

Ad-hoc 통신망의 에너지 효율적인 혼잡 제어 기법

(Energy Efficient Congestion Control Scheme in Ad-hoc Networks)

조 남 호 [†] 정 광 수 ^{**}
(Namho Cho) (Kwangsue Chung)

요 약 최근 기지국(Base Station)의 도움 없이 이동 단말기(Mobile Device) 간의 다중 무선 홉을 사용하여 송수신자 간의 데이터 전송을 가능하게 하는 Ad-hoc 통신망에 관한 연구가 활발히 진행되고 있다. 현재 인터넷 상에서 광범위하게 사용되고 있는 전송 규약인 TCP(Transmission Control Protocol)는 수동적인 혼잡 제어(Reactive Congestion Control)방식을 사용하여 망의 혼잡으로 인한 패킷 손실 발생 이전까지 혼잡 윈도우의 크기를 증가시킨다. 따라서 반복적인 혼잡과 그로 인한 패킷 손실로 인해 재전송을 반복하게 된다. 이와 같이 기존 TCP는 Ad-hoc 통신망에서 동작하는 이동 단말의 한정된 배터리 전원을 고려하지 않고 동작하기 때문에 이동 단말의 에너지를 불필요하게 낭비하는 문제를 가지게 된다.

본 논문은 Ad-hoc 통신망에서 이동 단말의 에너지 효율을 개선하기 위해 불필요한 재전송 방지 및 망 상태에 따라 전송률을 적절하게 조절하는 새로운 혼잡 제어 기법인 TCP-New Veno를 제안하였다. 제안한 TCP-New Veno는 이동 단말의 에너지 효율 및 대역폭의 사용률(Utilization)을 향상시킬 있도록 설계되었다. 또한 ns-2 시뮬레이터를 이용한 실험을 통해 제안한 TCP-New Veno에 의해 이동 단말의 에너지 효율이 향상되었음을 확인하였다.

키워드 : Ad-hoc 통신망, TCP, 이동 단말의 에너지 효율, 혼잡 제어

Abstract In recent years, there have been many researches about Ad-hoc Networks which is available to communicate freely between mobile devices by using multi-hop without any support of relay base or access point. TCP that used the most widely transport protocol in the Internet repeats packet loss and retransmission because it increases congestion window size by using reactive congestion control until packet loss occurs. As a result of this, energy of mobile device is wasted unnecessarily.

In this paper, we propose TCP-New Veno in order to improve the energy efficiency of mobile device. According to the state of network, the scheme adjusts appropriate size of congestion window. Therefore, the energy efficiency of mobile device and utilization of bandwidth are improved by the scheme. From the simulation by using ns-2, we could see more improved energy efficiency with TCP-New Veno than those with TCP in Ad-hoc Networks.

Key words : Ad-hoc Networks, TCP, Energy efficiency of mobile devices, Congestion control

1. 서 론

오늘날 무선 환경에서 인터넷을 사용할 수 있는 무선

인터넷의 보급으로 인해 그 사용자가 크게 증가하고 있다. 그로 인해 무선 환경을 고려한 많은 연구가 진행되고 있으며, 기지국의 도움 없이 이동 단말기 간의 다중 무선 홉을 사용하여 송·수신자 간의 데이터 전송을 가능하게 하는 Ad-hoc 통신망에 관한 연구도 활발히 진행되고 있다. Ad-hoc 통신망은 기존의 정적인 네트워크 구조에 구애받지 않고 언제 어디서나 신속하게 통신망을 구축 가능한 기술로써 다가오는 유비쿼터스(Ubiquitous) 컴퓨팅 시대에 필수적인 기반 기술이다.

본 연구는 한국과학재단 특정기초연구(R01-2005-0000-10934-0)의 지원에 의해 수행되었으며, 또한 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성 지원사업의 연구결과로 수행되었음

[†] 정 회 원 : 광운대학교 전자통신공학과
code23@nate.com

^{**} 정 회 원 : 광운대학교 전자통신공학과 교수
kchung@daisy.kwangwoon.ac.kr

논문접수 : 2005년 10월 17일

심사완료 : 2006년 5월 22일

하지만, Ad-hoc 통신망은 유선망과 다른 특성으로 인해 통신상의 많은 제약점을 가지고 있다. 예를 들면, Ad-hoc 통신망은 비신뢰적인 전송 매체인 무선 채널을 사용하기 때문에 높은 비트 에러율(High Bit Error Rate)과 망을 구성하는 이동 단말의 이동성으로 인한 동적인 토폴로지를 가지기 때문에 잦은 경로 변경(Route Change) 등과 같은 유선망과 다른 통신상의 제약점을 가지고 있다. 따라서 이러한 통신상의 제약을 극복하기 위한 많은 연구가 진행되고 있다[1,2].

Ad-hoc 통신망을 구성하는 이동 단말이 배터리 전원에 의존하여 동작하기 때문에, 이동 단말의 에너지 효율은 Ad-hoc 통신망에서 통신상의 제약점 중에 하나로 중요한 비중을 차지한다. 또한, Ad-hoc 통신망에서 이동 단말은 중단으로만 동작하는 것이 아닌 증계단으로도 동작하기 때문에 데이터 전송의 경로를 제공하는 중요한 역할도 수행한다. 이는 이동 단말의 수명(Life Time)은 망의 수명과 관련이 깊다고 할 수 있다. 따라서 이동 단말의 제한된 배터리 전원을 효율적으로 이용하기 위한 프로세싱 하중과 통신 프로토콜에 의한 에너지 소모를 줄일 수 있는 방법이 필요하다[3,4].

TCP는 현재 인터넷 상에서 광범위하게 사용되고 있는 전송 프로토콜로써 유선망의 신뢰적인 전송 매체를 고려하여 개발되었다. TCP의 혼잡 제어 방식은 수동적인 혼잡 제어 방식을 사용하여 망의 혼잡으로 인한 패킷 손실 발생 이전까지 혼잡 윈도우의 크기를 증가시킨다. 이러한 동작은 망의 혼잡을 반복적으로 발생시킴으로써 불가피한 패킷 손실이 발생하며, 손실된 패킷을 복구하기 위한 재전송을 반복하게 된다. 기존 TCP의 반복된 재전송은 유선망에서는 통신상의 제약이 되지 않지만, Ad-hoc 통신망에서는 이동 단말의 한정된 배터리 전원을 불필요하게 낭비하는 문제를 초래하게 된다[5,6].

본 논문에서는 Ad-hoc 통신망에서 이동 단말의 에너지 효율을 개선하기 위해 망 상태에 따라 전송률을 적절하게 조절함으로써 불필요한 재전송을 방지하는 TCP의 새로운 혼잡 제어 기법을 제안하였다. 제안된 혼잡 제어 기법은 망의 상태가 혼잡 상태인지 비혼잡 상태인

지를 판단하여, 혼잡 상태일 때는 보수적(Conservative)으로 전송률을 조절하고 비혼잡 상태일 때는 공격적(Aggressive)으로 전송률을 조절한다. 그럼으로써 이동 단말의 에너지 효율을 향상시켰을 뿐만 아니라, 망의 사용률도 향상시킬 수 있었다.

