

인터넷 스트리밍 서비스의 패킷 손실 억제를 위한 전송률 적응제어 시뮬레이션

윤인철, 권성기, 최우정, 장봉석
국립목포대학교, 정보공학부, 멀티미디어공학전공

Adaptive Rate Control Simulation for Packet Losses in Internet Streaming Services

Inchul Yoon, Sunggi Kwon, Woojeong Choi, Bongseog Jang
Major in Multimedia Eng., Div of Information Eng., Mokpo National University
E-mail : yunic2k@mokpo.ac.kr

요 약

본 논문에서는 인터넷 상에서 UDP 소켓 인터페이스를 이용하여 실시간 스트리밍 전송을 시뮬레이션 하였고 또한 수신자로부터 피드백 된 정보를 이용하여 송신자가 전송률을 제어하는 것을 시뮬레이션 하였다. 수신자로부터 제어정보를 피드백 받아 송신자가 전송률을 제어하는 경우와 송신자가 전혀 전송률제어를 하지 않는 경우의 패킷 손실 값을 비교했다. 결과로서, 피드백 되는 정보를 이용하여 제어를 하는 경우가 지속적인 패킷손실을 방지함을 보여 준다.

1. 서론

인터넷을 이용한 다양한 실시간 멀티미디어 서비스가 요구되어 지고 있다. 기존의 데이터 서비스 외로 비디오 또는 음성 서비스의 이용을 통한 멀티미디어 데이터의 실시간 전송 요구가 폭발적으로 증가하고 있다. 특히 화상 회의와 같은 실시간 비디오 서비스들은 일정 양의 대역폭, 지연시간, 품질을 보장할 수 있는 적절한 패킷 손실률과 같은 네트워크 자원의 보장을 필요로 한다.

실시간 데이터는 연속성을 지니기 때문에 스트리밍 전송 방식을 이용한다. 미디어 서버 같은 서비스 제공 주체는 지속적이며 일정한 간격으로 데이터를 사용자 수신 시스템에 전달하고 수신 시스템은 인터넷의 특성상 발생할 수 있는 지터, 지연, 패킷손실을 최소화한 상태에서의 플레이가 필요하다. 수신자 시스템에서는 버퍼를 활용하여 지터를 보상해 주는 경우도 필요하며 패킷 손실에 의해 발생한 에러제어에 대한 방안도 필요하다. 송신 시스템은 망의 상태를 고려하여 적

응적인 전송률 제어에 대한 기술이 필요하다. 또한 인터넷을 통해 다양한 특성을 갖는 멀티미디어 데이터를 동시에 서비스 해 주어야 하기 때문에 지연에 민감하지 않은 웹 트래픽 같은 경우와의 대역분배에 있어서 공평성도 고려 해 보아야 한다.[1]

실시간 적이고 지속적인 전송 특성을 고려할 때 기존의 인터넷 전송 프로토콜인 TCP나 UDP를 이용해서 전송할 경우 여러 가지 문제가 생긴다. TCP의 경우 손실 패킷에 대해 재전송을 하므로 신뢰성은 증가하지만 수신측에서는 연속적인 재생이 보장되지 않는다. UDP의 경우 손실 패킷에 대해 어떤 처리도 해주지 않으므로 연속적인 재생은 가능하지만 실시간 스트리밍의 품질이 저하된다.[2]

RTP(realtime transmission protocol)는 IETF RFC 1889에 명시된 표준 프로토콜이다.[3] RTP는 일반적으로 UDP의 상위에서 동작한다. RTP 헤더는 타임스탬프, 일련번호, 전송할 데이터의 종류 같은 정보를 포함한다. RTCP는 RTP의 제어 프로토콜이다. RTP를 이용한 서비스에 참여한 모든 참여자에게 멀티캐

