

유/무선 통신 환경에서 적응형 Snoop 모듈을 이용한 TCP 성능 분석

김명진 · 임세훈 · 김두용[†] · 김기완

[†]순천향대학교 전자공학과

Performance Analysis of TCP with Adaptive Snoop Module in Wired and Wireless Communication Environments

Myung Jin Kim, Sae Hoon Lim, Doo Yong Kim[†] and Kiwan Kim

[†]Dept. of Electronic Engineering, Soonchunhyang University

Abstract

TCP works well in wired networks where packet losses mainly occur due to congestion in data traffic. In wireless networks TCP does not differentiate packet losses from transmission errors or from congestion, which could lead to degrade the network performance. Several methods have been proposed to improve TCP performance over wireless links. Among them the Snoop module working at the base station is the popular method. In this paper, it is shown that the performance of Snoop largely depends upon the transmission link errors and the amount of data traffic. Also, our research shows that the local retransmission timeout value of Snoop can affect throughput. From the simulation results we suggest how to effectively use the Snoop algorithm considering data traffic and transmission link errors. It is expected that the proposed adaptive method will contribute to improving the network performance reducing the burden of the processes for data traffic.

Key Words : performance, TCP, congestion, snoop

1. 서 론

최근의 통신 환경은 유선 통신 환경의 비중이 감소하고 무선 및 이동 통신 환경의 비중이 폭발적으로 증가하고 있는 추세이다. 그러나 유선 통신 환경에서 개발된 인터넷 프로토콜인 TCP/IP 는 여전히 이러한 통신 환경에서 가장 중요한 통신 프로토콜이다. 그러므로 유/무선 통합 통신 환경에 적합하도록 인터넷 프로토콜들을 개발하고 개선하려는 노력이 필요하다. 이러한 노력 가운데 TCP 전송 프로토콜의 성능을 개선시키려는 연구가 활발히 진행되어 왔다[1].

인터넷 TCP 프로토콜은 전송 중에 발생하는 에러로 인한 패킷 손실과 네트워크 트래픽의 혼잡에 의한 패킷 손실을 구별할 수 없기 때문에 전송 중에 에러가 발생하기 쉬운 무선 통신 환경에서 제대로 성능을 발휘

할 수 없다. 예를 들면, 패킷 손실로 인해 타임아웃이 발생할 경우 TCP는 혼잡윈도우 크기를 1로 줄이고 혼잡제어 알고리즘인 slow-start 과정을 시작하게 된다. 따라서 타임아웃이 발생할 때 네트워크의 실제적인 혼잡이 발생한 것이 아니라 전송 중에 무선 링크의 에러로 인한 타임아웃이 발생한 것이라면 불필요하게 혼잡윈도우 크기를 줄이게 되므로 심각하게 데이터 처리율을 낮추고 이에 따라 전체적인 데이터의 지연을 초래하게 됨으로 네트워크의 성능을 훼손시키게 된다. 따라서 이러한 단점을 극복하고자 링크 계층에서 동작하는 Snoop 모듈이 일찍이 제안되고 있다[2].

출발지인 유선 호스트에서 목적지인 무선 이동 호스트로 패킷을 보낼 때 기지국에서 동작하는 Snoop 모듈은 패킷을 캐쉬에 보관한 다음 로컬 재전송 타이머를 설정한 후에 이동 호스트로 전송한다. 캐쉬에 보관된 패킷은 이동 호스트로부터 확인 응답이 도착하면 캐쉬에서 제거된다. 패킷 손실을 나타내는 중복된 확인응답

[†]E-mail : dooykim@sch.ac.kr

을 snoop 모듈이 전송 받는다면 캐쉬에 보관된 패킷을 이동 호스트로 재전송 한다. 이때 중복된 확인 응답은 기지국의 snoop 모듈에 의해 출발지로 보내지지 않기 때문에 불필요한 혼잡 제어 알고리즘이 실행되는 것을 막을 수 있다. 또한 Snoop 모듈에서 설정된 재전송 타이머에서 타임아웃이 발생할 경우에도 유선의 출발지 호스트의 재전송 타임아웃에 관계없이 캐쉬에 보관된 데이터를 재전송 한다. 이와 같은 방법으로 기지국에 있는 Snoop 모듈을 통해 무선 노드의 불안정한 링크 상태에서부터 발생할 수 있는 패킷 손실을 최소화 할 수 있다[3].

