3G-324M 환경에서 화상 프레임 정보에 기반한 다중비율 통신 스케줄링 기법

이호철, 윤현준°, 박성용
서강대학교 컴퓨터학과
manime@sogang.ac.kr, hjyun@dcclab.sogang.ac.kr, parksy@sogang.ac.kr

A Multirate Cyclic Loop Scheduling based on The Information of Video Frame in 3G-324M Environment

Hocheol Lee, Hyunjun Yun°, Sungyong Park
Department of Computer Science at Sogang University

요 약

무선 통신 환경에서 실시간 화상 통신을 위해 제안된 3G-324M 프로토콜은 회선 교환 통신을 사용함으로 인해 제한된 대역폭을 이용해야만 하므로 전송 시 여러 가지 제약이 발생하게 된다. 특히 송수신의 스케줄링을 효율적으로 하지 못할 경우 H.223 프로토콜에서 매 타임 슬롯마다 전송 가능한 최대 크기의 데이터를 전송 버퍼에 채워주지 못해 전송 지연 시간이 발생하게 된다.

본 논문에서는 이러한 문제들을 해결하기 위한 통신 스케줄링 기법을 제시한다. 이 통신 스케줄링 기법은 화상 프레임의 종류에 따라 크기가 크게 변한다는 사실에 기초해 H.223 프로토콜의 실행 비율을 임시적으로 증가시켜 지연 시간이 늘어나는 것을 최소화하고 이로 인해 내부 버퍼 사용량을 줄일 수 있도록 한다. 또 수신 버퍼에 처리해야 할 데이터가 많은 경우, 임시로 H.223 프로토콜의 실행 비율을 증가시켜 불필요한 수신 지연 시간이 발생하지 않도록 한다. 실험은 내부 버퍼의 사용량은 제안한 통신 스케줄링 기법이 다른 통신 스케줄링 기법들에 비해 효율적으로 관리되며 패킷의 손실률, 수신 단말기에서의 지연이 줄어드는 것을 보여준다.

1. 서 론

모바일 무선통신의 기술이 발전하면서 통신 속도가 빨라지고 다양한 서비스가 제공되면서 사용자들의 요구가 점점 증대되고 있는데, 그 중 하나가 바로 실시간 화상통신이다. 현재 실시간 화상통신을 위한 많은 멀티미디어 프로토콜이나 관련 기술이 제안되어 있고 이에 대한 구현 및 수정이 이루어지고 있지만, 이러한 기술들의 구현에 대한 여러 가지 고려 사항들은 대부분 이를 구현하는 구현자의 몫으로 남겨져 있고 에러 저항성, 호 설정 시간, 프레임 손실률, 프레임의 크기, 패킷 생성 및 제거 오버헤드, 프레임 압축 시 지연 시간, 전송 지연 시간 등이 있다[2][3].

현재 실시간 화상통신에서 사용할 수 있도록 제안된 프로토콜로서 대표적인 것으로 H.323 프로토콜[4]와 SIP(Session Initiation Protocol)[5]가 있으며, 3세대 휴대폰을 대상으로 개발된 3G-324M 프로토콜[6][7]이 있다. 3G-324M 프로토콜은 3세대 휴대 통신에서 사용할 수 있도록, 3GPP(3rd Generation Partnership Project)[9]에서 기존의 H.324 프로토콜[10]에 모바일 환경에서 사용할 수 있도록 옵션을 추가한 H.324M 프로토콜[11]에서 CODEC 부분을 수정하고 에러 발생 시 이를 처리하는 부분을 추가해 발표한 표준으로, 여러 가지 프로토콜이 함께 동작하도록 구성되어 있다. 그러나 실제 성능에 영향을 미치는 요소들 및 서비스 품질(Quality of Service, QoS)에 대한 정의는 하지 않았다.

3G-324M 프로토콜의 적용 타깃이 되고 있는 휴대 전화 시스템은 저 사양 CPU를 가진 임베디드 시스템에서 여러 가지 서비스를 효율적으로 시간 분배를 해 여러 가지 태스크를 처리할 수 있도록 다양한 통신 스케줄링 방식을 사용한다[13][14][15]. 3G-324M 프로토콜은 64 kbps라는 낮은 대역폭으로 매 타임 슬롯(time slot)마다 일정 크기의 데이터를 전송하기 때문에, 프로토콜의 태스크들과 다른 태스크들을 올바르게 스케줄링하지 못할 경우 대역폭을 제대로 활용하지 못하고 불필요한 헤더 데이터로 인한 대역폭의 낭비로 인해 내부 버퍼 사용량이 증가되고 전송 시간이 느려지게 된다.

통신 시스템에서는 데이터의 전송 및 수신을 위해 다양한 통신 스케줄링 기법을 이용한다. 이러한 통신 스케줄링 기법에는 환형 스케줄링 기법[8], 주기적 스케줄링 기법[12][17][19], 인터럽트를 이용한 주기적 스케줄링 기법[20], 라운드 로빈 스케줄링 기법[21]이 있다. 그러나 태스크 수행 시간을 정확히 예측하기 힘들거나, 태스크가 빠르게 실행이 되어야 하는 상황이 발생할 경우 이를 해결할 수 없는 문제가 발생하거나, 불필요한 태스크일지라도 해당 태스크가 완료되지 않고 계속해서 실행되어 중요한 태스크에 많은 시간을 부여하지 못하는 문제점 등이 존재한다.

본 논문에서 제시하는 통신 스케줄링 기법은 이러한 문제점 및 주의 사항을 반영해 전송 시 지연시간을 줄이되, 내부 버퍼 사용량을 다른 통신 스케줄링 기법들보다 적으면서 패킷 손실률이 적게 발생하도록 하는 것을 목표로 한다. 특히 3G-324M 프로토콜에서 이러한 문제점을 해결하는 데에 있어서 화상 프레임을 참조했다. 화상 프레임은 전송 데이터의 대부분을 차지할 정도로 크기가 크며 일정 시간 내에 전송을 해야 하는 특성을 가진다. 또 화상 프레임은 종류에 따라 크기가 다르기 때문에 프레임의 정보를 읽어 프레임에 따라 다르게 스케줄링을 할 필요가 있다.