스팅을 이용하여 응용 프로그램에서 이용할 정보를 주기적으로 제공한다. 전송한 패킷의 수, 손실 패킷의 수, 패킷간의 지터 등과 같은 통계 정보를 제공한다. 이러한 정보를 이용하여, 응용 프로그램에서 실시간 스트리밍 데이터의 전송률제어로서 활용할 수 있다. 즉, 패킷 손실의 증가를 수신 시스템으로부터 피드백 받게되면 송신을 위한 전송률을 줄일 수 있다. RTP는 엔드-to-엔드로 제공되는 프로토콜이므로 인터넷에서 발생하는 혼잡상황에 대처하기 위해서 송신 시스템의 응용 프로그램에게 수신 시스템으로부터의 피드백 정보는 유용하게 활용될 수 있으며 UDP에서 제공하지 못하는 혼잡 및 흐름제어를 응용 프로그램이 제공할 수 있다.

이러한 RTP/RTCP를 하위 계층으로 하고 RTCP의 피드백 정보를 이용하여 적응적인 실시간 스트리밍 전송률 제어를 위한 여러 가지 형태의 프로토콜 및 알고리즘이 최근에 제안되고 있다. 대표적인 것 중에 하나는 AIMD (additive increase / multiplicative decrease) 알고리즘이다.[4] 패킷 손실률이나 전송지연을 변수로 하고 그 변수의 미리 지정한 임계값을 이용하여 임계값 보다 값이 크게 되면 배수형태로 전송률을 감소시키고, 그 반대 상황에서는 가산형태로 전송률을 증가시킨다. 이러한 알고리즘을 이용한 프로토콜로서 RAP(rate adaption protocol)은 RTP 계층에 존재할 수 있으며 TCP 에서와 같은 ACK 메시지를 이용한다.[5] 시간만기나 lost된 packet은 패킷 번호를 통해서 알 수 있다. 만일 패킷의 loss가 탐지되지 않으면 송신측은 주기적으로 측정된 왕복지연시간의 함수로서 전송률을 가산 적으로 증가시킨다. 반면에 lost가 탐지되면 전송률을 배수형태인 반으로 줄이게 된다. 그림 1은 전형적인 스트리밍 서비스를 위한 인터넷 프로토콜 계층 모델을 나타낸다.

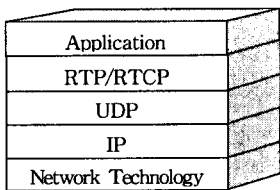


그림1. 인터넷 스트리밍 위한 대표적인 프로토콜

모든 피드백을 이용한 적응 적인 제어방식들은 TCP 우호적인 제어를 하도록 하여야 한다. 기존 웹 트래픽 같은 TCP를 이용하여 혼잡제어를 하는 경우에는, 혼잡상황이 발생하고 패킷 손실이 발생되면 제

전송을 하여야 하고 송신측의 전송률을 급격하게 줄여야 하기 때문이다. 그러므로, UDP를 사용한 적응적인 제어방식들은 TCP를 이용한 서비스에 대한 불공정한 대역점유가 가능하고 결국은 인터넷의 불안정성을 야기 시킬 수 있기 때문이다.[1]

본 논문은 초기연구 단계로서의 시뮬레이션 스타디이다. 본 연구의 궁극적인 목표는 TCP 우호적이면서 보다 효율적으로 스트리밍 서비스를 피드백 정보를 이용한 전송률 적응제어를 할 수 있는 방법에 대한 연구이다. 본 논문의 시뮬레이션 환경은 인터넷상에서 두 호스트를 설정하고 UDP 소켓 인터페이스를 구성한다. 패킷 손실, 지연, 수신된 패킷 수 등과 같은 제어정보를 피드백 할 수 있는 소켓 인터페이스를 이용해 수신 시스템이 송신 시스템에게 전달하고 송신 시스템은 피드백 된 정보를 이용하여 전송률을 재 설정한다. 패킷의 손실 정도에 따라서 AIMD 같은 형태의 전송률 결정을 하도록 하였다. 먼저, 2장에서는 인터넷 상에 구성되는 시뮬레이션 모델에 대해서 살펴본다. 3장에서는 시뮬레이션 결과에 대한 내용이며 4장에서는 결론과 향후 연구 계획에 대한 내용이다.

