韓國컴퓨터情報學會 論文誌
第14卷 第6號, 2009. 6.

2009-14-6-3-1

이동망 네트워크에서의 효율적인 TCP 알고리즘

홍 성 화*, 김 훈 기*

An Efficient TCP Algorithm in Mobile ADHOC Networks

Sung-Hwa Hong*, Hoon-Ki Kim*

요 약

TCP는 혼잡이 발생하는 유선망에서는 항상 패킷 손실이 발생하지만, 혼잡 손실과 전송 에러 손실을 분별할 수 있는 것은 아니다. 이는 송신자의 혼잡 윈도우 크기와 잃어버린 패킷의 재전송에 의해 무선망에서의 불필요한 TCP 성능 저하를 가져오게 된다는 가정을 유발한다. 또한 중복된 재전송은 이동 단말의 제한된 전원을 소비하게 한다. 본 논문에서는 네트워크 상태를 파악하고, 무선 애드혹 망에서의 에너지 효율과 TCP 성능 향상을 위한 혼잡을 방지하기 위한 알고리즘을 제안하였다. NS2를 이용하여 에너지 효율과 성능 향상을 이루어짐을 확인하였다.

Abstract

TCP assumes that packet loss is always happened by congestionlike wired networks because is can not distinguish between congestion loss and transmission error loss,. This assumption results in unnecessary TCP performance degradation in wireless networks by reducing sender's congestion window size and retransmitting the lost packets. Also, repeated retransmissions loed to waste the limited battery power of mobile devices. In this paper, we propose the new congestion control scheme that add the algorithms monitoring networks states and the algorithms preventing congestion to improve TCP throughput performance and energy efficiency in wireless ad-hoc networks. Using NS2, we showd our scheme improved throughput performance and energy efficiency.

▸ Keyword : TCP, ADHOC, Ubiquitous, Routing

• 제1저자 : 홍성화
• 투고일 : 2009. 04. 03, 심사일 : 2009. 05. 20, 게재확정일 : 2009. 05. 26.
* 동양공업전문대학 전산학부 소프트웨어공학과

Ⅰ. 서 론

오늘날 무선 통신 기술의 발달로 사용자가 원할 때 어디에서든 무선 통신을 이용하여 인터넷을 사용할 수 있는 환경이 제공 되고 있다. 이에 따라 무선 환경을 고려한 많은 연구가 진행되고 있으며, 무선 단말들로만 구성되어 데이터 통신을 하는 Ad-hoc 네트워크에 대한 연구도 활발히 이루어지고 있다. Ad-hoc 네트워크는 유선 네트워크와 달리 기지국의 도움 없이 무선 단말들로만 구성된 네트워크로서 무선 단말들은 상호간에 통신을 함에 있어 송신자, 수신자, 중계자 역할을 모두 수행하게 된다. 또한 무선 단말들로 이루어지기 때문에 언제 어디서나 신속하게 통신 네트워크를 구축할 수 있어 유비쿼터스 시대에 필수적인 기술이라 할 수 있다. 이러한 Ad-hoc 네트워크에서는 인터넷 검색, E-mail 송수신 등의 광대역 멀티미디어 서비스들을 지원하기 위해 종단(End to End) 간 신뢰성이 보장되는 TCP(Transmission Control Protocol) 프로토콜을 전송 프로토콜로 사용하고 있다.

그러나 인터넷에서 광범위하게 사용되고 있는 전송 프로토콜인 TCP는 전송 오류(Transmission Error)로 인한 패킷 손실이 거의 없는 유선 네트워크에 적합하게 설계되어 있어, 전송 오류가 상대적으로 빈번하게 발생하는 무선 환경의 특성을 고려하지 않고 있다. 따라서 무선네트워크에서 기존 TCP를 그대로 전송 계층 프로토콜로 사용할 경우, 성능이 크게 저하되는 결과를 초래하게 된다. 이의 원인은 TCP의 혼잡 제어 (Congestion Control)에서 찾을 수 있다[1][2]. TCP에서는 패킷 손실을 네트워크의 혼잡에 의한 것으로 간주하여 패킷 손실 발생 시 혼잡 제어를 수행하여 혼잡 윈도우를 줄임으로써 네트워크에 존재하는 데이터의 양을 조절하게 된다.

TCP는 패킷의 손실을 중복 ACK (duplicate Acknowledgment)와 Time Out에 의해 판단하게 되는데, 송신단에서는 일정 시간이 지나도록 수신단으로부터 ACK 패킷을 받지 못해 Time Out이 발생하거나 데이터 패킷의 손실을 의미하는 중복 ACK 패킷을 받을 경우 네트워크에 혼잡이 일어났다고 가정하여 전송 속도를 줄이게 된다. 즉 패킷 손실이 발생하기 전까지는 혼잡 윈도우의 크기를 증가시키고, 이로 인해 네트워크에 혼잡이 발생할 경우 혼잡 윈도우를 감소시킨 후 손실된 패킷을 재전송 하게 된다. 이와 같이 혼잡상황을 제외하고는 패킷 손실이 거의 발생하지 않는 유선네트워크의 경우, 패킷 손실이 발생하면 무조건 혼잡 상황으로 인식하여 혼잡 제어를 하는 TCP의 혼잡 제어가 잘 동작하게 된다.

