

화상 회의를 위한 온-라인 프레임 스케줄링 방법

*안성용^o *이정아 **심재홍
*조선대학교 컴퓨터공학부
**조선대학교 인터넷소프트웨어공학부
{dis^o, jalee, jhshim}@chosun.ac.kr

A On-Line Frame Scheduling Method for Video Conferencing

SeongYong Ahn^o *JeongA Lee **JeaHong Shim
*School of Computer Engineering, Chosun University
**School of Internet Software Engineering, Chosun University

요 약

화상회의 시스템에서 원활한 동영상의 전달을 위해서는 주어지는 처리시간에 대하여 그에 상응하는 품질을 보장하는 프레임 스케줄링 방법이 필수적이다. 영상 복원 작업에 주어지는 처리시간이 증가함에 따라 그 실행 결과로 얻어지는 영상의 품질이 함께 증가할 것이 요구된다. 본 논문에서는 영상회의 시스템에서의 압축된 영상의 복원 과정에서 프로세서의 처리시간에 따른 결과 영상의 품질에 대한 QoS 함수를 도출하고 이를 IRIS 스케줄링 방안에서 적용하는 온-라인 스케줄링 알고리즘을 제안한다. 본 논문에서 제시하는 알고리즘에 대한 모의실험의 결과는 주어진 시스템 성능에 대하여 영상회의에 참여한 모든 화자 영상의 총체적 화질을 향상시킬 수 있음을 보여준다.

1. 서 론

영상회의 시스템과 같은 연속 미디어 서비스에서는 프레임 레이트와 디스플레이 지연을 자기기 위한 방안을 미디어 내 동기화(Intra-stream Synchronization)라 정의하여 다양한 기술을 적용하고 있다[1]. 대개의 시스템에서 미디어 내 동기화를 위하여, 미디어 버퍼를 두고 미디어의 연속성 문제를 해결하고 있다. 이때, 가장 중요한 것은 미디어 버퍼내의 복수의 영상 중에서 어느 영상이 버퍼지느냐에 따라 디스플레이 되는 영상의 자연스러움의 정도가 많이 달라지기 때문에 프레임을 골라내는 작업은 매우 중요하여 여러 연구에서 I-Policy, E-Policy, S-Policy 등 다양한 방법이 소개되고 있다[2][3].

전체적인 시스템의 성능을 상대적으로 향상시키는 방법은 각 프레임에 대한 처리 시간을 절약하는 방법이다. 즉, 프레임의 처리 시간을 줄여 줄 경우 충분한 프레임 처리시간을 제공 받을 경우에 보장되는 최상의 화질을 얻을 수 없을지라도 납득할 만한 화질을 제공 받을 수 있다면 프레임의 처리시간을 줄여 줌으로써 시스템이 처리 가능한 프레임의 수를 늘릴 수 있다. 그렇게 하려면 프레임 처리하는 작업은 필요에 따라 처리를 완결하기 전이라도 강제로 중단될 수 있어야 하며, 또한 주어지는 처리시간에 대하여 그에 상응하는 결과 값의 품질이 보장되어야 한다. 즉, 영상 복원 작업에 주어지는 처리시간이 증가함에 따라 그 실행 결과로 얻어지는 영상의 품질이 함께 증가할 것이 요구된다. 즉, 결과 영상의 품질을 QoS 함수라 표현할 때, QoS 함수를 그래프로 나타내면 실행시간에 대하여 비감소 불룩한 형태를 가진다[4].

본 논문에서는 영상회의 시스템에서의 압축된 영상의 복원 과정에서 프로세서의 처리시간에 따른 결과 영상의 품질에 대한 QoS 함수를 도출하고 이를 IRIS 스케줄링 방안[6][7]에 적용하여 모의 실험함으로써 주어진 시스템 성능에 대하여 영상회의에 참여한 모든 화자 영상의 총체적 화질을 향상시킬 수 있음을 입증하였다.

본 논문의 구성은 다음과 같다. 우선, 제2장에서는 영상회의 시스템에서 스트림의 특성에 대하여 설명하고 제3장에서 영상회의 시스템의 프레임율 효율적으로 스케줄링 하기 위한 방안을 모색

하며, 제4장에서는 실제의 스케줄링 방안에서 이를 적용한 예와 그 결과를 보인다. 마지막으로 제5장에서 논문의 결론을 내린다.

