



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2009년07월17일
(11) 등록번호 10-0908287
(24) 등록일자 2009년07월10일

(51) Int. Cl.

H04L 12/56 (2006.01) H04L 12/26 (2006.01)

(21) 출원번호 10-2004-7016260

(22) 출원일자 2002년07월01일

심사청구일자 2007년06월29일

(85) 번역문제출일자 2004년10월11일

(65) 공개번호 10-2004-0102083

(43) 공개일자 2004년12월03일

(86) 국제출원번호 PCT/IT2002/000430

(87) 국제공개번호 WO 2003/088605

국제공개일자 2003년10월23일

(30) 우선권주장

T002A000326 2002년04월12일 이탈리아(IT)

(56) 선행기술조사문헌

US5404424 A

US6091709 A

US6147970 A

전체 청구항 수 : 총 26 항

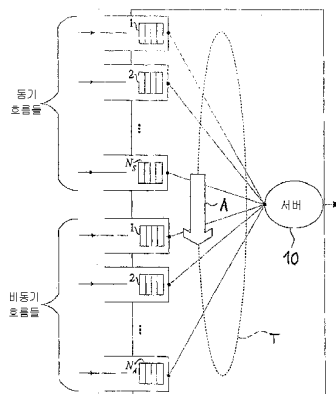
심사관 : 김세영

(54) 동기 및 비동기 패킷 흐름들 간의 공유 자원의 스케줄링

(57) 요약

각 동기 흐름($i = 1, 2, \dots, N_s$)는, 서버가 이동하기 전에 동기 흐름이 서비스될 수 있는 동안의 시간과 관련된 개별 동기 용량값(H_i)과 연관된다. 상기 값은 로컬 할당 기준 또는 글로벌 할당 기준을 따라서 선택될 수 있다. 각 비동기 흐름($j = 1, 2, \dots, N_M$)은, 개별 큐가 서비스될 권리를 갖도록 구성되는 지연을 나타내는 개별 제 1 값과 서버가 이전 사이클에서 개별 큐를 방문했던 순간을 나타내는 또다른 값에 연관된다. 그러면, 동기 흐름(h)에 할당된 각 큐는 상기 동기 용량값에 관련된 시간 동안 서비스되는 한편, 비동기 흐름(i)에 연관된 각 큐는 서버의 방문이 예상 순간 이전에 일어나는 경우에만 서비스된다. 동기 큐들의 서버의 방문(10)은 바람직하게는 2개의 연속적 사이클 동안 일어나서 이용가능한 자원의 사용을 최적화하여야 한다.

도 1



(72) 발명자

스카로네엔리코

이탈리아 아이-10148 토리노 274 비아 레이스 로물
리 텔레콤 이탈리아 소시에떼 퍼 아찌오니 내

스테이지오반니

이탈리아 아이-10148 토리노 274 비아 레이스 로물
리 텔레콤 이탈리아 소시에떼 퍼 아찌오니 내

특허청구의 범위

청구항 1

개별 연관 큐를 발생시키는 여러 정보 패킷 흐름들 사이에 공유되는 서비스 자원을 스케줄링하는 방법으로서, 상기 흐름들은 보증된 최소 서비스 속도(r_i)를 필요로 하는 동기 흐름들($i = 1, 2, \dots, N_s$)과 상기 동기 흐름들에 의해 사용되지 않고 남겨진 상기 자원의 서비스 용량을 사용하는 비동기 흐름들($j = 1, 2, \dots, N_A$)을 포함하고, 상기 방법은 서버(10)를 이용하고, 상기 방법은

- 서버(10)가, 개별 큐에 대한 방문 사이클을 완료하는데 서버(10)가 필요로 하는 시간을 식별하는 타깃 순환 시간값(TTRT)를 기초로 연속 사이클에서 상기 흐름들(i, j)에 연관된 상기 개별 큐를 방문하도록 하는 단계,
- 각 동기 흐름(i)을, 서버가 이동하기 전에 개별 동기 흐름이 서비스될 수 있는 최대 시간을 나타내는 개별 동기 용량값(H_i)과 연관시키는 단계,
- 각 비동기 흐름(j)을, 서비스될 권리를 갖기 위해 개별 큐를 위해 결정되어야 하는 값을 식별하는 제 1 개별 지연값(L_j), 및 서버가 이전 사이클에서 개별 큐에 방문한 순간을 나타냄으로써 상기 개별 큐를 위해 서버의 이전 방문 이후에 경과한 시간을 결정하는 제 2 개별 값(last_visit_time)과 연관시키는 단계,
- 상기 개별 동기 용량값(H_i)에 관련된 최대 서비스 시간 동안 동기 흐름(i)에 연관된 각 큐를 서비스하는 단계;
- 상기 서버의 방문이, 각 비동기 큐를 위한 최대 서비스 시간을 정의하는, 상기 타깃 순환 시간값(TTRT)과 상기 서버(10)의 이전 방문 이후에 경과한 시간 및 누적된 지연 간의 차가 양인 경우에만 비동기 흐름(j)에 연관된 각 큐를 서비스하는 단계; 및

- i) 수식들

$$\sum_{i=1}^{N_s} H_i + \tau_{\max} \leq TTRT$$

$$TTRT \geq \frac{\tau_{\max}}{1 - \sum_{h=1}^{N_s} r_h / C}$$

뿐만 아니라

- ii) 다음 수식 중 적어도 하나

$$H_i = \frac{r_i \cdot TTRT}{C}$$

$$H_i = \frac{(N_A + \alpha) \cdot r_i / C}{N_A + 1 - \sum_{h=1}^{N_s} r_h / C} \cdot TTRT$$

를 만족시킴으로써, i 번째 동기 흐름에 연관된 큐를 위해 상기 개별 동기 용량값(H_i)을 정의하는 단계를 포함하고, 여기서,

- H_i 는 i 번째 동기 흐름과 연관된 큐를 위한 상기 개별 동기 용량값(H_i)이고,

- N_s 는 상기 동기 흐름들의 개수이며,
- N_A 는 상기 비동기 흐름들의 개수이고,
- τ_{max} 는 상기 공유 서비스 자원에 의한 가장 긴 패킷 서비스의 길이이며,
- TTRT는 상기 타킷 순환 시간 값이고,
- C 는 상기 공유 서비스 자원의 서비스 용량이며,

- r_i 는,
$$\sum_{h=1}^{N_s} r_h / C < 1$$
 을 만족하는, i 번째 동기 흐름에 의해 필요한 최소 서비스 속도이고,

- α 는,
$$\sum_{h=1}^{N_s} r_h / C \leq 1 - \alpha$$
 를 만족시키는 파라미터인,

서비스 자원을 스케줄링하는 방법.

