

유무선 혼합 망에서 Snoop과 수신신호(received signal strengths)의 상호관계를 통한 네트워크 성능 개선

김진희*, 권경희*
*단국대학교 전자계산학과
e-mail : whitej2@dankook.ac.kr

The network efficiency improvement which uses the Snoop and Received signal strengths over wired-wireless networks

Jin-Hee Kim, Kyung-Hee Kwon
*Dept. of Computer Science, Dan-Kook University

요 약

다양한 무선 네트워킹의 출현은 유무선 혼합형태의 네트워크 망으로 변화되면서 기존의 유선 중심의 네트워크 기반으로 구현된 TCP에 많은 문제점을 가져왔다. 높은 비트오류율, 낮은 대역폭 그리고 높은 지연 등은 TCP/IP 통신을 효율적으로 지원하지 못하는 원인이 된다. 또한 BS(Base Station:기지국)에서 보낸 패킷이 MH(Mobile Node : 이동성을 갖는 노드)가 수신범위 밖으로 벗어나면서 손실로 이어질 경우 성능 저하의 원인이 되기도 한다. 본 논문에서는 MN의 ACK에 수신신호 관련 flag bit를 추가하면서 BS에서 추가된 ACK의 flag bit를 이용해서 패킷손실을 최소화하는 기법을 제안한다.

1. 서론

최근 몇 년 동안 인터넷 수요의 증가는 인터넷 서비스의 다양화를 가져왔고, 이러한 서비스는 무선통신으로 이어지면서 다양한 무선 네트워킹 기술을 등장하게 하였다. 무선통신의 등장으로 인터넷 환경은 자연스럽게 기존의 유선망과의 혼합으로 연결되면서 여러가지 해결해야 할 문제점을 가져왔다. 무선망에서의 노드의 이동성은 새로운 라우팅 기법과 주소할당방식을 요구하게 되었으며, 비트 오류율과 낮은 대역폭 그리고 높은 지연등은 유선망 중심의 TCP를 무선망을 고려하여 구현된 TCP의 발전으로 이어지게 하였다[2][3][4][5].

기존의 네트워크망은 유선망 형태로 대부분의 패킷 손실은 혼잡으로 인한 것으로 간주한다. 그러나 무선망은 앞서 이야기 했듯이 혼잡으로 인한 손실이기 보다는 낮은 대역폭과 비트에러, 노드의

이동성과 그로 인한 핸드오프 등으로 설명된다. 이는 패킷 손실시 복구하는 메카니즘 또한 유선과 무선 구간에 다르게 적용되어야 함을 의미한다. 패킷 손실시 재전송 메카니즘을 유선과 무선 구간에 따로 적용하면서 네트워크의 성능 향상을 높인 방법으로 BS(Base Station : 기지국)내에 Snoop module의 추가를 들 수 있다[2].

그러나 무선구간의 MN가 BS 중심으로 하여 전송범위를 벗어날 경우 패킷 손실과 그로 인한 BS에서 재전송은 무선 구간내 트래픽을 증가시키면서 성능 저하의 원인이 되기도 한다. 따라서 이러한 패킷 손실과 재전송을 MN의 RS(Receive Signal:수신신호) flag를 이용하여 최소화시키면서 네트워크의 성능을 향상시키는데 연구의 목적을 둔다.

2. 관련 연구

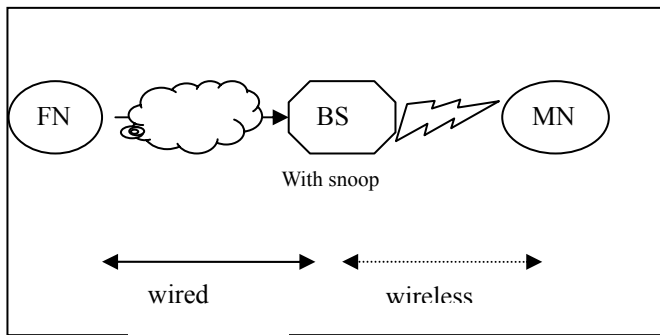
2.1 네트워크의 성능 향상을 위한 기존 연구

유무선 혼합 네트워크 망에서 네트워크 향상을 위한 TCP 관련 프로토콜을 크게 세가지 나누어 설명할 수 있다. End-to-End 프로토콜, split-connection 프로토콜, 링크 계층 프로토콜이며 본 논문에서 다룰 프로토콜은 split-connection 프로토콜로 Berkeley 대학에서 제안한 snoop protocol 이다.

