548,913 2000년04월13일 미국(US)
09/548,912 2000년04월13일 미국(US)
09/548,910 2000년04월13일 미국(US)

(71) 출원인 인터내셔널 비지네스 머신즈 코포레이션
미국 10504 뉴욕주 아몬크

(72) 발명자 배스브라이언미첼
미국노스캐롤라이나주27502에이펙스올드스터브리지드라이브4021
칼비그낙진루이스
미국노스캐롤라이나주27511캐리스프링할로우레인112
헤데스마르코
미국노스캐롤라이나주27612롤리그랜드매너코트#3084109
시젤마이클스티븐
미국노스캐롤라이나주27615롤리라우어리드라이브10625
베르플랑켄파브리스장
프랑스에프-06610라고드라우트데카스네스9152

(74) 대리인 안성탁
이광현
신정건

심사청구 : 없음

(54) 네트워크 프로세서가 흐름 큐의 연결 해제/재연결을 통해출력을 스케줄링하는 방법 및 시스템

요약

본 발명은 정보 단위들을 우선 순위화된 순서로 네트워크 프로세서에서 몇몇 상이한 레벨의 서비스를 수용하는 데이터 전송 네트워크로 이동시키는 방법 및 시스템에 관한 것이다. 본 발명은 정보 단위들의 다양한 소스들과 관련된 저장된 우선 순위에 따라 네트워크 처리 유닛으로부터의 처리 정보 단위들(또는 프레임들)의 이그레스를 스케줄링하는 방법 및 시스템을 포함한다. 양호한 실시예에서의 우선 순위는 저 레이턴시 서비스, 최소 대역폭, 가중 공정 큐잉 및 사용자가 장기간에 걸쳐 자신의 서비스 레벨을 계속해서 초과하는 것을 방지하는 시스템을 포함한다. 본 발명은 사용자로 하여금 희망하는 서비스 속도를 선택하게 하기 위해서 상이한 서비스 속도를 갖는 복수의 캘린더를 포함한다. 고객이 높은 대역폭의 서비스를 선택하는 경우에는, 그 고객은 더 낮은 대역폭을 선택했을 경우보다 더 자주 서비스되는 캘린더에 포함될 것이다.

대표도

도 1

명세서

기술분야

본 발명은 다양한 능력을 가진 각종 정보 처리 시스템이나 컴퓨터와 함께 링크하는 데 사용되는 통신 네트워크 장치와, 이러한 장치에서 데이터를 처리하는 컴포넌트 및 방법에 관한 것이다. 본 발명은 정보 단위들을 복수의 네트워크 처리 유닛에 결합된 흐름 제어 시스템에서 데이터 전송 네트워크로 MAC를 통해 분배하는 일을 스케줄링하는 개량된 시스템 및 방법을 포함한다. 더욱 상세히 말하자면, 본 발명은 가변 크기의 정보 패킷 또는 프레임을 처리하고 있는 복수의 사용자를 취급할 수 있는 스케줄링 기능과, 상이한 사용자들에게 부여된 복수의 상이한 우선 순위를 참작하면서 프레임을 흐름 제어 시스템에서 데이터 전송 네트워크로 제공하는 순서를 규정하는 기능을 포함한다.

배경기술

이어지는 본 발명의 설명은 판독기가 네트워크 데이터 통신과 이러한 네트워크 데이터 통신에 유용한 라우터 및 스위치에 대한 기본 지식을 가지고 있다는 전제 조건을 기초로 한다. 특히, 본 발명의 설명은 네트워크 동작을 여러 계층으로 나눈 네트워크 구조에 대한 국제 표준화 기구(ISO; International Standards Organization) 모델에 정통하다는 것을 전제로 한다. ISO 모델을 기반으로 한 통상적인 구조는 신호를 상위 계층으로 전달하는 물리적 경로 또는 매체인 계층 1(" L1" 이라고도 불림), 계층 2(또는 " L2"), 계층 3(또는 " L3"), ... , 네트워크에 링크된 컴퓨터 시스템에 상주한 애플리케이션 프로그래밍 계층인 계층 7로 구성된다. 본 명세서에서, L1, L2, L3과 같은 계층들에 대한 참조는 상기 네트워크 구조의 대응하는 계층을 나타낸다. 본 발명의 설명은 또한 패킷 및 프레임으로 알려진, 네트워크 통신에서 사용되는 비트열(bit string)에 대한 기본적인 지식을 가지고 있다는 전제 조건을 기초로 한다.

오늘날, 대역폭(또는 시스템이 단위 시간에 처리할 수 있는 데이터량)은 네트워크 동작에 있어서 중요한 고려 사항으로 부각되고 있다. 네트워크 상의 트래픽은 현저한 양으로 다양하게 증가되고 있다. 이전에, 일부 네트워크들은 전화 네트워크 상의 음성, 데이터 전송 네트워크 상의 디지털 데이터 등과 같이, 주로 일정한 종류의 통신 트래픽만을 전송하였다. 물론, 전화 네트워크는 음성 신호 이외에도, 제한된 양의 " 데이터" (예컨대, 라우팅 및 과금 목적을 위한 발신 번호 및 착신 번호)를 전송하였지만, 일부 네트워크들은 어느 때에 실질적으로 동종의 패킷들을 전송하여 왔다.

실질적인 트래픽 증가는 인터넷(월드 와이드 웹 또는 " www" 라고도 불리는, 짜임새 없이 링크된 컴퓨터들의 공중 네트워크)과, 사설 데이터 전송 네트워크로 보급된, 인터넷과 유사한 내부 네트워크(인트라넷이라고도 불림)에 대한 인기의 증가에 기인한 것이다. 인터넷과 인트라넷은 정보 및 최신 애플리케이션에의 원격 접속에 대한 날로 증가하는 요구를 만족시키기 위해서 다량의 정보를 원격지들간에 전송하는 기능을 포함한다. 인터넷은 지리적으로 흩어진 지역에 있는 다수의 사용자에게까지 폭발적으로 증가한 양의 원격 정보를 개방하고 있고, 전자 상거래와 같은 각종 새로운 애플리케이션을 가능하게 하고 있는데. 그 결과 네트워크의 부하가 상당히 증가되고 있다. 그 밖의 다른 애플리케이션들, 예컨대 이메일, 파일 전송 및 데이터베이스 접속 또한 네트워크의 부하를 증가시키고 있어, 일부 네트워크들은 고레벨의 네트워크 트래픽으로 인해 이미 과부하 상태에 놓여 있다.

음성 및 데이터 트래픽 또한 요즈음에는 네트워크 상으로 집중되고 있다. 데이터는 현재 인터넷 상에서 (인터넷 프로토콜, 즉 IP를 통해) 무료로 전송되고, 음성 트래픽은 통상적으로 최저 비용의 경로를 따라간다. VoIP[voice over IP(internet protocol)], VoATM[voice over ATM(asynchronous transfer mode)], VoFR(voice over frame relay) 등과 같은 기술들은 오늘날의 환경에서 음성 트래픽 전송에 대한 비용 효율적인 대체 기술이다. 이러한 서비스들은 새로 변하기 때문에, 산업계는 변경 비용 구조 등의 문제들과, 프로세서들간의 정보 전송에 있어서 서비스 비용과 서비스 품질간의 타협(trade off)에 관한 걱정들을 처리할 것이다.

서비스 품질면에는 용량 또는 대역폭(일정 기간에 수용될 수 있는 정보량), 응답 시간(한 프레임을 처리하는 데 걸리는 시간) 및 처리 유연성(상이한 캡슐화 또는 프레임 헤더 방법과 같이, 상이한 프로토콜 및 프레임 구성에 대한 응답)이 포함된다. 자원을 사용하는 사람들은 서비스 품질은 물론이고 서비스 비용을 고려할 것이며, 이들의 타협은 제시된 상황에 따라 달라진다. 사용자에게 각종 상이한 우선 순위 또는 스케줄링 알고리즘을 제공하여, 사용자가 보장된 대역폭(quaranteed bandwidth) 방식, 최선의 노력(best efforts) 방식, 피크를 위한 최선의 노력을 갖는 보장된 대역폭(guaranteed bandwidth with best efforts for peak) 방식 중에서 원하는 방식을 결정(결정한 방식에 대한 요금을 지불)하게 하는 것이 바람직하다. 또한, 대역폭 할당 시스템은 사용자가 선택 및 지불한 대역폭을 초과하는 용량을 요구할 경우에 이러한 요구를 거절함으로써, 사용자가 선택한 우선 순위 및 대역폭을 강요하는 시스템을 구비하는 것이 바

람직하다.

종래 기술의 일부 시스템들은 처리 시스템에서 출력되는 정보 단위들을 다양한 방법으로 처리한다. 그 중 한 방법은 큐들에게 공정한 기회를 부여하는 라운드 로빈 스케줄러를 사용하는 것이다. 다른 방법은 몇몇 상이한 레벨의 우선 순위와 각각에 대한 큐를 채택하는 것이다. 이러한 시스템의 경우, 사용자는 절대 우선 순위를 갖게 되는데, 여기서 최고 우선 순위의 작업은 먼저 처리되지만 최저 우선 순위의 작업은 처리되지 않을 수도 있다. 또 다른 출력 스케줄링 방법은 복수의 우선 순위화된 목록을 포함하는 것이다. 또한, 계층적인 패킷 스케줄링 시스템을 사용하는 방법도 알려져 있다. 심지어, 몇몇 상이한 스케줄링 방법을 사용하는 시스템도 있는데, 이러한 시스템의 경우, 정보 단위들이 데이터 전송 네트워크로 전송되는 순서는 상이한 스케줄링 기술들의 조합을 사용해서 결정한다.

다른 시스템들은 라운드 로빈 방식으로 구현된 가중 우선 순위 기술을 사용하며, 이것은 서비스 레벨을 규정하는 알고리즘을 기초로 해서 모든 큐들을 서브하는데, 일부 큐들은 다른 큐들에 비해 더 자주 서브된다. 심지어 이러한 가중 우선 순위 시스템도 할당된 서비스 레벨을 계속해서 초과하는 사용자에게 서비스를 제공하기도 하는데, 이것은 자주는 아니지만, 할당된 서비스 레벨을 초과할 때 마다 계속해서 서브하여, 시스템이 서비스 레벨 정책을 실행하는 것을 어렵게 만든다.

서브할 고객을 결정함에 있어서 패킷 또는 프레임의 크기를 고려함으로써 적정한 공정성을 서비스 시스템에 더할 수 있는데, 이것은 큰 프레임을 처리하고 있는 사용자는 시스템 용량의 더 많은 부분을 차지하므로 작은 프레임을 처리하는 사용자보다 적게 서비스를 받아야 한다는 점에서 그러하다. 종래 기술의 시스템들 중 일부는 자원을 할당함에 있어서 전송하는 패킷 또는 프레임의 크기를 고려하지만, 다른 시스템들은 그렇지 않다. 일부 통신 시스템들은 일정한 고정 크기의 패킷을 사용함으로써 패킷의 크기를 고려하는 것을 불필요하게 하지만, 다른 시스템들은 자원을 할당함에 있어서 패킷의 크기를 고려하지 않는다.

종래 기술의 다른 시스템들은 소위 비동기 전송 모드(ATM) 시스템에서와 같이 공통의 크기를 갖는 정보 단위들을 처리함으로써, 현재 또는 미래의 정보 단위의 우선 순위를 결정함에 있어서 정보 단위의 크기를 고려하지 않는다. 가중 구동 스케줄러를 가진 ATM 시스템은 ATM 시스템에서의 출력을 스케줄링하기 위한 공지된 해결책들 중 하나이다.

어떤 그러한 시스템에 있어서는, 프레임 크기에 상관없이 일정한 보장된 대역폭을 고객에게 할당하는 것과 같은 시스템 제약을 수용하는 동시에, 다음의 추가적인 특징, 즉 시스템의 파라미터가 상당히 계속 초과하는 것을 방지하면서 보장된 대역폭을 초과하는 피크 입력을 수용하고, 게다가 출력을 데이터 전송 네트워크로 제공함에 있어서 네트워크 프로세서의 용량을 효율적이고 공정하게 사용하는 특징을 제공하기 위한 매커니즘을 제공하는 것이 바람직하다.

원하는 바와 같은 상이한 서비스 타입 및 레벨을 허용하는 최대 유연성을 갖는 시스템을 제공하는 것이 바람직하다. 예컨대, 일부 사용자들은 최소 대역폭을 원하고, 다른 사용자들은 버스트를 허용하는 경우 외에는 최소 대역폭을 원하며, 또 다른 사용자들은 최소 대역폭의 존재 여부에 관계없이 " 최선의 노력(best effort)" 서비스를 제공하는 경제적인 서비스에 관심을 가질 수도 있는데, 스케줄링 시스템은 대역폭 또는 버스트 크기에 일정한 제한을 가할 수 있어야 한다. 스케줄링 시스템은 가변 패킷 길이를 수용하고 비사용 대역폭을 할당하는 가중 공정 큐잉 시스템을 제공하는 간단하고 효율적인 시스템의 특징들 중 일부 또는 모두를 갖는 것이 바람직하나, 불행히도, 그러한 시스템들은 종래 기술에 존재하지 않는다.

또한, 흐름 큐가 연결 해제되었다가 재연결되는 경우에, 그 큐에서는 그러한 일이 없었을 경우에 가질 수 있었던 것보다 더 좋은 우선 순위 또는 자리를 얻을 수 없는 것이 바람직하다.

또한, 스케줄을 계산함에 있어서, 서비스된 후 소정의 흐름에 대한 스케줄에서 새로운 순서에 대한 복잡한 계산과 관련된 하드웨어 비용을 피할 수 있도록 간단하고 효율적인 시스템으로 구현되는 것이 바람직하다.

이와 같이, 네트워크로 전송하기 위한 데이터 패킷을 처리하는 종래 기술의 시스템은 시스템의 융통성이나 그 동작 속도에 영향을 미치는 바람직하지 않은 불리한 조건과 제한 사항을 가지고 있다.

