

세션간 네트워크 코딩 기회를 인식하는 다중 경로 라우팅 프로토콜

최태종*, 강경란*, 조영종*, 방준호*

An Inter-Session Opportunistic Network Coding-aware Multipath Routing Protocol

Choi Tae Jong*, Kyungran Kang*, Young-Jong Cho, Bang, June-Ho

요약

무선 애드 hoc 네트워크는 제한된 에너지를 갖고 있는 노드들에 의해 자율적으로 구성되는 기반 구조가 없는 네트워크이다. 노드가 갖는 이런 제약사항은 데이터 전송 시 에너지 고갈로 데이터 전달이 갑작스럽게 실패하는 상황을 초래한다. 정적인 무선 애드 hoc 네트워크를 대상으로, 에너지 예약 기법 및 다중 경로 라우팅 기법을 이용하여 네트워크 내 노드들의 잔여 에너지가 적은 경우에도 안정적 데이터 전송을 지원하는 PAMP^[1]가 제안되어 있다. 본 논문은 PAMP에 다중 경로를 설정하는 과정에서 네트워크 코딩 기회가 있는 경로를 우선적으로 선정하여 중간 노드의 에너지를 절약할 수 있는 기법을 추가할 것을 제안한다. 시뮬레이션을 통한 성능 분석에서 본 논문에서 제안하는 기법이 기존의 PAMP와 센서 네트워크에서의 에너지 인식 다중 경로 설정 기법인 REAR에 비해 패킷 전송 성공률이나 노드의 에너지 소비에 있어서 우수한 성능을 보이고 있다.

Key words : ad hoc network, energy aware routing, multipath routing, network coding, inter-session

ABSTRACT

An ad hoc network consists of nodes with limited energy. Therefore, the data transmission can fail abruptly due to lack of energy of transmitting node. A previous work PAMP proposed to build multiple low-energy paths to support stable packet delivery exploiting the nodes with low energy. It has energy-reservation scheme and multi-path selection scheme for stationary wireless ad hoc networks. In this paper, we propose an extended version of PAMP^[1] by incorporating network coding opportunity in path selection process. The simulation results show that our proposed scheme shows better packet delivery ratio and lower energy consumption compared with PAMP and a legacy energy-aware multipath routing protocol REAR.

I. 서론

무선 메쉬 네트워크나 무선 센서 네트워크를 포괄하는 애드 hoc 네트워크에서는, 노드들의 제한적인 제한된 에너지로 인해, 데이터 전송 중 노드의 에너지

고갈로 데이터의 손실이 발생할 수 있는 문제점이 있다. 이런 문제점을 최소화하기 위해 경로 설정 시에 노드의 에너지를 설정 기준으로 활용하는 ‘에너지 인식 라우팅 프로토콜’들이 다수 제시되어 왔다. 에너지 인식 라우팅 프로토콜은 경로 상의 노드에서

※ 본 연구는 2010학년도 아주대학교 일반연구비 지원에 의하여 연구되었음.

※ 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(NIPA-2012-(H0301-12-2003)).

• 저자: 아주대학교 컴퓨터통신연구실, gz05__jj@nate.com, 준회원

* 아주대학교 컴퓨터통신연구실, korykang@ajou.ac.kr, 정회원, yjcho@ajou.ac.kr, 종신회원, june373@ajou.ac.kr

논문번호: KICS2011-12-624, 접수일자: 2011년 12월 20일, 최종논문접수일자: 2012년 7월 3일

소모하는 에너지의 양을 최소화하는 것에 목적을 두는 라우팅 프로토콜^[1,2]과 잔여 에너지가 많은 노드들을 중심으로 경로를 설정하여 네트워크의 운용시간을 향상시키는데 목적을 두는 라우팅 프로토콜^[3,4]로 나뉜다. 최근에는 잔여 에너지 정보를 알고 있는 다중 경로를 확보하거나 노드들의 잔여 에너지를 소스에서 필요로 하는 양만큼 예약하여 데이터 전달을 보장하고자 하는 방안들이 제안되었다.

무선 센서 네트워크를 대상으로 하는 REAR^[5]와 무선 단말 네트워크를 대상으로 하는 LAMOR^[6]를 노드들의 잔여 에너지를 고려하여 다중 경로를 설정하는 대표적인 사례로 들 수 있다. REAR는 잔여 에너지가 많은 노드들을 활용하여 주 경로를 설정하고, 노드들의 에너지가 고갈되는 상황을 예상하여 백업 경로를 설정해 두고, 주 경로의 데이터 전달이 실패하는 경우 백업 경로를 사용한다. 이런 기법은, 주 경로나 백업 경로 설정에서 에너지가 작은 노드들을 활용하지 못한다는 단점이 있으며, 백업 경로에 있는 노드들은 해당 세션이 종료될 때까지 다른 세션에서 사용할 수 없으므로 노드의 에너지를 효율적으로 활용하지 못하는 단점이 있다.

PAMP^[7]는 무선 센서 네트워크와 같이 정적인 무선 네트워크를 대상으로 낮은 에너지를 활용하기 위한 기법으로 제안되었다. 소스가 데이터 전송을 마치는데 필요한 에너지의 양을 예측할 수 있다는 가정하에, 가용 에너지가 작은 노드들을 활용하여 소스에서 목적지까지 k 개의 다중 경로를 설정한다. k 개의 경로 중 어느 하나만 사용하면 전체 데이터 전송을 마칠 수는 없으나, k 개의 경로를 모두 활용하면 가능하도록 다중 경로를 설정한다. 네트워크 내 각 노드들의 잔여 에너지가 낮더라도 이들을 모두 활용하여 경로를 설정할 수 있으므로 전체 네트워크의 지속 시간을 늘일 수 있다는 장점이 있다.

본 논문에서는 PAMP에 네트워크 코딩 기법을 추가 적용하여 성능을 추가 향상시키는 방안을 제안한다. 네트워크 코딩^[8]은 소스에서 목적지까지 다중 홉으로 전송되는 네트워크 환경에서 통신 자원을 효율적으로 사용함으로써 수율 (throughput) 향상, 지연 시간 감소, 그리고 에너지 절감 등의 이득을 얻을 수 있다고 이미 알려져 있다^[9,10].

그림 1이 네트워크 코딩의 이득의 예를 보여주고 있다. (b)에서 라우터는 A와 B로부터 수신된 x_1 과 x_2 를 XOR하여 하나의 패킷 $x_1 \oplus x_2$ 로 생성하고 이를 브로드캐스트하고 있다. 코딩된 패킷을 수신한 A와 B는 자신이 저장하고 있는 x_1 과 x_2 를 이용하여 디코

딩을 수행함으로써 상대방의 패킷을 획득할 수 있다. 이와 같이 네트워크 코딩은 무선 채널에서 브로드캐스트를 이용하여 이득을 얻을 수 있다.

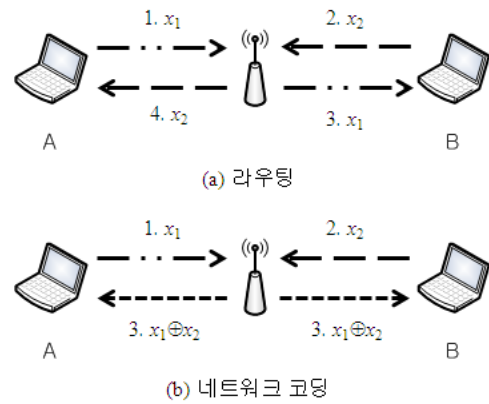


그림 1. 네트워크 코딩 이득 사례
Fig. 1. An example of network coding gain

본 논문에서 제안하는 기법은, PAMP를 기반으로 하되, 네트워크 코딩에 의한 이득을 얻기 위하여, PAMP의 다중 경로 설정 과정을 변경한다. 이미 설정되어 있는 경로 정보를 활용하여, 그림 1에 제시된 것과 같은 네트워크 코딩 이득을 볼 수 있는 경로를 우선적으로 선정한다. 경로 설정 요청 메시지가 소스에서 목적지까지 전달되는 과정에서, 네트워크 코딩 이익을 볼 수 경로가 있다면, 힌트 정보를 경로 설정 요청 메시지 내에 추가하여 목적지에 전달한다. 목적지에서는 최종 경로를 결정할 시에 네트워크 코딩 이익을 볼 수 있는 경로에 높은 우선순위를 부여한다. 제안하는 기법은 세션 간 발생하는 코딩 이득뿐만 아니라 네트워크 전체에서 발생하는 코딩 이득을 통해서 노드의 효율적 에너지 사용 및 네트워크의 운용 시간을 최대화할 수 있다.