그러나 유선 네트워크와 달리 무선 네트워크 환경은 일시적인 신호의 단절, 주파수 간섭, 잡음 등으로 불안정한 연결성을 갖으며 이로 인해 10-2 ~ 10-4의 높은 BER(Bit Error Ratio) 특성을 갖고 있어 패킷 전송 시 많은 손실이 발생하게 된다[3][4]. 이러한 무선 환경의 특성으로 인한 높은 BER에 대해 TCP는 알고리즘의 특성상 무선 네트워크에서 에러가 발생한 경우에도 이를 혼잡이 발생한 것으로 인식하여 Fast Retransmission, Slow Start등의 불필요한 혼잡 제어를 수행하게 된다. 따라서 유선 네트워크상의 고정 단말 간의 통신을 고려하여 설계된 기존의 TCP 프로토콜을 통신 환경이 급격하게 변화하는 무선 환경에 그대로 적용할 경우, TCP가 불필요한 혼잡 제어를 수행하게 되는 경우가 발생하게 되며, 전체적으로 링크 대역폭 이용률의 감소를 초래하여 결국 전송 성능 저하를 일으키게 되는 것이다.

이에 본 논문에서는 Ad-hoc 네트워크에서 이동 단말의 에너지 효율 및 전송 성능을 개선하기 위해 네트워크 상태에 따라 전송률을 적절하게 조절하고, 혼잡을 사전에 방지하여 불필요한 재전송을 줄이는 새로운 TCP 혼잡 제어 알고리즘을 제안한다. 제안한 알고리즘은 네트워크의 상태가 혼잡 상태인지 비혼잡 상태인지를 판단하여 이에 따라 전송률을 조절하게 되는데, NS-2 Network Simulator를 이용하여 기존 연구와 성능을 비교한 결과 제안한 혼잡 제어 알고리즘이 Ad-hoc 네트워크에서 에너지 효율 및 전송 성능에 있어 기존 알고리즘보다 우수한 성능을 보임을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서는 Ad-hoc 네트워크의 구성 및 무선 환경의 특성, Ad-hoc 네트워크에서 TCP가 사용될 때 발생하는 성능 저하 문제 및 이의 원인이 되는 TCP의 혼잡 제어 방법을 살펴보고, 기존 알고리즘의 문제점을 살펴본다. 3장에서는 본 논문에서 제안한 새로운 혼잡 제어 기법에 대해 설명한다. 4장에서는 모의 실험을 통해 본 논문에서 제안한 알고리즘을 TCP Reno, TCP Veno와 성능을 비교 분석하였다. 마지막으로 5장에서는 본 논문의 결론을 맺고 향후 연구 과제에 대해 기술한다.

Ⅱ. 관련 연구

2.1 TCP Reno

TCP Reno는 유선 네트워크에서 가장 널리 사용되는 TCP이며, Reno 의 혼잡제어는 Slow Start, Congestion Avoidance, Fast Retransmission, Fast Recovery로 구성되어 있다.

2.2 TCP Vegas

TCP Vegas는 송신측에서 전송한 패킷의 RTT를 측정하여 이를 기반으로 윈도우 크기를 조절하는 새로운 혼잡제어 알고리즘을 사용한다.

Vegas는 패킷 손실에 대한 정보를 바탕으로 네트워크의 혼잡제어를 수행하는 Reno와 달리 매번 측정된 RTT를 기반으로 하여 네트워크의 혼잡제어를 수행한다. 즉, RTT가 커지면 네트워크의 혼잡 정도가 증가된 것으로 판단하여 윈도우 크기를 선형적으로 감소시키고, RTT가 작아지면 네트워크의 혼잡 정도가 감소되었다고 판단하여 윈도우 크기를 선형적으로 증가시킨다.

2.3 TCP Veno

TCP Veno는 이름에서 알 수 있듯이 TCP Reno와 Vegas의 특성을 혼합한 방식으로, 혼잡 손실은 네트워크 상태의 혼잡으로 인해 패킷 전송 시간의 지연을 야기하지만 무선 손실은 무선 상에서 없어지는 패킷이므로 전송 시간의 지연이 발생하지 않는다는 점에 착안하여 만들어진 알고리즘이다[11]. 앞서 살펴보았듯이 TCP Vegas에서는 패킷 손실이 일어나는 것을 사전에 방지하기 위해 네트워크 상태를 측정하는데, Veno에서는 이를 이용하여 혼잡 손실과 무선 손실을 구분하는 근거로 사용한다.