무선 링크에서 TCP의 성능을 개선시키기 위한 많은 연구들이 크게 두 가지 방향으로 이루어져왔다[4-6]. 하나는 위에서 언급한 가장 널리 알려진 링크 계층에서 Snoop 모듈을 이용하는 것이며 또 다른 것은 Tang 이 발표한 바와 같이 무선 링크의 에러로 인한 손실과 혼잡으로 인한 손실을 구별하기 위한 방법 등을 제안하는 것이다[7]. 이러한 방법은 기존의 네트워크 인프라와 프로토콜 등을 수정하는 것이 필요하므로 실제 네트워크 상황에 적용하기가 어려운 단점이 있다.

따라서 본 논문에서는 현재 기지국에서 쉽게 적용 가능한 Snoop 모듈을 이용하여 데이터 트래픽과 무선 데이터 링크의 에러율 등이 Snoop 모듈의 성능에 어떤 영향을 미치는지를 분석하고 Snoop 모듈에서 데이터를 무선 노드로 전송할 때 패킷 손실로 인한 재전송 시간을 설정하기 위해 사용하고 링크 계층에서 동작하는 로컬 재전송 타임아웃 값에 따른 네트워크 성능도 분석한다.

본 연구를 위해 네트워크 시뮬레이터인 ns2에서 제공하는 Snoop 모듈을 이용하여 성능을 분석한다[8]. 그리고 시뮬레이션 실행을 통해 얻은 결과를 이용하여 Snoop 모듈을 효과적으로 사용하는 방법을 제시한다.

본 논문의 구성은 제 2장에서 ns2 네트워크 시뮬레이터가 허용하는 계층적 주소를 이용하여 유/무선이 결합된 시뮬레이션 모델에 대해 설명하고 이를 바탕으로 다양한 네트워크 시나리오의 시뮬레이션으로부터 얻은 결과를 분석하며 제 3장에서는 시뮬레이션 결과를 토대로 Snoop 모듈을 기지국에서 효율적으로 사용하기 위한 방안을 제시하고 그리고 제 4장의 결론으로 이루어져 있다.

2. 시뮬레이션 모델 및 결과

다음 그림 1은 본 논문에서 사용하는 유/무선이 결합되고 각 노드들 사이의 대역폭과 지연시간을 나타내

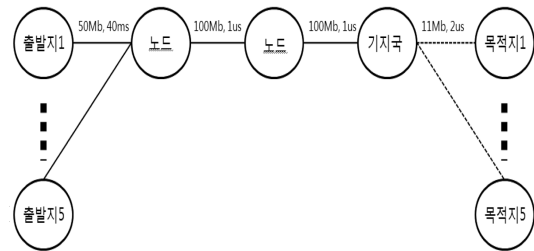


Fig. 1. Simulation model with wired and wireless nodes.

는 네트워크 시뮬레이션 모델을 보여준다. 그림으로부터 각 출발지의 데이터 트래픽은 경유 노드와 기지국을 거쳐 해당하는 이동 호스트인 목적지로 전송된다. 출발지와 기지국 사이는 유선으로 연결되어 있으며 기지국과 목적지 사이는 무선으로 연결되어 있고 Snoop 모듈은 기지국에서 동작한다고 가정한다. 그리고 무선 링크의 에러 모델은 Uniform 분포를 사용한다.