발명의 상세한 설명

본 발명은 종래 기술의 시스템들의 불리한 조건 및 제한 사항을 극복하기 위한 것으로서, 처리 시스템을 나오는 정보 단위들 또는 프레임들을 처리하고, 프레임들을 출력 포트로 보내서 데이터 전송 네트워크로 발송하는 간단하나 효과적인 방법을 제공한다. 본 발명은 복수의 사용자로부터 가변 길이의 패킷들이 처리되고 있는 시스템과, 일정 레벨의 서비스 계약이 사용자들 중 적어도 일부와 이루어진 경우에 특히 적용된다.

본 발명은 시스템의 사용자들에게 다양한 종류의 서비스 레벨을 제공하는 것을 가능하게 한다. 즉, 최소 보장된 대역폭이 어떤 한 사용자에게 제공되는 동시에 다른 사용자들은 공유 대역폭을 즐길 수 있으며, 피크 대역폭이 제한된 시간 동안 허용될 수 있고 최대 버스트 레벨 서비스가 사용자에게 제공될 수 있으며, 모든 것이 다른 사용자들에게 제공되는 서비스와 충돌하는 일이 없이 프로그램된다.

본 발명은 대역폭 자원의 효율적인 사용을 가능하게 하고, 서비스 레벨 계약을 이행하는 동시에 남은 대역폭을 효율적이고 공정하게 사용하는 것을 가능하게 하는 이점을 가지고 있다.

본 발명은 또한 버스트를 수용하면서 자원 할당을 실행할 수 있는 이점을 가지고 있다. 즉, 사용자는 제한된 기간 동안에 자신의 규정된 대역폭을 초과하는 버스트 속도로 트래픽의 일부를 전송할 수 있으나, 이러한 전송이 장기간에 걸쳐 이루어지는 경우에는 초과 사용에 대한 보상이 이루어질 때까지는 자신의 규정된 대역폭을 초과하여 전송하는 것이 금지된다. 이것은 사용자가 자신의 대역폭을 최대로 사용하지 않았을 때 비축되는, 각각의 사용자를 위한 흐름 큐에 대한 "크레디트(credit)" 시스템을 통해 달성된다.

본 발명은 나누기 연산이 없는 간단한 계산을 가능하게 하여, 서비스 흐름에 대한 큐 내의 새로운 위치를 계산하는 것을 지원한다.

본 발명은 또한 연결 해제 및 재연결시, 소정의 흐름이 그 연결 해제 덕으로 좋은 자리를 얻는 것을 방지할 수 있는 이점을 가지고 있다.

본 발명은 슬롯 거리를 사용하여 프레임의 크기 및 큐 가중치를 조정함으로써, 다음 자리를 신속하고 쉽게 계산하는 것을 가능하게 한다.

본 발명은 또한 버스트 사용의 실행 스코어를 유지하고 사용 가능한 " 크레디트" 를 계산 및 유지함으로써, 사용에 대한 시스템 제약을 강요하는 것을 제외하고는 당연히 버스트를 허용한다.

본 발명은 또한 소정의 타임 슬롯 또는 사이클 동안 스케줄링된 가장 늦은 서비스 요청에게 우선 순위를 주는 푸시 다운 스택을 제공하는 이점을 가지고 있다. 이것은 나중에 스케줄링된 사용자가 높은 우선 순위를 가질 것과, 그 사용자를 서브할 때의 소정의 지연이 낮은 우선 순위의 사용자를 서브할 때의 유사한 지연보다 더 큰 비율의 지연일 것을 전제로 한다. 슬롯 요청 이행에 후입선출(LIFO; last-in-first-out) 시스템을 사용함으로써, 시스템으로 하여금 시스템 과부하시 연속되는 서비스간의 정상 간격의 비율에 따라 서비스시 지각되는 지연을 최소화하게 하는 것을 가능하게 한다. 즉 스케줄링된 시간에서 처리될 수 있는 것보다 더 많은 작업이 가능하다.

도면의 간단한 설명

전술한 종래 기술의 불리한 조건 및 제한 사항 중 일부와, 본 발명의 목적 및 이점 중 일부와, 그리고 다른 목적 및 이점은 본 발명의 개선된 라우팅 시스템 및 방법을 나타내고 있는 이어지는 도면의 간단한 설명을 통해 당업자에게 명백해질 것이다.

도 1은 본 발명을 실시하는 데 유용한 DN 인큐 시스템 및 스케줄러를 보여주면서, 내장형 프로세서 컴플렉스를 포함하는 인터페이스 장치의 블록도이다.

도 2는 본 발명을 이해하는 데 유용한 DN 인큐(및 그 포함된 스케줄러)와 함께, 도 1에 도시된 타입의 내장형 프로세서 컴플렉스의 블록도이다.

도 3은 본 발명의 양호한 실시예에 따른 도 2의 스케줄러에서 가변 길이 패킷을 스케줄링하는 시스템을 나타낸다.

도 4는 본 발명의 양호한 실시예에 따른 도 3의 스케줄링 시스템에서 사용되는 타이머 기반 캘린더를 나타낸다.

도 5는 도 3 및 도 4의 스케줄러와 함께 사용되는 스케줄링 동작의 흐름도를 나타낸다.

도 6은 도 4의 에포크들이 어느 정도 상이한 시간 분해능을 갖는지를 보여주는 도면을 나타낸다.

도 7은 본 발명에 사용되는 최대 버스트 사양을 나타낸다.

도 8 내지 도 13은 본 발명의 스케줄러의 다양한 흐름도인데, 도 8 및 도 9는 다음 그린 시간을 계산하는 흐름도이고, 도 10 및 도 11은 버스트 크기 크레디트를 계산 및 갱신하는 흐름도이며, 도 12 및 도 13은 연결 해제 및 재연결로 인해 이익을 얻는 것을 방지하기 위한 큐 제어 블록의 노화를 나타낸다.

도 14는 본 발명의 스케줄러에 유용하고, 본 발명의 양호한 실시예에 따른 WFQ 캘린더를 나타낸다.

도 15는 도 3 및 도 4의 스케줄러와 함께 사용되는 스케줄링 동작 논리의 흐름도를 나타낸다.

실시예

이어지는 양호한 실시예의 설명에 있어서, 현재 발명자가 알고 있는 본 발명을 실시하는 최상의 방법을 매우 상세히 설명할 것이다. 그러나, 이러한 설명은 특정 실시예에서 본 발명의 개념을 광범위하고 일반적으로 교시하기 위한 것이지, 본 발명을 이 실시예로 제한하기 위한 것이 아니기 때문에, 특히 당업자라면 도면에 대해서 설명된 특정 구조 및 동작에 대한 많은 변형 및 변화가 이루어질 수 있음을 인정할 것이다.

도 1은 기판(10)과 그 기판 상에 집적된 복수의 하위 부품을 포함하는 인터페이스 장치 칩의 블록도이다. 하위 부품들은 상향 구성부와 하향 구성부로 배열되는데, " 상향" (" 인그레스" 라고도 불림) 구성부는 데이터 전송 네트워크에서 칩으로 인바운드(inbound)되는 데이터에 관련된 부품들에 관한 것이고, " 하향" (" 이그레스" 라고도 불림) 구성부는 아웃바운드(outbound) 방식(칩을 떠나 네트워크로)으로 데이터를 칩에서 데이터 전송 네트워크로 전송하는 기능을 가진 부품들에 관한 것이다. 데이터 흐름은 각각의 배열된 상향 및 하향 구성부를 따르므로, 도 1의 시스템에는 상향 데이터 흐름과 하향 데이터 흐름이 있다. 상향 또는 인그레스 구성부는 인큐-디큐-스케줄링 UP(EDS-UP) 로직(16), 다수의 다중화 MAC-UP(PMM-UP)(14), 스위치 데이터 무버-UP(SDM-UP)(18), 스위치 인터페이스(SIF)(20), 데이터 정렬 직렬 링크 A(DASL-A)(22) 및 데이터 정렬 직렬 링크 B(DASL-B)(24)를 포함한다. 본 발명의 양호한 실시예에서는 데이터 링크들이 사용되고, 다른 시스템들은 본 발명에 유리하게 사용될 수 있으며, 특히 이것들은 비교적 높은 데이터 흐름과 시스템 요구사항을 지원하는데, 그 이유는 본 발명이 양호한 실시예에서 사용되는 데이터 링크와 같은 특정 보조 장치로 제한되지 않기 때문이다.

시스템의 하향(또는 이그레스) 구성부는 데이터 링크인 DASL-A(26) 및 DASL-B(28), 스위치 인터페이스(SIF)(30), 스위치 데이터 무버-DN(SDM-DN)(32), 인큐-디큐-스케줄러(EDS-DN)(34) 및 다수의 이그레스용 다중화 MAC(PMM-DN)(36)을 포함한다. 기판(10)은 또한 복수의 내부 S-RAM, 트래픽 관리 스케줄러(이그레스 스케줄러라고도 알려짐)(40) 및 내장형 프로세서 컴플렉스(12)를 포함한다. PMM(14, 36)에는 각각의 DMU 버스를 통해 인터페이스 장치(38)가 결합된다. 인터페이스 장치(38)는 이더넷 물리적 장치(ENET PHT)나 비동기 전송 모드 프레임 장치(ATM FRAMER) - 양자 모두 잘 알려져 있고, 보통 상업적으로 입수 가능한 장치들의 예이다 - 와 같은 L1 회로에 연결하기에 적합한 어떤 하드웨어 장치일 수 있다. 인터페이스 장치의 타입 및 크기는 적어도 일부는 본 발명의 칩과 그 시스템이 결합되는 네트워크 매체에 의해 결정된다. 본 발명의 칩은 복수의 외부 D-RAM과 하나의 S-RAM을 사용할 수 있다.

본 명세서에는 특히 관련 스위칭 및 라우팅 장치들 밖으로의 일반적인 데이터 흐름이 건물에 설치된 와이어 및 케이블과 같은 전기 전도체를 통해 흐르는 네트워크가 개시되어 있지만, 그 네트워크 스위치들 및 부품들은 또한 무선 환경에서도 사용될 수 있다는 것을 고려해야 한다. 예컨대, 본 명세서에 개시된 매체 접속 제어(MAC) 부품들은 실리콘 게르마늄 기술로 만들어진 것과 같은 적절한 무선 주파수 장치들로 대체될 수 있으며, 따라서 그 개시된 장치는 무선 네트워크에 직접 연결될 수 있다. 이러한 기술이 적절히 사용되는 경우, 무선 주파수 장치들은 당업자에 의해 본 명세서에 기재된 VLSI 구조에 통합될 수 있다. 대안적으로, 무선 주파수 또는 다른 무선 응답 장치들, 예컨대 적외선(IR) 응답 장치들은 본 명세서에 개시된 다른 부품들과 함께 블레이드 상에 탑재되어, 무선 네트워크 장치에 유용한 스위치 장치를 구성할 수 있다.

화살표는 도 1에 도시된 인터페이스 시스템 내에서의 일반적인 데이터 흐름을 보여준다. ENET PHY 블록(38)을 떠나 DMU 버스를 경유해서 이더넷 MAC(14)로부터 수신된 데이터 또는 메시지 프레임(패킷 또는 정보 단위라고도 불림)은 EDS-UP 장치(16)에 의해 내부 데이터 저장 버퍼(16a)에 배치된다. 프레임은 정규 프레임이나 안내 프레임으로 식별될 수 있는데, 이것은 복수의 프로세서에서의 후속 처리 방법 및 위치에 관련된 것이다. 입력된 정보 단위 또는 프레임은 내장형 프로세서 컴플렉스 내의 복수의 프로세서 중 어느 한 프로세서에서 처리되며, 그 후, 처리 완료된 정보 단위는 스위치로 전송되어 네트워크 프로세서의 인그레스측으로 전달된다. 정보 단위가 네트워크 프로세서의 인그레스측에 수신되고 나면, 내장형 프로세서 컴플렉스 내의 복수의 프로세서 중 어느 한 프로세서에서 처리되며, 이그레스 처리가 완료되면, 처리 유닛(10) 중 트래픽 관리 스케줄러(40)를 통해 스케줄링된 후 PMM-DN 다중화 MAC(36)와 물리적 계층(38)을 통해 데이터 전송 네트워크로 전송된다.

도 2는 본 발명에 유리하게 사용될 수 있는 처리 시스템(100)의 블록도이다. 도 2에 있어서, 복수의 처리 유닛(110)은 디스패처(dispatcher) 유닛(112)과 완성(completion) 유닛(120) 사이에 위치한다. (도시되지는 않았지만, 본 발명의 데이터 처리 시스템에 결합된 스위치로부터의) 각각의 이그레스 프레임(F)은 하향 데이터 저장소(DN DS; DOWN data store)(116)에 수신 및 저장된 후, 디스패처 유닛(112)에 의해 순차적으로 제거되어, 그 프레임을 처리하는 데 사용 가능한 처리 유닛에 대한 디스패처 유닛(112)의 결정을 기초로 해서, 복수의 처리 유닛(110) 중 어느 한 처리 유닛에 할당된다. 디스패처 유닛(112)과 복수의 처리 유닛(110) 사이에는 하드웨어 분류기(118)가 개재된다. 복수의 네트워크 처리 유닛(110)에 의해 처리된 프레임은 흐름 제어 시스템을 통해 DN 인큐(34)에 결합된 완성 유닛(120)으로 전송된다. DN 인큐(34)는 PMM-DN MAC(36)과 DMU 데이터 버스를 통해 물리적 계층(38)(데이터 전송 네트워크)에 결합된다.

도 3의 스케줄러(40)는 단일 통합 스케줄러 시스템에서 스케줄링한 최소 대역폭 알고리즘, 피크 대역폭 알고리즘, 가중 공정 큐잉 기술 및 최대 버스트 크기에 따라, 네트워크 처리 유닛에서 데이터 전송 네트워크로의 프레임 전송을 스케줄링하는 기능을 허용하는 동작의 구조 및 방법을 제공한다.

