

네트워크 코딩의 개요

김재홍 | 박현철
한국정보통신대학교

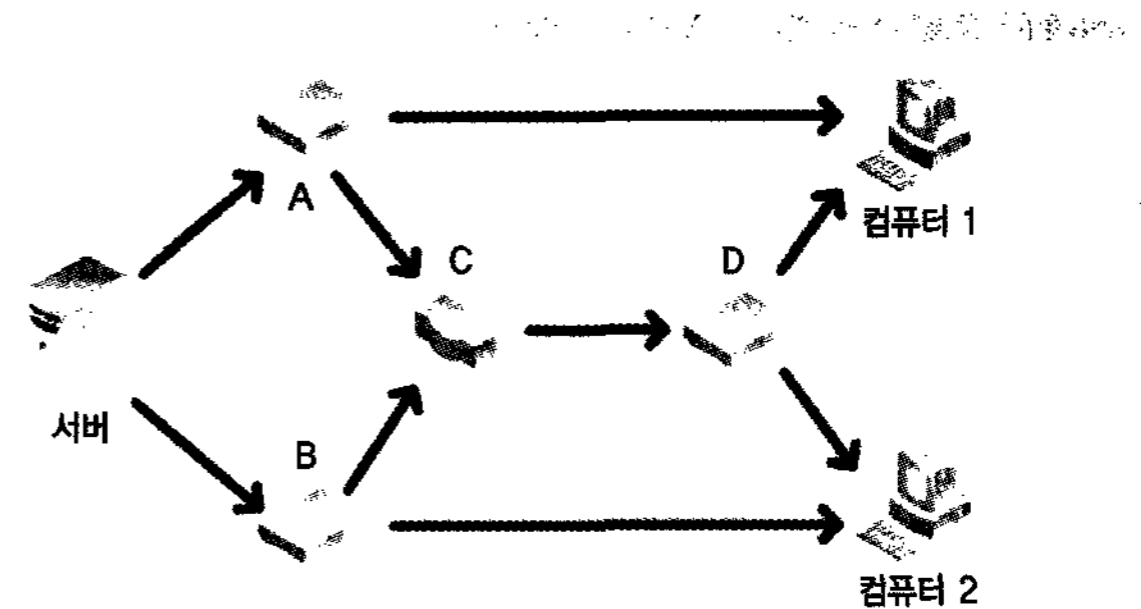
요약

대부분의 통신 네트워크에서는 정보 전달이 라우팅(routing)에 의해 이루어진다. 라우팅은 각 노드의 입력 링크를 통해 들어오는 정보를 단순히 출력 링크로 전달하는 것이다. 네트워크 코딩(network coding)은 네트워크의 노드에서 입력 정보의 전달 및 적절한 결합 등을 포함한다. 본고에서는 네트워크 코딩의 이론적인 배경에 대해 알아 보고, 네트워크 코딩을 실제 네트워크에 적용하기 위해 필요한 요소와 네트워크 코딩이 응용되는 예에 대해 알아본다.

1. 서론

네트워크 상에서 각 노드가 원하는 정보를 전달하는 기법을 라우팅이라고 한다. 이것은 각 노드에 들어오는 정보를 단순히 다른 노드로 전달하는 것이다. 반면에 네트워크 코딩은 각 노드에 들어오는 정보들의 적절한 결합을 통해 다른 노드로 전달하는 것이다. 네트워크 코딩은 라우팅에 비해 전송 효율을 높일 수 있는 장점이 있다. (그림 1)과 같은 butterfly 네트워크의 경우를 살펴 보자 [1]. 주어진 네트워크에서 주어진 시간 동안 각 컴퓨터는 서버로부터 동일한 양의 데이터를 받으려고 한다. 연결된 링크는 오류 없이 데이터를 복원할 수 있다고 가정한다.

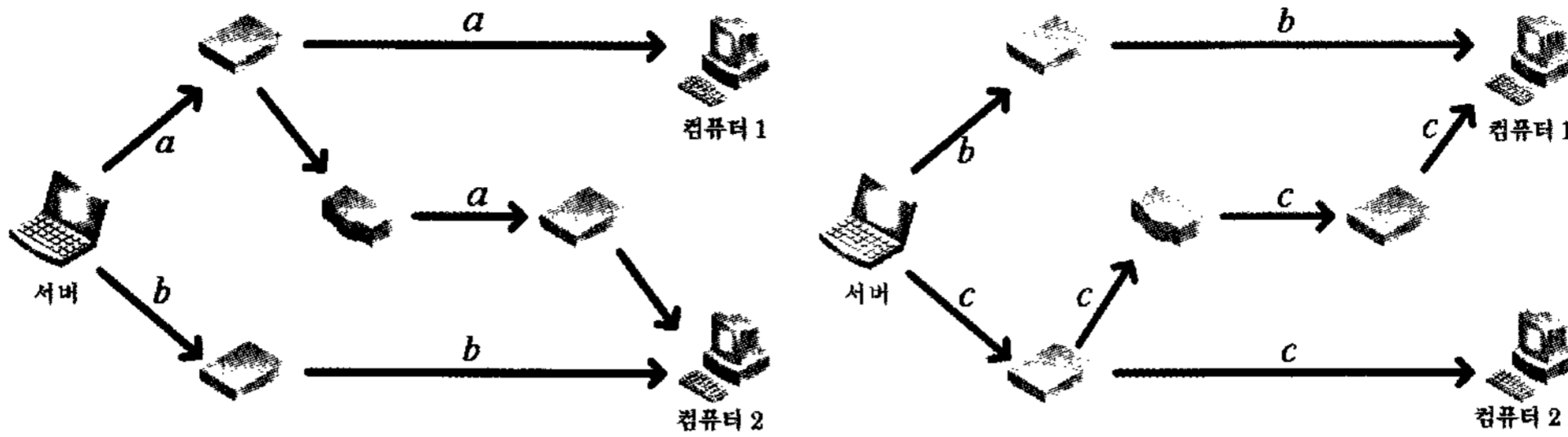
먼저 라우팅 기법으로 얻을 수 있는 최대 전송 효율을 고려한다. (그림 2)와 같이 첫번째 단위 시간 동안 서버는 컴퓨



