

IEEE 802.11 기반 메쉬 네트워크에서 다중 홉 트래픽을 위한 버퍼 관리 방식

종신회원 장길웅*

A Buffer Management Scheme for Multi-hop Traffic in IEEE 802.11 based Mesh Networks

Kil-woong Jang* *Lifelong Member*

요약

본 논문에서는 다중 홉으로 구성된 IEEE 802.11 기반의 메쉬 네트워크에서 노드 간에 패킷 전송 시 버퍼 오버플로우에 의한 패킷 손실을 줄이고 수신되는 패킷의 공정성을 높이기 위한 버퍼 관리 방식을 제안한다. 제안된 방식은 전송 패킷을 전달하는 중간 노드에서 인접 메쉬 라우터와 이동 노드로부터 전송된 패킷을 균등하게 수신하고, 각 메쉬 라우터에서 다중 홉을 거친 패킷 수신율을 향상시킴으로써 패킷 전송 지연과 노드의 에너지 소모를 줄일 수 있다. 제안된 방식은 기존의 IEEE 802.11 MAC 프로토콜 하에서 수정된 RTS/CTS를 이용하여 전송될 패킷의 크기를 미리 인지함으로써 패킷 손실과 패킷 공정성을 향상시킨다. 시뮬레이션을 이용하여 기존의 방식과 제안된 방식의 패킷 손실률, 각 메쉬 라우터에서 수신되는 패킷 양을 측정함으로써 제안된 방식의 성능을 비교 평가하였다.

Key Words : buffer management, multi-hop traffic, wireless mesh networks, packet fairness, packet loss

ABSTRACT

In this paper, we propose a buffer management scheme for decreasing the packet loss due to buffer overflow and improving the packet fairness between nodes in IEEE 802.11 based multi-hop mesh networks. In the proposed scheme, each mesh router that is an intermediate node receives fairly packet sent from neighboring mesh routers and mobile nodes, and it improves the reception ratio of multi-hop traffic of neighboring mesh routers. Therefore, the proposed scheme can reduce transmission delay and energy consumption. In order to improving the packet loss and the packet fairness, the proposed scheme uses the modified RTS/CTS under the IEEE 802.11 MAC protocol and reduces the packet loss by recognizing the packet size to send to the destination in advance. By using the simulation, we evaluated the proposed scheme in terms of the packet loss ratio and the number of received packet in each mesh router, and compare it to a traditional scheme.

1. 서론

최근 무선 메쉬 네트워크는 기존의 다양한 무선 네트워크를 통합한 차세대 네트워크로 빠르게 발전하고 있다. 무선 메쉬 네트워크는 광대역 홈 네트워크, 지역 네트워크, 기업 네트워크, 빌딩 자동화 등 다양한

분야의 무선 기술로 주목받고 있다. 무선 매체를 사용하는 노드는 전송 거리의 한계를 가진다. 무선 메쉬 네트워크에서는 전송 거리의 한계를 극복하기 위해서 전송된 패킷을 전달하기 위한 메쉬 라우터가 존재하며, 이동 단말에서 메쉬 라우터를 통해 목적지까지 패킷을 전송하는 다중 홉 구조로 이루어진다. 여기서 메쉬 라

* 한국해양대학교 데이터정보학과 네트워크연구실(jangkw@hhu.ac.kr)

논문번호 : KICS2008-12-561, 접수일자 : 2008년 12월 18일, 최종논문접수일자 : 2009년 4월 8일

우터는 중간 노드 기능과 더불어 다양한 네트워크 프로토콜을 처리할 수 있는 통합 기능을 함께 가진다^[1].

인터넷 사용자가 증가함에 따라 기존의 유선 네트워크로 네트워크 서비스를 제공하기에는 비용과 공간적인 제약이 많이 따른다. 이를 해결하기 위한 방법으로 네트워크를 하는 노드 간에 무선 주파수를 이용한 무선 메쉬 네트워크가 한 가지 방법으로 대두되고 있다. 즉 무선 메쉬 네트워크는 목적지까지 패킷을 전송하기 위한 다중 홉 구조로 이루어짐으로써 비용과 공간 문제를 줄일 수 있다. 그러나 네트워크의 크기가 증가함에 따라 노드 수가 증가하고 노드 간의 홉 수가 증가함에 따라 또 다른 문제가 발생할 수 있다. 즉, 홉 수가 증가함에 따라 라우팅 문제가 복잡해지고 인접한 노드의 수가 많아지면 채널 할당 및 충돌 문제가 발생하게 된다. 또한 다중 홉으로 인하여 포워딩 기능을 가진 메쉬 라우터에서는 버퍼 관리 문제가 발생한다. 현재의 버퍼 관리 방식은 링크 계층으로 패킷이 도착했을 때 패킷의 홉 수에 대해서는 전혀 고려하지 않는다. 즉 기존의 버퍼 관리 방식은 버퍼가 모두 사용되어 공간이 없으면 홉 수와 관계없이 새로 도착된 패킷은 손실이 발생하는 구조를 가진다. 패킷의 홉 수를 고려하지 않으므로써 전송된 트래픽에 대한 공평성 문제도 발생하게 된다. 기존의 애드 혹 네트워크에서는 포워딩 기능을 담당하는 중간 노드로 패킷이 전송되면 앞서 언급한 일반적인 버퍼 관리 방식을 사용하여 패킷을 버퍼에 저장한다. 인접 노드로부터 전송된 홉 수가 작은 패킷으로 인해 이미 버퍼가 모두 사용되어 버퍼 용량이 부족할 때 상대적으로 홉 수가 큰 패킷이 중간 노드에 도착했을 때 패킷 손실이 발생하게 된다. 홉 수가 큰 패킷의 경우 전송되는 과정에 의해 전송 지연이나 에너지 관점에서 홉 수가 작은 패킷에 비해 중요도가 높다. 따라서 홉 수가 큰 패킷의 손실은 네트워크 전체의 처리율이나 효율적인 면에서 많은 문제가 될 수 있다.

