

논문 2009-46TC-2-13

# 802.11 무선 네트워크에서 TCP 성능 향상을 위한 기법

( A Mechanism to improve the TCP performance in 802.11 Wireless Networks )

장복전\*, 김준환\*, 박용진\*\*

( Fu-Quan Zhang, Jun-Hwan Kim, and Yong-Jin Park )

## 요약

최근 802.11 무선 네트워크에서 TCP성능 향상을 위한 많은 연구가 이루어지고 있다. End-to-end 연결에서 TCP성능을 결정하는 가장 큰 요인으로는 hop-count와 RTT (Round Trip Time) 이다. 본 논문에서는 TCP의 성능을 향상하기 위해, hop-count와 RTT 변경 시 적절한 CWND값을 설정하여 TCP-CEV을 변경하는 기법을 설명한다. 본 논문에서 제안한 기법은 기본 TCP 구조를 따르는 대부분의 transport 프로토콜에 적합하고, 분석적인 과정을 통해 결과를 도출한다. 시뮬레이션을 통해 제안한 기법이 체인 토폴로지에서는 12%, 격자 토폴로지에서는 4.9% 이상 성능이 향상되었으며, TCP window가 특정 값에 수렴하는 것을 보인다.

## Abstract

Improving TCP performance has long been the focus of many research efforts in 802.11 wireless networks study. Hop count and Round Trip Time (RTT) are the critical sources which serious affect the TCP performance on end to end connection. In this paper, we analytical derived the affection and based on the analysis we propose TCP should Change its Expected Value (TCP-CEV) when hop count and RTT change by setting a reasonable CWND change rate to improve the performance. The proposed scheme is applicable to a wide range of transport protocols using the basic TCP mechanism, and the protocol behavior is analytically tractable. We show that our simple strategy improves TCP performance at least over 12% in a chain topology, 4.9% in a grid topology and improve the TCP convergence.

**Keywords :** 802.11 wireless networks, TCP, RTT, hop count, throughput

## I. 서론

무선 네트워크가 네트워크 분야의 주요 연구로 발전함에 따라 TCP를 사용하는 무선 네트워크에서 일어나는 현상에 주목하고 있고, 무선네트워크에서 TCP 사용이 적합한지에 대한 연구가 이루어지고 있다. TCP가 유선 네트워크에서 원활하게 동작하는 반면에 무선 네트워크에서 성능을 내기에는 적합하지 않다는 연구결과

가 많았다<sup>[1~5]</sup>. 메시 네트워크, 애드혹(ad-hoc) 네트워크, 센서 네트워크와 같은 무선 멀티홉(multi-hop) 링크 경로를 갖는 무선네트워크에서 기존의 TCP는 네트워크 혼잡, 라우팅 실패, 링크 고장 등을 구별할 방법이 없었다. TCP는 패킷손실의 주된 이유인 네트워크 혼잡을 고려하여, 혼잡제어 과정을 시작하고 slow start 주기로 진입할 것이다<sup>[3]</sup>. 이런 이유로, 무선 네트워크에서 TCP 사용을 위해 TCP를 수정하려는 많은 시도가 있었고, 이와 관련된 프로토콜이 제안되었다.

애드혹(ad-hoc) 무선 네트워크에서 몇 가지 transport 프로토콜은 동적 라우팅이 transport 기법에 어떤 영향을 주는지 고려하여 제작되었다. 그래서 다른 계층의 경로 상태정보 및 무선 채널 오류, 라우팅 변화, 연결실패, 무선 네트워크 이벤트를 이용하고자하는 시

\* 학생회원, \*\* 정회원, 한양대학교 전자컴퓨터통신공학부 (Dept. of Electronics and Communications, Hanyang University)

※ 이 논문은 2008년도 두뇌한국21사업에 의하여 지원되었음.

접수일자: 2008년11월7일, 수정완료일: 2009년2월12일

도가 있었다.

