

차량 포그 컴퓨팅에서 LT 코드를 이용한 블록체인 동기화

심 하 용*, 김 우 성°, 이 상 순*

LT Code Based Blockchain Synchronization for Vehicular Fog Computing

Hayong Shim*, Woosong Kim°, Sangsoon Lee*

요 약

5G 이동통신의 eMMB(Enhanced mobile broadband), mMTC(Massive Machine Type Communication), URLLC(Ultra-Reliable Low Latency Communication) 특성을 활용한 다양한 융합 기술이 빠르게 발전하고 있다. 특히, 자율주행 자동차는 5G 융합 기술로 많은 관심을 받고 있고, 최근 3GPP Release 16에서 LTE 기반의 V2X(Vehicle to Everything)통신이 5G NR(New Radio) 기반의 사이드 링크로 표준화 되었다. 차량 포그 컴퓨팅(Vehicular fog Computing)은 군집 자율주행으로 구성되는 일시적 로컬 클라우드로 지연에 민감한 대량의 센서 데이터 등을 빠르게 수집하고 분석하는데 필요하다. 안전한 군집 주행을 위해서는 차량 포그 컴퓨팅 내에 주행 및 안전 데이터의 무결성이 보장되어야 한다. 차량 블록체인(Vehicular Blockchain)은 차량 간 P2P 네트워크에서 생성, 유지되어 데이터의 무결성을 보장하는데 도움을 준다. 하지만, 차량 간 무선 통신의 간섭, 가변적인 차량 이동 속도에 따른 연결의 불안정성으로 인해 블록체인 데이터를 동기화하는데 어려움이 발생한다. 본 논문에서는 차량 포그 컴퓨팅 환경에서 차량 간 블록체인 블록 데이터 동기화를 신뢰성 있고 빠르게 동기화하기 위해 LT 코드를 활용한 방법을 제안하고 구현하였다. 이를 기반으로 테스트 베드를 구축하고 실험을 통해 실현 가능성을 확인하였다.

Key Words : V2X, Vehicular fog Computing, Vehicular Blockchain, LT Code, Blockchain Synchronization

ABSTRACT

Various convergence technologies that utilize eMMB(Enhanced mobile broadband), mMTC(Massive Machine Type Communication), and URLLC(Ultra-Reliable Low Latency Communication) characteristics of 5G mobile communication are rapidly developing. In particular, autonomous vehicles have received much attention for their 5G convergence technology, and in recent 3GPP Release 16, LTE-based Vehicle to Everything(V2X) communication has been standardized as 5G NR(New Radio) based sidelink. Vehicular Fog Computing is a transient local cloud consisting of autonomous driving platooning that is needed to quickly collect and analyze large amounts of delay-sensitive sensor data, etc. For safe vehicle platooning, the integrity of driving and safety data within Vehicular Fog Computing should be ensured. Vehicular Blockchain is created and maintained in an inter-vehicle P2P network to help ensure the integrity of the data. However, interference in inter-vehicle wireless communication, and the instability of connection due to variable vehicle moving speed, make it

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (no. NRF-2017R1C1B1006607).

• First Author : Gachon University Department of Computer Engineering, hys1753@gc.gachon.ac.kr, 학생회원

° Corresponding Author : Gachon University Department of Computer Engineering, woosong@gachon.ac.kr, 정회원

* Gachon University Department of Computer Engineering, sslee@gachon.ac.kr, 정회원

논문번호 : 202012-322-C-RU, Received December 22, 2020; Revised January 13, 2021; Accepted January 18, 2021

difficult to synchronize blockchain data. In this paper, we propose and implement a method utilizing LT codes to synchronize blockchain data between vehicles in a Vehicular Fog Computing environment with reliable and fast. Based on this, we build a test bed and confirmed that it can be realized through experiments.

1. 서론

최근 통계^[1]에 따르면 4차 산업혁명 핵심 기술 중 5G를 응용한 조기 기술 상용화 유망 분야에서 자율 주행 자동차(Autonomous Vehicle)가 1위로 알려졌다. 5G 이동통신 기술의 발전으로 기존 대비 높은 전송속도, 신뢰성, 초연결성, 초저지연 서비스를 제공할 수 있게 됨에 따라 Google(Waymo), Tesla 등의 자율 주행 자동차 관련 연구 기업들의 5G 기술 적용에 대한 노력이 시작되고 있다. 자율 주행을 위해 각 차량은 실시간으로 인근 차량 및 인프라, 등으로부터 각종 센서 데이터, 교통정보를 수신하고 처리하여 교통상황을 실시간으로 모니터링하기 위해 5G V2X 통신 기술을 이용하여 주변의 차량과 방대한 양의 데이터를 주고 받는다.

