

무선 환경에서 802.11 MAC의 MIB 정보를 이용한 TCP 성능 개선

*김기원, 신광식, 윤완오, 최상방
인하대학교 전자공학과

e-mail :objectkk@gmail.com, sangbang@inha.ac.kr

TCP Performance Improvement using 802.11 MAC MIB for Wireless Network

*Ki-Won Kim, Kwang-Sik Shin, Wan-Oh Yoon, Sang-Bang Choi
Dept. of Electronic Engineering
Inha University

Abstract

TCP applied to the wireless-wired integrated network is the one that was applied to the existing wired network. In the wireless-wired integrated network, both wireless and congestion loss can occur. When wireless packet losses occur, the congestion control of TCP causes performance degradation by reducing its transmission rate. In this paper, we propose the algorithm to distinguish the wireless packet loss from congestion packet loss using MIB of the 802.11 MAC which has been generally used recently in wireless links.

I. 서론

패킷 손실이 발생했을 때 TCP Reno는 네트워크 혼잡이 발생할 것으로 간주하고, 혼잡 제어 알고리즘에 의해 혼잡 윈도우를 반으로 줄임으로써 전송률을 반으로 줄이게 된다. 이것은 유선 링크로만 이루어진 기존 네트워크에는 적합하지만 네트워크 혼잡과는 무관한 무선 패킷 손실이 발생하는 네트워크에는 그렇지 못하다. 따라서 패킷 손실이 발생했을 때 무선 손실과 혼잡 손실을 구분하는 알고리즘이 필요하다. 본 논문에서는 현재 무선 링크에서 가장 많이 사용되는 802.11 표준의 MIB 정보를 이용하여 패킷의 손실 원인을 구분하는 알고리즘을 제안한다. 제안된 알고리즘은 기존의 다른 손실 구분 알고리즘들이 사용하는 ROTT나

RTT를 이용하지 않고 실제 전송을 담당하는 802.11의 로그 정보를 이용함으로써 손실 구분의 정확성을 향상시킨다. 손실 구분 능력이 향상됨으로써 불필요한 전송률 감소를 방지하고, 결과적으로 TCP 전송 성능을 향상시킬 수 있다.

II. 관련 연구

2.1 Bias 알고리즘

Bias 알고리즘은 패킷의 도착 간격 시간을 이용하여 패킷 손실을 구분한다. 이 알고리즘은 그림 1의 무선 손실 윈도우를 이용하여 동작하게 된다. T_{min} 은 패킷의 도착간격 시간의 최소값을 의미한다. n 개의 패킷 손실이 발생하였을 때, P_i 는 순차적으로 받은 마지막 패킷을 의미하고, P_{i+n+1} 는 패킷 손실 이후 비순차적으로 받은 패킷을 의미한다. 그리고 P_i 와 P_{i+n+1} 의 도착간격 시간은 T_i 가 된다. 각 패킷의 크기는 같다는 가정 하에 $(n+1)T_{min} \leq T_i \leq (n+2)T_{min}$ 이면, n 개의 손실된 패킷은 모두 무선 전송 오류로 추정하고, 만약 그렇지 않다면 혼잡 손실로 추정한다.[1] 본 논문의 시뮬레이션에서 비교하게 되는 mBias 알고리즘은 Bias 알고리즘을 최적화한 알고리즘이다.

2.2 Spike 알고리즘

Spike 알고리즘은 $B_{spikestart}$ 와 $B_{spikeend}$ 를 이용하여 Spike 상태를 설정하게 되는데 구하는 방법은 식(1), (2)와 같다. ROTT 값이 $B_{spikestart}$ 보다 커지게 되면 Spike 상태로 전환이 되고, ROTT 값이 $B_{spikeend}$ 보다 작아질 때까지 발생하는 패킷 손실은 모두 혼잡 손실

로 간주하게 된다. 반면, ROTT가 $B_{spikeend}$ 보다 작아지면 Nonspike 상태가 되고 다시 $B_{spikestart}$ 보다 커질 때까지 발생하는 패킷 손실은 모두 무선 손실로 간주한다. 단, $\alpha < \beta$ 이다.[2]

$$B_{spikestart} = rott_{min} + \alpha \times (rott_{max} - rott_{min}) \quad (1)$$

$$B_{spikeend} = rott_{min} + \beta \times (rott_{max} - rott_{min}) \quad (2)$$

III. 본 론

3.1 802.11 표준의 패킷 손실 과정

본 논문에서 다루고 있는 802.11 표준에서의 패킷 손실 과정은 다음과 같다. 송신측에서 패킷을 전송하고 ACK가 돌아오는 타이머를 진행시킨다. 패킷을 받은 수신측은 FCS 에러 체크를 하게 된다. 패킷에 문제가 없다면 ACK를 보내지 않는다. 패킷에 문제가 생겨 ACK를 받지 못하게 된 수신측에서는 타이머가 완료되고, 해당 패킷을 재전송하게 된다. 이와 같은 과정이 반복되어 재전송 횟수 제한에 다다르면, 송신측에서 해당 패킷의 전송을 포기하고 다음 패킷을 전송하게 된다. 이러한 패킷 손실 과정들은 MIB-II에 항목별로 기록되게 되어 있는데, 본 논문에서는 이렇게 802.11 표준에서 패킷 손실이 일어나는 과정 중 기록된 정보들을 이용하여 손실의 유형을 파악하는데 이용한다. dot11FailedCount(OID : 1.2.840.10036.2.1.1.5)과 dot11FCSErrorCount(OID : 1.2.840.10036.2.2.1.10) 속성을 이용하게 되는데 전자는 패킷 손실로 인한 재전송 시 재전송 횟수 제한에 도달하여 전송이 포기된 패킷의 수이다.[3] 후자는 FCS 에러가 발생한 패킷의 수를 나타낸다. 이러한 MIB 속성들을 열람하기 위해서는 그림 1.에서와 같이 TCP계층과 MIB를 관리하는 MAC mgmt 계층 간에 별도의 인터페이스가 필요하다.