에드혹(ad-hoc) 무선 네트워크에서 TCP-Vegas, TCP-Westwood, TCP-Jersey<sup>[6~8]</sup>는 링크 경로의 가능 대역폭을 판단하는 데에 중점을 두고 있다.

TCP-Eifel<sup>[9]</sup>은 재전송의 모호함 문제를 해결하기 위해 TCP의 timestamp option을 이용하였다. TCP Bus<sup>[10]</sup>은 버퍼링 능력을 고려하였고, TCP-ELFN<sup>[11]</sup>은 무선 환경에서 혼잡과 라우팅 변화를 구별하기 위해 Explicit Link Failure Notification (ELFN)을 제안하였다. ATCP와 ADTCP<sup>[12~13]</sup>는 다른 네트워크 이벤트로부터 혼잡을 구별하기 위해 end-to-end 네트워크의 상태 변화를 감시한다. 위에서 언급한 프로토콜들은 TCP 상태를 멈추고, 라우팅 변화에 의한 전송을 감소 없이 큰 값의 Congestion Window를 유지한다는 것을 기본 개념으로 하고 있다.

그밖에 다른 방법들도 있다. Ad hoc Transport Protocol (ATP)<sup>[14]</sup>은 윈도우 기반이 아닌 rate 기반의 프로토콜이다. Split TCP<sup>[15]</sup>는 짧은 TCP 연결이 많은 곳에 프록시를 설정하는 프로토콜이다.

위에서 언급한 기법들과 같이 무선 네트워크에서 TCP의 효율성을 높이기 위한 많은 연구가 있었지만, TCP는 특정 metric에 한해서만 성능이 좋다.

본 논문에서는 end-to-end 연결에서 TCP성능을 결정하는 가장 큰 요인인 hop-count와 RTT(Round Trip Time)를 분석한다. 분석을 통해 hop-count와 RTT가 변경 시 적절한 CWND 변화율을 적용함으로써 성능을 향상 시키고, 높은 처리량을 기대할 수 있다. 본 논문에서 제안한 기법은 기본 TCP 구조를 따르는 대부분의 transport 프로토콜에 적합하고, 분석적인 과정을 통해 결과를 도출한다. 시뮬레이션을 통해 제안한 기법이 체인 토폴로지에서 12%, 격자 토폴로지에서 4.9% 이상 성능이 향상되었으며, TCP window가 특정 값에 수렴하는 것을 보인다.

본 논문의 차례는 다음과 같다. II장에서는 배경과 관련연구에 대해 설명한다. III장 1절에서는 가정과 손실 이벤트율의 효과에 대해 설명한다. III장 2절에서는 TCP window 기법의 효과에 대해 설명한다. III장 3절에서는 제안한 기법에 대해서 설명한다. IV장에서 네트워크 시뮬레이션 환경을 설정하고, 시뮬레이션 결과를 보인다. 마지막으로 V장에서 결론과 미래연구에 대해 언급한다.

## II. 배경 및 관련연구

무선 네트워크에서 TCP window 기법은 패킷이 밀집되는 지역에서 신호 간섭과 충돌이 일어날 확률이 더 많다<sup>[3, 16]</sup>.

TCP 흐름에서 전송 간섭은 전송경로상의 TCP 데이터의 자기간섭, 전송의 양방향 경로상의 TCP 데이터와 ACK, 이렇게 두 부분으로 분류된다.

간섭 발생의 첫 번째 이유는 인근지역에서 발생하는 동시전송을 막기 위한 802.11 MAC 계층의 전송간섭에 의해서다. 802.11 MAC은 멀티홉(multi-hop) 토폴로지에서의 신호간섭을 완벽하게 처리하지 못한다. TCP 패킷이 많아지면 링크계층의 재전송도 매우 많아진다. 결론적으로 이는 더 많은 MAC 손실을 초래하게 된다. 최근 802.11 멀티홉(multi-hop) 네트워크에서 TCP Congestion Window가 작을수록 성능이 좋다는 많은 연구<sup>[2~3, 18~20]</sup>가 있었다.