2. 인터넷 시뮬레이션 모델

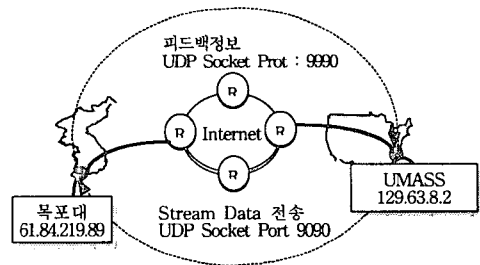


그림2. 인터넷 상의 시뮬레이션 구성도

그림 2와 같이 인터넷 상의 두 개의 호스트를 이용하여 시뮬레이션 하였다. 하나는 목포대학교 정보공학부 연구실에 있는 Solaris 버전2.6 이며 스트리밍 서비스 요청 시스템으로 구성하였다. 그러므로 스트리밍 데이터를 서버에 요청한 후 지속적으로 데이터를 받으며 서비스 중 도착 패킷에 대해 매초 당 지연, 지터, 도착 패킷 수, 손실 패킷 수를 계산하여 통계적인 값이나 축척된 값을 서버에게 피드백 시켜 준다. 다른 하나의 호스트는 미국 메사츄세츠 주립대 전산학과 CompaqAlpha 서버이며 Tru64Unix 버전이다. 이 호스트는 스트리밍 서버로 가상하고 서비스 요청 후 지

속적이며 일정한 패킷간격으로 데이터를 전송한다. 또한 피드백 된 정보를 활용하여 새로운 전송률을 결정하고 적응적으로 전송률을 변경한다. 전송률을 결정하고 변경하는 방법은 AIMD 형태의 간단한 알고리즘을 적용하였다.

두 개의 호스트는 모두 두 개의 UDP 소켓 인터페이스를 이용한다. 하나의 UDP 연결은 스트리밍 데이터의 송수신용이고 다른 하나는 피드백 정보를 송수신하기 위한 것이다. 서버는 스트리밍 데이터를 UDP로 수신 시스템에 전달하기 전에 각 전송될 데이터에 헤더정보를 추가하였다. 헤더에는 보낼 데이터의 일련번호만을 부가하였다. 그러므로 수신 시스템은 헤더의 일련번호를 검사하여서 패킷의 지연 및 손실에 대해서 측정할 수 있도록 하였다. 제어를 위한 피드백 정보의 데이터는 수신 시스템에서 매초마다 생성되며 피드백 정보의 내용은 일초동안 패킷의 손실된 개수, 일초 동안에 지연이 50ms 이상 된 패킷의 개수, 초당 도착된 패킷의 개수이다. 수신 시스템은 모든 도착하는 패킷들에 대해서 스트리밍 데이터의 일련번호를 참조하여 해당되는 정보를 계산하고 매초 결과를 파일에 저장해 둔다. 피드백 전송을 위한 UDP 소켓 프로그래밍은 매초 당 그 저장된 내용을 읽고 전송하도록 하였다.

서버는 수신된 피드백 정보를 읽고 전송률을 다시 계산한다. 전송률을 다시 계산하는 전송률 적응제어 알고리즘은 다음과 같다. 실제적으로는 시뮬레이션에서 전송되는 패킷간의 사이 간격을 서버가 계산하도록 하였다 (그림 3의 x 와 y 값).

1. 패킷손실 발생 한 경우는 이전 전송률 보다 0.8배 감소하도록 한다.[4] 그러므로 패킷 손실이 연속적으로 발생하면 지속적으로 전송률을 줄여준다.
2. 패킷의 손실이 감지되지 않으면 전송률을 가산적으로 증가시켜 준다. 가산 값은 5ms 혹은 10ms 증가하도록 하였다.

전체적인 시뮬레이션 절차는 그림 3에 요약하였다.