3.2 손실 구분 알고리즘

수신측에서는 패킷 i 를 받은 후 패킷의 크기를 T/M 인터페이스를 통해 dot11RTSThreshold 속성과 비교한다. 이 속성의 기본값은 2347 바이트이고, 전송하는 패킷이 값보다 크거나 같으면 dot11LongRetryLimit, 작으면 dot11ShortRetryLimit이 적용되어 재전송 횟수를 제한하게 된다. 그 후 패킷 $i-1$ 이 도착했을 때의 dot11FCSErrorCount _{$i-1$} 과 dot11FCSErrorCount _{i} 의 차 Δ dot11FCSErrorCount를 확인한다. 이 값이 패킷의 크기에 따른 재전송 횟수 제한(dot11LongRetryLimit, dot11ShortRetryLimit)보다 크거나 같으면 현재 받은 패킷 이전의 패킷이 무선 링크에서 전송이 포기되었다는 것을 알 수 있다. 즉, 패킷 $i-1$ 과 패킷 i 사이의 패킷이 무선 링크에서 손실되었음을 알 수 있다. 따라서 방금 패킷의 무선 손실이 있었음을 송신측에 알려 송신측의 불필요한 혼잡 제어를 방지해야 한다. 이것은 TCP 헤더의 사용하지 않는 비트 중 하나를 무선 손실을 나타내는 비트로 설정함으로써 가능해진다. 마지막으로 도착한 패킷의 ACK를 전송할 때 TCP 헤더에 미리 지정해둔 무선 손실을 나타내는 비트를 체크하여 전송하게 된다. 송신측에서는 3개의 중복된 ACK가 도착했을 때, 3개중 헤더에 무선 손실 비트가 설정되어 있는 ACK가 있으면 전송률을 줄이지 않고 계속 증가시킨다. 또한 송신측에서는 각각의 패킷을 전송하기 전에 dot11FailedCount 속성을 확인하여 바로 전에

보낸 패킷이 무선 링크에서 손실 됐는지를 확인한다. 이 값이 증가했을 경우 무선 손실이 발생한 것으로 판단하고 바로 재전송하게 된다.

IV. 시뮬레이션

4.1 시뮬레이션 결과

TCP Reno, mBiaz, Spike와 제안된 알고리즘에 적용하여 데이터 전송량을 비교하는 시뮬레이션을 하였다. 그림 2.는 각각의 에러율에 따른 전송률을 나타낸다. 에러율이 1%인 시나리오에서 TCP Reno에 비해 34%의 성능 향상을 보여 준다. 에러율에 10%인 경우 TCP의 전송률은 급격히 떨어지게 되고 비교가 무의미하다. 하지만 에러율이 높은 링크를 포함하는 연결에서는 원활한 전송을 위해 손실 구분 알고리즘이 반드시 필요하다는 것을 보여준다.

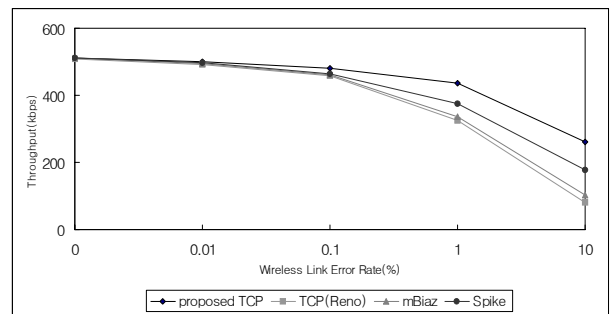


그림 1. 에러율에 따른 전송률.

V. 결론 및 향후 연구 방향

유·무선 통합 네트워크에서 TCP가 패킷의 무선 손실과 혼잡 손실을 구분하여 불필요한 전송률 저하를 막을 수 있다. 본 논문에선 제안된 손실 구분 알고리즘은 802.11 MAC의 MIB 정보를 이용하여 기존의 ROTT를 사용하는 기법에 비해 향상된 성능을 보였다. 그러나 하나의 노드에서 이중 프로토콜이 동시에 진행될 경우 성능이 저하되는 특성을 보인다. 이중 프로토콜의 동시 진행시 성능 향상 방법에 대한 연구가 더 진행되어야 할 것이다.

참고문헌

- [1] S. Biaz 외, "Discriminating congestion losses from wireless losses using interarrival times at the receiver", in Proc. IEEE Symp. Application-Specific Systems and Software Engineering and Technology, Richardson, TX, Mar. pp. 10-17, 1999.
- [2] Y. Tobe 외, "Achieving moderate fairness for UDP flows by path-status classification", in Proc. 25th Annu. IEEE Conf. Local Computer Networks(LCN 2000), Tampa, FL, pp. 252-261, 2000.
- [3] Bob O'Hara 외, "IEEE 802.11 handbook: A designer's Companion", Standards Information Network, IEEE Press, pp. 241-250, 2005.