2. Video Conferencing Stream의 특성

그림 1은 주어진 영상에 대하여 다섯 단계에 의하여 나타나는 14개의 말릿 블록을 생성하는 Wavelet 방안에 의하여 압축된 후 복원된 영상을 보여준다. 그림1에서는 최고주파 서브밴드를 위한 두개의 말릿 블록을 제외한 12개의 말릿 블록에 대한 영상만을 표시하였다. 그림에서 보는 바와 같이 복원에 사용된 말릿 블록의 수가 증가함에 따라 영상의 화질도 함께 증가함을 볼 수 있다. 주목할 만한 것은 여섯 개의 말릿 블록만을 가지고 복원된 영상일지라도 납득할 만한 화질을 제공한다는 것이다.

그림 1에 나타난 각 영상의 화질을 QoS 함수로 표현할 경우 말릿 블록의 증가에 따르는 영상의 화질 변화를 그림 2에 나타내었다. 여기서 사용한 QoS 함수는 $(S-N)/S = 1-N/S$ 로 여기서 N/S는 원 영상과 복원 영상의 신호 대 잡음비의 역수로서 S는 신호 성분율, N은 잡음 성분을 표현하고 있다. 이 때, S는 원영상의 신호 성분율로서 영상이 정해지면 함께 정해지는 상수로 나타나며 N은 말릿 블록의 수에 따르는 변수로서 말릿 블록의 수가 증가함에 따라 감소하는 특성을 가진다. 즉, 화질이 좋아짐에 따라서 N 값은 감소하게 된다. 그림에서 보는 바와 같이 화질의 변화는 말릿 블록 수의 증가에 대하여 점진적으로 증가하는 불룩한 형태의 그래프로 나타나고 있다.

영상의 처리 결과가 주어진 처리 시간의 증가에 따라 점진적으로 증가하는 불룩한 형태를 가지므로, QoS 함수의 그래프는 2차 함수나 지수함수로 모델링 할 수 있다. 본 논문에서는 그림 2의 그래프를 $f(x) = y - ae^{-(x-x_0)/n}$ 의 지수함수로 모델링 하였으며 그 결과 $f(x) = 0.97 - 0.5e^{-0.25(x-1)}$ 을 얻었다. 이는 시간에 대한 위상 천이를 무시하면 $f(x) = 1 - ae^{-ux}$ 의 형태로 표현되어 a 및 u의 값에 따라 최소 값으로 0.2 ~ 0.5의 수치를 가지며 최대값 1에 접근하는 그래프로 묘사할 수 있다.



그림 1 말릿 블록의 수를 증가하면서 복원된 영상

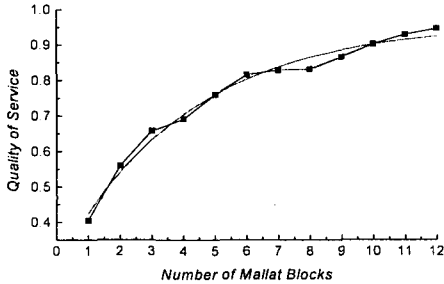


그림 2 말릿 블록 수의 증가에 따른 화질 변화

3. Video Conferencing Stream을 위한 Scheduling 방안

프레임 스케줄링 알고리즘의 목적은 프레임들의 총 QoS(모든 프레임들의 QoS들의 합)를 최대화($\max \sum_{i=1}^n f_i(r_i)$)시키는 것이며, 이를 위해 각 프레임의 복원에 할당할 서비스 시간을 결정하고, 또한 그들의 실행 순서를 결정하는 것이다. 그러나 프레임들이 언제 도착하고 그들의 만기가 얼마인지 사전에 알려져 있지 않으므로, 진정한 최적(optimal)의 총 QoS를 생성하도록 스케줄링한다는 것은 매우 어렵다(NP-hard). 따라서 더 이상의 프레임들이 도착하지 않는다는 가정 하에서 현재 시스템에 도착해 있는 프레임들을 상대로 총 QoS가 최대화되도록 스케줄링하는 on-line frame scheduling (OFS) algorithm을 제안한다.

OFS 알고리즘은 상위 계층으로 분리된 두 개의 알고리즘으로 구성된다. 상위 계층의 알고리즘은 프레임들의 만기를 고려하여 최대화된 총 QoS를 생성할 수 있도록 프레임들에게 할당할 서비스 시간을 결정한다. 하위 계층의 알고리즘은 프레임들의 복원 순서(프로세서의 할당 순서)를 결정할 후, 각 프레임에게 상위 계층에서 결정된 서비스 시간만큼만 프로세서를 할당한다. 하위 계층의 알고리즘으로 EDF(earliest deadline first) [5] 알고리즘을 사용한다.