간섭의 두 번째 이유는 전송경로 상의 TCP 패킷과 ACK에 의해서 발생한다. 양 노드 간 TCP 패킷 전송 시, 데이터 패킷과 ACK 패킷 서로의 전송에 간섭이 일어날 수 있다.

위와 같은 상황에서 혼잡을 줄이는 기법들이 있다. Singh<sup>[21]</sup>는 여러 개의 패킷에 대해서 하나의 Adaptive Delayed Acknowledgement (TCP ADA)를 전송한다. Fu<sup>[3]</sup>에서는 RED라고 부르는 링크 계층 능동적 큐 관리 알고리즘을 제안하였다. LRED는 ECN을 이용하여 TCP window를 고정시키고, Adaptive Pacing(AP)을 이용하여 전송률을 감소시킨다.

Chen<sup>[17]</sup>에서는 홉수(hop-count)에 따른 최적의 window 값을 제안하였다. 그러나 Adaptive Congestion Window Limit (CWL)는 대역폭을 충분히 이용할 수 없다.

이 문제를 분석하여, 제안한 기법에서 RTT와 홉수(hop-count)를 이용하여 혼잡을 줄임으로서 적절한 CWND 증가율을 보일 것이다.

## III. 분석 및 TCP-CEV 기법

### 1. 홉 수에 따른 손실률의 영향

그림 1은 멀티홉(multi-hop) 링크 토폴로지를 보여준다.  $S_0$ 은 송신노드,  $D_0$ 은 수신노드,  $1, 2, \dots, n-1$ 은 중간노드,  $P_i$ 는  $i$ 번째 노드의 패킷 손실률이다. 연결 0의 손실

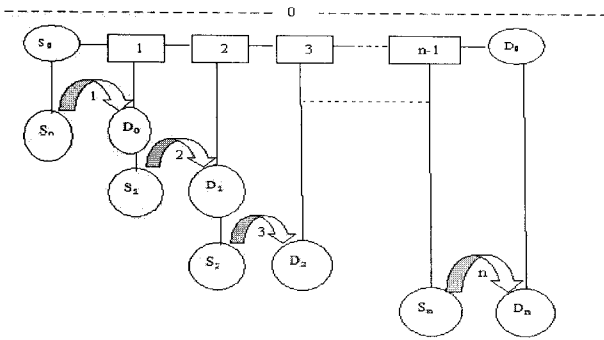


그림 1. 멀티홉(multi-hop) 링크 토폴로지  
Fig. 1. Multi-hop link topology.

를  $P_0$ 에 대한 수식은 아래와 같다.

$$P_0 = 1 - \prod_{i=1}^n (1 - P_i) \quad (1)$$

$P$  값이 매우 작을 경우에 수식은 아래와 같이 표현할 수 있다.

$$P_0 = 1 - (1 - p)^n \approx np \quad (2)$$

### 2. TCP window 기법의 영향

유명한 TCP-friendly equation은 TCP의 일정한 상태를 표현한 수학적 모델이다. 이는 오랜 시간 동안 얻은 평균 TCP windows 값을 설명하고 있고, [22]에서 평균 TCP 전송률을 나타내고 있다.

$$T = \frac{s}{R \sqrt{\frac{2P}{3} + t_{RTO} (3 \sqrt{\frac{3P}{8}}) p (1 + 32p^2)}} \quad (3)$$

$T$ 는 TCP의 전송률이고,  $s$ 는 패킷크기,  $R$ 은 RTT(round trip time),  $P$ 는 일정한 상태에서의 패킷 손실률이다.  $t_{RTO}$ 는 재전송시 사용되는 타임아웃 값이다.  $p$ 의 값이 매우 작은 경우 수식 (3)은 수식 (4)와 같이 간단히 표현할 수 있다.

$$T = \frac{s}{R \sqrt{\frac{2P}{3}}} \quad (4)$$

수식 (4)는 전송률의 상계를 나타낸다.

