Journal of KIIT. Vol. 18, No. 3, pp. 139-147, Mar. 31, 2020. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2020.18.3.139

사물인터넷 환경에서 공격에 대응 가능한 네트워크 코딩 연구

이 용*

Network Coding Method to Withstand Attack in IoT Environment

Yong Lee*

요 약

네트워크 코딩은 목적지에서 디코드할 수 있도록 중간 릴레이 노드들이 주변 노드들로부터 수집한 데이터를 결합하는 기술로서 네트워크 사용량이 증가하는 사물인터넷 환경에서 네트워크의 처리율을 향상시키기 위한 방안으로 연구되는 추세이다. 사물인터넷 환경에서 네트워크 코딩은 릴레이 노드가 주변 디바이스로 부터 온 패킷들을 혼합하여 전송하므로 네트워크 토폴로지가 신뢰할 수 있는 중간노드들로 이루어진 경우에만 제대로 동작할 수 있다. 그러나 노드들 사이에서 전자 서명이나 암호화 등의 보안 기술을 적용하더라도 악의적인 중간 릴레이 노드가 정상적으로 네트워크 구성에 참여하면서 유효하게 보이는 서명이나 암호를 사용하여 릴레이하는 패킷을 악의적으로 위변조하는 내부적 공격을 수행할 경우 이를 알아차리기 어렵게 된다. 따라서 본 논문에서는 네트워크 코딩으로 전송되는 패킷에 공격이 존재하는 경우에도 목적지 노드가 정당한 메시지를 복구해 낼 수 있는지에 대한 조건과 방법을 제시하고자 한다.

Abstract

Network coding is a technique that combines data collected from neighboring nodes by relay nodes so that they can be decoded at the destination, and has been researched as a method for improving network throughput in an IoT environment in which network usage is rapidly increasing. In IoT environment, network coding works well only when the network topology consists of reliable nodes because the relay node transmits a mixture of packets from neighboring nodes. However, even if security algorithms such as digital signature or encryption are applied, when a malicious node legitimately participates in the network configuration and performs an internal attack forging a relaying packet with a signature or cipher that appears to be valid, it becomes difficult to notice. In this paper, we propose the condition and method of whether the destination node can reconstruct the correct message even if there is an attack in the packet transmitted by network coding.

Keywords

network coding, IoT, reconstruction rate, redundancy, overlapped packet, error detection

* 프리랜서
- ORCID ID: https://orcid.org/0000-0002-8208-7335

· Received: Nov. 21, 2019, Revised: Jan. 08, 2020, Accepted: Jan. 11, 2020
· Corresponding Author: Yong Lee
Email: yleehyun@gmail.com

Ⅰ. 서 론

가상현실 게임(Virtual reality game)이나 u-헬스케어(u-health care), 홈오토메이션(Home automation), C-ITS(Cooperative Intelligent Transport Systems)와 같은 다양한 응용 서비스들이 동작하는 사물인터넷(IoT, Internet of Things) 환경은 디바이스 제작자, 인터넷 서비스제공자, 응용 서비스 개발자들에게 많은 관심을 받고 있다[1][2]. 사물인터넷 환경에서는 웨어러블 기기나 센서기기들을 통한 네트워크 사용량이 증가함에 따라 네트워크의 처리율을 향상시키기 위한 방안으로 네트워크 코딩(Network coding)을 적용하는 추세이다[2]. 네트워크 코딩은 목적지에서 디코드할 수 있도록 중간 릴레이 노드들이 여러 디바이스 소스로부터 받은 서로 다른 데이터들을 결합하는 기술이다[3][4]. 네트워크 신뢰도와 에러에 대한 유연함을 가지기 위해 소스 노드에서 싱크까지 여러 경로를 가지는 네트워크에서 주로 사용될 수 있으므로 센서기기나 웨어러블 기기들이 감지, 측정한 많은 양의 정보를 전송해야 하는 사물 인터넷 환경에 적합하다. 네트워크 코딩 구조에서 릴레이 노드는 패킷을 전송하기 전에 다른 소스로부터 온 패킷들을 혼합하여 각 전송에서 보내지는 정보의 내용을 양적으로 증가시키는 방법을 적용할 수 있다. 이런 방법은 전송되는 각 패킷이 가지는 정보의 양이 증가하며 일부 패킷이 손실되더라도 살아남은 패킷들로 인하여 정보의 복구가 가능하므로 통신 에러나 하드웨어 또는 릴레이의 오류에도 네트워크가 유연성을 가질 수 있도록 하는 장점이 있다. 이러한 특징은 센서기기들의 증가에 따라서 네트워크 용량의 증가와 네트워크 안정성이 필수적인 사물인터넷 환경에 네트워크 코딩의 적용을 적절하게 한다.

