
무선 메쉬 네트워크에서 TCP 성능 향상 기법

이혜림* · 문일영**

A TCP Performance Enhancement Scheme in Wireless Mesh Networks

Hye-Rim Lee* · Il-Young Moon**

요 약

무선 메쉬 네트워크는 무선 인프라 환경에서 고정이나 이동 중에 모든 노드가 통신을 할 수 있는 다중 경로를 가지는 새로운 네트워크이다. 하지만 무선 메쉬 네트워크는 유선망과 달리 무선 매체를 사용하기 때문에 경로손실, 간섭, 핸드오프 등으로 높은 패킷 손실률을 보이고 TCP(Transport Control Protocol) 알고리즘은 무선 메쉬 네트워크의 패킷손실 원인을 네트워크 내의 혼잡으로 인식하기 때문에 TCP 혼잡제어 알고리즘(Congestion Control Algorithm)을 실행하게 된다. 따라서 본 논문에서는 무선 메쉬 네트워크 환경에 적응력을 가지도록 기존의 TCP 혼잡제어 알고리즘의 혼잡 윈도우 값을 노드의 이동에 따라 유연하게 조절하는 새로운 TCP 혼잡제어 알고리즘을 제안한다.

ABSTRACT

Wireless mesh network is similar to ad-hoc network, so when transferred to the data packet in the wireless environment, interfered factor arise. When TCP(Transport Control Protocol) was created, however as it was design based on wired link, wireless link made more transmission error than wired link. It is existent problem that TCP unfairness and congestion collapse over wireless mesh network. But packet losses due to transmission errors are more frequent. The cause of transmission error in wireless ad-hoc network may be inexactly regarded as indications of network congestion. And then, Congestion Control Algorithm was running by this situation causes the TCP performance degradation. In this paper, proposed TCP can adaptively regulate the congestion window through moving node in the Wireless Mesh Network. And it enhanced the performance.

키워드

무선 메쉬 네트워크, TCP, 혼잡제어, 혼잡 윈도우, 모바일 노드

Key word

Wireless Mesh Network, TCP, TCP Congestion Control Algorithm, Congestion Window, Mobile Node

* 한국기술교육대학교 인터넷미디어공학부

접수일자 : 2010. 01. 25

** 한국기술교육대학교 인터넷미디어공학부 (iymoon@kut.ac.kr)

심사완료일자 : 2010. 02. 03

I. 서 론

무선인프라 환경에서 고정이나 이동 중에 기지국 대 기지국, 단말 대 단말, 기지국 대 단말 등 모든 노드가 통신을 할 수 있는 다중경로를 가지는 ‘메쉬형 토폴로지(Mesh topology)’를 무선망에 도입한 기술이다. 무선 메쉬 네트워크는 와이파이 기술과 접목하면서 광역 무선 LAN 구축을 더욱 쉽고 경제적으로 할 수 있게 됐다.

애드혹 통신 기술로부터 유래한 무선 메쉬 네트워크는 유선망과 달리 무선 매체를 사용하기 때문에 경로손실, 페이딩, 노이즈, 간섭, 핸드오프, Hidden 터미널 문제 등으로 높은 BER(Bit Error Rate)를 가지며, 노드들의 이동성으로 인해 동적인 토폴로지를 구성한다[1]. 하지만 유선 링크를 기반으로 설계된 TCP 알고리즘은 패킷의 모든 손실 원인을 네트워크 내의 혼잡으로 인식하기 때문에 패킷 유실이나 경로 이탈 등과 같은 전송에러의 한 손실도 혼잡 손실로 인식하여 TCP 혼잡제어 알고리즘을 실행하게 된다[1-3]. 이러한 TCP 동작은 혼잡 손실이 아닌 패킷 손실로 발생 할 경우, 상당한 성능 저하를 초래하게 된다[4].

이에 본 논문에서는, 유비쿼터스의 새로운 인프라로 주목받고 있는 무선 메쉬 네트워크에서 TCP 혼잡제어 알고리즘의 성능을 향상시키기 위해 혼잡 윈도우를 무선망에서도 적응력 있게 조절하는 연구를 하였다.