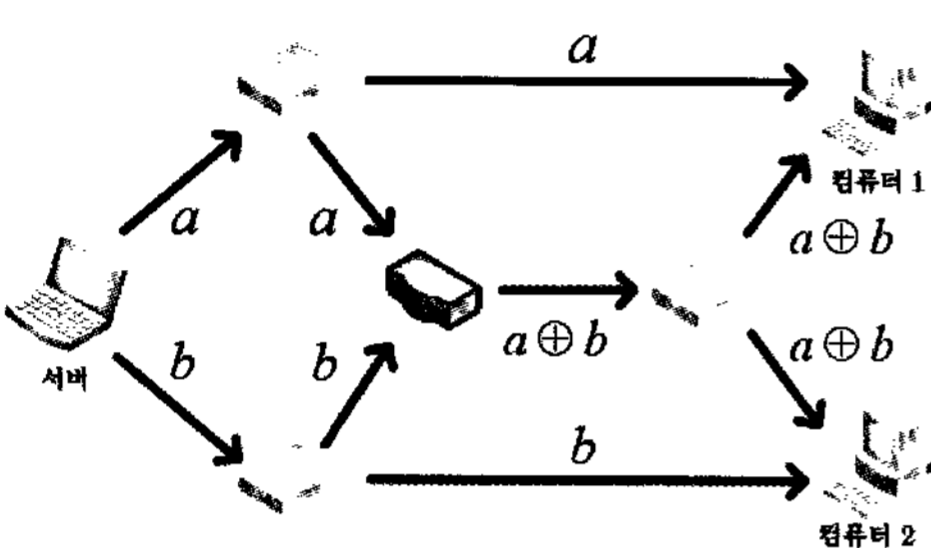
(그림 1) Butterfly 네트워크

터 1로 데이터 a 만 전송하고, 컴퓨터 2로 데이터 a 와 데이터 b 를 동시에 전송한다. 반대로 두번째 단위 시간 동안 서버는 컴퓨터 1로 데이터 a 와 데이터 b 를 동시에 전송하고, 컴퓨터 2로 데이터 c 만 전송하게 된다. 이 때, 두 단위 시간 동안 세 개의 데이터를 전송하였으므로, 주어진 네트워크 상에서의 라우팅 전송 효율은 1.5가 된다. 그러나, 서버에서 각 컴퓨터로 나가는 링크의 수와 각 컴퓨터로 들어오는 링크의 수는 2이므로, 주어진 네트워크에서 얻을 수 있는 최대 전송 효율은 2가 된다.

주어진 네트워크에서 최대 전송 효율을 얻지 못하는 이유는 노드 C와 노드 D 사이에서 한 단위 시간 동안 두 개의 데이터를 전송하지 못하기 때문이다. 이를 극복하기 위한 새로운 형태의 전송 기법이 필요하다. 이러한 전송 기법은 의외로 간단하다. 노드 C에서 노드 A와 노드 B로부터 들어오는 데이터 두 개를 XOR 연산(\oplus)을 수행한 후, 노드 D로 전송하는 것이다. (그림 3)과 같이 서버는 데이터 a 와 데이터 b 를 각 컴퓨터로 단위 시간 동안 전송할 수 있다. 컴퓨터 1은 노드 A를 통해서 데이터 a 를 수신하고, 노드 D로부터 데이



(그림 2) Butterfly 네트워크에서 라우팅 기법



(그림 3) Butterfly 네트워크에서 네트워크 코딩

터 $a \oplus b$ 를 수신한다. 이 때, 컴퓨터 1은 두 링크로부터 수신된 데이터로부터 XOR 연산을 통해 데이터를 얻을 수 있다. 같은 방법으로 컴퓨터 2는 데이터 a 와 데이터 b 를 동시에 얻을 수 있다. 이 때, 단위 시간 동안 두 개의 데이터를 전송하였으므로, 주어진 네트워크 상에서의 전송 효율은 2가 된다. 최대 전송 효율을 만족하는 전송 기법이 된다. 이러한 새로운 전송 기법이 **네트워크 코딩**이다. 주어진 네트워크에서 사용자가 원하는 최대 전송 효율을 얻을 수 있는 방법이다.

본 논문에서는 네트워크 코딩의 이론적인 접근과 실제적인 접근을 나누어 생각해 본다. 네트워크 이론에 관한 배경 지식을 2장에서 설명하고, 이를 바탕으로 네트워크 코딩의 이론적인 접근을 3장에서 다룬다. 그리고, 4장에서는 네트워크 코딩을 실제 네트워크에 적용하기 위한 노력들에 대해 알아본다. 현재 네트워크 코딩이 응용되는 분야에 대해서 5장에서 간략하게 알아본다.

II. 네트워크 이론

일반적인 네트워크의 정의와 네트워크의 분류에 대해서 간략하게 알아본다. 본 논문에서는 그래프 이론과 유한체 이론에 관한 자세한 언급은 피한다.

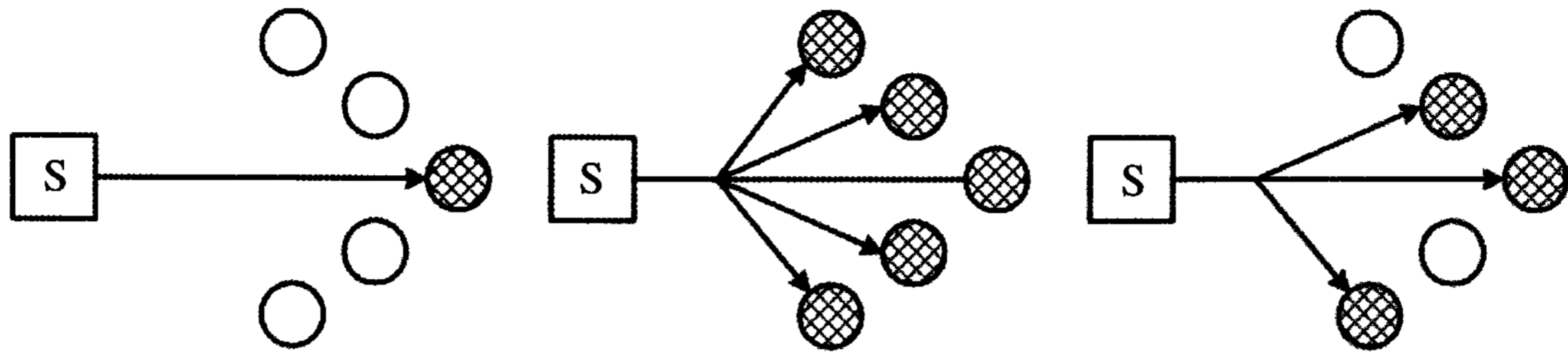
2. 1. 네트워크의 정의

네트워크 N 은 일반적으로 네트워크 노드의 집합 V , 각 노드를 연결하는 링크 e 의 집합 E , 각 링크 e 가 전송할 수 있는 정보의 양 $c(e)$ 로 구성된다. 노드의 집합 중에서 네트워크의 정보를 발생하는 노드를 소스 노드 (source)라고 하고, 정보를 수집하는 노드를 목적 노드 (destination)라고 한다. 본 논문에서의 네트워크는 하나의 소스 노드와 여러 개의 목적 노드를 가지고, 노드 사이 링크의 방향성이 있다고 가정한다. 방향성이 있는 네트워크에서 각 노드에서의 정보가 돌아 오는 사이클 (cycle) 없는 비순환 네트워크를 가정한다. 즉, 소스 노드에서 발생하는 정보는 한 방향으로 목적 노드까지 전달된다. 마지막으로 모든 링크가 전송할 수 있는 정보의 양은 단위 용량 (unit capacity)으로 가정한다.

