네트워크 코딩 기능을 갖춘 협력통신용 MAC 프로토콜에 관한 연구

장재신*

A study on the network coding enabled cooperative MAC protocol at Ad Hoc networks

Jae-shin Jang*

요 약

최근에는 Ad Hoc 네트워크에서 시스템 성능을 더욱 향상시키기 위해 협력통신에 대한 연구가 많이 진행되었다. 본 논문에서는 시스템 성능을 더욱 개선하기 위하여 협력통신을 지원하는 망구조에서 네트워크 코딩을 지원하는 NC-MAC 프로토콜을 제안하고 확률을 이용한 수학적인 접근방법을 사용하여 성능평가를 수행하였다. 도출된 성능평가 결과를 통해 IEEE 802.11 표준인 DCF 기법에 비해 시스템 처리량이 월등히 개선되며, 기존 연구에서 제안된 rDCF 기법 및 CO-MAC 기법보다도 최대 48%까지 시스템 처리량 성능이 증가함을 확인할 수 있었다.

ABSTRACT

Until now, much research on cooperative communications to increase system throughput at Ad Hoc networks has been provided. In this paper, in order to enhance system throughput much more at Ad Hoc networks, a network coding-enabled new MAC protocol, called NC-MAC protocol, is proposed and its performance is evaluated with a mathematical approach. Numerical results show that this scheme provides conspicuously enhanced system throughput compared to DCF scheme, and more increased system throughput by 48% than rDCF and CO-MAC schemes.

키워드

Ad Hoc 네트워크, 협력통신, NC-MAC, DCF, 도움노드, 네트워크 코딩

Ad Hoc networks, cooperative communications, NC-MAC, DCF, helper node, network coding

* 인제대학교 정보통신공학과

접수일자 2009. 03. 12
심사완료일자 2009. 07. 02

Ⅰ. 서 론

Ad Hoc 네트워크는 이동통신망이나 AP (access point)와 통신하는 무선 랜과는 달리 유사한 특징을 가진 모든 단말들이 무선채널을 통해 직접 통신하는 좀 더 진화된 통신망이라고 할 수 있다[1]. 그러나 Ad Hoc 네트워크를 포함한 모든 무선 통신망은 유선 통신망과 비교할 때 대역폭 제한이라는 치명적인 한계점을 안고 있다. 그래서 무선 통신망에서 해결해야 하는 중요한 과제 중의 하나는 시스템 성능을 증가시키는 것이며, 이를 달성하기 위해 데이터 링크 계층의 프레임 전송속도를 증가시키는 것이 중요하다. 프레임 전송속도를 높이기 위해서는 수신측에서의 SNR 값을 향상시키거나 수신기 성능을 향상시켜야 한다. 현재 물리계층 및 데이터 링크 계층으로 널리 사용하고 있는 표준이 IEEE 802.11 무선 랜이다. 이것은 CSMA/CA 프로토콜에 기반을 둔값을 향상기법이 침의 갬이 node 문제나 exposed node 문제를 원만하게 해결하면서 단일 홉에서 매우 우수한 성능을 제공하고 있기 때문이다.

WiBro 시스템, 무선 랜 시스템 등을 포함하여 최근 무선통신망에서는 시스템의 성능을 향상시키기 위하여 링크 어댑테이션 (Link adaptation) 기능을 사용한다. 이 기능을 실현하기 위해 물리계층에서는 다양한 변복조/채널코딩 기법을 제공하며, 채널상태에 따라서 적절한 채널 변복조/채널코딩 기법을 선택하여 사용하도록 하고 있다. 예를 들면, 채널상태가 양호할 때는 64QAM 변복조 기법을 사용하여 시스템 성능을 향상시키고, 채널상태가 불량할 때는 QPSK 변복조 기법을 사용하여 프레임 전송오류에 잘 견디도록 한다. 64QAM 변복조 기법을 사용하여 전송하면 QPSK 변복조 기법을 사용하여 전송하는 것보다 높은 SNR 환경에서 3배 더 많은 정보를 보낼 수 있다.

한편 무선통신망에서 링크 어댑테이션 기법에 기반을 두고 시스템 성능을 더욱 향상시키기 위해 Cooperation 통신 기법에 대한 연구가 많이 진행되고 있다. 좀 더 상세하게 설명하기 위해 그림 1과 같이 송신노드(N_s), 목적지노드(N_d), 도움노드(Helper node, N_h)로 이루어진 간단한 무선통신망을 고려하자. N_s에서 N_d로 직접 데이터를 전송할 경우에는 열악한 채널상태 때문에 1 Mbps로 전송할 수 있지만, N_s와 N_d 중간에 위치한 N_h가 프레임 중계기 역할을 수행할 경우, N_s와 N_h, 그리고 N_h와 N_d 사이는 11 Mbps로 전송할 수 있다고 하자. 이것처럼 N_h 노드를 통해서 N_s와 N_d가 통신을 한다고 하면, 평균전송속도는 $\frac{11 \cdot 11}{11+11} = 5.5\,\text{Mbps}$가 되며, 이 수치는 N_s와 N_d가 직접 통신할 때의 1 Mbps보다 더 큼을 알 수 있다.

즉, 채널상태가 우수한 도움노드의 도움을 받아서 채널 전송속도를 향상시키고, 패킷 전송시간을 감소시킴으로써 시스템 처리량을 향상시키는 접근방법을 협력 (Cooperative) 통신 이라고 한다.