제 2장에서는 무선 메쉬 네트워크와 TCP 혼잡제어 알고리즘에 관한 배경 및 관련연구를 조사하였고, 제 3장에서는 TCP 혼잡제어 알고리즘의 성능개선을 위해 무선 메쉬 네트워크에서 TCP 혼잡제어 알고리즘이 가지는 문제점을 지적하고 이를 개선하는 TCP 혼잡제어 알고리즘을 제안한다. 그리고 제 4장에서는 제안된 TCP 혼잡제어 알고리즘의 성능평가를 위해 기존의 TCP 알고리즘인 TCP Reno와 TCP New-Reno와 비교하여 시뮬레이션 결과분석을 하였고 이를 통해 기존의 TCP 알고리즘보다 제안된 TCP 혼잡제어 알고리즘이 훨씬 좋은 성능을 나타냈다는 것을 보인다. 마지막으로 제 5장에서는 본 논문의 결론과 향후 연구이슈에 대해 기술한다.

II. 관련 연구

2-1 무선 메쉬 네트워크 기술

무선 메쉬 네트워크는 최소한의 이동성을 유지하면서 무선 메쉬 네트워크의 백본 네트워크를 구성하는 메쉬 라우터와 이 메쉬 라우터에 접속하는 메쉬 클라이언트 두 가지 형태의 노드로 구성되어 있다. 메쉬 클라이언트는 기존의 클라이언트와 유사한 기능을 가지고 있지만 메쉬 라우터는 기존의 라우터와 큰 차이가 있다.

표 1. 메쉬 라우터의 기능 및 특징
Table 1 Character and Function of Mesh Router

메쉬 라우터 기능	메쉬 라우터의 특징
<ul style="list-style-type: none"> - 백본을 형성하는 주체 - 움직임이 거의 없음 - 메쉬 라우터 사이의 네트워크 접근 - 메쉬 라우터 <-> 메쉬 클라이언트 접근 	<ul style="list-style-type: none"> - 자가 망 구성, 자가 망 치유 - Multi-hop 라우팅 - 기존의 AP(Access Point)에 비하여 많은 기능이 요구됨 - 다른 네트워크와의 결합을 통해 또 다른 무선 메쉬 네트워크 완성

무선 메쉬 네트워크는 미국 군사기술을 민간용으로 전환한 것에서 시작하였으며, 무선 메쉬 네트워크 기능을 탑재한 무선 LAN 액세스 포인트는 전원연결만 되면 네트워킹이 가능함으로 설치가 편리하고, 유선망과의 연결 없이 네트워크 망의 확장이 용이하고, 이동성을 가진 클라이언트 노드들로 인해 동적인 토폴로지를 구성한다.

무선 액세스 포인트를 통해 각종 단말기를 연결하는 기존의 무선 네트워크는 수신 범위를 벗어나면 연결이 끊어지는 데 비해 무선 메쉬 네트워크는 그림 1과 같이 각 단말기들이 그물망처럼 연결되므로 네트워크를 혁신적으로 확장할 수 있다. 따라서 무선 메쉬 네트워크는 액세스 포인트가 커버해야 할 커버리지가 비교적 많은 광범위한 지역에서 인터넷 망을 사용할 때 효율적으로 커버리지를 확장하고 액세스 포인트의 설치비용을 절감하는 등의 효과가 있다.