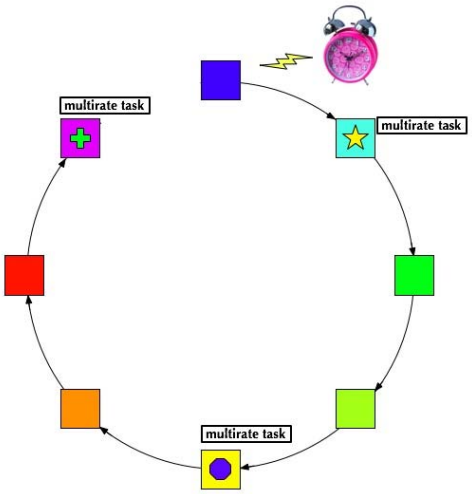
이 후의 본 논문은 다음과 같이 구성된다. 2장에서는 본 논문에서 제안하는 통신 스케줄링 기법인 화상 프레임 정보에 기반한 다중비율(Multirate Cyclic Loop, MCL) 통신 스케줄링 기법에 대하여 설명한다. 3장에서는 실험을 통해 제시한 통신 스케줄링의 성능을 다른 통신 스케줄링 기법들과 비교, 분석한다. 그리고 결론을 내리고 추후 연구과제에 대해 살펴본다.

2. 다중비율 통신 스케줄링 기법

2.1 개요

본 논문에서 제시하는 통신 스케줄링 기법은 CODEC에서 생성

된 프레임은 가능한 한 빠른 시간 내에 전송해야 하며, 내부 버퍼 사용량이 가능한 한 적어야하고, 프레임 전송 시 패킷의 손실률이 최소화시켜야 한다.



[그림 1] 다중비율 통신 스케줄링 기법

다중비율(Multirate Cyclic Loop, MCL) 통신 스케줄링 기법은 [그림 1]과 같이 주기적으로 태스크를 수행할 때, 외부적인 요인이나 내부적인 문제로 인해 빠르게 특정 태스크를 수행해야 하는 경우, 특정 태스크의 실행 비율을 변경시킬 수 있도록 제안한 휴리스틱 스케줄링 기법이다. 앞서 제시한 설계 목표를 달성하기 위해서 이 통신 스케줄링 기법은 다음 사항들을 참조해서 실행 비율이 동적으로 변경된다.

- CODEC에서 생성되는 프레임의 정보
- 타임 슬롯마다 패킷을 네트워크 계층으로 전송하는 전송 버퍼의 버퍼 사용량
- 네트워크 계층에서 수신한 데이터를 저장해 두는 수신 버퍼의 버퍼 사용량

H.263 CODEC에서 생성하는 화상 프레임의 내용은 Intra 프레임과 Inter 프레임으로 나누어진다. Intra 프레임은 I 프레임이라 하고 Inter 프레임은 P 프레임과 B 프레임으로 나뉜다. I 프레임은 정지 화상을 압축해 놓은 데이터로 데이터의 크기가 P나 B 프레임에 비해 매우 크다. P나 B 프레임은 I 프레임과 I 프레임 사이에 위치하게 되고, 두 I 프레임 사이의 움직임 벡터를 저장해 두기 때문에 I 프레임에 비해 상대적으로 작다.

화상 프레임 정보에 기반한 다중비율 통신 스케줄링 기법은 프레임의 내용이 어떤 것이라 할지라도 수신측의 복호기(decoder)는 파라미터로 설정된 초당 프레임 수(fps)에 맞추어 복호를 수행하기 때문에 전송 프레임의 종류에 따라 달라지는 프레임의 크기를 기반으로 태스크들의 실행 비율을 변경시킨다. 따라서 이러한 특성을 반영하지 못한 채 주기적으로 태스크를 수행할 경우, 크기가 큰 프레임 때문에 발생한 지연 시간으로 인해 전송 단말기에서 지연이 발생할 뿐만 아니라 수신 단말기에서도 제 시간에 데이터를 받지 못해 화상의 복호를 수행하지 못하는 일이 발생한다. 화상 프레임 정보에 기반한 다중비율 통신 스케줄링 기법에서는 생성되는 프레임이 I 프레임일 경우, 프레임을 AL 계층으로 전송하고 AL 계층에 있는 프레임들을 MUX-PDU 크기에 맞춰 하나의 스트림으로 만든 뒤 네트워크 계층으로 전송하는, H.223 프로토콜 태스크의 실행 비율을 높게 변경시킨다.

전송하고자 하는 MUX-PDU의 생성속도가 증가된다 할지라도, 전송 대역폭의 제한으로 인해 실제 전송되는 양에는 한계가 있다. 따라서 무조건 태스크의 실행 비율만을 증가시킬 할 경우, 이러한 제한과 더불어 영상과 음성 프레임을 조합해 스트림을 만들 때 스트림을 구성하는 영상과 음성 데이터 구성 비율에 문제가 발생한다. 이 경우, 헤더와 동기화를 위한 플래그 데이터가 불필요하게 많이 생성되어 오히려 내부 버퍼의 크기가 커질 뿐만 아니라, 프레임 전송 지연에도 영향을 미치게 된다. 그러므로 전송 버퍼의 사용량을 체크해 불필요하게 내부 버퍼가 커지지 않으면서도, 전송 지연 시간을 줄이도록 스케줄링 되어야 한다.

전송된 데이터를 수신하면 한 번에 하나의 MUX-PDU가 들어올 수도 있지만, MUX-PDU의 크기가 작은 경우 한 번에 여러 MUX-PDU가 수신된다. 이 경우 스케줄링을 효율적으로 수행하지 못한다면, 수신 버퍼에 MUX-PDU가 있음에도 불구하고 이를 제시간에 가져오지 못해, CODEC용 버퍼에 프레임이 존재하지 않는 일이 발생하게 된다. 이를 방지하기 위해 수신 측에서도 다중비율 통신 스케줄링 기법을 이용하여 MUX-PDU로부터 알맞은 AL 계층 데이터를 추출하고 상위 프로토콜들로 데이터를 전달해 주는 H.223 프로토콜의 태스크 실행 비율을 높게 변경시킨다.