본 논문은 다음과 같이 구성된다. 2장에서 Ad-hoc 통신망에서 기존 TCP의 문제점과 그 문제점을 해결하기 위한 관련 연구에 대해 살펴본 후, 3장에서는 본 논문에서 제안하는 새로운 혼잡 제어 기법에 대해 설명한다. 4장에서는 제안한 혼잡 제어 기법의 성능 평가를 위한 실험과 5장에서는 결론 및 향후 과제에 대해 기술한다.

2. 관련 연구

본 장에서는 Ad-hoc 통신망에서 기존 TCP의 문제점을 기술하며, 이러한 문제를 해결하기 위해 제안된 TCP-Veno에 대한 설명과 문제점에 대해서 기술한다.

2.1 Ad-hoc 통신망에서 기존 TCP의 문제점

기존 TCP는 패킷 손실 이후에 전송률을 조절하는 수동적인 방법으로 혼잡 제어를 수행 한다. 수동적인 혼잡 제어 방식은 패킷 손실 이후에 혼잡 윈도우의 크기를 줄임으로써 망의 혼잡을 완화하기 위한 방식으로, 패킷 손실 이전까지 전송률 증가를 위해 혼잡 윈도우의 크기를 계속적으로 증가시키게 된다. 이러한 수동적인 혼잡 제어 방식은 현재 망의 용량(Capacity)을 고려하지 않고 전송률을 증가시키기 때문에 망의 혼잡을 야기하며, 그로인한 반복적인 패킷 손실을 발생시키게 된다. 따라서 손실된 패킷의 복구를 위한 재전송을 반복하게 된다[7,8].

그림 1은 TCP-Reno의 혼잡 윈도우 변화를 나타내는 그림으로써 그래프의 곡선이 반복된 톱니파(Sawtooth Wave)의 모양임을 확인할 수 있다. 이와 같이 기존 TCP는 망의 상태를 고려하지 않고 패킷 손실 이전까지 혼잡 윈도우의 크기를 반복적으로 증가시키는 것을 확인할 수 있으며, 이를 복구하기 위한 반복적인 재전송이 발생한다는 것을 확인할 수 있다. 기존 TCP의 수동적인 혼잡 제어 방식은 유선망에서는 문제가 되지 않지만, Ad-hoc 통신망에서는 망의 혼잡으로 인한 패킷

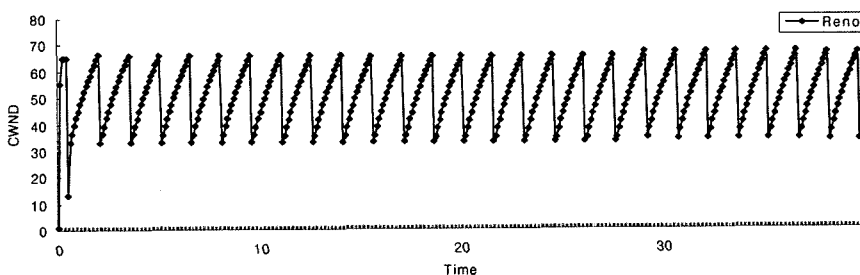


그림 1 TCP-Reno의 혼잡 윈도우의 변화

손실의 반복과 이를 복구하기 위한 반복적인 재전송에 의해 이동 단말의 에너지를 불필요하게 낭비하는 결과를 가지게 된다.

또한 기존 TCP는 유선망의 신뢰적인 링크를 기반으로 디자인된 프로토콜이기 때문에, Ad-hoc 통신망에서 비신뢰적인 무선 링크의 전송 에러로 발생하는 패킷 손실의 원인을 구별하지 못하고 혼잡에 의한 패킷 손실로 인식하는 문제점을 가지고 있다. Ad-hoc 통신망은 무선 링크에 의해 발생하는 잡음(Noise), 페이딩(Fading), 간섭(Interference) 등에 의한 높은 비트 에러율을 가짐으로써 전송 오류에 의한 패킷 손실이 발생할 수 있다. 하지만, 기존 TCP를 수정 없이 Ad-hoc 통신망에 적용하였을 경우, 기존 TCP는 패킷 손실의 원인을 구별하지 못하기 때문에 전송 에러에 의한 패킷 손실이 발생했을 때에도 망의 혼잡을 완화하기 위해 혼잡 제어를 수행함으로써 불필요하게 전송률을 감소하게 된다. 이러한 동작 방식으로 인해 Ad-hoc 통신망에서 기존 TCP는 제한된 대역폭을 효율적으로 사용하지 못하는 문제점을 갖게 된다.

2.2 TCP-Veno의 소개 및 문제점

TCP-Veno는 기본적으로 Reno의 동작 방식과 유사한 AIMD(Additive Increase Multiplicative Decrease) 방식으로 동작하지만, 2.1절에서 언급한 Ad-hoc 통신망에서 기존 TCP의 문제점을 해결하기 위해 추가적인 두 가지 알고리즘을 사용한다. 첫째로 망의 상태를 고려하지 않고 전송률을 증가시키는 수동적인 혼잡 제어 방식으로 인한 이동 단말의 에너지 효율 문제를 개선하기 위해 망의 상태를 고려하여 혼잡 윈도우의 증가폭을 조절하는 알고리즘이 있으며, 둘째로 패킷 손실의 발생 원인을 구별하는 알고리즘이 있다[9].

전송 초기 단계인 슬로우 스타트 단계에서의 TCP-Veno는 TCP-Reno와 같은 방식으로 지수적으로(Exponential) 혼잡 윈도우를 증가시키게 된다. 혼잡 윈도우의 크기가 슬로우 스타트 임계값에 이르게 되면, 혼잡 윈도우의 크기를 선형 증가 시키게 되는데, 이 때 그림 2의 TCP-Vegas 알고리즘을 사용하여 그림 3과 같이 혼잡 윈도우의 크기를 N값과 β값에 따라 점진적으로 증가시키게 된다. 그림 2의 TCP-Vegas 알고리즘에서 cwnd는 현재 혼잡 윈도우의 크기를 나타내며 Base-RTT는 연결 시간동안 측정된 최소 RTT를 의미한다.