본 논문에서는 다중 홉 메쉬 네트워크에서 다중 홉을 거친 패킷의 수신율을 향상시키고 전송된 패킷의 손실을 줄이기 위한 버퍼 관리 방식을 제안한다. 제안된 방식은 IEEE 802.11 MAC 프로토콜^[2]을 기반으로 수행되며 전송된 패킷 손실을 줄이기 위해 수정된 RTS/CTS 구조를 제안한다. 제안된 방식에서는 패킷 손실이 발생할 확률이 높은 다중 홉을 거친 포워딩 패킷을 위하여 버퍼 공간을 할당함으로써 기존의 방식에 비해 패킷 공평성을 높인다. 또한, 수정된 RTS/CTS를 이용하여 전송할 패킷 크기를 미리 설정함으로써 패킷 손실을 줄이게 되고 전송 지연 및 에너

지 소모를 줄일 수 있는 장점을 가지게 된다.

II. 관련 연구

다양한 네트워크 환경에서 패킷 공평성과 버퍼 관리를 위한 연구가 이루어졌다. Floyd et al.^[3]는 인터넷 환경에서 사용된 버퍼의 평균값을 측정함으로써 혼잡을 제어하는 방식을 제안하였다. 이 방식은 최대값과 최소값을 설정하여 버퍼에 저장된 패킷의 양이 최소값이 될 때까지 패킷을 버퍼에 저장하고 만약 저장된 패킷의 양이 최대값을 넘으면 입력되는 패킷은 손실이 발생하도록 설계되었다. 만약 저장된 패킷의 양이 최대값과 최소값 사이에 존재하면, 입력된 패킷은 버퍼의 패킷 증가 확률에 따라 버퍼에 저장되거나 손실이 발생한다. 이 방식에서는 패킷을 구별하는 방법을 제시하지 않기 때문에 패킷에 대한 공평성 문제를 해결하지 못하는 단점을 가진다. Lin et al.^[4]은 각 트래픽에 대한 정보를 게이트웨이의 버퍼에 저장함으로써 각 트래픽에 대하여 공평하게 버퍼가 사용되도록 하는 방식을 제안하고 있다. 그러나 앞서 기술한 두 가지 방식은 주로 인터넷 환경에서 라우터의 트래픽 혼잡을 제어하기 위해 설계된 것이며, 다중 홉 무선 네트워크 환경에는 적합하지 않은 방식이다.

다중 홉 무선 네트워크 환경 하에서 패킷 공평성과 혼잡 제어를 위한 연구도 수행되었다. Shakkottai et al.^[5]은 다중 홉 무선 네트워크에서 홉 단위의 혼잡 제어를 위한 시스템 모델을 제시하였으며, 전송된 트래픽을 제어하기 위한 홉 단위의 분산 혼잡 제어 방식을 제안하였다. Gambiroza et al.^[6]은 네트워크상의 노드 간에 사용된 채널 정보를 교환함으로써 각 노드의 최대 채널 사용 시간을 결정하는 공평성 알고리즘을 제안하였다. 이 방식은 다중 홉 애드 혹 네트워크에서 TCP 프로토콜을 수정 없이 각 트래픽의 공평성을 높이기 위한 분산 알고리즘이다. 이 두 방식은 전송된 트래픽에 대한 혼잡 제어 및 공평성 문제를 해결하기 위해 전송 프로토콜 관점에서 접근하였으며, 반면에 본 논문에서 제안하는 방식은 버퍼 관점에서 문제를 해결하고 있다.