2.2 전송을 위한 스케줄링 기법

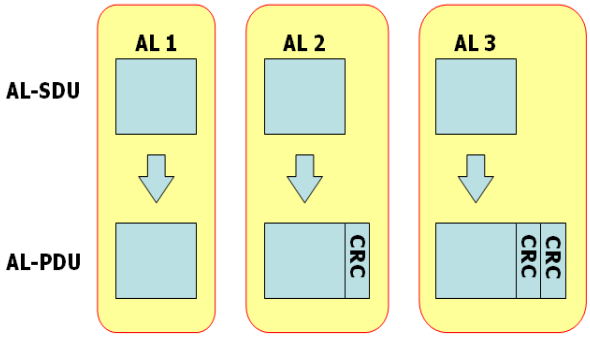
전송을 하기 위해서 우선 고려해야 할 것은 CODEC에서 생성된 프레임의 종류와 타임 슬롯마다 전송 가능한 크기이다. 타임 슬롯 간격을 20 ms로 할 경우, 한 번에 보낼 수 있는 데이터 양(버스트)은 160 bytes가 된다. 이는 3G-324M 프로토콜 표준안에서 권고하고 있는 MUX-PDU 크기와도 일치하는 값이다. 타임 슬롯 당 전송 가능한 크기를 S_U, 초당 전송 가능한 크기 즉 대역폭을 S_S, 그리고 타임 슬롯 간격을 T_P라 한다.

[표 1] H.263 평균 I 프레임 및 P 프레임의 크기

bitrate (kbps)	평균 I frame 크기 (bytes)	평균 P frame 크기(bytes)
13	1098.4	38.1
27.2	1962.9	122.8
44.7	2913.7	258.1
52.2	3670.3	394

[표 1]에서 알 수 있듯이 H.263의 I 프레임과 P 프레임의 크기는 큰 차이를 보인다. 전송률이 커지면서 차이는 더욱 더 커진다. I 프레임의 크기와 P 프레임의 크기를 S_I와 S_P라 한다. 음성 프레임의 경우, 전송률에 따라 크기가 조금씩 달라지지만, 본 논문에서는 12.2 kbps의 전송률을 사용하며 32 bytes의 크기를 가진다. 이를 S_A라 한다.

AL 계층으로 넘어온 프레임은 에러 체크를 위해 MUX 계층으로 전송할 데이터에 대해 CRC 값을 계산해 추가한다. [그림 2]처럼 AL1은 CRC 데이터를 붙이지 않고, AL2는 1 bytes, AL3은 2 bytes의 CRC 데이터를 추가하게 된다. AL2와 AL3의 CRC 데이터 크기를 S_{ACRC}와 S_{VCRC}라 한다.



[그림 2] AL-PDU의 구조

MUX 계층에서는 각 AL들의 AL-PDU들을 조합해 MUX-PDU를 생성한다. 이 때 MUX-PDU 헤더와 전송하는 패킷의 시작과 끝을 알려주는 동기 헤더(synchronization header)가 추가

되는데 각각 3 bytes와 4 bytes를 차지한다. S_{FLAG}를 데이터 이외의 모든 헤더 및 플래그가 차지하는 크기라 하면, 모든 CRC 값과 헤더들의 합으로 나타낼 수 있다.

$$S_{FLAG} = S_{ACRC} + S_{VCRC} + \text{MUX-PDU header} + \text{Sync header} \quad \text{<식 1>}$$

S_M을 MUX-PDU의 최대 크기라 하고, S_{VMAX}를 한 번에 MUX-PDU에 넣을 수 있는 최대 화상 프레임의 크기라 했을 때 <식 1>에서 구한 FLAG의 크기를 이용해 다음과 같이 계산할 수 있다.

$$S_{VMAX} = S_M - (S_A + S_{FLAG}) \quad \text{<식 2>}$$

S_M은 사용자가 설정할 수 있는 값으로 3G-324M 프로토콜의 표준안에서 권고하고 있는 값으로 160으로 설정하였다. <식 2>의 S_{VMAX}를 기준으로 S_I와 S_P를 다음과 같이 표현할 수 있다(단, α, β, δ, ρ 는 트레이드오프 파라미터).

$$S_I = \alpha S_{VMAX} + \beta \quad \text{<식 3>}$$

$$S_P = \delta S_{VMAX} + \rho \quad \text{<식 4>}$$

생성된 MUX-PDU는 각 프레임의 전송 마감시한(deadline)내에 전송을 해야 하는데, 각 프레임의 전송 마감시한을 T_D라 한다. 또 마감시한 내에 전송할 수 있는 타임 슬롯의 수를 N_T라 하면 다음과 같이 계산할 수 있다.

$$T_D = \frac{1\text{sec}}{fps} \text{ (단, } fps\text{는 초당 프레임 비율)} \quad \text{<식 5>}$$

$$N_T = \frac{T_D}{T_P} \quad \text{<식 6>}$$

이제 화상 프레임의 종류별 비율 설정 방법을 알아본다. [표 1]에서 알 수 있듯이, I 프레임의 경우 S_U보다 몇 배나 큰 크기를 가진다. 이 때 $\alpha > N_T$인 경우, 다음 화상 프레임이 CODEC에서 생성되어도, I 프레임을 전송하느라 생성된 프레임을 전송하지 못하는 지연 시간이 발생하게 된다. 따라서 I 프레임을 전송할 때 $\alpha > N_T$인 경우, 태스크 실행 비율을 높여 가능한 한 빨리 다음 프레임을 처리할 수 있도록 한다. 이 때 실행 비율은 다음과 같이 조정한다.

$$if(\alpha > N_T) \quad \text{실행비율} = \lceil \frac{S_U}{S_{VMAX}} \rceil \quad \text{<식 7>}$$
$$else \quad \text{실행비율} = 1$$

<식 7>에서 실행 비율을 $\lceil \frac{S_U}{S_{VMAX}} \rceil$로 설정하는 이유는, 음성 프레임의 경우는 매번 전송을 해야 하기 때문에 화상 프레임의 크기만을 고려해야 하기 때문이다. 대역폭의 제한으로 인해 아무리 빨리 전송을 하려 해도 S_U 이상 전송을 할 수는 없기 때문에 S_{VMAX}로 나눈 값을 이용한다. 반올림(ceiling)은 S_{VMAX}가 S_U의 약수가 아닐 경우, S_U의 크기만큼 MUX-PDU를 생성하지 못해 전송 시 효율이 떨어지는 것을 방지하기 위함이다.