2.1. 시뮬레이션 결과 및 분석

앞서 언급한 네트워크 모델을 이용하여 얻은 시뮬레이션 결과로부터 다음 그림들은 트래픽양과 링크 에러율에 따른 Snoop 기능의 효율성을 나타낸다. 그림 2에서 보는 바와 같이 출발지와 목적지가 각 한 쌍으로 트래픽 양이 상대적으로 많지 않을 때 링크 에러가 증가할 경우에 Snoop 기능을 사용하는 것이 훨씬 처리율을 높일 수 있다는 것을 알 수 있다. 반면에 그림 3으로부터 트래픽이 증가할 경우에는 Snoop 기능을 적용하여 얻는 처리율의 증가가 현저히 떨어진다라는 것을 알 수 있으며 그림 4로부터 더욱 많은 양의 트래픽이 존재한

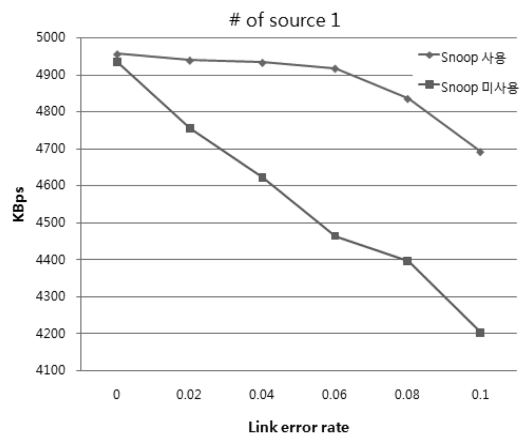


Fig. 2. Throughput for the number of sources = 1.

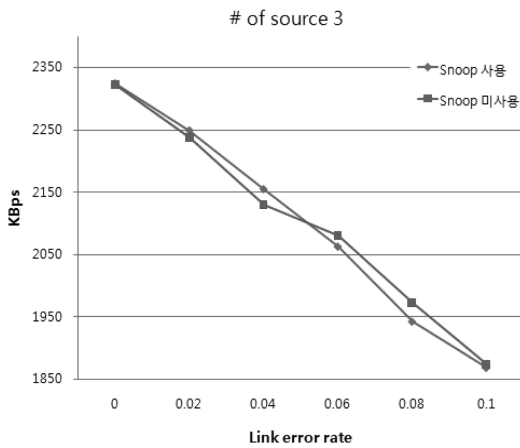


Fig. 3. Throughput for the number of sources = 3.

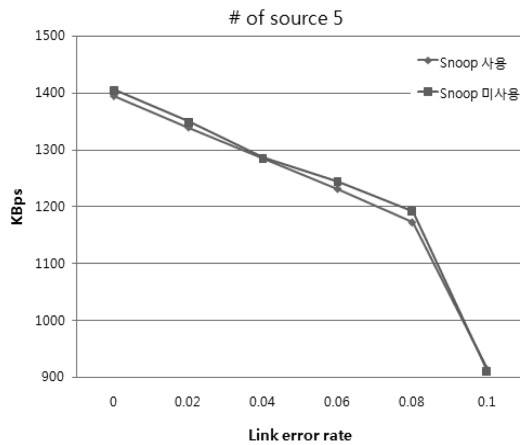


Fig. 4. Throughput for the number of sources = 5.

다면 오히려 Snoop 기능을 적용하는 것이 처리율을 감소시킬 수 있다는 것을 보여준다. 이러한 결과는 기지국에서 처리하는 트래픽양이 많고 링크 에러가 존재할 경우에 Snoop 기능을 사용하는 것이 오히려 기지국에 있는 서버의 트래픽 처리 능력에 부담을 주는 결과를 초래하여 데이터 처리율을 감소시킬 수 있다는 것을 보여준다.

따라서, Snoop 모듈을 실제 상황에 적용하기 위해서는 현재 기지국에서의 네트워크 트래픽 상황과 링크 에러율을 파악한 다음 이를 기준으로 Snoop 기능을 사용할 것인지를 판단하는 것이 필요하다. 현재 기지국은 트래픽과 링크 에러율을 실시간으로 측정할 수 있으므로 본 논문의 제3장에서 제안하는 방법을 쉽게 적용할 수 있다.

