

무선 에지의 링크 오류율을 이용한 개선된 TCP 혼잡제어

(An Enhanced TCP Congestion Control using Link-Error Rates at Wireless Edges)

오 준 석 [†] 박 탄 세 [†]
(Jun-Seok Oh) (Tan-Se Park)

박 창 윤 ^{**} 정 충 일 ^{***}
(Chang-Yun Park) (Choong-Il Jung)

요약 본 연구에서는 대부분의 무선 링크는 네트워크 에지에서 사용되고, NIC 드라이버가 무선 링크 오류율을 관찰하고 있다는 가정 하에, 기존 TCP의 혼잡제어를 수정한 TCP-L(TCP Link Aware)을 제안한다. TCP-L은 무선 링크 오류율을 이용해 유선 링크의 혼잡 손실률을 예측하여 무선 링크에서 세그먼트 손실이 발생했을 경우에 나타나는 성능 저하를 회피할 수 있다. 또한 실험을 통해 무선 링크 오류율이 높은 환경에서 다른 무선 TCP 보다 높은 처리율을 보이면서도 다른 유선 TCP 소스와의 형평성을 유지하는 것을 검증하였다. 본 연구의 접근 방법은 세그먼트 손실이 많거나 전송 지연이 산발적으로 발생하는 새로운 무선 환경, 즉 802.11 ACK-less와 인지 라디오(Cognitive Radio)와 같은 무선 링크에 효과적으로 적용될 수 있을 것으로 예상된다.

키워드 : 무선 TCP, 무선 네트워크, 혼잡제어

· 이 논문은 제36회 추계학술발표회에서 '무선 에지의 링크 오류율을 이용한 개선된 TCP 혼잡제어'의 제목으로 발표된 논문을 확장한 것임

· 이 논문은 2008년도 중앙대학교 학술연구비 지원에 의한 것임

[†] 학생회원 : 중앙대학교 컴퓨터공학부
ohjunseok@cmlab.cse.cau.ac.kr
parktanse@cmlab.cse.cau.ac.kr

^{**} 종신회원 : 중앙대학교 컴퓨터공학부 교수
cypark@cau.ac.kr
(Corresponding author임)

^{***} 정 회 원 : 여주대학 인터넷정보처리학과 교수
cijung@yit.ac.kr

논문접수 : 2009년 12월 24일

심사완료 : 2010년 3월 29일

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제16권 제7호(2010.7)

Abstract Assuming that a wireless link is mostly used at the network edge and the wireless NIC driver keeps monitoring the error rate of its link, this paper proposes an enhanced TCP congestion control, TCP-L (TCP Link-Aware). TCP-L predicts true congestion losses occurred inside the wired link area by utilizing the wireless link error rate. As a result, it mitigates performance degradation caused from TCP congestion control actions when segments losses occur in a wireless link. Experimental results show that TCP-L provides better performance and fairness in lossy wireless links than existing TCP congestion control schemes. Our approach utilizing the characteristic of the link at TCP could be well adapted to new wireless environments such as Cognitive Radio and ACK-less IEEE 802.11, where a frame may be delivered with a very long delay or lost in the link.

Key words : Wireless TCP, Wireless Networks, Congestion Control

1. 서론

전송 계층에서 많이 사용하는 프로토콜인 TCP는 무선 환경의 영향을 많이 받는다[1]. 초기 TCP는 유선 통신을 가정하여 만들어졌기 때문에, 무선 링크에서 오류가 발생할 때도 혼잡이 일어났다고 판단하여 전송 속도를 떨어뜨린다. 그 결과 통신 성능의 저하를 가져온다. 그래서 링크 에러가 발생하는 무선 환경에서 잘 적용하는 "Wireless TCP"의 설계를 위한 연구가 활발히 이루어지고 있다.

최근에는 새로운 오류 특성을 가진 무선 링크가 등장하였다. IEEE 802.11 ACK-less[2]는 802.11을 확장하여 전력 효율을 향상한다. 하지만 무선 링크 오류율에 직접적인 영향을 받는 특성이 있다. 인지 라디오(Cognitive Radio)[3]는 스펙트럼 자원을 효율적으로 사용하기 위해 채널 상황에 적응하는 기술이다. 스펙트럼을 센싱하거나 스펙트럼 핸드오프를 수행할 때 생기는 지연 시간은 전송 계층에서 전송 지연으로 인식하기 때문에 TCP의 성능에도 악영향을 미친다. 이러한 환경에서 성능을 향상시키기 위해서는 무선 환경에 적극적으로 적응하는 TCP가 필요하다.

