네트워크 코딩 기술

Network Coding in Technology

홍석범 · 노종선
Seokbeom Hong · Jong-Seon No

본 논문에서는 네트워크 상의 정보 전송에서 최근 들어 라우팅기법의 대안으로 대두되고 있는 네트워크 코딩기법에 대한 전반적인 내용을 소개하였다. 네트워크 코딩이란 네트워크 상의 각 중간 노드가 입력으로 들어온 정보들을 결합하여 생성된 새로운 정보를 출력으로 전송하는 기법이다. 이는 입력된 정보를 그대로 출력으로 전달하는 역할만을 수행하는 기존의 라우팅 기법을 확장한 것으로 생각할 수 있다. 일반적으로 멀티캐스트 네트워크에서 라우팅기법만을 이용해서는 전송가능한 최대 전송률을 달성하는 것이 불가능하다. 본 논문에서는 네트워크 코딩을 이용할 경우 멀티캐스트 네트워크에서 최대 전송률을 달성하는 전송이 가능하다는 것을 확인하고 이러한 전송이 가능하도록 하는 선형 네트워크 코드의 실제 설계방법에 대해서도 제시하였다. 또한 다양한 형태의 오류가 발생하는 네트워크에서 최적의 전송을 수행할 수 있도록 하는 네트워크 코드의 설계방법에 대해서도 서술하였으며 랜덤 네트워크 코딩과 무선 네트워크 상에서의 네트워크 코딩기법에 대해서도 소개하였다.

주제어: 네트워크 코딩, 라우팅, 멀티캐스트 네트워크

In this paper, we introduce the overview of network coding techniques which are recently on the rise for an alternative idea of routing scheme for data transmission on the network. Network coding is the scheme that each intermediate node on the network transmits the output symbol constructed by combination of the incoming symbols. Network coding can be viewed as extension of routing, the scheme that each node just forwards input symbol to output. It is impossible to achieve capacity with only routing scheme in general multicast network. In this paper, we confirm that network coding scheme can achieve capacity in multicast network, and suggest the design method of such linear network codes. We also present the design methods of optimal network codes on network with variety kinds of errors and introduce the concept of random network coding and network coding on the wireless network.

Keywords: Network coding, Routing, Multicast network

I. 서 론

기존에는 네트워크 상의 노드간에 정보를 전송하는 과정에서 단순히 각 노드가 입력된 정보를 그대로 출력으로 전달해 주는 라우팅기법이 주로 사용되었다. 그러나 최근 들어 네트워크 상의 노드에서 새로운 방식으로 동작하는 네트워크 코딩(network coding)기법[1]이 등장하였다. 네트워크 코딩이란 네트워크 상의 중간 노드(intermediate node)가 입력으로 들어온 정보를 그대로 전달해 주는 것이 아니라 입력정보들을 결합하여 생성된 새로운 출력정보를 다음 노드로 전달하는 기법이다. 또한 리시버 노드에서는 이렇게 결합되어 전달된 정보를 복호하여 소스 노드에서 전송한 정보를 얻어내게 된다. 네트워크 코딩은 기존의 라우팅기법을 확장한

논문번호: TR10-035, 논문접수일자:2010.04.19, 논문수정일자:2010.04.22, 논문게재확정일자:2010.05.12
홍석범, 노종선: 서울대학교

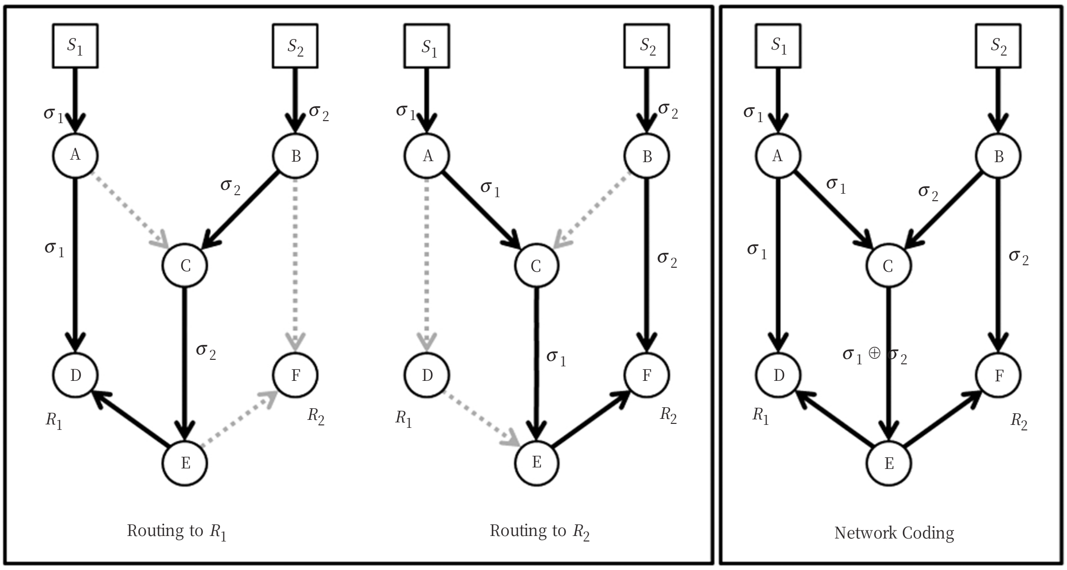

그림 1. 버터플라이 네트워크에서 라우팅기법과 네트워크 코딩기법간의 비교

개념으로 라우팅은 가장 간단한 네트워크 코딩의 예시로 생각할 수 있다.

네트워크 코딩을 사용하는 경우 라우팅만을 사용하는 경우에 비하여 전송효율 측면에서 장점을 가진다. 이는 그림 1의 버터플라이 네트워크(butterfly network)의 예를 통해서 확인할 수 있다. 그림 1의 네트워크는 2개의 소스 노드 S_1, S_2가 각각 가지고 있는 정보 σ_1, σ_2를 2개의 리시버 노드 R_1, R_2 양쪽 모두로 전송하는 것을 목표로 하는 네트워크이다. 분석의 편의를 위해 노드간의 정보 전송 시에 발생하는 시간 지연은 없는 것으로 가정하겠다. 그림에서 확인할 수 있는 바와 같이 각 노드에서 라우팅만을 사용하는 경우에는 1회의 단위 시간에 모든 정보를 전송하는 것은 불가능하다. 왜냐하면 링크 CE가 서로 다른 정보 σ_1, σ_2를 각각 전송하는 경로내에서 중복되기 때문이다.