최근 군집 자율 주행 자동차의 경우, 주행 안전을 위한 데이터 교환 시에, 응답 지연시간 및 기지국 트래픽을 줄이기 위해 차량 간 데이터를 중앙 서버를 거치지 않고 연결되어 있는 차량끼리 실시간으로 데이터를 처리하는 VFC (Vehicular Fog Computing)^[2] 기술이 주목받고 있다. 이때 자율 주행 자동차 간에 주고받는 센싱, 교통 정보, 개인 정보 데이터들의 무결성을 보장하기 위해 P2P 네트워크 기반의 분산 원장 시스템인 블록체인을 이용하여 수집한 데이터를 저장하고 공유하며, 상호 검증을 통해 신뢰를 제공해 주는 차량 블록체인 기술^[3]이 필요하다.

차량 블록체인은 블록체인 동기화에 있어 차량 통신에서 발생하는 잦은 연결 손실에 따른 블록 데이터 교환에 어려움이 있다. 이는 차량 블록체인의 forking 가능성을 높이고 단일 체인에 대한 차량 간 합의를 어렵게 한다. 또한, 제한적 저장장치를 가지는 VFC의 경우, 블록 개수의 증가에 따라 블록체인 데이터 저장 부하가 급격히 증가할 수 있다. 본 논문에서는 소스 코딩 기법을 이용한 차량 블록체인 동기화 기법을 제안한다. 대표적인 소스 코딩 기법 중에 하나인 LT 코드^[4]를 이용하여 차량들은 블록 데이터를 인코딩하여 분산 저장^[5] 및 전송함으로써, 단순 패킷 전송 시 발생할 수 있는 손실 데이터에 의한 블록 데이터 복원 지연 문제를 해결할 수 있다. 또한, 이웃 차량들에 대한 다중 세션을 통한 블록 데이터의 동시 송수신의 경우, 세

션 별 송수신 데이터 스케줄링의 간소화가 가능하다.

본 논문의 구성은 다음과 같은 순서로 기술한다. 2장에서는 V2X, 차량 포그 컴퓨팅, 블록체인 관련 연구 및 동향을 소개한다. 3장에서는 차량 포그 컴퓨팅 환경에서 블록체인 블록 동기화에 적용할 소스 코딩 기법을 설명한다. 4장에서는 소스 코딩 기법 중 하나인 LT 코드를 적용한 블록체인 블록 동기화를 실험하고, 차량 간 끊김 여부에 따른 동기화 시간을 비교 분석한다. 마지막 5장으로 결론을 맺는다.

II. 차량 포그 컴퓨팅 및 블록체인

본 장에서는 차량 포그 컴퓨팅 및 차량 블록체인에 관련 연구 및 동향에 대해 소개한다. 또한, 차량 블록체인에 대한 문제점을 제시한다.

2.1 차량 포그 컴퓨팅(Vehicular Fog Computing)

최근 비디오 스트리밍, AR, VR, IoT 등의 기술이 발전함에 따라 데이터 전송을 위한 네트워크 대역폭이 더 많이 필요하며, 지연시간에 대한 제약이 발생하고 있다. 이에 따른 해결 방안으로 시스코는 포그 컴퓨팅(Fog Computing)^[6]을 제안하였다. 포그 컴퓨팅이란 클라우드처럼 데이터 연산, 스토리지, 애플리케이션의 기능을 제공하지만 중앙 서버를 거치지 않고 연결 노드끼리 실시간으로 데이터를 처리하여 사용자는 클라우드처럼 서비스를 제공하는 분산형 아키텍처이다.

최근 포그 컴퓨팅 기술을 차량 네트워크에 적용하는 연구가 활발히 진행 중이며, 이를 차량 포그 컴퓨팅(VFC: Vehicular Fog Computing)^[7]이라 한다. 자율 주행 자동차 기술은 데이터 지연에 민감하며 안전한 주행 제어를 위해 약 1 Gbps의 센서 데이터를 실시간 분석하고 융합해야 한다^[8]. VFC는 이러한 자율 주행의 조건을 만족시킬 수 있도록 로컬 클라우드 서버를 차량 네트워크를 이용하여 생성, 유지 관리한다.