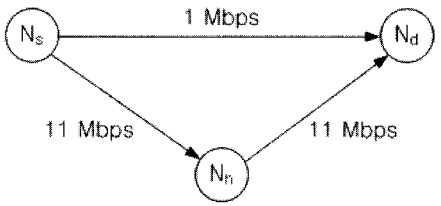

그림 1. 협력통신 예
Fig. 1 Example of cooperative communications

본 논문에서는 이러한 협력통신의 연구 추세에 발맞춰 무선통신망의 성능을 더욱 향상시키기 위해, 협력통신용 MAC 프로토콜로 참고문헌[9]에서 제안한 CO-MAC 기법을 기반으로 네트워크 코딩 기능을 갖춘 NC-MAC 기법을 제안하고 수학적 접근방법을 사용하여 제안된 프로토콜의 성능을 평가하고자 한다. 본 논문의 II장에서는 협력통신용 MAC 프로토콜과 네트워크 코딩에 관한 기존 연구동향을 기술하고, 본 논문에서 제안한 NC-MAC 프로토콜 기법을 III장에서 설명한 뒤, IV장에서 수학적 접근방법으로 제안된 기법을 분석하고, 수치계산 결과를 V장에서 제시한 뒤 VI장에서 끝맺는다.

Ⅱ. 관련 연구동향

현재 Ad Hoc 네트워크의 물리계층과 데이터 링크 계층으로 IEEE 802.11 무선 랜 표준[2]이 널리 사용되고 있으며, 차세대용 Ad Hoc 네트워크에서도 무선 랜

표준의 DCF 기법에 기반을 둔 프로토콜이 사용될 것으로 예상된다. 참고문헌[3]에서는 확률과 대기이론을 사용하여 DCF 기법의 성능평가를 수행하였다. 이 연구는 모든 노드들이 전송할 패킷을 항상 보유하고 있다는 포화상태에서 수행하였으며, 이진 지수함수 백오프 과정은 프레임 재전송 횟수가 최대 재전송 단계 (m)에 도달할 때까지 경쟁 윈도우 값을 이전 값보다 계속 2배씩 증가하도록 하였다. 그러나 이 모델은 최대 재전송 단계에 도달하고 난 뒤에도 재전송에 성공할 때까지 프레임을 계속 전송하는 문제점을 안고 있다.

참고문헌[4]에서는 참고문헌[3]이 안고 있는 문제점을 개선하여 IEEE 802.11 무선 랜 표준에 기술한 바와 같이 재전송 횟수가 최대 재전송 횟수(r)를 초과할 경우에는 재전송을 그만두고 해당 프레임을 폐기하도록 수정하였다. 참고문헌[3]과 참고문헌[4]에서 사용한 시스템 모델은 망 부하가 증가할 경우 패킷 처리량에서 큰 차이를 보임을 그림 2에 나타내었다. 이 결과는 본 논문에서 기본적으로 사용하고 있는 CO-MAC[9] 기법을 표 1에 기술한 시스템 환경에서 각각 모델링한 뒤, 성능 평가한 결과를 서로 비교한 것이다.

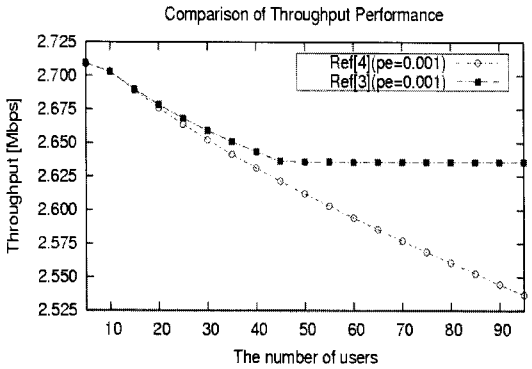

그림 2. 두 성능평가 모델 결과 비교
Fig. 2 Performance comparison of two models

무선 랜 표준에서는 채널 상태에 따라서 변복조 기법과 채널코딩 기법을 다르게 사용하는 링크 어뎁테이션 기능을 사용할 수 있다. 즉, 채널 상태가 양호할 경우에는 64QAM과 같은 대역폭을 효율적으로 사용하는 변복조 기법과 채널부호화 율이 높은 채널코딩 기법을 사용하여 전송속도를 높이고, 채널 상태가 양호하지 못할 경우에는 QPSK 기법과 채널 부호화 율이 낮은 채널코딩 기법을 사용하여 전송오류에 잘 견디도록 한다. 참고문헌[5]에서는 무선 랜 표준에서 링크 어뎁테이션을 사용하기 위해 필요한 프레임 규격과 절차를 제안하였다. 한편 무선 통신망에서 IEEE 802.11 표준의 기본 골격을 유지하면서 협력통신을 지원하는 MAC 프로토콜에 대한 연구도 많이 진행되었다[6]-[9].