본 논문의 구성은 다음과 같다. II장에서 본 논문의 기반이 되는 PAMP를 소개하고, 네트워크 코딩의 이익을 극대화하기 위한 데이터 전달 기법의 대표적인 사례인 COPE^[11], 그리고 네트워크 코딩 기회를 인식한 경로 설정 기법 DCAR^[12]을 소개한다. III장에서는 본 논문에서 제안하는 네트워크 코딩 기회를 활용하도록 PAMP를 확장하기 위해 고려해야 할 기술적인 요소들을 제시한다. IV장에서는 구체적으로 제안하는 기법을 소개하고 V장에서는 제안하는 기법의 성능 분석 결과를 제시한다. VI장에서는 향후 연구 방향을 제시하며 결론을 맺는다.

II. 관련 연구

2.1. PAMP

PAMP 이전에 제시된 에너지 인식 다중 경로 라우팅 프로토콜의 에너지 예약 기법은 경로 설정이 완료되지 않은 플로우에서 사용하게 될 에너지를 고려하지 않았다^{5,6}. 이는 다수의 RREQ 처리 시 노드가 예약할 수 있는 에너지보다 많은 에너지를 할당하는 에너지 과다 할당 문제를 발생시킨다. PAMP는 RREQ 처리 후 해당 경로에 지원해야 할 에너지를 예약하도록 하여 경로 설정이 완료되지 않은 세션에 대해서도 에너지 사용을 예측할 수 있도록 하였다.

PAMP에서는 노드가 복수개의 RREQ를 처리할 수 있도록 허용함으로써 가용 에너지가 많은 노드는 경로에 더 많이 참여할 수 있도록 하였다. 이를 위해서 PAMP에서는 구간 시퀀스 번호를 도입하였다. 구간 시퀀스 번호는 RREQ를 전달한 노드가 자신이 처리한 RREQ에 의한 경로 설정을 인식하기 위해 사용된다. 에너지 예약 기법은 일정량의 에너지를 경로에 할당하여 고정된 경로를 설정하는 기법이므로 각 노드는 자신이 지원하고 있는 각 경로에 구간 시퀀스 번호를 부여함으로써 여러 경로를 지원하더라도 각 경로를 분별할 수 있다. 이를 통해서 RREP가 경로 정보를 포함하여 전송될 필요 없이 중간 노드들이 수신된 RREP를 올바르게 인식하여 데이터 전달 경로와 에너지 할당량을 결정할 수 있다.

소스의 데이터 전달에 필요한 충분한 복수개의 경로를 설정하기 위해, 각 RREQ는 거쳐 온 노드들의 잔여 에너지 중 가장 낮은 에너지 값을 담고 있도록 한다. RREQ를 수신한 노드의 잔여 에너지가 기존 RREQ에 기록된 값보다 작으면 RREQ를 다음 노드로 전달하기 이전에 이 값을 자신의 잔여 에너지량으로 변경해야 한다.

그림 2는 PAMP의 라우팅 테이블 예제를 보여주고 있다. 구간 시퀀스 번호는 ISN, OSN 그리고 E_r 의 구성되는데 ISN은 이전 노드가 경로를 인식하기 위해 부여한 번호, OSN은 RREQ를 전달할 노드가 경로를 인식하기 위해 새로 부여한 번호, 그리고 E_r 는 노드가 경로에 할당한 에너지를 나타낸다. 각 노드는 그림과 같이 라우팅 테이블을 유지함으로써 동일 플로우의 RREQ를 수신하더라도 자신이 처리하지 않은 RREQ는 가용 에너지 내에서 추가적으로 경로를 지원할 수 있다.

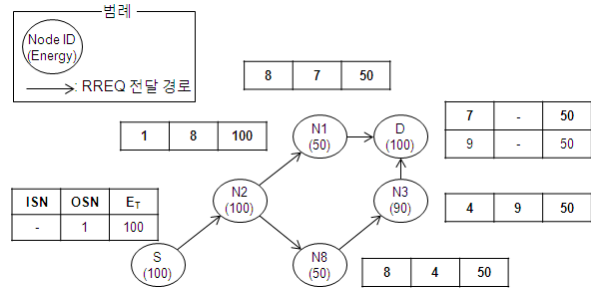


그림 2. PAMP의 라우팅 테이블
Fig. 2 A routing table of PAMP

그림 3은 PAMP의 에너지 예약 기법 예제를 보여주고 있다. 소스 S와 목적지 D 사이에는 세 개의 경로가 존재한다. 소스가 전송할 데이터의 에너지 요구량으로 100J을 요청할 때 단일 경로를 통해서는 데이터 전송 지원이 불가능하다. 이때 그림과 같이 PAMP의 에너지 예약 기법은 다중 경로의 예약된 에너지의 합이 소스의 에너지 요구량을 만족할 수 있도록 RREP가 소스 노드로 전달되는 과정에서 에너지 예약을 확정한다.

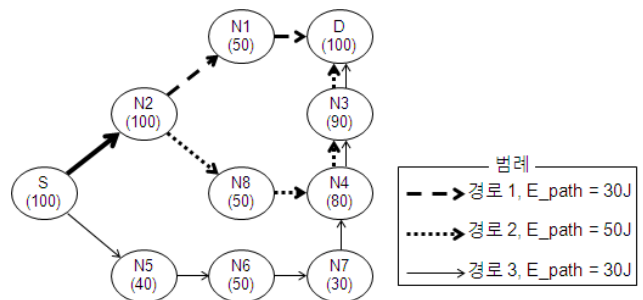


그림 3. PAMP의 에너지 예약 기법
Fig. 3. An energy reservation of PAMP

목적지에서는 경로 설정 시 RREQ가 통과한 경로상의 중간 노드들이 지원할 수 있는 에너지에 기반을 두어 소스의 에너지 요구량을 만족할 수 있도록 선택적으로 다중 경로를 설정한다. 이를 통해서 잔여 에너지가 적은 환경에서 다중 경로 설정을 이용하여 안정적인 데이터 전송을 지원할 수 있다.

2.2. COPE

COPE^[11]는 무선 매쉬 네트워크에서 네트워크 코딩을 사용하여 수율을 향상시킨 네트워크 코딩 시스템이다. COPE는 IP 계층과 MAC 계층 사이에 코딩 계층을 추가하여 코딩 기회를 탐지하고 이를 이용하여 한 번의 전송으로 다수의 패킷을 전달할 수 있도록 하였다. COPE는 감청, 기회적 코딩, 그리고 이웃

노드의 정보 획득 등의 세 가지 기법을 통해서 동작한다. 각각의 노드는 Promiscuous 모드를 사용하여 자신의 이웃 노드들이 전송하거나 수신하는 메시지를 감청한다. 이웃 노드의 정보 획득을 위해 일정 시간을 주기로 Reception Report를 교환한다. 기회적 코딩 기법은 수집된 이웃 노드의 정보를 이용해서 한 번의 전송으로 다수의 패킷을 전달할 수 있도록 이웃 노드들이 디코딩 가능한 선에서 최대한 많은 패킷을 코딩하여 전송한다.

COPE는 세 가지 시나리오에서 코딩 이득을 갖게 됨을 보였는데^[11] 감청 없이 코딩 이득을 갖는 경우, 감청을 통한 코딩 이득을 갖는 경우, 그리고 부분적 감청을 통한 코딩 이득을 갖는 경우이다. 그림 4에서는 COPE의 세 가지 코딩 이득을 보여주고 있다. (a)에서는 S1과 S2가 서로 자신이 갖고 있는 패킷 x 와 y 를 교환하는 상황을 보여준다. COPE의 기회적 코딩 기법을 통해 S1과 S2는 C로부터 코딩된 패킷 $x \oplus y$ 를 수신하고 이를 자신이 전송하였던 패킷과 디코딩을 수행함으로써 상대방의 패킷을 획득할 수 있다. 이때 발생하는 코딩 이득은 감청을 하지 않아도 획득할 수 있다.