시간 기반 캘린더는 최소 대역폭 및 최선의 노력(best effort) 피크 속도 요구 사항으로 패킷을 스케줄링하는 데 사용된다. 도 3에 나타낸 바와 같이, 3개의 시간 기반 캘린더가 상기와 같은 목적으로 사용되는데, 그 중 2개는 최소 대역폭용이고, 나머지 1개는 흐름 큐를 최대 최선의 노력(best effort) 피크 속도로 제한(피크 대역폭 실현)하는 데 사용된다. 최소 대역폭용으로 제공된 2개의 캘린더(LLS, NLS)는 최소 대역폭 QoS 등급 내에서 상이한 서비스 등급(즉, 저 레이턴시 및 정상 레이턴시)을 지원할 수 있도록 해준다.

전술한 캘린더에 있어서, 포인터는 흐름 큐의 캘린더 내 위치를 나타내는 데 사용된다. 또한, 시스템의 복수의 캘린더 내에 존재하는 단일 흐름 큐에 대해서 포인터가 없거나, 1개이거나, 2개일 수 있다. 통상적으로, 그렇지 않은 캘린더 내의 포인터들은 초기화되지 않았거나 비어 있지 않은 흐름 큐들을 나타낸다. 어느 한 흐름 큐에 대한 포인터가 시스템의 어느 한 캘린더 내에 존재하는 경우, 그 흐름 큐는 그 캘린더 " 내에" 존재한다고 말할 수 있다.

한 시간 주기는 한 스케줄러_틱(scheduler_tick)으로서 정의된다. 각 스케줄러_틱 동안에 한 단위의 대역폭이 서비스될 수 있다. 양호한 실시예에서는, 이러한 단위를 " 스텝(step)" 으로서 정의하며, 이것은 바이트 당(즉, 1/대역폭) 시간 단위들을 갖는다.

통신 시스템에 있어서, 각 흐름 큐에 대한 최소 대역폭 규격의 범위는 대략 몇 가지 등급의 크기로 분류된다. 즉, 일부 사용자들(또는 실제, 그 사용자들과 관련된 큐들)은 다량의 데이터를 전송하고 있기 때문에 고대역폭을 가질 것이고 따라서 그러한 대역폭에 대한 요금을 지불하고 있으며, 다른 사용자들은 언제라도 소량의 정보(대역폭)를 전송할 수 있는 이코노미를 선택하고 있다. 이러한 목적에만 전용적으로 사용되는 하드웨어량을 최소화하기 위해서, 양호한 실시예에서는 통신 시스템 및 서비스 레벨 계약(SLA)에 의해 요구되는 범위 및 정확도를 유지하면서 하드웨어 사용량을 줄일 수 있는 스케일링 기술이 사용된다.

양호한 일실시예에 있어서, 도 4에 나타낸 바와 같이, 각각의 타이머 기반 캘린더는 4개의 " 에포크(epoch)" 로 이루어진다. 각 에포크는 512개의 " 슬롯(slot)" 으로 이루어진다. 각 슬롯은 흐름 큐들에 대한 LIFO 포인터 스택을 포함한다. 어떤 2개의 슬롯간의 거리는 대역폭 측정량이고, 그 값은 에포크에 좌우된다. 양호한 실시예에 있어서는, 도 4에 나타낸 바와 같이, 각 에포크간에는 16의 스케일링 인수가 존재한다. 양호한 실시예에 있어서, 150 ns의 스케줄러_틱 지속 시간이 선택된 경우, 에포크 0에서 1개의 슬롯의 폭은 150 ns에 이동된 512 바이트의 대역폭 또는 대략 27 Gb/s를 나타내고, 한편 에포크 3에서 1개의 슬롯의 폭은 0.614 ms에 이동된 512 바이트의 대역폭 또는 대략 6.67 Mb/s를 나타낸다.

에포크 당 슬롯의 수 및 실시예에서 사용되는 에포크의 수는 하드웨어 비용과 설계 복잡성간에 절충되며, 본 발명의 범위를 제한하려는 것은 아니다. 당업자에게 명백한 바와 같이, 본 발명의 범위를 벗어나는 일이 없이, 에포크들, 에포크들간의 스케일링 인수 및 에포크 당 슬롯의 수에 대한 다양한 조합이 이루어질 수 있다.

현재 시간은 현재 스케줄러 시스템 시간에 대한 값을 유지하는 레지스터이다. 이 레지스터는 스케줄러_틱 마다 증분된다. 양호한 실시예에 있어서, 현재 시간 레지스터의 범위는 타이머 기반 스케줄러의 범위의 4배로 선택된다. 이것은 흐름 큐 제어 블록에서 발견된 시간 스탬프 필드들 중 하나(즉, NextRedTime이나 TextGreentime)와 현재 시간을 비교할 때 현재 시간 랩(wrap)을 결정할 수 있도록 해준다.

동작

도 5는 본 발명의 스케줄러의 동작을 나타내는 흐름도이다. 현재 포인터는 각 에포크 내에서 서비스 위치를 지시하는 데 사용된다. 각 스케줄러_틱 동안에, 현재 포인터가 지시하는 슬롯이 검사된다. 슬롯이 비어 있는 것으로 알려지면, 현

재 포인터는 다음의 비어 있지 않은 슬롯이나, 현재 시간에 대응하는 슬롯으로 이동한다. 슬롯들간의 거리는 에포크에 따라 변하므로, 현재 포인터는 현재 시간에 뒤떨어지지 않으면서 각 에포크 마다 상이한 속도로 이동한다. 슬롯이 비어 있지 않은 것으로 알려지면, 흐름 큐 에포크 후보가 알려진다. 각 에포크는 흐름 큐 에포크 후보가 알려져 있는지의 여부를 독립적으로 판정한다. 도 4에 나타낸 바와 같이, 흐름 큐 캘린더 후보는 최저 번호의 에포크가 제일 먼저 선택되는 절대 우선 순위 선택에 의해 에포크 후보들 중에 선택된다. 도 4에 나타낸 바와 같이, 선택 순서는 다음과 같다.

1. 에포크 0

2. 에포크 1

3. 에포크 2

4. 에포크 3

흐름 큐 에포크 후보가 선택되면, 흐름 큐 포인터는 LIFO 스택으로부터 디큐잉된다. 현재 포인터가 지시하는 슬롯이 이러한 디큐잉 동작 후에 비어 있지 않은 것으로 알려지면, 현재 포인터는 변함없이 남아 있게 된다. 그러나, 현재 포인터가 지시하는 슬롯이 이러한 디큐잉 동작 후에 비어 있는 것으로 알려지면, 현재 포인터는 다음의 비어 있지 않은 슬롯이나, 현재 시간에 대응하는 슬롯이나, 또는 흐름 큐 서비스 동작이 그 슬롯으로부터 디큐잉된 흐름 큐로 이동된 슬롯으로 이동한다. 현재 포인터는 이러한 가능한 거리들 중 최단 거리로 이동된다.

도 3에 나타낸 스케줄러 시스템은 복수의 흐름(210), 시간 기반 캘린더(220, 230, 250), 가중 공정 큐잉(WFQ) 캘린더(240) 및 타겟 포트 큐(260)로 구성된다.

흐름들(210)은 할당, 즉 관련 사용자가 선택하여 지불한 서비스 레벨을 기초로 해서 공통 시스템 특성을 공유하는 순서화된 프레임 목록들을 유지하는 데 사용되는 제어 구조들이다. 이러한 특성에는 최소 대역폭, 피크 대역폭, 최선의 노력(best effort) 대역폭 및 최대 버스트 크기의 서비스 품질(QoS) 요구 사항이 있다. 양호한 실시예는 통신 시스템의 QoS를 지원하기 위한 것으로 제공된 흐름 큐들 이외에도, 프레임들(즉, 필터링된 트래픽)을 폐기하고, 네트워크 프로세서 시스템의 이그레스에서 인그레스로 프레임 데이터를 랩핑하기 위한 것으로 규정된 흐름 큐들을 필요로 한다.

시간 기반 캘린더(220, 230, 250)는 최소 대역폭 및 최선의 노력(best effort) 피크 속도 요구 사항으로 패킷을 스케줄링하는 데 사용된다. 도 3에 나타낸 바와 같이, 3개의 시간 기반 캘린더가 상기와 같은 목적으로 사용되는데, 그 중 2개의 캘린더(220, 230)는 최소 대역폭용이고, 나머지 1개의 캘린더(250)는 흐름 큐를 최대 최선의 노력(best effort) 피크 속도로 제한(피크 대역폭 실현)하는 데 사용된다. 최소 대역폭용으로 제공된 2개의 시간 기반 캘린더(220, 230)[캘린더(220)는 저 레이턴시 서비스(LLS)용이고, 캘린더(230)는 정상 레이턴시 서비스(NLS)용임]는 최소 대역폭 QoS 등급 내에서 상이한 서비스 등급(즉, 저 레이턴시 및 정상 레이턴시)을 지원할 수 있도록 해준다.

가중 공정 큐잉(WFQ) 캘린더(240)는 최선의 노력(best effort) 서비스 및 최선의 노력(best effort) 피크 서비스(시간 기반 캘린더 220, 230 중 하나와 함께 사용될 경우)용으로 사용된다. 또한, WFQ 캘린더(240)는 최선의 노력(best effort) 서비스 QoS 등급 내에서 상이한 서비스 등급을 지원할 수 있도록 해주는 큐 가중치를 지원한다. 양호한 실시예에 있어서, 그러한 WFQ 캘린더는 지원된 매체 포트(출력 포트)의 수에 대응해서 40개가 있다. 40개의 그러한 포트의 선택은 하드웨어 비용과 설계 복잡성간에 절충안이며, 본 발명의 범위를 제한하려는 것은 아니다.

전술한 캘린더들의 각각에 있어서, 포인터(흐름ID)는 흐름 큐의 캘린더 내 위치를 나타내는 데 사용된다. 따라서, 흐름 0은 캘린더(220) 내의 흐름ID(221)를 갖고, 흐름 1은 캘린더(230) 내의 흐름ID(232)와 WFQ 캘린더(240) 내의 흐름ID(241)를 가지며, 흐름 2047은 캘린더(230) 내의 흐름ID(231)와 캘린더(250) 내의 흐름ID(251)를 갖는데, 모두 도 3에서 화살표로 표시되고 있다. 또한, 시스템의 복수의 캘린더 내에 존재하는 단일 흐름 큐에 대해서 포인터가 없거나, 1개이거나, 2개일 수 있다. 통상적으로, 그렇지 않은 캘린더 내의 포인터들은 초기화되지 않았거나 비어 있지 않은 흐름 큐들을 나타낸다. 어느 한 흐름 큐에 대한 포인터(또는 흐름ID)가 시스템의 특정 캘린더 내에 존재하는 경우, 그 흐름 큐는 그 특정 캘린더 " 내에" 존재한다고 말할 수 있다.

타겟 포트 큐들은 공통 포트 목적지 및 우선 순위를 갖는 순서화된 프레임 목록들을 유지하는 데 사용되는 제어 구조들이다. 양호한 실시예에 있어서는, 매체 포트(또는 출력 포트) 당 2개의 우선 순위가 제공되어, 상이한 서비스 등급, 소위 높은 우선 순위의 타겟 포트 큐와 소위 낮은 우선 순위의 타겟 포트 큐를 지원할 수 있도록 해준다. 2개의 우선 순위의 선택은 하드웨어 비용과 설계 복잡성간의 절충안이며, 본 발명의 범위를 제한하려는 것은 아니다. 또한, 양호한 실시예는 별도의 랩 큐(272)와 폐기 포트 큐(270)를 포함한다.

각각의 시간 기반 캘린더(220, 230, 250)는 복수의 에포크로 이루어지는데, 도 3에서는 각각의 캘린더에 대해서 4개의 에포크가 겹쳐진 직사각형들로 표현되어 있다. 도 4는 4개의 에포크(302, 304, 306, 308)와 함께, 이 에포크들을 위한 통상의 타이밍 장치를 보여주는데, 제1 에포크(302)(에포크 0으로 표시됨)는 스케줄러 틱(150 ns 마다 512 바이트를 허용)의 스텝을 갖고, 제2 에포크(304)는 제1 에포크(302)의 스텝의 16배의 스텝을 갖고, 제3 에포크(306)는 제2 에포크(304)의 스텝의 16배의 스텝을 가지며, 제4 에포크(308)는 제3 에포크(306)의 스텝의 16배의 스텝을 갖는다. 이와 같이, 제1 에포크(302)는 높은 우선 순위를 가지며(제2 에포크(304)의 16배의 서비스로 스케줄링됨), 비용 증가와 관련된 서비스 우선 순위의 계층을 형성한다. 현재 포인터[예컨대, 에포크(302) 내의 포인터(312)]는 큐에서 현재 처리 위치가 어딘지에 대해서 포인터를 제공하는 각 에포크와 관련된다. 에포크들을 통해 진행하는 본 시스템은 현재 포인터를 증분시켜야 하므로, 처리 방향은 에포크에서 낮은 곳에서 높은 곳으로 진행한다. 도 4에는 또한 현재 시간(320)과, 각 에포크 내의 스텝을 구동하고 클록(320)을 구동하는 스케줄러 틱(330)이 도시되어 있다.

우선 순위 선택은 절대 우선 순위 선택인데, 이것은 임의의 시간 간격 동안에는 단지 하나만이 서비스될 수 있으므로, 최고의 우선 순위를 가진 것만이 서비스된다는 것을 의미한다. 각 에포크 내의 현재 포인터가 데이터 흐름을 지시하면, 최저의 것(에포크 0)이 서비스된다. 에포크 0이 서비스가 필요없다면, 에포크 1이 서비스되고, 그 다음엔 에포크 2, 3, 4가 계속해서 서비스된다.

도 6은 도 4의 각 에포크에 대한 비트의 시간 분해능을 나타낸다. 즉, 이것은 후술하는 바와 같이, 현재 시간 저장소(23 비트의 카운터) 내의 어떤 비트들이 각 에포크와 관련되어 있는지를 보여준다.