참고문헌[6]에서는 CMAC 프로토콜을 제안하였으며, 오류제어기법과 결합한 FCMAC 프로토콜도 제안하였다. 이 연구결과는 현재 IEEE 802.11 무선 랜 표준의 변경을 적게 하면서 협력통신을 지원하는 MAC 프로토콜을 제안했다는 점에서 눈에 띄지만 도움노드를 선정하는 방법에 대해서 전혀 언급하지 않았다. 참고문헌[7]에서는 DCF 기법을 기반으로 협력통신을 지원하는 rDCF 기법을 제안하고 성능평가를 수행하였다. 이 연구에서는 도움노드를 설정하는 방법, 그리고 협력통신과 다중 전송속도를 구현하기 위해 기존 DCF에서 수정되어야 하는 부분들을 잘 언급하고 있다. 하지만 rDCF 기법을 수행하기 위해 변경되어야 할 제어프레임의 구조에 대한 설명이 없어서 성능평가에서 RRTS, RCTS 프레임이 전송해야 할 추가 정보들을 (N_s, N_r, N_d 사이의 전송속도 정보 등) 모두 고려했는지 확인할 수 없다. 참고문헌[8]에서 제안한 CoopMAC 프로토콜은 rDCF 프로토콜과 매우 유사하며, 도움노드를 선정하는 기준을 수식으로 표현한 것이 매우 주목할 만하다. 참고문헌[9]에서는 채널에서 프레임 전송오류가 발생할 경우 협력통신의 신뢰도를 높이기 위해 송신노드가 2개의 도움노드를 지정하여 RTS 제어프레임을 전송하는 CO-MAC 기법을 제안하였다. 이 기법은 rDCF 기법이나 CoopMAC 기법을 확장한 것으로 본 논문에서 제안하는 네트워크 코딩 기법은 CO-MAC 프로토콜에 기반을 두고 설계되었다.

한편 네트워크 코딩에 대한 연구도 많이 진행되었는데, 참고문헌[10]에서는 무선망에서 네트워크 코딩을 사용하기 위한 물리계층, 데이터 링크계층, 그리고 네트워크계층의 역할을 기술한 COPE 기법을 제안하였다. 이 기법에서는 모든 노드들이 인접노드가 전송하는 프레임의 헤더를 관찰하여 인접한 해당 노드들이 어떤 패킷을 보유하고 있는지 파악해야 한다. 파악된 정보는 reception report 메시지에 실어서 주기적으

로 인접노드들에게 방송하여 정보를 서로 공유한다. 시스템 성능을 높이기 위해 다수의 목적지노드에게 전송할 패킷을 보유하고 있는 송신노드는 한 번에 최대한 많은 패킷을 조합하여 전송할 수 있는 방법을 찾아내어야 한다. 그림 3에 나타낸 바와 같이 노드 B가 노드 A, C, D에게 전송할 패킷을 그림 3-(b)와 같이 보유하고 있고, 인코딩 기법은 Ex-OR 코딩 룰을 사용하는 경우를 생각하자. 먼저 노드 B가 P1과 P2 패킷을 결합하여 전송하는 경우, 노드 C는 P1을 사용하여 디코딩하면 P2를 수신할 수 있지만, 노드 A는 P2 패킷을 보유하고 있지 않기 때문에 P1 패킷을 디코딩할 수 없다. 만약 노드 B가 P1과 P3 패킷을 결합하여 전송하면, 유사한 절차에 따라서 노드 A와 C는 수신이 가능하지만 노드 D는 P4를 수신할 수 없다. 그러나 P1, P3 그리고 P4 패킷을 같이 결합하여 전송하면 노드 A, C, D가 모두 패킷을 수신하므로 성능향상을 달성할 수 있다.

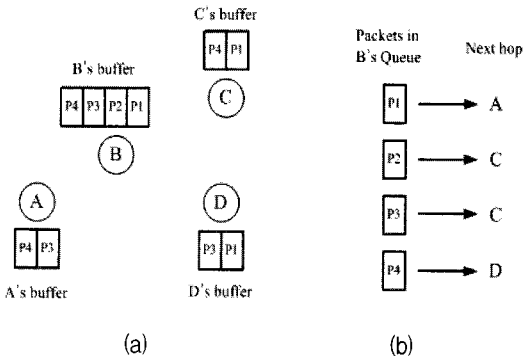

그림 3. COPE 기법의 네트워크 코딩 예
(a) B노드가 송신노드인 경우
(b) B노드 패킷의 목적지 노드
Fig. 3 Network coding example for COPE
(a) When node B is sender
(b) Destination nodes for the packet

그러나 그림 3과 같이 COPE 기법을 사용하여 네트워크 코딩을 구현하려면 사전에 모든 노드들이 Reception report 정보를 주고받아서 인접한 노드들이 보유하고 있는 패킷에 대한 정보를 알아야 하며, 송신노드 B가 Ex-OR 기법으로 결합된 패킷을 전송할 때 해당 패킷의 헤더에 어떤 패킷들이 결합되어 전송되는지 정보를 기록해야 한다. 후자에 대한 정보는 참고문헌[10]에서 기술하고 있지만, 전자에서 언급한 모든 노드들이 주기적으로 reception report를 교환하는 것이 망에 미치는 영향에 대해서는 참고문헌[10]에서 언급하고 있지 않다. 어쨌든 COPE 기법을 구현하기 위해서는 기존 표준을 크게 수정해야 하며, 주기적으로 교환해야하는 reception report 메시지 때문에 망에도 큰 부담이 예상된다. 그러나 본 논문에서 제안하는 NC-MAC 기법은 네트워크 코딩을 구현하기 위해 reception report 메시지를 사용하지 않기 때문에 망에 대한 부담이 없으며, 기존 협력통신용 MAC 프로토콜의 일부를 수정하여 구현할 수 있는 장점도 갖고 있다.

Ⅲ. 관련 연구동향

가. CO-MAC 프로토콜[9]

rDCF 프로토콜과 CoopMAC 프로토콜은 송신노드가 RTS 프레임을 전송할 때 한 개의 도움노드만 지정하여 전송하는데 비해 CO-MAC 프로토콜은 전송오류로 인해 RTS 프레임이 지정된 도움노드에게 제대로 전송되지 못하는 경우를 대비하여 두 개의 도움노드를 지정한다. 그렇지만 결국 협력통신에는 도움노드 1개가 참여하는 것이 특징이다. CO-MAC 프로토콜의 동작을 설명하기 위해 그림 4와 같은 다이아몬드 망 구조를 생각하자.