Sichitiu et al.^[7]은 기존의 다양한 방식을 공평성과 처리율 관점에서 분석하였으며, 전달을 담당하는 모든 중간 노드 관점에서 각 소스 노드에서 전송한 트래픽을 구별하기 위해 각각의 버퍼를 두어 관리하는 방식을 제안하였다. 그러나 무선 메쉬 네트워크와 같이 노드의 수가 많은 큰 규모의 네트워크에서 각 소스 노드를 위한 버퍼를 따로 두어 처리한다는 것은 실제로 불

가능하다. Nandiraju et al.^[8]은 멀티 홉 무선 네트워크에서 홉 수가 큰 패킷 전송을 위한 버퍼 관리 방식을 제안하였다. 이 방식은 패킷을 전달하는 중간 노드에서 각 소스 노드 당 버퍼 공간을 따로 설정하고, 입력되는 모든 트래픽에 대하여 도착률을 계산하여 패킷을 버퍼에 저장한다. 즉, 소스 노드의 도착률에 따라 사용할 수 있는 버퍼의 크기를 제어하는 방식을 제안하고 있다. 무선 애드 혹 네트워크에서 홉 수가 큰 패킷은 일반적으로 도착률이 낮으며, 반대로 홉 수가 작은 인접 노드의 패킷은 도착률이 높은 특성을 이용하였다. 그러나 홉 수가 큰 패킷에 대하여 패킷 공평성을 높게 유지하면 도착률이 높은 인접 패킷은 패킷 손실이 많이 발생하게 된다. 이 방식에서는 패킷 손실을 줄이기 위한 방법이 제시되지 않고 있으나, 본 논문에서 제안된 방식은 패킷 손실을 최소화하고 처리율을 높이기 위한 방식을 제안한다.

III. 제안된 버퍼 관리 방식

본 논문에서는 다중 홉으로 구성된 무선 메쉬 네트워크에서 패킷 전송 시 수신 노드의 버퍼 오버플로우로 인한 패킷 손실을 줄이고 수신되는 패킷의 공평성을 높이기 위한 버퍼 관리 방식을 제안한다. 제안된 방식은 패킷 공평성과 손실 관점에서 설계되었다.

다중 홉으로 구성된 메쉬 네트워크에서 홉 수가 상대적으로 작은 노드에서 보낸 패킷이 홉 수가 큰 패킷에 비해 메쉬 라우터의 버퍼에 저장될 가능성이 높으며, 홉 수가 큰 패킷일수록 버퍼 공간 부족으로 패킷 손실이 발생할 확률이 높다. 예를 들어 그림 1과 같이 하나의 링크만 가진 간단한 형태의 다중 홉 네트워크에서 시뮬레이션을 수행했을 때 그림 2와 같은 결과가 발생하였다. 이 결과에서 알 수 있는 것은 이동 노드는 작은 크기의 패킷을 짧은 홉 수로 자주 보내지만 메쉬 라우터에서는 이동 노드에서 들어온 패킷을 결집(aggregation)하여 큰 패킷으로 전송을 하기 때문에 수신하는 메쉬 라우터의 버퍼 공간부족으로 패킷 손실이 발생하게 된다. 따라서 홉 수가 증가할수록 패킷 손실은 더 증가하게 될 것이며, 홉 수가 큰 경우에는 그 만큼 더 많은 패킷을 재전송함으로써 전송 에너지와 지연이 증가하게 될 것이다. 이것은 홉 수를 많이 거친 패킷일수록 더 많은 비용이 들게 되며, 수신 패킷의 공평성이 깨지게 되는 것이다. 이를 해결하기 위해 제안된 방식에서는 인접 메쉬 라우터에서 전송된 홉 수가 큰 패킷을 위한 공간을 따로 관리함으로써 수신 패킷의 공평성을 높인다.

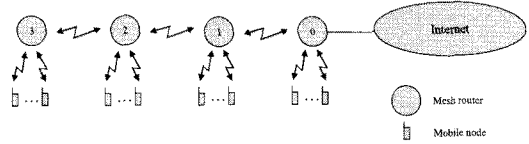


그림 1. 하나의 링크를 가진 메쉬 네트워크 구조

무선 메쉬 네트워크에서는 이동 노드로부터 전송된 패킷은 중간 노드를 통해 목적지까지 패킷을 전송한다. 이때, 전달 기능을 가진 중간 메쉬 라우터에서는 수신된 패킷을 버퍼에 저장하고 그 패킷을 서비스 처리율에 근거하여 패킷을 전송하게 된다. 그러나 이 과정에서 전송 충돌, 전송 에너지 증가 등 많은 문제점이 발생하게 된다. 따라서 이러한 문제를 줄이기 위해 수신된 패킷을 결집시키고 크기가 큰 형태의 패킷으로 전송하게 된다. 크기가 큰 패킷을 전송함에 따라 그 패킷을 수신하는 다음 메쉬 라우터에서 이동 노드로부터 많은 트래픽이 발생할 경우 버퍼 용량 부족으로 인하여 패킷 손실이 발생할 확률이 높아지게 된다. 이를 해소하기 위한 방법으로 두 메쉬 라우터 간에 미리 전송할 수 있는 패킷의 크기를 조정하여 전송하게 함으로써 패킷 손실을 줄일 수 있다. 즉 전송 요청이 이루어지는 과정에서 전송 가능한 패킷 크기를 설정함으로써 패킷 손실을 막을 수 있다.