도 7은 사용자로부터의 통신 파라미터를 나타낸다. 사용된 대역폭은 Y 축 상에 기입되고, 시간은 X 축 상에 기입된다. 사용자는 화살표(350)로 표시된 일관된 대역폭을 할당받을 수 있고, 또한 버스트 폭(370)으로 도시된 지속 기간 동안 화살표(360)로 표시된 피크 버스트 대역폭을 가질 수도 있다. 시간(380) 동안 대역폭 부재로 도시된 지연 또는 대기 시간은 피크 버스트 대역폭 사용에 대한 대가로서 부과될 수 있고, 또한 이하 더 상세히 후술하는 바와 같이 MBS로 알려진 크레디트의 사용에 의해 강요될 수 있다.

큐가 비게 되면, 그 큐는 연결 해제 형태로 캘린더로부터 제거된다. 어떠한 캘린더에도 존재하지 않는 큐가 프레임을 전송하기 시작하면, 그 큐는 연결(또는 이전에 프레임을 전송한 큐로 복귀하기 위한 재연결)로 불리는 프로세스에서 새로

운 큐로서 처리된다. 연결 해제 및 재연결 프로세스는 큐로 하여금 라인의 선두로 이동하게 함으로써, 각각 서비스한 후 계산된 우선 순위를 가지고 자신의 자리에 계속해서 있었다면 그 큐가 있었을 자리보다 앞으로 이동하게 하는 바람직하지 않은 결과를 초래할 수 있다.

현재 시간을 시간 기반 캘린더 내의 위치로 변환

현재 시간은 현재 시간, 에포크들간의 스케일링 인수 및 각 에포크에 사용된 슬롯수를 검사함으로써 타이머 기반 캘린더 내의 위치로 변환된다. 양호한 실시예에 있어서, 스케일링 인수는 16이고 각 에포크 마다 512개의 슬롯이 존재하므로, 각 에포크 내의 위치를 식별하기 위해서는 9 비트가 필요하다. 도 6에 나타낸 바와 같이, 비트 8 내지 비트 0은 에포크 0 내의 위치를 규정하는 데 사용되고, 비트 12 내지 비트 4는 에포크 1, 비트 16 내지 비트 8은 에포크 2, 그리고 비트 20 내지 비트 12는 에포크 3 내의 위치를 규정하는 데 사용된다.

흐름 큐를 추가할 경우 WFQ 캘린더 내의 위치 결정

본 발명은 가중 공정 큐의 사용을 고려할 수 있는데, 이 경우 최소 대역폭 고객에게 필요하지 않은 대역폭은 가중치 또는 우선 순위 및 프레임 길이에 기초하여 그 큐를 다음 서비스에 대한 순서를 결정하는 공식에 따라 공유하는 사용자들과 함께 최선의 노력(best effort)으로 사용될 수 있다.

어느 한 패킷이 흐름 큐로 인큐잉되어, 그 흐름 큐가 WFQ에 추가되면, 도 3의 WFQ 캘린더(240) 내의 위치는 다음의 방식들 중 하나로 결정된다.

1. WFQ 캘린더의 현재 포인터가 지시하는 위치에 추가.

2. WFQ 캘린더의 현재 포인터가 지시하는 위치 바로 앞의 위치에 추가.

3. 흐름 큐의 가중치 QD를 사용하여 현재 포인터가 지시하는 위치로부터의 거리를 결정.

양호한 실시예에 있어서, 흐름 큐의 가중치는 현재 포인터가 지시하는 위치로부터의 거리를 결정하는 데 사용된다. 그 거리 계산은 다음의 형식을 갖는다.

슬롯 거리 = Min((QD * S), 1)

여기서, S는 임의의 양의 정수 값인 스케일링 인수이다. 양호한 실시예에 있어서 스케일링 인수 S는 16이다.

현재 시간을 NextGreenTime으로 변환

NextGreenTime은 (피크 대역폭 실현 캘린더와 함께) 피크 대역폭 실현을 제공하기 위해서 WFQ 캘린더(240)가 사용하는 시간 스탬프 필드이다. 현재 시간을 NextGreenTime(양호한 실시예에서는 NxtGT.V, NxtGT.E)으로 변환하는 일은 흐름 큐 제어 블록의 피크 대역폭 서비스 속도 필드를 검사하는 일을 필요로 한다.

양호한 실시예에 있어서, PSD.E의 값은 다음과 같이 NxtGT.V 필드용으로 현재 시간으로부터 사용되는 비트들을 결정하는 데 사용된다.

PSD.E 현재 시간 비트

0 8 내지 0

1 12 내지 4

2 16 내지 8

3 20 내지 12

NxtGT.E는 PSD.E의 값과 같게 설정된다.

NextRedTime 또는 NextGreenTime과 현재 시간간의 비교(테스트 이후)

양호한 실시예에 있어서, 다음의 단계는 흐름 큐 제어 블록으로부터의 시간 스탬프와 현재 시간간의 비교를 가능하게 하는 방법을 포함한다.

1. 상기 비교를 수행하기 위해서 현재 시간으로부터 비트들을 선택한다. 이러한 설정은 비교될 시간 스탬프(NextRed Time 또는 NextGreenTime)로부터 " 도트 E" 필드를 검사하는 일을 필요로 한다.

도트 E 현재 시간 비트

0 8 내지 0

1 12 내지 4

2 16 내지 8

3 20 내지 12

2. " A" 와 " B" 의 선후 관계를 판정하기 위해서, 어떠한 연산 실행도 무시하기 위해서 먼저 B의 2의 여수를 만든 다음에, 그 결과를 A에 더한다. 그 결과가 제로가 아닌 경우에는 그 결과의 최상위 비트는 0이고, 따라서 A는 B보다 늦다. 그렇지 않은 경우에는 B가 A보다 늦다.

NextRedTime 또는 NextGreenTime을 시간 기반 캘린더 내의 위치로 변환

흐름 큐 제어 블록 내의 시간 스탬프 필드는 흐름 큐가 그 서비스 파라미터를 위반하는 것을 방지하는 수단의 일부로서 기능한다. 양호한 실시예에 있어서, " 도트 E" 필드는 에포크를 나타내고, " 도트 V" 필드는 그 에포크 내에서 위치 0으로부터의 거리를 나타낸다.

흐름이 그 피크 속도를 위반한 경우의 NextGreenTime 계산(NextGreenTime을 베이스로 사용)

양호한 실시예에 있어서, 흐름이 그 피크 속도를 위반한 경우에 NextGreenTime의 계산은 바이트 단위의 패킷 길이를 결정하는 BCI, 피크 서비스 속도 및 NextGreenTime의 현재 값을 검사함으로써 결정된다. 도 8에 있어서, FL은 BCI로부터 결정된 프레임의 바이트 길이를 나타낸다. 이하, 프로세스 블록을 설명한다.

프로세스 2는 스케일링 인수(도트 E)가 NextGreenTime과 피크 서비스 속도 양자 모두에 대해 동일한 경우의 슬롯 거리 계산(Temp)이다.

프로세스 4는 피크 서비스 속도의 스케일링 인수가 NextGreenTime의 스케일링 인수보다 큰 경우의 슬롯 거리 계산(Temp)이다.

프로세스 5는 피크 서비스 속도의 스케일링 인수가 NextGreenTime의 스케일링 인수보다 작은 경우의 슬롯 거리 계산(Temp)이다.

프로세스 7은 슬롯 거리(Temp)가 현재 스케일링 인수의 용량보다 큰 경우의 NxtGT.V 및 NxtGT.E 값의 계산이다(양호한 실시예의 경우, 도트 V 값은 511을 초과할 수 없음).

프로세스 10은 슬롯 거리(Temp)가 현재 스케일링 인수의 감소를 고려한 경우의 NxtGT.V 및 NxtGT.E 값의 계산이다. 이것은 바람직한데, 그 이유는 스케일링 인수가 작아질수록, 시간 베이스(time base)가 더 정확해지기 때문이다.

프로세스 11은 슬롯 거리(Temp)가 현재 스케일링 인수의 변경을 필요로 하지 않거나 허용하지 않는 경우의 NxtGT.V 값의 계산이다. NxtGT.E의 값은 변함없이 남아 있게 된다.

흐름이 그 피크 속도를 위반하지 않는 경우의 NextGreenTime 계산(현재 시간을 베이스로 사용)

양호한 실시예에 있어서, 흐름이 그 피크 속도를 위반하지 않는 경우에 NextGreenTime의 계산은 바이트 단위의 패킷 길이를 결정하는 BCI, 피크 서비스 속도 및 현재 시간을 검사함으로써 결정되며, 도 9에 나타내었다. 이하, 프로세스 블록을 설명한다. 도 9에 있어서, FL은 BCI로부터 결정된 프레임의 바이트 길이를 나타낸다.

프로세스 21은 슬롯 거리 계산(Temp)이다. 프로세스 블록 23, 25, 27 및 29는 피크 서비스 속도에 사용되는 스케일링 인수에 기초하여, 현재 시간 레지스터 내의 비트들로부터 베이스 시간(BaseT)의 값을 지정한다. NextGreenTime은 슬롯 거리, 스케일링 인수 및 베이스 시간으로부터 결정된다.

프로세스 31은 슬롯 거리(Temp)가 현재 스케일링 인수의 용량보다 큰 경우의 NxtGT.V 및 NxtGT.E 값의 계산이다(양호한 실시예의 경우, 도트 V 값은 511을 초과할 수 없음).

프로세스 34는 슬롯 거리(Temp)가 현재 스케일링 인수의 감소를 고려한 경우의 NxtGT.V 및 NxtGT.E 값의 계산이다. 이것은 바람직한데, 그 이유는 스케일링 인수가 작아질수록, 시간 베이스가 더 정확해지기 때문이다.

프로세스 35는 슬롯 거리(Temp)가 현재 스케일링 인수의 변경을 필요로 하지 않거나 허용하지 않는 경우의 NxtGT.V 및 NxtGT.E 값의 계산이다.

MBS 획득 크레디트 계산

양호한 실시예에 있어서, 제로가 아닌 최대 버스트 필드와 함께 사용 중인(QinUse=1) 흐름 큐는 그 흐름 큐가 비어 있을 경우에 토큰을 획득한다. MBSCredit 필드는 그 비어 있는 흐름 큐로 패킷이 인큐잉되는 경우에 갱신된다. 획득된 토큰수를 판정하기 위해서, NextRedTime과 현재 시간이 검사되며, 도 10은 이를 나타내고 있다.

프로세스 302, 303, 305 및 307에서는, NextRedTime에 의해 사용되는 스케일링 인수에 기초하여, 현재 시간으로부터 비트들을 선택하여, 흐름 큐가 비어 있었던 기간을 판정하는 데 사용되는 스케일드 시간(scaled time)(TimeA)을 작성한다.

판단 블록 308은 TimeA가 NextRedTime 시간 스탬프 필드가 나타내는 시간보다 늦은지의 여부를 판정한다. 이러한 목적을 위해서 NextRedTime을 사용할 경우에는 흐름 큐가 그 다음의 가능한 스케줄링 시간 후에도 계속 비어 있을 것을 요구한다. TimeA가 NextRedTime 시간 스탬프 필드가 나타내는 시간보다 늦지 않은 경우에는 더 이상의 조치가 취해지지 않는다.

판단 블록 309는 타이머 랩 상태를 처리하여, 프로세스 블록 311 및 310에서 흐름 큐가 토큰을 축적하고 있는 지속 기간(TimeT)을 계산하는 것을 허용한다.

프로세스 블록 313, 315 및 316은 MBSCredit.V에 대한 최종 계산이다. 판단 블록 312 및 314는 MBS 필드를 규정하는 데 사용되는 스케일링 인수로 인해서 TimeT를 조정할 필요가 있는지의 여부를 판정한다.

사용된 MBS 크레디트 계산

MBSCredit에 대한 새로운 값은 MBSCredit의 현재값, 프레임 길이를 결정하는 BCI 및 일관된 서비스 속도로부터 결정된다. 일관된 서비스 속도의 사용은 MBS 값을 계산할 때 사용되는 방법에 기인한다(등식에서 나눗셈이 복잡해지는 것을 방지). 양호한 실시예에 있어서, MBSCredit는 음의 값을 가질 수도 있다. 이하, 프로세스 블록을 설명한다. 도 11에서, FL은 BCI로부터 결정된 프레임의 바이트 길이를 나타낸다.

프로세스 402, 404 및 405는 길이 FL인 프레임에 사용된 토큰수를 결정한다. 판단 블록 401 및 403은 MBSCredit 필드를 규정하는 데 사용되는 스케일링 인수로 인해서 Temp를 조정할 필요가 있는지의 여부를 판정하는 데 사용된다.

프로세스 블록 406은 사용된 토큰수에 의해 MBSCredit.V의 값을 조정한다.

흐름 큐 제어 블록 노화(aging)

흐름 큐 제어 블록에서 스케일링 인수를 사용함으로써 시간 스템프 필드를 유지하는 데 필요한 하드웨어를 축소할 수 있다. 시간 스탬프가 너무 노화되어 적절하지 않은 때를 정확하게 결정하기 위해서는, 시간 스탬프 및 잔여 필드가 유효하지 않음을 나타내는 방법이 필요하다. 도 12 및 도 13은 시간 스탬프가 더 이상 유효하지 않은 흐름 큐 제어 블록을 표시하는 방법을 나타내고 있다.

스케줄러 시스템은 현재 사용 중인(QinUse=1) 모든 흐름 큐 제어 블록의 목록을 포함한다. 그러한 목록을 유지하는 여러 방법이 당업자에게 알려져 있다.

양호한 실시예에 있어서, 시간 스템프 필드의 도트 V 필드는 2 비트까지 확장된다. 이러한 추가적인 비트들은 시간 스탬프 필드가 갱신되는 경우에 현재 시간 포인터로부터 취해진다. 현재 시간 포인터로부터 사용된 비트들은 도 6에 나타낸 바와 같이 도트 E 필드의 값에 의해 결정되며, 다음과 같다.

도트 E 사용된 현재 시간 비트

0 10 내지 9

1 14 내지 13

2 18 내지 17

3 22 내지 21

타이머는 흐름 큐 제어 블록 검사 프로세스가 일어나는 때를 결정하는 데 사용된다. 도 12를 참조하면, '서비스 틱 노화(Aging Service Tick)' 검사를 시작한다. 이것이 진실이면, 프로세스는 계속해서 502에서 노화 목록으로부터 흐름 큐 제어 블록(QCB)을 선택한다. 프로세스는 계속해서 503에서 선택된 흐름 큐 제어 블록의 QinUse 필드를 검사한다.