TCP-Vegas는 각각 1과 3의 값을 갖는 a와 β값을

$$\begin{aligned}
 Expect &= cwnd / BaseRTT \\
 Actual &= cwnd / RTT \\
 N &= (Expect - Actual) * BaseRTT
 \end{aligned}$$

그림 2 TCP-Vegas의 알고리즘

이용하여 혼잡 윈도우의 크기를 조절한다. N값이 a보다 작으면 혼잡 윈도우의 크기를 1씩 증가시키며, N값이 a와 β값 사이에 있을 시 혼잡 윈도우의 크기를 고정하여 사용한다. 또한 N값이 β값보다 클 때, 혼잡 윈도우의 크기를 1씩 감소시킨다. 이와 같이 N값이 β값보다 클 때 혼잡 윈도우의 크기를 감소함으로써 망의 큐잉(Queuing) 되는 패킷의 수를 제한하게 된다. TCP-Vegas의 이러한 동작 방식은 N값이 β값보다 클 때, 망의 현재 상태를 혼잡 상태로 가정하기 때문이다. TCP-Vegas의 가정을 바탕으로 TCP-Veno는 그림 3과 같이 N값이 β값보다 큰 경우 혼잡 윈도우의 크기를 점진적으로 증가시킴으로써 망의 혼잡에 의한 패킷 손실을 방지하며 동작하게 된다.

```

// Additive Increase Phase
if(CWND ≥ ssthresh)
    if(N < β)
        CWND = CWND + 1
    else
        CWND = CWND + 1/2
    
```

그림 3 TCP-Veno의 선형 증가 알고리즘

TCP-Veno는 3개의 중복 ACK를 수신하거나 또는 Timeout의 발생에 의해 패킷 손실이 발생했음을 인지하게 된다. Timeout이 발생한 경우 TCP-Veno는 TCP-Reno와 똑같은 방식으로 손실된 패킷을 재전송하고 혼잡 윈도우의 크기를 1로 설정하여 슬로우 스타트하게 되며, 3개의 중복 ACK를 수신하면 그림 4와 같이 N값과 β값을 이용하여 패킷 손실의 원인을 파악 한 후 패킷 손실의 원인에 따라 혼잡 윈도우의 크기를 재설정하여 전송을 재개하게 된다. TCP-Veno는 N값과 β값의 비교로 패킷 손실의 원인을 판단하게 되는데, 판단 기준 또한 TCP-Vegas의 가정을 바탕으로 하고 있다. TCP-Veno는 패킷 손실에 의한 3개의 중복 ACK를 수신했을 때 N값이 β값보다 크면 혼잡에 의한 패킷 손실로 판단하게 되어 혼잡 윈도우의 크기를 반으로 줄임으로써 전송률을 감소시키며, N값이 β값보다 작으면 전송 오류에 의한 패킷 손실로 판단하게 되어 혼잡 윈도우의 크기를 거의 줄이지 않고 전송을 재개하게 된다.

```

// 3 Duplicative ACK received
if(N > β)
    CWND = CWND / 2 //Congestion Loss
else
    CWND = CWND * 0.8 //Non-congestion Loss
    
```

그림 4 TCP-Veno의 지수 감소 알고리즘

이와 같이 TCP-Veno는 TCP-Vegas 알고리즘을 이용하여 망의 상태를 모니터링 함으로써 망의 상태에 따라 전송률을 점진적으로 증가시키고, 패킷 손실의 원인을 파악하여 전송률을 조절함으로써 기존 TCP가 갖는 Ad-hoc 통신망의 문제점을 개선할 수 있다. 하지만, TCP-Veno는 망의 상태를 모니터링 하는데 있어서 TCP-Vegas의 알고리즘을 사용하기 때문에 TCP-Vegas의 문제인 BaseRTT를 정확히 측정할 수 없는 문제를 갖게 된다[10][11][12]. 따라서 망의 상태를 정확하게 측정하지 못하는 문제를 갖게 되며, 그에 따라 망의 대역폭 사용률이 떨어지는 문제점과 패킷 손실 원인 구별의 정확성이 떨어지는 문제를 갖게 된다. 또한 TCP-Vegas와 같이 BaseRTT에 의존함으로써 불공평성 문제도 갖게 된다.

3. TCP-New Veno

본 장에서는 2장에서 지적한 TCP-Veno의 문제점을 개선하기 위해 새롭게 제안한 TCP-New Veno의 알고리즘에 대해 기술한다. 제안한 TCP-New Veno는 TCP-Veno가 갖는 문제점을 해결하기 위해 망의 상태를 모니터링 하는 알고리즘을 추가한다. 추가된 알고리즘을 통해 망의 상태를 고려하지 않고 전송률을 증가시키는 수동적인 혼잡 제어 방식의 문제점을 개선하여 이동 단말의 에너지 효율을 향상시켰으며, 패킷 손실의 발생 원인을 구별함으로써 전송 에러에 의한 불필요한 혼잡 제어를 방지할 수 있다.

3.1 망 상태 모니터링 알고리즘

TCP-New Veno는 TCP-Veno가 갖는 BaseRTT의 문제를 보완하기 위하여 망의 상태를 모니터링 하는 알고리즘을 추가한다. 이 알고리즘을 통해 TCP-New Veno는 망의 상태를 안정, 혼잡 증가, 혼잡 감소로 나누어 현재의 망 상태를 구분하게 된다.

그림 5는 안정된 망 상태에서, 즉 중계 노드에서 큐잉 되는 패킷이 없이 송신자가 수신자로 n 패킷과 n+1 패킷을 중계 노드를 통해 각각 S_n 과 S_{n+1} 의 시간에 전송

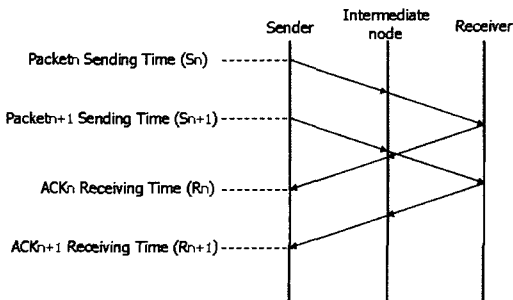


그림 5 안정된 망의 상태

하는 그림이다. n 패킷과 n+1 패킷의 전송에 대한 수신 확인으로 수신자는 각각 ACK_n 과 ACK_{n+1} 을 전송하며, 송신자는 두 ACK를 각각 R_n 과 R_{n+1} 의 시간에 수신하게 된다. 그림 5에서 볼 수 있듯이 중계 노드에서 큐잉 되는 시간 없기 때문에 n 패킷을 전송한 시간 S_n 과 n+1 패킷을 전송한 시간 S_{n+1} 의 간격은 ACK_n 을 수신한 시간 R_n 과 ACK_{n+1} 을 수신한 시간 R_{n+1} 의 간격과 같게 된다.

그림 6은 혼잡이 증가하는 망 상태에서 송신자가 수신자로 n 패킷과 n+1 패킷을 중계 노드를 통해 각각 S_n 과 S_{n+1} 의 시간에 전송하는 그림이다. 그림 6은 그림 5와 다르게 혼잡의 증가로 인한 중계 노드에서 패킷의 큐잉 지연 시간이 증가하기 때문에 n 패킷을 전송한 시간 S_n 과 n+1 패킷을 전송한 시간 S_{n+1} 의 간격은 ACK_n 을 수신한 시간 R_n 과 ACK_{n+1} 을 수신한 시간 R_{n+1} 의 간격보다 작게 된다.

그림 7은 혼잡이 감소하는 망 상태에서 송신자가 수신자로 n 패킷과 n+1 패킷을 중계 노드를 통해 각각 S_n 과 S_{n+1} 의 시간에 전송하는 그림이다. 그림 7은 혼잡의 감소로 인한 중계 노드에서 패킷의 큐잉 지연 시간이 감소하기 때문에 n 패킷을 전송한 시간 S_n 과 n+1 패킷을 전송한 시간 S_{n+1} 의 간격은 ACK_n 을 수신한 시간 R_n 과 ACK_{n+1} 을 수신한 시간 R_{n+1} 의 간격보다 크게 된다.