네트워크 코딩의 스킴은 네트워크 토폴로지가 신뢰할 수 있는 중간 노드들로 이루어진 경우에만 제대로 동작할 수 있다. 악의적인 노드가 참여하게 될 경우 중간 릴레이 노드로서 마음대로 정보를 조작하여 혼합할 수 있게 됨에 따라 악의적인 노드에 의한 정보의 위변조 위험 또한 증가하게 된다. 네트워크 구성에 있어서 신뢰성 있는 노드의 참여와 그 구별은 지속적인 과제이며 특히 사물인터넷의 경우 디바이스의 참여가 자유롭다는 측면에서 네트워크 코딩을 적용하는 경우에도 악의적인 노드로 인한 위험을 피할 수가 없게 된다. 네트워크 구성에서 악의적인 노드는 외부의 침입에 의한 경우도 존재하지만 네트워크 구성에 정당하게 참여한 내부 노드에 의한 위협도 발생한다. 내부 노드에 의한 위협의 경우 이미 신뢰받은 노드에 의한 공격이므로 그 진위를 파악하거나 발원지를 발견하기 더 어려운 문제가 있다. 전자 서명이나 암호화 등의 보안 기술을 적용하더라도 악의적인 중간 릴레이 노드가 정상적으로 네트워크 구성에 참여하면서 악의적으로 릴레이하는 패킷을 위변조하고 유효하게 보이는 서명이나 암호를 사용하는 내부적 공격을 수행할 경우 이를 알아차리기 어렵게 된다. 이런 경우 목적지 노드는 패킷의 공격 여부를 알지 못하고 수신된 패킷에 대하여 메시지 복구를 실행하게 된다. 이렇게 복구된 메시지는 정당한 패킷만으로 수행된 경우와 내부 공격에 의해 정당하게 보이는 공격 패킷에 포함된 경우 각각 다른 메시지로 나타나게 된다. 그러나 메시지는 모두 정당한 서명이나 암호화의 형태를 가지기 때문에 목적지 노드는 정당한 메시지를 구별해내기가 어려운 문제가 있다. 따라서 본 논문에서는 이런 경우에 목적지 노드가 정당한 메시지를 복구해 낼 수 있는지에 대한 기준과 방법을 제시하고자 한다.

Ⅱ. 관련 연구

사물인터넷이나 D2D(Device-to-Device) 통신 같은 환경은 멀티홉 네트워크 구조로 분류되고 급격히 증가하는 방대한 양의 데이터를 광범위한 사물인터넷을 통하여 효율적으로 신속하게 전달하는 것이 중요하다. 이런 환경에서는 네트워크 처리율이나 효율성을 향상시키는 것이 중요한 이슈이고 이를 위해 네트워크 코딩을 적용하는 연구가 활발히 진행되고 있다[2][5][6]. 이 장에서는 이와 관련된 연구를 살펴보고자 한다.

참고문헌 [7]에서는 사물인터넷 환경에서 센서 데이터를 블록체인 기술을 적용하여 전송하기 위하여 네트워크 코딩을 적용하는 방법을 제안하였다. 블록체인 기술이 많은 계산량과 블록에 대하여 합의를

이루는데 긴 지연 시간의 오버헤드가 동반되는데 비하여 네트워크 코딩 기술은 데이터 패킷을 조합하여 전송하므로 처리율을 높이고 블록체인의 문제점을 해결하는 데 도움이 되는 것을 보여준다.

디바이스가 무선 자원을 사용하는 환경에서 에너지 효율을 높이기 위해 네트워크 코딩을 적용하는 방안에 대한 연구도 활발하다[6][8][9]. 참고문헌 [6]에서는 무선 통신 채널 환경에서 통신 신뢰성을 높이고 전송 지연시간을 낮추기 위하여 업링크(Uplink)와 다운링크(Downlink)로 동시에 패킷을 전송할 수 있도록 네트워크 코딩을 적용하는 프로토콜을 제안하였다. 참고문헌 [8]에서는 무선 센서기기들의 에너지 소모를 줄이는데 관심을 가지고 정보를 전달할 노드가 이웃 노드들 중에서 데이터를 우선적으로 전달할 노드를 효율적으로 선택하기 위한 알고리즘을 제안하였다. 참고문헌 [9]에서는 사물인터넷 환경에서 효율적으로 많은 양의 데이터를 처리하고 데이터를 검색하기 위해 데이터의 저장에 네트워크 코딩을 적용하였다.

사물인터넷에서 다량의 데이터를 전송하기 위해 네트워크 코딩을 적용하여 처리율과 전송 효율성이 높아지는 것을 보여주는 연구가 많이 진행되고 있다[2][5]. 참고문헌 [10][11]에서는 사물인터넷이 여러 가지 다른 특성을 가지는 이종의 디바이스들로 구성된다는 점에 주목하고, 이런 환경에서 처리율을 높일 수 있는 네트워크 코딩 기술 적용방안을 제안하였다. 참고문헌 [12]에서는 메시지 전송 지연 시간을 단축시키는 방식으로 네트워크 코딩을 적용하고 분석하였다.

참고문헌 [13]에서는 전송되는 패킷에 오류검출(Error detection)과 오류수정(Error correction) 기술을 적용한 후 이를 인코딩하여 전송하는 방안을 제안하였다. 여기서는 중간노드들이 전송하는 패킷에 대하여 오류수정을 수행할 수 있도록 하여 처리율이 높아짐을 보여준다.

이와 같이 사물인터넷 환경에서 전송 효율을 높이고 네트워크의 신뢰성을 제공하기 위하여 네트워크 코딩을 적용하는 많은 연구가 수행되고 있다. 이 논문에서는 릴레이 구조로 이루어지는 사물 인터넷 환경에서 네트워크 코딩을 적용할 경우 중간노드에 의한 공격에 대비하기 위한 알고리즘을 제안하고자 한다.