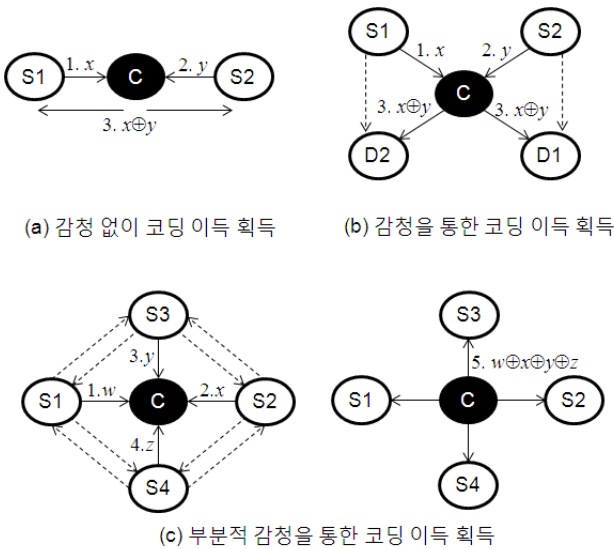


그림 4. COPE의 코딩 이득들
Fig. 4. Coding gains of COPE

(b)에서는 S1은 D1에게 패킷 x 를 S2는 D2에게 패킷 y 를 전송하는 상황을 보여준다. 만일 D2가 S1의 패킷 x 를 D1은 S2의 패킷 y 를 감청할 수 있다면, 코딩 포인트 C에서는 x 와 y 를 코딩하여 전송함으로써 코딩 이득을 획득할 수 있다. 이때 발생하는 코딩 이득은 감청을 통해서만 한다. 마지막으로 (c)

에서는 S1, S2, S3, 그리고 S4가 자신이 가진 패킷 w, x, y , 그리고 z 를 네트워크 내 자신을 제외한 다른 노드들에게 전송하려는 상황을 보여준다. 이때 각 노드에서는 몇몇 패킷은 감청을 통해 수신되고, 몇몇 패킷은 코딩을 통해서 수신된다. 예를 들어 S1의 경우, 패킷 y 와 z 는 감청을 통해서 수신할 수 있으며 패킷 x 는 코딩된 패킷을 디코딩함으로써 획득할 수 있다.

2.3. DCAR

DCAR^[12]는 소스로부터 목적지까지 데이터 전송이 가능한 경로와 어떤 경로를 통해 네트워크 코딩이 가능한 지 동시에 탐색할 수 있는 라우팅 프로토콜이다. 또한 두 홉 이상의 잠재된 코딩 기회를 갖는 경로를 탐색할 수 있도록 설계함으로써 COPE의 한계를 극복하였으며 보다 높은 수율을 갖도록 보장한다. 새로운 Metric인 CRM (Coding-aware Routing Metric)을 제시하여, 코딩 가능한 경로와 코딩이 불가능한 경로의 성능을 비교하고 성능이 우위인 경로를 선택하는 것을 원칙으로 한다.

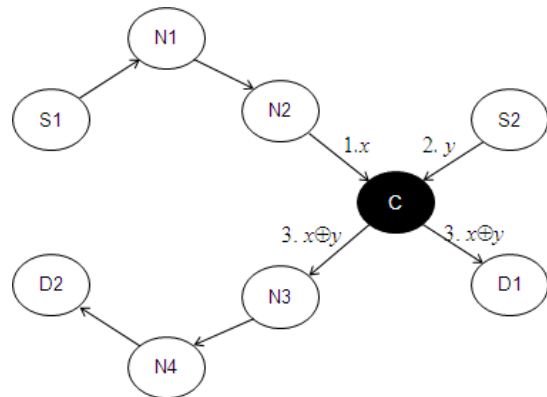


그림 5. 두 홉 이상의 잠재된 코딩 기회
Fig. 5. A potential coding opportunity beyond the two hop region

그림 5에서는 COPE에서 갖지 못하는 두 홉 이상의 잠재된 코딩 기회를 보여주고 있다. S1은 D1에게 패킷 x 를 S2는 D2에게 패킷 y 를 전송하는 상황을 보여준다. S1의 패킷 x 는 S1-N1-N2-C-D1을 통해 전송되며, S2의 패킷 y 는 C-N3-N4-D2를 통해 전송된다. 이때 D2는 S1의 패킷을 감청할 수 있으므로 C에서 패킷이 코딩되어 전달되더라도 디코딩을 수행할 수 있다. 그러나 COPE의 경우, C에서 이웃 노드의 정보 획득을 수행하면 N3에서는 패킷 x 를 감청하지 못하였으므로 코딩 불가능하다고 판단하여 잠재된 코딩 기회를 갖지 못한다. DCAR는 “General

Coding Condition"을 통해서 이런 기회를 섹출할 수 있도록 설계하였다.

Ⅲ. 기술적 고려 사항

3.1. 네트워크 코딩 인식 경로 탐색

네트워크 코딩을 통한 에너지 활용을 위해서는 먼저 어떤 경로를 통해서 네트워크 코딩 가능한지 탐색하는 방법이 필요하다. 이를 위해서 다음 네 가지 사항이 고려되어야 한다.

- 1) 어떤 토폴로지를 네트워크 코딩의 타겟으로 고려할 것인가
- 2) 네트워크 코딩을 할 수 있는 패킷들이 만족해야 할 조건은 무엇인가
- 3) 해당 경로가 네트워크 코딩이 가능한 경로임을 어떻게 목적지에게 알릴 것인가
- 4) 코딩 포인트의 이웃 노드들에게 디코딩을 위한 사전 동작을 어떻게 지시할 것인가

가. 어떤 토폴로지를 네트워크 코딩의 대상으로 고려할 것인가

COPE^[11]에서는 높은 코딩 이득을 갖기 위해서는 많은 패킷을 감청해야한다. 이는 같은 원리로 네트워크가 높은 코딩 이득을 갖기 위해서는 많은 노드들이 감청을 수행하도록 하는 것은 패킷을 전달할 때 더 많은 에너지를 소모하게 되며, 이는 본 논문의 목표와는 상치된다. 따라서 본 논문에서는 감청 없이 코딩 기회를 가질 수 있는 Chain 토폴로지를 네트워크 코딩 수행 토폴로지로서 고려한다.



그림 6. 고려하는 네트워크 코딩 토폴로지: Chain
Fig. 6. A network coding aware topology: Chain

그림 6은 Chain 토폴로지의 예제를 보여주고 있다. 네트워크 양 쪽 끝에 위치한 A와 B가 패킷을 상호 교환하는 상황을 가정한다. 네트워크의 중간 노드들인 N1, N2, ...은 A가 전송한 패킷과 B가 전송한 패킷을 동시에 갖게 되는 경우, 네트워크 코딩 포인트로 역할을 수행할 수 있다. 코딩된 패킷을 수신한 이웃 노드들은 자신이 전달하였던 패킷과 XOR을 하여 새로운 패킷을 획득할 수 있다. 이때 디코딩을 위해서 자신이 전달하거나 수신한 패킷을 일정

시간동안 저장하고 있어야 하기 때문에 각 노드는 이를 위한 임시 버퍼가 필요하다.

나. 네트워크 코딩을 적용하기 위한 패킷들의 필수 조건

Chain 토폴로지의 중간 노드에서 네트워크 코딩을 수행하기 위해서는 새로운 플로우의 다음 홉이 기존 플로우의 이전 홉으로 통과하고, 기존 플로우의 다음 홉이 새로운 플로우의 이전 홉을 통과해야 한다. 여기서 플로우란 어떤 소스로부터 목적지까지 데이터 흐름을 나타낸다.

그림 7에서는 네트워크 코딩이 가능한 경우와 불가능한 경우를 보여주고 있다. (a)의 네트워크 코딩이 가능한 경우에는 N2에서 경로 1의 이전 홉으로 경로 2의 패킷이 흐르고 있으며 경로 2의 이전 홉으로 경로 1의 패킷이 흐르고 있다. 그러나 (b)의 네트워크 코딩이 불가능한 경우에는 N2에서 경로 2의 이전 홉으로 경로 1의 패킷이 흐르고 있지만 경로 1의 이전 홉으로 경로 2의 패킷이 흐르고 있지 않아서 네트워크 코딩을 시도하면 N4는 디코딩을 할 수 없다.

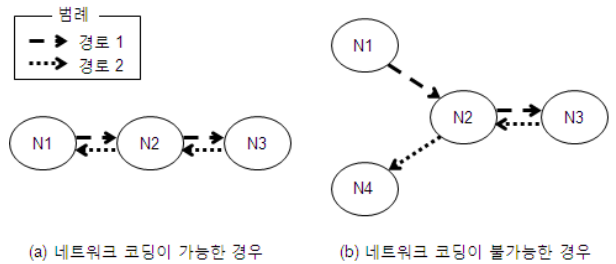


그림 7. 네트워크 코딩 가능 경우 및 불가능 경우
Fig. 7. A network coding possible and impossible case

다. 해당 경로가 네트워크 코딩이 가능한 경로임을 통지하기 위한 방안

PAMP에서는 목적지가 경로를 설정하기 때문에 네트워크 코딩을 통한 에너지 활용을 위해서는 목적지가 어떤 경로가 네트워크 코딩 가능 경로인지 알고 있어야한다. 이를 위해서는 DCAR^[12]에서 제안하였던 방법과 다르게 RREP가 아닌 RREQ가 전달되는 것만으로 해당 경로가 네트워크 코딩이 가능한 경로인지 파악할 수 있어야한다. 본 논문에서는 RREQ에 NC_{opp} 그리고 NC_{conf} 플래그들을 추가하였다. 이 플래그들은 인접 노드들이 해당 경로가 네트워크 코딩 기회를 갖는 지 아닌 지 판단하기 위해서

사용되어진다. 두 개의 플래그 모두가 1로 설정된다면, 목적지에서는 해당 경로가 네트워크 코딩 기회를 갖는다고 판단한다.

