

---

# 실제적인 트래픽 환경에서 애드 혹 네트워크용 협력통신 MAC 프로토콜 성능평가

장재신\*

Performance evaluation of cooperative MAC protocol at ad hoc networks  
under real traffic environments

Jaeshin Jang\*

## 요 약

본 논문에서는 기존의 RCO-MAC 협력통신용 프로토콜을 실제 환경에 더욱 근접한 모델을 사용하여 성능평가를 수행하였다. 트래픽 데이터가 지수함수 분포에 따라 발생하고, 발생한 데이터 패킷은 버퍼에 저장하는 모델을 사용하였으며, 성능평가 척도로는 시스템 처리량, 평균 액세스 지연시간, 그리고 평균 시스템 지연시간을 사용하였다. 성능평가는 컴퓨터 모의실험을 통해 수행하였으며 성능평가 결과로부터 RCO-MAC 프로토콜의 시스템 처리량 성능이 기존 rDCF 기법에 비해 18% 정도 우수함을 확인할 수 있었다. 그리고 기존연구에서 채택했던 트래픽 모델의 한계 때문에 확인하지 못했던 결과, 즉 RCO-MAC 프로토콜의 평균 시스템 지연시간 성능이 기존 rDCF 기법에 비해 우수함도 확인할 수 있었다.

## ABSTRACT

In this paper, the performance of a RCO-MAC protocol is evaluated using a different traffic model. Data frames are assumed to generate based on an exponential distribution from one source node and then directly to be stored into the input buffer in the source node. System throughput, average access delay and average system delay are used as a performance measure. Performance evaluation is carried out by the computer simulation. Numerical results show that the RCO-MAC scheme offers 18% higher system throughput than the rDCF scheme. In addition, the numerical results show that the RCO-MAC protocol also offers the better average system delay under any traffic load than that by the rDCF scheme, which was not obtained in the previous research due to the saturated traffic model.

## 키워드

협력통신, CRTS, 도움말노드, Ad Hoc 네트워크, MAC

## Key word

cooperative communications, CRTS, help nodes, ad hoc networks, MAC

---

\* 인제대학교 (icjoseph@inje.ac.kr)

접수일자 : 2010. 06. 03

심사완료일자 : 2010. 06. 20

## I. 서 론

에드 혹 네트워크는 기존 이동통신망 개념과는 달리 단말끼리 직접 통신이 가능한 망구조이다<sup>[1]</sup>. 최근에는 에드 혹 네트워크에서 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<sup>[2]</sup>. 협력통신용 MAC 프로토콜은 OSI 통신모델을 기준으로 제 2계층인 데이터링크 계층에 해당되며, 지금까지는 무선 랜 프로토콜인 IEEE 802.11 DCF (distributed coordination function) 프로토콜에<sup>[3]</sup> 기반을 두고 연구되고 있기 때문에 네트워크 계층에서 바라보면 아무 변화가 없기 때문에 투명하게 데이터 송수신 기능을 제공한다.

이러한 연구추세와 관련하여 저자는 협력통신용 MAC 프로토콜에 대한 연구를 이미 다수 수행하였다. 본 논문은 참고문헌<sup>[4]</sup> 연구결과의 확장으로써 채널오류에 강한 협력통신용 MAC 프로토콜인 RCO-MAC (reliable cooperative MAC) 프로토콜을 대상으로 좀 더 실제적인 통신환경에서 성능평가를 수행하였다. 참고문헌<sup>[4]</sup>에서는 시스템 최대 처리량을 계산하기 위해 트래픽 포화상태 모델을 사용하였다.

그러나 이 모델은 시스템 최대 처리량을 도출하는 경우에는 유용하지만 평균지연시간 성능을 제대로 도출할 수 없는 문제점을 안고 있다. 그래서 이전 연구결과에서는 평균 액세스 지연시간 성능측면에서 트래픽 부하가 상대적으로 클 경우 RCO-MAC 기법의 성능이 오히려 나빠짐을 확인했지만, 버퍼에서 대기하는 시간을 고려한 전체 평균 시스템 지연시간 성능을 계산하면 RCO-MAC 프로토콜의 성능이 기존 rDCF 기법에 비해 우수할 것이란 결과를 짐작하기만 했고, 결과를 보여주지는 못했다.