2. 2. 네트워크의 분류

하나의 소스 노드를 가지는 네트워크는 동일한 정보를 원하는 목적 노드에 따라 (그림 4)와 같이 유니캐스트 (unicast), 브로드캐스트 (broadcast), 멀티캐스트 (multicast)로 나누어 진다.

유니캐스트 네트워크는 하나의 소스 노드의 정보를 하나의 목적 노드로 전달하는 방법이다. 소스 노드에서 목적 노드로 전달할 수 있는 최대 전송 정보의 양은 서로 중복되지 않는 소스 노드를 포함하는 그룹과 목적 노드를 포함하는 그룹 사이의 전달 정보의 양과 관련이 있다는 min-cut max-flow 문제와 같다. 유니캐스트 네트워크의 최대 전송 효율



(그림 4) 유니캐스트, 브로드캐스트, 멀티캐스트 네트워크

(min-cut bound)을 얻을 수 있는 라우팅 기법은 Ford-Fulkerson 알고리즘으로 이미 오래 전에 해결되었다 [2].

브로드캐스트 네트워크는 하나의 소스 노드의 정보를 동일한 양으로 모든 목적 노드까지 전달하는 방법이다. 소스 노드에서 목적 노드로 전달할 수 있는 최대 전송 정보의 양은 확장 트리 (spanning tree) 문제와 같다. 브로드캐스트 네트워크의 최대 전송 효율을 얻을 수 있는 라우팅 기법은 Edmonds-Karp 알고리즘으로 해결되었다 [3].

멀티캐스트 네트워크는 하나의 소스 노드의 정보를 동일한 양으로 일부 목적 노드까지 전달하는 방법이다. 멀티캐스트 네트워크 상에서의 최대 전송 효율을 얻는 라우팅 기법을 찾는 문제는 Steiner 트리 문제와 같고, NP-hard로 알려져 있다 [4]. 멀티캐스트 네트워크에서는 라우팅 기법이 아닌 새로운 형태의 정보 전달 기법이 필요하다. 이러한 멀티캐스트 네트워크에서 새로운 형태의 정보 전달 기법이 네트워크 코딩이다. 네트워크 코딩 기법은 주어진 멀티캐스트 네트워크의 목적 노드가 원하는 정보 양을 항상 얻을 수 있다 [5]. 놀라운 사실은 앞서 살펴본 바와 같이 노드에서 입력 정보들의 간단한 선형 결합을 통해 멀티캐스트 네트워크의 최대 전송 효율을 얻을 수 있다 [6][7]. 다음 장에서는 비순환 멀티캐스트 네트워크에서 최대 전송 효율을 얻을 수 있는 기법이 항상 존재함을 보인다.

III. 네트워크 코딩의 이론적인 접근

멀티캐스트 네트워크의 코딩 기법에 대한 이론적인 접근을 위해 몇 가지 가정이 필요하다. 소스 노드는 하나의 심볼

을 하나의 링크를 통해 전송한다. 각 링크는 단위 시간 동안 하나의 심볼만 전송할 수 있고, 각 노드에서의 신호 지연 시간은 없다고 가정한다. 소스 데이터 심볼은 h 개이고, 목적 노드의 개수는 N 개이다 [8].

3. 1. 멀티캐스트 네트워크 코딩

(그림 5)와 같은 멀티캐스트 네트워크를 고려한다. 소스 노드는 S 이고, 소스 데이터 심볼은 α_1, α_2 이고, 목적 노드는 F, E, K 이다. 소스 노드에서 나가는 링크의 수는 2개이고, 각 목적 노드로 들어오는 링크의 수도 2개이다. 최대 전송 효율은 2가 된다. 즉, 단위 시간 동안 각 목적 노드로 전달되는 데이터의 양은 2이다.

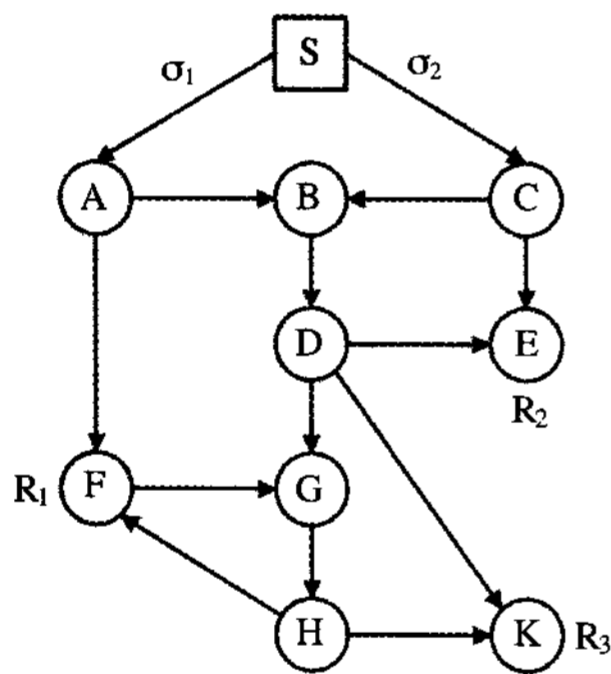
먼저 라우팅 기법을 통해서 최대 전송 효율을 얻을 수 있는지 알아보자. 소스 노드에서 출발한 두 개의 데이터 경로가 서로 겹치지 않으면서 목적 노드로 도착해야 한다. 목적 노드 F 는 데이터 α_1 을 링크 ($S \rightarrow A \rightarrow F$)를 통해서 얻을 수 있고, 데이터 α_2 를 링크 ($S \rightarrow C \rightarrow B \rightarrow D \rightarrow G \rightarrow H \rightarrow F$)를 통해서 얻을 수 있다. 목적 노드 E 는 데이터 α_1 을 링크 ($S \rightarrow A \rightarrow B \rightarrow D \rightarrow E$)를 통해서 얻을 수 있고, 데이터 α_2 를 링크 ($S \rightarrow C \rightarrow E$)를 통해서 얻을 수 있다. 마지막으로 목적 노드 K 는 데이터 α_1 을 링크 ($S \rightarrow A \rightarrow F \rightarrow G \rightarrow H \rightarrow K$)를 통해서 얻을 수 있고, 데이터 α_2 를 링크 ($S \rightarrow C \rightarrow B \rightarrow D \rightarrow K$)를 통해서 얻을 수 있다. 라우팅 기법으로 각 목적 노드로 두 개의 데이터를 전달하기 위해서는 두 개의 데이터가 동시에 들어오는 노드 B 와 노드 G 에서 시간을 나누는 것이 필요하다. 이로 인해 라우팅 기법으로는 최대 전송 효율을 얻을 수 없다.