이 플래그들이 어떻게 처리되어지는 지 알아보기 위해 중간 노드에서 새로운 RREQ를 수신한 상황을 고려해보자. R_{new} 는 노드에 도착한 새로운 RREQ를 나타내고, R_{out} 는 노드에서 처리되고 난 뒤 다음 노드로 전달하려는 RREQ를 나타낸다. 그리고 N_{prev} 는 RREQ를 중간 노드에 전달한 이전 홉 노드를 나타낸다. 중간 노드에서는 다음 세 가지 동작 중 하나를 수행한다.

- Case 1: R_{new} 의 NC_{opp} 가 0일 때, N_{prev} 로 전달되는 플로우가 있다면, R_{out} 의 NC_{opp} = 1로 설정
- Case 2: R_{new} 의 NC_{opp} 가 1일 때, N_{prev} 로 전달되는 플로우가 있다면, R_{out} 의 NC_{conf} = 1로 설정
- Case 3: R_{new} 의 NC_{opp} 가 1일 때, N_{prev} 로 전달되는 플로우가 없다면, R_{out} 의 NC_{opp} = 0으로 설정

그림 7의 상황을 다시 고려해보자. (a)에서 기존의 경로 1이 설정되어 있을 때, 경로 2가 서비스 경로 탐색중이라고 가정하자. N3으로부터 N2로 전달된 RREQ는 이전 홉 N3으로 전달되는 경로 1이 있으므로 NC_{opp} 를 1로 설정한다. 그리고 N2로부터 N1로 전달된 RREQ는 마찬가지로 이전 홉 N2로 전달되는 경로 1이 있으므로 NC_{conf} 를 1로 설정한다. (b)에서는 N3으로부터 N2로 전달된 RREQ는 이전 홉 N3으로 전달되는 경로 1이 있으므로 NC_{opp} 를 1로 설정한다. 그러나 N2로부터 N4로 전달된 RREQ는 이전 홉 N2로 전달되는 플로우가 없으므로 NC_{opp} 를 0으로 초기화한다.

노드에서 RREQ를 처리할 때 기존 플로우에 할당할 에너지도 고려해야 한다. 만일 기존 플로우에 할당할 에너지가 너무 많아서 새로운 플로우에 할당할 에너지가 매우 적다면, 이는 네트워크 코딩 기회를 통해 갖는 장점이 거의 없기 때문이다.

라. 코딩 포인트의 이웃 노드들에게 디코딩을 위한 사전 동작을 지시하기 위한 방법

네트워크 코딩 가능 경로를 통해서 데이터를 전송하기 전에 반드시 네트워크 코딩 조건을 만족하는 코딩 포인트의 이웃들에게 디코딩을 위한 사전 준비

를 할 수 있도록 공지해야 한다. 사전 준비를 위한 공지를 받은 이웃 노드들은 앞에서 살펴본듯이, 디코딩을 위해서 임시 버퍼에 자신이 전달하거나 수신한 패킷을 일정 시간동안 저장한다. 본 논문에서는 RREP에 NC 플래그와 MAC 계층에 NC_NOTI 컨트롤 메시지를 추가하였다. RREP를 전달하는 중 중간 노드에서 네트워크 코딩 조건을 만족하는 경우, RREP의 NC 플래그를 1로 설정하여 해당 RREP를 수신한 노드가 디코딩을 위한 준비를 수행하도록 지시한다. 이후 RREP의 NC 플래그는 다시 초기화되는데 이는 하나의 경로 상에 여러 개의 코딩 포인트가 존재할 경우를 위함이다. RREP는 유니캐스트 메시지가기 때문에 코딩 포인트로 RREP를 전달한 노드 역시 디코딩을 위한 사전 준비를 수행하도록 지시해야 한다. NC_NOTI는 이때 사용되는 메시지로써 코딩 포인트로 RREP를 전달한 노드에게 전송되며, 이 메시지는 네트워크 코딩이 가능한 플로우들의 리스트와 코딩 포인트의 정보를 포함하고 있다.

3.2. 네트워크 코딩 기반 데이터 전송

효율적인 네트워크 코딩 기반의 데이터 전송을 위해서는 다음 두 가지 사항이 고려되어야 한다.

1) 노드가 코딩된 패킷을 디코딩하기 위해서 필요한 정보는 무엇인가

2) 코딩 포인트에서는 코딩을 수행하기 위해서 어느 정도의 시간만큼 대기하는 것이 효율적인가

가. 코딩된 패킷을 디코딩하기 위해서 필요한 정보

네트워크에 흐르는 각각의 데이터 패킷은 소스 주소, 목적지 주소, 시퀀스 번호, 그리고 다음 홉 주소 등의 정보를 포함하고 있다. 코딩 포인트의 이웃 노드가 코딩된 패킷을 수신한 경우, 디코딩하기 위해서는 어떤 패킷들이 코딩되어 있는지에 대한 정보가 필요하다. 따라서 코딩된 패킷에는 위와 같은 정보를 보존하고 있어야 한다. 본 논문에서는 데이터 패킷에 네트워크 코딩 헤더를 추가하였는데, isNC 플래그와 코딩되는 패킷들의 <소스 주소, 목적지 주소, 시퀀스 번호, 다음 홉 주소> 등의 정보를 포함하고 있다. 네트워크 코딩 헤더는 IP 헤더와 MAC 헤더 사이에 추가하였다. 노드에서 패킷을 수신하였을 때 코딩 헤더의 isNC 플래그가 1로 설정되어 있는 경우, 노드는 자신의 임시 버퍼에 저장되어 있는 패킷과 코딩된 패킷의 정보를 비교하여 새로운 패킷을 적출해낸다.

나. 코딩 포인트에서 코딩을 수행하기 위한 패킷의 큐 대기 시간

만일 코딩 포인트에서 양 쪽의 이웃 노드들로부터 동시에 코딩될 패킷을 수신하지 못하면 네트워크 코딩은 발생하지 못한다. 무선 채널에서 서로 다른 플로우를 통해서 흐르는 패킷들은 서로 다른 경로를 통해 전달되기 때문에 서로 다른 지연 시간과 Jitter를 갖는다. 코딩 포인트에서 플로우 f 와 g 의 패킷이 네트워크 코딩 가능할 때, 플로우 f 의 패킷이 도착하였으나 아직 플로우 g 의 패킷이 도착하지 않은 상황을 고려해보자. 만일 코딩 포인트에서 네트워크 코딩을 발생시키기 위해 플로우 g 의 패킷이 도착할 때까지 플로우 f 의 패킷을 전달하지 않고 대기시킨다면, 플로우 f 는 불필요하게 긴 지연 시간을 갖게 될 것이다. 코딩 포인트의 에너지 절감을 위해서 네트워크 코딩을 발생시키는 것도 중요하지만 긴 대기 시간은 네트워크의 수율을 낮추기 때문에 적절한 시간만큼 대기 시간을 갖는 것이 효율적이다. 본 논문에서는 이 대기 시간을 네트워크 코딩 대기 시간 (τ_{coding})이라고 칭한다. τ_{coding} 은 전체적인 전달 지연 시간을 늘이는 부정적인 효과와 네트워크 코딩 기회를 증가시키는 긍정적인 효과를 가지므로, 적절한 값을 설정하는 것 또한 중요한 기술적인 고려 사항이 된다.

IV. 제안하는 기법

4.1. 개요

그림 8은 본 논문에서 제안하는 네트워크 코딩을 적용한 PAMP^[7]의 예제를 보여주고 있다. 그림의 코딩 포인트 C에서 네트워크 코딩을 수행하면 S1과 S2의 데이터 전달에 필요한 에너지를 절감할 수 있다. 이는 네트워크의 운용시간 및 신규 플로우에 대한 지원 시간이 향상될 수 있다. 또한 제안하는 기법은 세션 간 발생하는 코딩 이득뿐만 아니라 네트워크 전체에서 발생하는 코딩 이득을 통해서 코딩 포인트의 효율적 에너지 사용이 가능하다.