그래서 본 논문에서는 평균 시스템 지연시간 성능을 측정하기 위해 포화상태 모델이 아닌 좀 더 실제적인 트래픽 모델을 사용하여 성능평가를 수행하였다. 즉, 전송하고자 하는 트래픽 데이터 패킷은 지수분포에 따라서

계속 발생하며, 발생된 데이터 패킷은 먼저 각 송신노드가 보유하고 있는 입력 버퍼에 저장된다. 그리고 송신노드는 해당 버퍼 내에 데이터 패킷이 존재하면 CSMA/CA (carrier sense multiple access/ collision avoidance) 기법에 따라 전송절차를 수행한다.

본 논문의 구성은 다음과 같다. 먼저 II장에서는 협력통신 MAC 프로토콜과 관련된 기존 연구동향에 대해서 기술한다. III장에서는 RCO-MAC 프로토콜 동작을 자세하게 기술하며, IV장에서는 컴퓨터 모의실험에 의해 성능평가를 수행한 결과를 제시하고 기존 방법과 비교하여 성능이 증가함을 보여준 뒤, V장에서 결론으로 끝맺는다.

## II. 관련 연구동향

이제까지 저자가 파악한 바에 따르면 협력통신용 MAC 프로토콜에 대한 연구는 협력통신용 물리계층에 대한 연구에 비해 상대적으로 빈약하다. 이것은 열악한 무선채널 환경에서 시스템 처리량을 개선하기 위해 연구되고 있는 협력통신 기술이 물리계층 기법에 근간을 두고 있음을 의미한다. 여기에서는 최근까지 협력통신 MAC 프로토콜과 관련된 연구결과들에 대해서 살펴보고자 한다. 우선 참고문헌<sup>[5]</sup>에서는 무선 랜 프로토콜에서 링크 어댑테이션 기법을 사용할 수 있는 MAC 헤더구조를 중심으로 RBAR(receiver-based auto rate) 프로토콜을 제안하였다. 협력통신용 MAC 프로토콜에 대한 연구도 많이 진행되었는데, 우선 RBAR 기법에 기반을 두고 기존 무선 랜 표준을 최소로 변경하면서 협력통신을 지원할 수 있는 CMAC 기법이 제안되었다<sup>[6]</sup>. 이 연구에서는 두 노드 사이에서 전송오류가 발생하였을 때 재전송하는 절차와 또 CMAC과 채널코딩 기법을 통합한 FCMAC 기법도 제안하였다. 참고문헌<sup>[7]</sup>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<sup>[8]</sup>에서는 CoopMAC 기법을 제안하였는데, 이 기법은 rDCF 기법과 매우 유사하지만 도움노드를 선정하는 기준을 수학적식으로 깔끔하게 나타낸 점이 눈에 띈다. 참고문헌<sup>[9]</sup>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<sup>[10][11]</sup>에서는 채널경쟁단계에서 충돌에 의한 오류뿐만 아니라 열악한 무선채널에 의한 전송오류가 발생하는 환경에서 새로운 협력통신용 MAC 프로토콜을 제안하였다. 먼저 참고문헌<sup>[10]</sup>에서는 채널경쟁단계에서 수신 다이버시티 효과를 얻으면서 복잡 도는 rDCF와 동등한 수준을 달성하는 것을 목표로 CO-MAC 프로토콜을 제안하고 성능평가를 수행하였다. 또 참고문헌<sup>[4]</sup>에서는 협력통신용 MAC 프로토콜과 네트워크 코딩 개념을 결합한 NC-MAC 프로토콜을 제안하고 성능평가를 수행하였으며, rDCF 기법 대비 네트워크 코딩 이득이상으로 성능이 증가함을 보였다.