흐름 큐 제어 블록이 사용 중이 아닌 경우에는 프로세스는 블록 501로 돌아가서 다음 서비스 틱을 기다리며, 그렇지 않은 경우에는 프로세스는 계속해서 일관된 서비스 속도 필드(504)와 QinRed 필드(505)를 검사한다. 흐름 큐가 일관된 서비스로 지정되고 흐름 큐가 LLS 캘린더와 NLS 캘린더 중 어느 쪽에도 존재하지 않는 경우(QinRed=0)에는 506에서 노화 결정이 일어난다. 이하, " 노화 테스트(TestAge)" 를 위한 조치를 설명한다.

흐름 큐가 일관된 서비스 속도로 지정되지 않은 경우에는, 프로세스는 블록 507에서 피크 서비스 속도 필드를 검사한다. 피크 서비스 속도로 지정되지 않은 경우에는, 프로세스는 블록 501로 돌아가서 다음 서비스 틱을 기다린다. 피크 서비스 속도로 지정된 경우에는, 흐름 큐 제어 블록을 검사하여, 흐름 큐가 WFQ 캘린더(508) 또는 피크 대역폭 실현 캘린더(509)에 존재하고 있는지의 여부를 판정한다. 흐름 큐가 어느 쪽이든 존재하는 경우에는, 프로세스는 블록 501로 돌아가며, 그렇지 않은 경우에는, 블록 510에서 노화 결정이 일어난다. 이하, " 노화 테스트(TestAge)" 를 위한 조치를 설명한다.

도 13에 나타낸 바와 같이, 노화 테스트는 흐름 큐 제어 블록의 시간 스탬프 필드들 중 하나와 현재 시간을 입력으로 사용하며, QinUse 필드의 상태를 갱신하는 프로세스로 돌아간다. 도 5의 흐름도에 있어서, 이 결과는 블록 512 및 513에서 흐름 큐 제어 블록을 갱신하고 노화 목록으로부터 흐름 큐 제어 블록을 제거하는 데 사용된다.

도 13으로 되돌아가서, 노화 테스트 프로세스는 시간 스탬프 필드가 너무 노화되어 그 유용성을 유지할 수 없는지의 여부를 결정하는 데 현재 시간의 어떤 비트들이 사용되는지를 결정하기 위해서 스케일링 인수 E를 사용한다. 블록 600 내지 606은 이러한 일을 완수한다.

계속해서 블록 606 및 607에서는 도트 V 필드의 상위 비트들(위에서 정의되고 도 3에 나타낸 MM 비트들)과 현재 시간에서 선택된 비트들간의 비교가 이루어진다. 노화 테스트는 블록 607 및 608에서 시간 스탬프과 관련된 에포크가 최종 갱신된 이래로 한 번보다 많이 랩핑되었는지의 여부를 결정한다. 한 번보다 많이 랩핑되었다면, 시간 스탬프는 더 이상 사용될 수 없어 그 시간 스탬프 필드는 QinUse 비트를 제로로 설정함으로써 무효로 표시되며, 그렇지 않은 경우에는 QinUse 비트는 변함없이 남아 있게 된다(블록 609, 610).

가중 공정 큐잉(WFQ) 캘린더는 소위 " 최선의 노력(best effort)" 서비스에 사용되며, 시간 기반 캘린더와 함께 사용되는 경우에는 소위 " 최선의 노력 피크(best effort peak)" 서비스라고 한다. 즉, 최선의 노력 서비스는 보장된 대역폭을 제공하지는 않지만(시간 단위 마다 x 비트의 대역폭이 제공됨), 그 보장된 대역폭의 고객을 만족시킨 후에 남은 대역폭에 대해서는 다른 사용자들과 경쟁할 수 있다. 이러한 서비스는 보장된 대역폭 서비스에 비해 저수준의 서비스이고, 보통은 비용이 상당히 저렴하다. 소위 " 최선의 노력 피크" 서비스는 사용자가 가입한 보장된 서비스 수준 이상을 요구하는 경우에 자원의 잉여 대역폭을 경쟁적으로 할당한다. 따라서, 사용자는 최대 추가적인 5 Mbit, 서비스 중 총 15 Mbit의 총 피크 서비스를 위한 최선의 노력 서비스와 함께 10 Mbps의 서비스를 구입할 수 있다(최종 5 Mbit의 서비스는 그것이 사용 가능하고 가중 공정 큐잉이 다른 사용자들과의 공정한 분배가 그것을 허용하는 경우에만 제공됨).

또한, WFQ 캘린더는 최선의 노력 서비스 QoS 등급 내에서 상이한 서비스 등급들의 지원을 가능하게 하는 큐 가중치를 지원한다. 본 명세서에서 사용된 바와 같이, 큐 가중치는 소정의 사용자가 그의 서비스 레벨에 기초하여 할당받는 상대적인 우선 순위이다. 큐 가중치는 본 시스템과 함께 설명되는 가중 공정 큐잉 시스템에서 서비스간의 지연량과 관련된다. 양호한 실시예에 있어서는, 도 3에 나타낸 바와 같이, 그러한 WFQ 캘린더는 지원된 매체 포트수에 대응해서 40개가 있다. 40개의 그러한 포트의 선택은 하드웨어 비용과 설계 복잡성간에 절충을 필요로 하는 임의의 설계 선택권이며, 본 발명의 범위를 제한하려는 것은 아니다.

한 주기(또는 클록 간격)는 한 스케줄러_틱으로서 규정된다. 이것은 접속되어 있는 하드웨어의 응답 시간에 따라 150 ns 또는 165 ns이지만, 설계 파라미터와 하드웨어 능력에 따라 더 크거나 더 작을 수 있다. 스케줄러_틱 동안에 흐름 큐는 서비스를 위해 선택된다. 설명된 선택 알고리즘에 있어서, WFQ 캐린더는 최소 대역폭(일관된 서비스 속도)으로 지정된 모든 흐름 큐들이 서비스를 요구하지 않는 경우에 선택된다(즉, WFQ 캘린더는 스케줄러가 관리하는 남겨진 대역폭을 사용할 수 있다). 달리 설명하자면, 시간 기반 큐들(220, 230)은 그것들이 스케줄링되어 있고 전송할 정보를 가지고 있는 경우에 각 클록 간격 동안에 서브된다. 그렇지 않은 경우에는, WFQ 큐(240)가 그 클록 간격 동안에 서브된다. 타이머 기반 스케줄러와는 대조적으로, 네트워크 프로세서의 총 최소 대역폭이 관리될 수 있는 경우에, 각 WFQ 캘린더는 하나의 타켓 포트를 위한 최선의 노력 대역폭을 관리한다. WFQ 캘린더 또는 큐(240)의 목적은 이러한 최선의 노력 대역폭을 경쟁하는 흐름 큐들에게 공정하게 분배하는 것이다. 이것은 서비스를 위해 선택된 흐름 큐가 전송하는 바이트수를 고려하고 그 전송되는 바이트수에 기초하여 그 캘린더 내의 흐름 큐를 그 현재 위치로부터 떨어진 위치로 이동시킴으로써 달성된다. 즉, 흐름이 스케줄러_틱 동안에 전송하는 바이트가 많을수록, 다음 서비스 전에 그 캘린더

위로 더 높이 올라간다(또한 사이에 끼어드는 흐름들도 더 많아지고 따라서 주기로 길어진다).

양호한 실시예에 있어서, 도 14에 나타낸 바와 같이, WFQ 캘린더(240a)는 512개의 슬롯으로 구성된다. 각 슬롯은 흐름 큐들에 대한 LIFO 포인터 스택을 포함한다. 실시예에서 사용되는 슬롯의 수는 하드웨어 비용과 설계 복잡성간에 절충되며, 본 발명의 범위를 제한하려는 것은 아니다.

도 14를 다시 참조하면, 현재 시간은 현재 스케줄러 시스템 시간에 대한 값을 유지하는 레지스터이다. 이 레지스터는 스케줄러_틱 마다 증분된다. 양호한 실시예에 있어서, 현재 시간 레지스터의 범위는 타이머 기반 캘린더의 범위의 4배로 선택된다. 이것은 흐름 큐 제어 블록에서 발견된 시간 스탬프 필드들 중 하나(즉, TextGreentime)와 현재 시간을 비교할 때 현재 시간 랩(wrap)을 결정할 수 있도록 해준다.

현재 포인터는 WFQ 캘린더 내에서 서비스 위치를 지시하는 데 사용된다. 타이머 기반 캘린더와는 대조적으로, 현재 포인터는 스케줄러 시스템 시간과 관계가 없다.

동작

도 15의 흐름도에 나타낸 바와 같이, 각 스케줄러_틱 동안에, 각 타겟 포트 큐의 상태가 먼저 검사된다. 각 WFQ 캘린더는 한 쌍의 포트와 관련되는데, WFQ 포트 0은 높은 우선 순위 포트 0 및 낮은 우선 순위 포트 0과 관련된다. 라인 262에 입각해서 타겟 포트 큐의 임계값을 초과하는 경우에는, WFQ 캘린더는 스케줄러_틱 동안에 더 이상 아무런 조치도 취하지 않는다. (이 시스템은 역압(back pressure) 형태로 그 출력을 제한함으로써, 그 시스템이 처리할 수 없는 프레임들이 송출되는 것을 방지한다.) 타겟 포트 큐의 임계값을 초과하지 않는 경우에는, 현재 포인터가 지시하는 슬롯이 검사된다. 슬롯이 비어 있는 것으로 알려지면, 현재 포인터는 흐름 큐 WFQ 후보를 찾기 위해서 다음의 비어 있지 않은 슬롯으로 이동한다. 모든 큐들이 비어 있는 것으로 알려지면, 현재 포인터는 변경되지 않고 어떠한 후보도 알려지지 않는다. 슬롯이 이 한 캘린더 내에서 비어 있지 않은 것으로 알려지면, 그 슬롯에 저장된 흐름 큐 어드레스가 이 포트에 대한 WFQ 후보이다. 이와 유사하게 40개의 WFQ 캘린더의 각각은 그 관련된 타겟 포트 큐에 대한 후보를 찾을 수 있다.

타겟 포트 임계값 검사의 목적은 패킷이 WFQ 캘린더에서 타겟 포트 큐로 할당되는 속도를 제어하는 것이다. 타겟 포트 큐는 연결된 매체에 의해 지정된 속도로 유출할 것이므로, 임계값을 초과하는 타겟 포트의 상태를 검사하는 것은 타겟 포트 큐가 그 매체의 대역폭을 초과하는 속도로 패킷을 할당받지 못하도록 보장하는 매커니즘이다. 양호한 실시예에 있어서, 임계값은 바이트 크기로서 규정되며, 적절한 동작을 보장하기 위해서는 임계값을 적어도 매체에 규정된 최대 전송 단위(MTU)로 설정해야만 한다.

본 발명의 목적은 최소 대역폭와 함께 최선의 노력 대역폭 스케줄링 방법을 제공하는 것이므로, 단일 흐름 큐가 시간 기반 캘린더와 WFQ 캘린더 양자 모두에 존재하는 경우에, 상기 매커니즘이 적절한 기능을 위해서 필요하다.

최종 흐름 큐 선택은 모든 캘린더들(시간 기반 캘린더 및 WFQ 캘린더)에서 일어난다. 흐름 큐 WFQ 후보가 선택되면, 그 흐름 큐 포인터는 LIFO 스택으로부터 디큐잉된다. 현재 포인터가 지시하는 슬롯이 이러한 디큐잉 동작 후에 비어 있지 않은 것으로 알려지면(즉, 적어도 하나 이상의 엔트리가 그곳에 존재하면), 현재 포인터는 더 이상 변경되지 않는다. 현재 포인터가 지시하는 슬롯이 이러한 디큐잉 동작 후에 비어 있는 것으로 알려지면, 현재 포인터는 다음의 비어 있지 않은 슬롯으로 이동한다. 모든 슬롯들이 비어 있는 것으로 알려지면, 현재 포인터는 변경되지 않는다.

당업자라면 인식할 수 있는 바와 같이, WFQ 캘린더는 자신이 할당받은 모든 흐름 큐들에게 사용 가능한 대역폭을 분배한다. 또한, 흐름 큐에 가중치를 할당함으로써, 각 흐름 큐에 할당되는 대역폭의 상대적인 비율을 변경할 수 있다. 이것은 비어 있지 않은 서비스 흐름 큐가 배치된 위치를 계산함으로써 달성된다. 낮은 가중 인자는 소정의 프레임 길이에 대해 더 자주 서비스를 제공하는데, 이것은 관련된 흐름이 동일한 프레임 길이에 대한 높은 가중 인자보다 다음 서비스에 대해 " 사닥다리(ladder)" 위로 더 조금 이동된다는 것을 의미한다. 이와 유사하게, 긴 프레임을 제공하는 흐름은 동일한 가중 인자를 갖는 짧은 프레임을 제공하는 흐름보다 사닥다리 위로 더 많이 이동되는데, 이것은 동일한 우선 순위 또는 가중 인자를 받은 경우에, 짧은 프레임을 갖는 흐름이 더 자주 서비스를 받는다는 것을 의미한다.

WFQ 거리 계산

거리 계산은 흐름 큐가 디큐잉되는 위치로부터 캘린더에서 이동되는 슬롯수로서 규정된다. 양호한 실시예에 있어서, WFQ 캘린더에 대한 거리 계산은 다음과 같다.

슬롯 거리 = Max(Min(((프레임 길이)*QD*K), max_거리), 1)

여기서, 슬롯 거리는 정수이고, 프레임 길이는 바이트로 측정되고, QD는 흐름 큐 제어 블록에서 지정된 큐 가중치이고, Max_거리는 1이 적은 캘린더 내의 슬롯수로서 규정되며, K는 타겟 포트에 할당된 스케일링 인수이다. K의 지정은 타겟 포트의 MTU, WFQ 캘린더 내의 슬롯수 및 희망하는 QD 범위로부터 결정된다. 양호한 실시예에 있어서, K 값은 다음과 같이 권장된다.