P 프레임의 경우 크기가 대부분 $S_U \times N_T$보다 작기 때문에 태스크 실행 비율을 일반 주기에 맞춰 수행을 하지만, 만약 $\delta > N_T$인 경우가 있다면 그런 경우만 실행 비율을 높이게 된다.

$$if(\delta > N_T) \quad \text{실행비율} = N_T - \delta + 1 \quad \text{<식 8>}$$
$$else \quad \text{실행비율} = 1$$

이렇게 프레임에 기반을 두어 태스크 실행 비율을 높이는 이유는 I 프레임에 의해 전송 지연이 발생한 경우, 주기적으로 태스크를 수행하게 되면 지연이 된 채 남아있는 다른 프레임들은 계속해서 지연된 상태로 전송이 되기 때문에, I 프레임과 같이 크기가 큰 프레임의 지연 시간을 최대한 줄여 다음 프레임의 지연 시간의 발생을 최대한 줄이기 위함이다. 또 다른 이유는 패킷 마커에 의해 S_U만큼 데이터를 전송하지 못하는 경우를 줄이기 위해서이다. 일정 주기에 맞춰 S_U 크기만큼만 전송을 할 경우, 화상 프레임은 S_{VMAX} 크기에 맞게 잘려서 전송되고 마지막에 남은 작은 크기의 데이터는 S_U 보다 작은 크기만큼만 전송을 하게 된다. 즉 실제 전송할 데이터가 남아 있음에도 불구하고, 네트워크 대역폭을 충분히 활용하지 못하는 결과를 낳게 된다. 화상의 전송률이 큰 경우, 이로 인해 현재 전송 가능한 크기의 데이터가 전송되지 못해 다음 타임 슬롯까지 전송이 지연되는 결과가 발생할 수 있다.

하지만 지연 시간을 줄이기 위해 태스크 실행 비율을 높일 경우, 제한된 대역폭으로 인해 전송할 수 있는 크기가 제한되어 있기 때문에 버퍼의 크기가 계속해서 커지는 문제가 발생한다. 따라서 사용자가 설정 가능한 전송 버퍼 사용량의 임계값을 두어 임계값보다 전송 버퍼의 크기가 커질 경우, 실행 비율을 일반 주기와 동일한 값으로 조정하고 실행 비율이 빨라지지 않도록 처리를 해 준다. 지연 시간을 T_{delay}, 지연 시간의 임계값을 T_{thresh}, 전송 버퍼에 차지하고 있는 데이터의 양을 N_B, 그리고 전송 버퍼 데이터양의 임계값을 $N_{Bthresh}$라 한다. 지금까지 기술한 바를 정리하면 다음과 같다.

```
if ( T_delay < T_thresh ) {
    if ( 각 프레임의 전송 시작을 처음 시작하는 경우) {
        if (I 프레임) {
            if (α > N_T) 실행 비율 = ⌈ S_U / S_VMAX ⌉
            else 실행 비율 = 1
        } else {
            if (δ > N_T) 실행 비율 = N_T - δ + 1
            else 실행 비율 = 1
        }
    else if (I 프레임 이외의 프레임 전송 중 && 실행 비율 > 1)
        실행 비율 = 실행 비율 - 1
}
else if (N_B > N_Bthresh) 실행 비율 = 1
else 실행 비율 = ⌈ S_U / S_VMAX ⌉
```

[그림 3] 전송을 위한 스케줄링 기법

2.3 수신을 위한 스케줄링 기법

수신을 위한 스케줄링 역시 다중비율 통신 스케줄링 기법을 이용한다. 전송 시 프레임의 크기 차이로 인해, 작은 프레임의 경우 하나 이상이 전송된다. 일반적으로 I 프레임과 같이 큰 크기의 프레임이 전송된 후, P 프레임이나 B 프레임이 크기가 작아 이와 같이 전송된다. 이런 경우, 주기적인 태스크 스케줄링을 수행할 경우, 크기가 큰 프레임은 다음 프레임을 읽을 때까지는 CODEC용 버퍼로 전송되지 않는다. 따라서 이로 인해

프레임을 모두 수신하였음에도 불구하고 전송을 할 수 없는 일이 발생하게 된다. 이러한 문제를 해결하기 위해 수신 측에서도 수신 버퍼의 사용량을 참조해 실행 비율을 조정하도록 스케줄링 한다.

수신 버퍼에 있는 데이터의 양을 N_{RB}라 하고 CODEC용 버퍼에 있는 데이터의 양을 N_{VB}라 한다. 그리고 수신 버퍼에 있는 데이터양의 임계값을 $N_{RBthresh}$라 하고 CODEC 버퍼에 있는 데이터양의 최소값을 $N_{VBthresh}$라 한다. $N_{RBthresh}$는 적절한 값으로 설정을 하되 N_T번 보다 약간 큰 값을 유지해 무리하게 실행 비율이 높아지지 않도록 설정한다. $N_{VB} > N_{VBthresh}$인 경우는 실행 비율을 일반 주기에 맞춰서 동작하도록 1로 둔다. 하지만 $N_{VB} \leq N_{VBthresh}$인 경우, CODEC용 버퍼가 언더플로우가 될 확률이 높기 때문에, N_{RB}의 값이 $N_{RBthresh}$보다 큰 경우 실행 비율을 증가시킨다. N_{RB}의 값이 $N_{RBthresh}$의 값보다 작은 경우, 실행 비율이 1보다 크다면 그대로 유지시키고, 그렇지 않은 경우 실행 비율의 값을 1로 둔다. $N_{RBthresh}$보다 작은 경우 실행 비율을 증가시킬 경우 시스템의 부하가 커지지만, N_T번 수행 후에는 패킷 마커가 나올 확률이 높기 때문이다. 지금까지 기술한 바를 정리하면 다음과 같다.

```
if(N_VB > N_VBthresh) {
    실행 비율 = 1
}else{
    if( N_RB > N_RBthresh)
        실행 비율 = 실행 비율 + 1
    else
       if(실행 비율 > 1) 실행 비율 = 1
       else 실행 비율 그대로 유지
}
```