3. 시뮬레이션 결과

시뮬레이션 결과는 피드백을 통한 전송률의 적응제어 경우와 전혀 제어를 하지 않고 일정하게 지속적으로 패킷을 전송하는 경우와의 비교를 하였다. 시뮬레이션 초기의 전송률은 최대 약 9 kbps 로 설정하였다. 서버는 초당 평균 약 130 패킷을 전송하였고 스트리밍 데이터 사이즈는 패킷 당 800 byte 정도 설정하였다. 제어를 하지 않는 경우는 위의 전송속도가 시뮬레이션 동안 변화를 주지 않았고, 제어를 하는 경우는 2장에서 설명한대로 전송률을 변화 시켜 주었다. 시뮬레이션 전체 시간은 2만개의 패킷을 전송하도록 하였다. 한국시간으로 밤 11시 이후에 시뮬레이션을 하였고 미국시간으로는 아침 10시 정도이다. 인터넷의 트래픽이 어느 정도 인지는 정확히 알 수 없지만, ping test를 해본 결과 평균 280ms의 왕복 지연 시간을 보여 주었다.

그림 4는 제어를 하지 않는 경우 초당 도착된 패킷의 수와 손실된 패킷의 수를 보여 준다. 손실되는 패킷의 범위가 시간에 따라서 지속적으로 발생함을 보여 준다. 손실된 패킷의 전체 개수는 2만개 중 5,475개의 패킷의 손실이 있었다. 그러므로 시뮬레이션 기간 동안 약 30%의 손실률을 보여 준다. 그림 5는 제어를 한 경우에 초당 도착된 패킷의 수와 손실된 패킷의 수를 보여준다. 그림 4와 비교해서 지속적인 패킷의 손실이 발생하지 않음을 알 수 있다. 또한 전송되는 패킷의 수에 있어서 보다 가변성을 보여 주고 있다. 이 경우는 전체 2만개의 전송된 패킷 중 3,335개 패킷의 손실이 있었다. 손실률은 약 20% 정도 된다. 그림 6은 위의 두 경우에 대해서 패킷 손실 량

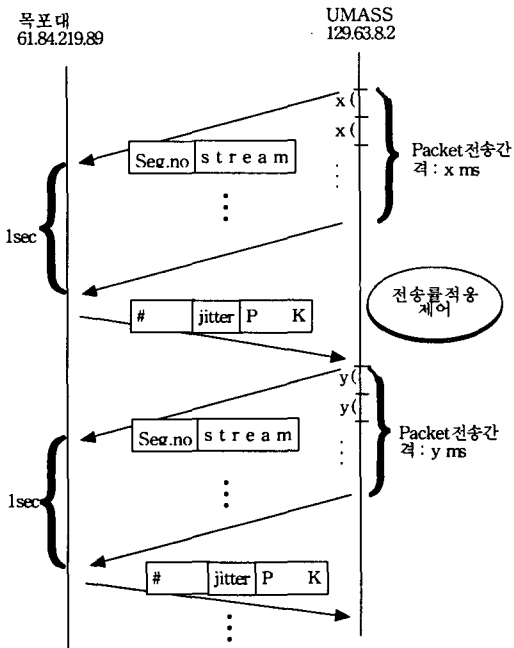


그림 3. 시뮬레이션 절차도

에 대한 비교 그림이다.

그림 7은 수신 시스템에서 지터를 측정 한 결과이다. 수신측의 저장 버퍼를 50ms로 가정하고 패킷 도착간격이 50ms 이상이 되는 패킷의 개수를 나타낸다. 제어를 한 경우가 하지 않는 경우에 비해서 패킷 간격 50ms 이상이 되어서 도착하는 경우가 더 발생함을 알 수 있다. 이러한 현상은 송신 시스템이 전송률을 제어하는 과정에서 패킷손실이 지속적으로 길게 유지되는 경우에 송신 시스템이 전송률의 감소를 계속적으로 시도하는 것과 연관이 있다. 이러한 지연을 수신 시스템에서 소화할 수 있도록 수신 시스템의 버퍼 크기를 조절하는 것이 필요하다는 것을 알 수 있다.