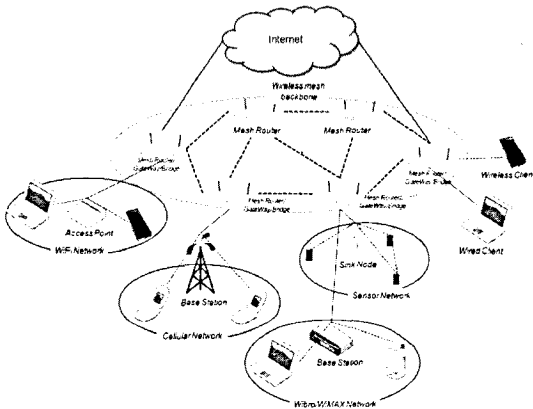


그림 1. 인프라스트럭처/백본 무선 메쉬 네트워크의 구조도

Fig. 1. Infrastructure/Backbone in Wireless Mesh Network Structure

2-2 TCP 혼잡제어 알고리즘

IP는 단순히 패킷만 전달하려는 목적으로 만들어졌기 때문에 TCP는 이러한 IP에 데이터 패킷을 안전하게 전달할 수 있도록 네트워크상의 여러 가지 변수들을 고려하여 최대한 목적지까지 정확히 전송되도록 유지시켜준다. 여러 가지 변수들 중에서 가장 TCP의 성능을 좌지우지 하는 것은 혼잡 윈도우이다. 혼잡 윈도우는 송신자가 데이터 패킷을 보낼 수 있는 크기를 말하며, 데이터 패킷은 혼잡 윈도우 사이즈에 맞춰 데이터 패킷을 전송한다. 이러한 혼잡 윈도우는 TCP의 혼잡제어 알고리즘에서 조절하게 된다. 즉, TCP 혼잡제어(congestion control) 알고리즘은 네트워크로 들어가는 정보 소통량을 조절하여 네트워크가 혼잡해지지 않게 조절하는 것을 말한다. 예를 들어, 정보 소통량이 과다한 것을 감지하여 패킷을 적게 보내면 혼잡 붕괴 현상이 일어나는 것을 막을 수 있다[5].

TCP는 흐름제어(Flow Control)와 혼잡제어 두개의 제어 메커니즘을 이용하여 End-to-End 간의 신뢰성 있는 전송을 보장한다. 흐름제어는 송신자가 수신자로부터 슬라이딩윈도우 크기를 받은 후 그것보다는 적게 보냄으로써 네트워크상의 흐름을 시간이나 크기단위로 조절하는 방법이고 혼잡제어는 송신자가 네트워크 상황을 보고 스스로 패킷을 조절하는 방법이다. 라우터가 부하를 감당할 수 없게 되어 심각한 지연이나 패킷이 손실

되는 경우를 혼잡이라고 한다. 빠른 LAN에서 느린 WAN으로 전송하거나 다수의 입력 스트림이 출력 수용량이 입력보다 적은 라우터에 도착했을 때 혼잡이 발생하는데 이런 문제를 해결하기 위한 방법을 혼잡제어라고 한다[6].

혼잡 윈도우는 송신자가 데이터 패킷을 보낼 수 있는 크기를 말하고, TCP는 혼잡 윈도우 값을 사용하여 송신단에서 전송하려는 패킷의 크기를 조절하여 송신측은 혼잡 윈도우 사이즈에 맞춰 데이터 패킷을 전송한다. 이러한 혼잡 윈도우는 TCP의 혼잡제어 알고리즘에서 조절하게 된다.

이러한 TCP 혼잡제어 알고리즘은 슬로우 스타트(Slow Start), 혼잡회피(Congestion Avoidance), 빠른 재전송(Fast Retransmission), 빠른 복구(Fast Recovery)인 4가지의 메커니즘으로 이루어져 있다.

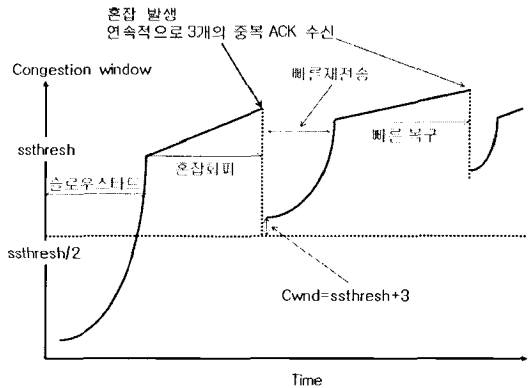


그림 2. TCP 혼잡제어의 메커니즘
Fig. 2. Mechanism of TCP Congestion Control

III. 무선 메쉬 네트워크에서 개선된 TCP 혼잡제어 알고리즘

3-1 무선 메쉬 네트워크에서 TCP 혼잡제어 알고리즘의 문제점

유선 네트워크는 물리적인 선을 통해 서로 연결되어 있기 때문에 패킷이 중간에 유실될 일도, 경로가 중간에 바뀌어 패킷이 손실될 일도 없다. 하지만 다수의 패킷들이 오고가는 라우터에는 한꺼번에 많은 양의 데이터가 오면 이를 해결하지 못하고 병목현상이 생겨 라우터 버