[그림 4] 수신을 위한 스케줄링 기법

3. 성능 평가

3.1 성능 평가 환경

실험은 3G-324M 프로토콜 프로토타입을 동작시킨 펜티엄4 3.0 GHz CPU, 1 GBytes 메모리인 총 3대의 시스템에서 이루어졌다. 두 대는 각각 3G-324M 프로토콜을 탑재한 단말기 역할을 수행하며, 다른 한 대는 필요에 따라 에러를 생성할 수 있도록 설계된 프록시 서버이다. 프록시 서버의 에러 생성은 실제 무선 환경 실험과 유사하게 수행하기 위해, Rayleigh Fading Channel 시뮬레이터[22]를 이용, 에러 패턴을 생성 후 이에 따라 에러를 생성하도록 하였고, 버스트에서의 에러 발생 분포는 정규 분포를 따른다[1]. 단 네트워크의 상태는 항상 일정한 것이 아니기 때문에, 네트워크상에서 발생하는 지연 시간은 없는 것으로 가정한다[18]. 또 PC의 경우 패킷 교환 네트워크를 기반으로 데이터 전송이 이루어지기 때문에 회선 교환 에뮬레이션 계층을 추가하였다.

제안한 통신 스케줄링 기법과의 비교를 위해 다음과 같은 세 가지 통신 스케줄링 기법을 함께 구현 및 실험하였다.

- 환형 스케줄링 기법: Basic Cyclic Loop(BCL) - 아주 빠르게 스케줄링을 수행하는 경우에 해당
- 주기적 스케줄링 기법: Periodic Cyclic Loop(PCL) - 타이머 간격을 얼마로 설정하느냐에 따라 지연 시간이나 내부 버퍼 사용량이 달라진다. 타이머 간격을 점점 빠르게 할 경우 지연 시간이 점점 줄어들어 BCL 스케줄링 기법과 비슷해지나 내부 버퍼의 사용량이 계속 줄지는 않는다.
- 라운드 로빈 스케줄링 기법: Round Robin(RR) - 통신 시스템에서 많이 사용되고 있으며, 제안한 통신 스케줄링 기법이 비선점형 스케줄링 기법이므로 선점형 스케줄링 기법인 RR 스케줄링 기법을 포함했다.


[그림 5] claire 및 carphone 데이터

실험에 사용한 화상 데이터는 [그림 5]에서 보는 바와 같이 화상 통화와 유사한 형태로 녹화된 claire와 carphone으로, 영상 실험에 많이 쓰이는 YUV 420 포맷의 비디오 시퀀스이다[16]. claire는 배경이 변하지 않는 상황에서 화자의 움직임만이 있는 데이터이고 carphone은 배경이 함께 움직이는 데이터로 둘 다 화상통화 환경에서 발생하는 영상과 유사한 형태를 나타낸다.

실험은 제안한 통신 스케줄링 기법을 검증하기 위해 다음과 같이 수행하였다. 첫 번째로 스케줄링에 따른 프레임의 지연 시간을 측정하기 위해, 패킷의 손실이 없는 상태에서 송신 단말기의 지연 시간 및 수신 단말기의 지연 시간을 측정하였고, 이때 각각의 통신 스케줄링 기법이 차지하는 버퍼의 양도 함께 측정하여 비교하였다. claire의 경우 carphone과 달리 시간이 지날수록 지연 시간이 증가하는 형태를 가진다. 따라서 패킷의 손실률이 크다. 이를 이용하여, 패킷 손실이 발생하기까지의 임계 시간을 0.5초, 1초, 1.5초 그리고 2초로 두고 임계 시간보다 더 오래 지연되고 있는 패킷의 손실이 발생하도록 하여 각각의 통신 스케줄링 기법 별 손실률을 측정하였다. 그리고 에러가 발생할 때, 각각의 통신 스케줄링 기법 별로 수신 단말기에서의 지연 시간을 측정하여, 송신 단말기에서 지연 시간이 있을 때, 수신 단말기에 지연 시간이 얼마나 더 지연되는지를 확인하였다.