상위 계층의 알고리즘은 더 이상의 프레임들이 도착하지 않는다는 가정 하에 현재 시스템에 도착해 있는 프레임들을 상대로 총 QoS를 최대화할 수 있도록 각 프레임에게 할당할 서비스 시간을 결정해야 한다. 이를 위한 알고리즘으로 참고문헌 [6]과 [7]에서 제안된 IRIS(Increasing Reward with Increasing Service) 태스크 스케줄링 알고리즘을 응용할 수 있다. [6]의 알고리즘은 계산 복잡도 $O(n^2)$ 을 가지며, [7]의 알고리즘은 최악의 경우 $O(n^2)$ 복잡도를 가지지만 평균적으로 $O(n)$ 에 가까운 계산 복잡도를 가지는 보다 향상된 알고리즘이다. 여기서 n 은 시스템 내의 태스크의 개수이다. 본 연구에서는 평균 계산 복잡도가 낮은 [7]의 알고리즘을 응용했다.

OFS 알고리즘은 더 이상의 프레임들이 도착하지 않는다는 전제 하에 프레임들의 만기를 고려하여 서비스 시간을 결정하기 때문에 새로운 프레임이 도착할 때마다 현재 서비스 중인 프레임의 복원을 중단하고, 총 QoS를 최대화하기 위해 프레임들에게 할당할 서비스 시간을 다시 결정 한다. 이 경우 이전 스케줄링 시점에서의 스케줄링 결과(프레임들의 서비스 시간)는 무시된다. 연속하여

빠른 주기로 시스템에 도착하는 프레임들의 특성과 시스템 실행 환경을 면밀히 분석해 보면, 거의 대부분 이전 스케줄링 시점에서 스케줄된 프레임들이 모두 서비스되기 전에 새로운 프레임들이 도착한다. 따라서 매 스케줄링 시 모든 프레임들의 서비스 시간을 결정하지 않고, 만기를 고려한 QoS 증가율이 큰 가장 몇 개의 프레임들에 대해서만 서비스 시간을 결정하는 것이 더 효율적이며 그 만큼 스케줄링 오버헤드를 줄일 수 있다[7].

프레임의 QoS 함수는 점진적으로 증가하는 볼록한 형태를 가지므로 서비스 시간을 많이 받으면 받을수록 동일한 서비스 시간에 대한 QoS 증가율은 점점 감소하게 된다. 이러한 특성을 고려하여 상위 계층의 알고리즘은 매 스케줄링 시점에서 전체 QoS를 최대화하기 위해 프레임들의 만기를 고려하여 프로세서 시간을 각 프레임에게 적절히 분산시켜야 한다. 이를 위해 상위 알고리즘은 bisection method [8]을 적용하여 모든 프레임들의 만기를 고려하면서 가능한 가장 큰 QoS 함수의 도함수 값(QoS 증가율)들 중 가장 작은 값을 결정한다. 이 값은 모든 프레임들의 만기를 증가 하면서 현 스케줄링 시점에서 획득할 수 있는 가장 큰 QoS 증가율이다. 이 값을 각 프레임의 도함수의 역함수의 매개변수로 사용하여 각 프레임에게 할당될 서비스 시간을 구할 수 있다. 이렇게 구해진 각 프레임의 서비스 시간 y_i 에 대해 항상 조건

$$\sum_{i=1}^k r_i = r_0$$

을 만족하는 k ($1 \leq k \leq n$)가 존재한다[7]. 여기서, n 은 현 스케줄링 시점(r_0)에서 시스템에 도착해 있는 프레임들의 개수이고, 이들 프레임들이 만기 순서로 관리된다고 가정할 때, r_k 는 k -th 프레임의 만기이다. 이러한 조건을 만족하는 k 개의 태스크들만 하위 계층의 알고리즘으로 넘긴다.

서비스 시간이 결정된 k 개의 선택된 프레임들은 하위 계층의 EDF 알고리즘에 의해 프레임들의 만기 순서로 프로세서가 할당된다. 이때 EDF는 각 프레임에게 상위 계층에서 결정된 서비스 시간(y_i) 동안 프로세서를 할당한다. 만약 EDF 의해 선택된 프레임들의 서비스가 모두 완료될 때까지 새로운 태스크가 도착하지 않으면, 서비스할 다음 프레임들을 선택하고 이들의 서비스 시간을 결정하기 위해 상위 스케줄링 알고리즘이 다시 수행된다.

