

채널오류에 강한 애드혹 네트워크용 협력통신 MAC 프로토콜에 관한 연구

정회원 장재신*

A Study on a Reliable Cooperative MAC Protocol for Ad Hoc Networks

Jaeshin Jang* *Regular Member*

요 약

본 논문에서는 열악한 무선채널 환경에서 시스템 처리량을 개선시킬 수 있는 협력통신용 MAC 프로토콜을 제안하고, 컴퓨터 모의실험을 통해서 성능평가를 수행하였다. 성능평가 척도로는 시스템 처리량과 평균지연시간을 사용하였으며, 성능평가 결과를 통해서 본 논문에서 제안한 기법이 기존 rDCF 기법과 비교할 때 시스템 처리량 측면에서 24% 정도 증가함을 확인할 수 있었다. 한편 평균지연시간 측면에서는 본 논문에서 제안한 기법이 단말기 수가 상대적으로 작을 경우에는 우수하지만, 단말기 수가 상대적으로 클 경우에는 미세하지만 열악한 것을 알 수 있었다. 그 이유는 단말기 수가 상대적으로 클 경우에는 채널경쟁에 의해 시스템 성능이 좌우되며, 본 논문에서 제안한 기법의 경우는 각 노드에서 수행하는 재전송 절차 때문에 CRTS 프레임을 전송한 후 설정하는 재전송 타이머의 설정시간이 기존 방식보다 큼에 기인한다. 그러나 버퍼 내 큐잉지연시간까지 고려하면 전체적인 시스템 지연시간은 rDCF 기법에 비해 감소할 것으로 예상된다.

Key Words : 협력통신, MAC, CRTS, 도움노드, Ad Hoc 네트워크, rDCF

ABSTRACT

In this paper, a cooperative MAC protocol that is reliable over bad wireless channel environments and thus can enhance the system performance much more is proposed. Its performance is evaluated with the computer simulation. The system throughput and average waiting delay are used as performance measures. According to numerical results, the proposed MAC protocol provides better system throughput by 24% than the traditional rDCF scheme. From the average waiting delay point of view, when the number of terminals is small, the proposed MAC protocol has the better performance in an average view-point than the rDCF scheme. However, when the number of terminals is large, the proposed MAC protocol provides a little worse performance than the rDCF scheme. That is because, when the number of terminals is large, the dominant factor affecting the system performance is channel contention procedure, which results in lots of collisions. However, if the queuing delays in the waiting buffers are included for calculating the average waiting delay, then the total system delay will be smaller than the rDCF scheme.

* 본 논문은 2009학년도 인제대학교 학술연구조성비 보조에 의한 것임

* 인제대학교 정보통신공학과 무선통신망 연구실 (icjoseph@inje.ac.kr)

논문번호 : KICS2010-01-041, 접수일자 : 2010년 2월 1일, 최종논문접수일자 : 2010년 5월 10일

I. 서 론

제한된 대역폭과 열악한 채널환경 등 유선망에 비해 열악한 조건을 안고 있는 무선통신분야에서도 그동안 눈부신 발전을 거듭하여 4세대 이동통신시스템에서는 최고 1Gbps까지 제공하고 있다. 그러나 이동통신망에서 단말기가 셀 경계지역에 위치해 있거나 또는 Ad Hoc 네트워크^[1]에서 통신을 수행하는 두 노드와의 거리가 멀거나 페이딩 등에 의해 잠시 채널상태가 악화되었을 경우에 발생하는 전송오류를 해결하는 문제는 여전히 풀어야 할 숙제이다. 한편 이동통신시스템에서는 셀 경계지역에서의 전송품질을 개선하기 위해 중계기를 사용하고 있지만, Ad Hoc 네트워크와 같이 단말기 이동성에 의해 망구조가 가변인 환경에서는 일반 중계기를 사용하여 열악한 무선채널문제를 해결할 수 없다.

최근 무선통신 품질을 개선하는 방법으로 협력통신(cooperative communications) 개념이 연구되고 있다. 협력통신이란 링크 어댑테이션을 사용하는 환경에서 두 노드 간의 채널환경이 급속히 나빠졌을 때 전송경로 상에서 두 노드 사이에 위치한 도움노드를 사용하여 송신노드와 도움노드, 그리고 도움노드와 목적지노드 사이의 전송경로 품질을 개선시키는 통신방법이다^[2]. 협력통신용 MAC 프로토콜은 OSI 통신모델을 기준으로 제 2계층인 데이터링크 계층에 해당되며, 지금까지는 무선 랜 프로토콜인 IEEE 802.11 DCF 프로토콜에^[3] 기반을 두고 연구되고 있기 때문에 네트워크 계층에서 바라보면 아무 변화가 없고, 따라서 투명하게 데이터 송수신 기능을 제공한다.