그러나 매우 간단한 네트워크 코딩기법을 적용하는 것만으로도 1회의 단위 시간동안 모든 정보를 전송하는 것이 가능하다. 그림과 같이 노드 C에서 입력 정보 σ_1, σ_2에 XOR(exclusive-or) 연산을 수행하여 노드 E로 정보를 전송하는 경우 링크 CE는 2가지 정보를 동시에 전송하게 되므로 라우팅에서 발생하던 링크의 중복 문제를 해결할 수 있다. 이러한 기법을 사용하는 경우 리시버 노드 R_1에서는 σ_1, $\sigma_1 \oplus \sigma_2$를 수신하게 되는데 이 정보에 다시 XOR 연산을 수행하여 σ_2를 복호할 수 있다. 또한 리시버 노드 R_2에서도 이와 유사한 방법으로 소스 노드에서 송신한 정보 σ_1, σ_2를 성공적으로 복원하는 것이 가능하다.

네트워크 코딩은 무선 자원 사용 측면에서도 장점을 가진다. 무선 네트워크 환경에서의 전송은 주로 동시에 범위 내에 있는 모든 노드로 정보를 전달하는 브로드캐스팅(broadcasting) 방식을 통해서 이루어진다. 따라서 라우팅만을 사용하는 경우에는 특정한 정보를 하나의 노드로 전달하기 위해 그 정보를 필요로 하지 않는 다른 노드들까지 이를 수신하게 되는 경우가 흔히 발생하게 된다. 반면 네트워크 코딩을 사용하는 경우 여러 정보를 결합하여 만든 패킷을 브로드캐스트하기 때문에 동시에 많은 수의 노드에 의미있는 정보를 전달하는 것이 가능하여 정보 전송에 필요한 총 전송횟수를 감소시킬 수 있다.

2개의 노드 A, C가 중계 노드 B를 통하여 서로 정보를 교환하는 경우의 예가 그림 2에 나타나 있다. 그림에서 확인할 수 있는 바와 같이 라우팅만을 사용하는 경우에는 적어도 4번 이상의 전송이 필요하지만 네트워크 코딩을 사용하는 경우에는 3번의 전송만으로 정보 교환을 완료할 수 있다. 이와 같이 네트워크 코딩을 사용하는 경우 필요한 정보 전송의 횟수가 줄어드는 효과를 얻을 수 있고 이에 따라 에너지 효율이나 시간 지연 등의 측면에서 라우팅을 사용하는 경우에 비해 이점을 가지게 된다.

또한 네트워크 코딩은 도청(wiretapping)과 같은 외부 공격에 대한 보안 측면에서도 장점을 가진다. 라우팅의 경우에는 임의의 1개의 링크에 도청을 수행하는 것만으로도 그 링크를 통해 전송되는 정보를 알아내는 것이 가능하다. 그러나 네트워크 코딩을 사용하여 각 링크를 통해 전송되는 정보가 입력정보들의 조합으로 이루어져 있는 경우 적은 수의 링크를 도청하는 것만으로는 소스 노드에서 송신한 정보를 복원해내는 것이 쉽지 않다.

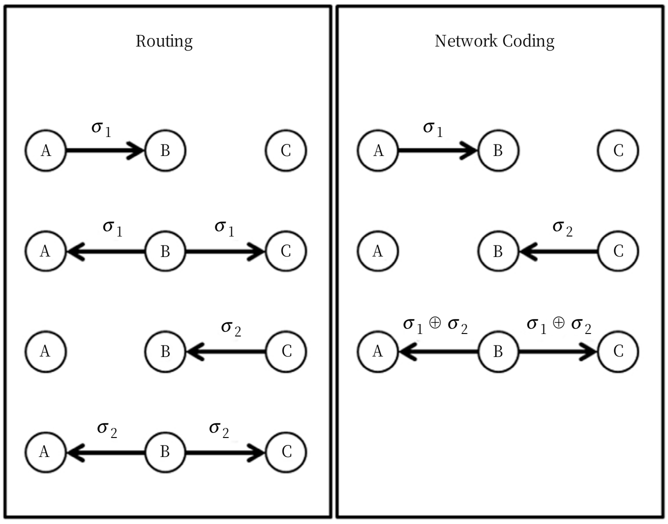

그림 2. 무선 중계 네트워크에서 라우팅기법과 네트워크 코딩기법간의 비교

네트워크 코딩이 가지고 있는 문제점으로는 노드에서의 높은 복잡도와 오류 전파(error propagation) 문제 등이 있다. 네트워크 상의 노드가 라우팅만을 수행할 때에는 정보를 수신하고 수신한 정보를 그대로 전달하는 역할만을 담당하기 때문에 매우 간단한 구조의 노드를 사용하는 것으로 충분하다. 그러나 네트워크 코딩을 사용하는 경우의 노드는 유한체(finite field) 상에서의 연산이 가능해야 하고 경우에 따라 추가적인 저장 공간을 필요로 하기도 한다. 또한 네트워크 내의 하나의 링크에서 오류가 발생하는 경우 이를 수신한 노드에서는 오류가 발생한 정보를 포함한 입력정보들을 결합하여 얻은 출력정보를 다른 노드로 전송하게 된다. 이로 인해 하위에 위치한 여러 노드들로 오류가 포함된 정보가 전송되고 이러한 과정이 반복되면서 빠른 속도로 많은 수의 링크에 오류가 전파된다. 이로 인해 리시버 노드에서 최종적으로 수신한 정보들을 복호하는 과정에서 오류를 정정하기가 상대적으로 더 어려워지게 된다.

네트워크 코딩의 개념은 2000년에 R. Ahlswede 등에 의해 처음으로 제안[1]되었으며 2000년대 중반 정도부터 본격적인 연구가 시작되어 현재까지도 매우 활발하게 연구가 진행되고 있다. 또한 네트워크 코딩 기법 자체에 대한 연구 이외에도 중계 네트워크, 분산 스토리지(distributed storage), 네트워크 정보이론, 채널 코딩, 보안 등 수많은 여타 연구 분야들과의 접목을 통한 연구도 지속적으로 진행되고 있다. 현재 네트워크 코딩은 P2P(peer-to-peer) 네트워크에서의 파일 다운로드 등의 분야에서 실제로 사용되고 있고 향후 IPTV와 같은 다양한 무선통신 시스템에서 활용될 것으로 기대되고 있다.