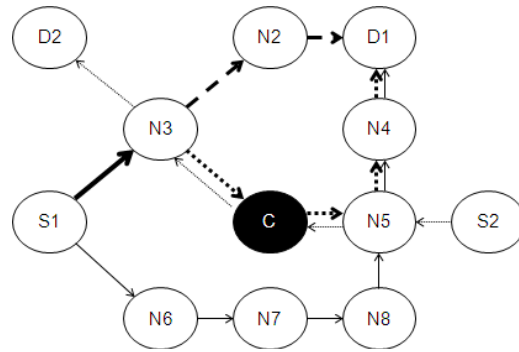


그림 8. 네트워크 코딩을 적용한 PAMP
Fig. 8. An example of incorporated PAMP with network coding

4.2. 서비스 경로 탐색

제안하는 기법은 PAMP^[1]의 에너지 인식 다중 경로 라우팅 기법에 네트워크 코딩 인식 라우팅 기법을 추가하여 확장하였다. 이는 서비스 경로 상에 에너지 예약과 동시에 RREQ만으로 어떤 경로를 통해서 네트워크 코딩이 가능한지 탐색하기 위해서다. 네트워크 코딩 인식 라우팅 기법은 III장에서 제시한 조건을 기반으로 수행한다.

```

checkNCOpportunity(RREQ p)
IF getNCOpp(p) == 1 AND getNCCConf(p) == 0 THEN
    /* the first case explained in III.1.다 */
    IF there is a flow whose next hop is prevHop(p)
    THEN
        setNCCConf(p)
        /* network coding available path */
    ELSE /* the second case explained in III.1.다 */
        resetNCOpp(p)
    ENDIF
ENDIF

IF getNCOpp(p) == 1 AND getNCCConf == 0 THEN
    /* the third case explained in III.1.다 */
    IF there is a flow whose next hop is prevHop(p)
    THEN
        setNCOpp(p)
    ENDIF
ENDIF
RETURN with p
    
```

그림 9. 중간 노드에서 RREQ를 수신했을 때 처리 과정
Fig. 9. The procedure when the intermediate node receives a RREQ

그림 9는 중간 노드에서 RREQ를 수신했을 때, 네트워크 코딩 인식 라우팅을 위한 처리과정을 보여주고 있다. 이는 앞서 설명하였던 내용과 동일하다.

4.3. 서비스 경로 설정

제안하는 기법은 네트워크 코딩 기회를 최대한 활용하는 것이 목적이기 때문에 지원 에너지가 많은

경로를 우선으로 선택하여 다중 경로를 설정하는 PAMP를 확장하였다. 제안하는 기법에서는 RREQ를 수신했을 때, 해당 경로가 네트워크 코딩 기회를 갖는지 아닌지를 구별하여 저장한다. 이후 경로 선택 시 네트워크 코딩 가능한 경로들의 집합에서 먼저 경로를 선택하여 설정한다. 만일 네트워크 코딩 가능 경로들을 모두 설정하여도 소스의 에너지 요구량을 만족하지 못한다면, 네트워크 코딩 기회를 갖지 못하는 경로의 집합에서 경로를 추가적으로 선택하여 설정한다.

```

/* The RREQ_LIST is divided into two parts: the first
part has the RREQs that represent network coding
possible paths and second part has the other RREQs */
/* put the newly received RREQ in the proper position
in RREQ_LIST */
pushRREQ(receivedRREQ p)
FOR noNCptr = a RREQ in RREQ_LIST DO
    IF getNCConf(noNCptr) != TRUE THEN
        BREAK
    /* select the RREQ that lies at the boundary of the
first and second parts in RREQ_LIST */
    ENDF
ENDFOR

IF getNCConf(p)==TRUE THEN
    /* if NCConf of RREQ p is 1 */
    FOR temp = a RREQ in RREQ_LIST between the
head and noNCptr DO
        IF E_PATH(temp) < E_PATH(p) THEN
            BREAK
        ENDF
    ENDFOR
ELSE
    FOR temp = a RREQ in RREQ_LIST between
noNCptr to the tail of the list DO
        IF E_PATH(temp) < E_PATH(p) THEN
            BREAK
        ENDF
    ENDFOR
ENDIF

INSERT(p, temp)
/* put the received RREQ in front of temp RREQ
in the RREQ_LIST */
STOP
    
```

그림 10. 목적지에서 RREQ를 수신했을 때 처리과정
Fig. 10. The procedure when the destination receives a RREQ

그림 10은 목적지에서 RREQ를 수신했을 때, 처리 과정을 나타낸다. RREQ의 NC_{conf}가 1로 설정되어 있는지 아닌지에 따라서 리스트 내 RREQ의 저장 위치가 달라진다. 포인터 noNCptr은 네트워크 코딩이 가능한 경로의 집합과 불가능한 경로의 집합을 구별하기 위해 사용된다. 리스트 헤더부터 포인터 noNCptr까지의 경로들은 네트워크 코딩이 가능한 경로들이며, noNCptr부터 리스트 테일까지의 경로들은 네트워크 코딩이 불가능한 경로들이다. 목적지에서는 경로 선택 시 리스트 헤더부터 시작해서 리스

트 테일까지 경로를 하나씩 선택하는데, 이 과정은 선택된 경로들의 지원 가능한 에너지 합이 소스의 에너지 요구량을 만족할 때까지 수행된다.

4.4. 네트워크 코딩 기반 데이터 전송

경로 설정 이후 소스는 라우팅 테이블을 이용하여 데이터를 전송한다. 본 논문에서 데이터는 MDC^[13]나 Digital Fountain^[15] 기법 등이 적용되어 전달 순서에 무관하다고 가정한다. 이런 기법들은 목적지에 일정 개수의 패킷이 전달되면 원래 데이터를 복원하는 것이 가능하다. 각 노드들은 매 패킷을 전달할 때마다 각 경로에 예약되어 있는 에너지양을 패킷이 송/수신 하는데 필요한 만큼 감소시키고 예약된 에너지양이 0이 되면 해당 경로로의 전달을 중단한다.

```

networkCoding()
p = dequeue()
/* fetch a packet p from the network interface queue */
FOR pt = a packet in the queue DO
    IF nextHop(p) == prevHop(pt) AND
nextHop(pt) == prevHop(p) THEN
        pt = pick()
        /* pick out pt from the queue that is
network-codable with p */
        p = doNetworkCoding(p, pt)
        /* perform the random linear coding */
        BREAK
    ENDF
ENDFOR
RETURN with p
    
```

그림 11. 코딩 포인트에서 데이터 전달 과정
Fig. 11. The data delivery procedure at the coding point

그림 11은 코딩 포인트에서 데이터 전달 과정을 보여주고 있다. 코딩 포인트에서 데이터 전달 시 이웃 노드가 디코딩할 수 있는 한에서 자신이 전달해야 할 패킷을 XOR하여 전달한다.

그림 12는 코딩된 패킷을 받은 노드에서 디코딩 수행 과정을 보여주고 있다. 자신이 가진 Native 패킷과 수신한 코딩된 패킷을 비교하여 새로운 패킷을 적출한다. 만일 디코딩에 실패한다면, 해당 패킷의 정보를 RequestMessage에 담아서 브로드캐스트를 통해 요청한다. 요청 메시지를 받은 이웃 중 해당 패킷을 가진 노드가 있다면, 이를 브로드캐스트를 통해 전송하여 이웃 노드의 디코딩 수행을 돕는다.


```

recvPacket()
  p = recv()
  IF destination(p) is not itself THEN
    /* if p is overheard packet */
    put p at the tail of overhearing queue
  ELSE
    IF isNetworkCodedPacket(p) == true THEN
      p = decodeNetworkCodedPacket(p)
      /* call the decoding function to get an
      innovative packet */
    ENDIF
  ENDIF
RETURN with p

decodeNetworkCodedPacket(codedPacket p)
  uid1 = getFirstCodedPacketUID(p)
  uid2 = getSecondCodedPacketUID(p)
  IF (overhearing queue is empty) THEN DO
    wait  $\tau_{coding}$ 
    /* extract the packet ids coded in the packet */
  FOR pt = a packet in the overhearing queue DO
    /* if the id of the overheard packet is either
    uid1 or uid2, decoding is possible */
    IF getUID(pt) == uid1 OR
      getUID(pt) == uid2 THEN
      p = decoding(p, pt) /* perform decoding */
      BREAK
    ENDIF
  ENDFOR
RETURN with p

```

그림 12. 중간 노드에서 디코딩 수행 과정
 Fig. 12. The decoding procedure at the intermediate node

V. 성능 분석

5.1. 네트워크 코딩 시뮬레이터 구현

본 논문에서 제안하는 기법을 ns-2 (Network Simulator version 2)^[15]를 사용하여 시뮬레이션하였다. 이를 위해 기회적 네트워크 코딩 모듈을 ns-2에 구현하였으며, 무선 채널의 손실, 혼잡, 그리고 경쟁 트래픽 등의 요인에 따른 네트워크 코딩 이득 변화를 분석할 수 있는 기능을 ns-2에 구현하였다.