제안된 방식은 다음과 같은 특징을 가진 네트워크에서 동작한다^[9]. 메쉬 라우터와 메쉬 라우터 간 전송 인터페이스와 메쉬 라우터와 이동 노드 간 전송 인터페이스는 서로 다른 것을 사용한다. 따라서 두 인터페이스 간에는 전송 충돌이 발생하지 않는다. 그러나 다수의 메쉬 라우터간이나 메쉬 라우터와 다수의 이동 노드 간에는 각 하나의 인터페이스를 사용하기 때문에 MAC 프로토콜을 이용하여 충돌을 방지해야 한다. 하나의 메쉬 라우터에서는 하나의 입력 버퍼를 이용

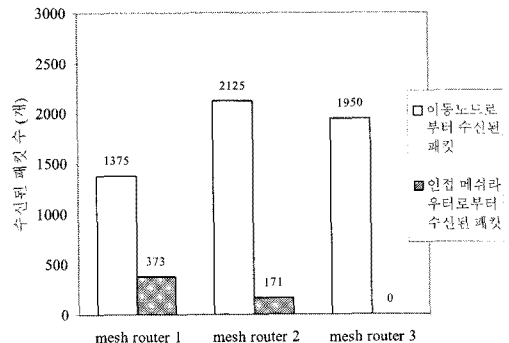


그림 2. 각 메쉬 라우터에서 수신된 패킷의 수

하며 인접 메쉬 라우터와 이동 노드에서 들어온 패킷은 하나의 버퍼에 저장된다.

제안된 버퍼 관리 방식의 동작을 기술하기에 앞서 버퍼 구조에 대하여 먼저 기술한다. 제안된 버퍼의 구조는 그림 3과 같다. 그림에 나타난 구조는 논리적인 메쉬 라우터의 버퍼 구조이다. 즉, 그림처럼 버퍼를 물리적으로 구분한 것이 아니라 저장되는 패킷의 양을 측정하여 그 크기만큼의 영역만 사용함을 의미한다. 버퍼는 크게 Primary와 Secondary 두 부분으로 나뉜다. Primary는 메쉬 라우터에 속해 있는 이동 노드와 인접한 메쉬 라우터에서 보낸 패킷을 우선적으로 저장하는 영역이며, Secondary는 Primary가 패킷으로 모두 채워졌을 때 그 다음 순서로 채워지는 영역이다. Primary는 다시 $ND_threshold$ 와 $MR_threshold$ 로 나뉜다. $ND_threshold$ 는 메쉬 라우터에 속해 있는 이동 노드가 그 메쉬 라우터로 패킷을 전송할 경우 우선적으로 패킷을 저장하는 영역이며, $MR_threshold$ 는 인접한 메쉬 라우터에서 전송한 패킷을 우선적으로 저장하는 영역이다. 두 영역에서 사용된 패킷의 양을 저장하기 위해 이동 노드나 인접 메쉬 라우터에서 전송한 패킷의 양을 가리키는 $ND_counter$ 와 $MR_counter$ 를 각각 둔다. 예를 들어, $MR_counter$ 의 값이 $MR_threshold$ 값보다 클 경우 메쉬 라우터에서 전송되는 패킷은 Secondary 영역으로 저장된다.

제안된 방식에서는 패킷 손실을 줄이기 위하여 IEEE 802.11 MAC 프로토콜의 수정된 RTS/CTS 구조를 제안한다. 본 논문에서 제안하는 수정된 RTS/CTS 구조는 그림 4와 같다. 송신측에서 수신측으로 패킷을 전송하기에 앞서 전송 권한을 얻기 위해 RTS를 전송한다. 이때 제안된 방식에서는 기존의 RTS 구조에서 전송하고자 하는 패킷의 크기를 나타내는 Packet size 필드를 추가한다. 패킷 크기를 전송함으로써 수신측에서는 전송되는 패킷의 크기를 미리 인지함으로써 자신의 버퍼에서 패킷 손실이 발생하는지 알 수 있다. 만약 송신할 패킷의 크기가 남은 버퍼 용량보다 클 경우 수신 노드는 남은 버퍼 용량만큼의 패킷만 전송하도록 CTS의 packet size 필드를 이용하여 송신 노드에게 알린다.