4. 실험

실험에서는 동적으로 도착하는 1~4개의 화상회의 스트림(Wavelet 압축방식 사용)을 복원하여 디스플레이하는 단일 프로세서 시스템을 시뮬레이션 한다. OFS 스케줄링을 위해 4개의 화상회의 스트림들을 사전에 분석하여 이들의 QoS 함수를 구하였으며, 이들은 각각 $f_1(x) = 1 - e^{-0.41x}$, $f_2(x) = 1 - e^{-0.33x}$, $f_3(x) = 1 - e^{-0.36x}$, $f_4(x) = 1 - e^{-0.31x}$ 이다. 사전에 분석한 각 스트림의 프레임들을 처리하기 위한 최악의 경우 실행 시간(Pentium III @800Mhz with 128MB of system memory)은 $c1 = 10.3$, $c2 = 11$, $c3 = 13.3$, $c4 = 15.4$ 이었다. 본 연구에서는 기존의 EDF를 수정하여 만기 전까지 중간 복원된 프레임의 이미지 역시 디스플레이 가능하도록 하였으며, 해당 프레임은 만기 전까지 서비스 받은 시간 (x_{ik}) 만큼의 프레임 품질 ($f_i(x_{ik})$)를 가진다.

그림 3은 시스템 내의 스케줄링해야 할 화상회의 스트림 수가 2개(NS2), 4개(NS4)일 때의 OFS와 EDF 스케줄링 알고리즘에 의해 스케줄된 모든 프레임들의 평균 QoS를 보여 준다. 그림에서 Uu 가 감소함에 따라 평균 QoS 역시 감소하고, 동일한 Uu 에서는 화상회의 스트림 수가 많을수록 QoS는 더 감소한다. 그림에서 또한 동일한 스트림 개수에 대해 Uu 가 매우 크거나(NS2인 경우 $Uu \geq 0.4$, NS4인 경우 $Uu = 1.0$) 또는 매우 작을 경우($Uu = 0.1$), 스케줄링 알고리즘에 상관없이 QoS는 비슷한 값을 가진다. 스트림의 개수가 동일할 경우 시스템 부하에 상관없이 항상 OFS의

QoS가 EDF보다 높다. 또한 Uu 가 감소함에 따라 OFS의 QoS는 점진적으로 감소하는 반면 EDF는 OFS에 비해 상대적으로 큰 폭으로 감소한다. 이 실험 결과를 통해 EDF의 경우 프레임들의 화질이 시스템 부하에 따라 민감하게 영향을 받는 대신, OSF의 경우 프레임들의 화질이 소폭으로 변화한다는 것을 확인할 수 있다. 또한 시스템 내의 스케줄링해야 할 스트림의 수가 증가할수록 OSF와 EDF에 의해 스케줄된 프레임들의 화질은 더 큰 차이를 보인다.

그림 4는 화상회의 스트림 수가 4개 일 때(그림 3의 NS4에 대한 실험 결과임) 4개의 스트림별 평균 QoS 중 가장 높은 QoS(Best-NS4)와 가장 낮은 QoS(Worst-NS4)를 보이고 있다. 그림에서 OFS는 시스템 부하가 적을 경우($Uu \geq 0.6$) Best QoS와 Worst QoS 차이가 크지 않다가 시스템 부하가 증가할수록 점점 그 격차가 벌어지다가 궁극적으로는($Uu = 0.1$) EDF와 동일한 격차가 된다. 반면, EDF의 경우 두 스트림간의 QoS 차이가 시스템 부하에 상관없이 매우 크다. 이 실험 결과를 통해 EDF의 경우 시스템 부하에 상관없이 특정 스트림의 화질은 매우 좋는데 비해 어떤 스트림은 매우 나쁜 화질을 가진다. 그러나 OFS는 시스템 부하가 매우 심할 때를 제외하고는 각 스트림별 QoS의 차이가 크지 않는 부수적인 효과를 함께 가진다.

그림 5에서는 스케줄링해야 할 스트림 수가 4개 일 때(그림 3의 NS4에 대한 실험 결과임), 누락되지 않고 디스플레이된 프레임들의 평균 품질과 디스플레이되지 않고 누락된 프레임들의 비율을 보여준다. 그림에서 각 스케줄링 알고리즘별 QDF는 실제 디스플레이된 프레임들만의 평균 QoS이므로, 그림 3의 모든 프레임들의 평균 QoS(NS4)에 비해 상대적으로 높은 값을 가진다. 그림에서 EDF의 경우 시스템 부하가 적을 경우($Uu \geq 0.7$) OFS와 비슷한 QDF 값을 가지지만, 시스템 부하가 증가할수록 OFS보다는 높은 QDF 값을 가진다. EDF의 경우 누락되는 프레임의 비율(RDF)이 OFS보다 훨씬 높게 나타난다. 이는 시스템 부하가 증가할수록 EDF에 의해 디스플레이되는 프레임들의 화질은 OFS보다 조금 더 선명하고 깨끗하지만 반대로 누락된 프레임의 수가 급격히 증가하여 화자의 움직임이 자연스럽지 못하거나 영상의 끊김이 일어나는 현상이 발생하게 됨을 의미한다.

