1999 年度 大韓電子工學會 夏季綜合學術大會 論文集 第 22 卷 第 1 號 99/6

위성 인터넷망에서의 프록시 서버 기법 비교

김용신*, 김영한*, 전경재**, 안재영**
*숭실대학교 정보통신 전자공학부, **한국전자통신연구원 무선방송기술 연구소

Comparison of Proxy Server Mechanisms in Satellite Internet

Yong-Sin Kim*, Young-Han Kim*, Kyeong-jae Cheon**, Jae-young Ahn**
*School of electronic engineering, Soongsil University
**ETRI Radio & Broadcasting Technology Lab.

요 약

위성 인터넷망에서의 성능 향상을 위한 프록시의 대표적인 기법인 Snoop과 Spoofing에서의 TCP 성능을 모의실험을 통해 비교하고, 전체 메모리 크기에 따른 TCP 전송 특성 변화를 살펴보았다.

Spoofing과 Snoop에서 모두 TCP 전송율이 향상됨을 확인할 수 있었고, Spoofing을 적용했을 경우 Snoop보다 높은 전송율을 얻을 수 있었으나 데이터를 저장하기 위한 버퍼 요구량이 매우 커졌다. 메모리 크기는 제한되어 있으므로 효율적인 메모리 분배 방법이 필요함을 고찰할 수 있었다.

1. 서론

GEO 위성망에서는 590ms에 달하는 RTT(round trip time)뿐만 아니라, 유선망에 비해 상대적으로 높은 BER(bit error rate)로 인한 불필요한 혼잡제어가 전송율을 저해한다. 또한, 대역폭을 추정하는 slow-start 알고리즘과 매 RTT마다 한 개의 패킷 크기만큼 증가시키는 congestion avoidance 알고리즘은 RTT값이 큰 위성망에서 TCP 성능을 제한한다.

위성망에서 TCP 성능 향상을 위해 TCPSAT WG(working group)에서는 기존의 표준들을 이용한 방법들을 RFC[1]로 정리하고 연구가 진행중인 새로운 기법들을 draft[2]로 제안하였다. 그러나, 기존의 표준들을 이용한 방법에서도 TCP는 여전히 낮은 전송율을 보이며[3], 새로운 기법들이 표준화가 되어 실제 구현에 적용되기까지는 많은 시간이 소요될 것으로 예상된다. 이에 즉각적으로 위성망에 적용하여 TCP 성능을 향상시킬 수 있는 방법으로 위성링크에서의 문제점을 보안할 수 있는 프록시 서버를 이용하는 방법이 제안되었으며, 현재 PILC (performance implications of link characteristics) WG에서 논의되고 있다. 위성망 환경에 적합한 프록시 서버 구조를 결정하기 위하여 2장에서는 프록시의 대표적인 기법인 Spoofing과 Snoop을 서술하고 모의실험을 통하여 TCP 성능에 대한 영향을 살펴본다. 3장에서는 Spoofing에서 시스템 메모리 크기 제한에 따른 TCP 성능을 알아보고, 4장에서 결론을 내린다.

2. 프록시 서버 방식 (Spoofing, Snoop)

프록시 서버는 동작 방법에 따라 '연결 유지'와 '연결 분리' 방식으로 구분된다. Berkeley에서 제안한 Snoop은 '연결 유지' 방식의 대표적인 프로토콜로 TCP의 종단간 semantics를 보장하면서 링크 손실에 대해서 로컬하게 재전송을 수행하여 송신단에서 재전송하는 것을 방지한다. Snoop은 현재 무선 LAN에서 선택적으로 적용되고 있으며 TCP 성능을 향상시키는 방법으로 PILC의 draft[4]에서 LTN(Long Thin Network) 환경에 적용하도록 권장되고 있다. '연결 분리' 방식 프로토콜인 Spoofing은 DirecPC에 적용되고 있으며 초기 TCPSAT에서 논의되다가 성격상 PILC에서 다루게되었다.