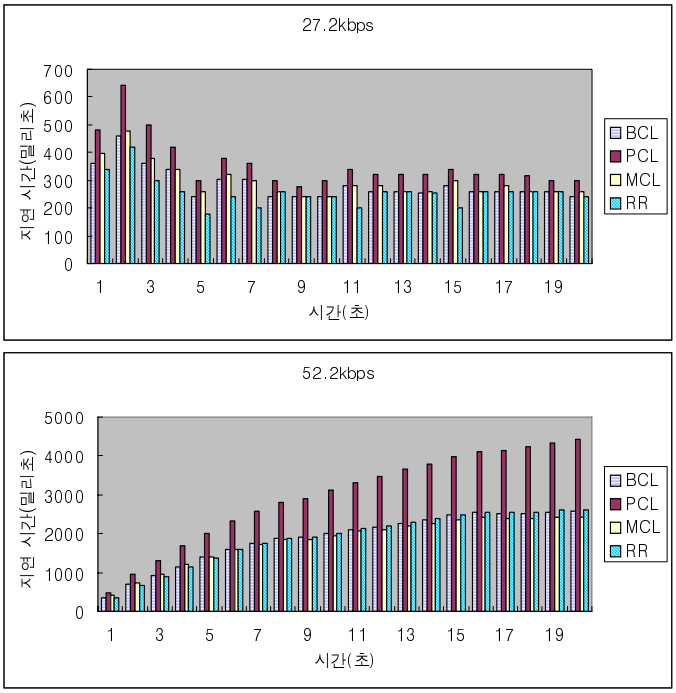
[표 2] 실험 파라미터

프로토콜	파라미터	설정값
H.245	비트레이트	64 kbps
CODEC	화상	H.263
	화상 원본 형식	QCIF
	화상 전송률(claire)	27.2, 44.7, 52.2 kbps
	화상 전송률(carphone)	26.5, 42.1, 49.8 kbps
	화상 GOP	5, 10(기본), 15, 20
	음성	GSM-AMR
	음성 비트레이트	12.2 kbps
H.223	지원 레벨	2
	MTE (Mux Table Entry)	{lcn0,rc ucf}
		{lcn1,rc ucf}
		{lcn2,rc ucf}
		{lcn1, rc 33}, {lcn2, rc ucf}
		{lcn1, rc 65}, {lcn2, rc ucf}
		{lcn1, rc 97}, {lcn2, rc ucf}
	MUX-PDU 크기	160 octets(bytes)
	패킷 손실 임계 시간	0.5, 1, 1.5, 2 sec

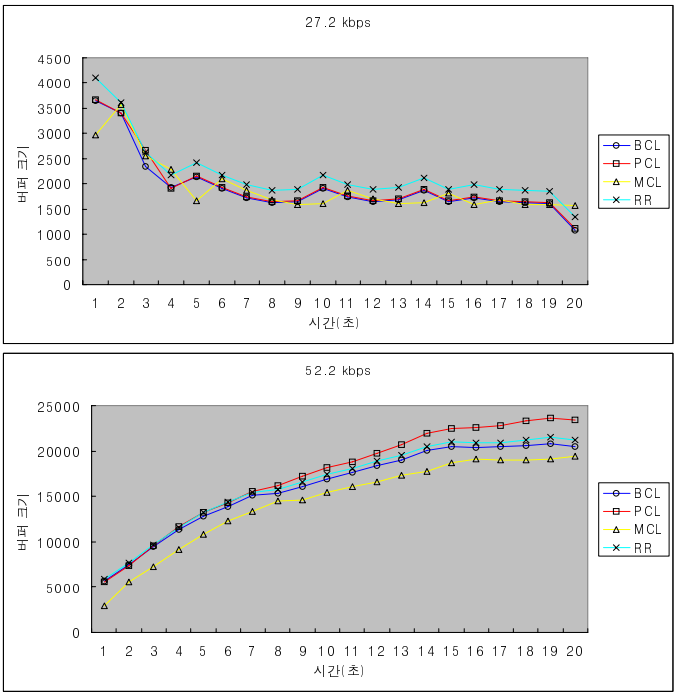
실험 수행 시, [표 2]와 같이 값을 설정하였다.

3.2 성능 측정 및 분석

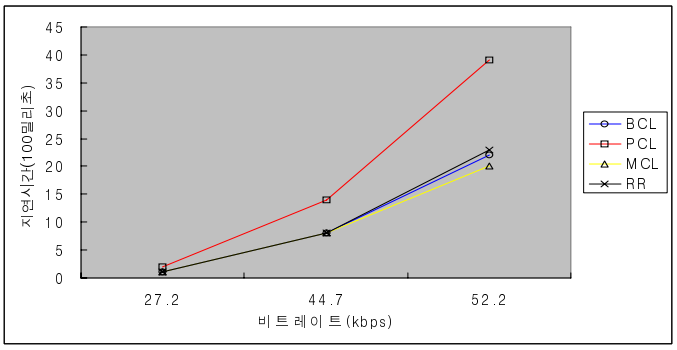
3.2.1 전송 단말기의 프레임 지연시간



[그림 6] 전송률에 따른 전송 단말기의 지연시간



[그림 7] 전송률에 따른 내부 버퍼 사용량



[그림 8] 전송률에 따른 수신 단말기의 지연시간

[그림 6]은 패킷을 손실시키지 않고 전송률이 점점 커질 때, 전송 단말기에서 각각의 통신 스케줄링 기법 별로 발생하는 전송 지연 시간을 측정한 것이다. 전체적으로 지연 시간은 PCL 스케줄링 기법을 제외하면 비슷한 형태를 보이고 있다. MCL 스케줄링 기법의 경우 20 ms마다 전체적으로 타이머가 호출이 되지만 지연 시간이 발생할 경우와 내부 버퍼 사용량이 증가되는 경우, H.223 프로토콜 태스크의 타이머 간격이 실행 비율에 따라 빨라지게 된다. 따라서 타이머가 호출될 때마다 모든 태스크들을 다 수행하지 않고 필요한 태스크의 실행 비율만을 높이게 되므로 다른 통신 스케줄링 기법들에 비해 불필요한 태스크의 호출이 적다. 또 일정량의 데이터가 차 있는 상태에서 실행 비율이 빨라지게 되므로 BCL 스케줄링 기법이나 RR 스케줄링 기법에 비해 MUX-PDU 생성 시 불필요한 플래그와 헤더를 생성하는 양이 적기 때문에 대역폭의 낭비도 적게 발생하게 된다. 이로 인해 타이머의 간격이 PCL 스케줄링 기법에 비해 작으면서도 실제 성능은 더 좋게 나타나게 된다.