참고문헌<sup>[4]</sup>에서는 열악한 무선채널 환경 때문에 전송오류가 빈번하게 발생하는 환경에서 유용하게 사용할 수 있는 협력통신용 MAC 프로토콜인 RCO-MAC 기법을 제안하고 성능평가를 수행하였다. 성능평가척도로는 시스템 처리량과 평균 액세스 지연시간을 사용하였으며, 최대 시스템 처리량을 계산하기 위해 포화상태 트래픽 모델을 사용하였다. 포화상태 트래픽 모델이란 모든 송신노드는 항상 전송할 패킷을 가지고 있다고 가정한 모델로서 최대 시스템 처리량을 도출할 때 흔히 사용하는 모델이다. 그러나 이 모델을 사용하면 버퍼에서의 대기시간을 포함한 평균 전체 지연시간을 제대로 구할 수 없다. 왜냐하면 트래픽 포화상태란 채널 가동률(utilization)이 1에 근접하거나 육박한 경우이기 때문이다.

### III. RCO-MAC 프로토콜<sup>[4]</sup>

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 프레임을 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 전송오류에 대응할 수 있도록 동작한다.

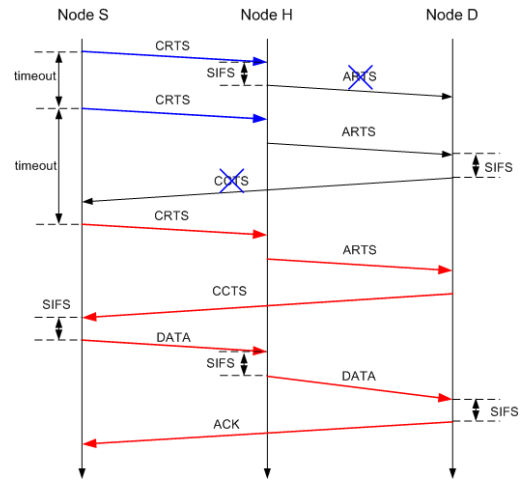
예를 들면(그림 1-(a) 참조), 도움노드(Node H)가 전송한 ARTS (acknowledge RTS) 프레임이 전송 중 오류가 발생하여 목적지노드(Node D)에서 제대로 수신하지 못했을 경우에는 송신노드(Node S)에서 운영하는 재전송 타이머 동작에 의해 CRTS(cooperative RTS) 프레임부터 재전송을 시도한다. 또 목적지노드에서 전송한 CCTS (cooperative CTS) 프레임이 전송 중 오류가 발생하였을 경우에도 동일한 동작에 의해 송신노드에서 CRTS 프레임을 재전송한다. 그러나 송신노드가 재전송 타이머의 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\*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\*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\*SIFS 만큼 기다리는 이유는, 목적지노드에서 SIFS 시간만큼 기다린 후 CCTS 프레임을 전송하기 때문에

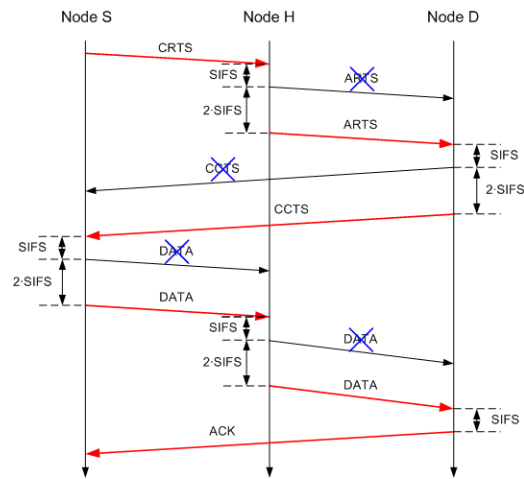
ARTS 프레임이 성공적으로 전송되었는지 여부를 판단하기 위해 가능한 짧은 시간도 충분한 시간이  $2 * SIFS$ 라고 판단하였기 때문이다.

본 논문에서는 도움노드를 선정하는 방법에 대해서는 기술하지 않았으며, 참고문헌<sup>[7]-[9]</sup>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, 랜덤값을 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 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.



(a)



(b)

그림 1. RCO-MAC 프로토콜 동작  
(a) rDCF 프로토콜 (b) RCO-MAC 프로토콜  
Fig. 1 Working procedure for RCO-MAC protocol  
(a) rDCF protocol (b) RCO-MAC protocol

DATA 프레임은 최대 N회까지 재전송할 수 있으며, 재전송 횟수를 관리하기 위해 변수 n을 사용하고, DATA 프레임을 전송하기 전에 이 값을 0으로 초기화한다.