Snoop과 Spoofing은 모두 유선망과 무선망의 중간에 위치한 프록시 서버에서 데이터 패킷을 캐쉬하고 링크 손실로 인해 재전송이 필요한 경우 송신단이 불필요하게 혼잡제어를 수행하지 않도록 재전송을 대행해주는 역할을 수행하는 공통점이 있다. 그러나, Spoofing은 추가적으로 spoofing ack를 생성해서 송신단에 전달한다. 그림 2.1에서 알 수 있듯이

Spoofing에서는 송신단에서 인식되는 RTT가 유선링크에서의 RTT로 줄어들게 되므로, RTT 또는 BER로 인한 TCP 성능 제한 요인을 제거할 수 있다. 특히, TCP는 slow-start 모드에서 낮은 전송율로 대역폭 추정을 시작하기 때문에 연결을 분리하여 인식되는 RTT값을 줄인 Spoofing은 위성망 환경에 적합하다. 그러나, spoofing ack를 전송한 후에 패킷들을 버퍼내에 계속 저장하고 있어야 하므로 위성 링크에서의 DB(delay*bandwidth)값에 해당되는 버퍼외에 링크 손실 복구과정 동안에 들어온 데이터를 저장할 수 있는 버퍼가 추가적으로 필요하다. Snoop에서는 재전송만을 수행하므로 송신단의 CWND(congestion window)만큼의 버퍼 크기만 필요하지만, 재전송 수행 후 Ack 패킷을 수신할 때까지 송신단은 휴지상태로 대기해야 하며 재전송 도중 송신단에서 timeout이 발생할 수 있는 단점이 있다. Snoop는 TCP end-to-end semantic을 요구하는 응용프로그램에 적합한 반면, Spoofing은 FTP, WWW 등의 bulk 데이터 전송등에 적합하다.

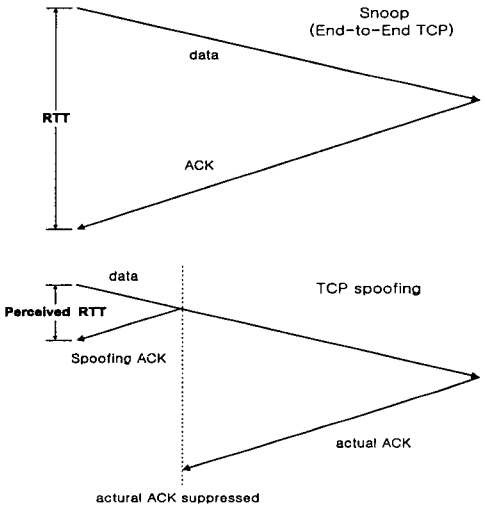
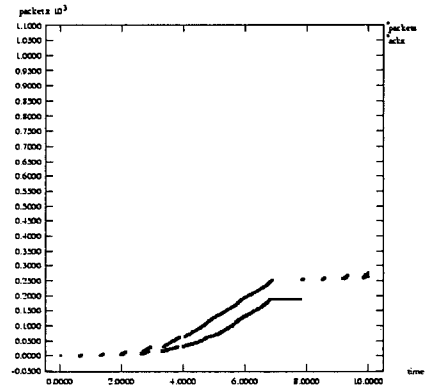