청구항 2

제 1 항에 있어서,

각 상기 연속 사이클 동안에, 상기 서버(10)는 상기 동기 흐름들($i = 1, 2, \dots, N_s$)에 연관된 모든 큐들에 대해 이중 스캔을 실행한 다음, 상기 비동기 흐름들 ($j = 1, 2, \dots, N_A$)에 연관된 큐를 방문하는 것을 특징으로 하는 서비스 자원을 스케줄링하는 방법.

청구항 3

제 2 항에 있어서,

- 개별 흐름에 이용가능한 서비스 시간량을 나타내는 또다른 값(Δ_i)를 각 동기 흐름(i)에 연관시키는 단계,
- 상기 이중 스캔의 주 사이클(main cycle) 동안에, 최대의 상기 또다른 값(Δ_i)과 동일한 시간 동안에 동기 흐름(i)에 연관된 각 큐를 서비스하는 단계, 및
- 상기 또다른 값(Δ_i)이 순수 양이라면, 상기 이중 스캔의 부 사이클(minor cycle) 동안, 동기 흐름(i)에 연관된 각 큐의 단 한 패킷만을 서비스하는 단계를 포함하는 것을 특징으로 하는 서비스 자원을 스케줄링하는 방법.

청구항 4

제 3 항에 있어서,

큐가 상기 이중 스캔의 주 사이클 동안에 방문될 때, 동기 용량(H_i)의 상기 개별 값만큼 상기 또다른 값(Δ_i)을 증가시키는 단계를 포함하는 것을 특징으로 하는 서비스 자원을 스케줄링하는 방법.

청구항 5

제 3 항 또는 제 4 항에 있어서,

서비스되는 각 패킷 만큼 전송 시간의 상기 또다른 값(Δ_i)을 감소시키는 단계를 포함하는 것을 특징으로 하는 서비스 자원을 스케줄링하는 방법.

청구항 6

제 3 항에 있어서,

- 큐가 비어 있을 때, 또는
 - 상기 또다른 값(Δ_i)에 의해 표현되는 이용가능 시간이 큐 앞에 있는 패킷을 서비스하기에 충분하지 않을 때,
- 상기 이중 스캔의 주 사이클 동안에 동기 흐름(i)에 연관된 각 큐의 서비스가 종료되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 방법.

청구항 7

제 6 항에 있어서,
개별 큐가 비어 있을 때, 상기 또다른 값(Δ_i)을 리셋하는 단계를 포함하는 것을 특징으로 하는 서비스 자원을 스케줄링하는 방법.

청구항 8

제 3 항에 있어서,
상기 이중 스캔의 부 사이클 동안에 주어진 서비스가 있는 경우 상기 또다른 값(Δ_i)의 서비스 시간을 감소시키는 단계를 포함하는 것을 특징으로 하는 서비스 자원을 스케줄링하는 방법.

청구항 9

제 3 항에 있어서,
- 동기 흐름(i)에 연관된 마지막 큐가 방문되었을 때, 또는
- 상기 동기 흐름들(i)에 연관된 모든 큐들의 용량들(H_i)의 합 이상의 시간이 상기 이중 스캔의 상기 주 사이클의 시작 이후 경과되었을 때,
상기 동기 흐름들(i)에 연관된 모든 큐들의 상기 이중 스캔 동안에, 상기 부 사이클이 종료되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 방법.

청구항 10

제 3 항에 있어서,
상기 또다른 값(Δ_i)를 0으로 초기화하는 단계를 포함하는 것을 특징으로 하는 서비스 자원을 스케줄링하는 방법.

청구항 11

제 1 항 내지 제 4 항 중 어느 한 항에 있어서,
상기 차이가 음인 경우, 비동기 흐름(j)에 연관된 각 상기 큐는 서비스되지 않고, 상기 차이의 값에 상기 지연(L_j)이 누적되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 방법.

청구항 12

제 1 항 내지 제 4 항 중 어느 한 항에 있어서,
- 큐가 비어 있을 때, 또는
- 이용가능 시간이 큐 앞에 있는 패킷을 전송하기에 충분하지 않을 때,
비동기 흐름(j)에 연관된 각 큐의 서비스가 종료되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 방법.

청구항 13

제 1 항 내지 제 4 항 중 어느 한 항에 있어서,
흐름이 활성화될 때, 상기 제 1 개별 값(L_j) 및 상기 제 2 개별 값(`last_visit_time`)은 각각 0 및 현재 사이클

의 개시 순간으로 초기화되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 방법.

청구항 14

개별 연관 큐를 발생시키는 여러 정보 패킷들 사이에 공유되는 서비스 자원을 스케줄링하는 시스템으로서, 상기 흐름들은 보증된 최소 서비스 속도(r_i)를 필요로 하는 동기 흐름들($i = 1, 2, \dots, N_s$)과 상기 동기 흐름들에 의해 사용되지 않고 남겨진 상기 자원의 서비스 용량을 사용하는 비동기 흐름들($j = 1, 2, \dots, N_A$)을 포함하고,