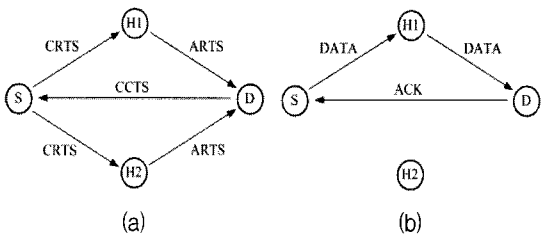

그림 4. CO-MAC 프로토콜 구성
(a) 1단계 전송 (b) 2단계 전송
Fig. 4 CO-MAC protocol
(a) 1st transmission (b) 2nd transmission

CO-MAC 프로토콜은 크게 2단계로 이루어져 있는데, 1단계는 CRTS (Cooperative RTS)와 ARTS (Ack RTS), 그

리고 CCTS (Cooperate CTS) 제어프레임을 주고받는 과정으로 이루어져 있다. 이때 IEEE 802.11 무선 랜의 CSMA/CA 기법을 사용하여 CRTS를 전송하고, 이 단계에서 도움노드를 선정한다. 후보 도움노드를 찾아내는 방법은 참고문헌[7][8]과 동일한 방법을 사용한다. 먼저 모든 통신노드들은 인접노드들이 전송한 모든 제어프레임들을 검사하여 해당 인접노드와 자신사이의 수신 SNR을 측정하고, 이를 기반으로 전송 가능한 전송속도를 계산하여 rate_info 테이블에 보관한다. 또한 각 노드는 직접 관리하고 있는 rate_info 테이블 내용을 주기적으로 Hello 메시지를 사용하여 인접노드들에게 전송하여 인접노드들과의 채널정보를 공유한다. Rate_info 정보를 수신한 각 노드는 이 정보를 기반으로 단일 홉(one hop) 또는 두 홉만큼 떨어져 있는 노드들과의 채널 정보(수신 SNR 또는 이에 해당하는 전송속도)를 help_list 테이블에 저장한다.

한편, 전송할 패킷이 상위계층으로부터 도착하면 송신노드는 먼저 help_list 테이블을 검색하여 목적지노드에게 직접 데이터 프레임을 전송할 때 소요되는 프레임 전송시간보다 도움노드를 통해서 목적지노드에게 전송할 때 프레임 전송시간이 짧은 도움노드 2개와 해당 링크간의 전송속도를 찾아낸다. 송신노드가 help_list 테이블을 검색한 뒤, 목적지노드와 관련하여 찾아낸 정보는 다음과 같다.

- 목적지노드 주소 (D)
- 도움노드1 주소 (H1)
- 도움노드2 주소 (H2)
- 송신노드와 도움노드1 간의 전송속도 (R_{sh_1})
- 송신노드와 도움노드2 간의 전송속도 (R_{sh_2})
- 도움노드1과 목적지노드 간의 전송속도 (R_{h_1d})
- 도움노드2와 목적지노드 간의 전송속도 (R_{h_2d})
- 송신노드와 목적지노드 간의 전송속도 (R_{sd})

CO-MAC 프로토콜을 사용하여 4개 통신노드가 제어프레임과 DATA 프레임을 주고받는 과정을 그림 5에 좀 더 자세하게 나타내었다.

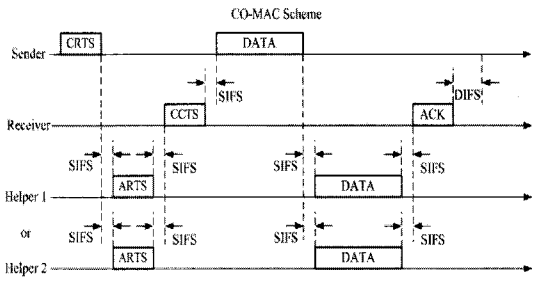

그림 5. CO-MAC 프로토콜의 프레임 전달과정
Fig. 5 Frame exchanges of CO-MAC protocol

한편 송신노드가 help_list 테이블로부터 찾아낸 정보를 CRTS 헤더에 저장한 뒤 도움노드에게 전송한다. 두 개의 도움노드 중에서 CRTS 프레임 헤더에 앞부분에 위치한 도움노드 (예를 들면, H1)가 묵시적으로 우선순위를 갖는다. CRTS 프레임을 수신한 노드 H1은 협력통신에 참여할 여유가 있으면, SIFS 이후에 ARTS 프레임을 목적지노드에게 전송한다. 만약 노드 H1이 여러 가지 이유로 (예를 들면, R_{sh_1}를 지원할 수 없는 경우) 협력통신에 참여할 수 없을 경우에는 ARTS 프레임을 전송하지 않는다. 그러면 두 번째 도움노드 H2는 이를 인지하고 곧바로 ARTS를 목적지노드에게 전송한다. CRTS 프레임을 수신한 도움노드는 수신 SNR 값을 측정한 뒤, CRTS 프레임에 기재되어 있는 R_{sh_i}를 제공할 수 있는지 확인하며, 더 높은 속도로 전송할 수 있으면 새로운 값을 ARTS에 기록한다. 한편 ARTS 프레임을 수신한 목적지노드는 수신 SNR 값을 측정한 뒤, R_{h_id} 값 또는 그 이상을 지원할 수 있는지 여부를 검사한다. 만약 지원 가능할 경우에는 가능한 R_{h_id} 값과 ARTS에 기재되어 있는 전송속도 값들을 기반으로 CCTS 프레임을 만들어서 SIFS 시간 이후에 송신노드에게 직접 전송한다. 한편 여러 가지 이유로 도움노드를 사용할 수 없다고 판단되거나, ARTS 프레임을 수신하지 못한 경우에는 SIFS 시간 이후 송신노드에게 CTS 프레임을 전송한 뒤 DCF 모드로 동작한다.