### 3. TCP-CEV 기법

멀티홉(multi-hop) 링크에서 수식 (1)과 수식(4)에 의한 원홉(one-hop)과 연결0 간의 연관값은 아래와 같다.

$$T_0 = \frac{R_1}{R_0} * \frac{1}{\sqrt{n}} * T_1 \quad (5)$$

$R_1$ 은 원홉(one-hop)의 RTT이고,  $R_0$ 은 연결 0의 RTT,  $n$ 은 홉수(hop-count)이다.  $T_0$ 는 연결 0에서 사용되는 전송률이다.  $T_1$ 은 원홉(one-hop)의 전송률이다.

이는 홉수(hop-count)와 RTT의 값이 변할 때, TCP 전송률은 수식 5에 따라 변한다는 것을 의미한다.

각 홉(hop)이 같은 RTT를 갖는다면,  $R_1/R_0$ 는  $1/n$ 이다. 이는 고정된 거리에 위치한 노드에 대해서는 성립이 되지만, 거리가 랜덤인 경우에는 성립이 되지 않는다. 단, TCP에서 RTT을을 제공하기 때문에, TCP에서 RTT를 쉽게 구현할 수 있다. 원홉(one-hop) 과 연결 0 사이의 연관값은 다음과 같다.

$$T_0 = \frac{1}{n} * \frac{1}{\sqrt{n}} * T_1 \quad (6)$$

RTT 차이를 고려하지 않는다면 수식 (7) 같이 간단히 표현할 수 있다.

$$T_0 = \frac{1}{\sqrt{n}} * T_1 \quad (7)$$

기존의 TCP 윈도우 메커니즘은 노드간의 홉수(hop-count)와 RTT를 고려하지 않고, 단순히 윈도우 사이즈를 조절함으로써 최대한 많은 패킷을 보내려고 하기 때문에 네트워크 상황이 변화하게 되면 성능이 크게 떨어질 수 있다.

이 문제점을 해결하기 위해 홉수(hop-count)와 RTT 변경 시 Change Expected Value (TCP-CEV) 을 고려한 적절한 CWND 변화율을 설정하여 높은 처리량을 얻을 수 있다.

TCP-CEV 프로토콜에서는 수식 5를 사용하여 CWND 증가율 조절한다.

RTT 변화와 홉수(hop-count)를 고려 시, TCP-CEV의 슬로우 스타트(slow start)는 다음과 같다.

방법 1.

$$CWND = CWND + \frac{R_1}{R_0} * \frac{1}{\sqrt{n}}$$

위 수식을 간단히 기술하면 다음과 같다.

방법 2

$$CWND = CWND + \frac{1}{n} * \frac{1}{\sqrt{n}}$$

RTT 상관없이 TCP-CEV의 슬로우 스타트(slow start)는 아래 수식으로 더 간단하게 표현 할 수 있다.

방법 3

$$CWND = CWND + 1/\sqrt{n}$$

시뮬레이션을 통해 성능 면에서 방법 1이 방법 3보다 좋다는 것을 증명을 할 것이다.

홉수(hop-count)가 적은 경우 TCP-NewReno의 슬로우 스타트(slow start)는  $CWND=CWND+1$ 이고, TCP 성능 면에서 최대 처리량을 얻을 수 있다. 이는 홉수(hop-count)가 작은 경우, MAC 계층 전송에서 자기간섭(self-interference) 문제가 발생하지 않기 때문이다. 그러므로 홉수(hop-count)가 작은 경우 홉수(hop-count)가 클 때와 달리, CWND 크기가 커질 때 TCP 성능에 영향을 주지 않는다. [17]에서 위의 사실이 확인되었다. 제안한 방법과 TCP-NewReno를 비교했을 때 같은 성능을 얻었고, 원홉(one-hop)일 때 최대 처리량을 얻을 수 있다. 언급한 바와 같이 홉수(hop-count)가 작은 경우 제안한 기법이 [2~3, 17~19]의 기법 보다 좋은 성능을 보임을 알 수 있다.