상기 시스템은 연속 사이클에서 상기 흐름들(i, j)에 연관된 개별 큐를 방문할 수 있는 서버(10)를 포함하고, 상기 시스템은,

- 상기 개별 큐의 방문 사이클을 완료하는데 서버(10)가 필요로 하는 시간을 식별하는 타깃 순환 시간값(TTRT)를 결정하는 단계;
- 개별 동기 흐름이 다음으로 이동하기 전에 서비스될 수 있는 최대 시간을 나타내는 개별 동기 용량값(H_i)을 각 동기 흐름(i)에 연관시키는 단계;
- 서비스될 권리를 갖기 위해 개별 큐를 위해 결정되어야 하는 값을 식별하는 제 1 개별 지연값(L_j), 및 서버가 이전 사이클에서 개별 큐에 방문한 순간을 나타냄으로써 상기 개별 큐를 위해 서버(10)의 이전 방문 이후에 경과한 시간을 결정하는 제 2 개별 값(last_visit_time)을 각 비동기 흐름(j)에 연관시키는 단계;
- 상기 개별 동기 용량값(H_i)에 관련된 최대 시간 동안 동기 흐름(i)에 연관된 각 큐를 서비스하는 단계;
- 각 비동기 큐를 위한 최대 서비스 시간을 정의하는, 상기 타깃 순환 시간값(TTRT)와 서버(10)의 이전 방문 이후에 경과한 시간 및 누적된 지연 간의 차가 양인 경우에만 비동기 흐름(j)에 연관된 각 큐를 서비스하는 단계; 및
- i) 수식들

$$\sum_{i=1}^{N_s} H_i + \tau_{\max} \leq TTRT$$

$$TTRT \geq \frac{\tau_{\max}}{1 - \sum_{h=1}^{N_s} r_h / C}$$

뿐만 아니라

- ii) 다음 수식 중 적어도 하나

$$H_i = \frac{r_i \cdot TTRT}{C}$$

$$H_i = \frac{(N_A + \alpha) \cdot r_i / C}{N_A + 1 - \sum_{h=1}^{N_s} r_h / C} \cdot TTRT$$

를 만족시키는 것에 의하여 i 번째 동기 흐름에 연관된 큐를 위한 상기 개별 동기 용량값(H_i)을 정의하는 단계를 수행하도록 구성되며, 여기서

- H_i 는 i 번째 동기 흐름과 연관된 큐를 위한 상기 개별 동기 용량값(H_i)이고,
- N_s 는 상기 동기 흐름들의 개수이며,
- N_A 는 상기 비동기 흐름들의 개수이고,
- τ_{max} 는 상기 공유 서비스 자원에 의한 가장 긴 패킷 서비스의 길이이며,
- TTRT는 상기 타킷 순환 시간 값이고,
- C 는 상기 공유 서비스 자원의 서비스 용량이며,

$$\sum_{h=1}^{N_s} r_h / C < 1$$

- r_i 는, $\sum_{h=1}^{N_s} r_h / C < 1$ 을 만족하는, i 번째 동기 흐름에 의해 필요한 최소 서비스 속도이고,

$$\sum_{h=1}^{N_s} r_h / C \leq 1 - \alpha$$

- α 는, $\sum_{h=1}^{N_s} r_h / C \leq 1 - \alpha$ 를 만족시키는 파라미터인,

서비스 자원을 스케줄링하는 시스템.

청구항 15

제 14 항에 있어서,

각 상기 연속 사이클 동안에, 상기 서버(10)는 상기 동기 흐름들($i = 1, 2, \dots, N_s$)에 연관된 모든 큐들에 대해 이중 스캔을 실행한 다음, 상기 비동기 흐름들 ($j = 1, 2, \dots, N_A$)에 연관된 큐를 방문하는 것을 특징으로 하는 서비스 자원을 스케줄링하는 시스템.

청구항 16

제 15 항에 있어서,

- 개별 흐름에 이용가능한 서비스 시간량을 나타내는 또다른 값(Δ_i)이 각 동기 흐름(i)에 연관되고,
- 상기 이중 스캔의 주 사이클(main cycle) 동안에, 동기 흐름(i)에 연관된 각 큐가 최대 또다른 값(Δ_i)과 동일한 시간 동안에 서비스되고,
- 상기 또다른 값(Δ_i)이 순수 양이라면, 상기 이중 스캔의 부 사이클(minor cycle)동안에, 본 시스템은 동기 흐름(i)에 연관된 각 큐의 단 한 패킷만을 서비스하는 것을 특징으로 하는, 서비스 자원을 스케줄링하는 시스템.

청구항 17

제 16 항에 있어서,

큐가 상기 주(major) 이중 스캔 사이클 동안에 방문될 때, 상기 또다른 값(Δ_i)은 상기 개별 동기 용량값(H_i)만큼 증가되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 시스템.

청구항 18

제 16 항 또는 제 17 항에 있어서,

상기 또다른 값(Δ_i)은 서비스되는 각 패킷의 전송 시간만큼 감소되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 시스템.

청구항 19

제 16 항에 있어서,

- 큐가 비어 있을 때, 또는
 - 상기 또다른 값(Δ_i)에 의해 표현되는 이용가능 시간이 큐 앞에 있는 패킷을 서비스하기에 충분하지 않을 때,
- 본 시스템은, 상기 이중 스캔의 주 사이클 동안에 동기 흐름(i)에 연관된 각 큐의 서비스가 종료되도록 구성되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 시스템.

청구항 20

제 19 항에 있어서,

개별 큐가 비어 있을 때, 상기 또다른 값(Δ_i)이 리셋되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 시스템.

청구항 21

제 16 항에 있어서,

상기 이중 스캔의 부 사이클 동안에 주어진 서비스가 있는 경우, 상기 또다른 값(Δ_i)은 서비스 시간의 양만큼 감소되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 시스템.

청구항 22

제 16 항에 있어서,

- 동기 흐름(i)에 연관된 마지막 큐가 방문되었을 때, 또는
 - 상기 동기 흐름들(i)에 연관된 모든 큐들의 용량들(H_i)의 합 이상의 시간이 상기 이중 스캔의 상기 주 사이클의 시작 이후 경과되었을 때,
- 상기 동기 흐름들(i)에 연관된 모든 큐들에 대한 상기 이중 스캔 동안에, 상기 부 사이클이 종료되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 시스템.

청구항 23

제 16 항에 있어서,

상기 또다른 값(Δ_i)이 0으로 초기화되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 시스템.