III. 사물인터넷 환경에서 공격에 대응가능한 네트워크 코딩 연구

3.1 시스템 모델

이 논문에서는 사물인터넷 환경에서 소스 노드와 목적지 노드 간의 단대단 오류 제어에 초점을 맞추고자 한다. 따라서 사물인터넷에 연결될 게이트웨이와 같은 중간노드들은 소스 노드와 목적지의 단대단 통신에 영향을 끼치지 않고 네트워크 코딩된 패킷의 생성에만 관여한다. 따라서 랜덤 네트워크 코딩(Random network coding)의 일반적인 방법으로 수신 패킷에 대해 랜덤 선형 조합(Random linear combination)을 적용하여 송신 패킷을 생성하여 전달한다. 이 논문에서는 전송 에러는 고려하지 않고 공격자의 공격에 의한 에러만을 고려한다. 네트워크 코딩에서는 악의적인 공격으로 인하여 인코딩된 패킷(즉, 조합의 형태)에 오류가 있다면 목적지 노드는 이를 제대로 복구할 수가 없게 된다. 우리는 목적지 노드가 인코딩된 패킷에서 오류가 있음을 발견하고 이를 원래대로 복구할 수 있는 방법을 연구하여 어느 정도까지의 공격수준에 대응할 수 있는지를 분석하고자 한다.

3.1.1 기호

이 장에서는 다음과 같은 표기를 사용한다.

- b : 소스 노드의 데이터 메시지가 전송을 위해 쪼개진 패킷 수
- n : 소스 노드가 b개의 패킷에 코드화 벡터로 선형방정식을 적용하여 나온 인코딩된 패킷 수
- m : 원본 패킷에 선형방정식을 적용하여 계산된 인코딩된 패킷에 포함된 리던던시의 수, $n-b$의 값을 가짐
- C_e : n개의 인코딩된 패킷 중에서 공격으로 인해 위변조된 패킷
- e : n개의 인코딩된 패킷 중에서 목적지에서 수신한 공격으로 인해 위변조된 패킷, 즉 C_e의 수

- r : n개의 인코딩된 패킷 중에서 목적지에서 수신한 정상 패킷 수
- *오버랩(overlapped)* 패킷 : 두 개의 서로 다른 디코딩 복구결과를 보이는 그룹에 공통으로 포함된 인코딩된 패킷들
- N_o : 목적지 노드가 수신한 n개의 인코딩된 패킷 중에서 오버랩 패킷 수
- $(P_1,P_2,...P_b)$: 원본 패킷들
- $(C_1,C_2,...C_n)$: 목적지 노드가 수신한 n개의 인코딩된 패킷들, $n=b+m$

3.1.2 소스의 메시지 전송

네트워크 코딩을 적용한 메시지 구성은 [4]의 정보 분산 스킴(Information dispersal scheme)을 기반으로 한다. 네트워크 코딩을 적용하여 패킷을 생성하는 소스 노드는 메시지 데이터를 b개의 패킷들로 자른다. 이 b개의 패킷은 아직 인코딩이 적용되지 않은 패킷이다[3][4][14]. 이 패킷들을 $x_i, i=1,2,...,b$로 정의하고 소스 노드는 이 패킷들에 선형방정식(Linear combination)을 적용하여 인코딩된 패킷 $c_j,\ j=1,2,...,b$으로 변환한 후 이를 네트워크로 전송한다[14]. 즉 c_j는 다음과 같이 나타낼 수 있다.

$$c_j=\sum_{i=1}^{b}a_{ji}x_i \tag{1}$$

여기서 a_{ji}들은 무작위로 선택된 계수들로서 갈로아 필드(Galois field), $GF(2^q)$에 대한 덧셈과 곱셈 연산으로 구성된다. 인코딩 벡터, $\overrightarrow{a_j}=(a_{j1},a_{j2},...,a_{jb})$는 c_j 패킷의 헤더에 포함되어 전송되며 헤더는 패킷이 복구될 때 사용된다[3][4].

3.1.3 목적지의 복구

목적지 노드는 인코딩된 패킷을 받았을 때, 코딩 계수를 사용하여 이를 복구한다. 각 인코딩 패킷은 b개의 보통 패킷들의 선형방정식으로 표현되므로 목적지 노드가 패킷을 수신하면 선형방정식을 이용한 디코딩과정을 수행하여 원본 메시지를 복구할 수 있게 된다. 이때 복구에 사용된 b개의 인코딩된 패킷은 소스 노드가 생성한 n개의 패킷의 부분집합이고 선형적으로 독립이며[14][15], 다음과 같이 나타낼 수 있다.

$$\begin{pmatrix}a_{11}\ a_{12}\ ...\ a_{1b}\\ a_{21}\ a_{22}\ ...\ a_{2b}\\ \vdots\ \ \vdots\ \ \ddots\ \vdots\\ a_{n1}\ a_{n2}\ ...\ a_{nb}\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ \vdots\\ x_b\end{pmatrix}=\begin{pmatrix}c_1\\ c_2\\ \vdots\\ c_n\end{pmatrix} \tag{2}$$

윗 식에서 c_j는 노드가 수신한 인코딩 패킷이며 대응하는 인코딩 벡터는 $\overrightarrow{a_j}=(a_{j1},a_{j2},...,a_{jb})$이다[3][4].