#### IV. 시뮬레이션 및 성능평가

IV장에서 TCP-CEV 프로토콜과 TCP-NewReno를 비교 평가한다. 시뮬레이션은 NS-2(Network Simulator 2)를 사용하였고, 체인 토폴로지에서 시뮬레이션을 하였다. 노드간 거리는 적당히 200미터로 고정하여 배치하였고, 6홉과 8홉, 두 가지 체인을 테스트하였다. 라우팅 프로토콜은 AODV(Ad hoc On demand Distance Vector)를 사용하였다. 제안한 기법이 TCP에서 성능 향상이 있음을 보일 것이다. TCP-NewReno를 수정하여, 경로상의 RTT와 hop-count를 고려한 TCP-CEV를 구현하였다.

##### 1. 체인 토폴로지

TCP 통신 시간은 0초부터 32초까지며, 모든 시뮬레이션은 35초 동안 동작한다. TCP-NewReno도 동일한 환경에서 테스트하였다.

홉수(hop-count)만 고려하였을 경우, 6홉 체인 토폴

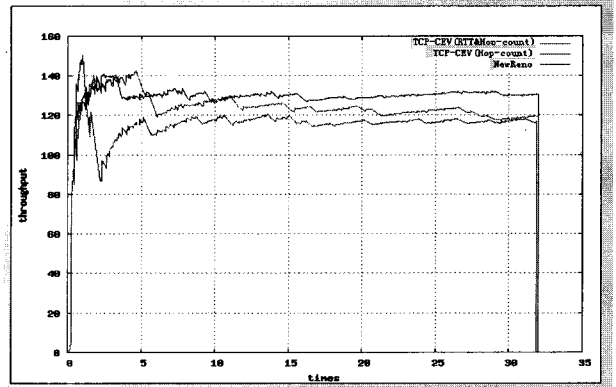


그림 2. 6-hop 체인 토폴로지에서 TCP-NewReno와 TCP-CEV의 처리량

Fig. 2. Throughput of TCP-NewReno and TCP-CEV in 6-hop chain topology.

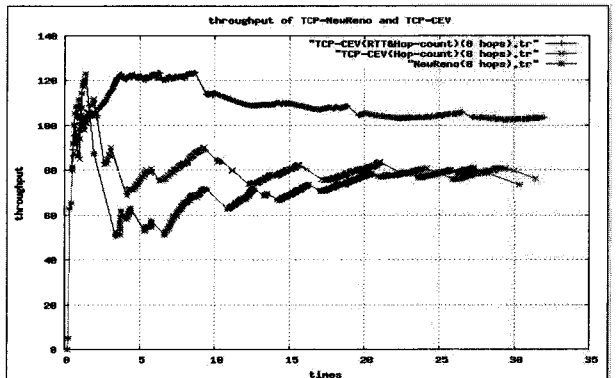


그림 3. 8-hop 체인 토폴로지에서 TCP-NewReno와 TCP-CEV의 처리량

Fig. 3. Throughput of TCP-NewReno and TCP-CEV in 8-hop chain topology.

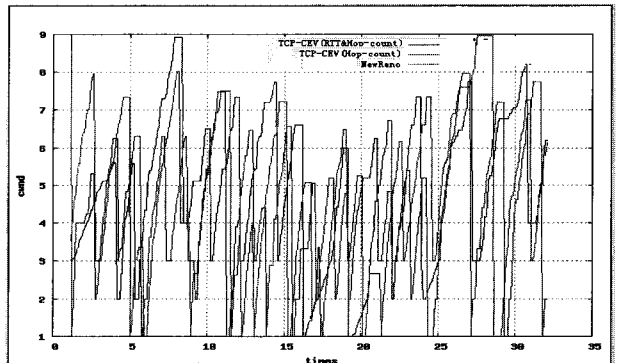


그림 4. 6-hop 체인 토폴로지에서 TCP-NewReno 와 TCP-CEV의 CWND

Fig. 4. CWND of TCP-NewReno and TCP-CEV in 6-hop chain topology.