네트워크 코딩 모듈은 감청, 기회적 코딩, 그리고 패킷 재요청 등의 세 가지 주요 기능을 기반으로 동작한다. 감청은 이웃 노드가 전송하거나 수신하는 패킷을 감청하여 제한된 시간 동안 저장하는 기능이다. 기회적 코딩은 한 번의 전송으로 이웃들에게 가능한 많은 정보를 제공할 수 있도록 패킷을 선별하여 코딩 하는 기능이다. 마지막으로 패킷 재요청은 노드가 디코딩에 실패할 경우, 이를 복구하기 위해 필요한 패킷을 요청하는 기능이다.

1) 감청 : 기회적 코딩으로 디코딩 실패 없이 많은 데이터를 네트워크에서 공유하기 위해서는 자신이 처리한 패킷뿐만 아니라 이웃이 처리한 패킷 역시

감청하여 수집해야 한다. 그러나 이런 데이터는 오직 디코딩을 위해 수집할 뿐이므로 오랜 시간 동안 저장하고 있으면 공간의 낭비가 발생한다. 따라서 감청된 데이터는 적절한 주기 동안만 저장한다.

2) 기회적 코딩 및 패킷 재요청 : 기회적 코딩의 목적은 이웃에게 한 번의 전송으로 많은 데이터를 전달하고자 하는데 있다. 이를 위해서 다음과 같은 가정을 갖는다. 첫 번째, 패킷의 크기는 모두 동일하다. 두 번째, 컨트롤 패킷과 다음 홉이 같은 패킷은 코딩 하지 않는다. 코딩 알고리즘은 노드에서 다음 홉이 다른 모든 패킷을 수집하여 진행한다. 패킷 재요청은 노드에서 감청된 데이터가 부족하여 수신된 패킷을 디코딩 할 수 없는 경우에 발생한다. 이 경우에 디코딩에 필요한 데이터를 이웃에게 브로드캐스트하여 요청하고 이웃은 요청 데이터 중 자신이 갖고 있는 패킷을 전송함으로써 해결할 수 있다.

네트워크 코딩 모듈은 Mac 802.11 모듈에 감청, 기회적 코딩, 그리고 패킷 재요청 등의 동작을 수행하는 함수를 추가하여 구현하였다. 그리고 Interface Queue 모듈에 코딩이 가능한 데이터를 선별하는 함수를 추가하였다. 그림 13은 네트워크 코딩 모듈의 전체적인 구성을 보여준다.

5.2. 성능 평가

ns-2를 이용하여 본 논문에서 제안하는 기법 (PAMP+NC), 기존의 PAMP, 무선 센서 네트워크용 에너지 인식 다중 경로 기법인 REAR를 시뮬레이션하였다.

네트워크 토폴로지는 두 가지로 나누어서 진행하였다. 첫 번째는 1000m x 1000m의 공간에 16개의 노드를 격자형으로 분포시켜 네트워크 형상을 구축한 그리드 토폴로지에서 성능 평가를 수행하였으며, 두 번째는 같은 공간에 128개의 노드를 랜덤하게 분포시켜 네트워크 형상을 구축한 랜덤 토폴로지에서 수행하였다. 시뮬레이션에 사용된 트래픽 패턴은 CBR (Constant Bit Rate)을 사용하였으며 소스는 512byte 크기의 패킷을 0.1초 간격으로 10분 동안 주기적으로 전송한다. 이때 패킷 하나를 전송하는데 소모하는 에너지는 0.12J, 수신하는데 소모하는 에너지는 0.04J로 설정하였으며, 컨트롤 메시지를 송/수신하는데 소모하는 에너지는 0.04J로 설정하였다. 각 노드는 250m의 송/수신 파워를 갖고 있으며, 노드의 초기 에너지 E_{max} 는 80J로 설정하였다. 따라서 단말이 에너지를 가득 가지고 있는 경우, 약 500개의 패킷을 전송할 수 있다. 코딩 포인트의 τ_{coding} 은 1 ms

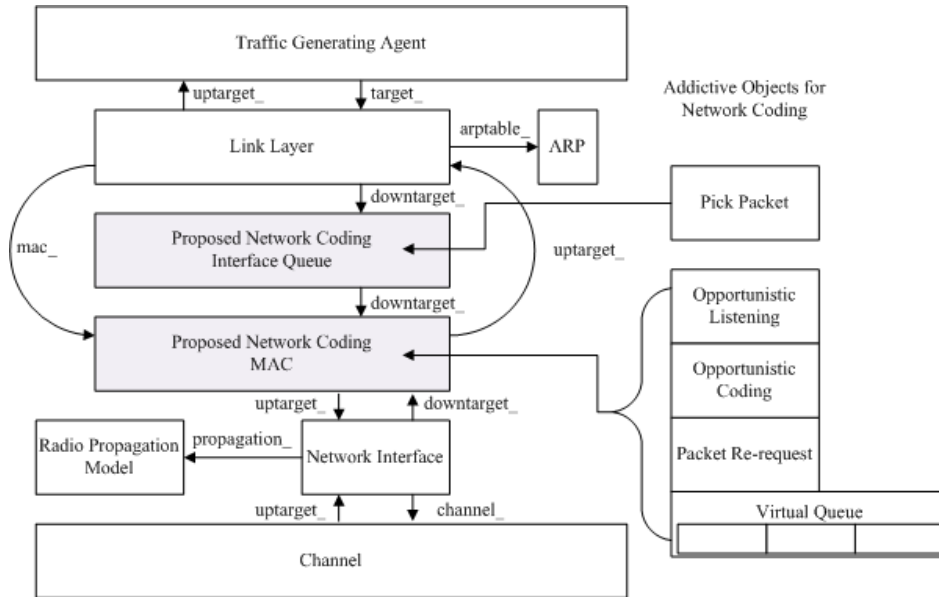


그림 13. 네트워크 코딩 모듈 구조
Fig. 13. The structures of network coding modules

로 설정하였다. 이는 불필요한 지연을 최소화하기 위함이다. 시뮬레이션에 사용된 트래픽은 총 400개의 패킷을 전송하는 경우를 대상으로 성능을 측정하였다.

시뮬레이션은 다음의 세 가지 평가 항목을 측정하였다. 첫째, 데이터 전송 성공률이다. 소스가 전송한 전체 데이터 패킷 수와 수신자가 수신한 데이터 패킷 수의 비율로서 백분율로 표현한다. 둘째, 생성된 경로의 개수이다. 플로우 별로 설정된 경로의 개수의 총 합을 나타낸다. 시뮬레이션이 시작되는 시점의 네트워크 내 노드의 에너지를 다르게 설정한다. 즉, 특정 초기 에너지에서 설정된 경로의 수가 0이 되는 기법이 1이상의 경로를 설정할 수 있는 기법에 비해 네트워크 지속 시간이 작다고 할 수 있다. 셋째, 코딩 포인트의 소비 에너지이다. 이는 PAMP와 PAMP+NC를 비교하기 위한 평가 항목으로, 네트워크 코딩을 활용하는 것이 에너지 잔존량을 높이는 데 기여하는지 확인하기 위함이다.

가. 그리드 토폴로지 - 2개의 플로우, 플로우당 패킷 400개 전송

그리드 토폴로지에서도 두 개의 플로우가 있을 때, 각각의 플로우는 400개의 데이터 패킷을 보내는 상황을 성능 평가하였다.

그림 14에서는 네트워크 내 노드들의 초기 에너지를 달리 설정하면서 측정한 생성된 경로 수를 보

여주고 있다. 에너지가 90%까지는 세 가지 기법 모두 다중 경로를 구축하여 데이터 전달을 지원하였으나, 그 이하의 에너지에서는 REAR가 경로를 설정하지 못한다. REAR는 잔여 에너지에 따라 경로에 참여할 확률을 결정하기 때문에, 초기 에너지가 작아지면 경로에 참여할 확률이 낮아지면서 경로 설정에 실패하게 된다. 이에 반해 PAMP와 PAMP+NC는 잔여 에너지가 작아도 경로에 참여할 수 있으므로 초기 에너지가 작아도 경로 설정이 가능하다.