다음으로는 노드에서의 코딩 기법을 통해 최대 전송 효율을 얻을 수 있는지 알아보자. 앞서 살펴본 바와 같이 노드 B 와 노드 G 의 데이터 정체 현상을 해결해야 한다. 노드 B 와

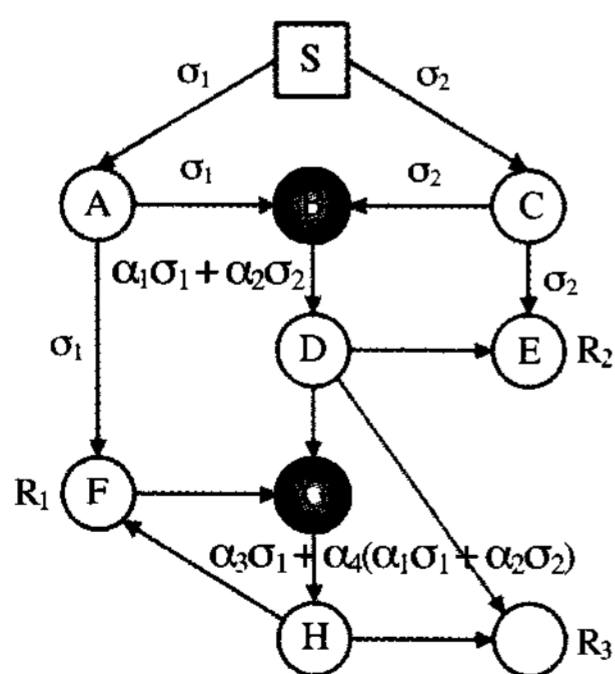
노드 G에서의 입력 데이터에 대한 선형 결합을 출력 데이터로 고려한다. (그림 6)과 같이 노드 B는 링크 BD로 $a_1\sigma_1 + a_2\sigma_2$ 을 전송하고, 노드 G는 링크 GH로 $a_3\sigma_1 + a_4(a_1\sigma_1 + a_2\sigma_2)$ 을 전송한다. 나머지 중간 노드는 들어오는 데이터에 대해 라우팅만 수행한다. 만약 목적 노드 F, E, K에서 두 개의 데이터 심볼 σ_1 과 σ_2 를 복원할 수 있도록 계수 $\{\alpha_k\}$ 를 선택한다면, 주어진 네트워크에서의 네트워크 코딩을 얻을 수 있다. 다음 절에서는 이러한 멀티캐스트 네트워크에서 네트워크 코딩을 항상 얻을 수 있는지에 대해 알아본다.

3. 2. 네트워크 코딩의 기본 정리

위와 같이 특정 중간 노드에서의 선형 결합만으로 주어진 멀티캐스트 네트워크의 최대 전송 효율을 항상 얻을 수 있는지 이론적으로 알아보자. 이를 위해 몇 가지 정의가 필요하다.



(그림 5) 멀티캐스트 네트워크



(그림 6) 멀티캐스트 네트워크 : 선형결합

정의 1. [Local Coding Vector]

링크 e 에 대한 local coding vector $c'(e)$ 는 링크 e 로 나가는 데이터에서 입력 데이터 심볼에 곱해지는 선형계수 벡터이다. 이 때, 벡터의 차원은 노드의 입력 링크의 수가 된다.

(그림 6)에서 링크 BD와 링크 GH에 대한 local coding vector는 각각 $c'(BD)=[\alpha_1 \ \alpha_2]$ 와 $c'(GH)=[\alpha_3 \ \alpha_4]$ 이다. 다른 링크에 대한 local coding vector는 입력 데이터를 그대로 출력하기 때문에 $c'(e) = 1$ 이 된다.

정의 2. [Global Coding Vector]

링크 e 에 대한 global coding vector $c(e)$ 는 링크 e 로 나가는 소스 데이터 심볼의 계수 벡터이다. 이 때, 벡터의 차원은 소스 데이터 심볼의 수가 된다.

(그림 6)에서 링크 BD와 링크 GH에 대한 global coding vector는 각각 $c(BD)=[\alpha_1 \ \alpha_2]$ 와 $c(GH)=[\alpha_3 \ \alpha_4]$ 이다.

정의 3. [Decoding Matrix]

소스 노드의 데이터 심볼을 σ_i 라고 하고, 목적 노드 j 의 i 번째 입력 링크의 심볼을 ρ_i^j 라고 하자. 목적 노드 j 에 대한 복호 행렬 (decoding matrix) C_j 는 목적 노드 j 에서의 global coding vector들을 입력 링크의 순서대로 행벡터로 나열한 행렬이다. 이 때, 목적 노드 j 는 소스 데이터 심볼을 복원하기 위해 다음과 같은 연립 일차 방정식을 풀어야 한다.

$$\begin{bmatrix} \rho_1^j \\ \rho_2^j \\ \vdots \\ \rho_h^j \end{bmatrix} = C_j \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_n \end{bmatrix}$$

(그림 6)에서 각 목적 노드에 대한 복호 행렬은 다음과 같다.

$$C_1 = \begin{bmatrix} c(AF) \\ c(HF) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \alpha_3 + \alpha_2\alpha_4 & \alpha_2\alpha_4 \end{bmatrix},$$

$$C_2 = \begin{bmatrix} c(CE) \\ c(DE) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \alpha_1 & \alpha_2 \end{bmatrix},$$

$$C_3 = \begin{bmatrix} c(DK) \\ c(HK) \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 \\ \alpha_3 + \alpha_2\alpha_4 & \alpha_2\alpha_4 \end{bmatrix},$$

여기서 목적 노드 번호 1, 2, 3은 각각 F, E, K을 나타낸다. 목적 노드에서 소스 데이터 심볼을 복원하기 위해서는 연립 일차 방정식의 해를 구해야 한다. 즉, 복호 행렬들의 역행렬이 존재해야 된다. 주어진 멀티캐스트 네트워크에서 코딩 기법의 존재성 문제는 모든 복호 행렬들의 역행렬이 존재하도록 계수 $\{\alpha_k\}$ 를 찾는 문제와 같다. 다음의 네트워크 코딩의 기본 정리를 얻는다.

[네트워크 코딩 기본 정리] 멀티캐스트 네트워크에서 복호 행렬 C_j 의 역행렬이 항상 존재하도록 계수 $\{\alpha_k\}$ 들이 항상 존재한다. 즉, 주어진 멀티캐스트 네트워크의 최대 전송 효율을 얻을 수 있는 코딩 기법은 선형결합의 형태로 항상 존재한다.