3.2 오버랩 패킷을 이용하여 공격에 대응 가능한 네트워크 코딩 복구 방안

3.2.1 오버랩 패킷

목적지 노드가 수신한 패킷 중에서 임의의 b개의 패킷을 대상으로 복구 연산을 수행하여 나온 결과 중에서 동일한 결과값을 나타내는 복구과정이 두 가지 이상인 경우 복구과정들의 모임을 그룹이라고 한다. 수신한 패킷에 오류가 없는 경우 모든 복구결과는 단일 결과값을 보이게 되고 모든 복구과정은 하나의 그룹을 구성한다. 복구결과가 두 가지인 경우 두 개의 그룹이 구성되고 한 그룹은 옳은 패킷들로 구성되고 다른 그룹은 공격받은 패킷을 포함하게 된다. 어떤 패킷들은 양쪽 그룹에 모두 포함되는데, 이 패킷들을 *오버랩* 패킷이라고 한다. 본 연구내용 설명을 위해 다음의 시나리오를 예로 든다.

○ 시나리오 : $GF=\{0,1,2,3\}$에서 모듈로 4 연산을 수행하는 예를 가정하자. 여기서 소스 노드는 메시지를 P_1, P_2의 두 개의 패킷으로 자르고 식(3)과 같이 선형방정식을 수행하여 C_1, C_2, C_3, C_4의 조합을 생성하여 전송한다. 이 메시지들은 전송 중에 중간 노드의 공격에 의해 C_4가 C_e로 변조된다고 가정하고, 변조된 C_e가 정당한 서명과 암호값을 가지고 있는 경우 목적지 노드에 도착하여 복구과정이 수행될 때까지 변조 여부를 알 수가 없게 된다. 네트워크는 전송으로 인한 오류는 없다고 고려하여 목적지 노드는 소스 노드가 전송한 모든 패킷을 수신

할 수 있다고 가정한다. 목적지 노드는 식(3)과 같은 인코딩 패킷들을 수신하였다고 하자.

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \\ a_{41} & a_{42} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \end{pmatrix} = \begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{pmatrix} \Rightarrow \begin{pmatrix} P_1 + 2P_2 = 0 : C_1 \\ 3P_1 + 2P_2 = 0 : C_2 \\ 2P_1 + P_2 = 1 : C_3 \\ 2P_1^e + 3P_2^e = 1 : C_e \end{pmatrix} \quad (3)$$

목적지 노드가 위의 수신한 인코딩된 패킷에 대하여 복구과정을 수행할 경우 각 2개 쌍의 방정식들로부터 다음의 6개의 복구결과를 얻을 수 있다.

- (C_1, C_2), (C_2, C_3), $(C_1, C_3) \Rightarrow P_1 = 2,\ P_2 = 1$
- (C_1, C_e), $(C_2, C_e) \Rightarrow P_1 = 2,\ P_2 = 3$
- $(C_3, C_e) \Rightarrow P_1 = \frac{1}{2},\ P_2 = 0$

이 결과에서 다음과 같은 2개의 그룹을 보여준다. 하나는 (C_1, C_2, C_3)으로 (C_1, C_2), (C_1, C_3), (C_2, C_3)으로부터 $P_1 = 2, P_2 = 1$의 결과를 생성하고, 다른 그룹은 (C_1, C_2, C_e)로 (C_1, C_e), (C_2, C_e)로부터 $P_1 = 2, P_2 = 3$의 결과를 생성한다. 공격 패킷 C_e가 포함된 세 경우를 제외한 나머지 세 경우는 모두 정상 패킷을 사용하여 수행된 복구과정이며 이들 세 가지 경우의 결과는 모두 동일하여 하나의 그룹을 형성한다. 공격 패킷 C_e를 포함하는 경우는 서로 다른 두 가지 결과를 보여준다. 이 예에서 C_1과 C_2는 양쪽 그룹에 모두 포함되므로 오버랩 패킷이 되고 $N_o = 2$가 된다.

결과에 다수의 법칙을 적용할 경우 $P_1 = 2$, $P_2 = 1$의 정상적인 복구결과를 선택하게 된다. 이 예에서 알 수 있는 것처럼 목적지 노드가 수신한 패킷 4개 중에서 원본 메시지 복구를 위해서는 2개의 메시지가 필요로 하므로 6개의 복구결과를 얻을 수 있으며 이로 인하여 전송 중에 공격받은 패킷이 존재하더라도 정상적인 패킷들만을 사용하여 복구된 메시지가 존재하게 된다. 정상적인 복구 메시지가 공격받은 복구 메시지의 수보다 많다면 목적지 노드는 다수의 법칙을 적용하여 이를 찾아낼 수 있게 된다. 그러나 공격받은 패킷의 수가 많거나 공격받은 패킷들을 포함하는 복구결과들이 더 많은 공통값을 보여주게 된다면 목적지 노드가 다수의 법칙을 적용하여 정상적인 복구결과를 찾는데 방해가 될 것이다.