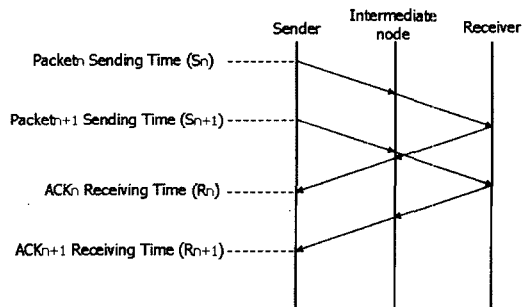


그림 6 혼잡이 증가하는 망의 상태

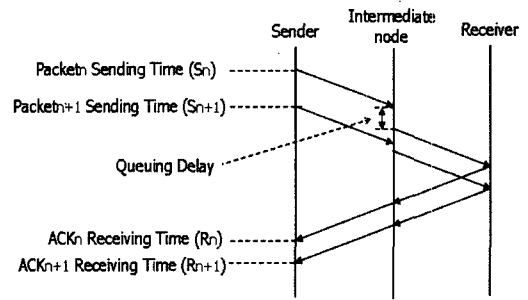


그림 7 혼잡이 감소하는 망의 상태

이와 같이 세 가지 그림을 바탕으로 TCP- New Veno는 망의 현재 상태를 모니터링 하기 위해 식 (1)의 Dq (Relative Queuing Delay)와 식 (2)의 ΔDq 을 추가한다.
 $Dq = ACK\ receiving\ interval - Packet\ sending\ interval$
 $= (R_n - R_{n-1}) - (S_n - S_{n-1})$ (1)

$$\Delta Dq = Dq_n - Dq_{n-1} \quad (2)$$

망이 안정된 상태에서는 패킷의 큐잉 지연 시간이 없기 때문에 ACK의 수신 시간 간격(ACK receiving interval)과 패킷의 전송 시간 간격(Packet sending interval)은 같게 된다. 따라서 Dq 는 0의 값을 가지게 된다. 또한 망의 혼잡이 증가하는 상태에서는 패킷의 큐잉 지연 시간이 증가하기 때문에 ACK의 수신 시간 간격은 패킷의 전송 시간 간격보다 커지므로, Dq 는 양의 값을 가지게 된다. 반대로 망의 혼잡이 감소하는 상태에서는 패킷의 큐잉 지연 시간이 감소하기 때문에 ACK의 수신 시간 간격은 패킷의 전송 시간 간격보다 작아지므로, Dq 는 음의 값을 가지게 된다. 또한 ΔDq 는 Dq 의 변화량을 나타냄으로써 혼잡 증가의 밀도(density)를 나타내게 된다. 이와 같이 식 (1)과 식 (2)를 이용하여 망의 상태를 그림 8과 같이 정리할 수 있다.

$Dq < 0$	// Congestion Decrease
$Dq = 0$	// Stable state
$Dq > 0$	// Congestion Increase
$\Delta Dq < 0$	// Congestion density Decrease
$\Delta Dq = 0$	// Stable state
$\Delta Dq > 0$	// Congestion density Increase

그림 8 Dq 와 ΔDq 에 따른 망의 상태

3.2 TCP-New Veno의 혼잡 제어 알고리즘

앞선 3.1절에서 설명한 망의 상태를 모니터링 하기 위한 알고리즘 Dq 와 ΔDq 를 설명하였다. 본 절에서는 Dq 와 ΔDq 를 이용하여 기존의 TCP-Veno를 보완한 TCP-New Veno의 혼잡 제어 알고리즘에 대해 설명한다.

기존의 TCP-Veno는 TCP-Vegas의 알고리즘을 이용한 N 값과 β 값의 비교를 통하여 망의 상태를 모니터링 하였다. 따라서 TCP-Veno는 TCP-Vegas가 갖는 BaseRTT를 정확히 측정할 수 없는 문제를 갖게 됨으로써 망의 상태를 정확하게 측정하지 못하는 문제를 갖게 된다. TCP-New Veno는 이러한 문제를 해결하기 위해 Dq 와 ΔDq 를 추가적으로 이용하여 망의 상태를 효율적으로 파악함으로써 그림 9와 같이 혼잡 윈도우의 크기를 조절한다.

```

If (N < β)
    Set cwnd = cwnd + 1
If (N ≥ β)
    If (Dq < 0)           Highly aggressive mode
        Set cwnd = cwnd + 1
    If (Dq = 0)         Aggressive mode
        If (ΔDq ≤ 0)
            set cwnd = cwnd + 1
        If (ΔDq > 0)
            set cwnd = cwnd + 1/2
    If (Dq > 0)         Conservative mode
        If (ΔDq ≤ 0)
            set cwnd = cwnd + 1/4
        If (ΔDq > 0)
            set cwnd = cwnd
    
```

그림 9 Dq 와 ΔDq 를 이용한 혼잡 윈도우 크기의 선형 증가

그림 9는 TCP-New Veno가 혼잡 윈도우의 크기를 선형 증가하는 그림으로써, 우선 N 값과 β 값을 비교하여 N 값이 β 값보다 작은 경우에는 TCP-Reno와 같이 혼잡 윈도우의 크기를 1씩 증가 시킨다. 또한 N 값이 β 값보다 큰 경우에는 Dq 와 ΔDq 를 이용하여 그림 9와 같이 단계적으로 혼잡 윈도우의 증가량을 나누어 조절한다.

첫 번째로 Dq 가 0보다 작을 때 TCP-Reno와 같이 혼잡 윈도우의 크기를 1씩 증가 시킨다. Dq 가 0보다 작다는 것은 패킷의 전송 시간 간격보다 ACK의 수신 시간 간격이 작다는 것으로써 망의 혼잡이 감소하는 상태를 의미한다. 이는 곧 잔여 대역폭의 증가를 의미하므로 망의 사용률을 올리기 위해 전송률을 1씩 증가 시킨다.

두 번째로 Dq 가 0과 같을 때 혼잡 윈도우의 증가량을 ΔDq 에 의해 결정하게 된다. Dq 가 0인 것은 패킷의 전송 시간 간격과 ACK의 수신 시간 간격이 같다는 것으로써 망의 상태가 안정된 상태를 의미하게 된다. 이때, 과거의 혼잡 증가량과 현재의 혼잡 증가량의 차를 나타내는 즉 혼잡의 증가 밀도를 나타내는 ΔDq 에 의해 혼잡 윈도우의 증가량을 조절하게 된다. ΔDq 가 0보다 같거나 작으면 혼잡의 증가폭이 감소 또는 일정한 상태를 의미하기 때문에 전송률을 증가 시키며, ΔDq 가 0보다 클 때, 혼잡의 증가폭이 증가하지만, TCP-Reno의 경우 이러한 상태 내에서도 전송률을 증가시키기 때문에 TCP-New Veno는 다른 플로우와의 경쟁력을 고려하여 전송률을 점진적으로 증가시킨다.