그림 1은 본 연구가 가정하는 통신망 환경을 보여주고 있다. 무선 링크는 네트워크 에지, 즉 라스트 마일 링크에서만 사용된다. 그리고 무선 링크에서 발생한 프레임 오류는 대부분의 NIC 드라이버에서 관찰되고, 평균 프레임 오류율은 파악되고 있다. TCP가 NIC에서 파악된 정보를 알 수 있으면, 세그먼트가 무선 링크에서 손실 되었는지, 중단 링크 상에서 혼잡 때문에 손실되었는지 추측할 수 있다. 이것을 이용해서 기존 TCP 혼잡

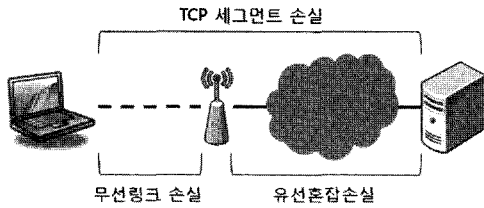


그림 1 무선 환경의 네트워크 에지의 손실

제어에서 세그먼트 손실이 발생하면 무조건 혼잡원도우를 줄이는 것을 개선할 수 있다.

이 논문에서는 TCP의 혼잡제어를 수정하여 무선 환경에서 적용하며 우수한 성능을 보이는 TCP-L(TCP Link Aware)을 제안한다. 이 방법은 TCP에서 발생한 전체 오류에서 무선 링크 오류의 영향을 분리하여 유선 링크의 혼잡 손실률을 추측한다. 이를 이용하여 혼잡에 의한 손실이 발생할 때만 혼잡제어를 하고 무선 링크 오류에 의한 손실이 발생했다고 예측되면 혼잡 제어를 하지 않아, 무선 환경에서도 좋은 성능을 낼 수 있다. 더불어 혼잡의 변별력도 좋기 때문에 전체 네트워크의 공정성을 해치지 않는다.

NS-2[4]로 시뮬레이션 한 결과 기존의 TCP들에 비해서 좋은 성능을 내면서 공정성을 해치지 않는 것으로 나타났다. TCP-L을 사용하여 무선 링크 오류율이 높은 환경 또는, ACK-less를 사용하는 무선 환경에서 성능의 효율을 향상할 수 있고, 인지 라디오 네트워크에서 오톱작과 성능 감소를 막을 수 있는 가능성을 보여주었다.

본 논문은 2장에서 무선 환경에서 TCP의 성능을 높이는 기존 기법들을 소개하고, 3장에서 제안하는 TCP-L 프로토콜의 개념과 핵심 알고리즘을 다룬다. 4장에는 무선 환경에서 기존 TCP 프로토콜과 성능 비교를 하고, 5장에서는 IEEE 802.11 ACK-less 환경에서의 실험결과를 보여준다. 그리고 6장에서 결론과 향후 연구 방향을 제시한다.

2. 관련 연구

무선 환경에서 발생하는 TCP의 성능 저하 문제는 여러 방향에서 연구되어왔다. 지금까지의 연구들을 분류하여 보면 종단(End-to-End), 링크 계층, 연결 분리(Split-Connection) 기법으로 나눌 수 있다[1].

종단 방식으로는 TCP Reno, New Reno, Vegas[5] 등의 방식이 있지만, 대부분 무선 환경에서 좋은 성능을 보이지 않는다. 무선에 잘 적용하는 기술로 TCP Westwood[6]가 있는데, RTT를 바탕으로 종단 대역폭을 추정하여 적당한 혼잡 윈도우 크기를 유지한다. 무선 링크 손실이 극심하지만 않다면, 우수한 성능을 유지한다.

링크 계층 방식으로는 재전송을 통해 신뢰성을 보장

하는 링크 계층 프로토콜이 있다. 하지만 링크 레벨의 재전송은 전송 딜레이를 증가시키기 때문에, TCP의 타임아웃에 영향을 미쳐서 성능의 저하를 일으킬 수 있다. 링크 계층 방식에 속하는 기법으로 Snoop[7]도 있다. 베이스 스테이션에 내장된 Snoop 에이전트가 버퍼링과 재전송을 통해 무선 TCP의 성능을 향상시키는 방식이다. 무선 호스트가 송신자가 아니면 TCP 프로토콜의 수정이 필요 없다는 장점이 있지만, 네트워크 프로토콜 계층을 침범하는 문제점이 있다.