2.2 차량 블록체인(Vehicular Blockchain)

V2X(Vehicle to Everything) 통신에 기반한 VFC의 경우, 공유 네트워크에서 데이터가 공유되기 때문에 데이터 보안이 주요 관심사 중 하나이다.^[9] 자율 주행을 위한 VFC의 경우 주변 상황을 인지하기 위해 각 차량의 위치, 속도와 같은 상태 등의 데이터를 공

유한다. 따라서, 데이터가 해킹될 경우 자율 주행에 있어 사고를 발생시킬 수 있다. 이와 같은 잠재적 위험이 있는 정보와 데이터를 보호하기 위해 퍼블릭 또는 프라이빗 P2P 네트워크에서 발생하는 모든 트랜잭션의 정보를 참여 주체 간에 공유된 디지털 원장인 블록체인을 기입함으로써 VFC에 기반한 자율 주행에 안전성을 높이는 연구가 진행 중이다.^[10]

차량 블록체인은 데이터 보안 또는 컴퓨팅 자원 접근에 있어서 권한 인증과 같은 절차를 통해 차량 주행의 안전을 도모할 수 있다. 또한, 데이터 보안에 있어 악의적 노드에 의한 데이터 조작 혹은 서비스 거부 공격 등을 막을 수 있을 뿐만 아니라 다수의 노드에 의한 합의 과정을 통해 블록을 생성하므로 데이터 자체에 대한 신뢰성을 가질 수 있다. 가령, 자율 주행 시 특정 선행 차량에서 도로 파손에 대한 거짓 정보 유통을 시도할 경우, 타 차량들이 해당 내용에 대한 블록 생성을 거부할 수 있어 차량 간에 공유되는 데이터의 신뢰성 보장할 수 있다.

하지만, 차량 블록체인 네트워크는 차량 에드혹 네트워크 위에서 동작하므로 차량 에드혹 네트워크의 영향을 받을 수 있다. 차량 에드혹 네트워크는 차량 간의 무선 통신으로 인한 채널 간섭 현상으로 데이터 전송률이 낮으며, 차량 이동 속도에 따라 연결성이 불안정하고 끊김이 자주 발생한다.

이러한 특성은 블록체인 네트워크 동작 시에도 동일하게 발생할 수 있다. 가령 그림 1과 같이 노드가 신규 데이터 블록을 이웃 차량으로부터 수신하고자 블록 요청 메시지를 전송한 이후에 이웃 차량의 이동으로 인해 블록 데이터를 수신 받지 못하거나, 많은 양의 블록 데이터를 수신 받을 경우, 다운로드 중간에 끊어지는 문제점이 있다. 이러한 블록 데이터 송수신 지연은 블록체인 동기화 및 합의 과정을 어렵게 한다.

따라서, 본 논문에서는 블록체인 네트워크 동작 시 블록 데이터를 소스 코딩 기법을 이용하여 동기화하여 V2X 통신의 끊김이 발생하더라도 빠르게 전체 블

록을 수신할 수 있는 방안을 제안한다.

III. 소스코딩 기반 블록체인 동기화

본 장에서는 차량 포그 컴퓨팅 환경에서 블록체인 블록 동기화에 적용하기 위한 소스 코딩 방법에 대하여 서술한다.

3.1 Fountain 코드

Fountain(분수) 코드는 Erased Probability의 확률로 정보가 제거되어 수신 측에서 어떤 정보가 전송되었는지 알 수 없는 Erasure 채널을 모델로 하는 네트워크 환경의 애플리케이션에서 최적의 성능을 위한 채널 부호화 기술이다. 또한, Fountain 코드는 rateless 코딩 방식으로 코딩 비율의 자유도가 높아 무한히 많은 양의 부호화 심볼을 전송해 에러 복원 능력이 탁월하다. 따라서, 에드혹 네트워크와 같은 무선 P2P 네트워크의 전송 애플리케이션에 자주 사용돼 왔다. 대표적인 Fountain 코드로 Luby 박사가 제안한 LT(Luby Transform) 코드와 Shokrollahi 박사의 Raptor 코드가 있다.

3.2 LT(Luby Transform) 코드

LT 코드^[4]는 인코딩 심볼을 생성하기 위해 소스 심볼을 선택하는 확률분포를 설계하여, 다른 Fountain 코드에 비해 디코딩 오버헤드 감소 및 디코딩 효율성 향상의 장점이 있다. Luby 박사가 정의한 LT 코드의 확률분포는 Ideal Soliton distribution, Robust Soliton distribution이다.