세 번째로 Dq 가 0보다 큰 경우는 패킷의 전송 시간 간격보다 ACK의 수신 시간 간격이 큰 것으로써 망의 혼잡이 증가하는 상태를 의미하게 된다. 따라서 TCP-New Veno는 작은 패킷 손실에 의한 재전송을 방지하기 위해 전송률을 ΔDq 에 따라 보수적으로 혼잡 윈도우의 크기를 조절한다.

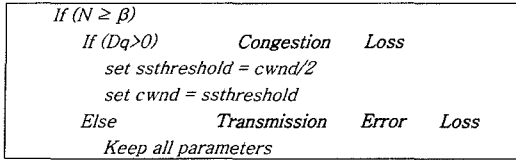


그림 10 Dq와 ΔDq를 이용한 혼잡 윈도우 크기의 지수 감소

그림 10은 패킷 손실에 의해 세 개의 중복 ACK를 수신했을 때 TCP-New Venos의 동작 방식을 나타낸 그림이다. TCP-New Venos는 혼잡 윈도우의 크기를 선형 증가시키는 단계에서 세 개의 중복 ACK를 수신하게 되면, 망의 현재 상태를 모니터링 하는 Dq와 ΔDq를 이용하여 현재의 패킷 손실의 원인을 구별하고, 패킷 손실의 발생 원인에 따라 그림 10과 같이 혼잡 윈도우의 크기를 조절한다. TCP-New Venos는 N값이 β값보다 크고, Dq가 0보다 큰 경우에 망의 상태가 혼잡으로 인식하기 때문에 이와 같은 구간에서는 TCP-Reno와 같이 혼잡 윈도우의 크기를 줄임으로써 혼잡 제어를 수행한다. 그 외의 구간에서는 모니터링 된 망이 혼잡 상태가 아니므로 전송 에러에 의한 패킷 손실로 인식을 하고 전송률을 고정하여 전송을 재개한다.

이와 같이 TCP-New Venos는 기본적으로 AIMD 방식으로 동작을 하며, 추가적으로 Dq와 ΔDq를 이용하여 망의 상태를 파악하고 그에 따라 전송률을 적절하게 조절함으로써 망의 혼잡에 의한 잦은 패킷 손실을 방지하며, 또한 패킷 손실의 원인을 구별하여 불필요한 혼잡 제어를 방지할 수 있다. 이러한 혼잡 제어 방식을 통해 TCP-New Venos는 기존의 TCP-Venos의 문제점을 개선함으로써 이동 단말의 에너지 효율 증가 및 망의 사용률을 증가시킬 수 있다. 또한 BaseRTT에 대한 의존성을 낮춤으로써 기존의 TCP-Venos가 갖는 불공평성 문제도 해결 할 수 있다.

4. 실험 및 성능 평가

본 장에서는 새로 제안한 TCP-New Venos의 성능 평가를 위해 전송 에러에 의해 패킷 손실이 발생하는 Ad-hoc 통신망에서 이동 단말의 에너지 효율 및 전송 성능 실험과 혼잡에 의해 패킷 손실이 발생하는 유선망에서 공평성 실험을 LBNL(Lawrence Berkely National Laboratory)의 ns-2(network simulator)를 사용하여 기존의 TCP-Reno, TCP-Venos와의 성능을 비교하였다[13].

4.1 Ad-hoc 통신망에서의 성능 실험

Ad-hoc 통신망은 유선망과 다르게 비선형적인 전송 매체를 사용함으로써 전송 에러에 의한 패킷 손실이 빈

번히 발생한다. 따라서 본 절에서는 전송 에러에 의해 패킷 손실이 발생하는 Ad-hoc 통신망에서 TCP-New Venos의 성능 평가를 위해 TCP-Reno, TCP-Venos의 전송 성능 및 이동 단말의 에너지 효율을 비교하였으며, 결과를 통해 TCP-New Venos가 기존의 연구에 비해 Ad-hoc 통신망에서 더 효율적으로 동작한다는 것을 확인하였다.

4.1.1 실험 환경

Ad-hoc 통신망에서 제안한 TCP-New Venos의 성능을 평가하기 위해 그림 11과 같은 실험 환경을 구성하여 성능 실험을 수행하였다. Ad-hoc 통신망에서 발생하는 전송 에러에 따른 이동 단말의 에너지 효율 및 전송 성능을 평가하기 위하여 이동 단말의 이동성을 배제한 환경으로 2Mbps 대역폭에 3ms의 전송 지연을 갖는 IEEE 802.11b MAC(Medium Access Control)을 사용한 무선 링크를 설정하였으며, 라우팅 프로토콜로는 AODV(Ad-hoc On-Demand Distance Vector)를 사용하였다[14,15]. 이동 단말의 전송 범위는 200m로 제한함으로써 송신자(S0)가 수신자(S2)로 직접 전송하지 않고 중계 노드(S1)를 통하여 전송 할 수 있도록 설정하였으며, 패킷 손실률에 따라 패킷을 강제적으로 버리는 에러 발생기(Error Generator)를 중계 노드에 설정함으로써 전송 에러에 의한 패킷 손실 환경을 만들었다. 또한 이동 단말의 초기 에너지는 100J(Joule)로 설정하였으며, 각각 패킷 전송 시 0.6W(Watt)를 소비하며 ACK 수신시 0.3W(Watt)를 소비하도록 설정하였다.

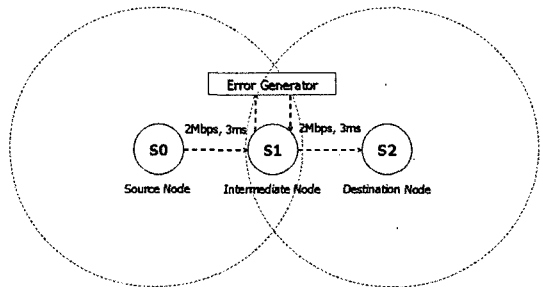


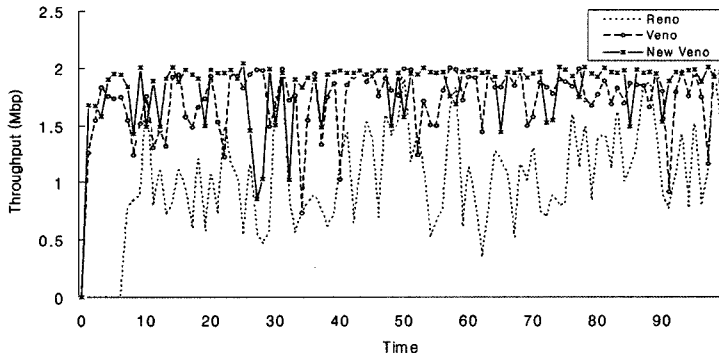
그림 11 실험 환경

4.1.2 전송 에러에 의한 패킷 손실률에 따른 성능 및 에너지 효율 실험

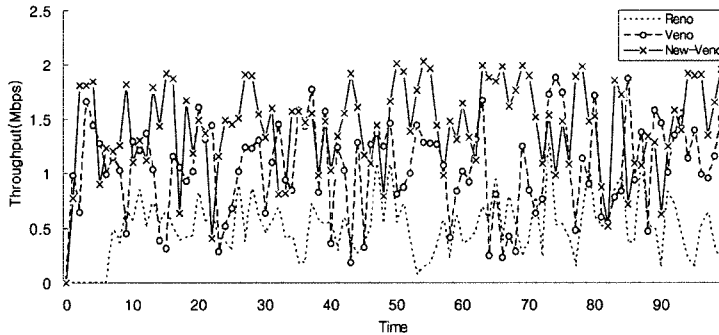
그림 12는 전송 에러에 의해 패킷 손실이 발생했을 때, 패킷 손실률에 따른 TCP-Reno, TCP-Venos, TCP-New Venos의 성능을 나타내는 그림이다. 그림 12(a)는 패킷 손실률이 2%일 때 각 프로토콜의 전송 성능을 나타낸 그림으로써 TCP-New Venos가 TCP-