Ⅱ. 네트워크 코딩의 기본 정리

네트워크 코딩의 기본 정리를 살펴보기에 앞서서 이와 관련된 몇가지 개념에 대하여 소개하도록 하겠다. 네트워크 상의 각 소스 노드로부터 전송되는 정보가 각각 하나씩의 리시버 노드로만 전송되는 형태의 네트워크를 유니캐스트 네트워크라 한다. 반면 각 소스 노드로부터 전송되는 정보가 여러 리시버 노드로 전송되는 경우의 네트워크는 멀티캐스트 네트워크라고 부른다. 만약 네트워크 상에 존재하는 모든 소스 노드들이 가진 정보가 전부 모든 리시버 노드로 전송되는 경우를 가정하면 리시버 노드가 1개이면 유니캐스트 네트워크, 2개 이상이면 멀티캐스트 네트워크가 된다.

네트워크 상의 소스 노드 S와 리시버 노드 R에 대하여 제거할 경우 S와 R사이의 연결이 끊어지도록 하는 링크들의 집합을 S와 R사이의 cut이라고 정의한다. 또한 cut에 속하는 링크들의 채널용량의 합을 cut의 값이라 하고 가장 작은 값을 갖는 cut 을 min-cut이라고 정의한다. 이 때 다음과 같은 정리가 성립한다.

- 정리 1(Min-Cut Max-Flow Theorem)[2]: 네트워크 상의 소스 노드 S와 리시버 노드 R 사이의 min-cut 값이 h라고 가정하자. 이 때 S에서 R로 전송가능한 최대 정보 전송률은 h가 된다. 즉 네트워크 상에서 min-cut의 값과 max-flow의 전송률은 일치한다. 만약 모든 링크가 단위 채널용량을 가지는 경우를 가정하면 S와 R사이에 h개의 서로 중복되지 않는 경로가 존재한다.

c의 채널용량을 가지는 하나의 링크는 단위 채널용

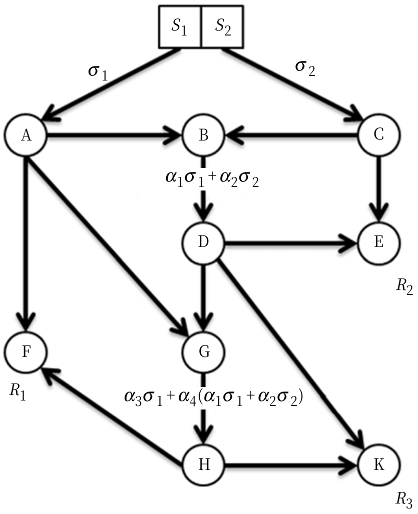

그림 3. 2개의 소스와 3개의 리시버를 가지는 멀티캐스트 네트워크

량을 가지는 c개의 평행한 링크로 변형하여 생각할 수 있다. 따라서 동일한 노드의 쌍 사이에 평행한 링크가 존재하는 것이 가능한 경우 모든 링크가 단위 채널용량을 가진다고 가정할 수 있다. 마찬가지로 r의 전송률을 가지는 소스 노드는 단위 전송률을 가지는 r개의 겹쳐 있는 소스 노드로 변형하여 생각할 수 있고 이에 따라 모든 소스 노드가 단위 전송률을 가진다고 가정할 수 있다. 따라서 이러한 가정 하에서 다음과 같은 네트워크 코딩의 기본 정리를 살펴보도록 하겠다.

- 정리 2(네트워크 코딩의 기본 정리)[1],[3]: h개의 단위 전송률을 갖는 소스 노드와 N개의 리시버 노드로 이루어진 방향성을 가지는 멀티캐스트 네트워크를 가정하자. 이 때 각 소스 노드를 $S_i(i=1,2,\ldots,h)$, 리시버 노드를 $R_j(j=1,2,\ldots,N)$라 명명하겠다. 또한 모든 링크는 단위 채널용량을 가지며 소스 노드들의 집합 $S=\{S_1,S_2,\ldots,S_h\}$로부터 임의의 리시버 노드 R_j 사이의 min-cut 값은 h라 가정하자. 즉 $S_i(i=1,2,\ldots,h)$와 R_j를 연결하는 h개의 경로들을 서로 중복되지 않게 설정하는 것이 가능하다. 이 때 각 노드가 충분히 큰 유한체 상에서 선형 네트워크 코딩을 사용하는 경우 전송률 h를 갖는 멀티캐스트를 가능하도록 하는 네트워크 코딩기법이 항상 존재한다.

여기서 선형 네트워크 코딩이란 각 노드에서 입력 정보들을 선형 결합하여 출력정보를 생성하는 네트워크 코딩기법을 말한다. 정리 2에서의 네트워크는 위에서 언급한 바와 같이 $S_i(i=1,2,\ldots,h)$와 R_j를 연결하는 h개의 중복되지 않는 경로들이 존재한다. 그러나 서로 다른 리시버 노드에 대한 경로들 사이에서는 중복되는 링크가 발생할 수도 있다. 라우팅기법만을 사용하는 네트워크의 경우 이와 같은 경로의 중복이 발생할 때에 전송률 h를 갖는 멀티캐스트를 수행하는 것이 불가능하다. 그러나 네트워크 코딩을 사용하는 경우 각 노드에서 적절한 선형 네트워크 코딩기법을 적용함으로써 최대 전송률 h를 달성하는 멀티캐스트를 수행할 수 있다는 것을 정리 2를 통해 알 수 있다.

Ⅲ.네트워크 코드의 설계

II장에서는 네트워크 코딩의 기본 정리에 따라 멀티캐스트 네트워크 상에서 최대 전송률을 달성할 수 있도록 하는 선형 네트워크 코드가 존재한다는 것을 확인하였다. 본 장에서는 이러한 선형 네트워크 코드의 실제 설계방법에 대하여 살펴볼 것이다.

선형 네트워크 코딩기법을 사용할 경우 각 노드에서는 입력으로 들어온 각각의 정보에 미리 정해진 계수를 곱한 뒤 이들을 전부 합하여 출력정보로 내보내게 된다. 이러한 과정에서 출력 링크 e에 대하여 위의 계수들을 모아서 만든 벡터를 로컬 코딩 벡터(local coding vector) $c^l(e)$라고 정의한다. 즉 로컬 코딩 벡터란 각 링크가 전송하는 정보가 입력정보들로부터 선형 결합되어 생성되는 방식을 나타내는 벡터로 생각할 수 있다. 또한 글로벌 코딩 벡터(global coding vector) $c(e)$는 링크 e를 통해 전송되는 정보가 소스 노드에서 전송한 정보들의 어떠한 선형 결합으로 표현되는지를 나타내는 벡터로 정의된다.