청구항 24

제 16 항에 있어서,

상기 차이가 음인 경우, 비동기 흐름(j)에 연관된 각 상기 큐는 서비스되지 않고, 상기 차이의 값에 상기 지연(L_j)이 누적되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 시스템.

청구항 25

제 14 항 내지 제 17 항 중 어느 한 항에 있어서,

- 큐가 비어 있을 때, 또는
 - 이용가능 시간이 큐 앞에 있는 패킷을 전송하기에 충분하지 않을 때,
- 비동기 흐름(j)에 연관된 큐의 서비스가 종료되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 시스템.

청구항 26

제 14 항 내지 제 17 항 중 어느 한 항에 있어서,

흐름이 활성화될 때, 상기 제 1 개별 값(L_j) 및 상기 제 2 개별 값(last_visit_time)은 각각 0 및 현재 사이클의 개시 순간으로 초기화되는 것을 특징으로 하는 서비스 자원을 스케줄링하는 시스템.

명세서

기술분야

<1> 본 발명은 패킷 통신 시스템에 관한 것으로, 특히 공유 자원의 스케줄링 기준, 즉 패킷이 발생될 때마다 자원이 어느 패킷에 할당되어야 하는지를 선택하기 위해 사용되는 기준에 관한 것이다.

<2> 본 발명에서 제공되는 솔루션은, 무선 자원 스케줄링(예: MAC(Medium Access Control level scheduling), 및 네트워크 노드에서의 계산 및 전송 자원의 스케줄링, 예컨대 인터넷 프로토콜 라우터(IP) 상의 다른 서비스 품질과의 흐름 스케줄링 둘 다를 위해 개발되어 왔다. 이하의 설명은 후자의 응용 예를 특히 기초로 하며, 이는 단지 예시로서 주어진 것이고 본 발명의 범위를 한정하지 않는다.

배경기술

<3> 최근 몇년 동안, 패킷 네트워크의 광범위한 응용 및 급속한 발전은, 구세대 패킷 네트워크(전자 메일, 웹서핑 등)에 의해 제공되는 종래의 서비스와 회로 스위칭 네트워크(실시간 비디오, 전화 등)을 위해 미리 확보된 새로운 서비스들을 소위 통합형 서비스 네트워크으로 통합하는 문제를 대두시키고 있다.

<4> 예컨대, 고정 패킷 네트워크 컴포넌트(코어(core) 네트워크)를 구성하기 위한 UMTS와 같은 시스템은 음성 및 데이터 서비스를 동시에 처리해야 하고, 이들이 실시간이든 아니든 새로운 서비스의 개발을 위한 지원을 제공해야 한다.

<5> 따라서, 통합형 서비스 네트워크는 다른 특성을 갖는 트래픽 흐름들을 처리할 수 있어야 하고, 동의된 기간 내에서 보장되어야 하는 사용자 및 서비스 제공자 간의 협의된 성능 인덱스 세트, 즉 적절한 서비스 품질을 각 타입의 흐름에 제공할 수 있어야 한다.

<6> 요구되는 서비스 품질을 제공함에 있어 핵심적 구성요소들 중 하나는 네트워크 노드 상에서 구현되는 스케줄링 시스템, 즉 노드 상에 존재하는 것들로부터 전송되는 패킷을 선택하기 위해 사용되는 시스템이다; 상기 시스템은, 공유 자원의 사용에서의 효율(예컨대, 전송 수단), 다양한 전송 흐름의 처리 및 높은 전송 속도를 필요로 하는 환경에서 사용할 수 있도록 하는 특성, 단순성, 및 여러 타입의 서비스를 제공할 수 있는 용량에 관한 유연성과 같은 현저한 특성을 명확히 구현해야 한다.

<7> 패킷 네트워크에서 일정 레벨의 서비스 품질(또는 QoS)을 보장하는 필요성은 꾸준히 증가하고 있다. 이러한 것은 예컨대, US-A-6 091 709, US-A-6 147 970 또는 EP-A-1 035 751의 문헌에서 알 수 있다.

<8> 실제 본 발명은 산업 발명 특허 출원 T02000A001000 및 이의 대응 출원 PCT/IT01/00536에서 설명된 솔루션의 개량이다.

<9> 이전의 솔루션은 기본적으로, 흐름들이 개별 연관 큐를 발생시키고 서버가 전송 허가를 할 때 서비스되는 여러 정보 패킷 흐름들 간에 공유되는 서비스 자원의 스케줄링에 적용된다.

발명의 상세한 설명

<10> 흐름들은 최소 서비스 속도 보장을 필요로 하는 동기 흐름들과, 동기 흐름에 의해 사용되지 않는 자원의 서비스 용량을 사용하는 비동기 흐름들로 대별된다. 당해 솔루션은 다음을 포함한다.

<11> - 연속 사이클 내의 흐름들과 연관되는 큐를 방문하는 서버를 제공하고, 큐 방문 사이클을 완료하기에 서버가 필요한 시간을 식별하는 TTRT라고 불리우는 타깃 토큰 순환 시간(또는 "회전")을 각 큐에 부여함.

<12> - 각 동기 흐름을, 전송 허가가 서버에 의해 철회되기 전에 동기 흐름이 서비스 될 수 있는 최대 시간을 나타내는 동기 용량값과 연관시킴.

<13> - 각 동기 흐름을, 개별 큐가 서비스될 권리를 갖기 위해 결정되어야 하는 지연을 나타내는 제 1 지연 값, 및 서버가 이전 사이클에서 개별 큐를 방문했던 순간을 나타내어 서버의 이전 방문에서부터 경과한 시간을 결정하는 다른 값(last_token_time)을 합한 것과 연관시킴.