LT 코드의 인코딩 및 디코딩 과정은 다음과 같다.

3.2.1 Encoding

1. 주어진 Degree 분포 d 에 따라 소스 심볼의 랜덤 d 를 선택한다.
2. 랜덤 d 에 따라 인코딩 심볼의 이웃 소스 심볼을 균일하게 선택한다.
3. 선택한 이웃 소스 심볼을 XOR 하여 인코딩 심볼을 생성한다.

3.2.2 Decoding

1. 인코딩 심볼 중 degree가 1인 심볼을 찾는다.
2. 해당 심볼을 XOR 되지 않는 소스 심볼이므로 그대로 복호화 한다.
3. 다른 Input 심볼 중 복호화 한 심볼을 이웃으로 가지는 심볼이 있으면 XOR 하여 복호화 한 심

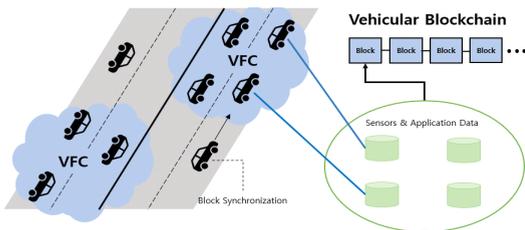


그림 1. 차량 포그 컴퓨팅 환경의 블록체인 구조
Fig. 1. Blockchain structure in Vehicular Fog Computing

불을 제거한다.

4. 1~3의 과정을 반복해 디코딩 한다.

Ideal Soliton distribution(d_i)는 소스 심볼 개수 k 에 대한 식은 다음과 같이 정의된다.

$$d_i = \begin{cases} 1/k & (\text{if } i = 1) \\ 1/i(i-1) & (\text{if } i = 2, \dots, k) \end{cases} \quad (1)$$

이는 최소한의 인코딩 심볼을 통해 오버헤드가 없고, 연산량이 최소화된 가장 이상적(Ideal)인 분포이다. 인코딩 심볼의 아직 처리되지 않은 이웃 소스 심볼들을 리플(ripple)이라고 하며, 그림 2와 같이 전체 인코딩 심볼에서 리플이 1인 디코딩 가능 심볼이 평균적으로 1개가 존재하고, 하나의 심볼이 디코딩된 후 나머지 심볼들의 리플 크기가 평균적으로 1개가 존재한다. 따라서 디코딩이 진행될수록 실패할 확률이 증가한다는 문제점이 있다. 이를 해결하기 위해 Luby 박사는 Robust Soliton distribution(d_r)을 같이 발표하였으며, 소스 심볼 개수 k 에 대한 식은 다음과 같이 정의된다.

$$R = \frac{k}{c \cdot \ln(k/\delta) \sqrt{k}} \quad (2)$$

$$N = \sum_{i=1}^k (d_i + d_r) \quad (3)$$

$$d_i = \begin{cases} 1/iR & (\text{if } i = 1, \dots, R-1) \\ \ln(k/R\delta)/R & (\text{if } i = R) \\ 0 & (\text{if } i = R+1, \dots, k) \end{cases} \quad (4)$$

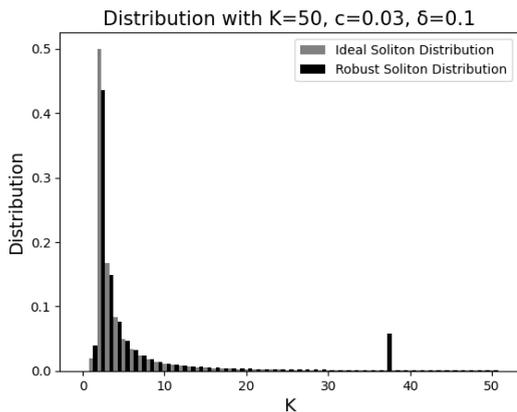


그림 2. $k=50, c=0.03, \delta=0.1$ 일 때 분포
Fig. 2. Distribution with $k=50, c=0.03, \delta=0.1$

$$d_r = (d_i + d_i) / N \quad (5)$$

이는 그림 2와 같이 리플이 1인 인코딩 심볼이 1개 이상 존재하고, 한 심볼이 디코딩된 후 리플의 개수를 Ideal Soliton distribution에 비해 늘려 전체적으로 오버헤드는 소폭 증가하지만, 디코딩 가능 확률을 높인다.