Reno 보다 좋은 성능을 보이며, TCP-Veno의 성능과는 비슷한 것을 확인할 수 있다. 하지만, 그림 12(b)에서 볼 수 있듯이 패킷 손실률이 증가하게 되면, TCP-New Veno가 TCP-Veno보다 성능이 뛰어난 것을 확인할 수 있다. 그림 12(c)와 같이 패킷 손실률이 0~10%인 구간에서 패킷 손실률이 증가할수록 TCP-New Veno의 평

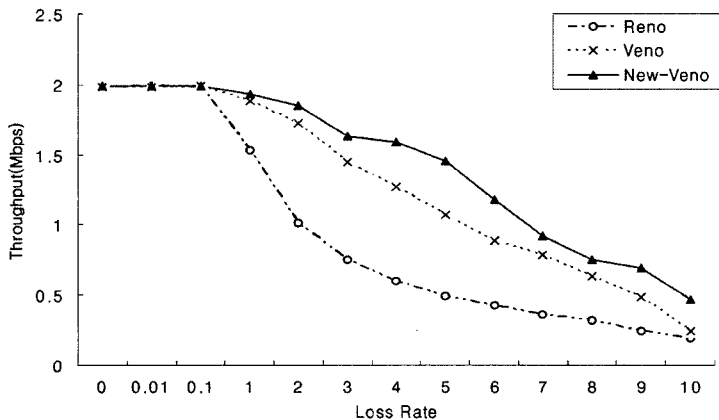
균 전송률이 더 높은 것을 확인할 수 있다. TCP-Reno의 성능이 가장 낮은 이유는 혼잡이 아닌 전송 에러에 의한 패킷 손실이 발생했을 때, 패킷 손실의 원인을 구별할 수 있는 알고리즘이 없기 때문에 불필요한 혼잡 제어를 수행하여 전송률을 낮춤으로써 가용 대역폭을 효율적으로 사용하지 못하게 되기 때문이다. 또한



(a) 패킷 손실률 2%에서의 전송률



(b) 패킷 손실률 5%에서의 전송률



(c) 패킷 손실률에 따른 평균 전송률

그림 12 전송 에러에 의한 패킷 손실률에 따른 성능

TCP-Veno의 성능 저하 원인은 TCP-Veno가 N값과 β값의 비고를 통해 패킷 손실의 원인을 구별하긴 하지만, 2.2절에서 설명한 것과 같이 BaseRTT의 문제로 인해 망의 상황을 파악하는데 한계 있기 때문이다. TCP-New Veno는 N값과 β값의 비교 이외에 Dq와 ΔDq를 이용함으로써 망의 상태를 더 정확히 판단할 수 있으며, 또한 패킷 손실의 원인을 더 정확히 구별함으로써 비교 대상의 프로토콜보다 가용 대역폭을 더 효율적으로 사용할 수 있도록 개선되었다는 것을 확인할 수 있다. 하지만, 패킷 손실률이 5%이상으로 증가하게 되면 패킷 손실로 인한 중복 ACK의 발생 빈도뿐만 아니라 Timeout의 발생 빈도 역시 증가하기 때문에 Timeout으로 인한 혼잡 제어의 빈도가 증가하여 모든 프로토콜의 성능이 급격히 저하된다.

$$Energy\ Efficiency\ (\eta) = \frac{Throughput}{Consumed\ Energy} (Kb/sJ) \tag{3}$$

각 프로토콜의 에너지 효율 실험을 위해 그림 11의 실험 환경에서 0~10%의 패킷 손실률에 따라 150초 동안 수행하였다. 식 (3)은 송신자가 전송한 데이터의 양과 그에 의해 소비된 에너지를 통해 이동 단말의 에너지 효율을 나타내는 식으로써 일정 에너지로 서비스할 수 있는 데이터의 양을 나타내게 된다. 이 식을 바탕으로 TCP-Reno, TCP-Veno 그리고 TCP-New Veno의 에너지 효율을 평가 한다.

그림 13는 TCP-Reno, TCP-Veno 그리고 TCP-New Veno의 패킷 손실률에 따른 이동 단말의 에너지 효율을 나타내는 결과이다. TCP-New Veno는 0~10%의 패킷 손실률 전체에서 TCP-Veno와 TCP-Reno보다 좋은 성능을 보이게 된다. 또한 평균 에너지 효율은 TCP-New Veno의 경우 약 32Kb/sJ, TCP-Veno의 경우 약 27Kb/sJ이며, TCP-Reno의 경우 약 17Kb/sJ을 나타낸다. 즉, TCP-New Veno는 TCP-Reno보다 약 2

배, TCP-Veno보다는 약 1.3배의 성능이 향상되었음을 확인할 수 있다. 패킷의 손실률이 5%일 때, TCP-Reno, TCP-Veno, TCP-New Veno의 전송률은 각각 약 0.5Mbps, 1.3Mbps, 1.6Mbps이며, 이 때 소비된 에너지의 양은 각각 약 41J, 44J, 43J이다. TCP-Reno의 경우 소비된 에너지가 작지만 전송한 데이터의 양이 작기 때문에 에너지 효율이 낮게 되는데 이는 곧 재전송에 의한 에너지 소비가 많았음을 의미하게 된다. TCP-Veno와 TCP-New Veno는 비슷한 에너지를 소비했지만 전송한 데이터의 양이 TCP-New Veno가 많기 때문에 TCP-New Veno가 더 좋은 에너지 효율을 보이게 된다. 이러한 결과는 TCP-New Veno가 Dq와 ΔDq를 이용하여 망의 상태를 모니터링 하고 전송률을 적절히 조절하기 때문이다. 주목할 점은 전송 에러에 의한 패킷 손실이 발생하지 않는 패킷 손실률 0%에서도 TCP-New Veno는 작게나마 더 좋은 에너지 효율을 보인다는 것이다. 이는 TCP-New Veno가 망의 상태를 적절히 파악하고 망이 혼잡 상황일 때, 전송률을 유지하거나 천천히 증가시킴으로써 기존의 TCP-Reno의 문제점인 잦은 패킷 손실에 의한 재전송을 방지하기 때문이다.