MTU (바이트) K

2048 1/64 th

9216 1/256 th

16384 1/1024 th

본 발명의 범위를 벗어나는 일이 없이 다른 K 값이 선택될 수 있다.

당업자라면 첨부된 도면과 함께 전술한 양호한 실시예에 대한 설명에 있어서 본 발명의 다양한 변형예들이 가능하다는 것을 알 수 있을 것이다. 예컨대, 프레임 크기 및 저장된 가중 인자에 기초한 가중화는 다른 공식으로 대체될 수 있다. 캘린더 내의 슬롯수 및 포트 당 캘린더수는 시스템의 구조를 수용하도록 변경될 수 있다. 에포크수 및 각 에포크의 스텝의 거리, 현재 시간 레지스터의 크기, 스케줄러_틱 속도는 본 발명의 범위를 벗어나는 일이 없이 변경될 수 있다. 추가적으로, 본 발명의 범위를 벗어나는 일이 없이, 시스템 구현에 있어서 다양한 변형예가 가능하고 우선 순위 시스템 및 다양한 알고리즘이 우선 순위를 결정하는 데 사용될 수 있다. 또한, 본 발명의 특징부들 중 일부는 대응하는 다른 특징부들의 사용없이도 사용될 수 있다. 따라서, 전술한 양호한 실시예에 대한 설명은 단지 본 발명의 원리를 예시적으로 설명하기 위한 것이지, 그것을 제한하기 위한 것이 아니다.

(57) 청구의 범위

청구항 1.

데이터 통신 시스템에서 복수의 소스로부터 상기 복수의 소스에 관해 저장된 정보에 기초하여 복수의 목적지로 전송되는 정보 단위들을 처리하는 시간 기반 캘린더와;

데이터 통신 시스템에서 복수의 소스로부터 상기 복수의 소스에 관해 저장된 정보에 기초하여 복수의 목적지로 전송되는 다른 정보 단위들을 처리하는 시간 독립적 캘린더와;

저장된 규칙에 기초하여 선택되는 단일 정보 단위를 상기 캘린더들 중 하나로부터 출력 목적지로 이동시키는 신호를 주기적으로 생성하는 타이머

를 포함하는 장치.

청구항 2.

제1항에 있어서, 상기 시간 독립적 캘린더는 정보 단위들의 흐름들을 처리하고, 각 흐름을 큐에 배치하여, 상기 흐름을 서비스한 후에 상기 흐름을 상기 큐 내의 상이한 자리로 이동시키며,

상기 장치는 흐름이 상기 시간 기반 캘린더에 자리를 가지고 있는지의 여부를 결정하고, 상기 시간 기반 캘린더에 그러한 자리를 가지고 있는 흐름이 연결 해제 및 재연결로 인해서 상기 시간 기반 캘린더에 더 좋은 자리를 갖게 되는 것을 방지하기 위한 매커니즘을 더 포함하는 것인 장치.

청구항 3.

제1항에 있어서, 상기 시간 독립적 캘린더는 정보 단위들의 흐름들을 처리하고, 각 흐름을 큐에 배치하고, 이동을 기다리는 프레임들의 순서와 각 프레임의 가중 인자 및 크기를 제공하며,

상기 장치는 어느 한 프레임이 상기 시간 독립적 캘린더에 의해 이동된 후 상기 프레임의 가중 인자 및 크기에 기초하여 상기 각각의 흐름에 대한 새로운 자리를 계산하기 위한 매커니즘을 더 포함하는 것인 장치.

청구항 4.

정보 단위들이 데이터 통신 시스템에서 복수의 소스로부터 복수의 목적지로 전송될 때 차례로 서비스를 위해서 데이터 흐름들을 큐들에 배치하는 단계와;

데이터 흐름이 상기 큐들에 이전의 자리를 가지고 있었는지의 여부를 결정하는 단계와;

상기 데이터 흐름이 상기 큐들에 이전의 자리를 가지고 있었다면, 할당받을 새로운 자리가 상기 이전의 자리보다 더 좋은 자리인지의 여부를 결정하는 단계와;

상기 할당받을 새로운 자리가 더 좋은 자리라면, 상기 이전의 자리를 사용하여 상기 흐름을 처리하는 단계와;

상기 할당받을 새로운 자리와 마찬가지로 상기 이전의 자리가 좋은 자리가 아니라면, 상기 새로운 자리를 사용하여 상기 흐름을 처리하는 단계

를 포함하는 방법.

청구항 5.

정보 단위들이 데이터 통신 시스템에서 복수의 소스로부터 복수의 목적지로 전송될 때 차례로 서비스를 위해서 데이터 흐름들을 큐들에 배치하는 단계와;

처리 준비가 된 상기 정보 단위들의 각각에 관한 우선 순위 정보를 수신하는 단계와;

처리 준비가 된 각 정보 단위를 각 정보 단위와 관련된 우선 순위 정보에 기초하여 몇몇 우선 순위화된 큐들 - 어떤 하나는 시간 기반이고 다른 하나는 시간 독립적임 - 중 하나에 배치하는 단계와;

연속하는 클록 사이클마다 저장된 규칙들에 기초하여 처리를 위해서 상기 큐들 중 하나를 선택하는 단계와;

연속하는 클록 사이클마다 알고리즘에 기초하여 처리를 위해서 선택된 큐로부터 하나의 정보 단위를 선택하는 단계와;

상기 선택된 정보 단위를 목적지를 향하여 전송하는 단계

를 포함하는 방법.

청구항 6.

제5항에 있어서, 상기 연속하는 클록 사이클마다 알고리즘에 기초하여 처리를 위해서 선택된 큐로부터 하나의 정보 단위를 선택하는 단계는 가중 공정 큐로부터 상기 선택을 행하고, 선택된 정보 단위의 크기 및 선택된 정보 단위의 가중 인자에 기초하여 상기 가중 공정 큐 내의 새로운 자리를 계산하는 단계를 더 포함하는 것인 방법.

청구항 7.

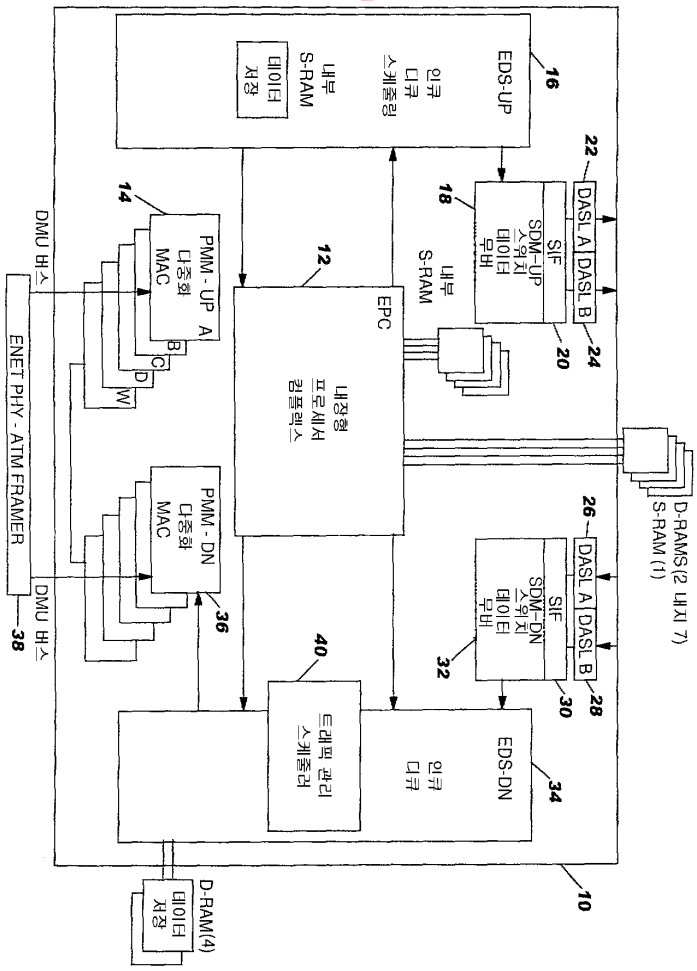
정보 단위들이 데이터 통신 시스템에서 복수의 소스로부터 복수의 목적지로 전송될 때 차례로 서비스를 위해서 데이터 흐름들을 큐들에 배치하는 단계와;

저장된 규칙들에 기초하여 데이터 흐름에 의한 피크 버스트 전송 허용 여부를 결정하는 단계 - 상기 결정 단계는 데이터 흐름에 대한 초기 크레디트를 계산하는 단계와, 시간이 경과함에 따라 그리고 상기 데이터 흐름 속도가 여전히 상기 데이터 흐름에 대해 설정된 임계값 이하일 경우에 상기 크레디트를 증가시키는 단계와, 시간이 경과함에 따라 그리고 상기 데이터 흐름 속도가 상기 임계값을 초과할 경우에 상기 크레디트를 감소시키는 단계와, 상기 크레디트값을 사용하여 소정의 시간에서의 피크 버스트 전송 허용 여부를 결정하는 단계를 포함함 -

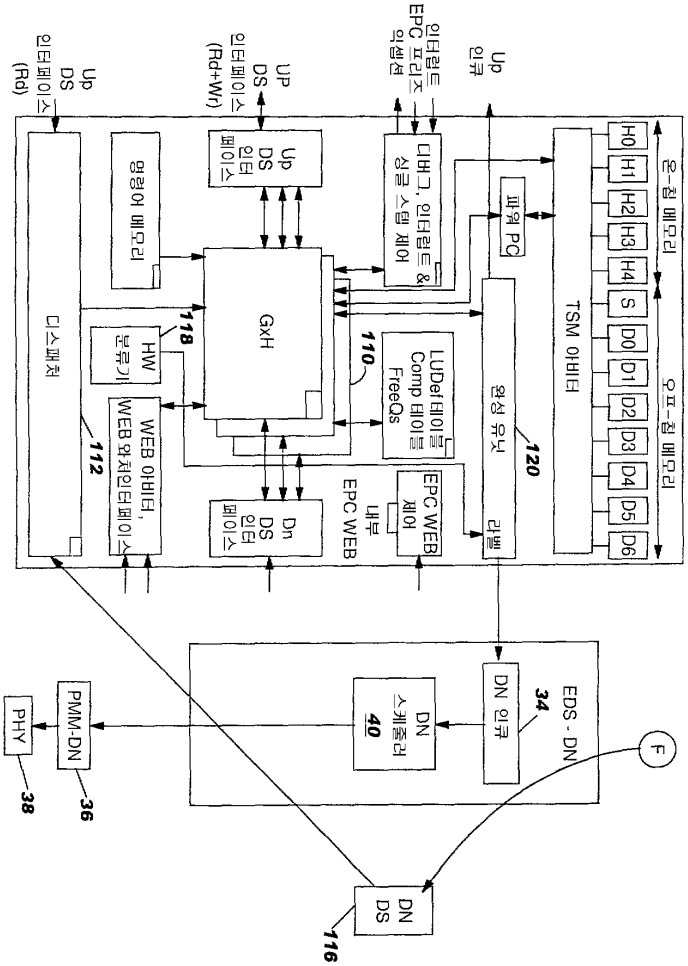
를 포함하는 방법.

도면

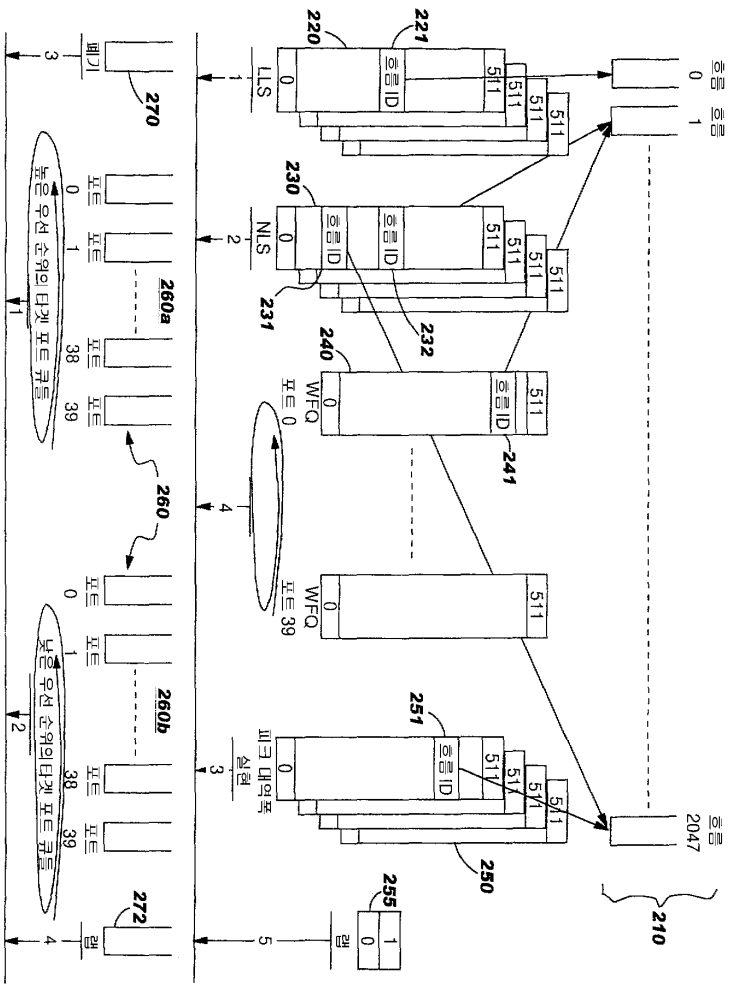
도면 1


EDS-UP
16
인큐
디큐
스케줄링
내부
S-RAM
데이터
저장
22
DASL A
DASL B
18
SIF
SDM-UP
스위치
데이터
유버
20
24
내부
S-RAM
14
PMM-UP A
다중화
MAC
B
C
D
W
DMU 버스
ENET PHY-ATM FRAMER
12
EPC
내장형
프로세서
컴플렉스
D-RAMS(2 내지 7)
S-RAM (1)
26
DASL A
DASL B
SIF
SDM-DN
스위치
데이터
유버
32
30
28
PMM-DN
다중화
MAC
36
38
DMU 버스
40
트래픽 관리
스케줄러
EDS-DN
인큐
디큐
34
10
D-RAM(4)
데이터
저장