2.1.1 Snoop

Snoop 은 기존의 유선망과 무선망 사이에 BS 을 두고 BS 의 link layer 에 별도의 snoop module 을 추가함으로써 무선 구간내의 높은 패킷 손실을 줄이기 위해 제안되었다. Snoop module 의 추가는 FN(Fixed Node: 고정 노드)와 MN 의 TCP 를 수정하지 않고 BS 의 라우팅 프로토콜의 수정만으로 성능을 향상시킨다[2].

아래 그림은 FN 과 MN 사이에서 Data 를 송수신하는 일련의 과정이다.



(그림 1) 유무선 통합 모델

BS 가 FN 로부터 수신한 데이터가 처음 수신한 패킷이라면 저장(caching)후 MN 로 전송하고 Snoop module 은 MN 에서 보내온 모든 ACK 에 대하여 기억하고 있다가 ACK 가 오면 버퍼에 저장되어있던 패킷을 제거하고 타임아웃이 발생하면 FN 과 상관없이 패킷 손실로 간주하여 이미 BS 에 cache 해 둔 패킷을 MN 로 재전송한다. 지역 FN 에서의 재전송이 아닌 BS 에서의 지역 재전송(local retransmit)을 하게 되는 것이다. 또한 BS 은 MN 에서 보낸 중복된 ACK 를 수신하면 FN 로 보내지 않음으로써 FN 에서의 불필요한 혼잡 메커니즘 호출을 방지 할 수 있다. 이처럼 Snoop 은 유선과 무선구간이 독립적 재전송을 실행하기 때문에 기존의 TCP 가 패킷 손실을 무조건 혼잡이라고 간주하여 전송속도를 낮추는 불필요한 혼잡 제어 실행을 방지한다. 그러나 무선 구간 내에서의 패킷 전송 및 재전송은 MN 노드가 전송범위 밖으로 벗어난 경우에도 이루어질 수 있다. 예를 들어 MN 가 수신범위 바로 경계선상에서 20m/s 의 이동 속도를 갖고 다른 BS 을 향해 이동한다고 하자. 이때 MN 가 ACK 를 기존 BS 에 보낼경우 FN 에서의

패킷 전송과 관계없이 BS 에서는 MN 로부터 ACK 을 받음으로써 다시 새로운 패킷 전송이 이루어진다. 따라서 BS 에서 출발한 패킷은 MN 가 이미 수신범위를 벗어난 상태이기 때문에 패킷 손실로 이어지고 이는 다시 재전송을 유발하면서 불필요한 트래픽을 조장하여 다른 노드에 영향을 미침으로써 성능저하의 원인이 될 수 있다.

2.2 제안하는 알고리즘

본 논문에서는 BS 과 MN 사이에서 패킷 전송시 수신신호를 이용하여 패킷 전송 여부를 결정하는 알고리즘을 제안한다.

수신신호는 MN 가 BS 로 부터 받는 signal 을 의미하는 것으로 BS 으로부터 멀리 떨어져 있으면 수신신호의 강도는 약해질 것이며 가까이 있으면 신호의 강도는 쉘 것이다. 이는 수신 신호가 BS 의 거리와의 관계로 설명됨을 의미한다. 수신신호는 송수신 안테나의 높이와 파장 그리고 거리와의 관계로 설명된다. 따라서 안테나의 높이가 수시로 변하는 것이 아닌 이상 실제 수신신호 변화에 영향을 주는 것은 거리라고 볼 수 있다[1]. 데이터의 전송 가능한 수신신호의 임계치(Threshold)를 $3.652e-10$ 라고 할 경우 MN 가 임계치 이하의 수신신호를 갖는다면 통신은 두절되는 것이다. 그러나 실제로 수신신호가 임계치 이하로 되기전에도 데이터 전송이 실패하는 경우가 있다. MN 가 이동으로 인해 BS 으로부터 점점 멀어지면서 데이터 송수신이 이루어지는 경우이다. 수신신호의 세기는 점점 약해질 것이며 disconnect 바로 직전에 날린 ACK 을 받은 BS 는 새로운 패킷을 MN 로 전송하게 될 것이다. 그러나 이미 MN 노드는 전송범위를 벗어난 상태이다. 전송된 패킷은 손실되고 지역 재전송이라는 BS(snoop)의 특징으로 재전송이 계속될 것이다. 무선 네트워크 상에 불필요한 트래픽만 조장하게 되는 것이다.