로지의 TCP-CEV의 처리량이 TCP-NewReno보다 5% 성능 이득이 있음을 그림 2에서 보여주고 있다. RTT와 홉수(hop-count)를 동시에 고려하였을 경우,

TCP-NewReno보다 12%의 이득이 있었다. 이는 제안한 기법이 TCP-NewReno보다 효율적임을 보여준다.

그림 3은 RTT와 홉수(hop-count)를 동시에 고려하였을 경우, 8홉 체인 토폴로지의 TCP-CEV의 처리량이 TCP-NewReno보다 성능에서 20%증가하였음을 보여주고 있다. 홉수(hop-count)만 고려한다면, TCP-CEV가 5% 더 성능이 좋다. 이는 제안한 프로토콜이 RTT와 홉수(hop-count)에 따라 적합한 CWND 증가율을 설정하였기 때문이다.

위 결과는 III장에서 분석한 결과를 보여주고 있다. 3장에서 언급했듯이 홉수(hop-count)가 적을수록 TCP 성능이 더 좋다는 것을 그림 3과 4에서 보여주고 있다.

시뮬레이션에서 얻은 TCP-CEV의 두 가지 장점은 일정한 처리량을 얻을 때까지 시간이 TCP-NewReno보다 빠르다는 것과 CWND가 TCP-NewReno에 비해 일정하다는 점이다. 그림 4에서 TCP-CEV의 CWND가 더 일정함은 보여주고 있다.

## 2. 격자 토폴로지

이번 장에서는 3x3, 4x4, 5x5, 6x6, 7x7 무선 격자 토폴로지에서 시뮬레이션하여 두 프로토콜을 비교하고, TCP-CEV와 TCP-NewReno의 성능 향상을 측정한다. RTT를 제외하고 홉수(hop-count)만 고려하여 비교하기 위해 수식 3을 적용하였다. TCP 통신 시간은 1초에서 899까지이며, 시뮬레이션은 898초 동안 동작한다. 송신 노드는 왼쪽의 중간에 위치하고 수신 노드는 오른쪽 중간에 위치하도록 배치하였다.

그림 5, 6과 표 1, 2는 격자 토폴로지에서 TCP-CEV와 TCP-NewReno의 처리량과 지연을 비교하고 있다.

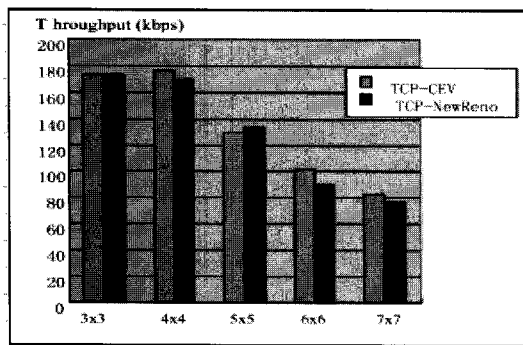


그림 5. NxN 격자 토폴로지에서 TCP-CEV와 TCP-NewReno의 처리량

Fig. 5. Throughput of TCP-CEV and TCP-NewReno in the NxN grid topologies.

표 1. NxN 격자 토폴로지에서 TCP-CEV와 TCP-NewReno의 처리량

Table 1. Throughput of TCP-CEV and TCP-NewReno in the NxN grid topologies

Grid	TCP-CEV ( kbps)	TCP-NewReno ( kbps)
3x3	173.6	173.6
4x4	176.8	170
5x5	129.2	133.65
6x6	101.7	91
7x7	82.7	77.3

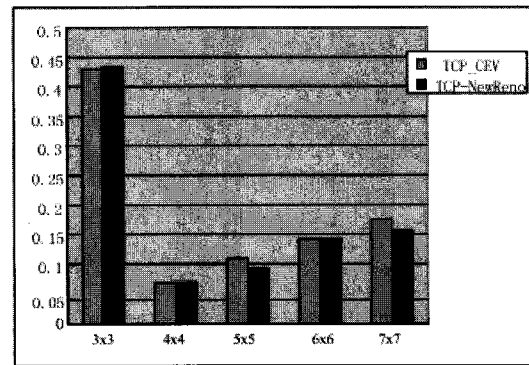


그림 6. NxN 격자 토폴로지에서 TCP-CEV와 TCP-NewReno의 지연

Fig. 6. Delay of TCP-CEV and TCP-NewReno in the NxN grid topologies.