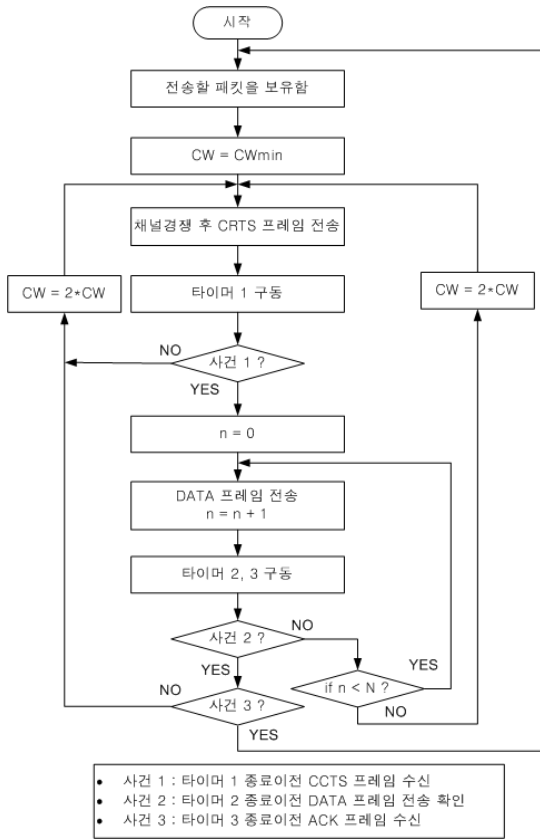


그림 2. 송신노드의 동작  
Fig. 2 Working procedure for source nodes

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 전송오류가 발생했다고 판단하고, DATA 프레임을 재전송한다. 만약 이 과정을  $N$ 회 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 전송오

류 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.

- 타이머 1:  $2 * SIFS + ARTS + (ARTS + 2 * SIFS) * (N-1) + CCTS + (CCTS + 2 * SIFS) * (N-1)$
- 타이머 2:  $2 * SIFS$
- 타이머 3:  $2 * SIFS + DATA + (DATA + 2 * SIFS) * (N-1) + ACK$

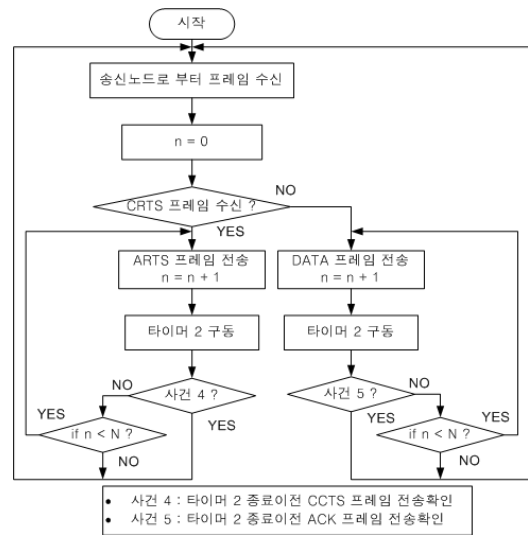


그림 3. 도움노드의 동작  
Fig. 3 Working procedure for helper nodes

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 프레임 전송하는 것을 감지하거나, 타이머 2가 종료되기 이전에 목적지 노드가 ACK 프레임을 전송하는 것을 감지하면 전송한 프레임이 성공적으로 수신되었다고 판단하고 도움노드 역할을 마무리한다. 그러나 타이머 2가 종료되기 이전에 목적지노드가 해당 제어 프레임을 전송하는 것을 감지하지 못하면 재전송 절차를 수행한다.