따라서 본 논문에서 제안하는 알고리즘은 MN 가 전송하는 ACK 에 수신신호 관련 flag bit 추가를 통해 BS 에서 MN 으로의 패킷 전송 유무를 결정하게 하는 것이다. 추가되는 flag bit 를 RS(Receive Signal) flag 로 표기한다. MN 가 BS 으로부터의 수신범위에서 벗어나고 있는지의 여부를 수신신호를 통해 예측하고 예측 여부를 ACK 의 RS flag 에 표기한다.

다음은 본 논문에서 제안하는 방법이다.

- 1) MN 가 1m/s 의 이동속도로 수신 범위 밖을 벗어날 때 MN 의 데이터 전송 가능한 수신신호의 임계치 값은 $3.652e-10$
- 2) 수신신호를 통해 ACK 에 RS flag 값을 셋팅
 - 이전 수신신호와 현재 수신신호 비교
 - 특정 범위내에서 수신신호가 확인되면 전송 범위를 벗어나기 바로 전으로 예측하여
 - 수신 신호가 이전수신 신호보다 크거나 같으

면 ACK의 RS flag를 '1' 셋팅해서 전송
 - 수신신호가 연속해서(count>=2) 이전신호보다 약하면 ACK의 RS flag를 '0' 셋팅해서 전송

다음은 BS에서의 추가된 부분이다.

- BS에서 수신한 ACK에 RS flag 값을 통해 다음 패킷의 전송 여부를 결정한다.
- '0' 일 경우 전송시 손실될 확률이 크므로 전송을 멈춘다.
- '1' 인 경우 계속해서 전송이 이루어진다.

본 논문에서는 네트워크의 성능 향상에 영향을 미칠 수 있는 수신신호의 특정범위를 알아 보려한다. 신호의 특정범위란 MN 수신신호의 세기가 점점 약해지면서 그 신호가 임계치값에 가까워지고 BS에서 보낸 다음 패킷이 손실될 것으로 예측되는 범위를 의미한다.

손실량은 줄이고 처리율은 기존 TCP Snoop 을 그대로 사용했을 경우와 같은 범위를 시뮬레이션을 통해 확인한다.

3. 시뮬레이션 모델

3.1 시뮬레이션 환경과 프로토콜

본 논문에서는 네트워크 시뮬레이터인 ns-2 를 이용하였으며, 유무선 혼합 환경으로 유선망과 무선망을 연결하는 BS에 snoop module 추가하였다

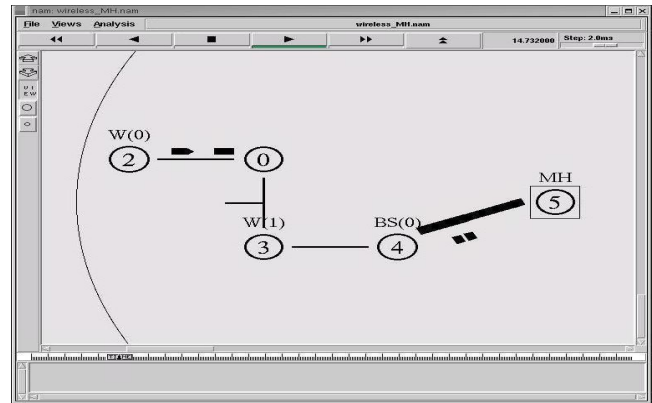
시뮬레이션 환경은 다음과 같은 가정하에 연구를 진행하였다.

첫째, TCP 타입은 TCP Reno 이다. 이는 TCP의 여러가지 구현 중 대표적이며 가장 많이 이용되는 것이 Reno이기 때문이다.

둘째, ACK에 대한 손실률을 0%로 하였으며, 데이터 송수신이 있는 동안 error가 없음을 가정한 것이다.

셋째, 데이터 송수신 방법은 반이중(half-duplex)으로 가정하였다.