네트워크 코딩 기본 정리는 [6]에서 처음 소개되었고, [7]에서 대수적인 방법으로 증명되었다. 그리고, [8]에서 sparse zeros lemma를 이용해 다시 쉽게 설명하고 있다. 본 논문에서는 유한체 (finite field) 이론을 배제한 채 설명하였다. 이러한 코딩 기법을 찾는 방법, 즉 계수 $\{\alpha_k\}$ 를 찾는 방법은 주어진 시간 안에 찾을 수 있음이 알려져 있다 [9]. 각 노드에서 계수 $\{\alpha_k\}$ 를 랜덤하게 사용해도 높은 확률로 최대 전송 효율을 만족하는 네트워크 코딩 기법이 된다 [10].

본 장에서는 멀티캐스트 네트워크 코딩은 네트워크 코딩, 즉 노드에서의 선형 결합을 이용하면, 최대 전송 효율을 얻을 수 있음을 알았다. 하지만, 이론적인 네트워크 코딩의 접근 방법은 각 노드 사이의 링크에 오류가 없음을 가정하였지만, 실제 네트워크에서 노드 사이 링크의 오류는 피할 수 없다 [11]. 실제 네트워크에 적합한 코딩 기법을 위해 주어진 네트워크에서 네트워크 코딩의 복잡도에 관한 다양한 고찰이 있었다 [12][13]. 다음 장에서는 실제 네트워크에 네트워크 코딩을 적용하기 위한 노력들에 대해 알아보자.

IV. 네트워크 코딩의 실제적인 적용

앞장에서 살펴본 네트워크 코딩은 이론적인 접근에 대해 알아 보았다. 이론적인 네트워크 코딩은 네트워크 전체의 데이터 동기가 맞고, 코딩 벡터와 복호 행렬을 얻는 방법이 중앙 제어 (centralized) 방식이었다. 즉, 주어진 네트워크에 맞는 네트워크 코딩을 고정시켜 놓고 사용하게 된다. 그러나, 실제 네트워크는 각 노드에서 패킷 손실, 지연으로 인해 '데이터 동기가 맞지 않고, 네트워크 연결 상태가 수시로 변한다. 따라서, 수시로 변하는 네트워크 연결 상태에 맞는 네트워크 코딩의 적용은 쉽지 않다. 실제 네트워크 상에서 네트워크 코딩을 적용하려면 이러한 많은 어려움이 있다. 네트워크 코딩을 실제 네트워크에 적용하기 위한 노력으로 랜덤 코딩 (random coding), 패킷 태깅 (packet tagging), 버퍼링 (buffering) 등이 있다.

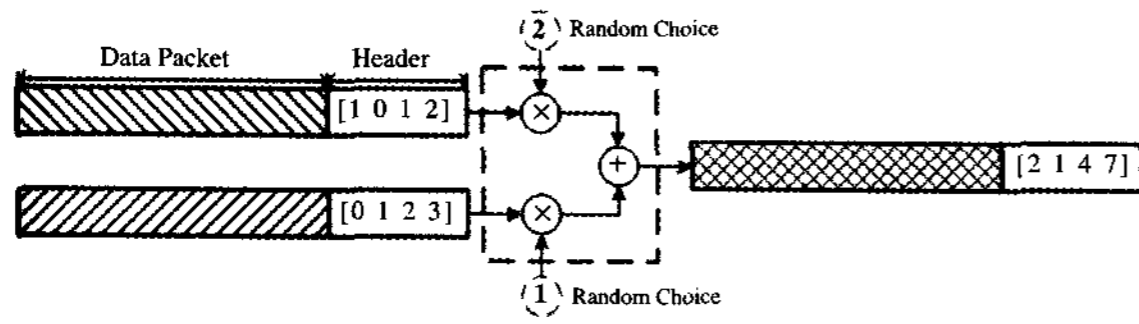
4. 1. 랜덤 코딩

네트워크 코딩은 각 목적 노드에서의 복호 행렬의 역행렬이 존재하도록 노드에서의 local coding vector를 잘 결정해야 한다. 이것은 전체 네트워크의 연결 상태를 알아야 가능한 것이지만, 네트워크의 연결 상태는 수시로 변화하기 때문에 실제적으로는 적합한 방법이 아니다. 주어진 네트워크 노드에서 local coding vector 성분을 랜덤하게 결정해도 된다 [14]. 이 때, 코딩 벡터의 성분들은 독립적이고 동일한 확률을 가진다. 예를 들면, 데이터 심볼이 유한체 2^8 에 속한 원소이고, 코딩 계수가 유한체 2^{16} 에 속한 원소이면, 복호 행렬은 99.6% 정도 가역 (invertible) 행렬이 된다. 필요한 노드에서 랜덤하게 local coding vector를 결정하고, 선형 결합만을 수행하면 된다. 그러므로, 랜덤 선형 코딩은 실제 네트워크에 네트워크 코딩을 적용하기 위해 필요하다.

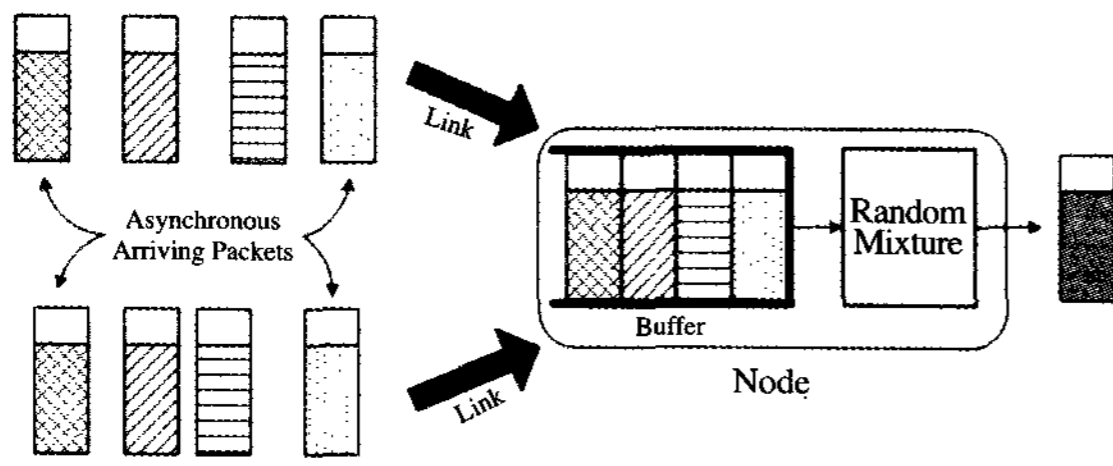
4. 2. 패킷 태깅

네트워크 코딩은 각 목적 노드에서 데이터 심볼을 복원하기 위해서 복호 행렬, 즉 global coding vector를 알고 있어야 한다. 링크 e 를 지나는 모든 데이터 패킷에 global coding vector를 붙여서 전송한다 [15]. 예를 들면, (그림 7)과 같이 코딩 벡터를 데이터 패킷 앞에 붙여서 링크를 통해 전송하

V. 네트워크 코딩의 응용



(그림 7) 랜덤 코딩과 패킷 태깅



(그림 8) 버퍼링

면 된다. 데이터 패킷이 충분히 길면, 코딩 벡터의 길이에 대한 부담은 크지 않다. 그리고, 목적 노드에서 네트워크의 연결 상태나 복호 행렬을 따로 얻을 필요가 없다. 이러한 이유로 패킷 태깅은 네트워크 코딩이 실제 네트워크에 적용 가능하도록 만들어준다.