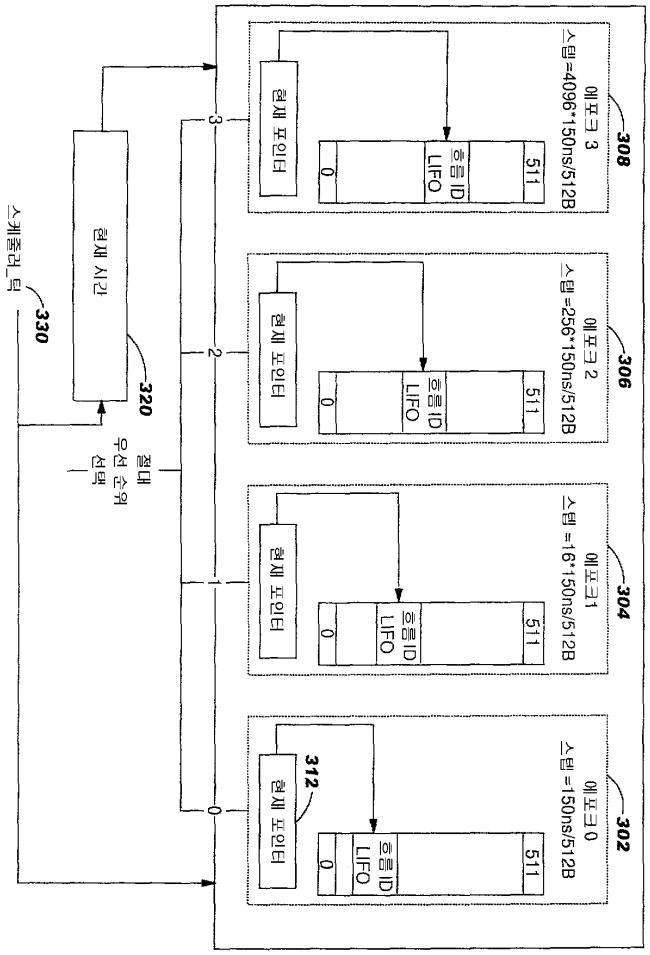
도면 2


Up DS 인터페이스 (Rd)
UP DS 인터페이스 (Rd+Wr)
인터럽트 EPC 프리즈 의심점
Up 인큐
온-칩 메모리
오프-칩 메모리
H0
H1
H2
H3
H4
S
D0
D1
D2
D3
D4
D5
D6
TSM 아비터
Up DS 인터페이스
디버그, 인터럽트 & 싱글 스텝 제어
파워 PC
영랑어 메모리
GxH
110
118
HW 분류기
LUDef 테이블 Comp 테이블 FreeQs
완성 유닛
120
디스패처
112
WEB 아비터, WEB 와치인터페이스
Dn DS 인터페이스
EPC WEB 제어
내부 EPC WEB
라벨
EDS - DN
DN 인큐
34
DN 스케줄러
40
F
PHY
38
PMM-DN
36
DN DS
116

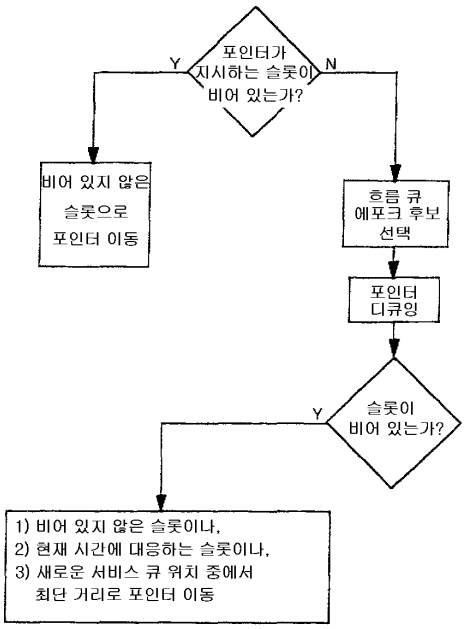
도면 3


흐름 0
흐름 1
흐름 2047
210
220
221
230
231
232
240
241
250
251
255
260
260a
260b
270
272
흐름 ID
511
LLS
NLS
WFQ
포트 0
포트 39
피크 대역폭 실현
램
폐기
높은 우선 순위의 타겟 포트 큐들
낮은 우선 순위의 타겟 포트 큐들
포트
0
1
2
3
4
5
38
39

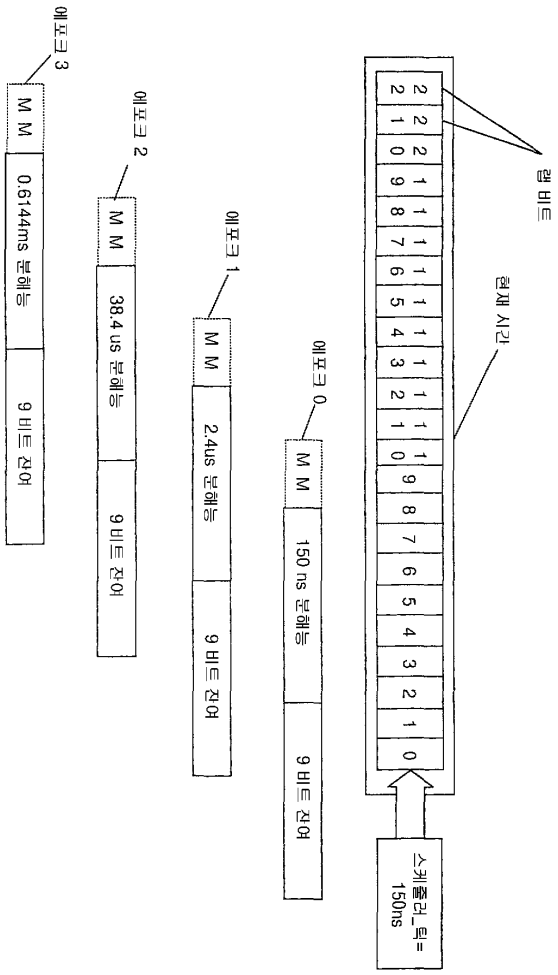
도면 4


에포크 3
스텝=4096*150ns/512B
308
현재 포인터
0
흥름 ID LIFO
511
에포크 2
스텝=256*150ns/512B
306
에포크1
스텝 =16*150ns/512B
304
에포크 0
스텝 =150ns/512B
302
312
3
2
1
0
현재 시간
320
절대 우선 순위 선택
스케줄러_틱
330

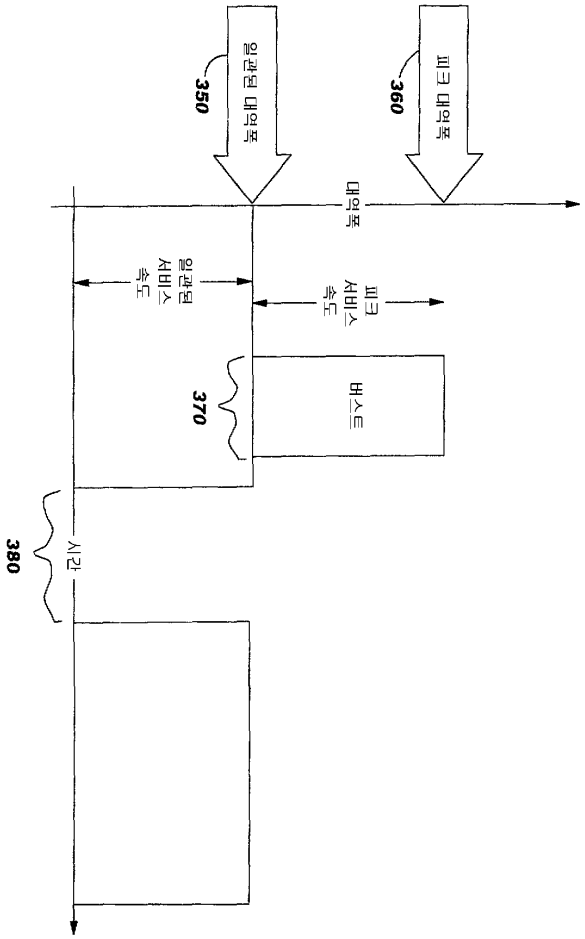
도면 5


포인터가
지시하는 슬롯이
비어 있는가?
Y
N
비어 있지 않은
슬롯으로
포인터 이동
흐름 큐
에포크 후보
선택
포인터
디큐잉
슬롯이
비어 있는가?
Y
1) 비어 있지 않은 슬롯이나,
2) 현재 시간에 대응하는 슬롯이나,
3) 새로운 서비스 큐 위치 중에서
최단 거리로 포인터 이동

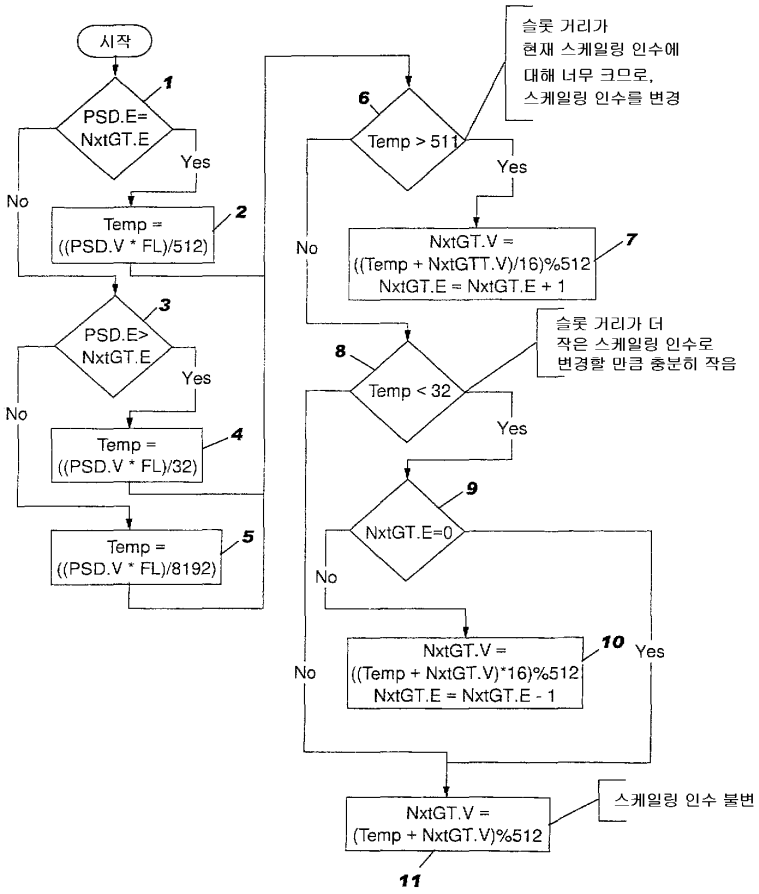
도면 6


에포크 3
M M
0.6144ms 분해능
9 비트 잔여
에포크 2
M M
38.4 us 분해능
9 비트 잔여
에포크 1
M M
2.4us 분해능
9 비트 잔여
에포크 0
M M
150 ns 분해능
9 비트 잔여
22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
랩 비트
결제 시간
스케줄러_틱= 150ns