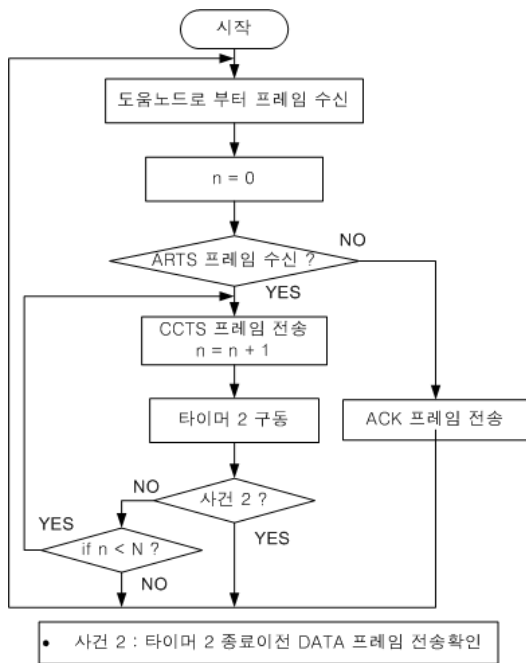


그림 4. 목적지노드의 동작  
Fig. 4 Working procedure for destination nodes

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 경우

에는 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 기본전송속도로 전송하며, DATA 프레임은 수신 SNR 값에 기반을 두어 변복조/코딩 기법을 결정하고, 전송속도를 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 논문에서는 참고문헌<sup>[4]</sup>에서 제안한 RCO-MAC 프로토콜의 성능을 컴퓨터 모의실험을 통해 평가하였다. 컴퓨터 모의실험은 C언어로 제공되는 SMPL 툴을 사용하였으며<sup>[12]</sup>, 성능평가 척도로는 시스템 처리량(system throughput), 평균 액세스 지연시간(average access delay), 평균 시스템 지연시간(average system delay)을 사용하였다. 시스템 처리량과 평균 액세스 지연시간은 참고문헌<sup>[4]</sup>에서 정의한 성능평가 척도와 동일하며, 평균 시스템 지연시간은 본 논문에서 새로 정의하였다. 시스템 처리량은 컴퓨터 모의실험 동안에 성공적으로 전송한 DATA 프레임의 양이며 단위는 Mbps이다. 평균 액세스 지연시간은 송신노드가 CRTS 프레임 전송시점부터 성공적으로 목적지노드로부터 ACK 프레임을 받을 때까지 소요된 시간이다.

한편 평균 시스템 지연시간은 평균 액세스 지연시간에다 버퍼에서의 대기 시간을 더한 값으로 송신노드가 상위계층으로부터 DATA 프레임을 수신한 시점부터 해당 프레임을 목적지노드에게 전송한 뒤, 목적지노드로

부터 ACK 프레임을 성공적으로 수신할 때까지 소요되는 시간을 의미한다. 트래픽 모델은 참고문헌<sup>[4]</sup>와 달리 송신노드에 DATA 프레임이 지수분포에 따라서 도착하는 모델을 사용하였으며, 상위계층으로부터 도착한 DATA 프레임은 해당 노드가 보유하고 있는 입력 버퍼에 곧바로 저장된다. 그리고 입력버퍼에 저장된 프레임은 FCFS(first come first service) 절차에 따라서 목적지노드에게 전송된다. 도착시간 간격의 평균값을 포함하여 성능평가에서 사용한 시스템 파라미터 값을 표 1에 나타내었다.

표 1. 시스템 파라미터  
Table. 1 System parameters

시스템 변수	값	시스템 변수	값
CRTS 크기	352 bits	SIFS	10 $\mu$ s
ARTS 크기	352 bits	DIFS	50 $\mu$ 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 $\mu$ s	기본전송속도	1 Mbps
재전송 횟수(N)	가변	프레임 전송오류(pe)	가변
데이터 프레임 도착간격	0.273 초	모의실험 수행시간	1000 초

여기에서  $R_{sh}$  는 송신노드(S)와 도움노드(H) 사이의 전송속도를 의미하며, 평균 DATA 프레임 도착간격 값은 1024 바이트 크기의 DATA 프레임이 90개 송신노드가 망 내에 존재할 때 전체 DATA 프레임 도착율이 2.7 Mbps 값이 되도록 선정하였으며, 2.7 Mbps 값은 참고문헌<sup>[4]</sup>에서 최대 성능처리량 값에 해당된다. 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.