3.2.2 제안하는 방안

소스 노드와 목적지 노드 간에 네트워크 코딩을 이용하여 통신이 이루어지는 경우 목적지 노드는 수신한 패킷에 선형방정식을 적용하여 디코딩 과정을 수행한다. 목적지 노드가 수신한 패킷이 모두 에러나 공격이 없는 정상적인 패킷이라면 이 패킷들을 이용하여 수행한 복구결과는 모두 동일한 결과를 보이게 되고 이는 정상적인 원본 패킷이 된다. 만약 수신한 패킷에 공격 패킷이 존재하는 경우 목적지 노드가 수행한 복구결과는 단일한 원본 패킷을 생성하지 못하고 다양한 결과를 보여주게 된다.

이 논문에서는 소스 노드가 전송한 메시지가 악의적인 중간노드의 공격을 받아 "정상적인" 패킷으로 보이도록 위 변조된 경우, 목적지 노드에서 수신한 패킷에 공격 패킷이 포함되더라도 정상적인 원본 메시지를 복구할 수 있는 경우를 보여준다. 목적지 노드에서는 공격 패킷으로 인해 발생하는 다양한 복구결과 중에서 정상적인 원본 메시지를 찾아낼 수 있게 된다. 이때 원본 메시지에 적용되는 인코딩 벡터의 수에 따라 확장되는 인코딩 패킷의 수에 따라 목적지 노드가 정상적인 원본 메시지를 찾을 확률이 다르게 나타나며 본 논문에서는 이를 분석하고자 한다. 네트워크는 전송으로 인한 오류는 없다고 고려하여 목적지 노드는 소스 노드가 전송한 모든 패킷을 수신할 수 있다고 가정한다.

목적지 노드에서 정상적인 결과값의 복구에 성공하기 위하여 수신하는 패킷에 포함된 공격 패킷의 수와 *오버랩* 패킷의 수의 관계를 살펴보면 다음과 같다.

- $e < m$일 때 {수신된 정상 패킷의 수 $> b$}가 되고, 정상적인 복구결과가 존재하게 된다.
- $e > m$일 때 {수신된 정상 패킷의 수 $< b$}가 되고 정상적인 복구결과는 존재하지 않는다.
- $e = m$인 경우, {수신된 정상 패킷의 수 $= b$}가 되고 정상적인 복구결과는 단 한 개 존재하게 된다. 이 경우 목적지 노드는 다수의 법칙을 적용할 수가 없다.

따라서 정상적인 복구결과가 존재하는 $e<m$의 경우에 대하여 *오버랩* 패킷의 수의 관계를 살펴본다. 모든 $(e<m)$에 대하여 $n-e>b$가 되고, 공격 패킷에 대하여 $e+N_o=b$인 경우 복구결과가 하나만 있으므로 그룹을 구성할 수 없고 다수의 법칙에 선택될 수 없다. $e+N_o>b$인 경우는 다음과 같이 3가지 경우를 고려할 수 있다.

i) $e+N_o<n-e$일 때 : $N_o<n-2e$, $r\geq e+1$이 되고, 수신된 패킷의 구성이 그림 1과 같게 된다.

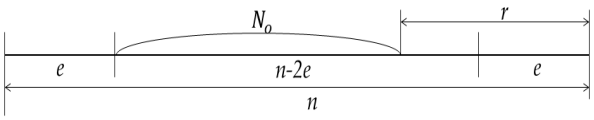

그림 1. $e+N_o<n-e$일 때 목적지에서 수신한 패킷의 구성

Fig. 1. Composition of packets received at the destination when $e+N_o<n-e$

따라서 공격받은 패킷을 포함하는 복구결과의 수를 R_A, 정상적인 패킷들로 수행된 복구결과의 수를 R_N이라고 하면 다음과 같은 관계가 성립한다.

$$R_A=\binom{e+N_o}{b}<\binom{n-e}{b}=R_N \tag{4}$$

이 경우 정상적인 복구 결과값을 찾을 수 있게 되고 $N_o<n-2e$이 된다.

ii) $e+N_o=n-e$일 때 : 두 가지 경우로 나누어 볼 수 있다. 먼저 $N_o\geq b$인 경우는,

$$R_A=\binom{e+N_o}{b}-\binom{N_o}{b}<\binom{n-e}{b}=\binom{e+N_o}{b}=R_N \tag{5}$$

가 되므로 정상적인 결과값을 찾을 수 있게 되고 $N_o=n-2e$이 된다. 다음으로 $b-e<N_o<b$일 때

$$R_A=\binom{e+N_o}{b}=\binom{n-e}{b}=R_N \tag{6}$$

가 되어 두 그룹의 구성 수가 같아져서 다수의 법칙을 적용할 수 없게 될 수 있으므로 정상적인 결과값을 찾는 것을 보장할 수 없게 된다.

iii) $e+N_o>n-e$일 때 : $N_o<b$일 때는 $e+N_o>n-e$이므로 정상적인 복구결과를 찾는 것을 보장할 수 없게 되므로 $N_o\geq b$ 일 때만 고려한다. 먼저 $e+N_o=n-e+1$인 경우를 살펴보면, $e=1$일 때,

$$\begin{aligned}\frac{R_A}{R_N}&=\frac{\binom{e+N_o}{b}-\binom{N_o}{b}}{\binom{n-e}{b}}=\frac{\binom{n}{b}-\binom{n-1}{b}}{\binom{n-1}{b}}\\&=\frac{1}{2}\frac{n}{(n-b)}=\frac{1}{2}\frac{(b+m)}{m}\end{aligned} \tag{7}$$