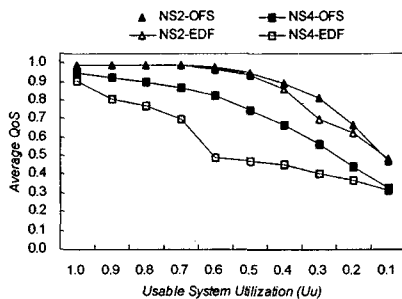


그림 3 Average Quality of Service

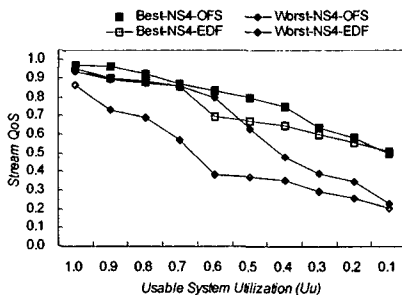


그림 4 Best vs. Worst Stream QoS (4 Stream)

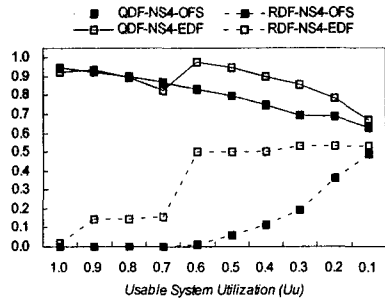


그림 5 Quality of Displayed Frames and Ratio of Dropped Frames (4 Streams)

5. 결론

본 논문에서는 영상회의 시스템에서의 압축된 영상의 복원 과정에서 프로세서의 처리시간에 따르는 결과 영상의 품질에 대한 QoS 함수를 도출하고 이를 IRIS 스케줄링 방안에 적용하는 온라인 스케줄링 알고리즘을 제안하였다. 알고리즘의 효율성을 검증하기 위하여 품질평가를 위한 시뮬레이션을 진행하였다. 실험 결과 제안된 스케줄링 기법을 사용할 경우 시스템 부하가 증가함에 따라 화질은 부드럽게 감소되면서 연속된 프레임들을 지속적으로 디스플레이함으로써 자연스런 화상의 동작을 볼 수 있음을 확인하였다. 또한 한 화면에 다자간 화상회의를 할 경우, 각 스트림별로 화질의 차이가 크게 발생하지 않는 부수적인 이점도 있다.

참고문헌

- [1] N. Laoutaris and I. Stavrakakis, "Intrastream Synchronization for Continuous Media Streams: A Survey of Playout Schedulers," *IEEE trans. on Networks*, Vol.16, No.3, pp.30-40, May 2002.
- [2] D. Stone and K. Jeffay, "An Empirical Study of Delay Jitter Management Policies," *Multimedia Systems*, Vol.2, No.6, pp. 267-279, Jan. 1995.
- [3] G. Jung, K. Yim, J. Shim, K. Choi et. al., "A Jitter Management Policy for Monitoring System using JPEG," *IEEE Proceedings on Consumer Electronics*, pp.214-215, Oct. 1996.
- [4] H. Aydin, R. Melhem, D. Moss, and P. Meja-Alvarez, "Optimal Reward-based Scheduling for Periodic Real-Time Tasks," *IEEE Trans. Computers*, Vol. 50, No. 2, pp. 111-130, Feb. 2001.
- [5] C.L. Liu and J.W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment," *J. ACM*, Vol. 20, No. 1, pp. 46-61, Jan. 1973.
- [6] J.K. Dey, J.F. Kurose, and D. Towsley, "On-line Scheduling Policies for a Class of IRIS (Increasing Reward with Increasing Service) Real-Time Tasks," *IEEE Trans. Computers*, Vol. 45, No. 7, 802-813, July 1996.
- [7] G. Jung, T. Kim, S. Park, and K. Choi, "A Low Complexity Dynamic Scheduling Algorithm for Real-Time Tasks," *IEE Electronic Letters*, Vol. 35, No. 24, pp. 2106-2108, Nov. 1999.
- [8] K. Choi and G. Jung, "Comment on On-line Scheduling Policies for a Class of IRIS Real-Time Tasks," *IEEE Trans. Computers*, Vol. 50, No. 5, pp. 526-528, May 200