TCP Veno가 혼잡 손실과 무선 손실을 구분하는 방식은 아래와 같다. 먼저 TCP vegas 에서 계산한 바와 같이 아래의 값들을 계산한다.

- Expected Throughput = CWND / BaseRTT
- Actual Throughput = CWND / RTT
- RTT = BassRTT + N / Actual Throughput
- N = Actual Throughput * (RTT - BaseRTT)

여기에서 CWND는 현재 TCP의 Window size를 의미하며, BaseRTT는 측정된 최소 RTT이고, RTT 는 현재 측정된 RTT이다. 만약 RTT 〉 BaseRTT 라면 병목 구간에 패킷이 축적되게 되는데, 이와 같이 축적된 패킷의 양이 N으로 표기된다.

TCP Veno는 위에서 측정된 N을 이용하여 현재 연결된 네트워크가 혼잡 상태인지 아닌지를 결정하게 되는데, 이는 Vegas 알고리즘에 기초하고 있다. 만약 패킷 손실이 발생했을 때 N 〈 β 라면, 네트워크가 혼잡하지 않은 상태로 간주하여 이때 발생한 패킷 손실은 무선 손실로 간주한다. 여기에서 β는 Vegas에서 혼잡을 판단하기 위한 Trigger로 사용하는 값으로 β= 3 의 값을 갖는다. 반면 N 〉 β 인 경우에 발생하는 패킷 손실은 네트워크 혼잡으로 인한 손실로 간주한다.

위와 같은 구분에 의해 TCP Veno는 아래와 같이 Additive Increase 구간에서 CWND의 값을 수정함으로써 혼잡 제어를 수행하게 된다.

- N 〉 β : 혼잡 손실로 간주된 경우

⇒ CWND = CWND + 1/2

(CWND 을 점진적으로 증가시켜 혼잡에 의한 손실 방지)

- N 〈 β : 무선 손실로 간주된 경우

⇒ CWND = CWND + 1

3개의 중복 ACK를 수신하거나 Time Out이 발생한 경우 TCP Veno는 패킷 손실을 인지하게 되는데, Time Out이 발생한 경우에는 TCP Reno와 동일한 방식으로 손실된 패킷을 재전송하고 혼잡 윈도우의 크기를 1로 설정하여 Slow Start 모드로 진입한다. 반면에 3개의 중복 ACK를 수신한 경우엔 아래와 같이 패킷 손실 원인을 파악 한 후 패킷 손실 원인에 따라 혼잡 윈도우의 크기를 재설정하여 전송을 재개한다.

- N 〉 β : 혼잡 손실로 간주된 경우

⇒ CWND = CWND / 2

(혼잡 윈도우를 반으로 줄임으로써 전송률을 감소시킴)

- N 〈 β : 무선 손실로 간주된 경우

⇒ CWND = CWND * 4/5

(혼잡 윈도우를 기존의 4/5로 변경 후 전송 재개)

TCP Veno는 네트워크의 상태를 모니터링하여 네트워크의 상태에 따라 전송률을 점진적으로 증가시키고, 패킷 손실 원인에 따라 전송률을 조절함으로 기존 TCP 가 무선 네트워크에서 사용될 때 발생하는 대역폭을 효율적으로 사용하지 못하는 문제를 개선할 수 있는 장점을 가진다. 그러나 네트워크의 상태를 모니터링 하는데 있어서 TCP Vegas의 알고리즘을 사용하기 때문에 TCP Vegas의 문제인 네트워크의 상황을 정확히 측정하지 못하는 문제점을 동일하게 갖게 된다.

이로 인해 TCP 의 전송 성능이 저하되게 되며, 재전송의 증가로 인해 에너지를 더 많이 소모하는 문제가 발생한다.

III. 제안 알고리즘

본 장에서는 2장에서 지적한 TCP Veno의 문제점을 개선하기 위해 새롭게 제안한 TCP New Veno 알고리즘을 설명

한다. TCP New Veno 알고리즘은 TCP Veno가 갖고 있는 문제점을 해결하기 위해 네트워크의 상태를 모니터링 하는 알고리즘을 추가 하였으며, 불필요한 재전송을 줄이기 위해 혼잡 상황을 사전에 방지하기 위한 알고리즘을 추가하였다.

3.1 네트워크 상태 모니터링 알고리즘

2장에서 살펴본 바와 같이 TCP Veno는 네트워크의 상태를 판단하는 데 있어 TCP Vegas의 알고리즘에 기초하고 있기 때문에, Vegas가 가지는 RTT의 부정확성 문제를 역시 가지게 된다. 즉 BassRTT에 의존하여 네트워크 혼잡을 조절하기 때문에, 부정확한 BassRTT를 사용하는 경우 Diff 값에 오류가 발생하고 되고, 이는 throughput에 직접적인 영향을 주게 된다. 이에 본 논문에서는 BassRTT에 대한 의존성을 줄이며, 네트워크의 상황을 더 정확히 판단하기 위해 바로 전 RTT와 현재의 RTT를 비교하여 네트워크 상태를 모니터링하며 그에 따라 구간별로 혼잡 제어에 대한 Parameter를 다르게 적용하는 알고리즘을 제안한다.