퍼가 오버플로우 되어 패킷들이 손실되는 경우가 발생한다. 이러한 경우를 대비하기 위해 TCP는 데이터 전송의 신뢰성을 보장하기 위한 흐름제어와 혼잡제어를 실행하게 된다. 위와 같은 네트워크의 혼잡상황일 때 TCP는 적응력 있게 패킷 량을 조절하여 데이터가 정확히 전달되도록 노력한다. 하지만 TCP를 물리적인 선을 사용하지 않는 무선 네트워크에서 사용할 때, 대부분의 전송 에러를 네트워크의 혼잡상황이라고 인식하여 TCP 혼잡제어 메커니즘을 실행한다. 따라서 무선 매쉬 네트워크에서 사용하는 TCP는 전송 에러에 대한 원인을 네트워크 혼잡 상황이라 국한하지 않으며, 정확히 인지하는 것이 가장 중요하다.

3-2 개선된 TCP의 혼잡제어 알고리즘의 진행방식 제안된 TCP의 혼잡제어 알고리즘의 순서도는 그림 3과 같다.

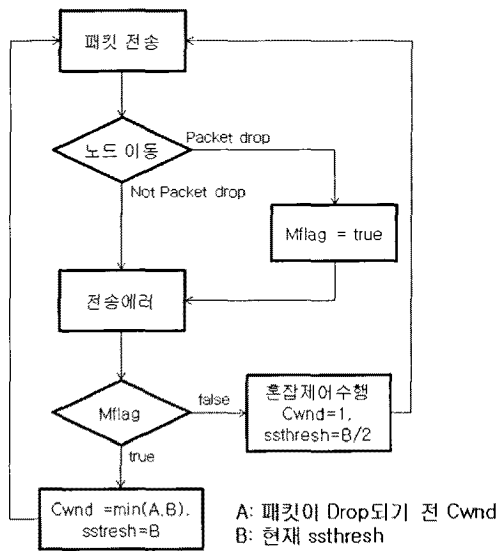


그림 3. 제안된 TCP 혼잡제어 알고리즘의 순서도
Fig. 3. Flow of Proposed TCP Congestion Control

정상적으로 패킷을 전송하다가 노드가 이동하게 되면 패킷이 Drop되어서 패킷이 손실된다. 이때 Mflag의 값은 true로 설정하여 노드가 이동해서 패킷이 Drop되었음을 알린다. 패킷이 손실됨에 따라 송신자는 중복 ACK를 받게 되고 중복 ACK가 3개 이상 수신되면 혼잡제어 메커니즘을 실행한다. 이 때 Mflag가 true면 노드가 이동

함에 따라 일시적으로 패킷이 전달되지 않은 것이며 네트워크에 많은 패킷이 몰리지 않은 것으로 판단하기 때문에 ssthresh값을 현재의 ssthresh/2로 줄이지 않고 그대로 유지하며, 패킷이 이동하기 전의 Cwnd값과 ssthresh값 중에서 작은 값으로 Cwnd를 대체한다. 그리고 Mflag가 false일 경우, 노드가 이동해도 패킷을 전송할 수 있는 범위 안에 있지만 중복 ACK를 3번 이상 수신했기 때문에 네트워크가 일시적으로 혼잡한 상태라 판단하고 정상적인 TCP 혼잡제어 메커니즘을 실행한다.

따라서 노드의 이동이 빈번한 무선 매쉬 네트워크에서 네트워크의 혼잡상황이 아닌, 경로가 변경됨에 따라 단순히 패킷이 유실될 경우, 송신 단에서는 혼잡 윈도우의 크기를 1로 만들지 않고 패킷을 보낼 수 있는 크기를 노드가 이동하기 전과 비슷하게 맞추는 것이다. 만약 노드의 경로 변경으로 패킷이 유실될 경우 혼잡 윈도우의 크기를 1로 줄이고 보낼 수 있는 패킷의 양을 송신측에서 조절한다면, 네트워크가 혼잡하지 않기 때문에 패킷을 많이 보낼 수 있는 여분의 혼잡 윈도우 크기가 있음에도 불구하고 혼잡제어 단계로 진입하는 문제가 생긴다.

IV. 시뮬레이션 및 결과 분석

4-1 시뮬레이션 모델

본 논문은 무선 매쉬 네트워크에서 개선된 TCP 혼잡제어 알고리즘의 성능 평가를 위해 다음과 같은 토폴로지 및 라우팅 그리고 각각의 값을 설정한다.