본 논문에서는 이러한 연구추세를 따르면서, 채널경쟁에 의한 전송오류 뿐만 아니라, 열악한 채널환경에 따른 전송오류도 발생하는 환경에서 시스템 처리량을 증가시킬 수 있는 새로운 협력통신용 MAC 프로토콜인 RCO-MAC(reliable cooperative MAC) 기법을 제안하고, 컴퓨터 모의실험을 통해서 성능평가를 수행하였다. 본 논문의 구성은 다음과 같다. 먼저 II장에서는 협력통신 MAC 프로토콜과 관련된 기존 연구동향에 대해서 기술한다. III장에서는 본 논문에서 제안한 RCO-MAC 프로토콜에 대해서 설명하며, IV장에서는 컴퓨터 모의실험에 의해 성능평가를 수행한 결과를 제시하고 기존 방법과 비교하여 성능이 증가함을 보여준 뒤, V장에서 결론으로 끝맺는다.

II. 관련연구 동향

먼저 참고문헌^[4]에서는 무선 랜에서 링크 어댑테이

션을 사용할 수 있는 MAC 헤더 규격과 절차에 대해서 기술한 RBAR(receiver-based auto rate) 기법을 제안하였다. 특히 본 논문의 관련연구동향에서 기술할 기존 협력통신용 MAC 프로토콜들도 모두 RBAR 기법에 기반을 두고 제안되었으며, 성능분석 결과 비교를 통해서 협력통신을 사용하지 않은 경우대비 어느 정도 성능이 개선되는지 판단할 기준 잣대로서 널리 사용되고 있다.

협력통신용 MAC 프로토콜에 대한 연구도 많이 진행 되었는데, 우선 RBAR 기법에 기반을 두고 기존 무선 랜 표준을 최소로 변경하면서 협력통신을 지원할 수 있는 CMAC 기법이 제안되었다^[5]. 이 연구에서는 두 노드 사이에서 전송오류가 발생하였을 때 재전송하는 절차와 또 CMAC과 채널코딩 기법을 통합한 FCMAC 기법도 제안하였다. 참고문헌^[6]에서는 기존 DCF 기법을 협력통신용으로 확장한 rDCF (relay DCF) 기법을 제안하였는데, 여기에는 도움노드를 선정하는 절차와 협력통신을 지원하기 위해 기존 DCF에서 변경되는 제어프레임들을 설명하고 (RRTS1, RRTS2, RCTS), 이 제어프레임들을 사용하여 송신노드, 수신노드 그리고 도움노드 사이에서 프레임 전송하는 절차에 대해서 기술하였다. 참고문헌^[7]에서는 CoopMAC 기법을 제안하였는데, 이 기법은 rDCF 기법과 매우 유사하지만 도움노드를 선정하는 기준을 수학적식으로 깔끔하게 나타낸 점이 눈에 띈다. 참고문헌^[8]에서는 채널경쟁단계에서 수신 다이버시티 이득을 얻기 위해 기존 rDCF 기법을 확장한 것으로, 기존에는 제어프레임을 사용하여 삼각형 형태의 핸드셰이크를 수행하던 절차를 다이아몬드 형태의 핸드셰이크로 변경하였다.

이제까지 연구결과를 살펴보면 모두 채널경쟁단계에서 충돌에 의한 오류는 발생하지만 열악한 무선채널에 의한 전송오류는 발생하지 않는다고 가정하고 성능평가를 수행하였으며, 각 연구에서 제안한 기법들은 모두 RBAR 기법과 성능을 비교하였다. 한편 참고문헌^{[9][10]}에서는 채널경쟁단계에서 충돌에 의한 오류 뿐만 아니라 열악한 무선채널에 의한 전송오류가 발생하는 환경에서 새로운 협력통신용 MAC 프로토콜을 제안하였다. 먼저 참고문헌^[9]에서는 채널경쟁단계에서 수신 다이버시티 효과를 얻으면서 복잡도는 rDCF와 동등한 수준을 달성하는 것을 목표로 CO-MAC 프로토콜을 제안하고 성능평가를 수행하였다. 또 참고문헌^[10]에서는 협력통신용 MAC 프로토콜과 네트워크 코딩 개념을 결합한 NC-MAC 프로토콜을 제안하고 성능평가를 수행하였으며, rDCF 기법 대

비 네트워크 코딩 이득이상으로 성능이 증가함을 보였다.