3.2.2 전송 단말기의 내부 버퍼 사용량

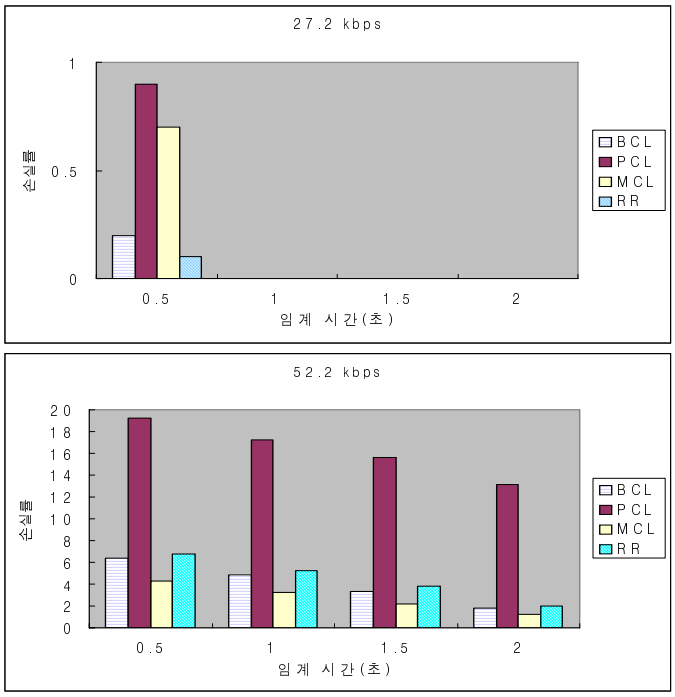
[그림 7]은 각각의 통신 스케줄링 기법 별 내부 버퍼의 사용량을 측정한 것이다. MCL 스케줄링 기법의 버퍼 사용량이 낮은 전송률에서는 다른 기법과 비슷하지만 전송률이 높을 경우에는 우수하다. MCL 스케줄링 기법을 사용할 경우 필요한 경우에만 특정 태스크 비율을 조정하기 때문에 모든 태스크가 매번 불필요하게 수행되는 것을 막을 수 있으며 빠른 실행 비율로 인해 불필요한 플래그 및 헤더가 생성되는 것을 최소화할 수 있어 다른 통신 스케줄링 기법들에 비해 내부 버퍼 사용량이 좋아진다.

3.2.3 수신 단말기의 지연시간

수신 단말기에서는 수신한 프레임을 복호하는데 필요한 데이터의 버퍼에 데이터가 항상 존재해야 하는 것이 중요하다. 이러한 점을 이용해, 수신 단말기의 지연 시간은 버퍼의 데이터가 하나도 포함되어 있는 않는 횟수를 기준으로 시간을 측정할 수 있다. 3.2.1절에서 볼 수 있듯이 지연 시간이 전송률이 증가하면서 커지기 때문에 수신 단말기에서의 지연 시간 역시 [그림 8]처럼 동일하게 증가하는 형태를 보인다. 또한 20 ms 타이머 간격을 가지는 MCL 스케줄링 기법을 사용하였을 때 5 ms 타이머 간격을 가지는 PCL 스케줄링 기법이나 BCL 스케줄링 기법 그리고 RR 스케줄링 기법과 비교해 수신 단말기에서 지연 시간이 커지지 않음을 알 수 있다.

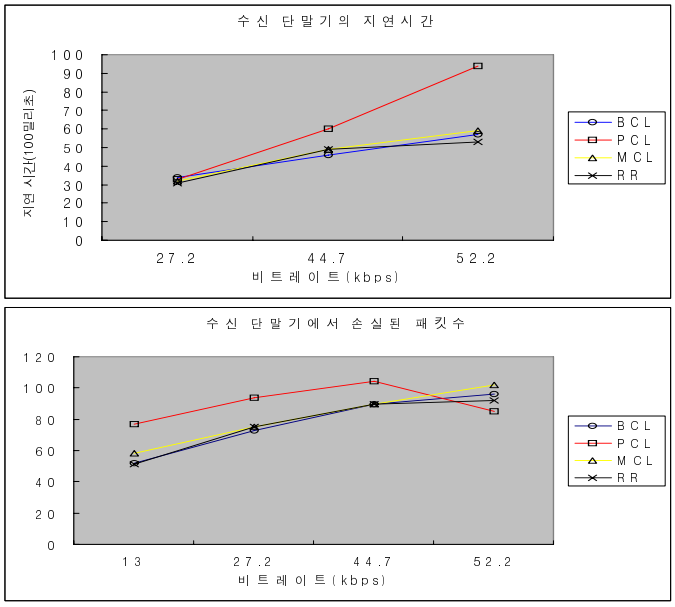
3.2.4 패킷 손실률

3.2.1, 3.2.2, 3.2.3의 실험 결과와 같이, 전송률이 증가하게 되면, 대역폭에 비해 CODEC이 생성하는 프레임의 크기가 커서, 통신 스케줄링 기법과 관계없이 지연 시간과 내부 버퍼 사용량은 증가한다. 따라서 패킷의 손실이 불가피한 경우가 발생한다. 실험에서는 주어진 지연 임계 시간을 기준으로 패킷의 손실 여부를 판단했다. 임계 시간이 가장 적은 0.5초의 경우([그림 9]의 상단), 모든 통신 스케줄링 기법에서 패킷 손실이 발생하였는데, 이는 초기 I 프레임의 크기가 이후에 나오는 I 프레임의 크기보다 커서, 패킷의 손실 비율이 전송 초기가 후기보다 더 많이 발생하기 때문이다. 높은 전송률에서는 초기의 I 프레임 크기가 너무 커서 어떠한 통신 스케줄링 기법을 써도 지연 시간이 발생하는 것을 막기가 힘들다([그림 9]의 하단). 임계 시간을 더 크게 늘린다면 지연 시간을 줄일 수 있겠지만, 전송 단말기와 수신 단말기간의 시간 차이가 많이 나기 때문에 좋은 방식은 아니다. MCL은 전송률이 높을 때 더욱 좋은 성능을 보여준다.



[그림 9] 전송률과 임계 시간에 따른 패킷 손실률

3.2.5 에러 발생 시 지연 시간



[그림 10] 전송률에 따른 수신 단말기의 지연시간과 손실된 패킷수

[그림 10]하단의 수신 시 패킷 손실은 통신 에러로 인해 CRC 에러가 발생해 손실된 패킷의 수를 측정한 것이다. 전송 시 패킷의 손실이 발생하고 통신 에러로 인해 발생한 CRC 에러 때문에 패킷 손실이 발생하기 때문에, 수신 단말기의 지연 시간이 에러가 없을 때보다 더 커지게 된다. 실제 전송한 크기가 크게 차이가 나지 않으므로 각각의 통신 스케줄링 기법마다 큰 차이가 발생하지는 않지만, PCL 스케줄링 기법처럼 지연 시간의 발생이 있어, 다른 통신 스케줄링 기법보다 전송 기간이 길 경우 네트워크에 노출되는 기간이 길어지므로 에러 역시 발생이 증가하는 것을 확인할 수 있다.