먼저 본 논문에서 제안한 알고리즘을 적용하기 위해 추가된 파라미터는 아래와 같다.

- Pre_RTT : 바로 직전의 RTT

본 알고리즘에서 Pre_RTT와 RTT의 관계를 통해 네트워크의 상황을 구분하는 방법은 아래와 같다.

1〉 현재의 RTT ≥ Pre_RTT 인 경우 : 현재의 RTT가 이전 RTT보다 커진 경우는 네트워크상에 flow가 증가하여 지연이 더 커진 경우라고 볼 수 있다. 즉, 바로 직전보다 네트워크의 상태가 더 혼잡해지고 있음을 의미한다고 할 수 있다. 따라서 이 경우엔 혼잡이 경감되도록 flow를 조절할 필요가 있다.

2〉 현재의 RTT 〈 Pre_RTT 인 경우 : RTT가 작아지는 경향을 보이는 것은 네트워크상에 flow가 줄어들고 있어 지연이 감소됨을 의미한다. 따라서 이 경우 혼잡이 줄어들고 있으며, 이전 상황보다는 네트워크의 가용량이 늘어났다고 볼 수 있다. 이 경우엔 늘어난 가용량을 사용하도록 flow를 증가시킬 수 있는 구간으로 정의한다.

본 논문에서는 위에서 말한 바와 같이 이전 RTT와 현재의 RTT를 비교하여 상기와 같이 네트워크의 상태를 정의하며 이에 따라 아래와 같이 혼잡 제어를 수행하게 된다.

```
If ( N <  β )   // non - congestion 인식 상황
    CWND = CWND + 1
If ( N > β )    // congestion 인식 상황
    If ( RTT >= Pre_RTT )
        // 네트워크의 혼잡이 증가되고 있는 경우
        CWND = CWND + 1/2
    Else  //  네트워크의 혼잡이 감소하고 있는 경우
        CWND = CWND + 1
```

위의 알고리즘에서 현재의 RTT가 이전 RTT보다 커진 경우 네트워크의 혼잡이 이전보다 증가되고 있음을 나타낸다. 따라서 이 구간에서는 네트워크 상의 flow 증가분을 기존 알고리즘인 TCP Veno와 동일하게 1/2로 증가 시키게 된다.

반면 현재의 RTT가 Pre_RTT보다 작은 경우엔 네트워크의 혼잡이 경감되고 있는 경우이므로, 이 구간에서는 네트워크의 가용량을 사용하기 위해 CWND를 조금 더 크게 증가시키게 된다.

3.2 사전 혼잡 방지 알고리즘

본 논문에서는 혼잡 윈도우를 TCP의 상태에 따라 저장된 RTT값으로 제어하여 혼잡을 회피하도록 제안된 혼잡 제어 알고리즘[12]을 적용하였고 그 알고리즘을 그림 1에서 보였다. 이 사전 혼잡 방지 알고리즘은 혼잡이 발생했을 때의 RTT값을 기억해, 이후 RTT값이 이 값보다 커지는 경우 CWND가 증가하는 것을 방지함으로써 혼잡 발생과 그로 인한 재전송을 줄이기 위해 제안된 방법이다.

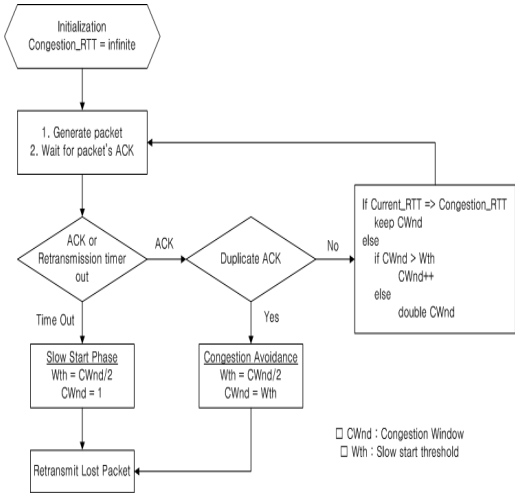


그림 1. 기 제안된 혼잡제어 알고리즘
Fig. 1. Pre-propsed congestion control algorithnm

이 알고리즘을 적용하기 위해 추가된 파라미터는 아래와 같다.

• Max_RTT : 수신된 RTT 중 최대값으로 초기값은 0으로 설정한다.