제안된 버퍼 관리 방식의 동작은 이동 노드 관점과

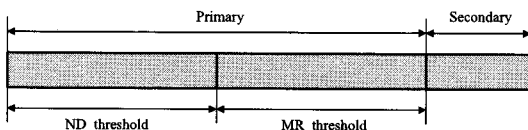
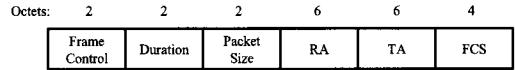
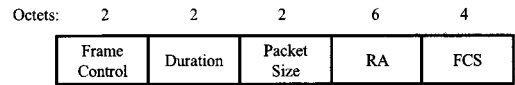


그림 3. 제안된 방식의 논리적인 버퍼 구조



(a) RTS structure



(b) CTS structure

그림 4. 수정된 RTS/CTS 구조

인접 메쉬 라우터 관점에 따라 두 가지로 구분된다. 수정된 RTS/CTS 메커니즘을 이용하여 제안된 버퍼 관리 방식은 다음과 같이 동작한다. 먼저 이동 노드 관점에서 보면, 이동 노드에서 RTS를 통해 길이가 K인 패킷을 전송하고자 한다면, 수신측인 메쉬 라우터에서는 자신의 버퍼 상태를 검사한다. 만약 $ND_counter$ 의 값이 $ND_threshold$ 의 값보다 큰 상태이면 이동 노드의 패킷을 수신하지 않는다. 즉, CTS를 전송하지 않음으로써 노드의 패킷을 수신하지 않는다. 그렇지 않을 경우에는 전송되는 패킷의 길이와 $(ND_threshold - ND_counter)$ 의 값을 비교한다. 비교 결과에서 패킷의 길이가 작을 경우에는 K 크기의 패킷 전체를 수신하고 $ND_counter$ 값을 변경한다. 패킷의 길이가 클 경우에는 다시 패킷의 길이와 $(ND_threshold + Secondary - ND_counter)$ 의 값을 비교한다. 패킷의 길이가 작을 경우에는 K 크기의 패킷 전체를 수신하고 $ND_counter$ 값과 Secondary 값을 변경한다. 패킷의 길이가 클 경우에는 $(ND_threshold + Secondary - ND_counter)$ 의 크기만큼의 패킷만 송신 노드에게 CTS를 이용하여 요청한다. 패킷을 수신한 메쉬 라우터는 $ND_counter$ 값과 Secondary 값을 변경한다. 여기서 Secondary 값은 사용된 만큼 값을 줄여 나간다. 메쉬 라우터에서 인접 메쉬 라우터로 버퍼링된 패킷을 전송하게 되면 우선적으로 Secondary 값을 전송한 크기만큼을 증가하게 된다.

인접 메쉬 라우터 관점에서 제안된 방식은 다음과 같이 동작한다. 인접 메쉬 라우터에서 RTS를 통해 길이가 K인 패킷을 전송하고자 한다면, 수신측인 메쉬 라우터에서 자신의 버퍼 상태를 검사한다. 만약 $MR_counter$ 의 값이 $(MR_threshold + Secondary)$ 의 값보다 작을 경우 전송되는 패킷의 길이와 $(MR_threshold + Secondary - MR_counter)$ 의 값을 비교한다. 패킷의 길이가 작을 경우에는 K 크기의 패킷 전체를 수신하고 $MR_counter$ 값과 Secondary 값을 변경한다. 패킷의 길이가 클 경

우에는 $(MR_threshold + Secondary - MR_counter)$ 크기만큼의 패킷만 송신 노드에게 CTS를 이용하여 요청한다. 패킷을 수신한 메쉬 라우터는 $MR_counter$ 값과 $Secondary$ 값을 변경한다. 앞서 $MR_counter$ 의 값과 $(MR_threshold + Secondary)$ 의 값을 비교했을 때 비교 결과가 클 경우에는 $ND_counter$ 의 값과 $ND_threshold$ 의 값을 비교한다. $ND_counter$ 값이 작을 경우에는 $(ND_threshold - ND_counter)$ 크기만큼의 패킷만 송신 노드에게 요청한다. 패킷을 수신한 메쉬 라우터는 $MR_counter$ 값과 $ND_threshold$ 값을 변경한다. $ND_counter$ 값이 클 경우에는 인접 메쉬 라우터로부터 패킷을 수신하지 않는다. 앞선 이동 노드 관점과 마찬가지로 패킷 수신을 하지 않을 때는 CTS를 전송하지 않으며, 동작 과정에서 변경되는 $ND_threshold$ 와 $Secondary$ 값은 사용된 만큼 값을 줄여 나간다. 또한 메쉬 라우터에서 인접 메쉬 라우터로 버퍼링된 패킷을 전송하게 되면 $ND_threshold$ 값을 먼저 증가시키고 그 다음으로 $Secondary$ 값을 전송한 크기만큼을 증가하게 된다. 지금까지 기술된 제안된 버퍼 관리 동작에 대한 알고리즘은 그림 5와 6에서 다시 기술한다.

```

if a node will send a K size packet:
if (ND_counter < ND_threshold)
if (ND_threshold - ND_counter >= K)
allow to send the K size packet
save the packet in the buffer and update
ND_counter
else
if ((ND_threshold + Secondary) - ND_counter >=
K)
allow to send the K size packet
save the packet in the buffer
update ND_counter and Secondary
else
allow to send the ((ND_threshold + Secondary)
- ND_counter) size packet
save the packet in the buffer
update ND_counter and Secondary
end if
end if
else
disallow to send the packet
end if