토폴로지는 그림 4와 같은 구조로 이루어져 있으며, 홉 수를 4, 8, 16개로 늘리면서 성능 값을 측정한다. 홉 수가 변하여도 기본 구조는 그림 4와 같이 매쉬 라우터와 매쉬 클라이언트로 이루어진다. 시뮬레이션 시간은 총 120초로, 노드들은 40초 이후가 되면 서서히 움직이기 시작한다. 그리고 30~50초 후 다시 경로를 변경하고 노드들이 서버 쪽으로 이동한다. 즉, 70~90초 사이에 서버와 거리가 멀어진 클라이언트들은 연결이 완전히 단절되어 패킷을 전송하지 못하고, 90초 이후엔 거의 모든 노드들이 다시 제자리를 찾아 서버와 연결되어 패킷을 재전송하게 된다.

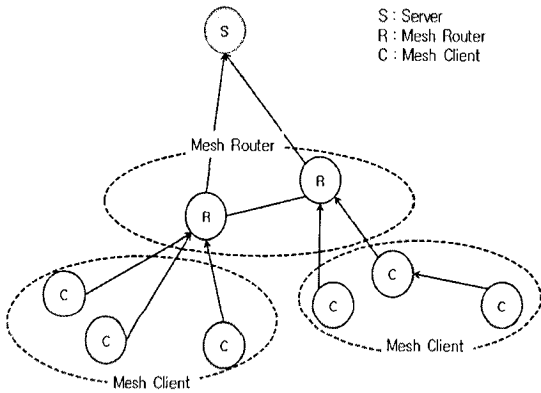


그림 4. 기본 토폴로지의 구성도
Fig. 4. Basic TCP Topology Structure

표 2. 노드 개수와 노드 종류
Table 2 A Kind of Node and Node Count

노드 종류 \ 노드 개수	4개	8개	16개
서버	1	1	1
메쉬 라우터	1	2	4
메쉬클라이언트	2	5	11

라우터 역할을 하는 노드들은 메쉬 라우터의 특성으로 움직이지 못하지만, 클라이언트 노드들은 자유로이 이동하게 된다. 클라이언트의 움직임은 일정한 이동성을 가지고 있으며, 클라이언트 노드가 이동할 때, 대부분의 패킷이 전송되지 않는다.

이 과정을 통해 경로 변경으로 인한 패킷 유실을 기존의 TCP는 혼잡상황으로 인식하게 되지만 제안된 TCP는 패킷이 경로변경으로 유실되었음을 인지하여 경로이동에 따른 효과적인 TCP 알고리즘의 성능을 분석할 수 있다.

4-2 시뮬레이션 결과분석

그림 5와 그림 6은 노드 개수가 4개 일 때, 송신 단에서 받은 제안된 TCP와 TCP Reno, TCP New-Reno의 패킷량과 처리량을 비교한 그래프이다.

그림 5와 그림 6에서, 슬로우 스타트, 혼잡회피, 빠른 재전송 단계를 실행하는 TCP Reno와 많은 양의 데이터가 손실되었을 때 빠르게 대처하는 TCP New-Reno는 수

신된 패킷량과 처리량에서 큰 차이를 보이지 않는다. TCP New-Reno는 기존의 TCP Reno에서 버스트한 패킷 손실을 해결하기 위해 추가된 알고리즘이기 때문에 노드의 수가 적어 다수의 노드에서 많은 양의 패킷 손실이 일어나지 않는 시뮬레이션에서는 성능 차이가 크지 않았다.