3.3 차량 블록체인 동기화

새로운 노드가 그림 1과 같이 차량 포그 컴퓨팅 환경의 블록체인 네트워크에 접속하여 해당 네트워크를 사용하기 위해서 블록체인 데이터가 없거나 일부만 가진 새로운 노드는 차량 포그 컴퓨팅 환경의 블록체인 네트워크와 블록체인을 일치시켜야 한다. 이를 블록체인의 블록 동기화라 하며, 블록 우선(Block-First) 방식과 헤더 우선(Headers-First) 방식이 있다.

블록 우선 방식은 단일 풀 노드에 부족한 블록 데이터를 요청하여 동기화하는 방식으로 동기화 과정이 단순하다는 장점이 있다. 하지만, 블록 우선 방식은 단일 동기화 노드에 의존함으로 인한 동기화 속도에 제한, 동기화 노드의 연결이 끊어지거나 부적절한 블록을 전송할 경우 다른 풀 노드로부터 재시작이 이루어질 수 있으며, 순차적으로 블록을 전송받지 못할 경우 고아 블록이 생성되어 메모리 점유 공격을 받을 수 있다는 단점이 있다.

반면, 헤더 우선 방식은 풀 노드로부터 블록체인의 모든 블록 헤더를 먼저 다운로드하고, 헤더 정보를 이용하여 다른 여러 풀 노드들로부터 블록 데이터를 동기화하는 방식이다. 따라서, 헤더 우선 방식은 한 노드에 의존하지 않으므로 블록 우선 방식에 비해 속도가 빠르다는 장점이 있다.

본 연구에서는 그림 3과 같이 LT 코드를 사용하지 않은 블록 동기화(No LT-Coding), 사용한 블록 동기화(LT-Coding), 두 가지 시나리오를 구성하였다.

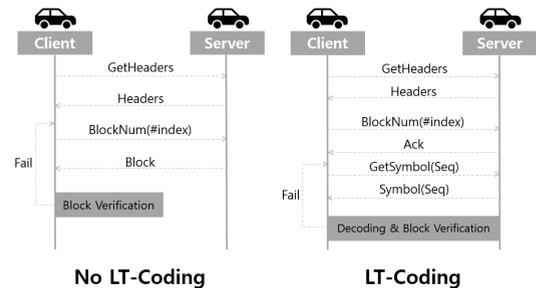


그림 3. 차량 블록체인 동기화 절차
Fig. 3. Vehicular blockchain synchronization

두 시나리오 모두 헤더 우선 방식을 사용하며, GetHeaders, Headers 메시지를 통해 헤더 체인을 동기화한다. 이후 LT 코드를 사용하지 않는 시나리오는 인근 풀 노드들에게 필요한 블록 헤더 해시를 BlockNum 메시지에 담아 전송하고 해당 블록 데이터를 수신받아 검증한다. 반면, LT 코드를 사용한 시나리오의 경우 소스 코딩 적용 여부에 따른 블록체인 동기화 성능을 분석하기 위해 직접 블록 데이터를 수신하는 것이 아닌 인코딩 심볼을 수신하고, 디코딩해 원본 블록으로 재 조합하는 절차를 추가한다.

IV. 실험

본 논문에서 그림 3과 같은 VFC 환경에서 LT 코드를 이용한 블록체인 동기화 기능을 검증하고자 다음과 같이 시뮬레이터를 구현하였다. 서버 i9 3.6GHz, 32GB MEM와 운영체제로 5.4.0-52 generic kernel과 64bit Ubuntu 20.04 LTS을 사용하였으며, VFC 환경 구현을 위해 Core(Common Open Research Emulator) 7.0.1 및 EMANE 1.2.6을 이용하여 가상 차량 노드를 구현하였다. 또한, EMANE의 RF Pipe Model(Frequency 2.4GHz, Bandwidth 20 MHz, Datarate 54Mbps)에 기반한 IEEE 802 a/g Model을 사용해 7개의 차량 플랫폼 환경을 구성하였다.

차량은 블록체인 블록 전송 시, V2X 통신의 비지속적 연결 특성을 감안하여 UDP 프로토콜을 이용하여 블록을 보낸다. 이때 전송 세그먼트는 MTU Size 보다 작은 1024byte로 하여 전송하며, 패킷 손실 발생으로 인해 블록 전송에 실패할 경우, 해당 블록을 재 전송한다.