이 된다. 식 (7)으로부터

$$\frac{R_A}{R_N}=\begin{cases}\frac{1}{2}(\frac{b}{m}+1)<1, & m>b\text{일때}\\ \frac{1}{2}(\frac{b}{m}+1)\geq 1, & m\leq b\text{일때}\end{cases} \tag{8}$$

을 구할 수 있다. 식 (8)로부터 $m>b$일 때 $e=1$을 적용하면

$$R_A=\binom{e+N_o}{b}-\binom{N_o}{b}<\binom{n-e}{b}=R_N \tag{9}$$

이 되어 정상적인 결과값을 찾을 수 있게 되고 이 때 $N_o=n-2e+1$이 된다. $m\leq b$인 경우는 $e=1$을 적용하면

$$R_A=\binom{e+N_o}{b}-\binom{N_o}{b}\geq\binom{n-e}{b}=R_N \tag{10}$$

이 되므로 정상적인 복구결과를 찾는 것을 보장할 수 없다.

다음으로 $e\geq 2$인 경우를 살펴본다. $N_o\geq b$일 때,

$$\begin{aligned}R_A&=\binom{e+N_o}{b}-\binom{N_o}{b}>\binom{n-e}{b}\\&=\binom{e+N_o-1}{b}=R_N\end{aligned} \tag{11}$$

이 되므로 정상적인 복구결과를 찾는 것을 보장할

수 없다. 마지막으로 $e+N_o > n-e+1$인 경우, $n-e < e+N_o-1$이 되므로 $N_o \geq b$일 때,

$$R_A = \binom{e+N_o}{b} - \binom{N_o}{b} > \binom{n-e}{b} = R_N \quad (12)$$

이 되므로 정상적인 복구결과를 찾는 것을 보장할 수 없게 된다.

앞의 (i), (ii), (iii)에서 살펴본 e, m, N_o의 관계에 의해 정당한 복구결과를 얻는데 필요한 조건에 따라 다음과 같은 알고리즘을 얻을 수 있게 된다.

▪ For any b and e<m

if $((m > b)$ && (e==1) && $(N_o \leq n-1))$
  Can find the correct decoding result
else if $((e < \frac{1}{2}m)$ && $N_o \leq n-2e))$
  Can find the correct decoding result.
else if $((\frac{1}{2}m < e < \frac{n}{2})$ && $(N_o \leq n-2e-1))$
  Can find the correct decoding result.
else
  Cannot find the correct decoding result

IV. 성능분석 및 결과

4.1 성능분석 척도 및 결과

앞 장의 분석결과와 알고리즘에 따라 다음과 같은 성능분석 척도를 얻을 수 있다.

$$\text{복구율} = \frac{\text{성공적인 복구의 수}}{\text{전송된 메시지로 가능한 모든 복구의 경우}} = \frac{1+\sum_{e=1}^{e \leq \frac{m}{2}}(n-2e+1)+\sum_{e>\frac{m}{2}}^{e<\frac{n}{2}\, e<m}(n-2e)}{1+\sum_{e=1}^{e \leq n}(n-e+1)} \quad (13)$$

$$\begin{aligned}\text{총비용} &= \text{메시지복구비용} + \text{오류처리비용} \\ &= \text{복구율} \times \text{전송비용} \\ &\quad + \text{복구실패율} \times \text{총패킷의수} \times \text{재전송비용}\end{aligned} \quad (14)$$

식 (13)은 메시지 복구율을 나타내며, 식 (14)에서는 성공적인 메시지 복구비용과 복구에 실패할 경우 재전송비용을 고려한 총 복구비용을 나타낸다.

그림 2~그림 5는 리던던시, m과 공격 패킷의 수, e에 대한 네트워크 코딩의 복구율과 총 복구비용의 변화를 보여준다.

그림 2는 $e=2$일 때 네트워크 코딩에 따라 발생하는 리던던시, m의 수에 따른 복구율을 원본 패킷의 수, b의 변화에 대하여 보여준다. 이 그래프에서 $m=4$인 경우부터 복구율이 안정세를 보이는 것을 알 수 있다. 그림 3은 $m=5$일 때 공격 패킷의 수, e가 증가함에 따르는 복구율의 변화를 b에 대하여 보여준다.

