

QoS 지원 멀티미디어 서버 프로토타입 설계

남상준 김태윤
고려대학교 컴퓨터학과
{sjnam, tykim}@netlab.korea.ac.kr

The Prototype Design for the Multimedia Server supporting QoS

Sang-Jun Nam Tai-Yun Kim
Dept. of Computer Science and Engineering, Korea University

요 약

QoS(Quality of Service)는 멀티미디어 서비스 제공시 필수적이다. 그러나 기존의 QoS는 운영체제, 전송 시스템, 네트워크에 대한 연구가 진행되었으며 이는 다양한 네트워크와 시스템에 대한 환경을 지원하기에는 부족하였다. 이에 대해 본 논문에서는 멀티미디어 서버 시스템의 효과적인 방법을 제안한다. 멀티미디어 데이터를 서버의 전송 구조를 변경하고 멀티미디어 전용 큐를 도입하여 기존의 서버 시스템보다 나은 모델을 제안한다.

1. 서론

인터넷 기반 전자 상거래의 발전은 기업의 컴퓨터 시스템 및 네트워크를 설계하고 관리하는 일을 매우 복잡하게 변화시켰다. 더욱이 수 만명의 인트라넷 사용자와 인터넷의 대규모 사용자를 고려해야 하는 서버의 경우 최적의 성능, 신뢰성, 보안성, 확장성, 유연성 뿐만 아니라 다양한 멀티미디어 서비스를 제공해야만 한다.

QoS(Quality of Service)는 원격 교육, 가상 박물관, 원격 진료, 주문형 비디오, 재택 근무, 디지털 방송, 화상 통화 등의 멀티미디어 서비스 제공시 필수적이다. 그러나 현재의 서버 시스템은 다음과 같은 한계점을 가진다.

첫째 멀티미디어 데이터의 효과적인 분배를 제어하지 못한다. 둘째, 텍스트 위주의 전송 방식을 지금까지 고수하고 있다.

그래서 본 논문에서는 멀티미디어 데이터의 전송을 지원하기 위해 기존의 서버 시스템의 전송 과정을 수정하고 전송 큐를 2개를 둬으로써 이를 해결하는 모델을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구로서 Tenet 모델과 일반적인 서버 전송 매커니즘을 알아보고, 3장에서 새로운 멀티미디어 데이터 전송을 지원하기 위한 일반적인 서버의 전송 경로를 줄이고 2개의 큐를 사용함으로써 효율적 전송을 지원하는 모델을 제안한다. 마지막 4장에서 본 논문의 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구

2.1 기존의 QoS 지원을 위한 Tenet 모델

버클리 대학에서 개발한 Tenet 구조[1]는 일반적인 목적의 실시간 네트워크 모델을 제시하고 있는 멀티미디어 스트림을 실시간 통신에 속하는 경우로 정의하고 있다. Tenet 모델은 그림 1과 같은 구조를 가진다.

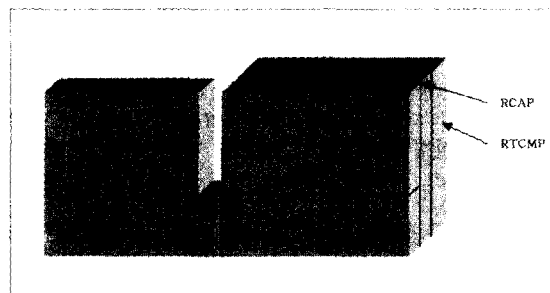


그림 1 Tenet 모델

Tenet 모델은 데이터 전송과 제어 전송의 두 가지로 구성된다. 데이터 전송은 RTIP(실시간 인터넷 프로토콜), RMTP(실시간 메시지 전송 프로토콜) 그리고 CMTP(연속적인 미디어 전송 프로토콜)로 구성된다. 제어 전송 프로토콜은 RCAP(실시간 채널 관리 프로토콜)과 RTCMP(실시간 제어 메시지 프로토콜)이

다. RTIP는 패킷 스케줄러를 담당하고, RCAP는 채널 설정, 접속 허가 테스트 등을 담당한다. RTCP는 에러 처리를 담당하며 인터넷 프로토콜의 ICMP에 대응되는 역할을 한다.