이와 같이 Ad-hoc 통신망에서 이동 단말의 에너지 효율은 시간당 전송한 데이터의 양과 이 때 전송에 의해 소비된 에너지에 의해서 결정되게 된다. 시간당 전송한 데이터의 양이 많더라도 그 데이터를 전송하는 에너지의 소비가 많다면 상대적으로 에너지 효율이 낮게 되며, 시간당 전송한 데이터의 양이 적더라도 에너지의 소비가 적다면 상대적으로 에너지 효율이 높을 수도 있다. 이와 같은 관점에서 TCP-New Veno는 망의 상태가 안정적일 때 전송률을 크게 증가시키고, 망의 상태가 혼잡할 때 전송률을 유지 또는 점진적으로 증가시키며 동작한다. 그럼으로써 불필요한 패킷 손실에 의한 재전송을 방지할 수 있기 때문에 일정 데이터를 전송하는데

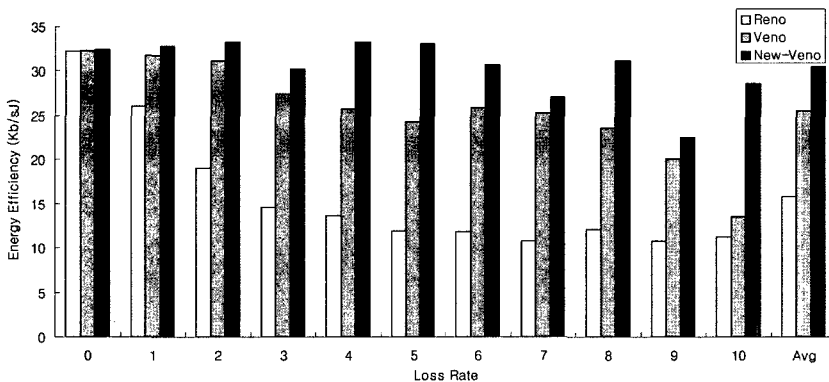


그림 13 패킷 손실률에 따른 에너지 효율

있어서 소비되는 에너지의 양을 줄일 수 있었다. 또한 패킷 손실이 발생했을 때 패킷 손실의 원인을 구별하여 그 원인에 따라 혼잡 제어를 수행함으로써 불필요한 전송을 저하를 방지하기 때문에 일정 시간에 많은 데이터를 전송할 수 있다. 즉 TCP-New Veno는 망의 상태를 파악하여 전송률을 조절함으로써 높은 전송률을 유지하고 전송 에너지를 적게 소비하기 때문에 이동 단말의 에너지 효율 측면에서도 높은 성능을 나타내게 된다.

4.2 유선망에서의 공평성 실험

본 실험은 TCP-New Veno의 공평성을 위한 것으로, 기존의 TCP-Veno가 갖는 불공정성 문제를 TCP-New Veno는 Dq와 ΔDq를 사용함으로써 개선할 수 있음을 보이기 위한 것이다.

4.2.1 실험 환경

그림 14와 같이 유선망을 사용하여 S1~S10의 송신자와 D1~D10의 수신자로 망을 구성하였다. S1의 송신자는 D1의 수신자로, S2의 송신자는 D2의 수신자로 데이터를 전송한다. 이러한 방법으로 각각 송신자는 R0과 R1을 통해 각각의 수신자로 데이터를 전송하며, 이때에 사용되는 프로토콜은 실험에 따라 5쌍의 송수신자로 2개의 그룹으로 나뉘어 실험한다. 또한, 4.1절의 실험과는 다르게 공평성 실험을 위해 전송 에러에 의한 패킷 손실 발생은 배제하였다.

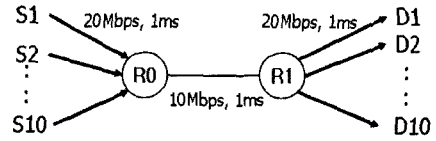
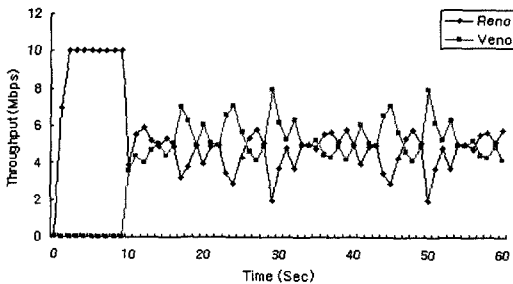


그림 14 공평성 실험을 위한 실험 환경

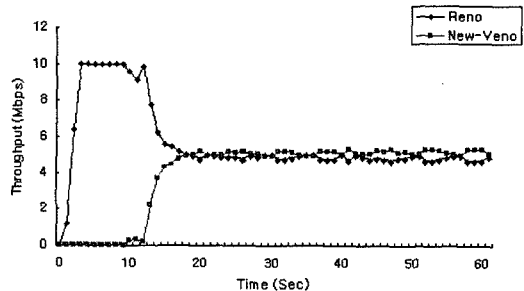
4.2.2 TCP-New Veno의 공평성 실험

그림 15는 TCP-New Veno와 TCP-Reno 플로우의 공평성에 대한 실험 결과로써, 그림 15(a)에서 확인할 수 있듯이 기존 TCP-Veno는 TCP-Reno 플로우와의 공평성 문제를 갖게 된다. 하지만, TCP-New Veno는 BaseRTT에 의존하지 않음으로써 그림 15(b)와 같이 TCP-Reno 플로우와의 불공평성 문제가 개선되었음을 확인할 수 있다.

그림 16(a)와 같이 RTT가 서로 다른 두 TCP-Veno 플로우 간의 경쟁에서 불공평성 문제가 발생한다는 것을 확인할 수 있다. TCP-Veno는 N값이 β값보다 클 경우 전송률을 점진적으로 조절하게 되기 때문에 N값을 구성하는 BaseRTT에 의해 혼잡 윈도우의 크기를 적극 또는 소극적으로 전송하는 시점이 틀리게 되고, 패킷 손실의 원인을 구별하는 구간이 서로 다르기 때문에 상대적으로 RTT가 긴 플로우가 망의 점유하는 결과를 갖게 된다. 하지만, TCP-New Veno의 경우 그림 16(b)

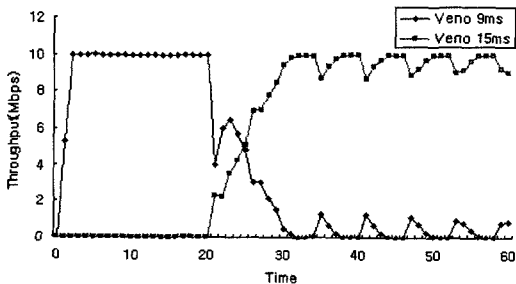


(a) TCP-Reno vs. TCP-Veno

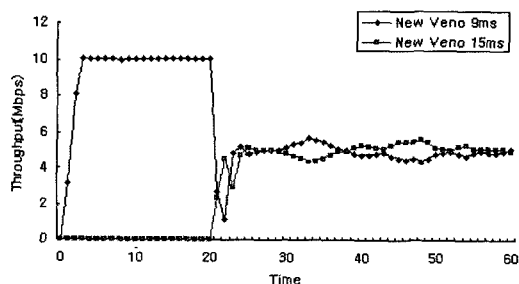


(b) TCP-Reno vs. TCP-New Veno

그림 15 TCP-Reno와 TCP-Veno, TCP-New Veno 플로우 간의 공평성 실험



(a) TCP-Veno



(b) TCP-New Veno

그림 16 RTT가 서로 다른 두 플로우 간의 공평성 실험

와 같이 RTT가 서로 다른 두 플로우간의 공평성 문제가 개선되었음을 확인 할 수 있다.