- <14> - 전송한 동기 용량값과 동일한 최대 시간 길이동안 동기 흐름에 연관된 각 큐를 서비스함.
- <15> -서버의 방문이 예상 순간 이전에 일어나는 경우에만 비동기 흐름에 연관된 각 큐를 서비스함. 이전에 일어나는 경우는 상기 TTRT 시간과 서버의 이전 방문 이후에 경과한 시간 및 누적된 지연 간의 차이로부터 얻어진다.
- <16> 상기 차이가 양이면, 이는 각 비동기 흐름에 연관된 각 큐를 위한 최대 서비스 시간을 정의한다.
- <17> 이상에서 언급된 솔루션은 작동의 관점에서 완전히 만족적인 것으로 증명되었다. 그러나, "출원인"에 의해 얻어지는 경험은 본 솔루션이 본 발명에서 설명되는 바와 같이 더욱 발전되고 향상될 것이라는 것을 제시하였다.
- <18> 본 발명은 다음의 측면에서 특히 적용된다.
- <19> - 다른 타입의 서비스를 제공하는 한편 낮은 계산 비용을 유지하는 가능성: Intserv(IETF 사양에서와 같은 통합형 서비스(Integrated Services)) 또는 Diffserv(IETF 사양에서와 같은 차등형 통합형 서비스(Differentiated Integrated Services))를 갖는 IP 네트워크들과 같은 사용자들 위해, 또는 MAC 레벨 스케줄링 알고리즘(W-LAN 시스템들, 제3세대 무선-이동 서비스들)과 같은 무선 자원 스케줄링 시스템들을 위해 서비스 품질을 보장해야 하는 컴퓨터 네트워크 응용을 위한 중요한 특성;
- <20> - 동기 트래픽을 위한 각 흐름의 버퍼들의 최대 점유, 최대 큐등록(queuing) 지연 및 다양한 흐름들의 비트 전송률의 보장 가능성;
- <21> - 특히 서비스 통합 네트워크에 있어서, 동시에 2개의 다른 타입의 서비스들을 제공하기 위한 용량, 속도 보장(동기 흐름에 적당) 및 공정 큐등록(비동기 흐름에 적당)에 관한 유연성;
- <22> - 전송 흐름의 분리 가능성, 즉 이는 단일 흐름에 제공되는 서비스가 다른 흐름들의 존재 및 동작으로부터 독립적으로 되도록 함;
- <23> - 전송될 패킷을 선택하기 위해 필요한 동작들의 수에 관한 낮은 계산 복잡도; 이런 특징은 하드웨어로 구현 가능한 관점에서도 고속 전송 속도 및 다수 전송 흐름들의 처리를 필요로 하는 환경에서 사용될 수 있도록 함;
- <24> - 복잡한 절차에 의존해야 함 없이 자원을 재분배함으로써 동작 파라미터들(예컨대, 존재하는 흐름들의 수)의 변화를 처리할 수 있다는 의미에서 적응성;
- <25> - 분석적 설명용이성, 즉 이는 시스템 동작의 완전한 분석적 설명을 제공하는데, 이는 서비스 품질 측정이 시스템 파라미터에 관련될 수 있도록 한다.
- <26> 다른 중요한 특성은 공평성, 즉 속도 보장 서비스를 수신하는 전송 흐름들과 공정 큐등록 서비스를 수신하는 전송 흐름들 모두를 동일한 방식으로 다루어서, 다른 길이의 패킷들이 있는 경우에도 요구되는 것에 비례하는 서비스 레벨을 각각에게 제공하는 가능성이다.
- <27> 본 발명의 목적은 전송된 태양에 특별한 관심을 가지고 이전에 언급된 이미 공지된 솔루션을 더욱 개발하는 것이다.
- <28> 본 발명에 따르면, 상기 목적은 이후의 청구의 범위에서 특히 언급되는 특성들을 갖는 스케줄링 방법을 사용하여 달성될 수 있다.
- <29> 본 발명은 관련 시스템을 또한 언급한다.
- <30> 간략히 설명하자면, 본 발명에서 주어진 솔루션은 본 특허출원에서 소개되는 명칭, 즉 패킷 시간 토큰 서비스 규칙(Packet Timed Token Service Discipline) 또는 PTTSD로 정의될 수 있는 스케줄링 시스템을 작동시킨다.
- <31> 현재, 상기 스케줄링 시스템은 패킷-컴퓨터 네트워크 스위칭 노드 상에서 동작하도록 설계되고, 단일 전송 채널을 여러 전송 흐름들로 다중화할 수 있다.
- <32> 본 시스템은 2개의 다른 타입의 서비스를 제공한다: 보장된 최소 서비스 속도를 필요로 하는 전송 흐름들(이하, "동기 흐름들")에 적당한 속도 보장 서비스, 및 최소 서비스 속도에 대한 어떤 보장도 필요로 하지 않지만 이용 가능한 더 큰 전송 용량으로부터 이익을 받는 전송 흐름들(이하, "비동기 흐름들")에 적당한 공정 큐잉(fair-queueing) 서비스. 본 시스템은 후자를 제공하나, 동기 흐름들에 의해 사용되지 않은 전송 용량의 공평 공유를 갖는다.
- <33> 노드 상의 각 전송 흐름 입력으로부터의 트래픽은 자신의 큐(동기 또는 비동기 큐)에 삽입되어 전송되도록 취해진다. 서버는 고정 순환 순서로 큐를 방문하고, 각 방문 때마다 정확한 타이밍 제한에 따라서 확립된 서비스 시

간을 각 큐에 부여한다.

초기에 서버는 1번 회전하는 동안에 동기 큐를 2번 방문하여 주 사이클과 부 사이클 또는 복원 사이클을 완성하며 다음으로 비동기 큐를 방문하기 위해 이동한다.

실시예

<35> 본 발명에서 제시되는 스케줄링 시스템은 단일 전송 채널을 여러 전송 흐름들로 다중화할 수 있다.

<36> 본 시스템은 2개의 다른 타입의 서비스를 제공한다: 보장된 최소 서비스 속도를 필요로 하는 전송 흐름들(이하, i 동기 흐름들. 단, $i = 1, 2, \dots, N_S$)에 적절한 속도 보장 서비스, 및 서비스 속도에 어떤 보장도 필요로 하지 않는 전송 흐름들(이하, j 비동기 흐름들. 단, $j = 1, 2, \dots, N_A$)에 적절한 최선 노력 서비스(best-effort service). 본 시스템은 후자를 제공하지만, 동기 흐름들에 의해 사용되지 않는 전송 용량의 공평한 공유를 갖는다.