전체 2만개의 패킷을 전부 수신하는데 까지 걸리는 시간을 수신측에서 측정한 값을 보면, 제어를 하게 됨으로서 지연이 길어지므로 전체 패킷이 도착하는 시간이 길어진다. 그러므로 수신 시스템의 충분한 크기의 버퍼용량 및 초기 playing out 시간의 최적한 설정 값 등이 필요하다고 보인다.[6]

4. 결론 및 향후 연구계획

실시간 스트리밍 데이터를 전송하기 위해서 수신 시스템의 피드백 정보를 이용하여 송신 시스템에서 전송률을 적응 적으로 제어하는 시뮬레이션 결과를 보았다. 시뮬레이션은 엔드-to-엔드 응용 S/W가 UDP 상위에서 동작하는 형태이며 두 개의 소켓 인터페이스를 이용하여 데이터 전송과 주기적인 피드백 정보 송수신을 구현하였다. 패킷의 손실에 대해서 송신 시스템이 적응 적인 전송률 제어를 하도록 하였고 그 결과를 제어하지 않는 경우와 비교하였다.

제어를 함으로서 패킷의 손실률이 감소함을 볼 수 있었고 장시간 발생하는 지속적인 패킷 손실에 대한 방지가 가능함을 볼 수 있었다. 그러나 제어를 함으로서 패킷의 지연이 더 발생하게 되고 제어하지 않는 경우에 비해 주어진 시간 내에 수신 시스템에 도착하지 못하는 경우가 조금 더 발생한다. 그러므로 수신측의 버퍼가 지터를 흡수할 수 있도록 충분한 사이즈를 가져야 할 것이다.

시뮬레이션을 통해서 확인 한 것처럼 패킷의 손실은 지속적이며 다소 장시간 버스트하게 발생한다는 것을 알 수 있다. 비록 일대일 연결만을 가정했고 많은 횟수의 시뮬레이션 데이터의 분석 작업이 아직 안된 상태이지만 혼잡상황이 발생할 수 있는 상황을 가정해서 시뮬레이션 했기 때문에 그와 같은 패킷 손실 패턴은 인정할 수 있다.

본 연구의 궁극적 목표는 TCP 우호적이면서 피드백 정보를 활용한 가장 효율적인 적응적 전송률제어 알고리즘 및 프로토콜을 개발하는 것이다. 추후 연구로는 본 시뮬레이션에서 사용한 간단한 전송률 제어 알고리즘을 더욱 보강할 예정이다. 왕복전송 지연을 활용하여 전송지연을 전송률을 변화시키는데 적용하고자 한다. 그리고, 패킷손실의 지속적이며 버스트한 패턴을 이용하여 예측제어합수를 개발하여 보다 효과적으로 전송률을 제어하는 방법을 연구할 예정이다.[7] 패킷손실의 시작을 미리 결정할 수 있는 요인을 알 수 있으면 보다 적절한 전송률을 계산할 수 있고 또한 지연시간도 보다 최소화 할 수 있을 것이다.

References

- [1] J. Widmer, R. Denda, and M. Mauve, "A Survey on TCP-Friendly Congestion Control", IEEE Network, May/June 2001.
- [2] J.F. Kurose and K.W. Ross, "Computer Networking A Top-Down Featuring the Internet", Addison Wesley, 2001.
- [3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, Jan. 1996.
- [4] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS Control of Multimedia Applications based on RTP", Computer Communications, vol. 19, no. 1, Jan. 1996.
- [5] R. Rejaie, M. Handley, and D. Estrin, "An end-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet", IEEE INFOCOM'99, Mar. 1999.
- [6] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks", IEEE INFOCOM'94, 1994.
- [7] 오창운, 장봉석, "ABR 서비스를 위한 트래픽 예측 제어 알고리즘 연구", 한국인터넷정보학회 논문지, vol. 1, no. 2, 2000년 12월.