한편 목적지노드로부터 CCTS 프레임을 수신한 송신노드는 선정된 도움노드에게 DATA 프레임을 지정된 속도 (R_{sh_i})로 전송하며, 도움노드는 목적지노드에게 DATA 프레임을 R_{h_id}로 전송한다. 도움노드로부터

DATA 프레임을 수신한 목적지노드는 SIFS 시간 이후에 ACK 프레임을 직접 전송하여 2단계 데이터 전송을 모두 완료한다.

나. NC-MAC 기법 (CO-MAC + 네트워크 코딩)

본 논문에서 제안한 NC-MAC (Cooperative MAC with Network Coding) 프로토콜은 네트워크 코딩을 사용할 수 있도록 CO-MAC 프로토콜을 수정한 것이다. COPE 네트워크 코딩 프로토콜의 경우는 망 내의 모든 노드들이 인접노드들이 이미 보유하고 있는 패킷에 대한 정보를 reception report 메시지를 통해 사전에 확보해야 한다. 또 네트워크 코딩에 대한 정보를 전달하기 위해 네트워크 코딩된 패킷의 앞부분에 추가적으로 COPE 헤더를 부착해야 하기 때문에 망 부하 증가를 야기하면서, 또한 복잡하다. 하지만 본 논문에서 제안하는 NC-MAC 기법은 reception report 메시지를 사용하지 않으며, COPE 헤더와 같은 추가적인 헤더를 이용하지 않고, 단지 기존 MAC 헤더 내에 1비트만 추가로 할당할 수 있으면 구현이 가능하기 때문에 매우 간단한 것이 장점이다.

먼저 제어 프레임과 DATA 프레임 헤더 내에 NC 필드(Network coding indicator 1비트를 할당하여 1 (0)이면 사용 (사용하지 않음))함을 나타냄)를 할당하여 네트워크 코딩 사용 유무를 나타내도록 한다. 네트워크 코딩은 송신노드와 목적지노드가 모두 동의할 때만 사용하며, NC-MAC 프로토콜을 사용하여 통신하는 예를 그림 6에 나타내었다.

먼저 송신노드 S가 목적지노드 D에게 전송할 패킷이 발생하면, 헤더 내의 NC 필드를 1로 설정하여 CRTS 프레임을 전송한다. CRTS 프레임을 수신한 도움노드는 네트워크 코딩 모드로 동작함을 인지하고 ARTS 프레임의 NC 필드를 1로 설정한 뒤 목적지노드에게 전송한다. ARTS 프레임을 수신한 목적지 노드 D는 노드 S에게 전송할 패킷이 있으면 CCTS 프레임의 NC 필드를 1로 설정하여 전송하며, 전송할 패킷이 없을 경우에는 NC를 0으로 설정하여 CCTS 프레임을 전송한다.

송신노드는 NC 필드가 1인 CCTS 프레임을 수신하면 목적지노드가 네트워크 코딩을 수행하겠다는 것으로 판단하고 DATA1을 도움노드에게 전송한다. 이때 DATA1 프레임의 NC 필드를 1로 셋팅하여 도움노드이 네트워크 코딩을 수행하도록 한다. 또 다른 노드들이 데이터 전송 기간에 끼어드는 것을 막기 위하여 헤더 내의 Duration 필드 값을 다음과 같이 수정하며, DATA2의 크기는 CCTS를 통해서 받을 수 있다.

$$\text{Duration} = \text{DATA1} + \text{DATA2} + (\text{DATA1} \oplus \text{DATA2}) + 2\text{ACK} + 5\text{SIFS}$$

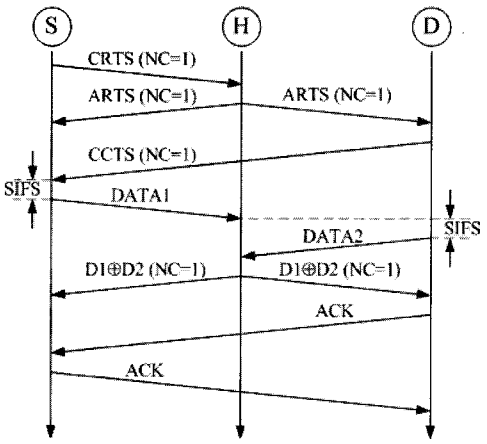

그림 6. NC-MAC 프로토콜 동작 예
Fig. 6 Example of NC-MAC

한편 NC 필드가 1인 DATA1 프레임을 송신노드로부터 수신하면 도움노드는 목적지노드가 DATA2를 전송하기를 기다린다. 목적지노드는 송신노드가 DATA1을 전송한 후 SIFS 시간만큼 기다렸다가 DATA2를 전송한다. 도움노드가 목적지노드로부터 DATA2를 수신하면 DATA1 프레임과 DATA2 프레임을 결합하여 DATA1⊕DATA2 프레임을 생성한 뒤, 해당 프레임을 송신노드와 목적지노드에게 방송한다. 이때 네트워크 코딩을 사용하여 결합된 프레임을 표시하기 위해 프레임 헤더에는 NC 필드를 1로 설정한다. 한편 결합된 프레임을 수신한 송신노드와 목적지노드는 이미 보관하고 있는 프레임을 사용하여 결합된 프레임을 디코딩할 수 있으며, 수신한 DATA 프레임의 오류여부를 체크하고 성공적으로 수신하였을 경우에는 ACK 프레임으로 응답하면 전체 동작이 마무리된다.