2.2. Snoop 모듈의 Round-Trip Time

일반적으로 RTT(Round-Trip Time) 값을 이용 하여 RTO(Retransmission Time Out) 값을 설정할 때 아래의 식(1)을 사용한다[9].

$$RTO = SRTT + f * RTTVAR \quad (1)$$

여기서 SRTT 는 smoothed round-trip time, f는 정수 값, 그리고 RTTVAR 은 round-trip time variance를 나타낸다.

본래, 참고문헌[10]에서 초기에 f 값으로 2를 제안했으나 후에 연구를 통해 4를 제안하였다. 그런데 이러한 연구 결과를 얻은 네트워크 환경은 주로 유선 인터넷으로 효과적인 혼잡제어 알고리즘을 위해 실제적인 f 값으로 4를 제안하는데 목적이 있었다. 따라서 이때의 혼잡은 네트워크상의 수용 능력을 넘어 발생한 많은 양의 트래픽에 기인한 것으로 간주한다.

한편, 무선 환경에서 TCP 프로토콜의 혼잡제어의 경우 실제 혼잡이 발생하지 않은 상황인데도 불구하고 무선 링크 에러로 말미암아 특정 패킷의 손실로 인해 타임아웃이 발생할 수 있다. 이로 인해 TCP의 혼잡제어 알고리즘이 작동하여 송신 윈도우의 느린 시작 과정 등으로 복귀하여 불필요하게 송신 윈도우의 크기를 줄이므로 네트워크의 성능을 저하 시킨다. 그러므로 Snoop 모듈에서 기존의 RTT 값 적용의 문제점은 실제 증가된 RTT 값으로 인해 한 패킷의 손실에 대한 타임아웃이 어떻게 발생했는지에 대한 정보가 없는 것이다. 즉, 무선 링크의 에러 때문인지 혹은 실제 네트워크상의 트래픽으로 인한 혼잡이 발생하여 RTT 값이 증가하여 타임아웃이 일어난 것인지에 대한 구별된 정보가 없는 것이다. 본 논문에서는 만약 구별하여 적용할 수 있다면 일정 부분의 성능 향상을 꾀할 수 있다는 것을 보여준다. 다음 그림들은 ns2에서 제공하고 있는 MySnoop.tcl 파일을 이용하여 본 논문에서 제안하는 방법을 적용하여 얻은 결과를 나타낸다. 그림 5로부터 링크 계층 Snoop 모듈의 로컬 재전송 루틴에서 사용하는 RTO의 f값을 2로 설정할 경우 처리율에서 평균 약 5%이상의 증가 효과를 보여주고 있다. 이것은 그림 6의 RTT값에서 알 수 있듯이 f값을 2로 설정하는 것이 4로 설정할 때보다 더 작은 RTO값을 얻을 수 있으므로 이것이 전체적인 처리율의 증가현상으로 나타나는 것을 알 수 있다. 이러한 현상은 재전송이 혼잡에 기인한 것이 아니고 링크 에러로 인한 손실 패킷에 대한 타임아웃 값을 상대적으로 일찍 발생시킬 수 있으므로 f

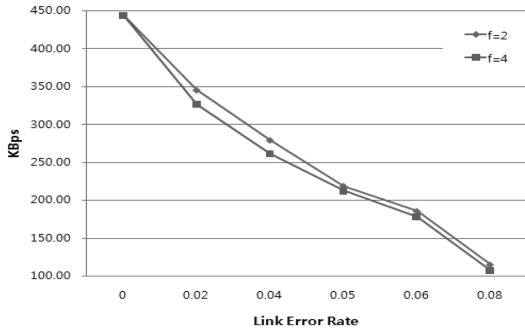


Fig. 5. Throughput with the values of f.