• Congestion RTT: 혼잡이 발생했을 시의 RTT를 의미하며, 패킷의 손실이 네트워크가 혼잡 상황에서 발생한다는 가정 하에서 혼잡 상황의 재 발생을 회피하기 위해 사용하는 파라미터이다. 초기값은 ∞로 설정된다.

알고리즘의 상세 동작은 아래에 설명한다.

1. 패킷을 정상적으로 수신했을 경우

Max_RTT = max (RTT, Max_RTT)

패킷을 수신한 경우 현재 RTT를 수신한 패킷의 RTT로 설정하게 되며, Max_RTT값은 현재 RTT와 기존 Max_RTT 중 큰 값으로 설정한다.

2. Slow Start 상태인 경우

If (RTT 〉= Cogestion_RTT)
Keep CWND
Else CWND = CWND + 1

Slow Start 단계에 있는 경우 현재 TT를 Congestion RTT와 비교하여 이에 따라 CWND 를 현 상태로 유지하거나 증가시키게 된다.

3. Additive Increase 상태인 경우

If (RTT 〉= Congestion_RTT)
Keep CWND
Else CWND = CWND +1

Slow Start 단계와 마찬가지로, RTT가 Congestion RTT보다 작은 경우에만 CWND를 증가시킨다.

4. 3 duplicate ACK를 수신한 경우

Congestion_RTT = Max_RTT
Max_RTT = 0

세 개의 중복 ACK의 수신은 혼잡 상태를 의미하기 때문에, 혼잡 상태의 RTT를 기억하기 위해 Congestion RTT를 이때의 Max RTT 값으로 갱신하고, Max RTT 값을 초기화한다.

본 논문에서는 위에서 설명한 네트워크의 상황을 판단하는 알고리즘과 사전 혼잡 회피 알고리즘을 동시에 적용하여 혼잡에 의한 패킷 손실율을 감소시키고, 혼잡 손실과 무선 손실을 더 정확히 구별하여 혼잡 제어를 수행함으로써 불필요한 재전송을 줄이고 에너지 효율 및 전송 성능 향상을 유도한다.

IV. 제안 알고리즘의 성능 시뮬레이션

본 논문에서 제안한 New Veno의 성능을 평가하기 위해, LBNL(Lawrence Berkely National Laboratory)의 ns (네트워크 simulator) 2.31을 이용하였다[8].

제안한 New Veno의 성능은 TCP Reno, TCP Veno의 성능과 비교하였다.

제안한 New Veno의 성능을 평가하기 위해 그림 2과 같은 실험 환경을 구성하여 성능 평가를 진행하였다.

그림 2. 시뮬레이션 네트워크 토폴로지
Fig. 2. Network topology of the simulation

전체 네트워크 사이즈는 1000 x 500이며 무선 네트워크의 대역폭은 2Mbps이고, 전달 지연은 TwoRay Ground 모델을 사용하였다. 이동 단말의 전송 범위는 250m로 제한하였으며, 송신자가 직접 수신자에게 패킷을 전달하지 않고 중계 단말을 통해 전송할 수 있도록 단말간 거리는 200m로 설정하였다. MAC은 802.11 MAC을 사용하였고 Routing Protocol은 AODV를 사용하였다. 전송 에러에 의한 단말의 에너지 효율 및 전송 성능을 평가하기 위해 단말의 이동성은 배제하였으며, BER을 0.01 ~ 0.1로 변경하여 BER에 따른 성능 평가를 진행하였다. 에너지 효율을 평가하기 위해 단말의 초기 에너지는 1000J(Joule)로 설정하였으며, 패킷 전송시 0.6W(Watt), 패킷 수신시 0.3W를 소비하도록 설정하였다. 성능 평가는 200초 동안 진행하였다.

4.1 패킷 에러율에 따른 전송 성능 비교

먼저 패킷 에러에 따른 각 TCP의 전송 성능을 비교한다.