4. 3. 버퍼링

네트워크 코딩은 각 노드에 도착하는 데이터 패킷의 동기가 정확하다고 가정한다. 그러나, 실제 네트워크에서는 패킷 손실, 패킷 충돌, 트래픽 초과 등으로 인해 각 노드에 도착하는 데이터 패킷의 수는 일정하지 않다. 또한 소스 노드에서는 단위 시간 동안 데이터 심볼을 연속적으로 만들어낸다. 각 노드에서 패킷 수신 동기가 맞기 어렵다. 소스 노드에서 데이터 심볼을 발생하는 단위 시간을 generation이라고 한다 [15]. 각 패킷의 헤더에 generation 순서를 붙인다. (그림 8)은 버퍼링을 나타낸 그림이다. 각 노드에 도착한 패킷은 generation 순서에 따라 buffering을 이용해 동기를 잡는다. 과 같이 두 개의 링크로부터 들어오는 패킷을 generation에 따라 버퍼에 순서대로 쌓는다. 그리고, generation에 따라 랜덤 코딩해서 링크로 내보낸다. 각 노드에서 나중에 도착한 이전 generation의 패킷은 버린다.

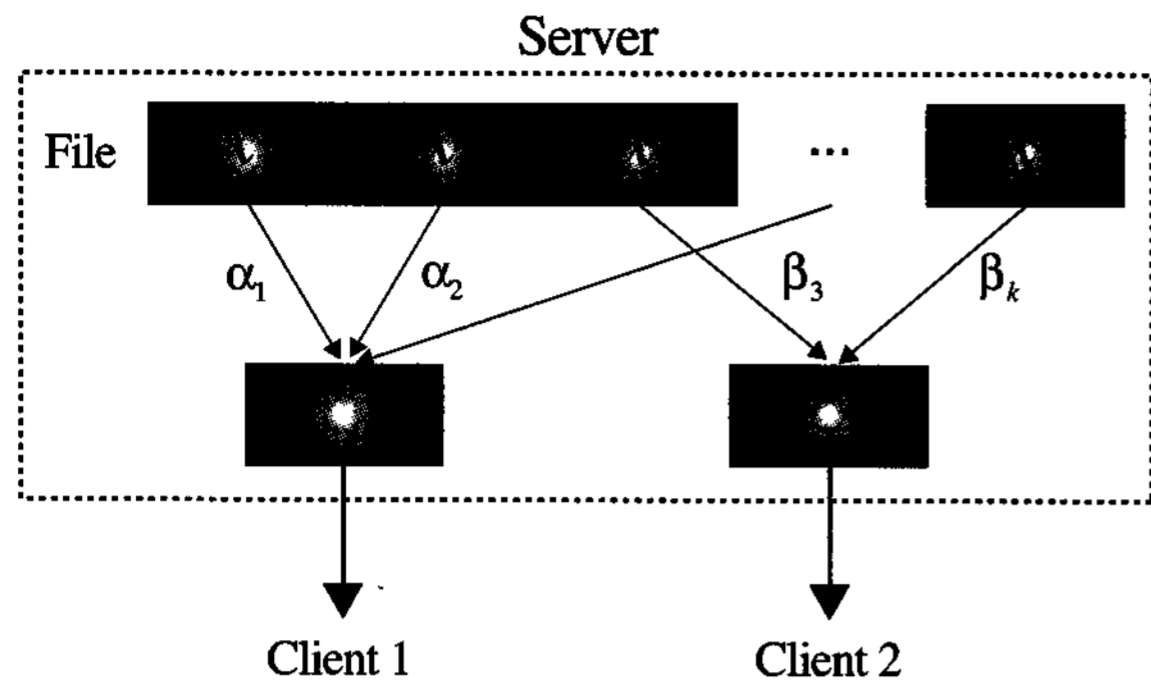
네트워크 코딩의 실제 적용을 위해서는 아직 많은 어려움이 있다. 최근 네트워크 코딩에 대한 다양한 응용이 소개되었다. 본 논문에서는 네트워크 코딩이 응용될 수 있는 P2P 네트워크 상에서의 파일 다운로드, 무선 네트워크 상에서의 네트워크 코딩 기법 및 네트워크-채널 결합 코딩 기법에 대해 설명한다. 그리고, 그 밖의 네트워크 코딩 응용에 대해 간략하게 언급한다.

5. 1. 파일 다운로드 : Avalanche

네트워크를 이용하여 서버로부터 하나의 컴퓨터로 파일을 다운로드 하는 것은 유니캐스트 네트워크로 볼 수 있고, 많은 사용자가 동일한 데이터 파일을 다운로드 하는 것은 멀티캐스트 네트워크로 간주할 수 있다. 이러한 관점에서 볼 때, 네트워크 코딩은 데이터 전송 효율을 증가시키고, 파일 다운로드 시간을 줄일 수 있다. 네트워크 코딩에 기반한 새로운 파일 다운로드 프로토콜이 Avalanche이다 [16]. (그림 9)와 같이 파일을 동일하게 h 개의 조각으로 나눈다. 각 사용자는 랜덤 코딩 파일 조각 하나만 다운로드 한다. 이 때, 각 데이터 패킷의 global coding vector는 헤더에 붙인다. Global coding vector를 이용하여 각 사용자가 원하는 데이터 파일을 복원할 수 있다. Avalanche의 평균 파일 다운로드 시간은 코딩을 적용하지 않았을 때에 비해 약 20~30% 개선됨을 알 수 있다.

5. 2. 무선 네트워크

두 개의 노드가 서로의 데이터를 중간 노드를 거쳐서 전달하는 무선 네트워크를 고려한다. 하나의 노드가 무선 자원을 사용하고 있을 때, 다른 노드는 사용할 수 없다는 것을 가정한다. (그림 10)과 같이 코딩이 없을 때는 한 노드가 다른 노드에게 전달하기 위해서는 중간 노드를 통해 2번의 단위 시간이 소요된다. 따라서, 상호 간의 데이터를 공유하기 위해서는 4번의 단위 시간이 필요하다. 그러나, 코딩이 있을 때는 서로의 데이터를 공유하기 위해서는 3번의 단위 시간이면 충분하다. 중간 노드는 각 단위 시간 동안 얻은 두 데이



(그림 9) 파일 다운로드 : Avalanche

터를 XOR 연산을 해서 다음 단위 시간에 양쪽 노드로 전송한다. 이것은 무선 자원 효율성과 데이터 전송 효율 측면에서 장점을 가진다. 그러나, 채널의 감쇄 등으로 인한 무선 네트워크에서의 링크 오류는 피할 수 없다.