그림 3의 예시를 통해 로컬 코딩 벡터와 글로벌 코딩 벡터의 개념을 확인할 수 있다. 그림 3의 네트워크에서 네트워크 코딩이 이루어지는 코딩 포인트(coding point)는 링크 BD와 GH이다. 이들의 로컬 코딩 벡터

는 각각 $c^l(BD) = [\alpha_1, \alpha_2]$, $c^l(GH) = [\alpha_3, \alpha_4]$이고 글로벌 코딩 벡터는 각각 $c(BD) = [\alpha_1, \alpha_2]$, $c(GH) = [\alpha_3 + \alpha_1\alpha_4, \alpha_2\alpha_4]$가 된다. 선형 네트워크 코드를 설계하는 문제는 각 코딩 포인트에 대한 로컬 코딩 벡터 및 글로벌 코딩 벡터를 설계하는 문제로 변형하여 생각할 수 있다.

소스 노드 S_i와 리시버 노드 R_j를 연결하는 경로의 마지막 링크의 글로벌 코딩 벡터를 i번째 행으로 갖는 행렬 A_j를 정의하면 다음과 같은 식이 성립한다.

$$\begin{bmatrix} \rho_1^{\,j} \\ \rho_2^{\,j} \\ \cdots \\ \rho_h^{\,j} \end{bmatrix} = A_j \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \cdots \\ \sigma_h \end{bmatrix}$$

이 때 σ_i는 S_i로부터 전송된 소스 심볼이며 $\rho_i^{\,j}$는 S_i와 R_j를 연결하는 경로의 마지막 링크를 통해 전송되는 정보를 의미한다. 즉 $i = 1,2,..,h$에 대하여 σ_i는 소스 노드 S_i가 송신한 정보이고 $\rho_i^{\,j}$는 리시버 노드 R_j가 실제로 수신하는 정보를 나타낸다. 따라서 행렬 A_j는 h개의 소스 노드들로부터 리시버 노드 R_j로의 전송 행렬(transfer matrix)이 된다. 그림 3의 네트워크에서 각 리시버 노드에 대한 전송 행렬을 구해 보면 다음과 같다.

$$A_1 = \begin{bmatrix} 1 & 0 \\ \alpha_3 + \alpha_1\alpha_4 & \alpha_2\alpha_4 \end{bmatrix},\ A_2 = \begin{bmatrix} 0 & 1 \\ \alpha_1 & \alpha_2 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} \alpha_1 & \alpha_2 \\ \alpha_3 + \alpha_1\alpha_4 & \alpha_2\alpha_4 \end{bmatrix}$$

위 결과를 통해 각 전송 행렬들의 행벡터는 글로벌 코딩 벡터로 구성되는 것을 확인할 수 있으며 이에 따라 전송 행렬 내의 각 원소는 로컬 코딩 벡터의 계수들에 대한 식으로 표현됨을 알 수 있다. 리시버 노드 R_j에서 수신한 정보들로부터 기존의 소스 심볼들을 복원해내기 위해서는 전송 행렬 A_j가 역 행렬을 가져야 한다. 이에 따라 최대 전송률 h를 달성하는 멀티캐스트를 수행하기 위해서는 모든 리시버 노드에 대한 전송 행렬이 전부 역행렬을 가지는 행렬이 되도록 글로벌 코딩 벡터를 설계하여야 한다.

이에 따라 정리 2의 결과를 대수적으로 서술해 보면 로컬 코딩 벡터의 계수들의 집합 $\{\alpha_k\}$를 충분히 큰 유한체 상에서 선택할 경우 모든 전송 행렬 A_j가 역행렬을 가지도록 선택하는 것이 가능하다. 다시 말해 $f(\{\alpha_k\}) = \det(A_1)\det(A_2)\cdots\det(A_N) \neq 0$이 되도록 하는 $\{\alpha_k\}$가 존재한다. 이는 sparse zeros lemma[4]에 의해 쉽게 증명된다.

또한 네트워크 상에서의 각 전송이 단위 지연시간을 가진다고 가정할 때 전송 행렬 A_j는 다음과 같이 표현할 수 있다.

$$A_j = C_j(I\text{-}A)^{-1} B$$

이 때 행렬 A는 네트워크 상의 링크들간의 관계를 나타내는 행렬이며 링크의 총 개수 m에 대하여 $m \times m$ 크기의 행렬이다. 행렬 A의 각 원소는 링크의 쌍에 대응되는데 인접하지 않은 링크들에 대응되는 원소는 0의 값을 가지며 인접한 링크들에 대응되는 원소는 로컬 코딩 벡터의 계수 $\{\alpha_k\}$의 값을 가진다. 행렬 B는 소스 노드들로부터 나오는 h개의 링크와 네트워크의 각 링크가 어떠한 방식으로 연결되어 있는지를 표현하는 $m \times h$ 행렬이고 행렬 C_j는 리시버 노드 R_j로 들어가는 h개의 링크와 네트워크의 각 링크간의 연결관계를 나타내는 $h \times m$ 행렬이다. 위 식을 네트워크 코드의 시스템 방정식이라고 부른다.

IV. 오류가 발생하는 경우의 네트워크 코딩

지금까지는 네트워크에서 정보 전송 시에 오류가 발생하지 않는다는 가정하에서 최적의 네트워크 코딩을 수행하는 방법에 대하여 알아보았다. 그러나 실제로 네트워크에서 정보를 전송하는 상황에서는 다양한 형태의 오류가 발생할 수 있다. 전송된 정보에 오류가 추가되면서 변형된 정보가 수신되기도 하고 패킷이 정상적으로 전송되지 못하고 소실(erasure)되기도 하며 때로는 정보를 전송하는 링크 자체가 동작하지 않는 경우도 발생한다. 본 장에서는 이와 같이 다양한 형태의 오류가 발생하는 네트워크에서 어떻게 네트워크 코드를 설계해야 하는지에 대하여 알아보도록 하겠다.

일반적으로 오류가 발생하는 네트워크에서는 채널 코딩을 이용하여 오류를 정정하는 기법이 많이 사용된다. 만약 모든 중간 노드들이 복호화 및 부호화를 수행할 수 있는 능력이 있고 복잡도나 시간 지연 측면에서 제약이 전혀 없는 이상적인 네트워크에서는 네트워크 코딩과 채널 코딩을 각각 분리하여 최적의 기법을 설계함으로써 최적의 전송을 수행하는 것이 가능하다. 그러나 실제로 사용되는 현실적인 네트워크의 경우 채널 코딩과 네트워크 코딩을 종합적으로 고려하여 설계해야만 최적의 전송을 수행하는 것이 가능하다. 따라서 오류가 발생하는 네트워크에서 최적의 전송을 수행하기 위해서는 네트워크 코딩기법 뿐 아니라 적용되는 채널 코딩 기법을 동시에 고려해서 설계해야 할 필요가 있다.