```

그림 5. 이동 노드 관점의 알고리즘

```

if a neighboring mesh router will send a K size packet:
if (MR_counter < MR_threshold + Secondary)
if ((MR_threshold + Secondary) - MR_counter >= K)
allow to send the K size packet
save the packet in the buffer
update MR_counter and Secondary
else
allow to send the ((MR_threshold + Secondary) -
MR_counter) size packet
save the packet in the buffer
update MR_counter and Secondary
end if
else
if (ND_counter < ND_threshold)
allow to send the (ND_threshold - ND_counter)
size packet
save the packet in the buffer
update MR_counter and ND_threshold
else
disallow to send the packet
end if
end if

```

그림 6. 인접 메쉬 라우터 관점의 알고리즘

IV. 성능평가

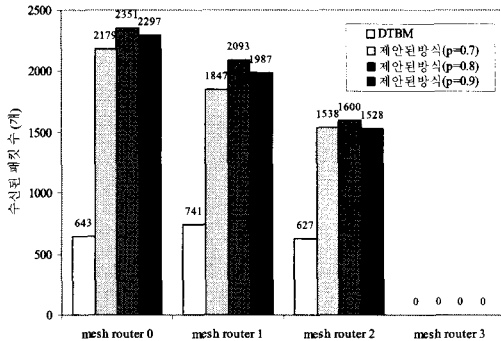
제안된 버퍼 관리 방식에 대한 성능을 평가하기 위해 시뮬레이션을 수행하였다. 또한, 제안된 방식을 비교 평가하기 위해 기존의 방식과 비교하였다. 기존 방식은 수신 버퍼가 모두 사용 중일 경우 새로운 패킷이 수신되면 패킷 손실이 발생하는 방식(Drop tail buffer management scheme: DTBM)이다. 시뮬레이션에 사용되는 네트워크는 앞서 기술한 그림 1과 같은 구조를 가진다. 사용되는 네트워크 구조는 4개의 메쉬 라우터가 사용되고 각 메쉬 라우터에서는 다수의 이동 노드를 가지며, 이동 노드에서 전송되는 패킷은 각 메쉬 라우터를 통해 인터넷으로 전송된다. 실제 메쉬 네트워크에서는 각 메쉬 라우터와 연결되는 인접 메쉬 라우터는 다수로 구성된다. 그렇지만 본 논문에서는 시뮬레이션의 복잡성을 줄이기 위해 메쉬 라우터간의 링크를 하나만 가지는 구조를 이용하였다. 만약 다수의 링크가 수직 교차 주파수를 사용하거나 공간적으로 충분히 떨어져 있다면 하나의 링크를 가진 네트워크 구조에서 얻어진 결과는 시스템 전반의 동작을 대신할 수 있다⁶⁾. 하지만, 다수의 링크가 전송 범위 내에 존재한다면, 전송 충돌이나 간섭이 증가함에 따라 본 논문에서 제시하는 처리율보다 낮게 나타날 수 있다.

시뮬레이션 상에서 사용된 각 메쉬 라우터(0, 1, 2, 3)에 속하는 이동 노드 수는 4, 3, 4, 4개로 설정하였다. 이동 노드가 메쉬 라우터로 전송하는 패킷 크기는 1000 byte 이며, 각 이동 노드에서 전송되는 패킷 도착률을 3초로 한 포아송 분포를 따른다. 메쉬 라우터에서 사용되는 버퍼의 크기는 50 Kbyte 이며, 서비스율은 IEEE 802.11 프로토콜의 2 Mbps로 설정하였다. 성능평가에서 각 메쉬 라우터에 입력된 패킷은 이동 노드에서 전송한 패킷과 이전 메쉬 라우터에서 보낸 패킷으로 구분한다. 본 논문에서는 이동 노드에서 전송된 패킷을 노드 패킷, 이전 메쉬 라우터에서 보낸 패킷을 포워드 패킷이라고 칭한다.