III. RCO-MAC 프로토콜

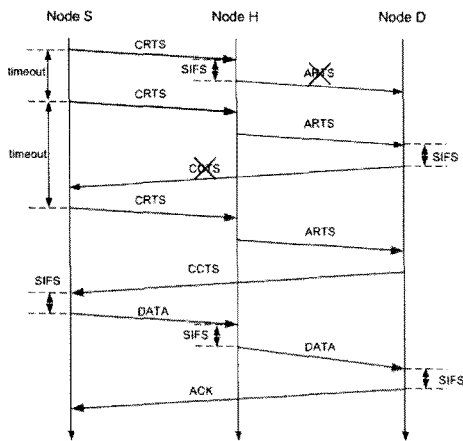
본 논문에서 제안한 RCO-MAC 프로토콜은 기존 DCF 기법에 기반을 두고 설계하였으며, 기존 협력통신 MAC 프로토콜인 rDCF의 동작절차는 그림 1-(a)에 나타내었고, 본 논문에서 제안한 RCO-MAC 프로토콜의 동작절차를 그림 1-(b)에 비교할 수 있게 나타내었다. RCO-MAC 프로토콜은 전송오류가 빈번하게 발생하는 경우에는 해당 프레임의 전송한 노드가 직접 재전송을 수행하는 것이 시스템 성능 측면에서 유리하다는 개념에 기반을 두고 제안되었다. 그래서 전송오류가 발생했다고 판단되었을 때에는 각 노드에서 효율적으로 최대한 빨리 재전송 기능을 사용해서 전

송오류에 대응할 수 있도록 동작한다. 예를 들면(그림 1-(a) 참조), 도움노드(Node H)가 전송한 ARTS 프레임이 전송 중 오류가 발생하여 목적지노드(Node D)에서 제대로 수신하지 못했을 경우에는 송신노드(Node S)에서 운영하는 재전송 타이머 동작에 의해 CRTS 프레임부터 재전송을 시도한다. 또 목적지노드에서 전송한 CCTS 프레임이 전송 중 오류가 발생하였을 경우에도 동일한 동작에 의해 송신노드에서 CRTS 프레임을 재전송한다. 그러나 송신노드가 재전송 타이머의 종료에 의해 프레임 전송오류가 발생한 사실을 확인하고 대응하기 때문에 전송오류 복구에 시간이 많이 소요되는 단점을 안고 있다. 한편, 이러한 단점을 개선하기 위해 제안된 RCO-MAC 프로토콜의 경우는(그림 1-(b) 참조), 도움노드가 전송한 ARTS 프레임에 오류가 발생하여 $2 \times \text{SIFS}$ 시간동안 목적지 노드에서 CCTS 프레임을 전송하지 않거나 아무런 동작을 수행하지 않을 경우에는 도움노드가 이 사실을 인지하고 ARTS 프레임을 직접 재전송한다. 각 노드에서 긴급하게 수행하는 프레임 재전송은 $2 \times \text{SIFS}$ 시간단위로 수행하며, 최대 N회까지 진행할 수 있다. 그러나 N회 재전송에서도 전송오류가 복구되지 않을 경우에는 송신노드에서 재전송 타이머 종료에 의해 CRTS 프레임을 재전송하는 절차를 수행한다. 여기에서 전송오류가 발생한 ARTS 프레임을 재전송하기 위해 $2 \times \text{SIFS}$ 만큼 기다리는 이유는, 목적지 노드에서 SIFS 시간만큼 기다린 후 CCTS 프레임을 전송하기 때문에 ARTS 프레임이 성공적으로 전송되었는지 여부를 판단하기 위해 가능한 짧으면서도 충분한 시간이 $2 \times \text{SIFS}$ 라고 판단하였기 때문이다.

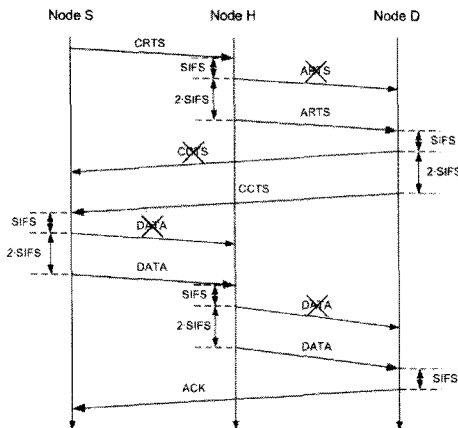
본 논문에서는 도움노드를 선정하는 방법에 대해서는 기술하지 않았으며, 참고문헌^[6-8]에서 제안한 방법과 동일한 방법을 사용할 수 있다. 지금부터는 RCO-MAC 기법에 따라서 동작하기 위해서 송신노드, 목적지노드, 그리고 도움노드에서 수행해야 하는 절차를 순서대로 자세하게 설명한다.

먼저 송신노드에서 수행할 절차를 그림 2에 나타내었다.