<37> N_S 및 N_A 는 음이 아닌 정수이고, 각 동기 흐름 $i = 1, \dots, N_S$ 는 최소 서비스 속도(r_i)와 동일한 서비스 속도를 필요로 하고, 동기 흐름에 의해 요구되는 서비스 속도들의 합은 채널의 용량 C 를 넘지 않아야 함

$$\left(\sum_{i=1}^{N_S} r_i \leq C \right) \text{을 가정해야 한다.}$$

<38> 노드 상에 입력된 각 전송 흐름으로부터의 트래픽은 자신의 큐에 삽입(동기 또는 비동기 큐들은 나중에 설명된다)되어 전송되도록 취해진다. 서버(10)은 고정 순환 순서(이상적으로는 제적 T 와 화살표 A 를 갖는 도면에서 도시된 순서)로 큐들을 방문하여, 매 방문 때마다 정확한 타이밍 제한에 따라 확립된 서비스 시간을 각 큐에 부여한다.

<39> 본 발명에서의 방법은 큐로 순환적으로 방문하기 전에 초기화 단계를 포함한다. 본 방법이 이하 설명된다.

<40> 초기화

<41> 먼저, 시스템에 동작 조건들에 관한 정보를 주는 것이 필요하다: 얼마나 많은 동기 흐름들이 있는지(일반적으로 N_S), 각 동기 흐름에 의해 요구되는 전송속도는 얼마인지, 얼마나 많은 비동기 흐름들이 있는지, 타깃 순환 시간(TTRT), 즉 서버가 모든 큐들을 한번 방문하는 동안의 완전한 사이클이 얼마나 오래동안 지속되어야 하는지에 관한 정보가 주어져야 한다.

<42> 동기 흐름들

<43> 적절한 할당 정책에 따라서, 각 동기 흐름들 $i, i = 1, \dots, N_S$ 는 변수 H_i (동기 용량)에 연관되고, 상기 H_i 는 서버가 전송 허가를 취하기 전에 동기 흐름의 트래픽이 전송될 수 있는 최대 시간을 나타낸다. 가능한 할당 정책들이 이하 설명된다. 초기에 0인 변수 Δ_i 가 각 동기 흐름에 연관되고, 흐름에 이용가능한 전송 시간 양을 저장한다.

<44> 비동기 흐름들

<45> 각 비동기 흐름 $j, j = 1, 2, \dots, N_A$ 는 2개의 변수, 제 1 개별 지연값(L_j) 및 제 2 개별 값(last_visit_time _{j})과 연관된다; 첫 번째 변수는 비동기 큐 j 가 서비스될 권리를 갖기 위해 결정되어야 하는 지연 또는 지체를 저장한다; 두 번째 변수는 서버가 이전 사이클에서 비동기 큐 j 를 방문한 순간을 저장한다. 이들 변수들은 0과 흐름이 활성화될 때 진행 중인 순환이 시작되었던 순간으로 각각 초기화된다.

<46> 이러한 방식의 진행은 비동기 흐름들이 반드시 시스템 개시에서만 아니라 어떤 순간에도 활성화될 수 있다는 것을 의미한다.

<47> 주 사이클(major cycle) 동안 $i = 1, \dots, N_S$ 를 갖는 일반 동기 큐 i 로의 방문

<48> 동기 큐는 변수 Δ_i 의 최대값과 동일한 시간동안 서비스될 수 있다. 상기 변수는 큐가 주 사이클 동안에 방문될 때 H_i (초기화동안에 결정된 값)만큼 증가되고, 전송되는 각 패킷의 전송시간만큼 감소된다.

<49> 주 사이클동안의 큐의 서비스는, 큐가 비어있을 때(변수 Δ_i 가 리셋되는 경우), 또는 (Δ_i 의 현재 값에 의해 표현되는) 이용가능 시간이 큐 앞에 있는 패킷을 전송하기에 충분하지 않을 때 종료된다.

<50> 부 사이클(minor cycle) 동안에 일반 동기 큐 $i, i = 1, \dots, N_s$ 로의 방문

<51> 변수 Δ_i 가 순수 양의 값인 경우 부(또는 복원) 사이클 동안에 동기 큐는 단 하나의 패킷을 전송할 수 있다. 전송이 일어나면, 변수 Δ_i 는 전송시간만큼 감소된다.

<52> $j = 1, 2, \dots, N_s$ 를 갖는 일반 비동기 큐 j 로의 방문

<53> 서버의 방문이 예상 순간 이전에 일어나는 경우에만 비동기 큐는 서비스될 수 있다. 서버의 방문이 있을 것인지를 사전에 계산하기 위해, 타겟 순환 시간 TTRT로부터 이전 방문 이후 경과한 시간 및 누적된 지연 L_j 를 뺀다.

<54> 이 차이가 양이면, 이는 비동기 큐가 서비스될 권리를 갖는 시간이고, 이 경우 변수 L_j 는 리셋된다.

<55> 이 차이가 음이면, 서버는 지연되고, 큐 j 는 서비스될 수 없다; 이 경우 지연은 변수 L_j 에 저장된다. 큐가 비어있을 때, 또는 (패킷이 전송되는 때마다 감소되는) 이용가능 시간이 큐 앞에 있는 패킷을 전송하기에 충분하지 않을 때, 비동기 큐 서비스는 종료된다.

<56> 순환 동안의 방문 시퀀스

<57> 한 순환 동안에 모든 동기 큐들(주 및 부 사이클들)에 대해 이중 스캔이 행해진 다음, 비동기 큐들이 방문된다. 부 사이클은 다음 경우 중 하나가 일어날 때 종료된다.

<58> - 마지막 동기 큐가 방문되었을 경우;

<59> - 주 사이클 개시 이후 모든 동기 큐들의 용량의 합보다 크거나 같은 기간이 경과한 경우.