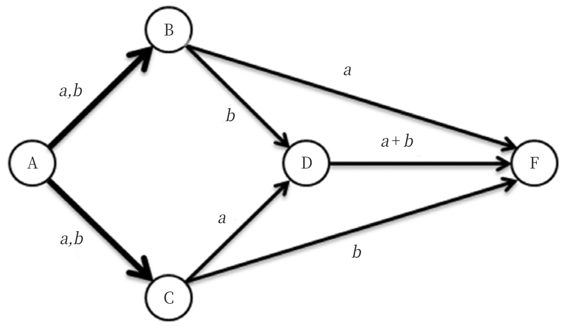

그림 4. 임의의 1개의 링크가 손상되더라도 2비트의 정보 전송이 가능한 네트워크

1. 링크 손상(link failure)

네트워크 상에서 정보를 전송하는 과정에서는 특정 링크가 파손되어 해당 링크를 통한 정보의 전송이 영구적으로 불가능해지는 경우도 발생한다. 이러한 링크의 파손이 발생했을 때의 가장 간단한 해결책은 손상된 링크를 제외하고 남은 네트워크에서 정보 전송의 경로를 재설정하는 것인데 이러한 방법은 매우 비효율적이다. 그러나 네트워크 코딩기법을 사용하는 경우 링크 손상이 발생하였을 때에 중간 노드에서의 동작을 변경하지 않고도 정상적인 정보 전송을 수행하는 것이 가능하다. 또한 소스와 리시버 사이의 min-cut의 값이 h인 네트워크에서 최대 t개의 링크가 임의로 손상되는 경우를 가정했을 때 적절한 네트워크 코드를 사용함으로써 전송률 h-t를 달성하는 최적의 전송이 가능하다[5],[6].

이와 관련된 예제로써 그림 4의 네트워크[7]를 살펴 보도록 하겠다. 그림 4에서 링크 AB, AC의 채널용량은 2비트이고 그 외의 링크들은 1비트의 채널용량을 가지며, min-cut의 값은 3비트가 된다. 이 네트워크에서는 임의로 최대 1개의 링크가 손상되고 링크가 손상된 경우에는 어떠한 링크가 손상되었는지 확인이 가능하다고 가정하겠다. 또한 손상된 링크를 통해서는 전부 0의 값이 전송된 것과 동일하게 동작하는 것으로 간주하겠다. 이러한 가정하에서 그림 4의 네트워크에서는 임의의 1개의 링크가 손상되더라도 항상 리시버 노드 F로 2비트의 정보를 전송하는 것이 가능하다는 것을 쉽게 확인할 수 있다.

지금부터는 min-cut의 값이 h인 네트워크에서 임의로 최대 t개의 링크가 손상되는 경우에 전송률 h-t를 달성할 수 있도록 하는 네트워크 코드의 설계방법에 대해서 알아볼 것이다. III장에서 살펴보았던 바와 같이 오류가 없는 네트워크에서 최대 전송률 h를 달성하기 위해서는 모든 리시버 노드에 대한 $h \times h$ 크기의 전송 행렬들이 역 행렬을 가지도록 설계하여야 한다. 만약 최대 t개의 링크가 손상되는 경우에 최대 전송률 h-t를 달성하고자 하는 경우를 고려해 보면 소스 노드들은 다음과 같은 변환 식에 따라 h-t개의 정보를 h개의 소스 심볼로 변환하여 전송을 수행하여야 한다.

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \cdots \\ \sigma_h \end{bmatrix} = \begin{bmatrix} \beta_{1,1} & \cdots & \beta_{1,h-t} \\ \cdots & \ddots & \cdots \\ \beta_{h,1} & \cdots & \beta_{h,h-t} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \cdots \\ \tau_{h-t} \end{bmatrix}$$

위 식에서 $\beta_{i,j} (i=1,2,\cdots,h, \ \ j=1,2,\cdots,h\text{-}t)$로 이루어진 $h \times (h\text{-}t)$ 크기의 선형 변환 행렬을 B라고 정의하겠다. 이와 같이 변환된 심볼들을 전송하는 네트워크에서의 실질적인 전송 행렬은 기존의 전송 행렬 A_j에 대하여 $A_j\ B$와 같이 표현된다. 또한 네트워크 상의 임의의 링크에 손상이 발생하면서 행렬 A_j의 일부 원소들의 값에도 변형이 발생한다. 예를 들어 그림 5의 네트워크의 경우를 생각해보면 링크 손상이 발생하기 이전의 리시버 노드 R_1과 R_2에 대한 전송 행렬은 다음과 같다.

$$A_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \alpha_3 & \alpha_4 \\ \alpha_1\alpha_5 & \alpha_5\alpha_2 + \alpha_6\alpha_3 & \alpha_6\alpha_4 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & 0 & 1 \\ \alpha_1 & \alpha_2 & 0 \\ \alpha_1\alpha_5 & \alpha_5\alpha_2 + \alpha_6\alpha_3 & \alpha_6\alpha_4 \end{bmatrix}$$

그러나 실제로 링크 손상의 효과를 반영한 전송 행렬은 기존의 전송 행렬과 비교하여 일부 원소들의 값이 변경된다. 그림 5의 네트워크에서 링크 DF가 손상된 경우를 가정하면 전송 행렬 A_1, A_2는 다음의 A'_1, A'_2와 같이 변형된다.

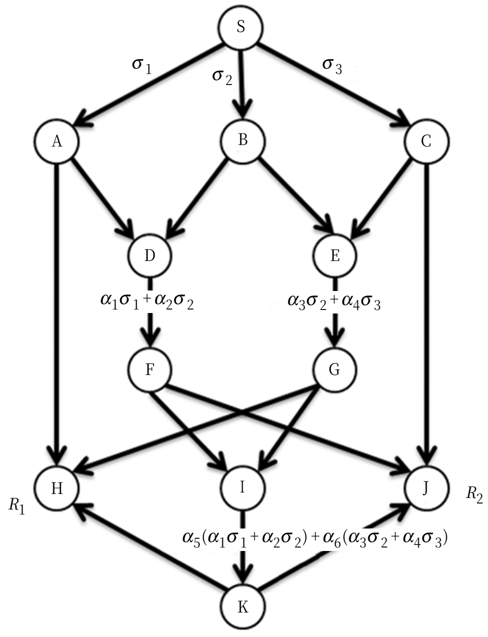

그림 5. 3개의 소스 심볼을 2개의 리시버로 전송하는 멀티캐스트 네트워크

$$A_1' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \alpha_3 & \alpha_4 \\ 0 & \alpha_6\alpha_3 & \alpha_6\alpha_4 \end{bmatrix},$$

$$A_2' = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & \alpha_6\alpha_3 & \alpha_6\alpha_4 \end{bmatrix}$$