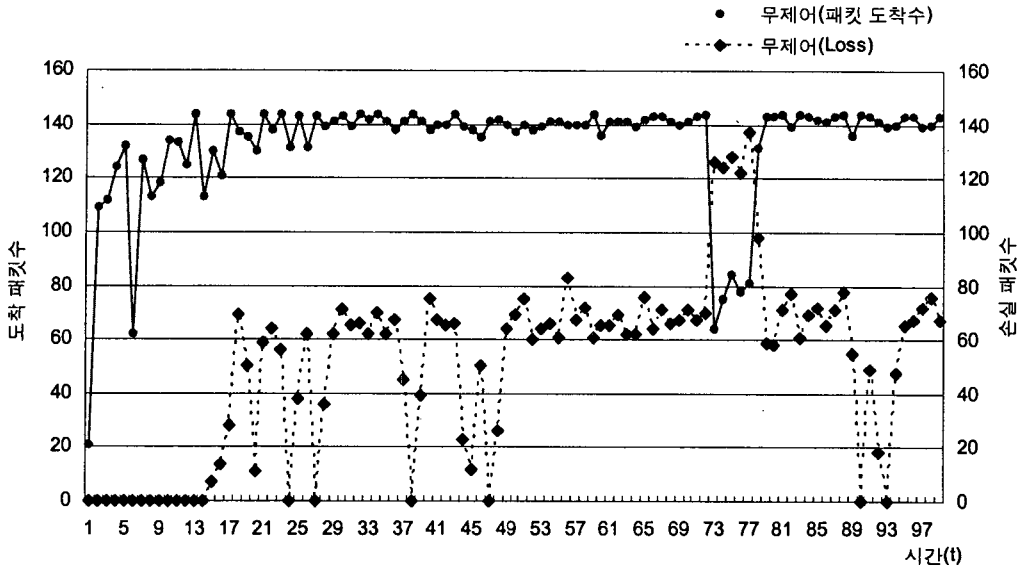


그림 4. 전송률 적응 제어하지 않는 경우: 패킷 손실과 도착 패킷수 관계

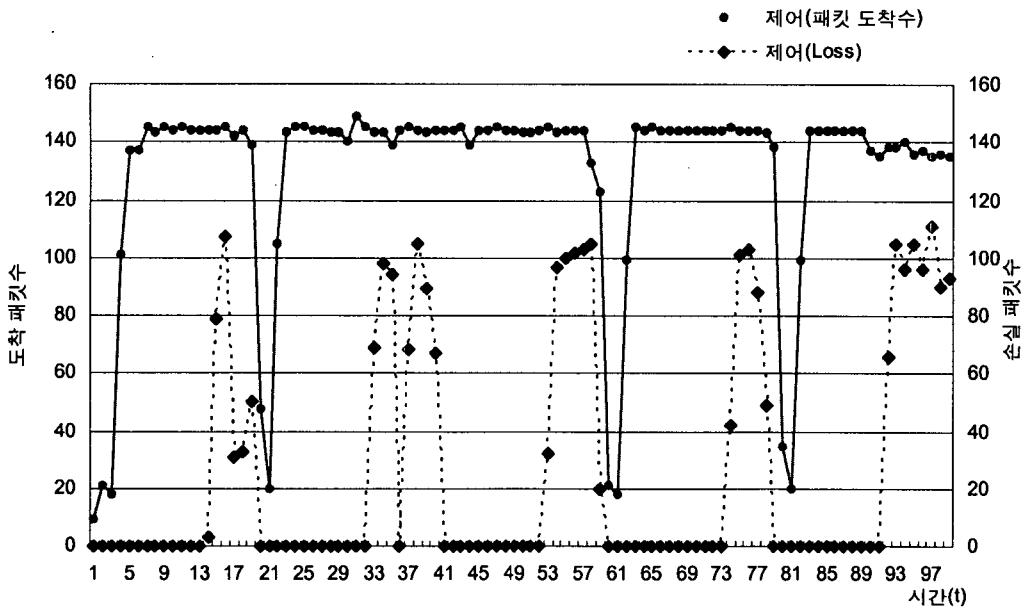


그림 5. 전송률 적응 제어 경우: 패킷 손실과 도착 패킷수 관계