연결분리 방식은 유·무선 연결을 분리하여 해결책을 모색하는 방식이다. 대표적인 방식은 TCP-II[8]가 있다. 이는 각 링크의 특성에 맞게 연결의 트래픽 인자를 정할 수 있고 이를 통해 무선 링크의 불안정성에 따른 문제를 무선 링크 연결에 국한시켜 성능을 향상시킨다. 하지만 종단의 의미구조(Semantic)를 파괴하게 된다.

본 연구에서는 제안하는 TCP-L은 종단 방식을 따른다. 종단 방식은 별도의 외부 구성요소 없이 TCP의 수정만으로 알고리즘을 적용할 수 있다. 또한 링크 계층의 정보에 접근하여 링크 오류율 정보를 직접 사용한다. 이것은 다른 계층으로의 침범이지만, 동작 과정의 간섭이 아니라 단순한 정보 참조이기 때문에 계층 침해 수준이 낮다.

3. TCP-L(TCP Link Aware)

3.1 무선 링크 오류 인지

802.11의 NIC 드라이버는 네트워크 상황을 모니터링하여 그 정보에 대한 통계를 유지하고 있다. 링크 계층에서 프레임이 송신할 때, 최대 재전송 한계를 넘어서면 송신을 포기하며 송신 실패 횟수가 1씩 증가한다. 송신의 실패는 ACK 회신 실패이므로 무선 링크에서 오류가 발생했다고 볼 수 있다. 그리고 전체 송신 프레임 수에 대한 통계도 유지하고 있다. 송신 실패 횟수를 전체 프레임 수로 나누어 근사적인 무선 오류율을 구할 수 있다.

리눅스의 커널 네트워크 장치 드라이버는 링크 계층의 통신 상태에 대한 정보를 가지고 있는 가상 파일인 `/proc/net/dev`에서 알아낼 수 있다. 또한 대부분의 OS가 제공하는 SNMP API를 사용해 MIB의 정보를 얻어올 수 있고, 최근 많이 연구되는 Cross-Layer 기법을 사용할 수도 있다.

3.2 TCP 혼잡 손실률 예측

혼잡 손실률은 TCP에서 발생한 세그먼트 오류율과 링크 오류율의 차를 구해 근사적으로 구할 수 있다. TCP에서 송신한 전체 세그먼트 개수, 오류로 인해 재전송 한 세그먼트 개수를 알고 있으면 TCP 계층에서 발생한 세그먼트 오류율을 계산할 수 있다. 또한 링크 계층에서 발생한 전체 송신 프레임 개수, 전송 실패한 세그먼트 개수를 알고 있으면 MAC 계층에서 발생한

프레임 오류율 역시 계산 가능하다. 두 오류율의 차는 무선 링크를 제외한 종단간에 발생한 오류율, 즉 혼잡 손실률을 뜻한다. 또한 혼잡 손실률과 TCP 세그먼트 오류율의 비는 전체 세그먼트 손실에서 혼잡 손실이 차지하는 비를 나타낸다. 최종적으로 구한 이 비율은 TCP에서 혼잡제어 수행 여부의 척도가 된다.

$$Loss_{tcp} = \frac{retx_segment_count}{tx_segment_count}$$

$$Loss_{link} = \frac{failed_count}{tx_frame_count}$$

$$Loss_{congestion} = Loss_{tcp} - Loss_{link}$$

$$P_{congestion} = \frac{Loss_{congestion}}{Loss_{tcp}}$$

일반적으로 혼잡제어를 통해 윈도우의 크기를 줄이는 동작은 세그먼트 손실을 감지하여 재전송을 할 때 일어난다. 표 1은 기존 TCP에 혼잡제어 동작 여부를 검사하기 위해 TCP Reno의 혼잡제어를 수정한 의사코드다. TCP Reno의 경우 빠른 재전송 또는, 타임아웃에 의해 혼잡제어가 동작한다. 무선 오류 검출의 목적은 경우에 따라 혼잡제어를 하지 않는 것이므로, 혼잡제어를 수행할 확률인 $P_{congestion}$ 를 계산하여 혼잡제어 여부를 결정한다.