4. 결론

통신 기술의 발전이 빨라지고, 이를 이용하는 사용자들의 요구가 증가되면서, 음성 통신을 넘어 화상통신에 대한 관심이 증가되고 있다. 무선 통신 환경에서 실시간 화상 통신을 위해 제안된 3G-324M 프로토콜은 회선 교환 통신을 사용함으로 인해 64 kbps라는 제한된 대역폭을 이용해야만 하므로 전송 시 여러 가지 제약이 발생하게 된다. 특히 대역폭에 비해 크기가 큰 화상과 음성용 프레임을 전송할 때, 효율적으로 스케줄링을 하지 못할 경우 프레임들을 제 시간에 상대 단말기로 전송하지 못해 전송 지연 시간이 발생하게 되고 이로 인해 내부 버퍼 사용량이 증가하고 대역폭의 낭비를 초래하는 문제가 발생하게 된다.

이러한 문제들을 해결하기 위해 제시한 통신 스케줄링 기법은 화상 프레임의 종류에 따라 크기가 크게 변한다는 사실에 기초해, 전송 시 화상 프레임의 정보를 확인 후 크기가 큰 프레임의 경우 H.223 프로토콜의 실행 비율을 임시적으로 증가시켜 지연 시간이 늘어나는 것을 최소화하고 이로 인해 내부 버퍼 사용량을 줄일 수 있도록 하였다. 또 수신 시 버퍼에 많은 양의 프레임 데이터가 있음에도 불구하고 스케줄링이 제대로 되지 않아 제 시간에 복호를 못하는 일이 없도록 수신 버퍼에 처리해야 할 데이터가 많은 경우, 임시로 H.223 프로토콜의 실행 비율을 증가시켜 불필요한 수신 지연 시간이 발생하지 않도록 하였다. 제안한 스케줄링 기법이 지연 시간을 줄이는 데는 전송률에 따라 가장 좋은 결과를 보여주지 못했지만 20 ms마다 동작함에도 불구하고 가장 성능이 좋은 통신 스케줄링 기법에 거의 근접하는 것을 확인할 수 있었고, 특히 내부 버퍼 사용량 및 패킷 손실률 면에서 좋은 결과를 보여주었다.

본 논문에서 수행한 실험을 확장하여 하드웨어 CODEC의 사용, 실제 상황에서의 실험, 패킷 교환 네트워크 방식의 프로토콜과의 비교 등 다양한 환경에서 수행해보고 여러 가지 사항을 종합적으로 고려해볼만 할 것이다.

참고문헌

[1] J.Pons and J.Dunlop, "Bit Error Rate Characterisation and Modelling for GSM", Global Telecommunications Conference, IEEE, vol. 6, Page(s):3722 - 3727, 1998
[2] Hamid Gharavi and Lajos Hanzo, "Scanning the Issue: Special Issue on Video Transmission for Mobile Multimedia Applications", Proceedings of the IEEE, vol. 87, no. 10, 1999
[3] Performance Issues in Mobile Video Telephony, http://www.wireless-center.net/Wireless-Internet-Technologies-and-Applications/Performance-Issues-in-Mobile-Video-Telephony.html
[4] ITU-T: Packet-based multimedia communication systems. ITU-T Recommendation H.323, 1998; www.itu.org
[5] RFC 3261: SIP: Session Initiation Protocol. June 2002; www.ietf.org
[6] Jabri. M.A. "The 3G-324M protocol for conversational video telephony" IEEE MultiMedia, vol. 11, Page(s):102 - 105, 2004
[7] David J. Myers, "Mobile Video Telephony: For 3G Wireless Netoworks" McGraw-Hill, 2004
[8] Joel L. Wolf, Mark S. Squillante, John J. Turek, Philip S. Yu and Jay Sethuraman, "Scheduling Algorithms for the Broadcast Delivery of Digital Products", IEEE Transactions on Knowledge and Data Engineering, vol. 13, Page(s): 721 - 741, 2001
[9] 3rd Generation Partnership Project; www.3gpp.org
[10] ITU-T: Terminal for low bit-rate multimedia communication. ITU-T Recommendation H.324, 1998; www.itu.org
[11] ITU-T: Terminal for low bit-rate multimedia communication Annex H: Mobile multilink operation. ITU-T Recommendation H.324, 1998; www.itu.org
[12] Matt W. Mutka, "Using Rate Monotonic Scheduling Technology for Real-Time Communications in a Wormhole Network", Parallel and Distributed Real-Time Systems, IEEE, vol. 0, Page(s): 194 - 199, 1994
[13] Karen Ellison, "Scheduling Algorithms for systems with hard deadlines", Embedded system programming, 1995
[14] Lonnie Vanzandt, "Scheduling Sporadic Events", Embedded system programming, 2002
[15] R.C. Lacovara, "Scheduling Real-Time Program", Embedded system programming, 1999
[16] Jack, Keith, "Video demystified: a handbook for the digital engineer 3rd ed.", LLH-technology publishing, 1995
[17] W.J. Dally and C. L. Seitz, "The Torus Routing Chip", Journal of Distributed Computing, vol. 1, Page(s): 187-196, 1986
[18] Igor D.D. Curcio and Ari Hourunranta, "QoS of Mobile Videophones in HSCSD Networks", Proceedings of Eight International Conference, IEEE, vol. 0, Page(s): 447 - 451, 1999
[19] Naoko Kosugi, Kazunori Takashio and Mario Tokoro, "Modification and Adjustment of Real-Time Tasks with Rate Monotonic Scheduling Algorithm", Parallel and Distributed Real-Time Systems, IEEE, vol. 0, Page(s): 98 - 103, 1994
[20] David Kalinsky, "Context Switching", Embedded system programming, 2001
[21] Yao Liang, "A Simple and Efficient Scheduling Mechanism Using Minimized Cycle Round Robin", IEEE International Conference, vol. 4, Page(s): 2384 - 2388, 2002
[22] Christos Komninakis, "A Fast and Accurate Rayleigh Fading Simulator"; http://www.ee.ucla.edu/~chkomn