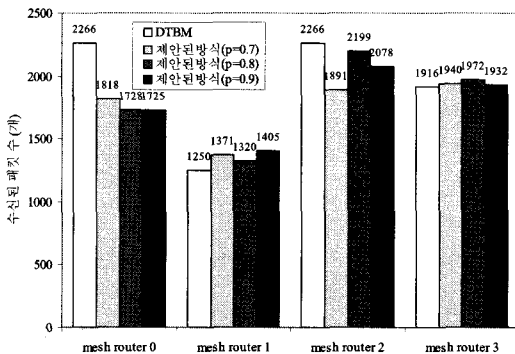
제안된 버퍼 방식을 평가하기 위해 Primary와 Secondary의 비율과 ND_threshold와 MR_threshold의 비율을 각각 다르게 설정하였으며, 시뮬레이션 결과에서 사용된 각 비율은 다음 식과 같다.

$$p = \text{Primary} / (\text{Primary} + \text{Secondary}) \quad (1)$$

$$q = \text{ND_threshold} / \text{Primary} \quad (2)$$



(a) 포워드 패킷

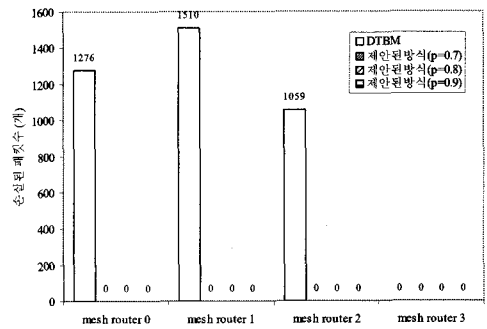


(b) 노드 패킷

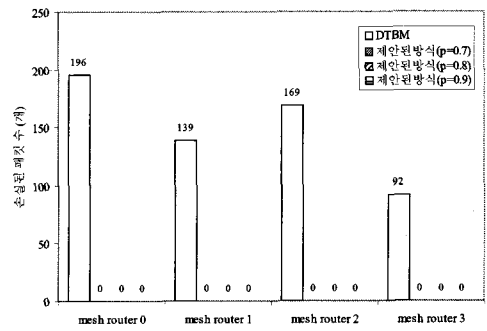
그림 7. 각 메쉬 라우터에서 수신되는 패킷의 수(q = 0.5)

그림 7은 ND_threshold와 MR_threshold를 같은 값으로 설정하였을 때 각 메쉬 라우터에서 수신되는 포워드 패킷과 노드 패킷의 수를 나타낸 것이다. 기존의 방식에서 수신된 노드 패킷에 비하여 포워드 패킷이 상당히 적음을 알 수 있다. 이것은 노드 패킷이 버퍼를 대부분 차지함으로써 포워드 패킷은 버퍼, �버플로우에 의해 손실이 발생함으로써 나타난 현상이다. 반면에 제안된 방식은 포워드 패킷을 위한 버퍼의 공간 분할로 인하여 포워드 패킷과 노드 패킷의 비가 비슷한 형태로 전송됨을 알 수 있다.

그림 8은 ND_threshold와 MR_threshold를 같은 값으로 설정하였을 때 각 메쉬 라우터에서 손실된 패킷의 수를 나타낸 것이다. 기존의 방식에서 손실되는 패킷은 대부분 포워드 패킷이 차지하고 있으며 제안된 방식을 사용했을 때는 손실되는 패킷이 발생하지 않음을 볼 수 있다. 제안된 방식에서는 수정된 RTS/CTS 방식의 MAC 프로토콜을 병행하여 전송되는 패킷 크기를 미리 조절함으로써 패킷 손실이 발생하지 않는다.



(a) 포워드 패킷

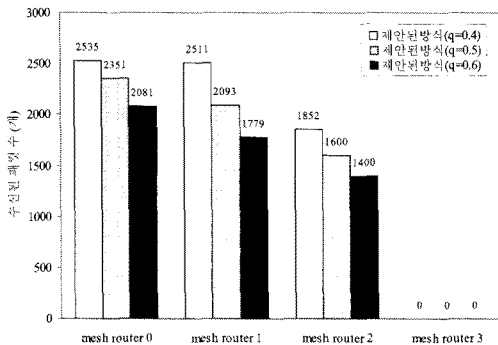


(b) 노드 패킷

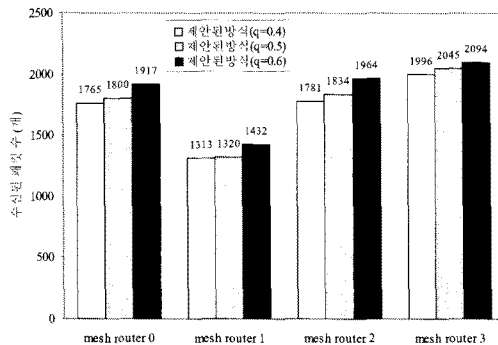
그림 8. 각 메쉬 라우터에서 손실된 패킷의 수(q = 0.5)

그림 7과 8에서 기존의 방식을 사용했을 때 이전 메쉬 라우터와 이동 노드에서 수신된 총 패킷의 수는 제안된 방식을 이용했을 때 보다 더 많은 패킷이 수신 되었음을 알 수 있다. 제안된 방식에서는 패킷 손실이 발생하지 않기 때문에 기존의 방식에 비해 총 수신 패킷의 수는 줄어들게 된다. 이것은 제안된 방식이 전송 에너지 감소와 재전송에 의한 전송지연을 줄일 수 있다는 것을 보여준다.