실험 시나리오로는 구성된 차량 플랫폼 환경에서 기존에 차량 블록체인 네트워크에 참여하고 있는 6개 차량 노드는 추후 VFC에 참여한 1개 차량 노드의 UDP 연결 요청에 따라 세션을 설정하고 그림 3과 같이 LT 코드를 사용하지 않는 블록 동기화(No LT-Coding), LT 코드를 사용한 블록 동기화(LT-Coding), 두 가지 시나리오에 대해 실험하였다. 본 실험에서는 실제 비트코인 블록 100개 (110.1 MB)를 동기화하기 위한 샘플 블록으로 사용하였다. 상기 블록은 블록 생성률에 따라 수십 초 혹은 수십 분 동안 생성될 수 있는 정보량이다.

표 1은 상기 비트코인 블록 100개 각각을 LT 코드를 이용하여 심볼 크기 1024byte로 인코딩 및 디코딩 작업을 수행하는 평균 시간이다. 소스 심볼 개수를 k , 인코딩 심볼 개수를 m 이라 하였을 때 Overhead(O)

표 1. 임의의 100개 블록 평균 인코딩 및 디코딩 시간
Table 1. Average Encoding, Decoding time for random 100 blocks.

Overhead(O)	0.15	0.2	0.25
Average Encoding Time(Sec)	3.40	3.57	3.76
Average Decoding Time(Sec)	8.31	8.93	9.39

는 $(m \div k) - 1$ 로 한다.

LT 코드는 수신된 인코딩 심볼의 개수가 kN 개일 때 최소 디코딩 성공 확률은 $1 - \delta$ 이다^[11]. 해당 테스트 블록 (블록당 평균 소스 심볼의 개수 $\approx 1,100$, $c = 0.03$, $\delta = 0.1$)의 경우 $N \approx 1.08$ 이므로 O 가 0.08에서 최소 디코딩 성공 확률이 0.9임을 확인할 수 있다. 따라서, LT 코드 테스트 시 0.08 이상의 오버헤드를 사용한다. 표 1에서와 같이 오버헤드 증가는 인코딩, 디코딩 지연을 유발하지만 특히 디코딩 복잡도를 늘려 디코딩 지연이 좀 더 발생함을 알 수 있다.

그림 4는 연결 손실이 발생하지 않는 환경에서 6개의 차량 노드로부터 블록 동기화를 진행할 때 소요된 평균 동기화 시간을 보여 준다. 총 비트코인 100개 블록의 평균 블록 동기화하는데, LT 코드($O = 0.15$, $c = 0.03$, $\delta = 0.1$)를 사용한 경우와 사용하지 않은 경우 각각 26.1초 16.5초의 동기화 지연이 발생했다. 소스 코딩 기반의 블록체인 동기화는 원래 소스 데이터에 비해 데이터 오버헤드가 존재하며, 코딩 된 소스 데이터를 노드에 분산 저장하여 이웃 노드로부터 데이터를 수신하는데 디코딩에 필요한 심볼의 수신이 늦어질 경우 전체적인 동기화 성능이 저하된다.

하지만, 차량 간 블록체인 동기화 시 각 차량의 속도 및 지형지물에 의해 연결 손실이 발생할 수 있다. 차량 간 통신의 연결 및 연결 소실은 exponential

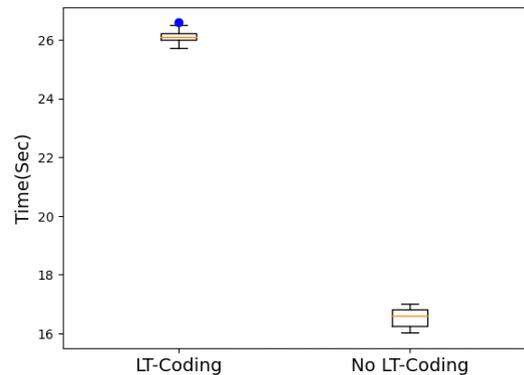


그림 4. 끊김이 발생하지 않는 환경의 블록 동기화 시간.
Fig. 4. Block synchronization without disconnection

distribution을 따른다. 본 연구에서는 각 각의 connection period, disconnection period를 $f(t) = \lambda e^{-\lambda t}$ 을 통해 정하며, 각 period의 λ 값을 λ_c, λ_d 로 표시한다.