Ⅳ. 성능평가

본 논문에서 제안한 NC-MAC 프로토콜에 대한 성능평가는 참고문헌[4]에서 제안한 확률 기반의 수학적 기법을 사용하였다. 이 기법은 포화상태에서 시스템의 최대 성능을 구하기 위해 사용된 해석적 기법이며, 포화상태란 모든 단말의 전송 버퍼에 전송할 패킷이 항상 존재한다고 가정한 상태를 의미한다. 본 논문에서는 송신노드가 채널경쟁을 수행하기 위해 전송하는 CRTS 프레임에서만 전송오류가 발생한다고 가정하였다.

성능평가에 앞서 다음과 같은 몇 가지 변수를 정의한다.

- τ : 한 개의 노드에서 CRTS 프레임 전송 확률
- n : 송신노드로부터 송신도달거리 내에 있는 총 노드 수
- p_c : CRTS 프레임 전송 시 프레임 충돌 확률
- p_e : CRTS 프레임 전송 시 프레임 전송 오류 확률
- p_f : CRTS 프레임 전송 실패 확률
- m : 경쟁 윈도우 최대 단계, $\mathrm{CW}_{\max} = 2^m \mathrm{CW}_{\min}$
- r : 이진 지수함수 백오프 과정에서 최대 재전송 횟수

CRTS 프레임 전송을 실패하는 확률을 rDCF, CODE, 또는 CoopMAC 프로토콜에서는 $p_f'(=p_e+p_c-p_e \cdot p_c)$로 나타낼 수 있지만, CO-MAC 프로토콜에서는 CRTS 프레임 전송오류가 H1 노드와 H2 노드에서 모두 일어나는 경우에만 발생하기 때문에, $p_f(=p_e^2+p_c-p_c \cdot p_e^2)$로 나타낼 수 있다. CO-MAC 프로토콜에서 사용하는 경쟁 윈도우 값은 다음과 같은 관계를 갖는다.

$$W_i = \begin{cases} 2^i \mathrm{CW}_{\min}, & 0 \le i \le m \\ 2^m \mathrm{CW}_{\min}, & m < i < r \end{cases} \tag{1}$$

그러면 참고문헌[4]의 연구결과로 부터 한 노드에서의 CRTS 프레임 전송확률 τ을 다음과 같이 나타낼 수 있다.

$$\tau = \sum_{i=0}^{r} \alpha_{i,0} = \frac{1-(p_f)^{r+1}}{1-p_f} \alpha_{0,0} \tag{2}$$

여기에서 $\alpha_{i,j}$는 참고문헌[4]에 나타낸 2차원 천이확률 모델의 평형상태에서의 확률 값을 의미하며, 전체 확률의 합은 1이란 조건으로 부터 확률 $\alpha_{0,0}$을 구할 수 있다. 그러면 식 (2)와 또 다른 독립적인 식 (3)을 사용하여 CRTS 프레임 전송확률 τ을 구할 수 있다.

$$\begin{aligned} p_c &= 1-(1-\tau)^{n-1} \\ p_f &= p_e^2 + p_c - p_c \cdot p_e^2 \end{aligned} \tag{3}$$

위 두 식으로부터 구한 CRTS 프레임 전송확률 τ을 기반으로, 망 내에 n개의 통신노드가 CSMA/CA 기법에 기반을 두고 무선 채널 상에 CRTS 프레임을 전송할 확률 P_{tr}을 다음과 같이 구할 수 있다.

$$P_{tr} = 1-(1-\tau)^n \tag{4}$$

한편 채널에 CRTS 프레임을 전송했다는 조건 하에서 CRTS 프레임이 프레임 전송오류와 충돌 없이 성공적으로 전송될 확률 P_s는 다음과 같이 구할 수 있다.

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}}\left(1-p_e^2\right) \tag{5}$$

마지막으로 CRTS 프레임을 전송하였을 경우 성공적으로 전송되었을 때의 슬롯시간 T_s^{NC}, 그리고 충돌이나 전송오류 등으로 인해 실패했을 때의 슬롯시간 T_c^{NC}을 다음과 같이 나타낼 수 있다.

$$\begin{aligned} T_s^{NC} = {} & \mathrm{CRTS} + \mathrm{ARTS} + \mathrm{CCTS} + \mathrm{DATA}(\mathrm{L}, \mathrm{R_{sh}}) \\ & + \mathrm{DATA}(\mathrm{L}, \mathrm{R_{hd}}) + \mathrm{DATA}(\mathrm{L}, \min(R_{sh}, R_{hd})) \\ & + 2\mathrm{ACK} + 7\mathrm{SIFS} + 8\delta + \mathrm{DIFS} \end{aligned} \tag{6}$$

$$T_c^{NC} = \mathrm{CRTS} + \mathrm{DIFS} + \delta \tag{7}$$

식 (6)에서 $\mathrm{DATA}(\mathrm{L}, \mathrm{R_{sh}})$는 크기가 L인 DATA 프레임을 R_{sh} 속도로 전송할 때 소요되는 프레임 전송

시간을 의미하며, 제어프레임(예를 들면 CRTS)은 표 1에 기술한 크기의 제어프레임을 기본 속도 (Basic Rate)로 전송할 때 소요되는 전송시간을 의미한다. 그리고 δ는 전파지연시간으로 성능평가에서는 무시하였다. 따라서 네트워크 코딩을 사용한 NC-MAC 프로토콜의 시스템 처리량 성능은 식 (8)과 같이 나타낼 수 있다.

$$S = \frac{P_s P_{tr} E[P]}{(1-P_{tr})\sigma + P_s P_{tr} T_s^{NC} + P_{tr}(1-P_s) T_c^{NC}} \quad (8)$$

$$E[P] = \mathrm{DATA}(\mathrm{L}, \mathrm{R}_{\mathrm{avg}}) + \mathrm{DATA}(\mathrm{L}, \min(\mathrm{R}_{\mathrm{sh}}, \mathrm{R}_{\mathrm{hd}}))$$

위 식에서 R_{avg}는 $(R_{sh}+R_{hd})/2$을 나타내고, σ는 빈 슬롯시간의 크기를 나타낸다.