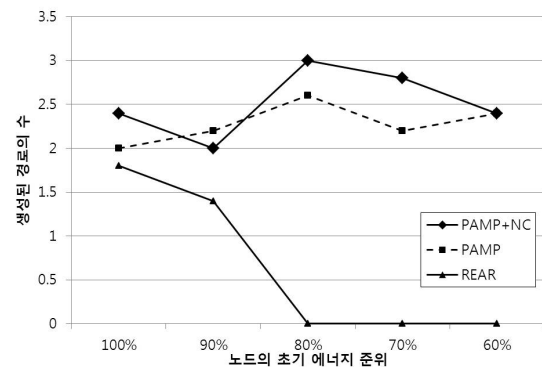


그림 14. 그리드 토폴로지: 생성된 경로의 수
Fig. 14. Grid topology: the number of paths built for data delivery

그림 15에서는 네트워크 내 노드들의 초기 에너지를 달리 설정하면서 측정한 패킷 전송 성공률을 보여주고 있다. PAMP와 PAMP+NC는 라우팅 기법으로서 신뢰성 있는 데이터 전송을 제공하지 않고

MAC 계층에서의 충돌 등에 의한 데이터 손실을 보완하는 기법을 갖고 있지 않다. REAR는 주 경로가 실패했는지 파악하기 위한 ACK은 정의가 되어 있으나 신뢰성 있는 데이터 전달 알고리즘이 마련되어 있지 않다. 그래서, 세 가지 기법 모두 70% 이하의 패킷 전송 성공률을 보이고 있다.

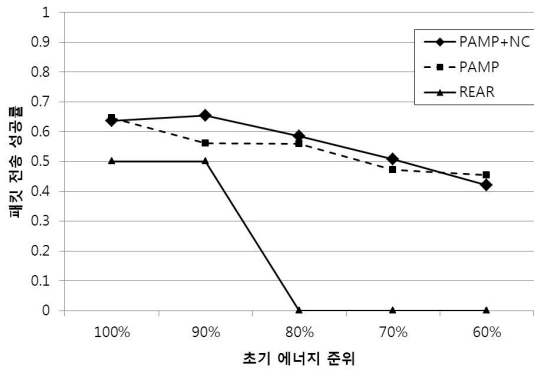


그림 15. 그리드 토폴로지: 패킷 전송 성공률
Fig.15. Grid topology: packet delivery ratio

REAR가 PAMP나 PAMP+NC에 비해 낮은 패킷 전송 성공률을 보이고 있다. REAR는 주 경로가 실패했을 시에 백업 경로로 전환하는 판단을 하기 전까지 버려지는 패킷들의 수가 많으나, PAMP와 PAMP+NC 모두 각 경로로 전송해야 할 데이터의 양이 미리 정해져 있어서 소스나 중간 노드가 자율적으로 결정하면 되므로 경로 전환하는데 지연 시간이 없어서 이로 인한 패킷 손실이 발생하지 않는다.

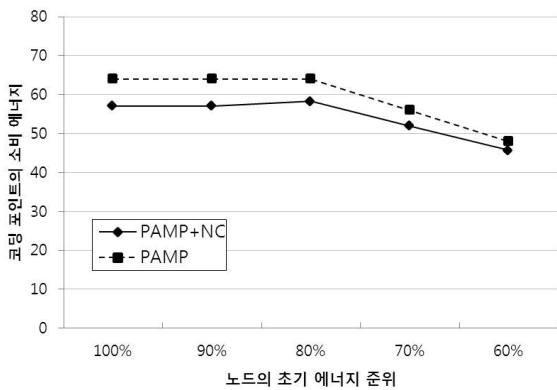


그림 16. 그리드 토폴로지: 코딩 포인트의 소비 에너지
Fig. 16 Grid topology: Energy consumption at the network coding points

그림 16에서는 PAMP+NC에서 코딩 포인트로 설정된 노드의 에너지 소비량을 보여주고 있다. 해당 노드는 PAMP에서도 경로 상에 존재하여 데이터 전

달을 위해 에너지를 소모하므로, 그림 15에서 두 기법을 비교하여 보여주고 있다. PAMP+NC가 PAMP를 적용할 시에 소비되는 에너지 대비 91.5%에 해당하는 에너지를 평균적으로 소비한다. PAMP+NC의 경우, 네트워크 코딩 기회를 활용하여 데이터 전달 횟수를 감소시킴으로써 에너지 소모가 적게 나타난 것이다.

나. 랜덤 토폴로지 - 4개의 플로우, 플로우당 패킷 200개 전송

마지막으로 랜덤 토폴로지에서 네 개의 플로우가 있을 때, 각각의 플로우는 200개의 데이터 패킷을 보내는 상황을 성능 평가하였다. 네트워크의 노드 수는 128개로 늘려서 성능 평가를 수행하였다.

그림 17에서는 랜덤 토폴로지에서 네트워크 내 노드의 초기 에너지를 다르게 설정하였을 때 생성된 경로 수를 보여주고 있다. PAMP+NC는 가용 에너지가 높은 경로보다 네트워크 코딩이 가능한 경로를 우선적으로 설정하기 때문에, PAMP에 비해 차이는 크지 않지만 보다 많은 경로를 설정하게 되는 경향이 있다. REAR는 그리드 토폴로지의 경우와 마찬가지로 이유로 초기 에너지가 80% 이하인 경우에는 경로 설정이 이루어지지 않는다.

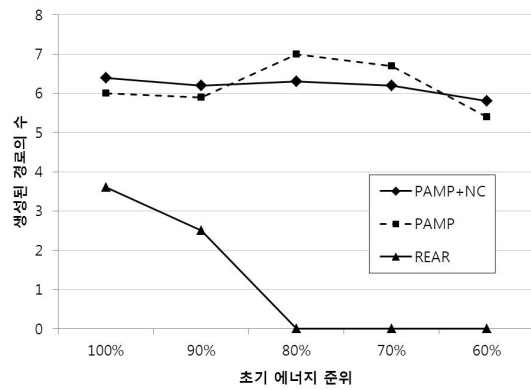


그림 17. 랜덤 토폴로지: 생성된 경로의 수
Fig. 17. Random topology: the number of paths built for data delivery

그림 18에서는 패킷 전송 성공률을 보여주고 있다. 랜덤 토폴로지이므로 그리드 토폴로지에 비해 생성되는 경로의 수가 많으므로, 보다 다양한 노드들이 데이터 전달에 참여하게 되어 MAC 계층에서의 충돌의 수가 감소하여 패킷 전송률이 그리드 토폴로지에 비해 우수하게 나타난다.

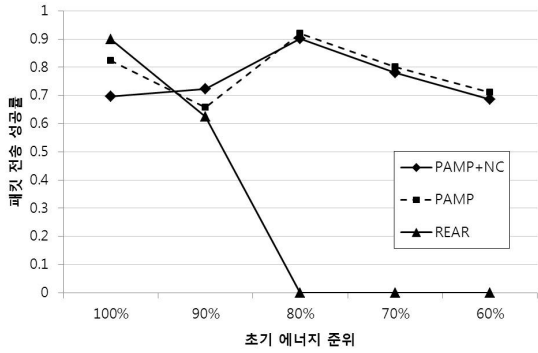


그림 18. 랜덤 토폴로지: 패킷 전송 성공률
Fig.18. Random topology: packet delivery ratio

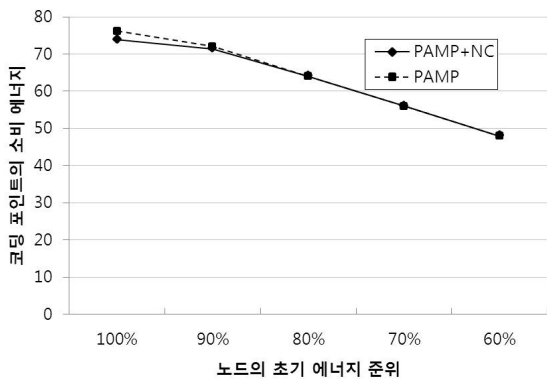


그림 19. 그리드 토폴로지: 코딩 포인트의 소비 에너지
Fig. 19. Grid topology: Energy consumption at the network coding points

그림 19에서는 코딩 포인트의 소비 에너지를 보여주고 있다. 그림에서 보여주는 바와 같이, 제안하는 기법이 기존 PAMP에 비해 에너지 소모량에 별 차이가 없다. 이는, 랜덤 토폴로지의 경우 네트워크 코딩 포인트로 설정된 노드가 그리드 토폴로지에 비해 실제 코딩을 적용할 기회가 갖지 못해서 에너지 소모량 감소가 크게 적게 나타난 것이다.

다. 네트워크 코딩 대기 시간 효과 분석

본 논문에서 제안하는 기법은 코딩 포인트에서 네트워크 코딩 기회를 활용함으로써 에너지 절감 효과를 기대할 수 있으나, 네트워크 코딩 기회를 기다리는 코딩 대기 시간 (τ_{coding})을 발생시켜 타 기법들에 비해 전체 패킷 전달 지연 시간을 증가시키는 효과를 불러올 수 있다.