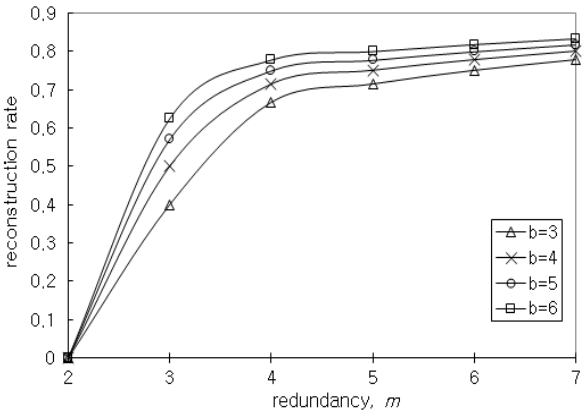

그림 2. 네트워크 코딩에 따라 발생하는 리던던시, m의 수에 따른 복구율의 변화($e=2$)

Fig. 2. Reconstruction rate according to the number of redundancy of network coding($e=2$)

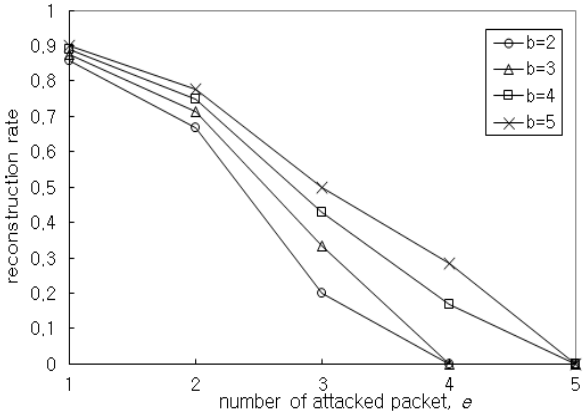

그림 3. 공격패킷의 수, e의 증가에 따른 복구율의 변화 ($m=5$)

Fig. 3. Change in reconstruction rate with increasing number of attack packets, $e(m=5)$

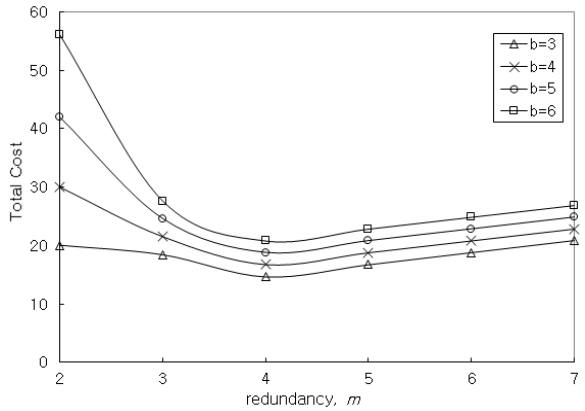

그림 4. 네트워크 코딩에 따라 발생하는 리던던시, m의 수에 따른 복구비용의 변화($e=2$)
Fig. 4. Change in reconstruction cost according to the number of redundancy($e=2$)

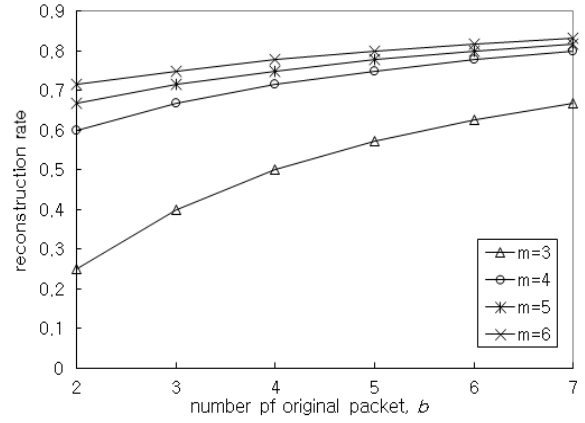

그림 5. 원본 패킷의 수의 증가와 리더던시의 증가에 따른 복구율의 변화($e=2$)
Fig. 5. Change in reconstruction rate with increasing number of original packets and increasing redundancy($e=2$)

그림 4는 $e=2$일 때 네트워크 코딩에 따라 발생하는 리던던시, m의 수에 따른 복구비용의 변화를 원본 패킷의 수, b에 대하여 보여준다. 이 그래프에서 $m=4$일 때 원본 패킷의 수에 관계없이 복구비용이 최소임을 보여준다.

그림 5는 원본 패킷의 수와 리던던시의 증가에 따른 복구율의 변화를 알 수 있는데, 리던던시, m이 4이상일 때, 원본 패킷의 수에 증가에도 복구율이 안정적임을 알 수 있다.

4.2 목적지에서 네트워크 코딩에 대한 복구예

앞의 시나리오에 대하여 <3.2.2>절의 알고리즘을 적용하여 알고리즘이 제대로 동작하는지를 살펴본다. 이 예에서 $b=2$, $n=4$, $m=2$, $e=1$, $N_o=2$(이 예에서는 C_1, C_2가 됨)가 된다. C_3나 C_e 중에 하나가 공격 패킷으로 의심되는데, 다수의 법칙에 의해 C_1, C_2, C_3가 정당한 패킷이 되고 C_e가 공격 패킷으로 판정된다. 알고리즘에 적용하면 $(e=1)<(m=2)$가 되고 $n-e=3>b$, $n-2e=2\geq b$, $N_o=2\leq n-2e$가 되어 정상적인 복구결과를 얻을 수 있음을 알 수 있다.