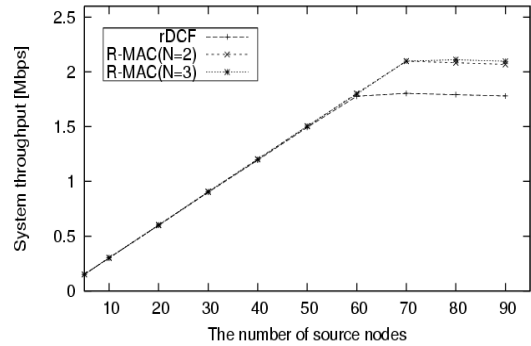


그림 5. 시스템 처리량 성능 비교  
Fig. 5 Performance comparison of system throughput

그림 5는 기존 rDCF 기법과 RCO-MAC 기법의 시스템 처리량 성능을 비교하였다. 열악한 채널환경에 기인한 프레임 전송오류는 10% 정도 발생하는 환경을 고려하였으며, 전송범위 내 송신노드 개수의 변화에 따른 시스템 처리량을 도출하였다. RCO-MAC 기법의 경우는 송신노드 수가 70개 정도일 때 최대 성능 값을 제공함을 알 수 있으며, 기존 기법인 rDCF는 60개에서 최대 성능 값을 가짐을 알 수 있다. RCO-MAC 기법에서 재전송 횟수가 2회인 경우와 3회인 경우는 큰 차이가 없지만 rDCF 기법과 비교하면 송신노드 수가 70개 이상인 구간에서는 대략 18% 성능개선이 발생함을 확인할 수 있다.

그림 6은 재전송 횟수가 3회인 RCO-MAC 기법에서 프레임 전송오류의 변화에 따른 시스템 처리량 성능을 보여준다. 프레임 전송오류 값이 작은 경우는 (0.01보다 작은 경우) 송신노드 수가 90개에 이를 때까지 시스템 처리량이 포화상태에 이르지 않고 계속 증가함을 알 수 있다. 그러나 프레임 전송오류가 0.1인 경우에는 송신노드 수가 70개인 지점에서 시스템 처리량이 최댓값을 가지며 그 이후에는 하락함을 알 수 있다.

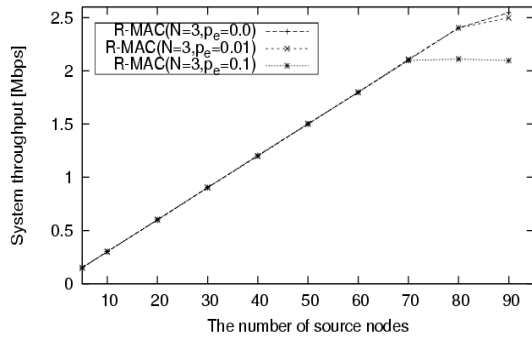


그림 6. 오류확률 변화에 따른 시스템 처리량 비교  
Fig. 6 Comparison of system throughput for various frame error probabilities

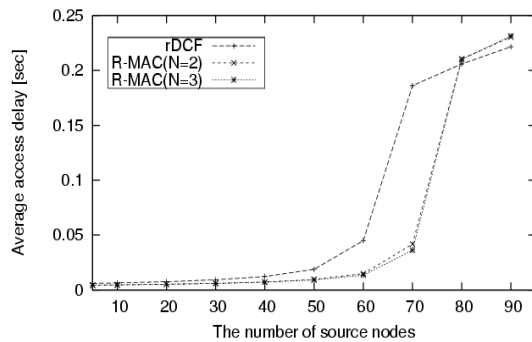


그림 7. 평균 액세스 지연시간 성능 비교  
Fig. 7 Performance comparison of average access delay

그림 7은 기존 rDCF 기법과 RCO-MAC 기법의 평균 액세스 지연시간 성능을 프레임 전송오류가 0.1인 무선 채널 환경에서 비교하였다. 참고문헌<sup>[4]</sup>에서 확인했던 바와 같이 트래픽 부하가 낮은 경우에는 프레임을 전송한 노드가 재전송을 담당하는 RCO-MAC 프로토콜의 지연시간 성능이 우수하지만 트래픽 부하가 증가하면(그림 7에서는 송신노드의 개수가 80개 이상인 경우) RCO-MAC 프로토콜의 지연시간 성능이 rDCF 기법과 비교하여 더 나빠짐을 확인할 수 있다. 이유는 트래픽 부하가 증가하면 충돌로 인한 전송오류가 차지하는 비율이 무선채널 악화로 인한 전송오류보다 증가하며, rDCF 기법이나 RCO-MAC 기법 모두 송신노드가 RTS 프레임 또는 CRTS 프레임을 재전송한다.