그림 20은 그림 4(a)에 제시된 것과 같은 네트워크 코딩 기회를 활용하는 경우의 성능을 분석한 결과를 보여주고 있다. 세 개의 노드 중 양 끝에 위치한 노드들이 0.25초 간격으로 100개부터 1000개까지

의 데이터를 생성하여 전송하였고, 중간에 위치한 노드는 네트워크 코딩 기회를 갖기 위해 코딩 지연 시간만큼 기다려서 코딩을 수행하였다. 가로 축은 네트워크 코딩 대기 시간과 패킷 구간 간의 비율을 의미한다. 즉, 네트워크 코딩 대기 시간을 0.025, 0.05, 0.10, 0.25, 0.5로 증가시켜 가면서 네트워크 코딩 기회를 계수하여 전송된 패킷 대비 네트워크 코딩이 적용된 횟수의 평균값을 그래프로 나타내었다.

그림 20에서 보는 바와 같이 네트워크 코딩 대기 시간이 패킷 구간과 동일할 때 가장 많은 네트워크 코딩 기회를 제공했고, 이보다 증가하게 되는 경우 네트워크 코딩 기회 증가분이 크게 감소하고 있다.

중단간 전달 지연 시간을 고려하면, 네트워크 코딩 대기 시간만큼 순증가하게 되므로, 네트워크 코딩 대기시간이 패킷 구간의 100%를 넘어가는 경우에는 네트워크 코딩을 통한 이득보다는 손실이 더 증가하게 된다고 예상할 수 있다.

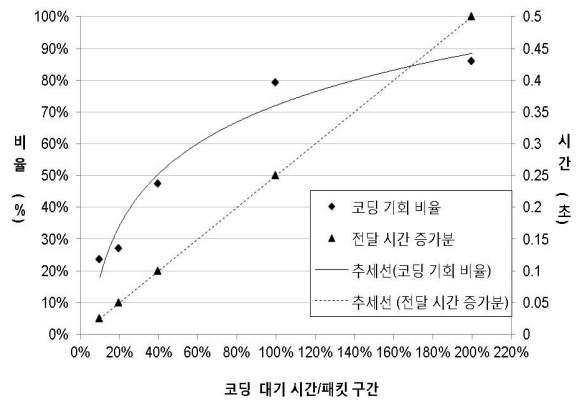


그림 20. 코딩 대기 시간에 따른 코딩 기회 비율의 변화와 패킷 전달 시간 증가분의 변화
Fig. 20. Network coding waiting time vs. Coding opportunity and end-to-end delay increase

VI. 결 론

본 논문에서는 기존에 무선 애드 혹 네트워크 내의 잔여 에너지가 작은 낮은 노드들을 활용하는 다중 경로 설정 기법인 PAMP에 네트워크 코딩을 결합하여 에너지 절감 효과를 높이기 위한 방안을 제시한다. 다중 경로 설정 시에 잔여 에너지의 양뿐만 아니라 소스로부터 목적지까지 전달되는 과정에서 기존의 설정된 다른 세션과의 네트워크 코딩 기회 여부를 고려하여 경로를 설정하도록 하는 것이 본 논문에서 제시하는 기법의 핵심이다.

기존의 무선 센서 네트워크를 대상으로 한 에너지

인식 다중 경로 기법인 REAR와 PAMP, 그리고 제안하는 기법과의 시뮬레이션을 통한 성능 분석에서 제안하는 기법이 개선된 성능을 보임을 확인하였다. 다만, 네트워크 코딩 기회를 활용하여 성능 개선을 얻을 수 있는 것이 네트워크 토폴로지에 영향을 받는다는 것을 관찰하였다. 그리드 토폴로지와 같이 특정 노드가 다수의 세션의 경로에 참여할 확률이 높은 경우에 네트워크 코딩을 활용할 수 있는 기회가 많아서 랜덤 토폴로지의 경우에 비해 성능 개선 효과가 더 높았다. 다만, 네트워크 코딩 기회를 갖기 위한 지연 시간이 에너지 절감 효과를 발생시킬 수 있으나 전체 종단간 지연 시간을 증가시키게 되므로 패킷 구간 대비 100% 이내의 값을 설정해야 한다.

향후 연구로는 네트워크 코딩 대기 시간으로 인한 에너지 절감 효과 대비 전체 종단간 지연 시간 증가에 미치는 효과를 보다 다양한 토폴로지와 다양한 세션 시나리오들을 구성하여 분석할 것이다. 또한, 기존의 설정된 경로를 고정값이 아닌 변화 가능한 요소로 간주하여, 본 기법을 네트워크 코딩의 기회를 더 확장할 수 있는 경로 설정 기법으로 발전시켜 나갈 것이다.

References

- [1] J. Gomez and A. Campbell, "Power-Aware Routing Optimization for Wireless Ad Hoc Networks," *Proc. of High Speed Net. Workshop*, 2001.
- [2] Y. Xue and B. Li, "A Location-aided Power-aware Routing Protocol in Mobile Ad Hoc Networks," *Proc. of IEEE Global Telecommunication Conference*, pp. 2837 - 2841, 2001.
- [3] Wang Kun, Xu Yin-Long, Chen Guo-Liang, and Wu Ya-Feng, "Power-aware on-demand routing protocol for MANET," *Proc. of ICDCSW*, 2004.
- [4] M. Maleki, K. Dantu, and M. Pedram, "Power Aware Source Routing Protocol for Mobile Ad Hoc Networks," *Proc. of International Symposium of Low Power Elect. and Design*, pp.72 - 75, 2002.
- [5] Hossam Hassanein and Jing Luo, "Reliable Energy Aware Routing in Wireless Sensor Networks," *Proc. of IEEE DSSNS*, 2006.
- [6] Liansheng Tan, Ling Xie, King-Tim Ko, Ming Lei, and M. Zukerman, "LAMOR: Lifetime-Aware Multipath Optimized Routing Algorithm for Video Transmission over Ad Hoc Networks," *Proc. of IEEE VTC 2006-Spring*. pp.623-627, 2006, .
- [7] Jin Seok Yang, Kyungran Kang, Young-Jong Cho, and Sung Yoon Chae, "PAMP: Power-Aware Multi-Path Routing Protocol for a Wireless Ad hoc Network," *Proc. of IEEE WCNC*, pp.2247-2252, 2008.
- [8] R. Ahlswede, Ning Cai, S. -Y.R. Li, R. W. Yeung, "Network information flow," *IEEE Trans. on Information Theory*, 46(4), pp.1204-1216, 2000.
- [9] M. Azhar Iqbal, B. Dai, B. Huang, A. Hassan, and S. Yu, "Survey of network coding-aware routing protocols in wireless networks," *Journal of Network and Computer Applications*, 34(6), pp.1956-1970, 2011.
- [10] N. Gaddam, S.A. Gathala, D. Lastine, and A. Somani, "Energy Minimization through Network Coding for Lifetime Constrained Wireless Networks," *Proc. of ICCCN*, 2008.
- [11] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Medard, and Jon Crowcroft, "XORs in the Air: practical wireless network coding," *IEEE/ACM Trans. on Networking*, 16(3), pp.497-510, 2008.
- [12] Jilin Le, Jhon C.S. Lui, and Dah-Ming Chiu, "DCAR: distributed coding-aware routing in wireless networks," *IEEE Transactions on Mobile Computing*, 9(4), pp.596-608. April 2010.
- [13] Vivek K Goyal, "Multiple Description Coding: Compression Meets the Network," *IEEE Signal Processing Magazine*, 18(5), pp.74-93, September 2001.
- [14] D. J. C. MacKay, "Fountain codes," *IEE Proceedings-Communications*, 152(6), pp.1062-1068, December 2005.
- [15] The Network Simulator: NS-2, <http://www.isi.edu/nsnam/ns>, 2009.

최 태 종 (Choi Tae Jong)



2010년 2월 안양대학교 학사
2012년 2월 아주대학교 석사
<관심분야> 네트워크 코딩, 멀티캐스트, 무선 네트워크

조 영 종 (Young-Jong Cho)



1985년 2월 KAIST 석사
1990년 2월 KAIST 박사
1996년 3월~현재 아주대학교 교수
<관심분야> 멀티캐스트, 무선 네트워크, 트래픽 모델링

강 경 란 (Kyungran Kang)



1994년 2월 KAIST 석사
1999년 2월 KAIST 박사
2004년 3월~현재 아주대학교 부교수
<관심분야> 멀티캐스트, 이동 네트워크, 전송 통신

방 준 호 (Bang, June-Ho)



2010년 2월 아주대학교 학사
2012년 8월 아주대학교 석사
<관심분야> 무선 네트워크, 네트워크 신뢰성