표 1 개선된 TCP Reno 버전 혼잡제어의 의사코드

```

if ( n DUPACK received )
    calculate P_congestion
    X = a random number between 0~1
    if ( P_congestion>X )
        ssthresh = cwin/2
        cwin = ssthresh
    endif
endif

if ( timeout expires )
    calculate P_congestion
    X = a random number between 0~1
    if ( P_congestion>X )
        cwin = 1;
        ssthresh /= 2;
    endif
endif
    
```

3.3 서버 TCP에의 적용

혼잡제어는 TCP 송신단에서 수행하기 때문에, TCP 세그먼트 오류율과 무선 링크 오류율을 바탕으로 혼잡제어를 하는 방식 역시 송신단에서 구현해야 한다. 무선 노드는 클라이언트일 경우가 많기 때문에 클라이언트가 무선 링크의 상황을 서버에게 피드백 하는 방법이 필요하다.

이를 위해 TCP 클라이언트는 클라이언트의 무선 링크 오류율 $Loss_{link}$ 를 서버 TCP에게 송신한다. $Loss_{link}$ 의 전송을 위해서 확장한 TCP 헤더를 사용한다. TCP

헤더는 최대 40바이트까지 확장이 가능하므로, 32비트 실수와 함께 기존의 확장 옵션을 사용할 수 있다. 링크 오류율은 급격히 변화하지 않기 때문에 일정한 주기로 피드백 한다.

4. 실험 결과 및 분석

혼잡제어의 중요한 두 가지 성능 척도는 처리율과 공정성이다. 이 두 가지 요소를 검증하기 위하여 그림 2와 같은 토폴로지를 구성한다. 이 환경에서 TCP1과 TCP2는 100초 동안 FTP를 통해 대용량 데이터를 송신한다. 또한 무선 링크 에러의 영향을 관찰하기 위해 무선 바이트 에러율을 0에서 5×10^{-4} 까지 변화시키며 실험한다. 실험의 정확성과 효율성을 위해 NS-2를 사용하며, TCP 프로토콜은 모두 TCP-Linux[9]를 사용한다.

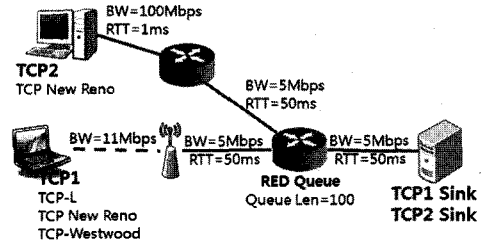


그림 2 실험 토폴로지

그림 3은 TCP1에서 세 가지의 TCP 프로토콜에 대한 유효 처리율(Goodput)을 보여준다. TCP-L은 혼잡 손실과 무선 링크 손실을 잘 구분하여 오류에 강한 성능을 나타낸다. 높은 오류율에서는 혼잡의 영향이 줄어들면서 포화 처리율과 차이가 줄어드는 모습을 보인다. TCP NewReno는 1.5×10^{-4} 부터 TCP 재전송의 오버헤드와 재전송한 세그먼트 손실로 인한 타임아웃으로

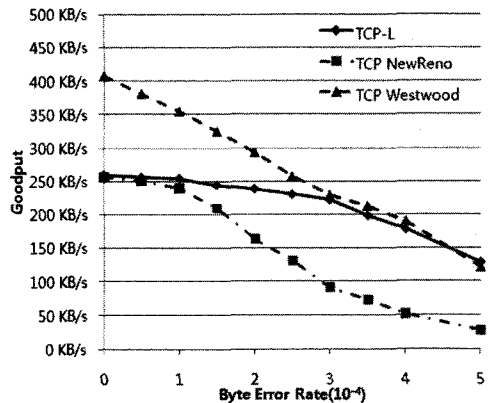


그림 3 TCP1의 유효 처리율

인해 유효 처리율의 저하를 보인다. 한편 TCP Westwood는 낮은 오류율에서 높은 처리율을 보이고 있지만, TCP2와의 공정성을 깨뜨린다.