넷째, 노드의 이동은 BS을 중심으로 전송범위 밖으로 벗어났다가 들어오는 경우로 가정하였다.



(그림 2) Topology

3.2 시나리오

트래픽 발생시 2 번이 소스(source) 노드가 되고 5 번은 목적(destination)지 노드가 된다.

노드 5 번은 이동성을 갖는 노드로서 본 시뮬레이션에서는 이동속도를 도보(1m/s)로 하는에서부터 차량(10m/s~25m/s)을 통한 이동까지 고려하여 테스트하였다. 시뮬레이션 시간은 50 초로 하였으며 10 초동안 정지된 상태에서 데이터 송수신이 이루어지다가 10 초 경과 후 MN가 수신 가능한 범위(250m) 밖으로 벗어났다가 다시 들어오는 경우로 하였다.

4. 시뮬레이션 결과

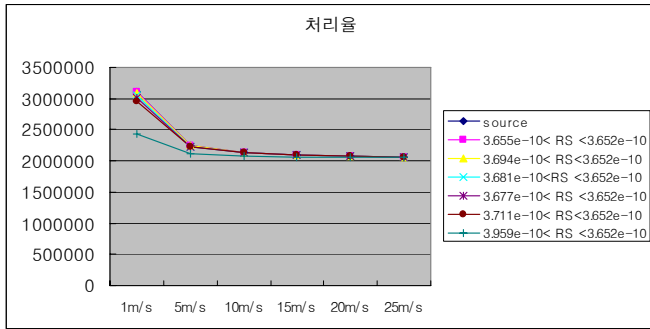
이동 속도가 빠를수록 수신노드에서의 처리율(Throughput)은 적어진다. 빠른 속도로 MN가 수신 범위 밖을 벗어나면 당연히 패킷 처리율은 적어진다. 그러나 단순히 처리율이 적어지는 것으로만 생각할 것은 아니다. MN가 임계치에 가까운 수신신호상에서 첫번째 데이터에 대한 ACK를 BS에 보내면 BS가 두번째로 보낸 데이터는 손실될 확률이 커지게 된다. 이미 데이터를 받을 시점에서 MN는 수신범위를 벗어났기 때문이다. 이는 불필요한 재전송을 유발하게 되고 이러한 재전송은 다른 노드가 데이터 전송을 위한 채널 확보를 어렵게 하는 원인이 될 수도 있다. 따라서 본 논문에서는 이동 속도별로 MN에서 패킷 처리율과 손실량을 체크해보고 본 논문에서 제안하는 기법을 적용 후 처리율과 손실량을 체크해 봄으로써 불필요한 재전송과 손실량을 줄이려고 한다.

MN에서 패킷 수신 가능한 거리는 최대 250m이며 이때의 수신 신호는 3.652e-10 이다. 따라서 패킷 수신이 가능하려면 3.652e-10 이상의 신호의 강도를 가져야 한다. 아래 그림(3)과 그림(4)는 현재의 수신신호를 확인하고 MN의 이동에 따라 다음 위치가 영역 밖으로 벗어날 것으로 예상되는 신호의 범위를 다양하게 해서 테스트한 결과이다. 성능평가를 위해 BS에서의 전송량과 MN에서 처리율, 그리고 패킷 손실량 통해 확인해 보았다.

그래프에서 source란 기존의 TCP와 Snoop를 적용한 경우이다.

packet size		1460 bytes	
Traffic type		FTP	
TCP Type		TCP Reno	
Pysical Layer	Propagation Model (Reference distance)	Free space	1/r^2(100m)
		Two-ray ground	1/r^4(250m)
	802.11's		
	Direct Sequence Spread Spectrum(DSSS)		
802.11 base on CSMA/CA			
Topology, Mobility Model		670 * 670 grid	
		1m/s(도보)~ 25m/s(자동차)	