송신노드는 상위계층으로부터 전송할 패킷을 수신하면 기존 DCF 기법의 CSMA/CA 기법에 기반을 두고 채널경쟁을 시작한다. 그래서 먼저 경쟁 윈도우(CW: contention window) 값을 초기화시키고, 랜덤 값을 발생시킨 뒤 해당된 빈 슬롯에서 CRTS (cooperative RTS) 제어프레임을 전송한다. 그리고 전송한 CRTS 프레임에 전송오류가 발생할 경우를 대비하여 재전송 타이머 1을 구동한다. 타이머 1 값은



(a) 기존 방식 (rDCF)



(b) RCO-MAC 프로토콜

그림 1. RCO-MAC 프로토콜 동작

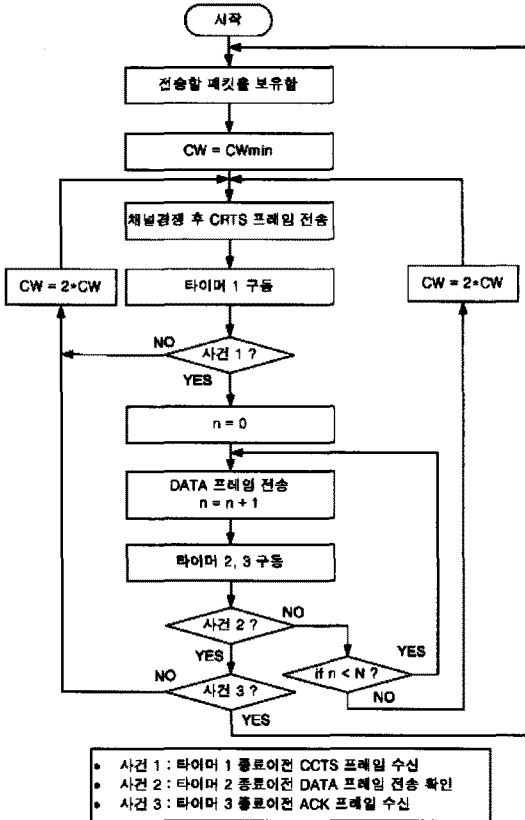


그림 2. 송신노드의 동작

CCTS 프레임을 성공적으로 수신할 때까지 소요되는 최고 시간 값으로 설정하며 기존 DCF에서는 일반적으로 ACK 프레임을 수신할 때까지 소요되는 시간으로 설정한다. 만약 타이머 1이 종료될 때까지 CCTS (cooperative CTS) 프레임을 수신하지 못하면 기존 DCF 기법처럼 경쟁 윈도우 값을 두 배로 증가시켜서 재전송을 실시한다. 그러나 타이머 1이 종료되기 이전 목적지 노드로부터 CCTS 프레임을 수신하면, 타이머 1을 종료시키고 DATA 프레임 전송을 준비한다. DATA 프레임을 전송한 후, DATA 프레임 전송이 실패한 것을 인지하면 DATA 프레임을 재전송한다. 여기에서 DATA 프레임 전송을 성공했느냐 실패했느냐는 SIFS 시간 이후에 도움노드가 DATA 프레임을 목적지노드로 전송하는지 여부를 감지함으로써 알 수 있다. 한편 기존 DCF 프로토콜에서는 DATA 프레임 전송시도에서 전송오류가 발생해도 이를 인지하지 못하며, 다만 RTS 프레임을 전송한 후 구동한 재전송 타이머가 종료될 때까지 ACK 프레임을 수신하지 못하면 RTS 프레임을 재전송한다. DATA 프레임은 최대 N회까지 재전송할 수 있으며, 재전송 횟수를 관리

하기 위해 변수 n을 사용하고, DATA 프레임을 전송하기 전에 이 값을 0으로 초기화한다. DATA 프레임을 도움노드로 전송한 후에는 변수 n 값을 1만큼 증가시키며, 재전송 절차에 의해 전송오류를 복구하기 위해 타이머 2와 타이머 3을 각각 구동한다. 타이머 2는 DATA 프레임 재전송 절차를 관리하기 위한 용도이며, 타이머 3은 지정된 시간까지 목적지노드로부터 ACK 프레임을 수신하지 못했을 때, CRTS 프레임을 사용하여 재전송하는 절차를 관리하기 위함이다. 즉, DATA 프레임을 전송한 후 두 배의 SIFS 시간 동안 (타이머 2에 설정된 값) 도움노드가 DATA 프레임을 목적지 노드로 전송하는 것을 감지하지 못하면 전송 오류가 발생했다고 판단하고, DATA 프레임을 재전송한다. 만약 이 과정을 N회 반복할 때까지 DATA 프레임 전송을 성공하지 못하거나 타이머 3이 종료될 때까지 ACK 프레임을 수신하지 못하면 경쟁 윈도우 값을 두 배로 증가시킨 뒤, CRTS 프레임을 재전송하여 전송오류 문제를 해결한다. 한편, 위에서 언급했던 세 개의 타이머를 구동할 때 설정하는 값은 다음과 같다. 여기에서 제어프레임(예를 들면 CCTS 프레임)은 해당 제어프레임을 기본전송속도로 전송할 때 소요되는 전송시간을 의미한다.

$$\text{타이머 1} : 2 * \text{SIFS} + \text{ARTS} + (\text{ARTS} + 2 * \text{SIFS}) * (\text{N}-1) + \text{CCTS} + (\text{CCTS} + 2 * \text{SIFS}) * (\text{N}-1)$$

$$\text{타이머 2} : 2 * \text{SIFS}$$

$$\text{타이머 3} : 2 * \text{SIFS} + \text{DATA} + (\text{DATA} + 2 * \text{SIFS}) * (\text{N}-1) + \text{ACK}$$

RCO-MAC 프로토콜에 따라서 도움노드에서 수행하는 절차를 그림 3에 나타내었다.