RTIP는 RTIP 데이터그램을 정의하고 패킷 전송에 필요한 버퍼를 확보하고 rate 제어, 지터 제어, 패킷 스케줄링 등을 수행한다. QoS 보장은 결정적 또는 통계적 보장의 두 가지 방법을 제공하는데 결정적 방법은 하드 실시간(hard real-time) 처리를 한다. 통계적 방법은 정해진 지연시간에 대하여 몇 % 이상 지연되지 않는 것을 보장한다.

2.2 일반적인 서버 전송 모델

그림 2는 현재 리눅스 운영체제 시스템에서의 네트워크 입출력 시스템의 계층구조를 보여 준다. 애플리케이션에서 데이터를 요구하고 그 데이터가 네트워크 인터페이스로 전송되는 일반적인 과정이다.

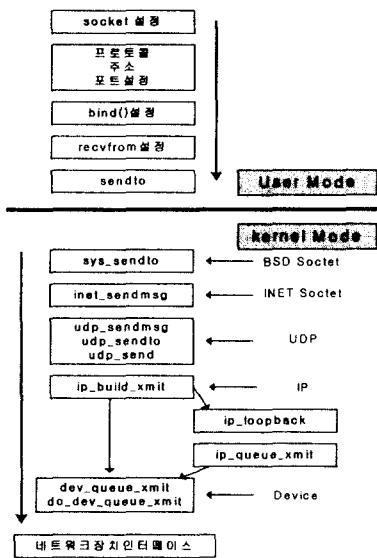


그림 2 리눅스의 전송 메커니즘

그림 2는 Bare 리눅스의 UDP 기반 전송 메커니즘을 보여주는 그림이다.

일반적 전송과 마찬가지로 소켓 설정을 한 후 프로토콜, 주소, 포트 설정을 마치면 바인더를 한 후 recvfrom 함수를 기다린다. recvfrom을 받으면 바로 sendto 함수로 데이터를 전송하게 된다.

일반적인 서버 시스템은 사용자 공간에서 네트워크 장치까지 패킷 전송시마다 데이터 복사가 발생한다. 그림 2의 메커니즘은 작은 크기의 입출력을 행하는 전통적인 텍스트와 바이너리 파일은 전송이 잘 이루어진다. 그러나 멀티미디어 데이터(오디오, 비디오, 애니메이션 등)와 같은 캐싱 속성을 가진 데

이터들은 기대하는 것만큼의 효과를 보지 못한다.

또한 이 서버 시스템은 메모리 공간을 많이 사용한다. 데이터는 종종 이것이 재 사용될 수 있음에도 불구하고 대체될 수 있다. 성능상의 불이익이 인여 복사로 인해 이루어진다. 데이터 복사를 최소화하는 연구는 이전부터 시작되었다. 보호되는 영역을 만들어 물리적 데이터 전송을 최소화하는 fbufs[2], 수신측의 커널 영역에 식별자를 두어 네트워크 장치와 애플리케이션의 버퍼 사이의 경로를 줄이는 Splice[3], 성능 향상을 위해 프로그래밍 언어 수준의 제어와 데이터 구조를 명확히 하는 SPIN[4], mmbuf(multimedia mbuf)를 사용하여 새로운 데이터 구조를 커널 버퍼로 생성해 멀티미디어 데이터 전송을 지원하는 MARS[5] 메커니즘 등이 있다.

그러나 이러한 메커니즘들은 공통적으로 다음과 같은 단점을 가진다.