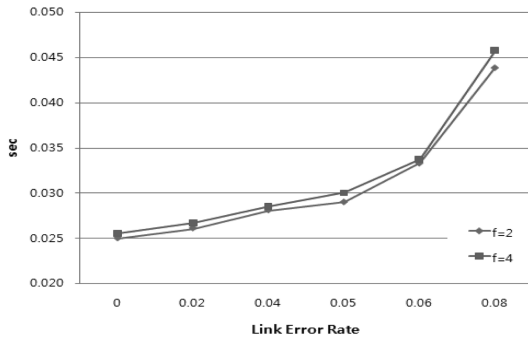


Fig. 6. Local round-trip time of snoop with the values of f.

값으로 2를 사용하는 경우가 재전송이 Snoop 모듈에서 일찍 발생하므로 이는 곧 작은 지연으로 이어지고 이로 인해 더 나은 처리율을 얻을 수 있다는 것을 보여 준다. 따라서 많은 양의 데이터 트래픽에 의한 네트워크 혼잡 상황이 아니라 링크 에러에 따른 손실 패킷이 발생하여 타임아웃이 발생한다면 RTO의 f값을 2로 설정하여 가능한 빨리 패킷을 재전송하므로 처리율을 높일 수 있는 것을 알 수 있다.

3. 개선된 Snoop 모듈 운영 방법

본 논문에서는 제 2장의 시뮬레이션 수행으로부터 얻은 결과를 토대로 Snoop 모듈을 기지국에서 사용할 때 그림 7과 같은 새롭게 개선된 Snoop 모듈의 적용 방법을 제안한다.

그림에서 보는 바와 같이 기지국에 패킷이 도착하면 현재 기지국에서 네트워크를 통해 측정된 트래픽과 무선 링크의 에러 상태 정보와 그리고 그동안 기지국을 운용하면서 얻은 트래픽과 링크 에러율에 따른 Snoop

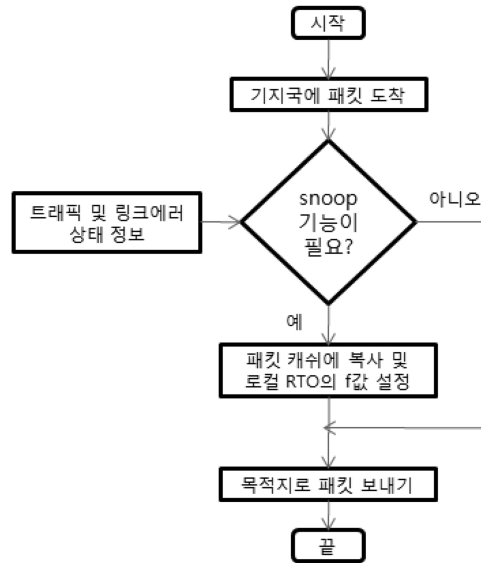


Fig. 7. Adaptive snoop module.

기능의 운영 결과를 이용하여 Snoop 기능을 적용할지를 판단한다. 따라서 이러한 정보를 토대로 만약 Snoop 기능이 필요하다면 Snoop 모듈을 사용하여 패킷을 목적지로 보낸다. 또한 위에서 살펴본 바와 같이 Snoop 모듈의 로컬 재전송 타임아웃값을 어떻게 설정하느냐에 따라 네트워크 성능에 영향을 미칠 수 있으므로 로컬 재전송 타임아웃값을 목적지의 무선 통신 환경에 적합하게 설정하여 패킷을 목적지로 전송한다.

그러나 Snoop 기능이 필요 없다고 판단된다면 Snoop 모듈을 통하지 않고 즉 패킷들을 캐쉬에 복사하여 저장하지 않고 바로 목적지로 보내어 기지국에 있는 서버의 부담을 덜어준다. 따라서 Snoop 모듈을 사용하지 않기 때문에 많은 양의 데이터를 서버의 캐쉬에 저장하지 않아 데이터 처리 속도를 높일 수 있으므로 전체적인 네트워크의 성능 향상에 기여할 수 있다.