5. 3. 네트워크-채널 결합 코딩

네트워크 코딩 기법은 각 링크의 오류가 없다는 가정에서 출발한다. 실제 네트워크에서는 링크 오류를 피할 수 없다. 각 링크에 오류 정정 부호인 채널 코딩을 사용하면, 링크 오류를 원하는 만큼 줄일 수 있다. 즉, 링크 오류가 없는 네트워크의 데이터 전송 효율을 높이는 네트워크 코딩 기법과 네트워크의 데이터 전송 효율은 떨어지지만, 각 링크 오류를 줄일 수 있는 채널 코딩 기법을 같이 고려하는 것이 네트워크-채널 결합 코딩이다. 양방향 릴레이 (two-way relay) 채널에서의 터보 네트워크 부호 [17]와 다중 접속 (multiple access) 채널에서 LDPC 부호 [18], convolutional 부호 [19]에 관한 연구가 진행되었다.

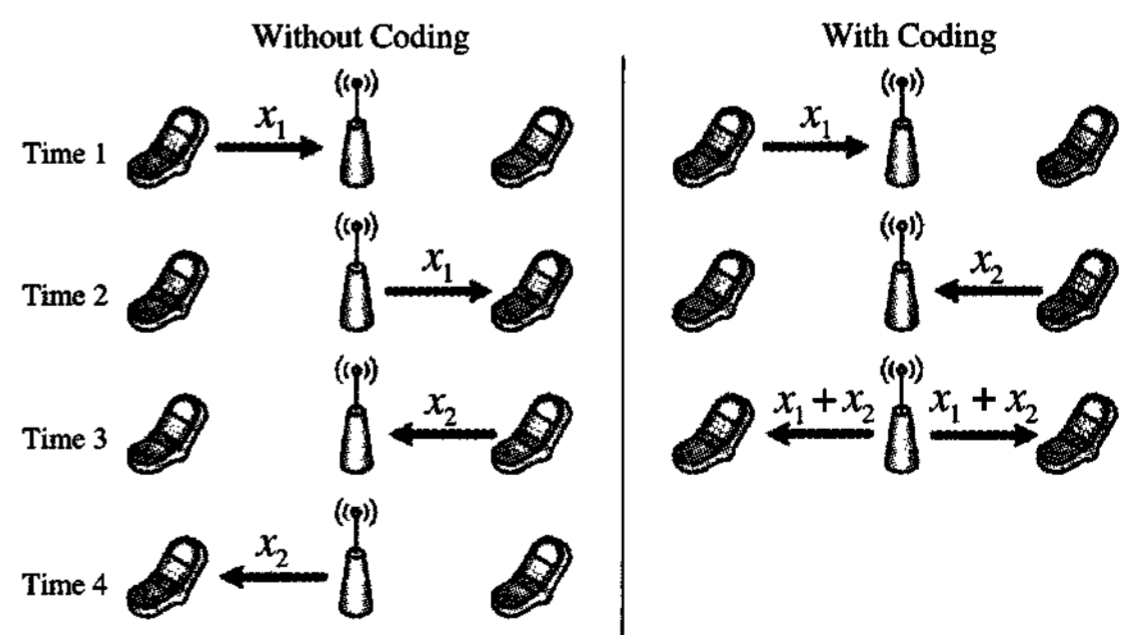
양방향 릴레이 채널에서의 터보 네트워크 부호에 대해서 간략하게 알아보자 [17]. 양방향 릴레이 채널 모델은 사용자 (mobile station, MS)와 기지국 (base station, BS)은 릴레이 (relay station, RS)의 도움으로 각각 독립적인 2개의 채널을 통해 서로에게 데이터를 전송한다. 이 채널의 특징은 각 노드는 다른 노드들에게 동시에 정보를 전달한다. 터보 네트워크 부호는 크게 두 가지로 구성된다. 네트워크 부호는 릴레이에서 사용자와 기지국으로부터 온 데이터를 각각 복호 비트를 XOR하여 인코딩한 후, 전송한다. 터보 부호는 분산 터보 부호 (distributed turbo codes)의 의미이다. 즉, 구성 부

호인 컨볼루션 부호 (convolutional codes)를 사용자와 릴레이 각각 가지고, 기지국은 터보 복호하여 데이터를 얻는 방식이다 [20]. (그림 11)은 터보 네트워크 부호의 블록도이다. 터보 네트워크 부호는 첫번째 시간에 사용자가 컨볼루션 부호 데이터를 릴레이와 기지국으로 전송한다. 두번째 시간에 기지국이 컨볼루션 부호 데이터를 릴레이와 사용자에게 전달한다. 세번째 시간에 릴레이는 각각의 복호 데이터를 XOR하여 기지국과 사용자에게 전송한다. 각각의 수신기는 상대방의 데이터를 터보 복호하여 얻어낸다.

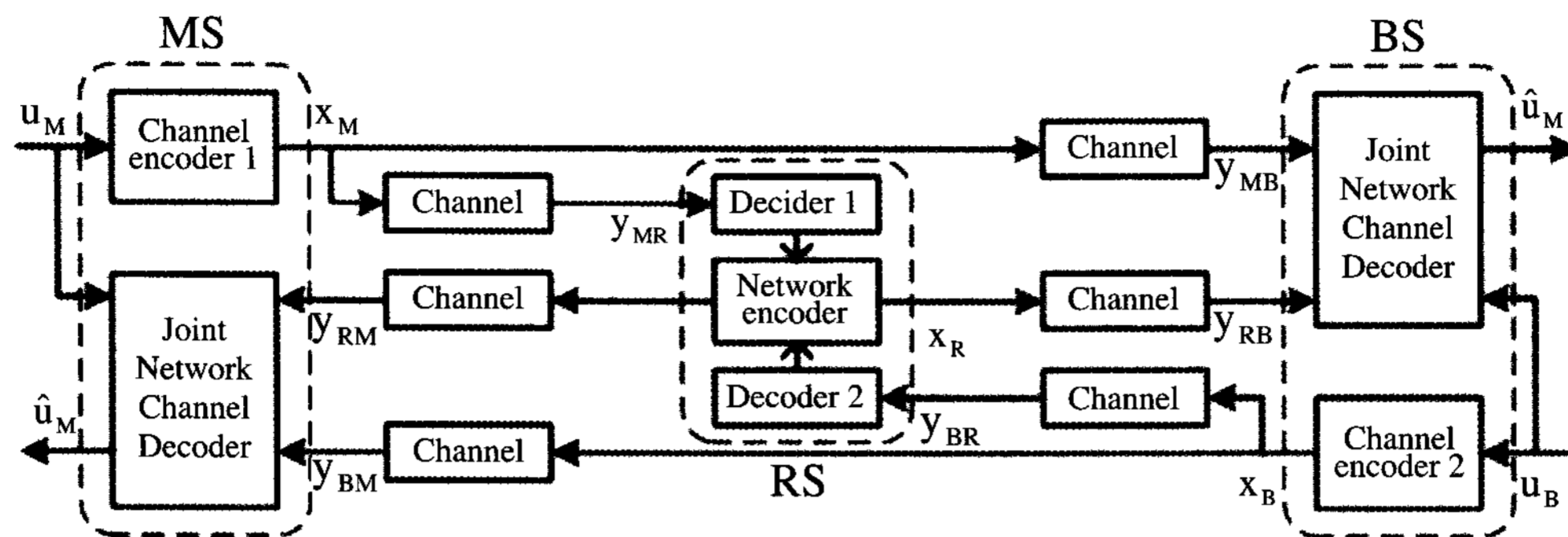
이 밖에도 네트워크 코딩의 응용 분야는 더 있다. 주어진 네트워크의 소스 노드에서 상관성이 있는 두 데이터를 각 목적 노드에 전송할 때 최대 전송 효율을 만족하는 네트워크-소스 결합 코딩 기법에 대한 연구가 진행되고 있다 [21]. 네트워크 코딩은 네트워크 보안 [22]과 센서 네트워크 [23]에 적용해 볼 수 있다.