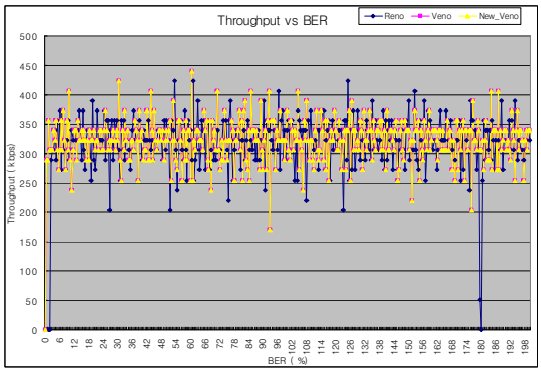


그림 3. BER 2%에서 TCP Throughput 비교
Fig. 3. A TCP Throughput comparison in BER 2%

그림 3~5는 전송 에러에 의해 패킷 손실이 발생했을 때 시간에 따른 TCP Reno, TCP Veno, TCP New Veno의 전송 성능을 보여준다. 그림 2는 패킷 에러율이 2% 일 때의 각 TCP에 대한 전송 성능을 나타낸 것으로 TCP Veno 및 TCP New Veno의 성능이 TCP Reno에 비해 안정적임을 볼 수 있다. 그림 4는 패킷 에러율이 5% 일 때 각 TCP의 전송 성능을 보여주는 것이며, 그림 5는 패킷 에러율이 10% 일 때 각 TCP 의 전송 성능을 보여준다. TCP New Veno는 BER 2%에서는 TCP Veno와 비슷한 성능을 보인다. 하지만 그림 4과 그림 5에서 보는 바와 같이 패킷 에러율이 증가하게 되면, TCP New Veno가 TCP Veno 및 TCP Reno에 비해 더 나은 성능을 보이는 것을 알 수 있다.

BER이 높아질수록 TCP Reno의 성능이 낮은 이유는 전송 에러에 의한 패킷 손실이 발생했을 때, 이를 혼잡과 구별할 수 있는 알고리즘이 없기 때문에 불필요하게 혼잡 제어를 수행하여 전송률을 낮추게 됨으로 인한 것이다. TCP Veno는 네트워크의 상태를 모니터링하여 혼잡과 비혼잡에 의한 손실을 구분하는 알고리즘이 추가되었기 때문에 TCP Reno에 비해 더 나은 성능을 보인다. 그러나 앞서 언급한 바와 같이 네트워크의 상황을 정확히 알지 못하기 때문에 제안한 알고리즘보다 낮은 성능을 보이게 된다.

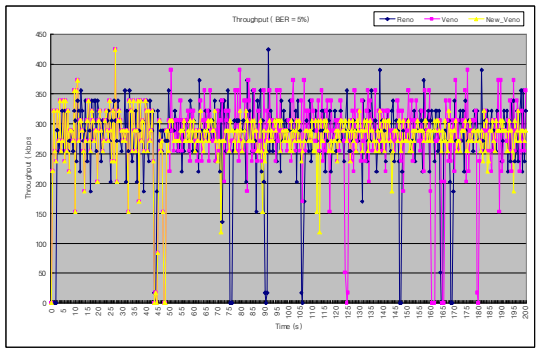


그림 4. BER 5%에서 TCP Throughput 비교
Fig. 4. A TCP Throughput comparison in BER 5%

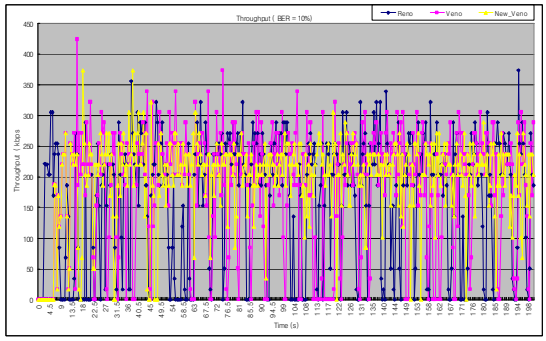


그림 5. BER 10%에서 TCP Throughput 비교
Fig. 5. A TCP Throughput comparison in BER 10%

그림 6는 에러율이 0 ~ 10% 까지 변할 때 각 TCP 프로토콜의 전송 성능을 비교한 것이다. 패킷 에러율이 낮을 때는 세 프로토콜이 비슷한 성능을 보이지만, 패킷 에러율이 증가할수록 제안한 TCP New Veno의 성능이 더 우수함을 볼 수 있다.

제안한 TCP New Veno의 경우 사전 혼잡 방지 알고리즘의 적용으로 혼잡 상황을 사전에 방지할 수 있게 되었고, 네

트워크 모니터링 알고리즘을 통해 손실 발생 시 혼잡 손실인지 무선 손실인지 더 정확하게 파악하여 패킷 손실의 원인에 따라 적절하게 혼잡 제어를 수행함으로써 더 나은 성능을 보이게 된다.

그러나 패킷 에러율이 높아질수록 패킷 손실로 인한 중복 ACK의 발생과 Time Out 발생 빈도가 증가하기 때문에 전체적인 성능 저하가 발생하게 된다.

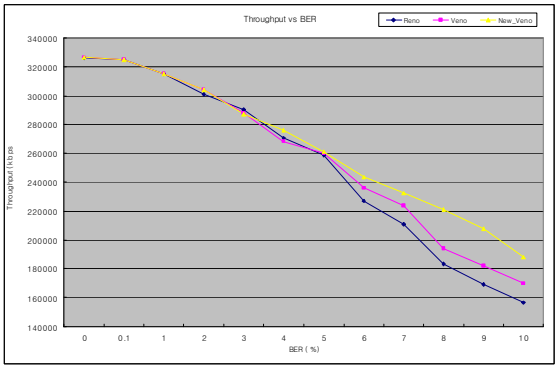


그림 6. BER에 따른 TCP Throughput 비교
Fig. 6. TCP Throughput comparison in different TCP

그림 7은 패킷 에러율에 따른 각 TCP 프로토콜의 receive ratio를 비교한 것이다. Receive ratio는 수신 단말이 수신한 패킷을 송신 단말이 전송한 패킷으로 나눈 것으로 전송 효율을 보여주는 것이다. 그림 6에서 보는 바와 같이 패킷 에러율이 높을수록 TCP New Veno의 성능이 더 우수함을 볼 수 있다.