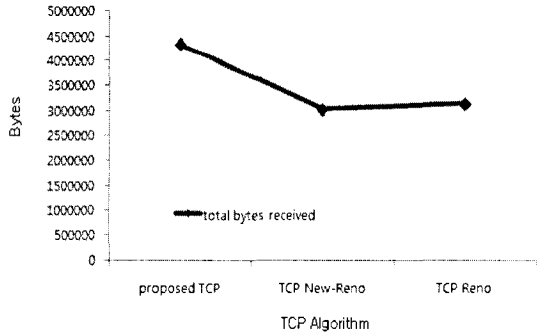


그림 5. 노드개수가 4개 일 때 수신된 총 패킷량
Fig. 5. Total Bytes received of Node Count 4

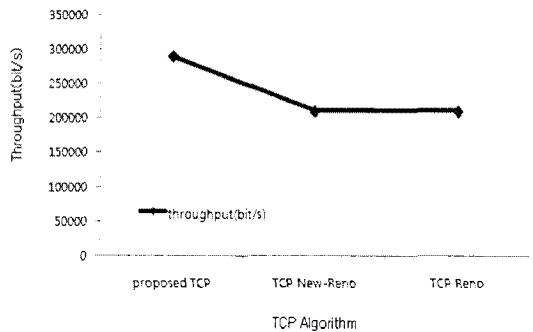


그림 6. 노드개수가 4개 일 때 처리량
Fig. 6. Throughput of Node Count 4

반면에 제안된 TCP 알고리즘은 기존의 알고리즘에 비해 상당히 많은 양의 패킷을 수신하였고, 그에 따라 처리량도 좋아지는 것을 확인할 수 있다. 이는 노드가 이동해서 서버와의 연결이 단절된 후 다시 연결을 시작했을 때 보낼 수 있는 패킷 사이즈를 줄이지 않고 그 전과 비슷하게 유지함으로써 사용할 수 있는 가용 윈도우를 활용했기 때문이다.

그림 5, 6과 같이 그림 7, 8과 그림 9, 10은 노드 개수가 8개, 16개 일 때, 송신 단에서 받은 제안된 TCP와 TCP

Reno, TCP New-Reno의 패킷 량과 처리량을 비교한 그래프이다.

그림 7, 8과 그림 9, 10은 그림 5, 6에 비해 TCP Reno와 TCP New-Reno와 패킷 수신량의 차이를 줄인 것을 알 수 있다. 이는 패킷을 송신하려는 클라이언트 노드수가 많아짐에 따라 많은 패킷의 손실에 강한 TCP New-Reno의 특성을 나타낸 결과라 할 수 있다. 하지만 TCP New-Reno가 많은 노드의 패킷 손실에 강하더라도 빠른 복구를 실행하는데 있어 혼잡 윈도우의 크기를 느리게 높이는 결점이 있기 때문에, 노드가 이동하고 재 연결될 때는 보낼 수 있는 데이터의 크기를 빠르고 효율적으로 사용한 제안된 TCP가 평균적으로 많은 양의 데이터를 전송하는 것을 알 수 있다.