VI. 결 론

본 논문에서는 주어진 네트워크의 최대 전송 효율을 얻을 수 있는 네트워크 코딩에 대해서 알아 보았다. 멀티캐스트 네트워크에서는 라우팅 기법만으로는 최대 전송 효율을 얻을 수 없으며, 이에 새로운 형태의 전송 기법이 필요하게 된다. 네트워크 코딩은 멀티캐스트 네트워크에 항상 존재하게 되고, 최대 전송 효율을 항상 만족함을 이론적으로 접근하였다. 그리고, 실제 네트워크에서 네트워크 코딩 기법을 적



(그림 10) 무선네트워크



(그림 11) 양방향 릴레이 네트워크에서 터보 네트워크 부호

용하기 위해 랜덤 코딩, 패킷 태깅, 버퍼링 등에 대해서 알아보았다. 또한 다른 분야에 응용되고 있는 네트워크 코딩 기법에 대해서도 알아보았다.

소스 노드가 여러 개일 때, 즉 multi-source 멀티캐스트 네트워크에서의 네트워크 코딩이 존재하는지 [24], 각 목적 노드가 원하는 데이터의 양이 같지 않을 때, 네트워크 코딩이 존재하는지 여전히 이론적으로 마련되어 있지 않다. 이처럼 다양한 환경에 따른 최대 전송 효율을 만족할 수 있는 네트워크 코딩에 관한 연구는 네트워크, 통신, 전산 등과 관련되어 현재도 활발히 진행 중에 있다.



- [1] P. A. Chou and Y. Wu, "Network coding for the internet and wireless networks," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 77-85, Sep. 2007.
- [2] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399-404, 1956.
- [3] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the Association for Computing Machinery*, vol. 19, no. 2, pp. 248-264, Apr. 1972.
- [4] K. Jain, M. Mahdian, and M. R. salavatipour, "Packing Steiner trees," *Symposium on Discrete Algorithms*, 2003.
- [5] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204-1216, July 2000.
- [6] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371-381, Feb. 2003.
- [7] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782-795, Oct. 2003.
- [8] C. Fragouli and E. Soljanin, *Network coding fundamentals*, now Publishers, Inc., 2007.
- [9] S. Jaggi, et al., "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973-1982, June 2005.
- [10] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *Journal of the ACM*, vol. 27, no. 4, pp. 701-717, Oct. 1980.
- [11] R. Koetter and F. Kschischang, "Coding for errors and erasures in random network coding," *IEEE International Symposium on Information Theory (ISIT2007)*, Nice, France, June 2007.
- [12] C. Chekuri, C. Fragouli, and E. Soljanin, "On average throughput and alphabet size in network coding," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2410-2424, June 2006.

[13] M. Langber, A. Sprintson, and J. Bruck, "The encoding complexity of network coding," *IEEE/ACM Transactions on Networking*, vol. 14, pp. 2386-2397, June 2006.

[14] T. Ho, et al., "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413-4430, Oct. 2006.

[15] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," *Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2003.

[16] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," *INFOCOM*, Miami, FL, Mar. 2005.

[17] C. Hausl and J. Hagenauer, "Iterative network and channel decoding for the two-way relay channel," *ICC*, Istanbul, Turkey, June 2006.

[18] C. Hausl, et al., "Iterative network and channel decoding on a tanner graph," *Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sep. 2005.

[19] L. Xiao, T. E. Fuja, J. Kliewer, and D. J. Costello, "A network coding approach to cooperative diversity," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3714-3722, Oct. 2007.

[20] B. Zhao and M. C. Valenti, "Distributed turbo coded diversity for relay channel," *Electronics Letters*, vol. 39, no. 10, pp. 786-787, May 2003.

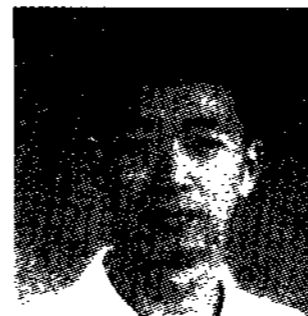
[21] T. Ho, M. Medard, M. Effros, and R. Koetter, "Network coding for correlated sources," *CISS*, Princeton, NJ, Mar. 2004.

[22] C. Gkantsidis and P. R. Rodriguez, "Cooperative security for network coding file distribution," *INFOCOM*, Barcelona, Spain, Apr. 2006.

[23] D. Petrovic, K. Ramchandran, and J. Rabaey, "Overcoming untuned radios in wireless networks with network coding," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2649-2657, June 2006.

[24] X. Yan, R. W. Yeung, and Z. Zhang, "The capacity region for multi-source multi-sink network coding," *ISIT*, Nice, France, June 2007.

약 력



2003년 성균관대학교 수학과 석사
 2005년 포항공과대학교 수학과 이학석사
 2005년 ~ 현재 한국정보통신대학교 공학부 석박사통합과정
 관심분야: MIMO, 오류정정부호

김재홍



1983년 연세대학교 전자공학과 공학사
 1985년 연세대학교 전자공학과 공학석사
 1997년 Georgia Institute of Technology 공학박사
 1985년 ~ 2002년 삼성전자 책임연구원, 수석연구원
 2002년 ~ 현재 한국정보통신대학교 공학부 부교수
 관심분야: MIMO, OFDM, 오류정정부호

박현철