5. 결론 및 향후 과제

Ad-hoc 통신망은 비신뢰적인 무선 채널을 사용함으로써 무선 채널에 의한 높은 비트 에러율을 가지며, Ad-hoc 통신망을 구성하는 이동 단말의 에너지 자원은 한정적인 특징을 갖는다. 기존 TCP는 유선망을 기반으로 디자인되어, Ad-hoc 통신망의 특성으로 인해 발생하는 에러를 구별하지 못하는 문제를 가지며, 이동 단말의 제한된 에너지 자원을 고려하지 않고 동작하는 문제가 있다. 이러한 문제를 해결하기 위해 TCP-Veno는 TCP-Vegas 알고리즘을 사용하여 망의 상태를 파악하고, 망의 상태에 따라 전송률의 증가폭을 조절하여 기존 TCP의 불필요한 패킷 손실 및 재전송의 반복을 방지하며, 패킷 손실이 발생했을 때 패킷 손실의 원인을 구별함으로써 패킷 손실의 원인에 따라 적절히 혼잡 제어를 수행한다. 하지만, TCP-Veno는 망의 상태를 모니터링 하는데 있어서 TCP-Vegas의 알고리즘을 그대로 사용하기 때문에 TCP-Vegas의 BaseRTT 문제를 갖게 되고, 망의 상태를 정확히 파악할 수 없는 문제가 발생한다.

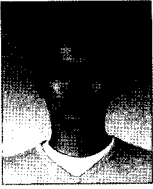
따라서 본 논문에서는 Ad-hoc 통신망과 관련된 기존 연구들의 한계를 보완하기 위한 TCP-New Veno를 제안하였다. TCP-New Veno는 기본적으로 AIMD 방식으로 TCP-Veno와 유사하게 동작을 하며, TCP-Vegas 알고리즘 이외에 추가적으로 Dq 와 ΔDq 를 이용하여 망의 상태를 파악한다. 추가된 알고리즘에 따라 전송률을 적절하게 조절함으로써 망의 혼잡에 의한 잦은 패킷 손실을 방지하며, 패킷 손실이 발생하였을 경우 패킷 손실의 원인을 구별하여 불필요한 혼잡 제어를 방지할 수 있다. 이러한 혼잡 제어 방식을 통해 TCP-New Veno는 이동 단말의 높은 에너지 효율을 가지며, 망의 사용을 또한 증가시킬 수 있었다. 또한 BaseRTT에 대한 의존성을 낮춤으로써 기존의 TCP-Veno가 갖는 불공평성 문제도 해결 할 수 있다.

시뮬레이터를 이용한 실험을 통해서 Ad-hoc 통신망 및 유선망에서 TCP-New Veno의 성능을 검증하고, 관련 연구와의 성능 비교를 위해 이동 단말의 에너지 효율 및 전송 성능, 공평성에 대해 비교 분석하였다. 실험 결과를 통해 제안한 TCP-New Veno는 전송 에러에 의해 패킷 손실이 발생하는 Ad-hoc 통신망에서 높은 에너지 효율성을 가지며, 높은 전송 성능을 확인할 수 있었다. 향후, 유선망과 다른 Ad-hoc 통신망의 또 다른 특성인 이동 단말의 이동성으로 인한 경로 변경 및 연결 단절과 같은 특성을 반영하기 위한 방법의 연구가 수행되어야 하고, 따라서 이동성이 잦은 환경에서의 실험

이 수행되어야 할 것이다.

참고 문헌

- [1] Mobile Ad-hoc Networks Discussion Archive-Data, <http://www1.ietf.org/mail-archive/working-groups/manet/current/maillist.html>
- [2] H. Balakrishnan, V. N. Padmanabhan, S. Seshan and R. H. Katz, "A Comparison of Mechanisms of Improving TCP Performance over Wireless Links," *Proceeding of ACM Transactions On Networking*, pp.756-769, December 1997.
- [3] C. Jone, K. M. Sivalingam, P. Agrawal and J. C. Chen, "A Survey of Energy Efficient Network Protocol for Wireless Networks," *Proceeding of ACM Computer Communication Review*, pp. 14-43, October 1997.
- [4] H. Singh, S. Saena and S. Singh, "Energy Consumption of TCP in Ad Hoc Networks," *Wireless Networks*, pp. 531-542, September 2004.
- [5] H. Balakrishnan, S. Seshan, E. Amir and R.H. Katz, "Improving TCP/IP Performance over Wireless Networks," *Proceeding of ACM MOBICOM'95*, November 1995.
- [6] G. Holland and N. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks," *Proceeding of ACM MOBICOM'99*, August 1999.
- [7] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," RFC-2001, January. 1997.
- [8] V. Jacobson and M. Karels, "Congestion Avoidance and Control," *Proceeding of ACM SIGCOMM'88*, pp. 314-319, August 1988.
- [9] 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.
- [10] L. Brakmo and L. Peterson, "TCP Vegas: End-to-end Congestion Avoidance on a Global Internet," *IEEE Communication*, p. 1465-1480, October 1995.
- [11] T. Henderson and E. Sahouria, "On Improving the Fairness of TCP Congestion Avoidance," *Proceeding of IEEE GLOBECOM'99*, pp.539-544, November 1998.
- [12] W.Feng and S. Vanichpun, "Enabling Compatibility Between TCP Reno and TCP Vegas," *Proceedings of the Symposium on Applications and the Internet'2003*, p.301, January 2003.
- [13] The network simulator ns-2, <http://www.isi.edu/nanam/ns/>
- [14] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," *Proceeding of IEEE Workshop on Mobile Computing Systems and Applications*, February 1999.
- [15] M. Gast, "802.11 Wireless Networks: The Definitive Guide," O'REILLY, April 2002.



조 남 호

2003년 명지대학교 전자공학부 학사
2006년 광운대학교 전자통신공학과 석사
관심분야는 인터넷 QoS, 유비쿼터스 컴
퓨팅, Ad-hoc & Wireless 통신망



정 광 수

1981년 한양대학교 전자공학과 학사. 1983
년 한국과학기술원 전기 및 전자공학과
석사. 1991년 미국 University of Florida
전기공학과 박사(컴퓨터공학전공). 1983
년~1993년 한국전자통신연구원 선임연
구원. 1991년~1992년 한국과학기술원
대우 교수. 1993년~현재 광운대학교 전자공학부 교수(정보
통신 연구원)