V. 성능평가 결과

성능평가 결과를 도출하기 위해 본 논문에서 사용한 시스템 파라미터를 표 1에 나타내었다. 성능평가를 위해 그림 4와 같이 송신노드, 목적지노드, 그리고 2개의 도움노드로 구성된 망을 고려하였다. 또, 송신노드와 목적지노드는 기본 전송속도 (Basic Rate)로 프레임을 전송하며, 송신노드와 도움노드, 그리고 도움노드와 목적지노드 간에는 11 Mbps로 전송한다고 가정하였다. 성능평가 척도로 사용한 시스템 처리량은 식 (8)에 NC-MAC, CO-MAC과 rDCF의 경우는 11 Mbps를 곱한 값을 사용하였으며, DCF의 경우는 송신노드와 목적지노드가 직접 통신하기 때문에 기본 속도인 1 Mbps를 곱한 값을 사용하였다.

표 1. 시스템 파라미터
Table. 1 System parameters

시스템 변수	값	시스템 변수	값
CRTS 크기	352 bits	SIFS	10 μs
ARTS 크기	352 bits	DIFS	50 μs
CCTS 크기	304 bits	CWmin	32 slots
ACK 크기	304 bits	CWmax	1024 slots
DATA 크기	1024 bytes	R_{sh}, R_{hd}	11 Mbps
Slot 시간(σ)	20 μs	Basic Rate	1 Mbps
RTS/CTS	160/112	r	6
		m	5

그림 7과 8에는 본 논문에서 제안한 네트워크 코딩 기능을 갖는 NC-MAC 프로토콜에 대한 성능평가 결과를 나타내었다. 먼저 그림 7에는 다양한 프레임 전송오류 환경에서 노드 수의 변화에 따른 시스템 처리 성능을 나타내었다. 그림으로 부터 망 내에 위치한 노드 수가 증가함에 따라 시스템 처리량이 감소함을 알 수 있는데, 이것은 본 성능평가에서는 모든 단말이 포화상태에 있다고 가정하였기 때문에 단말 수가 증가할수록 충돌확률이 증가하기 때문이다.

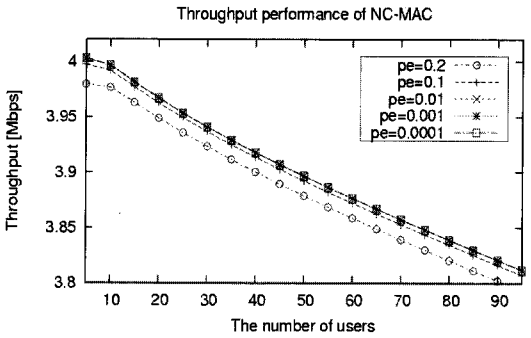

그림 7. 단말 개수에 따른 시스템 처리 성능
Fig. 7 System throughput for various number of terminals

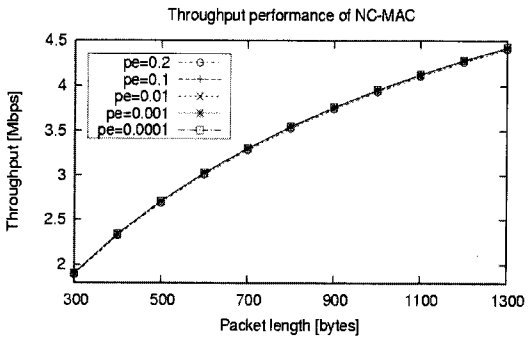

그림 8. 프레임 크기 변화에 따른 시스템 처리 성능
Fig. 8 System throughput for various frame sizes

그림 8은 노드의 수를 10으로 고정한 상태에서 DATA 프레임의 크기 변화에 따른 시스템 처리량을 나타낸다. DATA 프레임의 크기가 증가하면 전체적으로 제어 프레임이 차지하는 오버헤더 비율이 줄어들기 때문에 시

스템 처리량이 증가한다. 한편 프레임 전송오류 값 변화에 대해서는 시스템 처리량 변화가 거의 없는 것처럼 보이지만, 이것은 그림 7과 비교할 때 시스템 처리량의 변화폭이 프레임 전송오류에 따른 시스템 처리량 변화폭보다 훨씬 큰 이유에 기인한 것이며, 프레임 전송오류가 증가함에 따라 그림 7과 같이 시스템 처리량이 일부 감소함을 확인할 수 있었다.

그림 9에는 참고문헌[7]에서 제안한 rDCF 기법, 참고문헌[9]에서 제안한 CO-MAC 기법과 본 논문에서 제안한 네트워크 코딩 기법을 갖는 NO-MAC 기법과의 성능평가 결과를 비교하였다. 성능평가 결과를 통해 rDCF 및 CO-MAC 기법보다 시스템 처리량이 최대 48% 정도 개선됨을 확인할 수 있다. 성능평가에서 사용한 네트워크 구조에서 네트워크 코딩이득은 개략적으로 1.33 (=4/3)으로 33% 정도이지만, NC-MAC 기법에서는 제어프레임을 중복해서 사용하지 않기 이득이 48%로 증가한 것으로 예측된다.