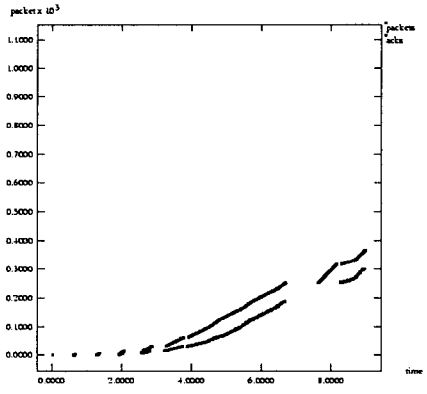
그림 1. Snoop과 Spoofing에서의 패킷 흐름도

좀더 정량적인 비교를 위해 모의 실험을 통해 TCP 전송율을 측정하였다. NS (Network Simulator)[5]를 이용하여 위성망의 중심국에 Spoofing과 Snoop 모듈을 추가한 후 시뮬레이션을 수행하였다. 유선 링크 대역폭, 위성 링크의 다운링크 대역폭, 단말국 수는 가변적으로 설정하였으며, 위성 링크의 업링크 대역폭은 9.6 Kbps로 설정하였다. 위성 링크의 BER은 10^{-6}으로 uniform 분포를 가정하였다. 데이터 패킷을 저장하기 위한 버퍼 크기는 Snoop의 경우 65 Kbytes, Spoofing의 경우 제한을 두지 않은 경우와 65 Kbytes로 설정한 경우를 비교하였다.

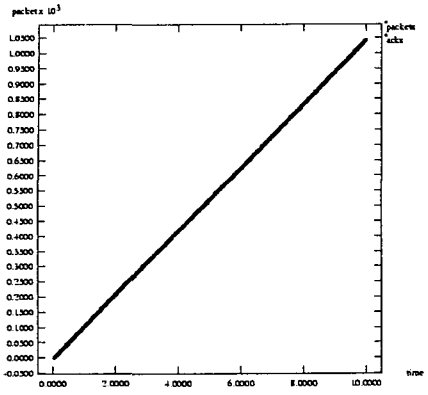
그림 2는 1 Mbps 대역폭 환경에서 각 프로토콜별 TCP 트래픽 전송 특성을 보여준다. 그림 2(b)는 Snoop 모듈을 사용했을때의 TCP 전송 특성을 보여준다. 2(a)에서 중복 Ack 수신시 송신단에서 전송율 저하가 발생하는데 비해 수신단으로부터의 중복 Ack는 TCP 송신단에 도달하기전에 중심국(기지국)의 snoop 모듈에서 처리하여 불필요한 혼잡제어를 막는다. Spoofing 모듈을 사용했을 경우 Snoop 보다 높은 전송율을 보였다. 중심국에서 송신단으로 spoofing ack를 생성하여 전송하므로 위성링크의 상태와 무관하게 TCP 전송이 이루어진다. 링크 손실에 대해서는 Snoop과 동일하게 중심국 버퍼에 저장된 패킷을 재전송하지만 복구과정동안에도 송신단은 전송을 계속할 수 있는 특성을 보인다. 대역폭 비대칭 환경에서는 그림 2(d)에서 볼 수 있듯이 단말국 출력 버퍼에서 Ack 패킷의 drop이 발생하지만 TCP 성능에는 영향을 주지 않았다.



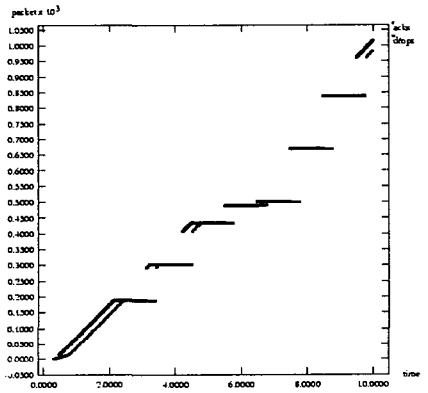
(a) TCP Tahoe



(b) Snoop



(c) Spoofing (data/spoofing ack)



(d) Spoofing (실제 Ack)

그림 2. 프로토콜에 따른 TCP 전송 특성

그림 3은 전송 파일 크기에 따른 TCP 전송율을 보여준다. Spoofing은 Snoop에 비해 크게 두 가지 장점이 있다. 첫째, 짧은 파일 전송시 전송율 증가율이 월등히 빠르다. 여러개의 짧은 TCP connection을 만들어서 전송되는 WWW 환경에서 월등히 높은 전송율을 얻을 수 있음을 알 수 있다. 둘째, 링크 손실 복구과정중에도 전송율이 감소되지 않는다. 또한, 연결 분리의 효과로 손실이 없는 환경에서 보다 훨씬 높은 전송율을 얻을 수 있었다.