도움노드의 경우는 송신노드로부터 프레임을 수신하자마자 수행해야 할 역할이 시작된다. 우선 재전송 절차를 관리하기 위해 n 값을 0으로 설정하며, 수신한 프레임이 CRTS 프레임이면, 목적지노드에게 ARTS(acknowledge RTS) 프레임을 전송한 뒤 n 값을 증가시키며, 수신한 프레임이 DATA 프레임일 경우에는 목적지노드에게 수신한 DATA 프레임을 전달한 뒤, n 값을 1만큼 증가시킨다. 두 경우 모두 전송오류에 따른 재전송 절차를 관리하기 위해 타이머 2를 구동시키며, 타이머 2가 종료되기 이전에 목적지노드가 CCTS 프레임을 전송하는 것을 감지하거나, 타이머 2가 종료되기 이전에 목적지 노드가 ACK 프레임을 전송하는 것을 감지하면 전송한 프레임이 성공적

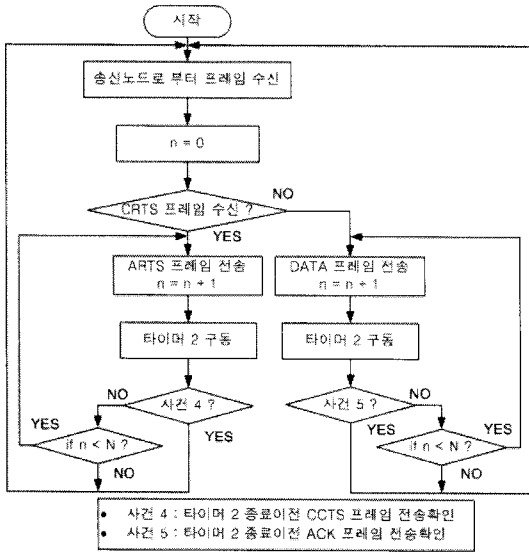


그림 3. 도움노드의 동작

으로 수신되었다고 판단하고 도움노드 역할을 마무리한다. 그러나 타이머 2가 종료되기 이전에 목적지노드가 해당 제어 프레임을 전송하는 것을 감지하지 못하면 재전송 절차를 수행한다.

마지막으로 목적지노드에서 수행할 절차를 그림 4에 나타내었다. 목적지노드도 도움노드로부터 프레임을 수신하자마자 목적지노드의 역할을 시작한다. 도움노드로부터 프레임을 수신하면 재전송 절차를 관리하기 위해 n 변수 값을 0으로 설정한 뒤, 수신한 프레임이 ARTS 프레임이면, 송신노드에게 직접 CCTS 프레임을 전송한 뒤, n 값을 1만큼 증가시키고 CCTS 프레임 재전송 절차 관리를 위해 타이머 2를 구동한다. CCTS 프레임 재전송 절차 관리는 도움노드에서의 ARTS 프레임 또는 DATA 프레임 재전송 관리 절차와 유사하다. 만일 도움노드로부터 수신한 프레임이 ARTS 프레임이 아닐 경우에는 DATA 프레임이므로, 송신노드에게 직접 ACK 프레임을 전송한다. 그러나 ACK 프레임을 전송한 후 송신노드로부터 이것에 대한 응답을 받을 수 없기 때문에 기본적으로 ACK 프레임은 재전송하지 않는다.

그러나 무선 채널 악화에 따른 전송오류가 빈번하게 발생할 경우에는 목적지노드가 송신노드에게 무조건 ACK 프레임을 2번 이상 전송하여 전송 신뢰성을 높일 수도 있다. RCO-MAC 프로토콜은 무선 랜에서 사용하고 있는 DCF 기법처럼 제어프레임은 기본전송 속도로 전송하며, DATA 프레임은 수신 SNR 값에 기반을 두어 변복조/코딩 기법을 결정하고, 전송속도를