그림 5는 연결 손실이 발생하는 차량 통신 환경($\lambda_d = 1.0$)에서 O 에 따른 비트코인 100개 블록의 평균 동기화 시간이다. O 가 적을수록 패킷 loss로 인해 수신 받은 심볼의 개수가 최소 디코딩 가능 심볼 개수보다 부족해 Decoding에 실패하게 된다. 따라서, O 가 적을수록 전체 블록 동기화 시간이 긴 것을 확인할 수 있다.

또한, λ_c 에 따른 평균 연결 지속시간은 λ_c 가 커질수록 연결 지속시간이 짧아진다. 이로 인해 패킷 손실이 더 빈번하게 발생하게 되어 λ_c 가 1.0일 때보다 2.0일 때 평균 블록 동기화 시간이 길게 측정되었다.

그림 6은 연결 손실이 발생하는 차량 통신 환경에서 λ_c, λ_d 변화에 따른 LT 코드($O = 0.15, c = 0.03, \delta = 0.1$)를 사용한 동기화, 사용하지 않은 동기화 시나리오에 대한 임의의 블록 100개의 평균 블록 동기화 시간이다. λ_c 값이 작아 연결이 길게 유지될수록 LT 코드를 사용한 시나리오가 사용하지 않은 시나리오보다 평균 블록 동기화 시간이 길게 걸렸다. 반면, 연결 시간이 짧아질수록 LT 코드를 사용한 시나리오의 블록 동기화 시간이 짧게 걸린 것을 확인할 수 있다. 이는 LT 코드를 사용하지 않은 시나리오의 경우 잦은 패킷 손실로 인해 블록 데이터의 재전송이 자주 발생되지만, LT 코드를 사용한 시나리오의 경우 오버헤드 심볼을 수신해 디코딩 가능하므로 동기화하는 시간을 단축시킬 수 있기 때문이다.

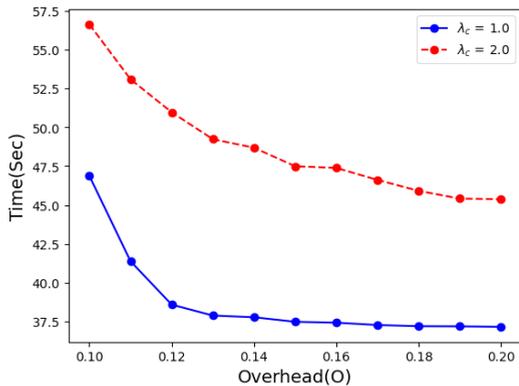


그림 5. 끊김이 발생하는 환경에서 오버헤드 별 블록 동기화 시간.
Fig. 5. Average block synchronization time by overhead in disconnect occurs.

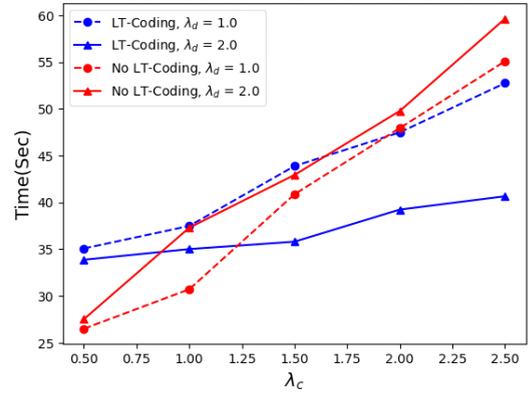


그림 6. 끊김이 발생하는 환경에서 LT 코드를 사용한 동기화, 사용하지 않은 동기화 각각의 시나리오에 대한 평균 블록 동기화 시간.
Fig. 6. Average block synchronization time for each LT-Coding and no LT-Coding scenarios in environment where disconnect occurs.