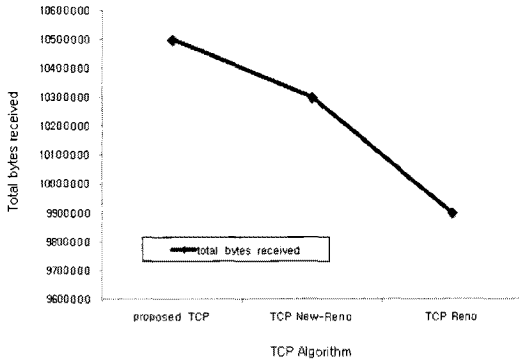


그림 7. 노드개수가 8개 일 때 수신된 총 패킷량
Fig. 7. Total Bytes received of Node Count 8

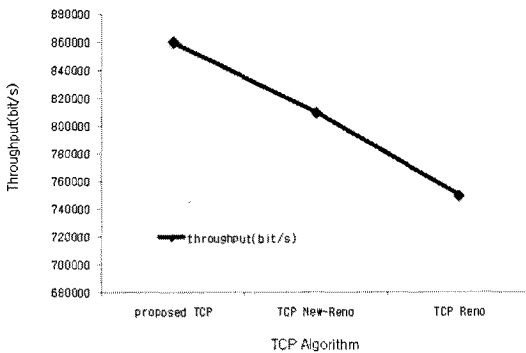


그림 8. 노드개수가 8개 일 때 처리량
Fig. 8. Throughput of Node Count 8

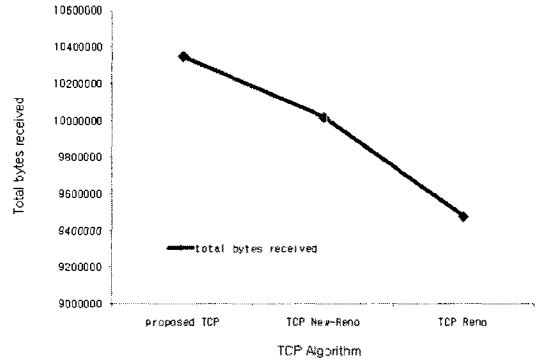


그림 9. 노드개수가 16개 일 때 수신된 패킷량
Fig. 9. Total Bytes received of Node Count 16

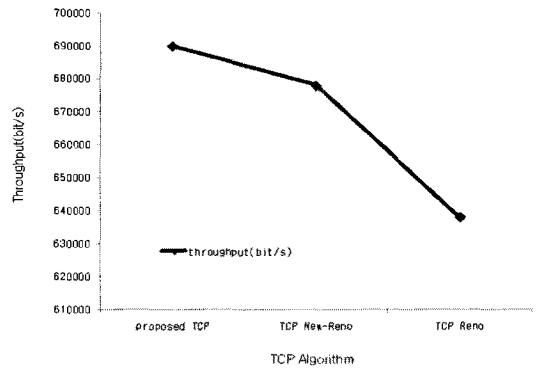


그림 10. 노드개수가 16개 일 때 처리량
Fig. 10. Throughput of Node Count 16

V. 결론

본 논문에서 제안된 TCP는 노드가 이동함에 따라 생기는 패킷 손실에 대해 기존의 혼잡제어를 실행하지 않고 적응력 있게 혼잡 윈도우의 크기를 조절하여 노드가 움직여서 일시적인 패킷 손실이 발생해도 윈도우 사이 즈를 그전과 비슷하게 유지하여 평균적인 패킷 처리량을 높였다. 또한 TCP 혼잡제어 알고리즘이 이동성을 가진 무선 메쉬 네트워크 환경에서 얼마만큼 영향을 미치는지 알아보았고, 대표적인 방식인 TCP Reno, TCP New-Reno를 논문에서 제안한 TCP 혼잡제어 알고리즘과 동일한 상황에서 모의 실험하였으며, 제안된 TCP가

무선 메쉬 네트워크 환경에서 최적의 성능을 발휘하는 것을 볼 수 있었다.

본 논문을 바탕으로 무선 환경은 제약이 많고 Hidden 터미널 문제 등의 여러 문제들이 고려되어야 한다. 따라서 많은 연구를 통해 신뢰성을 유지하면서 혼잡상황을 정확하게 이해할 수 있는 알고리즘을 지속적으로 연구해야 한다.

참고문헌

- [1] G. Xylomenos and G. C. Polyzos, "TCP Performance Issues over Wireless Link," *IEEE Communications Magazine*, vol. 39, pp.52-58, April 2001.
- [2] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," RFC-2001, Jan. 1997.
- [3] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno, and SACK TCP," *Comput. Commun. Rev.*, 1996.
- [4] C. Barakat, E. Altman, and W. Dabbous, "On TCP Performance in a Heterogeneous Network: A Survey," *IEEE Communications Magazine*, Jan. 2000.
- [5] L. Patterson, Larry and S. Bruce, Davie, *Computer Networks* 3rd edition, Morgan Kaufmann, pp. 468-478, 2003.
- [6] W. Richard Stevens, *TCP/IP Illustrated*, Reading MA, volume 1. Addison-Wesley, 1994.

저자소개

이혜림(Hye-Rim Lee)



2008년 8월 : 한국기술교육대학교
인터넷미디어공학부 졸업
(공학사)
2008년 9월 ~ 현재 : 한국기술
교육대학교 대학원
정보미디어공학과 재학

※ 관심분야: 무선 TCP, 무선 메쉬 네트워크, 무선 인터넷, 모바일 네트워크 응용

문일영(II-Young Moon)



2000년 2월 : 한국항공대학교 항공
통신정보공학과 (공학사)
2002년 2월 : 한국항공대학교
대학원 항공통신정보
공학과 (공학석사)

2005년 2월 : 한국항공대학교 대학원 정보통신공학과 졸업(공학박사)

2004년 ~ 2005년 : 한국정보문화진흥원 선임연구원

2005년 3월 ~ 현재 : 한국기술교육대학교

인터넷미디어공학부 조교수

※ 관심분야: 무선 인터넷 응용, 무선 인터넷, 모바일 IP