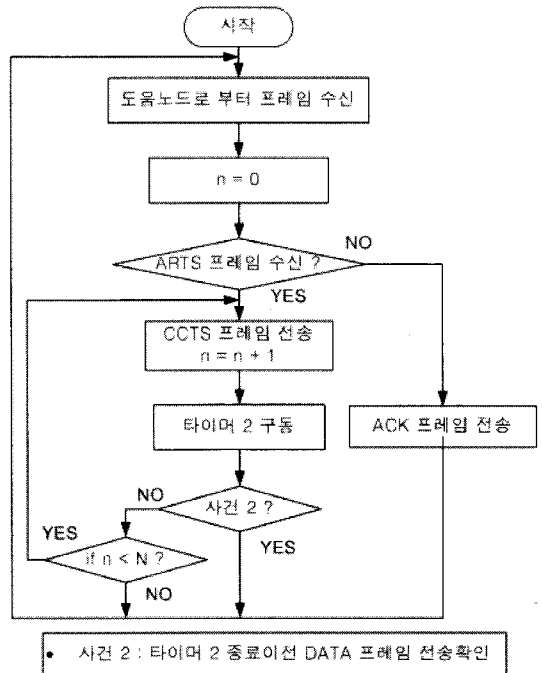


그림 4. 목적지노드의 동작

결정하는 링크 어댑테이션 기법에 따라 전송한다. 그리고 DATA 전송 대비 제어 프레임이 차지하는 비율을 줄이기 위해 CCTS 프레임과 ACK 프레임은 도움노드를 통하지 않고 목적지노드가 송신노드에게 직접 전송한다. 이것이 가능한 이유는 앞에서 설명한 바와 같이 제어프레임은 전송오류에 더 잘 견디는 변복조 기법을 사용하여 전송하기 때문이다.

IV. 성능평가 및 결과

본 논문에서 제안한 RCO-MAC 프로토콜의 성능을 분석하기 위해서 컴퓨터 모의실험을 사용하였다. 컴퓨터 모의실험은 SMPL 틀을 사용하였으며^[11], 전송할 데이터 프레임은 항상 존재하는 트래픽 포화상태를 가정하였다. 트래픽 포화상태는 참고문헌^{[6]-[8]}에서도 성능평가를 수행하기 위해 사용한 적이 있다. 성능평가에서 사용한 시스템 파라미터를 표 1에 나타내었다. 여기에서 R_{sh} 는 송신노드(S)와 도움노드(H) 사이의 전송속도를 의미하며 컴퓨터 모의실험은 1,000 초 동안 수행하였다.

성능평가 척도로는 시스템 처리량 (그림에서 average transmission rate 으로 표기)과 프레임 평균 지연시간 (그림에서 average access delay로 표기)을 사용하였으며, 시스템 처리량은 컴퓨터 모의실험을 수

표 1. 시스템 파라미터

시스템 변수	값	시스템 변수	값
CRTS 크기	352 bits	SIFS	10 μ s
ARTS 크기	352 bits	DIFS	50 μ s
CCTS 크기	304 bits	CWmin	32 slots
ACK 크기	304 bits	CWmax	1024 slots
DATA 크기	1024 bytes	R_{sh}, R_{hd}	11 Mbps
슬롯 시간	20 μ s	기본전송속도	1 Mbps
재전송 횟수 (N)	가변	프레임 전송오류(p_e)	0.1

행한 시간동안 성공적으로 전송한 총 데이터 양으로 정의하였으며 bps 단위로 나타내었다. 프레임 평균지연시간은 큐잉시간을 제외하고, 전송 시도 순간부터 성공적으로 전송이 완료될 때까지 기다린 시간으로 정의하였으며 버퍼 내 큐잉시간은 제외하였다. 프레임 전송오류 확률 p_e 는 프레임 길이와 사용한 변복조 기법에 따라서 다르기 때문에 제어프레임과 DATA 프레임을 다르게 설정해야 옳지만 본 논문의 성능평가에서는 간단한 경우를 우선적으로 살펴보기 위해 제어프레임과 DATA 프레임의 프레임 전송오류가 동일하다고 가정하였다. 이 전송오류확률은 특별하게 언급하지 않았을 경우에는 0.1이며, 각 그림에서 사용한 값을 그림 내에 표기하였다.

그림 5는 기존 rDCF 기법과 본 논문에서 제안한 RCO-MAC 기법의 시스템 처리량 성능을 비교하였다. 열악한 채널환경에 기인한 프레임 전송오류는 10% 정도 발생하는 환경을 고려하였으며, 전송범위 내 단말기 수의 변화에 따른 시스템 처리량을 도출하였다. 단말기 수의 변화에 따른 시스템 처리량의 변화는 참고문헌^{[9][10]}에서 제시한 결과와 유사한 형태임을 알 수 있다. rDCF 기법과 비교할 때 프레임 전송오류