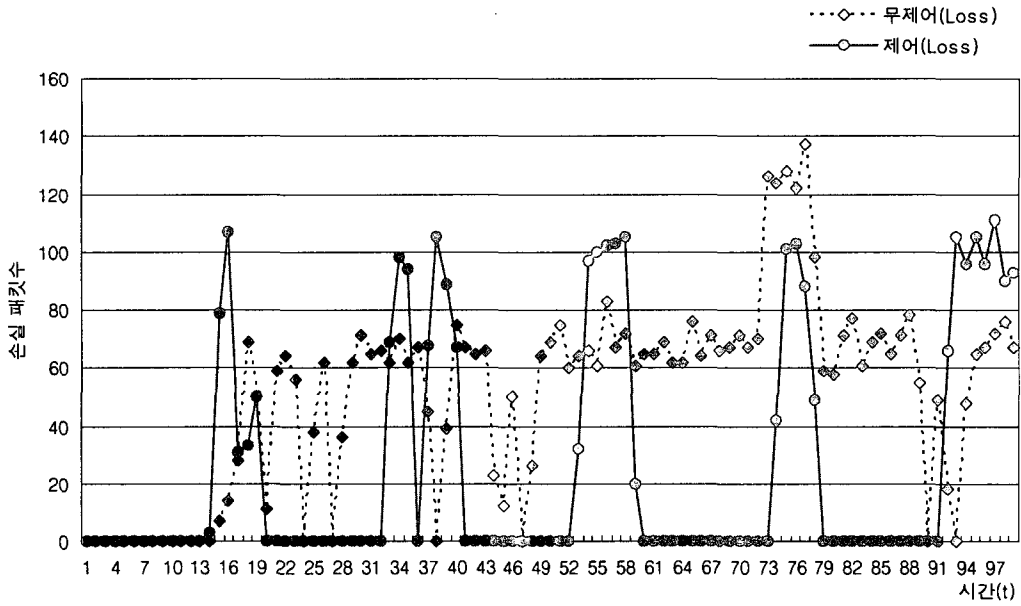


그림 6. 제어와 제어하지 않은 경우 패킷 손실 비교

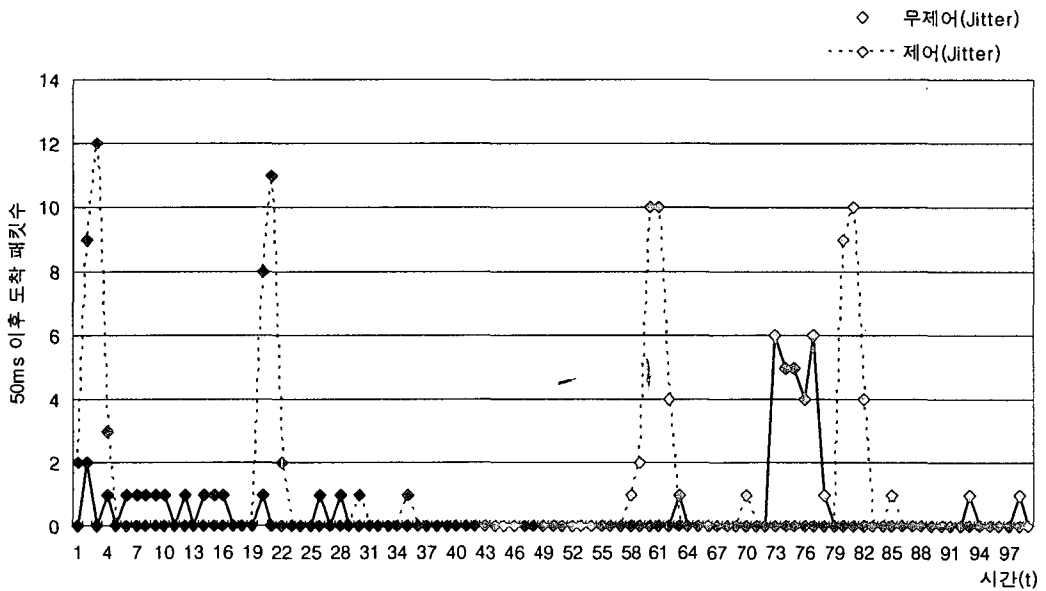


그림 7. 50ms 이후 도착한 패킷수 비교