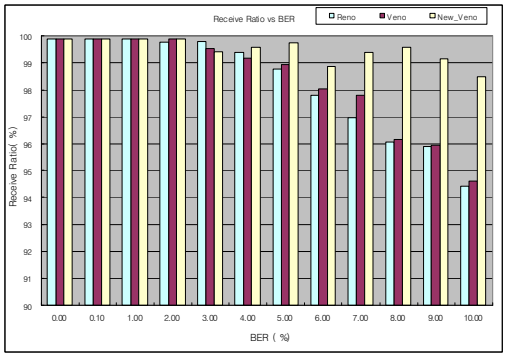


그림 7. BER에 따른 TCP receive ratio 비교
Fig. 7. TCP receive ratio comparison in different TCP

4.2 패킷 에러율에 따른 에너지 효율 비교

다음으로 각 TCP 프로토콜의 에너지 효율을 비교하기 위해 동일한 실험 환경에서 0 ~ 10%의 패킷 에러율에 따른 에너지 효율을 비교하였다.

에너지 효율은 아래와 같이 계산 할 수 있다.

$$Energy Efficiency(\eta) = \frac{Throughput}{Consumed Energy}(Kb/sJ)$$

에너지 효율은 송신 단말이 전송한 데이터의 양을 송신 단말이 소모한 총 에너지로 나눈 값으로 표현되며, 이동 단말이 일정 에너지로 서비스할 수 있는 데이터의 양을 나타내기 때문에 값이 더 클수록 에너지 효율이 좋다고 할 수 있다. 위의 식을 이용하여 TCP Reno, TCP Veno 및 제안한 알고리즘인 TCP New Veno의 에너지 효율을 평가하였다.

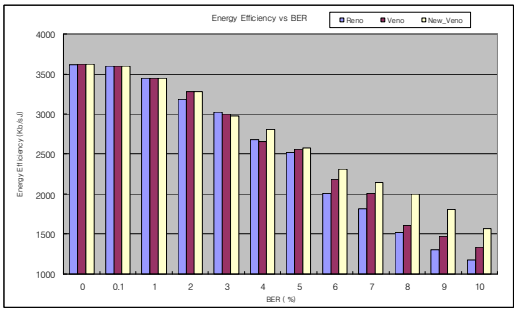


그림 8. BER에 따른 TCP 에너지 효율 비교
Fig. 8. TCP Energy Efficiency comparison in different BER

그림 8은 각 TCP 프로토콜의 패킷 에러율에 따른 단말의 에너지 효율을 보여준다. TCP Throughput의 결과와 비슷하게 BER이 낮을 때 세 프로토콜의 에너지 효율은 거의 동일한 성능을 보였으나, 패킷 에러율이 높아질수록 제안한 알고리즘인 TCP New Veno의 효율이 더 높음을 알 수 있다. 이는 TCP New Veno에서 네트워크의 상황을 더 정확하게 파악하여 혼잡 손실과 무선 손실을 더 정확히 구분하고, 이를 바탕으로 손실의 원인에 따라 적절하게 혼잡제어를 수행함으로 인해 패킷 손실을 감소시키기 때문이다. 또한 사전 혼잡 방지 알고리즘의 적용으로 혼잡으로 인한 패킷 손실이 줄어들고, 이로 인해 재전송 횟수도 줄어들기 때문에 더 나은 성능을 보이게 되는 것이다.

V. 결 론

Ad- hoc 네트워크는 신뢰할 수 없는 무선 채널을 사용함

으로써 무선 환경의 특성으로 인한 높은 패킷 에러율을 가지며, 전원이 한정적인 배터리를 사용하는 이동 단말들로 구성되는 특징을 가진다. 기존 TCP 는 신뢰성 있고 전원의 제약이 없는 유선네트워크를 기반으로 디자인되어, 무선 네트워크에 사용될 경우, 무선 환경의 특성으로 인해 발생하는 에러와 혼잡을 구별하지 못해 무선 환경에서 사용될 경우 성능 저하를 야기한다. 또한 이동 단말이 송신자, 수신자, 중계자 역할까지 수행함으로 인해 무선 단말의 수명이 네트워크 수명에 지대한 영향을 미치게 된다. 이러한 문제를 해결하기 위해 본 논문에서는 에너지 효율적이면서 전송 성능을 높일 수 있는 TCP New Veno 알고리즘을 제안하였다. 본 알고리즘은 네트워크의 상태를 파악하여, 네트워크의 상태에 따라 전송률을 조절하고, 패킷 손실이 발생했을 시 그때의 RTT를 기억하여 이후 RTT가 저장된 RTT값보다 큰 경우 CWND를 증가시키지 않음으로써 사전에 혼잡을 방지하는 기법이다. 제안한 알고리즘의 성능 평가를 통해 기존 TCP Reno 및 TCP veno에 비해 더 나은 성능을 보임을 확인하였다.