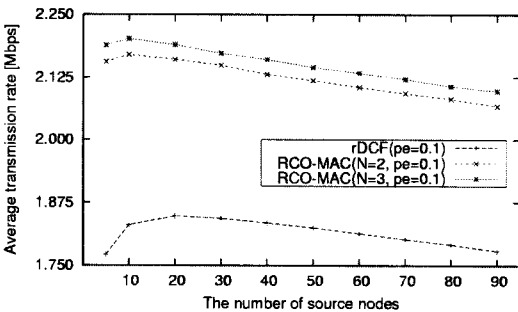


그림 5. 시스템 처리량 비교

가 10%인 채널환경에서는 시스템 처리량 성능이 약 24% 개선됨을 확인할 수 있다. 그리고 각 노드에서 수행하는 재전송 절차에서 재전송 횟수가 2회인 경우와 3회인 경우는 큰 차이가 없음을 알 수 있다. 즉 10% 정도의 재전송 환경에서는 각 노드에서 2회만 재전송해서 기존 방식에 비해 시스템 성능을 크게 증가시킬 수 있음을 확인할 수 있다.

한편 평균지연시간 측면에서의 성능평가 결과를 그림 6에 나타내었다. 먼저 전송범위 내 단말기 수가 상대적으로 적을 경우에는 그 차이는 미비하지만 본 논문에서 제안한 RCO-MAC 기법이 우수함을 알 수 있다. 그러나 단말기 수가 상대적으로 많을 경우에는 기존 rDCF 기법에 비해 평균지연시간이 조금 더 큼을 알 수 있었다. 이것은 단말기 수가 많을 경우에는 시스템 성능이 무선채널 악화에 따른 프레임 전송오류보다도 CRTS 프레임을 사용하여 채널경쟁을 하는 단계에서 발생하는 충돌에 의해 좌우되기 때문이다. 그리고 CRTS와 DATA 프레임을 전송한 후 송신노드가 운영하는 재전송 타이머의 설정시간 값이 각 노드에서 자체적으로 수행하는 재전송 절차 때문에 rDCF 기법보다 더 큰 값으로 설정하기 때문에 평균지연시간이 증가한다. 하지만 본 논문의 성능평가에서는 제외하였지만 만약 버퍼에서 대기하는 큐잉시간까지 고려한다면, 전체적인 시스템 평균지연시간은 감소할 것으로 예측된다. 왜냐하면 RCO-MAC 기법의 시스템 처리량이 기존 rDCF에 비해 24%정도 증가하였으며, 시스템 처리량이 증가하면 버퍼에서 기다리는 큐잉지연시간이 감소하기 때문이다.

그림 7에는 본 논문에서 제안한 RCO-MAC 프로토콜에서 최대 재전송 횟수를 3회로 고정된 뒤, 열악한 채널환경에 따른 프레임 전송오류 값의 변화에 따른 시스템 처리량의 변화를 비교하였다. 전송오류가 없을 경우 송신노드의 수가 5개 정도일 때 최대 2.7 Mbps 정도의 성능을 달성할 수 있음을 확인할 수 있

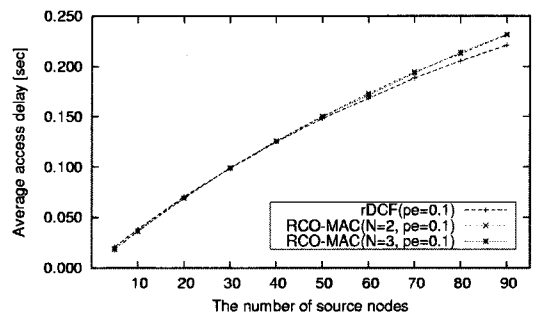


그림 6. 평균지연시간 비교

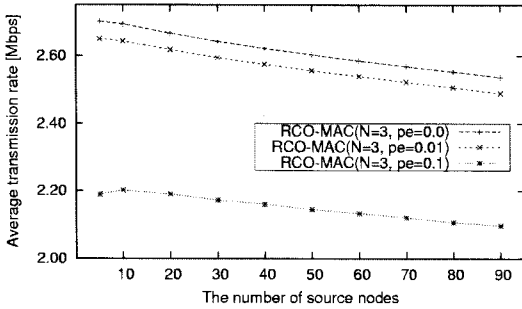


그림 7. 시스템 처리량 성능

으며, 전송오류가 증가함에 따라서 시스템 성능이 급격하게 감소함도 확인할 수 있다.

그림 8에는 전송오류 변화에 따른 평균지연시간 성능을 비교하여 나타내었다. 지금까지 제시한 수치계산 결과를 통해 살펴보면, 열악한 채널환경에 의해 발생하는 프레임 전송오류 때문에 발생하는 시스템 성능 감소 문제를 개선하기 위해 각 노드에서 수행하는 재전송 횟수의 최댓값 N 을 무작정 증가시키는 것이 적절한 해결책이 아님을 확인할 수 있다. 왜냐하면 N 값을 증가시키면 시스템 처리량은 개선되겠지만 평균지연시간이 증가하여 실시간 특성을 갖춘 트래픽을 수용할 수 없기 때문이다.