표 2. NxN 격자 토폴로지에서 TCP-CEV와 TCP-NewReno의 지연

Table 2. Delay of TCP-CEV and TCP-NewReno in the NxN grid topologies.

Grid	TCP-CEV(s)	TCP-NewReno(s)
3x3	0.42917	0.43222
4x4	0.06980	0.07191
5x5	0.10999	0.09313
6x6	0.14296	0.14175
7x7	0.17543	0.15728

두 가지 프로토콜을 비교하였을 때, 평균 처리량은 TCP-CEV가 5% 이득이 있었고, 지연은 거의 비슷하였다.

## V. 결론 및 미래연구

지금까지 간단하지만 분석적인 방법으로 TCP-CEV

프로토콜을 소개하였다. 이 기법은 멀티홉(multi-hop) 정적 체인 토폴로지에서 패킷 전송률 변경 시 RTT와 홉수(hop-count) 정보를 사용하여 처리량을 향상시켰다. 모바일 네트워크에서 좋은 성능을 내기 위해서는 정적 무선 네트워크에서의 좋은 성능이 선행되어야 한다.

본 논문은 여러 부분에 기여하였다. 특히, TCP 성능에 영향을 주는 요인들을 분석하였다. 기본 TCP 구조를 따르는 대부분의 transport 프로토콜에 적용할 수 있다. 원홉(1-hop) 연결에서 최대 처리량을 얻었고, 성능 면에서 12% 이상 이득이 있었으며, 처리량이 일정한 값에 수렴한다.

다음 연구에는 모바일 환경에서 파워, 노드의 이동 속도 등을 고려하여 TCP-CEV 성능을 평가할 것이다.