따라서 선형 변환으로 생성된 소스 심볼과 링크의 손상으로 인한 효과를 모두 고려할 때의 리시버 노드 R_j로의 전송 행렬은 $A_j'B$가 된다. 이 때 행렬 A_j'는 링크 손상이 발생한 위치에 따라서 그 값이 변화하는 행렬이다. 이에 따라 임의로 최대 t개의 링크가 손상되는 경우에 최대 전송률 h-t를 달성하기 위해서는 모든 링크 손상 패턴에 대한 행렬 A_j'에 대해 행렬 $A_j'B$가 h-t의 계수(rank)를 갖도록 $\{\alpha_k\}$, $\{\beta_k\}$를 설계해야만 한다. 이러한 설계 과정은 확장된 LIF(Linear Information Flow) 알고리듬을 적용함으로써 수행할 수 있다.

2. 랜덤 오류 및 소실

네트워크 상의 각 링크에서는 랜덤하게 발생한 오류가 전송된 정보에 영향을 미치면서 변형된 정보가 수신단으로 전송되는 경우가 발생한다. 또한 링크의 대기열(queue)이 가득차거나 지연시간이 길어져서 일정 시간 내에 정보가 수신단까지 도착하지 못하는 등의 이유로 인해 전송된 정보가 소실되기도 한다. 본 절에서는 이러한 랜덤 오류 및 소실이 발생하는 상황에서의 네트워크 코딩 기법에 대해서 알아보도록 하겠다.

Min-cut의 값이 h인 유니캐스트 네트워크의 경우, 소스 노드들로부터 리시버 노드까지 중복되지 않는 h개의 경로가 존재한다. 이러한 네트워크의 경우 라우팅과 함께 기존의 채널 코딩기법을 사용하는 것만으로도 최적의 전송을 수행할 수 있다. 그러나 일반적인 형태의 멀티캐스트 네트워크에서는 서로 다른 리시버 노드로의 경로들 사이에 중복이 발생하기도 하며 이러한 경우에는 최적의 전송을 위해서 네트워크 코딩기법이 반드시 필요하다.

최대 t개의 임의의 링크에서 오류 또는 소실이 발생하는 네트워크에서 모든 리시버가 수신한 정보들로부터 소스에서 송신한 정보들을 정확히 복원하는 것이 가능하다면 이 때 사용되는 네트워크 코드를 t-오류 정정이 가능한 네트워크 코드라고 한다. 이러한 네트워크 코드는 기존의 채널 코딩기법과는 달리 소스 노드 뿐 아니라 중간 노드들에서도 정보의 가공 및 결합이 이루어진다. 이에 따라 네트워크 코딩기법을 사용하는 경우에는 하나의 링크에서 발생한 오류가 네트워크 상의 여러 링크로 빠르게 전파되는 오류 전파 문제가 발생하기도 한다.

네트워크 코딩기법을 사용하는 경우에도 기존의 채널 코딩에서 사용되던 것과 유사한 형태의 bound가 성립한다[8]. 소스 및 네트워크에서 사용되는 유한체의 크기를 q라 하고 길이가 k인 입력정보로부터 길이 h의 부호어(codeword)를 생성하여 전송하는 네트워크 코드

를 가정하자. 이에 따라 소스에서 생성되는 부호어의 종류는 q^k개가 된다. 또한 각 리시버 노드가 t개 이하의 링크에서 오류 또는 소실이 발생하는 경우에는 항상 정확한 복호를 수행할 수 있다고 가정하자. 위의 가정을 만족하는 네트워크 코드에서는 채널 코딩의 경우와 동일한 맥락으로 다음과 같은 bound가 성립하게 된다.

- *Hamming Bound*:

$$q^k \le \frac{q^h}{\sum_{k=0}^{t} \binom{h}{k} (q-1)^k}$$

- *Singleton Bound*:

$$q^k \le q^{h-2t}$$

네트워크 코드의 복호 과정에서도 채널 코드의 경우와 같이 최소 거리 복호(minimum distance decoding)를 사용하는 것이 가능하다[9]. 이러한 복호 기법을 적용하기 위해 먼저 네트워크에서의 Hamming 거리를 정의하도록 하겠다. 리시버 노드 R_j에서 수신하게 되는 정보들로 이루어지는 h의 길이를 갖는 벡터 y_j는 길이 h의 부호어 벡터 x와 길이 m의 오류 벡터 z에 관한 식으로 표현된다. 이 때 m은 전체 네트워크 상에 존재하는 링크의 수를 의미한다. 이러한 벡터들의 관계식은 III 장에서 언급한 네트워크 코드의 시스템 방정식에 의해 다음과 같은 형태로 나타난다.

$$y_j(x,z) = C_j(I-A)^{-1}(Bx+z) = A_j x + F_j z$$

여기서 $F_j = C_j(I-A)^{-1}$이며 이는 오류 벡터가 네트워크의 출력에 어떠한 영향을 미치는가를 나타내는 행렬이다. 오류 벡터 z의 Hamming weight가 $w_H(z)$일 때, z의 네트워크에서의 Hamming weight $w^j(z)$는 다음과 같이 정의된다.

$$w^j(z) = \min_{z':F_j z' = F_j z} w_H(z')$$

또한 2개의 출력 벡터 y_j와 y_j'에 대하여 네트워크 Hamming 거리 $d^j(y_j, y_j')$는 다음과 같은 식과 같이 정의된다.

$$d^j(y_j, y_j') = \min_{z:F_j z = y_j - y_j'} w_H(z)$$

네트워크 코드의 최소 거리 복호기법에서는 각 리시버 노드가 수신된 출력 벡터 y_j에 대하여 네트워크 Hamming 거리 $d^j(y_j, A_j x)$가 최솟값을 가지도록 하는 부호어 x를 찾아내어 복원하게 된다. 이러한 네트워크 코드의 최소 거리는 $d_{\min} = \min_j \min_{x \ne x'} d^j(A_j x, A_j x')$과 같이 정의되고 오류 및 소실이 발생하는 링크의 수가 $d_{\min}/2$보다 적은 경우에는 최소 거리 복호기를 이용하여 정확한 오류 정정을 수행하는 것이 가능하다.