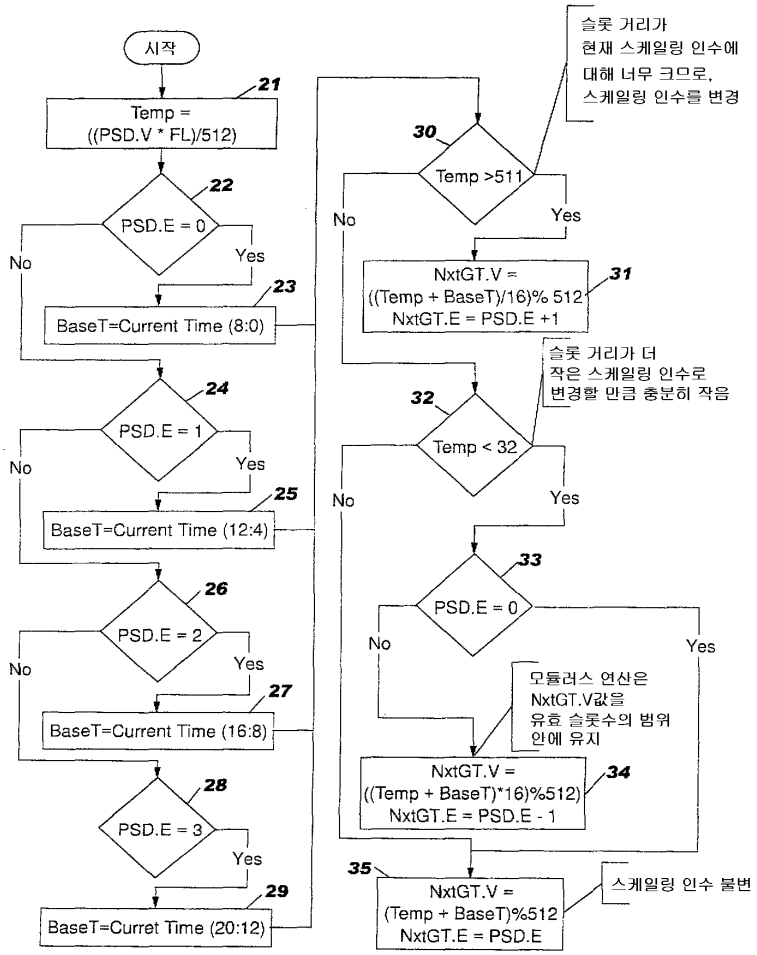
도면 7



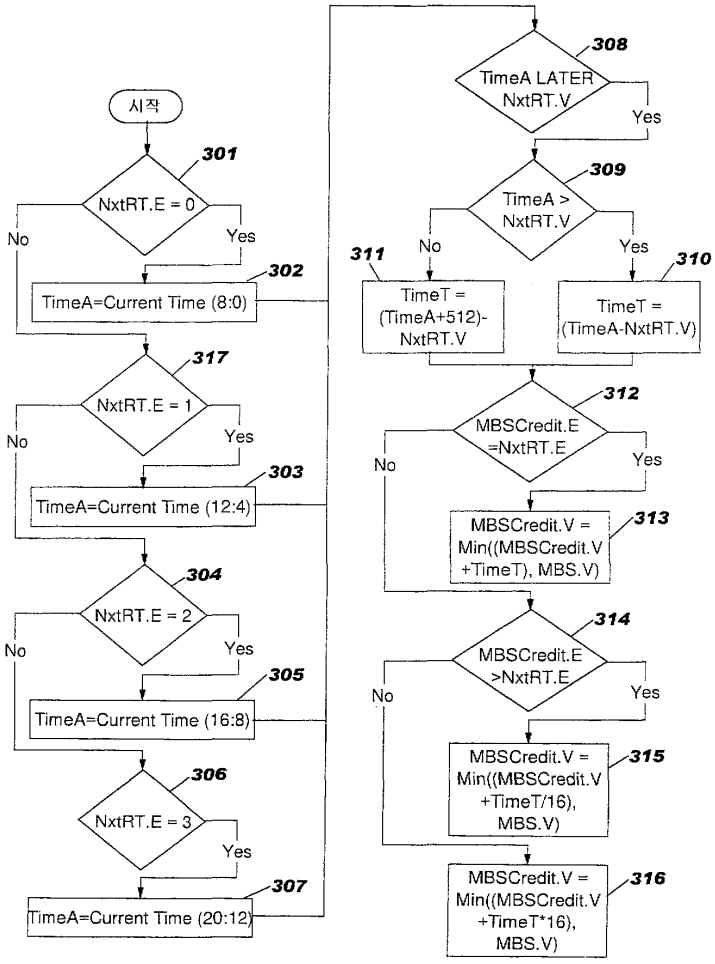
도면 8


시작
1
PSD.E= NxtGT.E
Yes
No
Temp = ((PSD.V * FL)/512)
2
3
PSD.E> NxtGT.E
Yes
No
Temp = ((PSD.V * FL)/32)
4
Temp = ((PSD.V * FL)/8192)
5
6
Temp > 511
Yes
No
슬롯 거리가 현재 스케일링 인수에 대해 너무 크므로, 스케일링 인수를 변경
NxtGT.V = ((Temp + NxtGTT.V)/16)%512
NxtGT.E = NxtGT.E + 1
7
8
Temp < 32
Yes
No
슬롯 거리가 더 작은 스케일링 인수로 변경할 만큼 충분히 작음
9
NxtGT.E=0
No
Yes
NxtGT.V = ((Temp + NxtGT.V)*16)%512
NxtGT.E = NxtGT.E - 1
10
NxtGT.V = (Temp + NxtGT.V)%512
스케일링 인수 불변
11

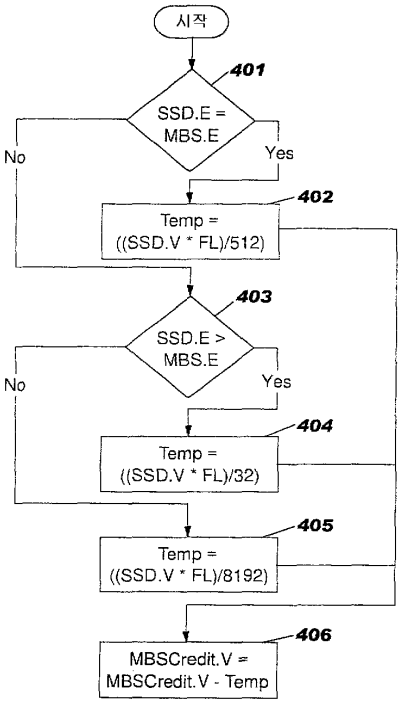
도면 9


시작
21
Temp =
((PSD.V * FL)/512)
22
PSD.E = 0
No
Yes
23
BaseT=Current Time (8:0)
24
PSD.E = 1
No
Yes
25
BaseT=Current Time (12:4)
26
PSD.E = 2
No
Yes
27
BaseT=Current Time (16:8)
28
PSD.E = 3
Yes
29
BaseT=Curret Time (20:12)
30
Temp >511
No
Yes
슬롯 거리가
현재 스케일링 인수에
대해 너무 크므로,
스케일링 인수를 변경
31
NxtGT.V =
((Temp + BaseT)/16)% 512
NxtGT.E = PSD.E +1
32
Temp < 32
No
Yes
슬롯 거리가 더
작은 스케일링 인수로
변경할 만큼 충분히 작음
33
PSD.E = 0
No
Yes
모듈러스 연산은
NxtGT.V값을
유효 슬롯수의 범위
안에 유지
34
NxtGT.V =
((Temp + BaseT)*16)%512)
NxtGT.E = PSD.E - 1
35
NxtGT.V =
(Temp + BaseT)%512
NxtGT.E = PSD.E
스케일링 인수 불변

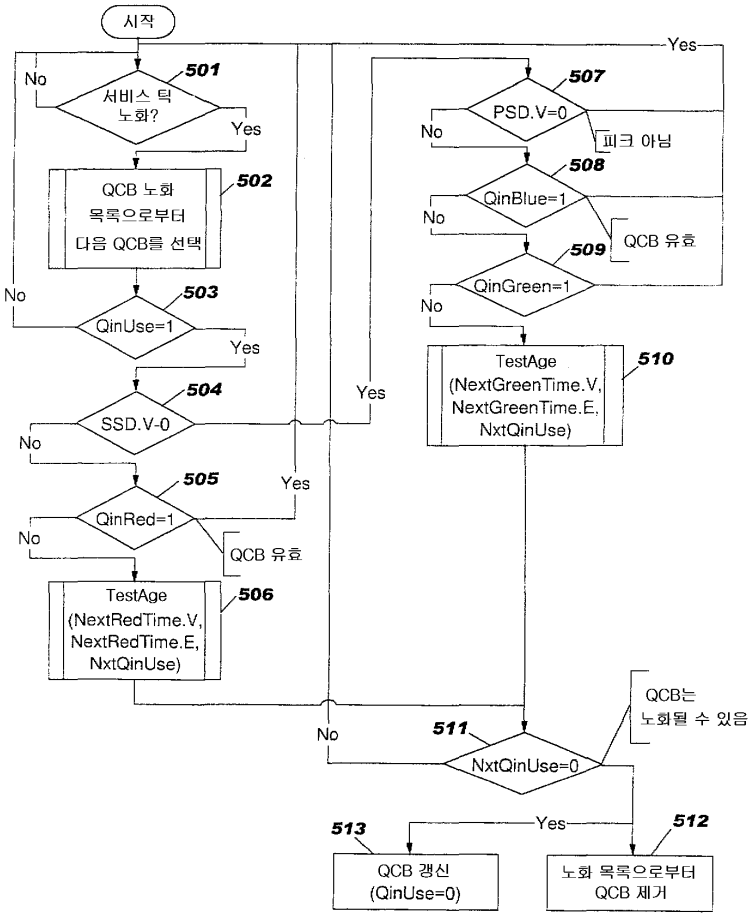
도면 10


시작
301
NxtRT.E = 0
No
Yes
302
TimeA=Current Time (8:0)
317
NxtRT.E = 1
No
Yes
303
TimeA=Current Time (12:4)
304
NxtRT.E = 2
No
Yes
305
TimeA=Current Time (16:8)
306
NxtRT.E = 3
Yes
307
TimeA=Current Time (20:12)
308
TimeA LATER NxtRT.V
Yes
309
TimeA > NxtRT.V
No
Yes
311
TimeT = (TimeA+512)-NxtRT.V
310
TimeT = (TimeA-NxtRT.V)
312
MBSCredit.E =NxtRT.E
No
Yes
313
MBSCredit.V = Min((MBSCredit.V +TimeT), MBS.V)
314
MBSCredit.E >NxtRT.E
No
Yes
315
MBSCredit.V = Min((MBSCredit.V +TimeT/16), MBS.V)
316
MBSCredit.V = Min((MBSCredit.V +TimeT*16), MBS.V)

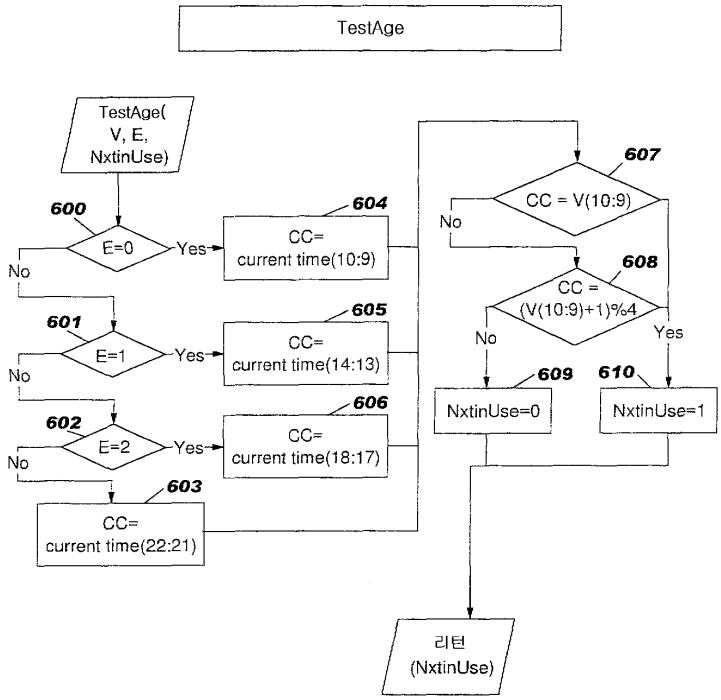
도면 11


시작
401
SSD.E = MBS.E
No
Yes
402
Temp = ((SSD.V * FL)/512)
403
SSD.E > MBS.E
No
Yes
404
Temp = ((SSD.V * FL)/32)
405
Temp = ((SSD.V * FL)/8192)
406
MBSCredit.V = MBSCredit.V - Temp

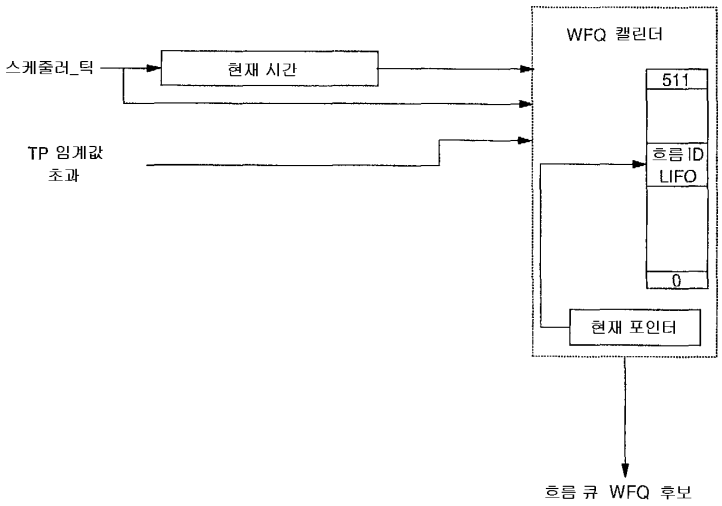
도면 12


시작
501
서비스 탁 노화?
No
Yes
502
QCB 노화 목록으로부터 다음 QCB를 선택
503
QinUse=1
No
Yes
504
SSD.V-0
No
Yes
505
QinRed=1
No
Yes
QCB 유효
506
TestAge (NextRedTime.V, NextRedTime.E, NxtQinUse)
507
PSD.V=0
No
Yes
피크 아님
508
QinBlue=1
No
Yes
QCB 유효
509
QinGreen=1
No
Yes
510
TestAge (NextGreenTime.V, NextGreenTime.E, NxtQinUse)
511
NxtQinUse=0
No
Yes
QCB는 노화될 수 있음
512
노화 목록으로부터 QCB 제거
513
QCB 갱신 (QinUse=0)

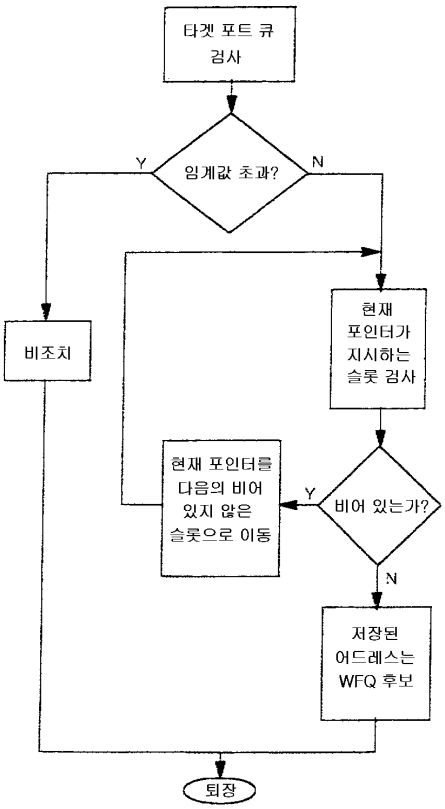
도면 13


TestAge
TestAge(V, E, NxtinUse)
600
E=0
Yes
No
604
CC= current time(10:9)
601
E=1
Yes
No
605
CC= current time(14:13)
602
E=2
Yes
No
606
CC= current time(18:17)
603
CC= current time(22:21)
607
CC = V(10:9)
No
608
CC = (V(10:9)+1)%4
No
Yes
609
NxtinUse=0
610
NxtinUse=1
리턴 (NxtinUse)

도면 14


스케줄러_틱
현재 시간
TP 임계값 초과
WFQ 캘린더
511
흐름 ID LIFO
0
현재 포인터
흐름 큐 WFQ 후보

도면 15


타겟 포트 큐 검사
임계값 초과?
Y
N
비조치
현재 포인터가 지시하는 슬롯 검사
비어 있는가?
Y
N
현재 포인터를 다음의 비어 있지 않은 슬롯으로 이동
저장된 어드레스는 WFQ 후보
퇴장