<60> 분석적 보장

<61> 다음의 부등식에 의해 동기 용량들은 타겟 순환 시간 TTRT 및 가장 긴 패킷 τ_{\max} 의 전송 길이에 링크되는데, 이는 항상 다음과 같아야 한다:

$$\sum_{i=1}^{N_s} H_i + \tau_{\max} \leq TTRT \quad (1)$$

<62> 동기 흐름들을 위한 최소 전송 속도

<63> 전제 (1)에 있어서, 본 명세서에서 설명되는 시스템은 다음의 규준화된 전송 속도가 각 동기 흐름을 위해 보장될 것이라는 것을 보장한다.

$$\gamma_i = \frac{N_A + 1}{N_A + \sum_{h=1}^{N_s} X_h + \alpha} \cdot X_i$$

<64> 단, $X_i = H_i/TTRT$

<65> $\alpha = \tau_{\max}/TTRT$

<66> 그리고, 일반 동기 큐 i 가 전혀 비워지지 않는 임의의 시간 $[t_1, t_2]$ 가 주어지면, $[t_1, t_2]$ 에서 큐 i 로부터 수신된 서비스 시간 $W_i(t_1, t_2)$ 는 다음의 부등식을 만족하는 것 또한 가능하다.

<69>
$$\gamma_i^*(t_2 - t_1) - W_i(t_1, t_2) \leq A_i < \infty \quad (2)$$

<70> 단,

<71>
$$\Lambda_i = \begin{cases} H_i \cdot (2 - \gamma_i) + (1 + \gamma_i) \cdot \tau_i & \text{단 } H_i \geq \tau_i \\ \tau_i + 2 \cdot H_i & \text{단 } H_i < \tau_i \end{cases}$$

<72> 그리고, τ_i 는 흐름 i에 대한 가장 긴 패킷의 전송 시간이다.

<73> 전술한 수식 (2)은, 본 명세서에서 설명되는 타입의 i 동기 흐름 시스템에 의해 공급되는 서비스는 동일 흐름이

본 발명에 설명된 시스템에 의해 다루어지는 채널의 용량의 γ_i 배와 동일한 용량을 갖는 개인 전송 채널의 소유자라면 경험하는 서비스와는 Λ_i 이상 다르지는 않는다는 것을 확립한다. 따라서, Λ_i 는 이상적 상황에 대한 최대 서비스 차이를 나타낸다.

<74> 따라서, 동기 흐름은 대기 시간(latency)이라고 불리는 파라미터를 특징으로 할 수 있는데, 이는 다음과 같이 계산된다:

<75>
$$\Theta_i = \begin{cases} \left(2 + \frac{\tau_i}{H_i} \right) \frac{N_A TTRT + \tau_{\max} + \sum_{h \in S} H_i}{N_A + 1} + \tau_i - H_i, & \text{단 } H_i \geq \tau_i \\ \left(2 + \frac{\tau_i}{H_i} \right) \frac{N_A TTRT + \tau_{\max} + \sum_{h \in S} H_i}{N_A + 1}, & \text{단 } H_i < \tau_i \end{cases}$$

<76> 또는, $N_A \rightarrow \infty$ 인 경우

<77>
$$\Theta_i^* = \begin{cases} \left(2 + \frac{\tau_i}{H_i} \right) TTRT + \tau_i - H_i, & \text{단 } H_i \geq \tau_i \\ \left(2 + \frac{\tau_i}{H_i} \right) TTRT, & \text{단 } H_i < \tau_i \end{cases}$$

<78> 본 명세서에서 설명된 솔루션을 구현하는 스위칭 노드가 주워지는 경우, 상기 노드 상의 동기 흐름에 대한 트래픽 입력이 소위 "리키 버킷(leaky-bucket)"의 파라미터 (σ , ρ)에 의해 제한되면, 다음의 보장이 주어질 수 있다:

<79> a) 동기 흐름에 대한 단일 노드 상의 최대 지연

<80> 각 패킷은 다음보다 크지 않은 지연을 갖는다:

<81>
$$D = \sigma / \rho + \Theta_i$$

<82> b) 동기 흐름에 대한 노드 상의 최대 메모리 점유

<83> 동기 흐름 패킷 내의 패킷들에 의해 점유되는 메모리 양은 다음과 같다:

<84>
$$B = \sigma + \rho \Theta_i$$

<85> c) 동기 흐름에 대한 N 노드들의 라우트 상의 최대 지연

<86> $\Phi_1 \dots \Phi_N$ 을 본 명세서에서 설명되는 시스템을 구현하는 N 스위칭 노드들이라고 하고, Θ_i^j 를 Φ_j 노드들 각각

에서 계산되는 대기시간이라고 하면,

$$\overline{\Theta}_i = \sum_{j=1}^N \Theta_i^j$$

<87>

<88> 제 1 노드 상의 트래픽 입력이 리키 버킷 파라미터들 (σ , ρ)에 의해 제한되면, N 노드들을 가로지르기 위한 패킷에 대한 최대 지연에 대한 상한을 정의하는 것이 가능한 경우에 있어서, 상기 상한은 다음과 같다:

$$D_N = \sigma/\rho + \overline{\Theta}_i$$

<89>

<90> 값 $\Theta_i^* \geq \Theta_i$ 은 상기 3개의 보장들 a), b), c)의 각각에서 사용될 수 있다; 이는, 활성 비동기 흐름들의 수에 의존하지 않는 한계들이 계산될 수 있다는 것을 의미한다.

<91> 파라미터 선택

<92> 동기 흐름들이 요구된 것 이상의 최소 서비스 속도를 받는 것을 보장하는 능력은 동기 용량들 H_i , $i = 1, \dots, N_s$ 의 정확한 선택에 따른다. 각 동기 흐름 i 가 최소 전송 속도 r_i 를 필요한 것으로 가정하면, 다음의 부등식을 만족하는 동기 용량들을 할당하는 것이 필요하다.

$$\gamma_i \geq r_i/C$$

(3)

<93>

<94> 로컬 및 글로벌 할당으로 각각 불리우는 2개의 다른 방식에 따라서, 본 명세서에 설명된 솔루션은 동기 용량들을 할당한다.