V. 랜덤 네트워크 코딩

본 논문에서 지금까지 소개한 네트워크 코딩 기법은 네트워크의 구조에 따라 미리 설계된 네트워크 코드를 사용하여 정보를 전송하였다. 그러나 네트워크의 규모가 크고 복잡한 경우에는 네트워크 코딩을 설계하는 과정의 복잡도가 매우 높아지게 된다. 또한 실제로 사용되는 네트워크 중에서는 수시로 노드와 링크가 추가되거나 삭제되어 네트워크의 구조 자체가 실시간으로 변화하는 네트워크도 존재한다. 이러한 경우에 변화된 구조에 따라 매번 새로운 네트워크 코드를 설계하는 것은 매우 비효율적이다. 이러한 문제점을 개선하기 위해 제안된 기법이 랜덤 네트워크 코딩이다.

랜덤 네트워크 코딩을 수행하는 네트워크에서는 리시버 노드를 제외한 모든 노드들이 입력정보에 랜덤한 선형 결합을 수행하여 출력정보를 생성하게 된다. 이러한 랜덤한 선형 결합은 각 노드에서 독립적으로 이루어지며 각 노드는 출력정보와 함께 이 정보의 글로벌 코딩 벡터를 전송한다. 즉 각 노드에서는 입력정보에 랜덤한 선형 결합을 수행한 뒤 이로부터 입력 정보의 글로벌 코딩 벡터를 갱신하여 출력 정보와 함께 전송하게 되고 리시버 노드에서는 수신한 글로벌 코딩 벡터를 이용하여 소스 심볼을 복원하게 된다. 이러한 선형 랜덤 네트워크 코딩기법을 사용할 경우 사용되는 유한체의 크기가 증가함에 따라 1에 기하급수적으로 근접하는 확률로 멀티캐스트 네트워크의 최대 전송률을 달성할 수 있다[10].

- 정리 3: 소스로부터 전송되는 각 심볼들이 유한체 $\mathbb{F}_q$ 상의 원소이며 $\mathbb{F}_q$상에서 균등한 확률로 선택된 계수들을 이용하여 선형 랜덤 네트워크 코딩을 수행하는 네트워크를 생각해 보자. 또한 네트워크의 각 소스는 서로 독립적이거나 선형 상관특성을 가지는 경우를 가정하겠다. 네트워크 상의 리시버 노드의 개수가 $N(N<q)$개이고 링크의 개수를 m개라 할 때 이러한 랜덤 네트워크 코드는 적어도 $(1-N/q)^m$의 확률로 멀티캐스트 네트워크의 최대 전송률을 달성할 수 있다.

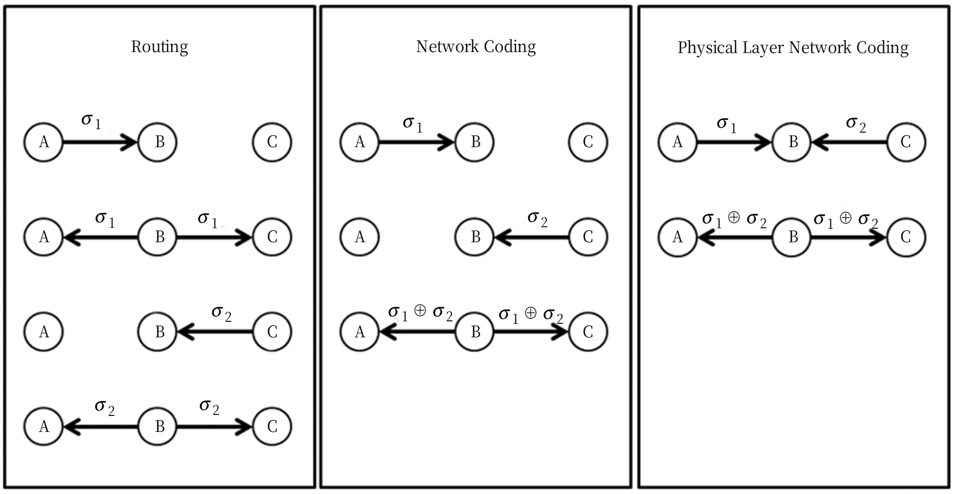

그림 6. 무선 중계 네트워크에서 물리 계층 네트워크 코딩기법과 여타 기법간의 비교

위의 정리는 네트워크 코드에서 사용되는 로컬 코딩 벡터의 계수들 가운데 일부만 랜덤하게 선택하고 나머지 계수는 고정된 값을 사용하는 경우에도 성립한다. 단 멀티캐스트 네트워크에서 이러한 고정된 계수들을 포함하는 최대 전송률을 달성하는 해가 존재하는 경우에만 성립하게 된다. 이 때 정리 3에서 m의 값은 전체 링크의 수가 아닌 랜덤하게 생성된 계수와 연관된 링크의 수를 나타낸다. 정리 3에서는 네트워크 내의 각 소스가 독립적이거나 선형 상관특성을 가지는 경우만을 가정하였지만 임의의 상관특성을 가지는 소스의 경우에도 최대 전송률의 멀티캐스트 전송이 성공할 확률의 하한이 유도되었다[10]. 또한 네트워크 상의 링크에서 오류가 발생하거나 정보가 소실되는 상황에서의 랜덤 네트워크 코딩의 설계방법도 [11]에서 제시되었다.

VI. 무선 네트워크 코딩

지금까지 본 논문에서는 주로 유선 네트워크에서 네트워크 코딩을 수행함으로써 어떠한 이득을 얻을 수 있는지에 대하여 알아보았고 또한 네트워크 코딩을 이용하여 멀티캐스트 네트워크에서 전송가능한 최대 전송률을 달성할 수 있다는 것을 확인하였다. 그러나 유선 네트워크의 경우와는 달리 무선 네트워크에서 네트워크 코딩을 수행하는 경우에는 라우팅에 비하여 얼마만큼의 성능 향상을 얻을 수 있는지 그리고 어떠한 네트워크 코드를 사용하는 것이 최적인지에 대해 아직 잘 알려져 있지 않다. 그러나 I장에서도 언급한 바와 같이 무선 네트워크에서 네트워크 코딩기법을 적용함으로써 무선 자원 사용 측면에서 이득을 얻을 수 있으며 이외에도 시간 지연 측면이나 수시로 상태가 변화하는 네트워크에서의 적응성 측면 등에서도 이점을 가진다는 것이 확인되었다.