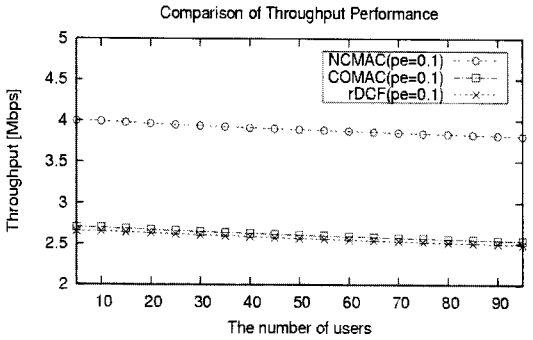

그림 9. 단말 개수 변화에 따른 성능비교
Fig. 9 Throughput comparison for various terminals

그림 10은 DATA 프레임 크기의 변화에 따른 시스템 처리량을 DCF[2], rDCF[7], 그리고 CO-MAC[9] 기법과 비교한 것이다. 이 결과에서 DCF가 상대적으로 성능이 매우 낮은 것은 성능평가에서 가정한 그림 4의 망구조에서 송신노드와 목적지노드 간의 직접경로가 상대적으로 멀어서 프레임 전송속도가 매우 낮은 1 Mbps 전송 환경을 고려했기 때문이며, 직접경로의 채널 환경이 개선되면 시스템 처리량 차이가 그림 10에 나타낸 것보다 줄어든다. 어쨌든 성능평가 결과를 통해 네트워크 코딩을 사용한 NC-MAC 기법이 기존의 다른 MAC 기법보다 시스템 처리량 측면에서 성능이 월등하게 우수함을 확인할 수 있다.

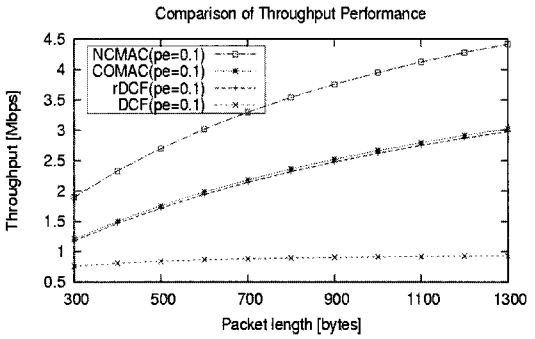

그림 10. 프레임 크기 변화에 따른 성능비교
Fig. 10 Throughput comparison for various frame sizes

VI. 결론

본 논문에서는 협력통신에 사용할 수 있는 CO-MAC 기법에 네트워크 코딩을 함께 구현한 NC-MAC 기법을 제안하고 수학적인 접근방법을 사용하여 성능평가를 수행하였다. 성능평가 척도로는 시스템 처리량을 사용하였으며, IEEE 802.11 규격의 핵심인 DCF 기법, 그리고 기존 연구에서 협력통신용 MAC 프로토콜로 제안된 rDCF 기법, 그리고 CO-MAC 기법과 성능비교를 수행하였다. 협력통신을 사용할 경우 DCF 기법에 비해서는 성능이 많이 개선되며, 프레임 전송오류가 큰 환경에서는 rDCF 기법에 비해 CO-MAC 기법이 높은 성능을 보이며, CO-MAC 프로토콜에 네트워크 코딩을 구현하였을 경우 그 성능이 최대 48%까지 개선됨을 확인할 수 있었다. 차후 연구로는 본 연구에서 제안한 프로토콜을 컴퓨터 모의실험을 통해 본 연구에서 얻지 못했던 좀 더 다양한 환경에서의 성능평가 결과를 도출할 계획이다.

참고문헌

[1] IETF MANET Working Group. Available from: <http://www.ietf.org/html.charters/manet-charter.html>.

[2] IEEE std 802.11-1997, part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, June 1997.

[3] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," IEEE J. of Select. Areas, pp. 537~547, March 2000.

[4] J. W. Tantra, C. H. Foh, and A. B. Mnaouer, "Throughput and delay analysis of the IEEE 802.11e EDCA saturation," in Proc. of IEEE ICC, June 2005.

[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 J. of Select. Areas, Vol. 25, No. 2, pp. 340 ~ 353, 2007.

[9] 장재신, "Ad Hoc 네트워크에서 Cooperative MAC 프로토콜에 관한 연구," 한국해양정보통신학회, 제13권, 제8호, pp.1561-1570, 2009.

[10] S. Katti, H. Rahul, D. Katabi, W. H. M. Medard, and J. Crowcroft, "XORs in the Air: Practical wireless network coding," in Proc. of ACM SIGCOMM, 2006.

저자소개

장 재신 (Jae-shin Jang)

1990년 2월 동아대학교 전자공학과 학사
1992년 2월 한국과학기술원 전기및 전자공학과 석사
1998년 2월 한국과학기술원 전기및전자공학과 박사
1997년 7월 ~ 2002년 2월 (주)삼성전자 네트워크사업부 책임연구원
2008년 8월 ~ 2009년 7월 Iowa 주립대 방문연구원
2002년 3월 ~ 현재 인제대학교 공과대학 정보통신공학과 조교수
※관심분야 : 이동통신망, 무선 랜, 무선인터넷, 센서네트워크, QoS 스케쥴러