<95> 로컬 할당

<96> 동기 용량들은 다음과 같이 선택된다:

$$H_i = \frac{r_i \cdot TTRT}{C}$$

<97>

<98> 이러한 방식에 있어서, 요구되는 전송 속도들이 다음의 부등식을 만족하면 부등식(1)이 만족된다:

$$\sum_{h=1}^{N_s} r_h/C \leq 1-\alpha$$

(4)

<99>

<100> 각 동기 흐름은 다음과 같은 표준화된 서비스 속도로 보장된다:

$$\gamma_i = \frac{[N_A + 1] \cdot r_i/C}{N_A + \sum_{h=1}^{N_s} r_h/C + \alpha}$$

(5)

<101>

<102> 수식(5)에 의해 주어진 γ_i 의 값은 부등식(3)을 만족한다.

<103> 글로벌 할당

<104> $N_A > 0$ 을 필요로 하는 본 방식에 따르면, 동기 용량들은 다음과 같이 선택된다:

$$H_i = \frac{(N_A + \alpha) \cdot r_i / C}{N_A + 1 - \sum_{h=1}^{N_s} r_h / C} \cdot TTRT$$

<105>

<106> 글로벌 할당 방식에 있어서, 요구되는 전송 속도의 합 또한 부등식 (4)를 만족해야한다. (4)가 만족되면, 동기

흐름의 규준화된 서비스 속도는 $\gamma_i = r_i / C$ 이다.

<107> 글로벌 방식은 로컬 방식보다 채널의 전송 용량을 더 많이 사용하는 것을 보장한다. 이는, 비동기 흐름 전송을 위해 더 많은 대역폭을 남겨두어 동기 흐름들에 적은 용량을 할당하기 때문이다.

<108> 한편으로는, 글로벌 방식의 사용은 모든 동기 용량이 시스템 내에 있는 흐름들(동기 또는 비동기)의 개수가 변화할 때마다 재계산되어야 한다는 것을 의미한다; 그러나, 로컬 방식의 사용은 용량들이 시스템 내의 흐름들의 개수와 독립적으로 확립될 수 있다는 것을 의미한다.

<109> TTRT의 선택

<110> 다음의 방식은 본 발명에 따른 솔루션에서 TTRT의 선택을 보여주기 위해 주어질 수 있다.

<111> 다음의 부등식을 만족하는 요구되는 전송 속도를 갖는 동기 흐름 세트가 주어진다면

$$\sum_{h=1}^{N_s} r_h / C < 1$$

<112>

<113> TTRT는 다음의 부등식에 따라서 선택되어야 한다:

$$TTRT \geq \frac{\tau_{\max}}{1 - \sum_{h=1}^{N_s} r_h / C}$$

<114>

<115> 아래에서 분석적으로 설명되는 유사부호(pseudo-code)는 본 발명에서 주어지는 바와 같은 시스템의 동작을 설명한다.

<116> 흐름 초기화

<117> Sync_Flow_Init (synchronous flow i)

<118> { $\Delta_i = 0;$

<119> Select_synchronous_bandwidth $H_i;$

<120> }

<121> Async_Flow_Init (asynchronous flow j)

<122> { $L_j = 0;$

<123> last_visit_time_j = start_of_curr_revolution;

<124> }

<125> 주 사이클동안 일반 동기 큐 $i, i = 1, \dots, N_s$ 로의 방문

```

<126> Major_Cycle_Visit (synchronous flow i)
<127> {       $\Delta_i += H_i$ ;
<128>         q= first_packet_tranmission_time;
<129>         while ((  $\Delta_i \geq q$ ) and (q > 0))
<130>             {      transmit_packet (q);
<131>          $\Delta_i -= q$ ;
<132>                 elapsed_time += q;
<133>             }
<134>         if (q=0)  $\Delta_i = 0$ ;
<135>     }
<136> 부 사이클동안 일반 동기 큐 i, i = 1...Ns로의 방문
<137> Minor_Cycle_Visit (synchronous flow i)
<138> {      q= first_packet_tranmission_time;
<139>         if (q > 0))
<140>             {      transmit_packet (q);
<141>          $\Delta_i -= q$ ;
<142>                 elapsed_time += q;
<143>             }
<144>         if (q=0)  $\Delta_i = 0$ ;
<145>     }
<146> 일반 비동기 큐 i, i = 1... Ns로의 방문
<147> Async_Flow_Visit (asynchronous flow j)
<148> {      t= current_time;
<149>         earliness= TTRT-Lj - (t-last_visit_timej);
<150>         if (earliness > 0)
<151>             {      Lj= 0;
<152>                 transmit_time= earliness;
<153>                 q= first_packet_transmission_time;
<154>                 while ((transmit_time>=q) and (q > 0))
<155>                     {
<156>                         transmit_packet (q);
<157>                         transmit_time -= q;
<158>                     }

```



```

<159>         }
<160>         else Lj = -earliness;
<161>         last_visit_timej = t;
<162>     }
<163>     순환동안의 방문 시퀀스
<164>     PTSD revolution ()
<165>     {         elapsed_time= 0;
<166>         for (i=1 to Ng) Major_Cycle_Visit (i);
<167>         i= 1;
<168>         while((elapsed_time<sum(Hh)) and (i<=Ng))
<169>         {
<170>             if (Δi>0) Minor_Cycle_Visit (i);
<171>             i ++;
<172>         }
<173>         for (j=1 to Nk) Async_flow_visit (j);
<174>     }

```

<175> 본 발명의 범주를 벗어나지 않고 이상의 자세한 사항들은 변경될 수 있음이 자명하다.

산업상 이용 가능성

<176> 상기 내용 중에 포함되어 있음.

도면의 간단한 설명

<34> 본 발명에 대한 이후 설명은, 본 발명을 제한하지 않는 첨부된 도면을 참조로 주어지는데, 상기 도면은 본 발명에 따라서 작동하는 시스템의 동작 기준을 설명하는 단일 블록 다이어그램을 포함한다.

도면

도면1