<표 1> wireless-시뮬레이션 모델



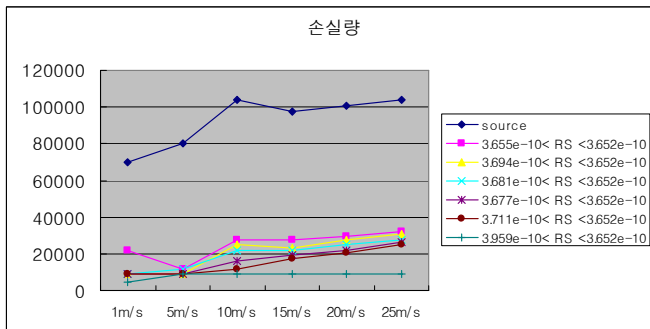
(그림 3) MN에서의 패킷 처리율

그림(3)은 MN가 다음 패킷을 보내지 말라고 해야 할 수신신호의 범위를 여러 개의 구간으로 나누어서 테스트 한 결과이다. 가장 하단의 선은 약한 신호로 해서 다음 데이터의 수신가능성이 희박할 것으로 예상된 범위를

$$3.959e-10 < RS < 3.652e-10$$

으로 했을 때이다. 다른 그래프에 비해 그 범위를 지나치게 넓게 좁으로써 실제로 데이터를 수신할 가능성이 많음에도 불구하고 데이터를 보내지 말라고 하면서 MN에서 패킷 처리율이 기존 TCP(with snoop) 코드를 수정하기 전보다 떨어지는 것을 볼 수 있다.

그래프들 중 가장 위에 나타나는 그래프는 알고리즘 적용전이고 나머지는 신호의 범위를 점점 좁히면서 테스트한 그래프이다. 처리율은 이동 속도가 빠를수록 논문에서 제안한 기법 적용 전과 크게 다르지 않다.



(그림 4) MN에서의 패킷 손실량

그림(4)의 결과는 패킷 손실량을 나타낸 것으로 기존 알고리즘을 그대로 사용했을 때의 그래프가 손실량이 가장 많은 것으로 나타난다.(가장 위) 반면에 알고리즘을 적용했을 때의 그래프는 하단에 보이는 것으로 손실량(재전송포함)이 현저하게 줄어든 것으로 볼 수 있다.

기존 TCP/IP(with snoop)를 그대로 적용했을 때와 수정된 알고리즘을 적용했을 때 MN에서의 처리량은 같고 패킷 손실량이 가장 적은 범위는 $3.694e-10 < Pr < 3.652e-10$ 으로 확인할 수 있다.

실제 테스트는 좀더 범위를 좁게 해서 ($3.655e-10 < Pr < 3.652e-10$)도 진행해 보았으나 이동속도가 아주 느릴 경우(1m/s)를 제외하고는 손실량을 많이 줄이지 못하는 것으로 확인 되었다.

4. 결론

본 논문에서는 하나의 MN에서 데이터 수신량과 손실량등을 MN의 이동속도별로 테스트해 보았다. 시뮬레이션 결과 이동속도가 빠른 수록 데이터 손실량이 많아지는 것을 확인할 수 있었고 이런 손실량을 줄이기 위해 개선된 알고리즘을 제안하였다.

MN의 이동속도가 1m/s인 경우를 제외하고 이동속도가 빠른 경우 처리율은 같고 손실량이 현저하게 줄어드는 것을 확인하였다. MN에서의 패킷 손실은 BS에서 재전송을 유발하게 되고 불필요한 트래픽 조장으로 전체적인 네트워크 성능 저하의 원인이 되기도 한다. 따라서 패킷 손실을 줄인다는 것은 네트워크 성능 향상으로 설명 될 수 있다.

참고문헌

- [1] William Stallings, "Wireless Communications & Networks", Prentice Hall, 2004.
- [2] Yun-Joo Kim, meejeong Lee and Jae-Young Ahn, "Limited Indirect Acknowledgement for TCP Performance Enhancement over Wireless Networks", 정보과학회 논문지, 제 30권 제 2호, 2003. 4
- [3] HALA ELAARAG, "Improving TCP Performance over Mobile Networks", ACM Computing Surveys, Vol. 34, No.3, September 2002, pp. 357-374
- [4] V.Anantharaman, Sivakumar, "Atra : A Framework for Improving TCP Performance over Ad-hoc Networks, Technical Report, GNAN Research Group - <http://www.ece.gatech.edu/research/GANA,> march 2002
- [5] Ajay Kr. Singh and Sridhar Iyer, "ATCP: Improving TCP Performance over Mobile Wireless Environments," In Fourth IEEE Conference on Mobile and Wireless Communications Networks, Stockholm, Sweden, September 2002.