이때 재전송을 결정하는 사건은 재전송 타이머 종료이며 (RCO-MAC 기법의 경우는 타이머 1 또는 타이머 3이 여기에 해당), RCO-MAC 프로토콜의 경우는 해당 프레임을 전송한 노드가 해당 프레임에 대한 재전송 기능을 담당하기 때문에 타이머 1이나 타이머 3의 설정 값이 중간노드에서 재전송을 수행하지 않는 rDCF 기법에 비해 크다. 이 사실 때문에 충돌에 의한 프레임 전송오류가 증가하는 환경에서는 RCO-MAC 기법의 평균 액세스 지연시간이 rDCF보다 더 크게 된다.

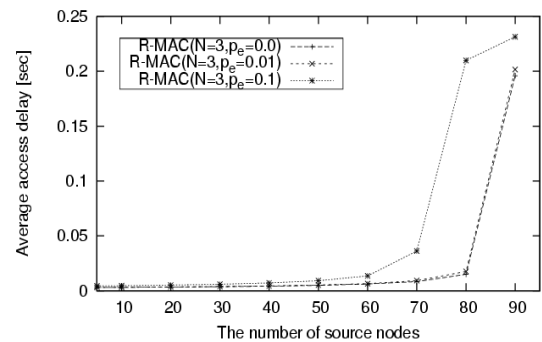


그림 8. 오류확률 변화에 따른 평균 액세스 지연시간 성능 비교  
Fig. 8 Comparison of average access delay for various frame error probabilities

그림 8은 프레임 전송오류 변화에 따른 RCO-MAC 기법의 평균 액세스 지연시간 성능을 비교한 것이다. 프레임 전송오류가 작을 경우에는 해당 프레임을 전송한 노드에 의해 짧은 시간 내에 재전송을 수행하므로 평균 액세스 지연시간이 작고, 프레임 전송오류 확률이 0과 0.01인 경우 값이 유사하지만, 프레임 전송오류가 0.1인 경우에는 평균 액세스 지연시간이 큰 폭으로 증가함을 보여 준다.

그림 9는 기존 rDCF 기법과 RCO-MAC 기법의 평균 시스템 지연시간 성능을 서로 비교하였다. 여기에서 프레임 전송오류 확률은 0.01인 무선채널 환경에서 전송범위 내 송신노드 개수의 변화에 따른 시스템 지연시간 성능을 도출하였다. 참고문헌<sup>[4]</sup>에서 예측했던 것처럼 버퍼에서의 대기시간을 포함한 시스템 지연시간을 도출한 결과 RCO-MAC 기법의 시스템 지연시간 성능이 rDCF 기법에 비해 우수함을 알 수 있다. 이것은 송신노드 수가 증가하는 경우에도 RCO-MAC 기법의 시스템



처리량 성능이 우수한 결과로부터 유추할 수 있었다. 이 결과는 트래픽 부하가 증가하는 경우 송신노드에서 CSMA/CA 기법의 절차에 따라서 프레임 전송하는 과정에서 전송오류에 의해 재전송 횟수가 지정된 최대값을 초과하여 전송되지 못하고 폐기되는 DATA 프레임의 개수가 RCO-MAC 기법에서 rDCF 기법과 비교할 때 감소함을 암시한다.

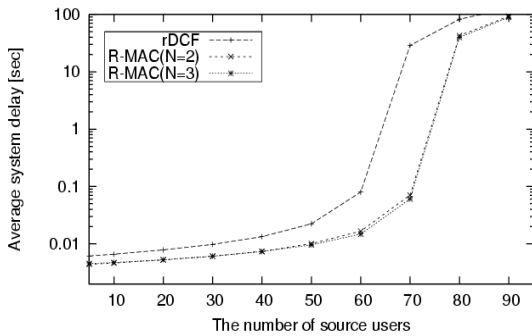


그림 9. 평균 시스템 지연시간 성능 비교  
Fig. 9 Performance comparison of average system delay

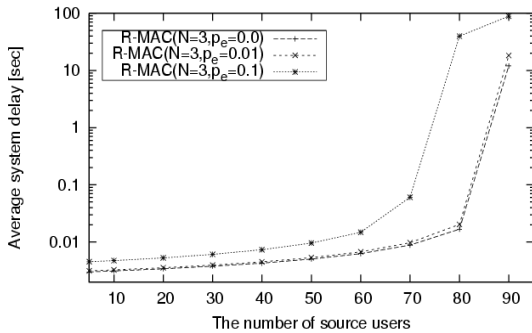


그림 10. 오류확률 변화에 따른 시스템 지연시간 성능 비교  
Fig. 10 Comparison of average system delay for various frame error probabilities