그림 4는 TCP2에 대한 TCP1의 공정성을 나타낸다. 공정성은 TCP1의 유효 처리율을 TCP2의 유효 처리율로 나누어 구한다. TCP1과 TCP2가 공정하게 동작한다면 병목 링크의 가용 자원을 균등하게 사용하여, TCP1과 TCP2의 유효 처리율의 비는 1이 된다. TCP-L과 TCP New-Reno는 오류율이 없을 때부터 TCP2와의 공정성을 잘 지키며, 오류가 증가하고 성능이 저하되면서는 공정성이 떨어진다. 반면 TCP Westwood는 무선 오류가 없는 환경에서 굉장히 높은 성능을 보였지만, ACK 압축 현상으로 인해 TCP2와의 공정성을 깨뜨린다. 3×10^{-4} 부터 TCP2와 균형을 맞추며 TCP-L과 비슷한 추세를 보이지만, 공정성을 지키기 때문이 아니라 무선 오류의 영향으로 생기는 현상이다.

TCP-L은 전체 오류율에서 무선링크의 영향을 분리하면서 혼잡 윈도우를 선별적으로 줄이므로, 높은 오류

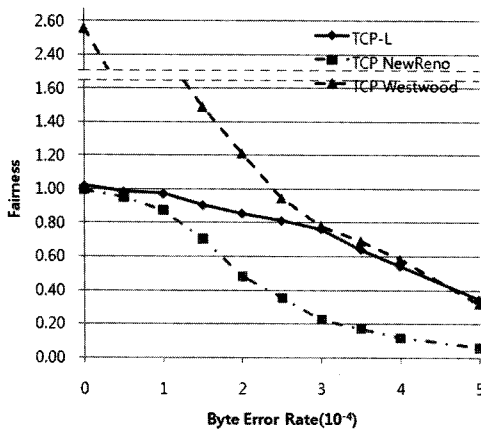


그림 4 TCP2에 대한 TCP1의 공정성(TCP1÷TCP2)

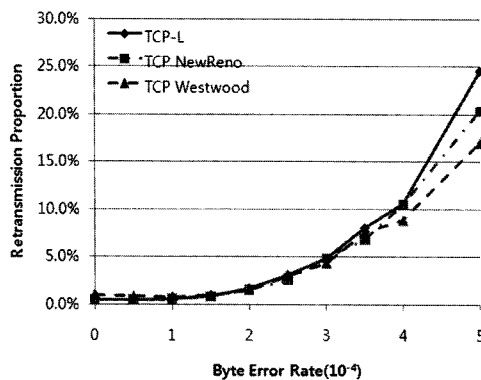


그림 5 TCP1의 재전송률

율에서는 다른 방식들에 비해 더 많은 세그먼트를 전송할 수 있다. 그림 5는 TCP1에서 전체 전송 세그먼트에 비해 재전송한 세그먼트가 차지하는 비율을 나타낸다. 실험 결과에 따르면 재전송 세그먼트의 비율은 다른 프로토콜과 유사하고, 무선 링크 에러율이 아주 높은 경우에서도 큰 차이를 보이지 않는다. 이를 통해 높은 오류율에서도 세그먼트 재전송으로 인한 효율성 저하가 적다는 것을 알 수 있다.

5. IEEE 802.11 ACK-less 유니캐스팅에서 TCP-L 적용

802.11 ACK-less 유니캐스팅은 MAC의 ACK를 제거하여, 현재의 무선 LAN 사용환경과 같은 링크 상태가 양호한 환경에서 통신 성능과 사용전력 효율을 향상시키기 위해 제안된 기법이다. 하지만 무선 링크 손실을 링크 계층에서 복구하지 않고, TCP의 재전송에 의존하기 때문에 오류율이 높은 장거리 통신에서는 혼잡 윈도우의 감소로 인한 처리를 저하가 문제가 된다. 한편 TCP-L은 무선 링크 계층에서 발생한 손실을 인지할 수 있기 때문에 이런 환경에서도 좋은 성능을 유지할 수 있다.