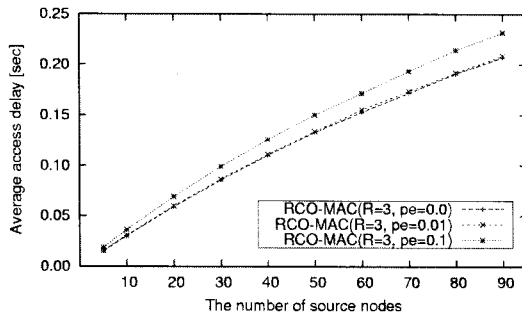


그림 8. 평균지연시간 성능

V. 결 론

기술의 발달로 오래전부터 무선통신시스템의 숙제였던 낮은 전송속도 문제가 많이 극복되고 있다. 그러나 송수신 노드가 멀리 떨어져 있거나 페이딩 현상이 발생하는 환경에서는 열악한 전송환경 때문에 필연적으로 전송속도가 감소한다. 본 논문에서는 이러한 환경에서 사용할 수 있는 새로운 협력통신용 MAC 프로토콜을 제안하였다.

컴퓨터 모의실험에 의한 성능평가 결과에 따르면

프레임 전송오류가 10%인 환경에서는 기존 rDCF 기법과 비교하여 시스템 처리량 성능이 약 24% 정도 개선됨을 확인하였으며, 평균 지연시간 측면에서도 단말기 수가 상대적으로 적을 경우에는 미약하지만 지연시간 성능이 개선됨을 확인할 수 있었다. 그러나 단말기 수가 상대적으로 클 경우에는 재전송 타이머의 설정 값이 크기 때문에 비록 규모는 작지만 지연시간 성능이 더 나빠짐을 확인할 수 있었다. 그러나 시스템 처리량 측면에서는 성능이 월등하게 우수하므로 버퍼에서 기다리는 큐잉지연 시간을 포함하여 지연시간 성능을 계산하면 rDCF 기법에 비해 우수할 것으로 예상된다.

차후 연구에서는 본 논문에서 제안한 RCO-MAC 기법을 다른 성능평가 방법을 통해서 계속 성능평가를 수행하면서 미흡한 부분을 보완할 계획이다.

참 고 문 헌

- [1] IETF MANET Working Group. Available from: <http://www.ietf.org/html.charters/manet-charter.html>.
- [2] A. Nosratinia, T. E. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Commun. Mag.*, Vol.42, No.10, pp.74-89, Oct. 2004.
- [3] IEEE Std 802.11-1997, Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, June 1997.
- [4] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks", in *Proc. of ACM/IEEE MOBICOM'2001*, Italy, June 2001.
- [5] Sai Shankar N, Chun-Ting Chou, and Monisha Ghosh, "Cooperative communication MAC (CMAC) - A new MAC protocol for next generation wireless LANs," in *Proc. of IEEE Int. Conf. on Wireless Networks, Communications and Mobile Computing*, Hawaii, June 2005.
- [6] H. Zhu and G. Cao, "rDCF: A relay-enabled medium access control protocol for wireless Ad Hoc Networks," *IEEE Trans. on Mobile Computing*, Vol.5, No.9, pp.1201-1214, September 2006.

- [7] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. S. Panwar, "CoopMAC: A cooperative MAC for wireless LANs," *IEEE JSAC*, Vol.25, No.2, pp.340-353, 2007.
- [8] K. Tan, Z. Wan, H. Zhu and J. Andrian, "CODE: Cooperative medium access for multirate wireless Ad Hoc network," in *Proc. of IEEE SECON'2007*, 2007.
- [9] 장재신, "Ad Hoc 네트워크에서 Cooperative MAC 프로토콜에 관한 연구," *한국해양정보통신학회논문지*, 13(8), pp.1561-1570, 2009년 8월.
- [10] 장재신, "네트워크 코딩 기능을 갖춘 협력통신용 MAC 프로토콜에 관한 연구," *한국해양정보통신학회논문지*, 13(9), pp.1819-1828, 2009년 9월.
- [11] M. H. MacDougall, *Simulating computer systems Techniques and tools*, The MIT Press, 1992.

장 재 신 (Jaeshin Jang)

정회원



1990년 2월 동아대학교 전자공학
학과

1992년 2월 한국과학기술원 전
기 및 전자공학과 석사

1998년 2월 한국과학기술원 전
기 및 전자공학과 박사

1998년 7월~2002년 2월 (주)

삼성전자 정보통신총괄 책임연구원

2008년 8월~2009년 7월 Iowa 주립대 방문연구원

2002년 3월~현재 인제대학교 공과대학 정보통신공
학과 부교수

<관심분야> WMAN, 무선 메쉬네트워크, 센서네트
워크, 무선인터넷, QoS 보장