그림 10은 프레임 전송오류 변화에 따른 RCO-MAC 기법의 평균 시스템 지연시간 성능을 비교한 것이다. 프레임 전송오류가 작을 경우에는 해당 프레임을 전송한 노드에 의해 짧은 시간 내에 재전송을 수행하므로 지연시간이 작고, 프레임 전송오류 확률이 0과 0.01인 경우 값이 유사하지만, 프레임 전송오류가 0.1인 경

우에는 평균 시스템 지연시간이 큰 폭으로 증가함을 보여준다.

## V. 결 론

본 논문에서는 참고문헌<sup>[4]</sup>에서 저자가 제안했던 RCO-MAC의 지연시간 성능을 좀 더 자세하게 측정하기 위해 다른 트래픽 모델을 사용하여 성능평가를 수행하였다.

컴퓨터 모의실험에 의한 성능평가 결과에 따르면 프레임 전송오류가 10%인 환경에서는 기존 rDCF 기법과 비교하여 시스템 처리량 성능이 약 18% 정도 개선됨을 확인하였으며, 평균 지연시간 측면에서도 단말기 수가 상대적으로 적을 경우에는 미약하지만 평균 액세스 지연시간 성능이 개선됨을 확인할 수 있었다. 그러나 단말기 수가 상대적으로 클 경우에는 채널 액세스 과정에서 충돌에 의한 전송오류가 열악한 무선채널에 따른 전송오류보다 크며, RCO-MAC 기법의 경우 송신노드에서 운영하는 재전송 타이머의 설정 값이 크기 때문에 평균 액세스 지연시간 성능이 더 나빠짐을 확인할 수 있었다. 그러나 버퍼 대기시간을 포함하는 평균 시스템 지연시간 성능을 도출한 결과 rDCF 기법에 비해 우수한 것을 확인할 수 있었다.

차후 연구에서는 본 논문에서 제안한 RCO-MAC 기법을 대기이론을 이용하여 수학적으로 분석하는 연구를 수행할 계획이다.

## 참고문헌

- [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 ] 장재신, “채널오류에 강한 애드혹 네트워크용 협력 통신 MAC 프로토콜에 관한 연구,” 한국통신학회 논문지, 35(6), pp. 577-584, 2010년 6월.
- [ 5 ] 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.
- [ 6 ] 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.
- [ 7 ] 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.
- [ 8 ] 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.
- [ 9 ] 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.
- [10] 장재신, “Ad Hoc 네트워크에서 Cooperative MAC 프로토콜에 관한 연구,” 한국해양정보통신학회논문지, 13(8), pp. 1561-1570, 2009년 8월.
- [11] 장재신, “네트워크 코딩 기능을 갖춘 협력통신용 MAC 프로토콜에 관한 연구,” 한국해양정보통신학회논문지, 13(9), pp. 1819-1828, 2009년 9월.
- [12] M. H. MacDougall, Simulating computer systems: Techniques and tools, The MIT Press, 1992.

## 저자소개

### 장 재신 (Jaeshin Jang)



1990년 2월 동아대학교 전자공학과  
학사

1992년 2월 한국과학기술원  
전기및전자공학과 석사

1998년 2월 한국과학기술원 전기및전자공학과 박사

1997년 7월 ~ 2002년 2월 (주)삼성전자  
네트워크사업부 책임연구원

2008년 8월 ~ 2009년 7월 Iowa State University  
방문연구원

2002년 3월 ~ 현재 인제대학교 공과대학  
정보통신공학과 부교수

※관심분야: 이동통신망, 무선 랜, 무선인터넷, 센서  
네트워크, QoS 스케줄러