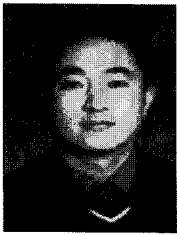
### 참 고 문 헌

- [1] M. Gerla, K. Tang, and R. Bagrodia, "Tcp performance in wireless multihop networks," in Proc. IEEE International Workshop on Mobile Computing Systems and Applications (WMCSA'99), New Orleans, Louisiana, USA, Feb. 1999.
- [2] Z. Fu, X. Meng, and S. Lu, "How bad tcp can perform in mobile ad hoc networks," in Proc. IEEE International Symposium on Computers and Communications (ISCC'02), Taormina, Italy, July 2002.
- [3] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on tcp throughput and loss," in Proc. IEEE Infocom 2003, San Francisco, California, USA, Apr. 2003.
- [4] Wang Haitao, Song Lihua. Performance of TCP in ad hoc network and its improvement polices [J]. Journal of NorthwestUniversity, 2004, 34(5): 442-445.
- [5] MANET Working Group. Mobile ad hoc networks [EB/OL]. . 2000-04-08
- [6] L. Brakmo, S. O'Malley, and L. Peterson. TCP Vegas: New techniques for congestion detection and avoidance. In Proceedings of the SIGCOMM '94 Symposium (Aug. 1994) pages 24-35.
- [7] TCP Westwood: End-to-End Bandwidth Estimation for Enhanced.. - Casetti, Gerla, al. (2002)
- [8] Kai Xu; TCP-Jersey for wireless IP communications Kai Xu; Ye Tian; Ansari, N. Selected Areas in Communications, IEEE Journal on Volume 22, Issue 4, May 2004 Page(s): 747 - 756
- [9] Reiner Ludwig The EifelAlgorithm: Making TCP Robust Against Spurious Retransmissions Reiner Ludwig Randy H.Katz Appears in ACM Computer Communications Review ,Vol. 30, No. 1, January 2000.
- [10] Donglojun TCP-BuS: Improving TCP Performance in Wireless Ad Hoc Networks - Donglojun Kimt Toby (2000)
- [11] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," in Proc. ACM Mobicom '99, Seattle, WA, 1999.
- [12] J. Liu and S.Singh, "ATCP: TCP for mobile ad hoc networks," IEEEJ-SAC, vol.19, no.7, pp. 1300 - 1315, 2001.
- [13] Z. Fu, X. Meng, and S. Lu, "A transport protocol for supporting multimedia streaming in mobile ad hoc networks," IEEE journal on selected areas in communications, vol. 21, no. 10, pp. 1615 - 1626, Decemeber 2004.
- [14] Sundaresan K, Anatharaman V, Hsieh H, Sivahkumr R. ATP: A reliable transport protocol or ad-hoc networks. In: Proc. of the ACM MobiHoc2003. 2003. 64-75.
- [15] Kopparty S, Krishnamurthy S, Faloutsos M, Tripathi SK. Split TCP for mobile ad hoc networks. In: Park JT, Ejiri M, eds. Proc. of the IEEE Global Telecommunications Conf. (GLOBECOM 02). IEEE, 2002. 138-142.
- [16] K. Chen, Y. Xue, S. Shah, and K. Nahrstedt, "Understanding bandwidth-delay product in mobile ad hoc networks," Special issue on protocol engineering for wired and wireless networks, Elsevier Computer Communications, vol. 27, pp. 923 - 934, 2004.
- [17] K. Chen, Y. Xue, and K. Nahrstedt, "On setting TCP's congestion window limit in mobile ad hoc networks." in Proc. IEEE ICC 2003, Anchorage, Alaska, May, 2003.
- [18] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?" IEEE Communications Magazine, pp. 130 - 137, June 2001.
- [19] K. Chen and K. Nahrstedt, "Limitations of equation-based congestion control in mobile ad hoc networks," in Proc. IEEE WWAN, Tokyo, Japan, March 2003.
- [20] Kitae Nahm, Ahmed Helmy, Jay Kuo. Cross-layer Interaction of TCP and Dynamic Routing in Multi-hop 802.11 Networks, ACM MobiHoc 2005.

[21] Singh, A.K. Kankipati, K.TCP-ADA: TCP with adaptive delayed acknowledgement for mobile ad hoc networks .Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE

[22] J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe, "Modeling TCP throughput: A simple model and its empirical validation," in Proc. ACM SIGCOMM '98, Vancouver, CA, September 1998, pp.303 - 314.

저 자 소 개



장 복 전(학생회원)  
 2000년 NorthEast Univ.  
 전자공학과 학사 졸업.  
 2005년 ShenYangLiGong Univ  
 컴퓨터공학과 석사 졸업.  
 2006년~현재 한양대학교  
 전자통신컴퓨터공학과  
 박사 재학.

<주관심분야 : 통신, 컴퓨터, 신호처리, 반도체>



김 준 환(학생회원)  
 2007년 안양대학교 디지털미디어  
 공학과 학사 졸업.  
 2007년~현재 한양대학교  
 전자통신컴퓨터공학과  
 석사 재학  
 <주관심분야 : 컴퓨터통신, 무선  
 네트워크>



박 용 진(정회원)  
 1969년 와세다대학교 전자통신  
 공학과 학사 졸업.  
 1971년 와세다대학교 전자통신  
 공학과 석사 졸업.  
 1978년 와세다대학교 전자통신  
 공학과 박사 졸업.

1979년~현재 한양대학교 전자통신컴퓨터공학부  
 교수  
 2001년 4월~2006년 와세다대학교 대학원 국제  
 정보통신대학원 강사  
 2002년 10월~2006년 10월 한국인터넷협회 이사  
 2003년 2월~현재 한국첨단망협회  
 (국내 연구개발망 추진기관) 이사  
 2005년 1월~현재 IEEE Asia-Pacific Region  
 이사  
 2005년 2월~현재 IEICE 한국 지부 대표  
 2009년~현재 IEEE 아시아·태평양지역 회장  
 <주관심분야 : 컴퓨터통신, 이동 데이터 통신>