- 커널내 새로운 구조체 생성으로 과도한 메모리 할당이 이루어진다.
- 디스크 장치와 네트워크 장치 사이에 새로운 구조체의 생성으로 인한 다른 장치와의 호환성 결여 문제가 발생한다.
- 원래의 운영체제가 지원하는 커널 버퍼 이상의 할당인 경우 시스템에 치명적인 다운 현상이 발생한다.

본 논문에서는 이러한 현상들을 줄이고 기존의 운영체제에 맞도록 데이터 전송 구조를 제안하고 이러한 전송 구조를 통해 내려온 패킷을 2개의 큐를 두어 멀티미디어 패킷과 일반적인 데이터 패킷으로 구분하여 효율적인 네트워크 전송 구조를 지원한다.

3. 제안된 전송 구조 메커니즘

기존의 서버 전송 구조는 애플리케이션과 커널 영역 사이의 데이터 복사와 문맥 교환의 오버헤드가 발생한다. 이러한 오버헤드를 줄이기 위해 일반적인 서버 시스템에 존재하는 데이터 전송 단계를 줄임으로써 전송상의 향상을 보고자 한다. 다음에 전송할 데이터가 무엇인지 아는 멀티미디어 데이터인 경우 일반적인 서버 시스템에서는 반복적으로 사용자 모드와 커널 모드를 넘나들면서 데이터를 이동시키며 불필요한 데이터 복사와 문맥 교환으로 인한 오버헤드가 발생하게 된다. 리눅스 기반의 일반적인 UDP 전송 메커니즘의 커널 경계 부근에서 데이터 전송 경로는 위의 그림 2와 같다. 그림 2는 sendto 함수를 호출하면서 커널 모드와 사용자 모드의 경계에서 데이터 복사가 발생하는 모습을 볼 수 있다. 그러나 그림 3과 같은 메커니즘을 사용한 전송 메커니즘에서는 send 함수를 커널 내부에 추가함으로써 데이터 복사를 줄이는 과정을 볼 수 있다.

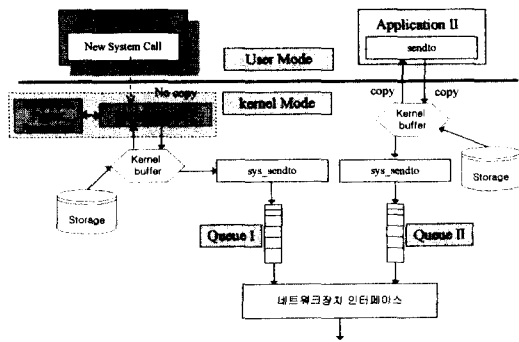


그림 3 제안된 멀티미디어 지원 모델

그림 3과 같은 모델을 가진 운영체제는 현재 존재하고 있지 않다. 제안된 모델의 전송 메커니즘은 다음과 같다.

멀티미디어 데이터 전송 서버에서 동작하는 애플리케이션은 그림 3과 같이 애플리케이션 I과 애플리케이션 II라고 가정하였다. 애플리케이션 I은 멀티미디어 전송을 위한 일반적 데이터 전송 경로를 줄여 데이터 복사를 줄였고 이 패킷은 큐 I으로 들어가게 하였다. 멀티미디어 데이터 전송 애플리케이션 I에서의 전송 과정은 다음과 같다.

제안된 모델의 애플리케이션 I은 클라이언트에서 멀티미디어 서비스 요청을 받은 서버 시스템은 서비스 할 멀티미디어 데이터의 위치와 정보를 새로운 시스템 콜을 이용해 커널로 내려 보내준다. 이 시스템 콜은 사용자 영역에서 받은 멀티미디어 데이터 정보를 패킷 프로세서에게 보내 적당한 헤더를 붙고 처리한다. 이 패킷을 같은 커널 영역에서 UDP 계층으로 넘겨주고 UDP 계층은 자신의 헤더를 붙인 후 큐 I을 통해 네트워크 인터페이스로 옮겨주게 되는 것이다. 이러한 과정으로 다음 패킷부터 마지막 패킷까지 커널 영역에서 RTP 패킷 생성과 UDP 전송을 하는 일련의 작업을 처리하게 된다.