그림 6은 TCP-L이 ACK-less 환경에 어떻게 적용하는지 알기 위한 실험의 토폴로지이다. 표 2는 실험을 통해 측정된 유효 처리율과 전력 효율이다. 1578 바이트의 프레임용 TCP-L과 ACK-less를 사용하여 전송하는 경우 TCP-L의 유효 처리율은 265KB/s로써, ACKed보다 350배 높은 세그먼트 오류 환경에서 무려 72%의 성능을 유지한다. 한편 전송하는 프레임 크기가 작으면 무선 링크의 손실률이 낮기 때문에[10], 전송 성능의 유지에 유리하다. 138바이트의 프레임으로 실험한 결과, TCP-L과 ACK-less는 61KB/s의 성능으로 ACKed의 56KB/s보다 9% 가량 우수한 성능을 낸다. 또한 전력효율은 어

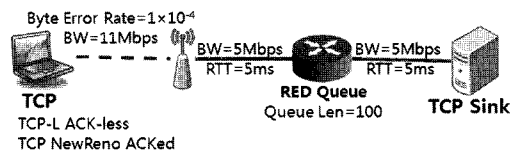


그림 6 ACK-less 실험 토폴로지

표 2 무선 링크 오류율 1×10^{-4} 에서 TCP들의 성능

	138 Bytes		1578 Bytes	
	유효 처리율	전력 효율	유효 처리율	전력 효율
TCP-L, ACK-less	61KB/s	299KB/J	265KB/s	1224KB/J
TCP NewReno, ACKed	56KB/s	148KB/J	365KB/s	760KB/J

는 경우에서든 TCP-L과 ACK-less가 항상 우수하며, 최대 102% 우수한 성능을 보인다. 이와 같이 TCP-L은 MAC 레벨에서 ACK를 제거해 전력효율을 향상시키면서 우수한 성능을 낼 수 있다.

6. 결론

무선 환경에서의 성능 저하는 TCP의 약점으로 꼽혀 왔다. 하지만 혼잡제어를 개선한 TCP-L은 무선 링크 오류 인지와 TCP 혼잡 손실을 예측을 통해 무선 환경에 잘 적용할 수 있다. 특히 오류가 심해질수록 다른 TCP 혼잡제어에 비해 우수한 처리율을 보이면서도 네트워크의 공정성을 해치지 않는다.

TCP-L은 IEEE 802.11 ACK-less 유니캐스팅에도 잘 적응하는 모습을 보인다. ACK-less와 인지 라디오를 비롯하여 속속 등장하는 새로운 환경의 네트워크에서는 그에 알맞은 TCP가 필요하다. TCP-L은 이런 새로운 환경에 적용할 수 있는 가능성이 있으며, 앞으로 추가적인 연구를 통해 개선 및 검증을 해야 한다.

참 고 문 헌

- [1] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan, and Y. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," *IEEE/ACM Transactions on Networking*, vol.5, issue 6, pp.756-769, 1997.
- [2] Jun-Seok Oh, Hyun Yang, and Chang-Yun Park, "Performance and Energy Saving of IEEE 802.11 ACK-less Unicasting using MAC Level Multicasting," *Proc. of the KIISE Korea Computer Congress 2008*, vol.35, no.1(D), pp.406-411, 2008. (in Korean)
- [3] Ian F. Akyildiza, Won-Yeol Lee, Mehmet C. Vuran, and Shantidev Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol.50, issue 13, pp.2127-2159, 2006.
- [4] The Network Simulator - ns-2, <http://www.isi.edu/nsnam/ns/>
- [5] L.S. Brakmo, and L.L. Peterson, "TCP Vegas: end to end congestion avoidance on a global Internet," *IEEE Journal on selected Areas in communications*, vol.13, no.8, pp.1465-1480, 1995.
- [6] Claudio Casetti, Mario Gerla, Saverio Mascolo, M.Y. Sansadidi, and Ren Wang, "TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks," *Wireless Networks*, vol.8, issue 5, pp.467-479, 2002.
- [7] Hari Balakrishnan, Srinivasan Seshan, Elan Amir, and Randy H. Katz, "Improving TCP/IP performance over wireless networks," *Proc. of the 1st annual international conference on Mobile compu-*

ting and networking, pp.2-11, 1995.

- [8] A. Bakre, and B.R. Badrinath, "I-TCP: indirect TCP for mobile hosts," *15th IEEE International Conference on Distributed Computing Systems (ICDCS'95)*, pp.136-143, 1995.
- [9] D. X. Wei, and P. Cao, "NS-2 TCP-Linux: An NS-2 TCP Implementation with Congestion Control Algorithms from Linux," *Proc. of the 2006 workshop on ns-2: the IP network simulator*, no.9, 2006.
- [10] Dan Duchamp and Neil F. Reynolds, "Measured Performance of a Wireless LAN," *Proc. of the 17th IEEE Conference on Local Computer Networks*, pp.494-499, 1992.