그러나, Spoofing은 Snoop에 비해 요구되는 버퍼 크기가 매우 크다. 버퍼 크기를 Snoop과 동일하게 64 Kbytes로 제한한 경우에는 전송율이 감소되었다. 3장에서 버퍼 크기에 대한 제한이 있는 경우에 대해서 자세히 고찰한다.

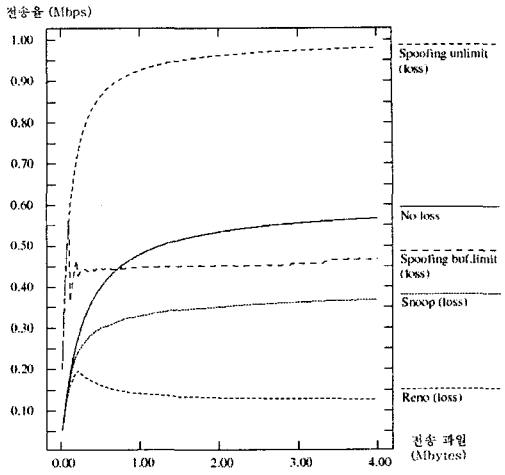
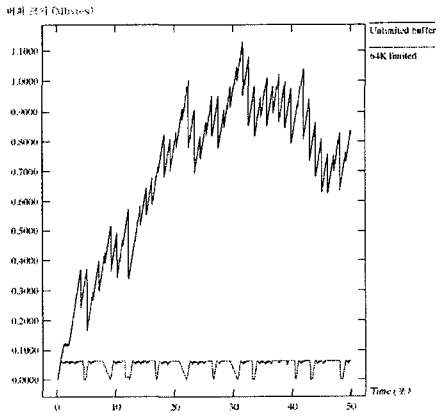


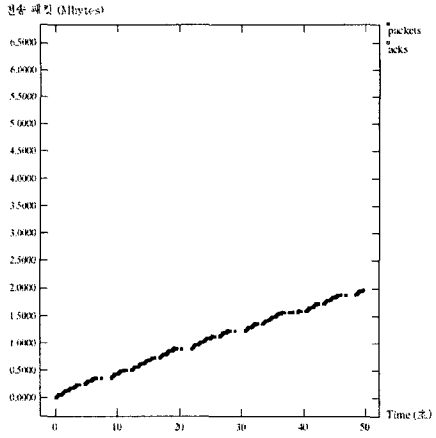
그림 3. 적용 프로토콜에 따른 TCP 성능

3. Spoofing에서의 버퍼 크기 제한

그림 4는 1Mbps 대역폭에서 Spoofing을 적용했을 때 단일 TCP 연결에 대한 성능을 보여준다. 4(a)는 버퍼의 크기에 제한이 없는 경우와 64 Kbytes로 제한된 경우에서의 버퍼크기 변화를 보여준다. 버퍼 크기 제한이 없는 경우 손실 복구과정 중에 다시 손실이 발생하면 버퍼크기가 계속 증가되는 결과를 보였다. 버퍼 크기에 제한이 있는 경우, 버퍼가 full이 되면 Spoofing은 spoofing ack를 전송할 수 없기 때문에 전송단은 더 이상 데이터 패킷을 전송할 수 없다. 그림 4(b)에서 볼 수 있듯이 버퍼의 제한이 없는 경우에 비해 낮은 전송특성을 보였다.