애플리케이션 II는 멀티미디어 데이터가 아닌 나머지 일반적인 데이터 전송을 지원하는 경로이다. 애플리케이션 II는 기존의 전송 경로를 통하게 하여 큐 II로 패킷을 들어가게 하였다.

그림 3과 같이 제안된 새로운 멀티미디어 지원 모델의 이점은 다음과 같다.

- 입출력의 개선
애플리케이션에서 새로운 시스템 콜 한번의 호출로 인해 데이터 전송을 커널 내부로 옮겨 이에 대한 불필요한 데이터 복사와 문맥 교환을 줄였다.
- 멀티미디어 패킷 처리
멀티미디어 패킷을 생성함에 있어, 기존의 방식은 사용자 애플리케이션에서 이루어지는 것을 커널 내부로 내장함으로써 처리 속도의 향상을 이점을 얻고자 한다.
- 커널 메커니즘의 수용
최소한의 커널 메커니즘의 수정으로 효율적인 멀

티미디어 데이터의 서비스를 지원한다.

- 차별화된 큐 관리
큐를 2개로 두어서 멀티미디어 데이터 처리 전용 큐를 사용함으로써 향후 패킷 스케줄러의 효율적 알고리즘의 도입으로 성능 향상을 기대한다.

4. 결론 및 향후 과제

본 논문에서는 멀티미디어 데이터의 네트워크 전송 지원을 위한 방안으로 새로운 모델을 제안하였다. 새로운 제안된 모델을 이용하여 실제 멀티미디어 데이터 생성과 전송은 커널 내부의 새로운 시스템 콜이 담당하게 되고, 애플리케이션은 휴지 상태로 들어갈 수 있도록 설계하였다. 따라서 커널 공간에서 사용자 공간으로의 데이터 복사가 필요하지 않고, 이를 수행하기 위한 멀티미디어 데이터 전송의 사용자 프로세스를 스케줄링 할 필요가 없는 성능상의 향상이 있었다. 또한 멀티미디어 큐를 만들어 기존의 큐와 차별화함으로써 멀티미디어 전용 서버로 구현하기 쉬운 모델을 제안하였다. 향후 연구 과제는 이러한 모델을 본 논문에서 제안한 모델을 구현하고 멀티미디어 데이터 전송에 적합한 큐 알고리즘을 연구하여 멀티미디어 서버 시스템의 핵심 기술로 활용할 수 있을 것이다.

참고 문헌

- [1] A. Banerjee, D. Ferrari, B. Mah, M. Moran, D. Verma and H. Zhang, "The Tenet Real-time Protocol Suite: Design, Implementation, and Experiences", IEEE/ACM Transactions on Networking, Vol. 4, No 1, pp.1-10, Feb. 1996
- [2] P. Druschel and L. L. Peterson, "Fbufs: A highbandwidth cross-domain transfer facility", In Proceedings of the Fourteenth ACM Symposium on Operating System Principles, pp. 189-202, Dec. 1993.
- [3] Kevin Fall and Joseph Pasquale, "Improving Continuous-Media Playback Performance with In-kernel Data Paths", Proceedings of the International Conference on Multimedia Computing and Systems, May 14-19, Boston, Massachusetts. IEEE-CS, pp. 100-109, 1994.
- [4] B. N. Bershad, S. Savage, P. Pardyak, E. G. Sirer, M. E. Fiuczynski, D. Becker, C. Chambers and S. Eggers, "Extensibility safety and performance in the SPIN operating system", Proceedings of the fifteenth ACM symposium on Operating systems principles, pp. 267-283, December 1995.
- [5] Buddhikot, M. and Chen, X. "Project MARS: WWW based Scalable, Interactive Multimedia Recording and Playback Services", Winner of the first prize at the Research Poster Competition, ACM SIGCSE/SAC'97, San Jose, CA, Feb 27 - Mar 1, 1997.