4. 결 론

최근 스마트폰과 노트북 등을 이용한 무선 통신의 비중이 폭발적으로 증가하면서 한층 더 인터넷 TCP 전송 프로토콜의 성능 개선이 요구된다. 많은 개선 노력 가운데 기존의 TCP 기능을 수정하지 않고 무선 통신 링크의 에러에 취약한 성능을 나타내는 단점을 보완하고 TCP와 쉽게 연동하여 사용할 수 있으며 링크 계층에서 동작하는 snoop 모듈이 제안되고 있다. 그러나 본 논문의 시뮬레이션 결과를 통해 Snoop 모듈의

성능은 데이터 트래픽 양과 무선 링크의 에러율의 크기에 따라 처리율에 큰 영향을 준다는 것을 알 수 있다. 또한 Snoop 모듈에서 혼잡에 의한 패킷 손실이 아니라 무선 링크 에러로 인한 패킷 손실로 인해 Snoop 모듈에서 사용하는 로컬 재전송 타임아웃이 발생할 때 목적이인 이동 호스트가 기지국을 중심으로 한 휴 정도 떨어져 있다고 가정할 경우 RTO의 값을 2로 설정함으로써 상대적으로 빨리 재전송이 일어나 결과적으로 데이터의 처리율을 높일 수 있다는 것을 알 수 있다. 그러므로 Snoop 모듈을 사용할 경우 데이터 트래픽 상황과 무선 링크의 에러율 등을 고려하는 것이 필요하다. 그렇지 않을 경우 특정한 트래픽과 무선 링크의 에러에서 오히려 Snoop 모듈을 사용하는 것이 기지국에 과부하를 주어 네트워크 성능을 감소시킬 수 있다는 것을 보여 준다.

따라서 본 논문에서 제안한 방법으로 Snoop 모듈을 적용한다면 기지국에서 서버의 데이터 처리 능력에 대한 부담을 완화할 수 있으므로 네트워크 자원을 효율적으로 사용하는데 기여할 수 있을 것이다.

감사의 글

이 논문은 2011학년도 순천향대학교 교수 연구년제에 의하여 연구하였음

참고문헌

1. 김대영, “미래인터넷 개념 및 현황,” 정보와 통신, vol. 27, Oct. 2010, pp. 3-10.
2. H. Balakrishnan, S. Seshan, and R. Katz, “Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks,” ACM Wireless Networks,

- vol. 1, Dec 1995.
3. N. Vaidya et al., “Delayed duplicate acknowledgements: a TCP-unaware approach to improve performance of TCP over wireless” Technical Report 99-003, Dept. of computer Science, Texas A&M University, Feb. 1999.
4. S. Lee, S. Park, J. Yu, and C. Hong, “A TCP Acceleration Algorithm for a Wireless Link using Rate Adaptation Based on Round-Trip-Time and Virtual Receiver Window Information,” IEEE Comm. Letters, vol. 13, No. 9, Sept. 2009, pp. 655-657.
5. Y. Wu, Z. Niu, and J. Zheng, “A network-based solution for TCP in wireless systems with opportunistic scheduling,” in Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '04), vol. 2, pp. 1241-1245, Barcelona, Spain, September 2004.
6. M. Chen and A. Zakhor “Aio-TFRC : a light-weighted rate control scheme for streaming over wireless,” Proc. IEEE WirelessCom Symp. Multimedia over Wireless 2005.
7. J. Tang, G. Morabito, I. Akyildiz, and M. Johnson, “Rcs: A Rate Control Scheme for Real-time Traffic in Networks with High Bandwidth-delay Products in High Bit Error Rates,” in Proc. IEEE INFOCOM, Alaska, USA, Apr. 2001, pp. 114-122.
8. <http://www.isi.edu/nsnam/ns>
9. W. Stallings, Data and Computer Communications, Prentice Hall, 2004.
10. V. Jacobson, “Congestion Avoidance and Control,” Proc. SIGCOMM '88, Computer Communication Review, Aug. 1988.

접수일: 2011년 8월 29일, 심사일: 2011년 9월 8일,
 게재확정일: 2011년 9월 15일