무선 네트워크에서의 정보 전송은 주로 브로드캐스팅기법을 이용하여 이루어진다. 이러한 브로드캐스팅 기법을 사용할 경우 전송범위내에 있는 모든 노드로 정보가 전송되기 때문에 여러개의 노드가 동시에 전송을 수행하는 경우에는 정보를 수신하는 노드에서 여러 정보가 결합된 신호를 수신하는 경우도 발생할 수 있다. 수신 노드에서 이 중 하나의 정보만을 복원하고자 하는 경우에는 다른 노드에서 보낸 정보들이 간섭(interference)으로 작용하여 복원을 방해하는 요소로서 작용한다. 일반적으로 이러한 간섭현상은 무선 네트워크에서의 정보 전송에서 발생하는 가장 큰 문제점 중 하나이다. 그러나 오히려 이러한 신호간의 간섭을 이용하여 보다 효율적인 전송을 수행할 수 있는 기법이 있는데 그것이 바로 물리 계층 네트워크 코딩[12]이다.

물리 계층 네트워크 코딩을 이용하는 경우 중간 노드에서는 여러 정보가 물리 계층에서 결합된 형태의 신호를 수신하는데 이를 수신한 중간 노드는 수신한 정보를 복호하지 않고 그대로 다른 노드로 전달해 주게 된다. 즉 일반적인 네트워크 코딩기법에서는 노드에서 각 정보들을 하나씩 수신한 뒤 로컬 코딩 벡터의 계수에 따라서 결합을 수행하는데 이와 달리 물리 계층 네트워크 코딩에서는 동시 전송으로 인해 발생하는 간섭을 오히려 정보를 결합하는 방식으로 활용하는 것이다.

물리 계층 네트워크 코딩을 사용하는 네트워크의 예시는 그림 6에 나타나 있는데, 이는 그림 2의 예시를 확장한 것이다. 그림 2에서와 마찬가지로 모든 전송을 마치는 데까지 라우팅만을 사용하는 경우에는 4번의 단위 시간을 네트워크 코딩기법을 사용하는 경우에는 3번의 단위 시간을 필요로 한다. 그러나 물리 계층 네트워크

코딩을 사용하는 경우 노드 B에서는 노드 A와 C로부터 수신하는 정보를 2번의 단위 시간에 걸쳐 수신할 필요 없이 동시에 수신하는 것이 가능하다. 이와 같이 물리 계층 네트워크 코딩을 이용하여 단 2번의 단위 시간에 모든 전송을 완료하는 것이 가능하다.

VII. 결 론

본 논문에서는 네트워크 상의 각 노드들이 수신된 정보를 결합하여 출력정보를 생성하는 네트워크 코딩 기법에 대해 소개하였다. 네트워크 코딩을 이용할 경우 라우팅 기법만으로는 달성할 수 없는 멀티캐스트 네트워크에서의 최대 전송률을 달성할 수 있다는 것을 확인하였고 이를 가능하도록 하는 선형 네트워크 코드의 설계 방법에 대해서도 알아보았다.

또한 다양한 형태의 오류가 발생하는 네트워크에서 어떠한 방식으로 네트워크 코딩을 수행해야 하는지에 대해서도 살펴보았고 랜덤 네트워크 코딩 및 무선 네트워크 상에서의 네트워크 코딩기법에 대해서도 간략하게 소개하였다.

본 논문에서 소개한 내용 이외에도 네트워크 코딩 기법과 관련하여 지금까지 수많은 연구가 이루어졌고 현재도 활발한 연구가 진행중인 상황이다. 또한 빠른 속도로 진행되는 통신 기술의 발달과 더불어 머지 않아 다양한 통신 시스템에 네트워크 코딩기법이 널리 사용될 것으로 예상된다.

[참고문헌]

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, Vol. 46, No. 4, Jul. 2000, pp. 1204-1216.

[2] L. R. Ford Jr. and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, Vol. 8, 1956, pp. 399-404.

[3] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, Vol. 49, No. 2, Feb. 2003, pp. 371-381.

[4] R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang, "Network coding theory: A tutorial," *Foundation and Trends in Communication and Information Theory*, Vol. 2, 2006, pp. 241-381.

[5] T. Ho, M. M?dard, and R. Koetter, "An information-theoretic view of network management," *IEEE Trans. Inf. Theory*, Vol. 51, No. 4, Apr. 2005, pp. 1295-1312.

[6] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, "Resilient network coding in the presence of Byzantine adversaries," *INFOCOM*, May 2007, pp. 616-624.

[7] S. E. Rouayheb, A. Sprintson, and C. N. Georghiades, "Simple network codes for instantaneous recovery from edge failures in unicast connections," *ITA ̀06*, Feb. 2006.

[8] R. W. Yeung and N. Cai, "Network error correction, I: Basic concepts and upper bounds," *Communications in Information and Systems*, Vol. 6, No .1, 2006, pp. 19-35.

[9] Z. Zhang, "Linear network error correction codes in packet networks," *IEEE Trans. Inf. Theory*, Vol. 54, No. 1, Jan. 2008, pp. 209-218.

[10] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, Vol. 52, No. 10, Oct. 2006, pp. 4413-4430.

[11] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Inf. Theory*, Vol. 54, No. 8, Aug. 2008, pp. 3579-3591.

[12] S. Zhang, S. -C. Liew, and P. P. Lam, "Physical-layer network coding," *ACM MobiCom*, Sep. 2006, pp. 24-29.

[13] C. Fragouli and E. Soljanin, *Network Coding Fundamentals*, Now Publishers Inc., 2007.

[14] C. Fragouli and E. Soljanin, *Network Coding Applications*, Now Publishers Inc., 2007.

홍 석 범
(Seokbeom Hong)

2002. 3 ~ 2007. 2: 서울대학교 전기공학부 학사
2007. 3 ~ 2009. 2: 서울대학교 전기컴퓨터공학부 석사
2009. 3 ~ 현재: 서울대학교 전기컴퓨터공학부 박사과정
관심분야: 네트워크 코딩, 압축 센싱, 오류정정부호
E-mail: fousbyus@ccl.snu.ac.kr
Tel: +82-2-880-8437
Fax: +82-2-880-8222

노 종 선
(Jong-Seon No)

1981. 2: 서울대학교 전자공학과 학사
1984. 2: 서울대학교 전자공학과 석사
1988. 2: University of Southern California
전기공학과 박사
1988. 2 ~ 1990. 7: Hughes Network Systems,
Senior MTS
1990. 8: 건국대학교 전자공학과 부교수
1999. 7 ~ 현재: 서울대학교 전기컴퓨터공학부 교수
관심분야: 오류정정부호, 의사불규칙수열, 암호학,
시공간부호, LDPC부호, 무선통신시스템
E-mail: jsno@snu.ac.kr
Tel: +82-2-880-1809
Fax: +82-2-880-8185