그림 9는 Primary의 비율을 전체 프레임의 80%로 두었을 때 제안된 방식에서 ND_threshold 값에 따라 수신되는 패킷의 수를 측정하는 것이다. 노드 패킷을 위한 ND_threshold 값이 클수록 노드 패킷의 수는 증가하고 반대로 포워드 패킷의 수는 감소됨을 알 수 있다. 만약 입력되는 패킷의 수를 정확히 측정하거나 예상할 수 있다면 ND_threshold 값을 동적으로 변화시킴으로써 입력되는 패킷을 효과적으로 처리할 수 있다. 그림 7에서의 마찬가지로 메쉬 라우터 3은 네트워크 모델의 끝단에 위치함으로써 포워드 패킷은 수신 되지 않는다.



(a) 포워드 패킷



(b) 노드 패킷

그림 9. 제안된 방식에서 ND_threshold 값의 변화에 따른 수신 패킷의 수 (p = 0.8)

V. 결 론

본 논문에서는 IEEE 802.11 기반의 무선 메쉬 네트워크에서 다중 홉 트래픽을 위한 버퍼 관리 방식에 대하여 제안하였다. 제안된 방식은 수정된 RTS/CTS 구조를 가진 MAC 프로토콜을 이용하여 인접 메쉬 라우터와 이동 노드의 전송 패킷을 효율적으로 수신하기 위한 버퍼 관리 방식이다. 수정된 RTS/CTS를 이용하여 전송할 수 있는 패킷 크기를 미리 인지함으로써 수신하는 노드에서는 패킷 손실을 줄일 수 있다. 홉 수가 작은 이동 노드의 패킷과 홉 수가 상대적으로 큰 인접 메쉬 라우터의 패킷 공평성을 위해 수신 노드 버퍼의 공간을 따로 관리함으로써 패킷 공평성을 향상시킬 수 있었다. 제안된 방식은 홉 수가 큰 인접 메쉬 라우터의 패킷 수신율을 향상시켜 패킷 전송지연을 줄임으로써 패킷 처리율 및 재전송으로 인한 에너지 효율을 향상시킬 수 있는 장점을 가진다. 시뮬레이션을 이용한 성능평가에서 패킷 손실률과 수신되는 패킷의 양을 기존의 방식을 비교했을 때 제안된 방식이 상대적으로 인접 메쉬 라우터에서 전송한 패킷의 수신율이 높음을 볼 수 있었으며, 패킷 손실률도 향상됨으로 알 수 있었다. 향후에는 수신되는 패킷의 수신율을 정확히 예측함으로써 제안된 방식의 버퍼 관리 방식의 threshold 값을 동적으로 변화시켜 더욱 효율적인 버퍼 관리 방식에 대한 연구를 하고자 한다.

참 고 문 헌

- [1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," Computer Networks. Jan. 2005.
- [2] IEEE Std. 802-11. "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification," ISO/IEC 8802-11:1999 (E), Aug., 1999.
- [3] S. Floyd, and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance," IEEE/ACM Transactions on Networking, vol.1, no.4, August 1993.
- [4] D. Lin, and R. Morris, "Dynamics of Random Early Detection," In Proceedings of ACM SIGCOMM 97.
- [5] Y. Yi, and S. Shakkottai, "Hop-by-hop congestion control over a wireless multi-hop network," In Proceedings of IEEE INFOCOM

'04, Hong Kong, March 2004.

- [6] V. Gamberoza, B. Sadeghi, and E. W. Knightly, "End to End Performance and Fairness in Multihop Wireless Backhaul Networks," In Proceedings of ACM Mobicom 2004.
- [7] J. Jun, and M. L. Sichitiu, "Fairness and QoS in multihop wireless networks," in Proc. Of the IEEE Vehicular Technology Conference (VTC), October 2003.
- [8] N. S. Nandiraju, D. S. Nandiraju, D. Cavalcanti, and D. P. Agrawal, "A Novel Queue Management Mechanism for Improving Performance of Multihop Flows in IEEE 802.11s based Mesh Networks," In Proceedings of IEEE IPCCC 2006.
- [9] R. Bruno, M. Conti, and E. Gregori, "Mesh Networks: Commodity Multihop Ad Hoc Networks," IEEE Communications Magazine, Vol. 43, No. 3, March 2005.

장길웅 (Kil-woong Jang)

중신회원



1997년 2월 경북대학교 컴퓨터
공학과 졸업

1999년 2월 경북대학교 컴퓨터
공학과 석사

2002년 8월 경북대학교 컴퓨터
공학과 박사

2003년 3월~현재 한국해양대학

교 데이터정보학과 부교수

<관심분야> 네트워크 프로토콜, 센서네트워크, 메쉬
네트워크