V. 결 론

사물인터넷 환경이 활발하게 진행되면서 네트워크 용량 증가에 대한 필요성 역시 급증하고 있다. 네트워크 용량을 증가시키기 위한 여러 가지 방안이 논의되고 있으며, 네트워크 코딩도 그 중 하나이다. 이 논문에서는 사물인터넷 환경에서 네트워크 코딩을 적용하여 패킷을 전송했을 때 중간 노드의 공격에 의해 목적지에서 수신한 패킷들 중에 오류가 존재하더라도 여러 개의 디코딩 결과 중에서 정상적인 결과를 찾아낼 수 있는 방안을 제시하였다. 악의적인 노드가 위조한 패킷은 유효하게 절차를 거쳐서 정상적인 패킷과는 구별할 수 없으나 이 논문에서 제안하는 방법은 목적지 노드에서 수신한 패킷 중에 공격받은 패킷이 존재하더라도 디코딩 과정에서 정상적인 결과를 찾을 수 있는 기준과 이에 대한 분석을 보여준다. 결과에 따르면 리던던시가 4 이상일 때 원본 패킷의 증가에 상관없이 복구율과 총복구비용이 모두 안정적이므로 목적지에서 메시지를 복구할 확률이 높아짐을 알 수 있다.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things:A Survey on Enabling Technologies, Protocols and Applications", IEEE Communication Surveys and Tutorial, Vol. 17, No. 4, pp. 2347-2376, 4th Quarter 2015.

[2] J. Li, Y. Liu, Z. Zhang, J. Ren, and N. Zhao, "Towards Green IoT Networking: Performance

Optimization of Network Coding Based Communication and Reliable Storage", IEEE Access, Vol. 5, pp. 8780-8791, May 2017.

[3] R. Ahlswede, N. Cai, S. Y. R. Li, and W. Yeung, "Network Information Flow", IEEE Trans. on Information Theory, Vol. 46, No. 4, pp. 1204-1216, Jul. 2000

[4] D. Boneh, D. Freeman, B. Waters, and J. Katz, "Signing a Linear Subspace: Signatures for Network Coding", Public-Key Cryptography 2009, pp. 68-87, Mar. 2009

[5] K. Lei, S. Zhong, F. Zhu, K. Xu, and H. Zhang, "An NDN IoT Content Distribution Model With Network Coding Enhanced Forwarding Strategy for 5G", IEEE Transactions on Industrial Informatics, Vol. 14, No. 6, pp. 2725-2735, Jun. 2018.

[6] V. N. Swamy, P. Rigge, G. Ranade, A. Sahai, and B. Nikolić, "Network coding for high-reliability low-latency wireless control", 2016 IEEE WCNCW, Doha, Qatar, Apr. 2016.

[7] M. Cebe, B. Kaplan and K. Akkaya, "A Network Coding based Information Spreading Approach for Permissioned Blockchain in IoT Settings", ACM Mobiquitous'18, pp. 470-475, Nov. 2018.

[8] J. H. Kim, D. B. Park, and H. Y. Song, "Network Coding-Based Information Sharing Strategy for Reducing Energy Consumption in IoT Environments", Journal of KICIS, Vol. 41, No. 4, pp. 433-440, Apr. 2016.

[9] C. H. S. Oliveira, Y. Ghamri-Doudane, C. E. F. Brito, and S. Lohier, "Optimal Network Coding-Based In- Network Data Storage and Data Retrieval for IoT /WSNs", IEEE 14th NCA, Sep. 2015.

[10] R. Hallousha, H. Liub, L. Dong, M. Wud, and H. Radha, "Hop-by-hop Content Distribution with Network Coding in Multihop Wireless Networks", Digital Communications and Networks, Elsevier, Vol. 3, No. 1, pp. 47-54, Feb. 2017.

[11] J. Wang, Z. Liu, Y. Shen, H. Chen, L. Zheng, H. Qiu, and S. Shu, "A distributed algorithm for inter-layer network coding-based multimedia multicast in Internet of Things", Computers & Electrical Engineering, Elsevier, Vol. 52, pp. 125-137, May 2016.

[12] G. Y. Lee, "Delay Improvement of Expedited Data from Erasure Coding", Journal of KIIT, Vol. 15, No. 4, pp. 95-103, Apr. 2017.

[13] J. Li, T. Li, J. Ren, and H. C Chao, "Enjoy the Benefit of Network Coding: Combat Pollution Attacks in 5G Multihop Networks", Wireless Communications and Mobile Computing, Hindawi, Vol. 2018, Dec. 2018.

[14] M. Rabin, "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance", Journal of the ACM, Vol. 36. No. 2. pp. 335-348, Apr. 1989.

[15] S Marinkovic and E Popovici, "Network coding for efficient error recovery in wireless sensor networks for medical applications", 1st Int. Conf. on Emerging Network Intelligence, Oct. 2009.

저자소개

이 용 (Yong Lee)

1997년 8월 : 연세대학교 컴퓨터과학과(이학석사)
2001년 2월 : 연세대학교 컴퓨터과학과(공학박사)
2001년 ~ 2003년 : 한국정보보호진흥원 선임연구원
2004년 ~ 2005년, 2009년 ~ 2012년 : 코넬대학교, 방문연구원
2005년 ~ 2007년 : 삼성전자 통신연구소 책임연구원
2007년 ~ 2011년 : 충주대학교 전자통신공학전공 조교수
2020년 3월 현재 : 프리랜서
관심분야 : 네트워크 보안, 차세대 인터넷, IoT보안, 이동통신망 보안, 정보보호