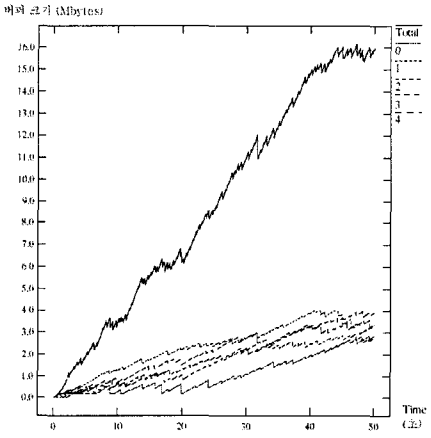
(a) 시간에 따른 버퍼 크기 변화



(b) 버퍼 크기 제한시 전송 특성

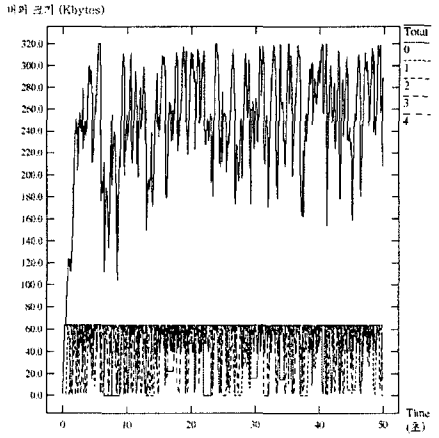
그림 4. 버퍼 크기에 따른 TCP 전송 특성 (1 Mbps, 1 TCP)

그림 5는 5개의 TCP connection들이 10 Mbps 위성 링크를 공유할때의 실험 결과이다. 버퍼 크기의 제한이 없을 경우 시간에 따른 버퍼 크기는 계속 증가하는 결과를 나타내었다. 반면, 각 TCP connection당 64 Kbytes로 버퍼 크기를 제한하였을 경우 최대 버퍼 크기가 일정한 값을 나타내었으나, TCP 전송 특성은 그림 4(b)에서 처럼 버퍼 제한으로 인한 전송율 저하 현상을 볼 수 있었다.



(a) 버퍼 크기 제한이 없을 경우

버퍼를 너무 작게 할당할 경우 그림 4(b)처럼 TCP 전송율이 감소되며 버퍼 크기를 너무 크게 할당하면 그림 5(a)처럼 전체 시스템 버퍼 요구량이 너무 커지는 문제가 있다.



(b) 64 Kbytes로 버퍼 크기 제한시

그림 5. 시간에 따른 버퍼 크기 변화 (10 Mbps, 5 TCP)

4. 결론

모의 실험을 통하여 비대칭 위성망 환경에서 Spoofing과 Snoop 프로토콜을 적용하였을 경우의 TCP 트래픽 특성을 관찰하였다. Spoofing과 Snoop에서 TCP 성능이 향상됨을 확인하였다. Slow-start 구간, 손실 복구 과정, 대역폭 비대칭 환경등에서 Spoofing은 Snoop에 비해 높은 성능을 나타내었다. 그러나, 데이터 패킷의 저장을 위한 버퍼 요구량을 무제한 수용할 수 없고 각 TCP 연결들이 사용하게 될 버퍼 크기가 가변적이기 때문에 효율적인 버퍼 분배 알고리즘에 대한 연구가 필요하다.

참고문헌

[1] Mark Allman, Dan Glover, "Enhancing TCP Over Satellite Channels using Standard Mechanisms", RFC 2488, January 1999

[2] Mark Allman, Spencer Dawkins, "Ongoing TCP Research Related to Satellites", INTERNET DRAFT, March, 1999

[3] 김용신, 김영한, 전경재, 최훈, "위성망에서의 TCP 성능 분석", 하계종합학술대회, 한국통신학회, p441-444, 1998

[4] G. Montenegro, Sun Microsystems Inc, "Long Thin Networks", INTERNET DRAFT, February, 1998

[5] S. McCanne, S. Floyd, "the LBNL Network Simulator", http://www-nrg.ee.lbl.gov/ns/