또한, λ_d 가 1.0일 때보다 2.0일 경우 LT 코드를 사용하지 않은 시나리오에서 블록 동기화 시간이 오히려 길게 측정되었다. 이는 연결 손실 시간은 짧으나 이로 인해 연결 손실 횟수는 증가하게 되었기 때문으로, 이는 블록 데이터 복구 실패 확률을 높이고 결국 동기화 지연을 유발한다. 반면 LT 코드를 사용한 시나리오에서는 연결 손실 시간이 줄어들수록 동기화 시간이 짧게 측정되었다. 비 연결 시간이 짧아짐에 따라 연결이 더 빈번하게 발생하여 단위 시간당 수신할 수 있는 심볼 수가 증가하였기 때문이다. 위와 같은 결과로 LT 코드를 사용한 블록 동기화의 경우, 패킷 손실로 인한 블록 수신 실패를 줄여 블록 재전송 확률을 줄일 수 있어 효율적임을 확인할 수 있다. 따라서 빈번한 연결 손실이 있는 V2X 통신 환경에서 소스 코딩을 이용한 블록체인 동기화가 유리함을 알 수 있다.

V. 결론

본 논문에서는 차량 간 블록체인 블록 동기화 시 LT 코드를 이용한 동기화 방식을 제안하고, UDP를 통한 블록 동기화 프로토콜을 설계하였다. 테스트 베드 구현을 통해 연결 손실이 빈번하게 일어나는 V2X 통신 환경에서 패킷 손실이 자주 발생하는 경우 LT 코드를 통한 블록 동기화 방식이 사용하지 않았을 때보다 동기화 시간이 단축되어 효율적임을 확인하였다. 또한, 각 노드가 인코딩된 심볼 일부만 저장해 블록체인의 스토리지 요구 사항을 줄여 차량의 한정적인 자원을 더욱 효율적으로 사용 가능하다.

References

- [1] D. H. Kim, S. J. Park, and C. S. Leem, "An industry-service classification development of 5g-based autonomous vehicle applications," *JSEBS*, vol. 24, no. 2, pp. 91-112, May 2019.
- [2] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860-3873, Jun. 2016.
- [3] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things J.*, vol. 6, no. 2, pp. 1495-1505, Apr. 2019.
- [4] M. Luby, "LT codes," in *Proc. The 43rd Annu. IEEE Symp. Foundations of Comput. Sci.*, pp. 271-280, Vancouver, Canada, Nov. 2002.
- [5] M. Dai, S. Zhang, H. Wang, and S. Jin, "A low storage room requirement framework for distributed ledger in blockchain," *IEEE Access*, vol. 6, pp. 22970-22975, Mar. 2018.
- [6] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. MCC '12*, pp. 13-16, New York, USA, Aug. 2012.
- [7] Y. Xiao and C. Zhu, "Vehicular fog computing: Vision and challenges," in *Proc. 2017 IEEE PerCom Workshops*, pp. 6-9, Kona, USA, Mar. 2017.
- [8] intel, *Technology and Computing Requirements for Self-Driving Cars* (2014), Retrieved Dec. 10. 2020, from <https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/automotive-autonomous-driving-vision-paper.pdf>
- [9] Ms. Taiyaba, M. A. Akbar, B. Qureshi, M. Shafiq, M. Hamza, and T. Riaz, "Secure V2X environment using blockchain technology" in *Proc. EASE '20*, pp. 469-474, New York, USA, Apr. 2020.
- [10] S. Rahmadika, K. Lee, and K. Rhee, "Blockchain-Enabled 5G autonomous vehicular

networks," in *Proc. 2019 ICSECC*, pp. 275-280, Bandung, Indonesia, Aug. 2019.

- [11] D. J. C. MacKay, "Fountain codes," *IEE Proc. - Commun.*, vol. 152, no. 6, pp. 1062-1068, Dec. 2005.

심 하 용 (Hayong Shim)



2015년 3월: 현재 가천대학교
컴퓨터공학과 학사과정
<관심분야> 블록체인, 채널코딩, 5G 이동통신
[ORCID:0000-0003-3063-535X]

김 우 성 (Wooseong Kim)



2000년 2월: 서울시립대학교 전
전자공학과 학사
2004년 2월: 한국과학기술대학
교 정보공학과 석사
2012년 3월: UCLA 컴퓨터 과
학 박사
2015년 3월: 현재 가천대학교
부교수
<관심분야> 블록체인, P2P 네트워크, 5G 이동통신
[ORCID:0000-0003-0955-3421]

이 상 순 (Sangsoon Lee)



1982년 2월: 인하대학교 전자
공학과 학사
1986년 2월: 인하대학교 전자
계산학과 석사
2005년 2월: 인천대학교 컴퓨
터공학과 박사
1994년 2월: 현재 가천대학교
컴퓨터공학과 부교수
<관심분야> 컴퓨터네트워크, IoT, 시스템소프트웨어
[ORCID:0000-0001-6680-2637]