본 논문에서는 에너지 효율을 측정하기 위해 이동 단말의 이동성을 배제하고 성능 평가를 진행하였다. 향후 단말이 빈번하게 이동함으로 인한 경로 단절이 발생할 경우를 반영하기 위한 연구가 수행되어야 하고, 다른 Routing Protocol 환경에서의 성능 평가가 이루어져야 할 것이다.

참고문헌

〔1〕 V. Jacobson, "Congestion Avoidance and Control", In Proceedings of ACM SIGcomm'88, Vol. 18, No. 4, p314-329, August 1988

〔2〕 W. R. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmission, and Fast Recovery Algorithms", IETF, RFC 2001, Jan. 1997

〔3〕 H. Balakrishnan, "Challenges to reliable Data Transport over Heterogeneous Wireless Networks", PhD Thesis, University of California at Berkeley, 1998.

〔4〕 G. Xylomenos, et al. "TCP performance issues over wireless links",IEEE Commun. Mag., p2-14, Fourth quarter 2000

〔5〕 W. R. Stevens, "TCP/IP Illustrated", Vol. 1, Addison-Wesley, Nov. 1994

〔6〕 V. Jacobson, "Modified TCP Congestion Avoidance Algorithm", LBNL Technical Report, April 1990.

〔7〕 L. Brakmo, S. O'Malley and L. Peterson, "TCP Vegas: New Techniques for Congestion Detection and Avoidance", Proceeding of ACM SIGCOMM '94, pp. 24-35, August 1994.

〔8〕 The 네트워크 simulator (NS-2) *http://www.isi.edu/nsnam/ns/*

〔9〕 D. B. Jonson, D A. Maltz, U. Hu, "The dynamic source routing protocol for mobile ad hoc networks", IETF Interne Draft. *http://www.ietf.org/internet-drafts /draft-ieft-manet-der-08.txt, 2003*

〔10〕 L. Chen and W. B. Heinzelman, "QoS-aware Routing Based on BandWidth Estimation for Mobile Ad Hoc Networks", IEEE Journal on Selected Area in Communications (JSAC '05), vol. 23, pp.561-572, Mar.2005.

〔11〕 C. P. Fu, S. C. Liew, "TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks", IEEE Journal of Selected Areas in Communications, February 2003.

〔12〕 최지현, 김대영, 김관웅, 정경택, 천병실, "네트워크 효율 향상을 위한 개선된 TCP 혼잡 제어 알고리즘," 전자공학회 논문지, 2003.8월

〔13〕 E. M. Royer and C. K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", IEEE Personal Communications, vol.6, pp.46-55, Apr.1999

〔14〕 X. Zou, B. Ramamurthy, and S. Magliveras "Routing Techniques in Wireless Ad Hoc Networks - Classification and Comparison",The Sixth World Multi-conference in Systemics , Cybernetics, and Informatics(SCI02), vol.4, Jul. 2002

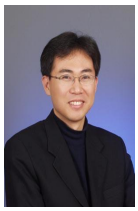
〔15〕 C.E. Perkins and P.Bhagwat "Highly Dynamic Destination Sequenced Distance-Vector Routing (DSDV) for Mobile Computers",ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM'94), pp.234-244, Aug.1994

〔16〕 P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc

Networks", Proceedings of IEEE International Multi Topic Conference (INMIC'02),pp.62-68, Dec. 2002

[17] S. Vutukury and J.J. Garcia-Luna-Aceves, "MDVA: A Distance vector Multipath Routing Protocol", In Proceeding of IEEE INFOCOM (INFOCOM '01), vol. 1, pp. 557-564, Apr.2001.

저 자 소 개



홍 성 화

1996: 고려대학교 전자계산학과 이학사.
2002: 한국항공대학교 정보통신공학과 공학석사.
2008: 고려대학교 전자컴퓨터공학과 공학박사
현 재: 동양공업전문대학 소프트웨어학과 전임강사
관심분야: 유비쿼터스 네트워크, 홈 네트워크, 무선네트워크, 임베디드 시스템,

김 훈 기

1988: 한양대학교 전자공학과 공학사.
1990: 한양대학교 전자공학과 공학석사.
2002: 한양대학교 전자공학과 공학박사
현 재: 동양공업전문대학 소프트웨어학과 조교수
관심분야: 임베디드 시스템, 센서 네트워크, 로봇 S/W, 통신 시스템