2025
통신망운용관리 학술대회 논문집
Proceedings of KNOM Conf. 2025

일자 2025년 4월 24일(목) ~ 26일(토)

장소 대전 한국과학기술정보연구원 (KISTI)

주최 한국통신학회 통신망운용관리연구회

주관 한국과학기술정보연구원, 강원대학교 정보통신연구소

KNOM KICS 1996 통신망운용관리연구회 한국통신학회
KICS 한 국 통 신 학 회
KiSTi www.kisti.re.kr 한국과학기술정보연구원 Korea Institute of Science and Technology Information
강원대학교 정보통신연구소

초대의 말씀

한국통신학회 통신망운용관리연구회(KNOM)는 **2025년 통신망 운용관리 학술대회 (KNOM Conference 2025)**를 개최하여 통신망 운용관리 최신 연구개발 현황에 대한 관련분야 학자, 연구원, 네트워크 관리자, 실무담당자들과 토론의 장을 마련하고자 연구논문을 모집합니다.

5G/6G, AI/Cloud, Big Data, Datacenter, Blockchain, Security 기술들을 포함하여, 통신망 전반에 대한 운용 관리와 서비스/기술분야의 최신 연구 공유의 장을 가지려고 합니다. 각 분야 전문가 여러분의 적극적인 논문 투고와 발표를 기대하며, 연구소, 학계, 통신사업자를 포함한 산업계에서 많은 분들이 참석하여 최신 기술을 공유하고 토론하는 좋은 기회가 되기를 바랍니다. 그리고, 선정된 모든 논문은 추가심사를 거쳐 KNOM Review(KCI 등재학술지)에 게재됨을 알립니다.

또한, 본 학술대회에 통신망 운용관리와 관련된 연구소, 하계, 통신서비스 사업자, 통신기기 제조업체 등에서 많은 분들이 참석하여 실제적인 기술을 습득하고 토론할 수 있는 좋은 기회가 될 수 있기를 바랍니다.

2025. 4.

2025 통신망운용관리학술대회 운영위원장 조부승

한국통신학회 통신망운용관리연구회 위원장 김우태

2025 통신망운용관리학술대회 운영위원회

운영위원장	조부승(KISTI 센터장)
학술 프로그램	김경백(전남대학교 교수)
튜토리얼	옥기상(KT)
현장진행	김기현(KISTI)
홍보 및 출판	김명섭(고려대학교 교수)
등록 및 예산	최미정(강원대학교 교수)
자문	석승준(경남대학교 교수) 홍충선(경희대학교 교수) 최덕재(전남대학교 교수) 송왕철(제주대학교 교수) 주홍택(계명대학교 교수)

등록비

구분	On-line 참석	현장 참석	등록비 포함사항			
			발표논문집 (온라인)	프로그램북	점심식사	만찬
일반	300,000 원	300,000 원	○	○	○	○
학생	150,000 원	150,000 원	○	○	○	

◆ 발표논문집 : 4 월말까지 KNOM Conference 홈페이지 다운로드 가능

◆ 점심식사 : 4 월 25 일(금) / 1 회 제공

◆ 만찬 : 4 월 24 일(목)

학술대회 등록방법

- 등록 사이트 : http://kics.or.kr

- 등록 기간

사전등록/저자등록	2025 년 3 월 24 일(월) ~ 2025 년 4 월 23 일(화)
일반등록/ 현장등록	2025 년 4 월 24 일(목)

종합일정표

- 1 일차 : 2025 년 4 월 24 일(목)
- 발표장소 : KISTI 본관 강당 및 KNOM 화상회의 시스템
  - On-line : 추후 제공
  - Off-line : KISTI 본관

시간	구분	발표자
13:00 ~ 15:00	등록	
15:00 ~ 16:00	TS1. 블록체인	좌장 : 김명섭 교수 (고려대학교)
16:00 ~ 16:10	Coffee Break	
16:10 ~ 16:20	개회사	조부승 센터장 (KISTI)
16:20 ~ 17:20	(초청강연) 양자암호통신과 활용기술	심규석 박사 (KISTI)
18:00 ~ 20:00	만찬	

● 2 일차 : 2025 년 4 월 25 일(금)

● 발표장소 : KISTI 키움관 및 KNOM 화상회의 시스템

- On-line : 추후 제공
- Off-line : KISTI 본관

시간	구분	발표자
09:30 ~ 10:30	등록	
10:30 ~ 11:30	TS2. 인공지능 기반 관리	좌장 : 김경렬 박사 (KT)
11:30 ~ 13:30	점심	
13:30 ~ 14:30	TS3. 네트워크 관리 자동화	좌장 : 옥기상 박사 (KT)
14:30 ~ 15:30	Poster. 장소 : 키움관 컨퍼런스룸 로비	좌장 : 김기현 박사 (KISTI)
15:30 ~ 15:40	Coffee Break	
15:40 ~ 16:40	(튜토리얼) 초전도 기반 양자컴퓨팅	구자승 박사 (표준과학연구원)
16:40 ~ 17:00	우수논문 시상식	

● 3 일차 : 2025 년 4 월 26 일(토)

● 발표장소 : KISTI 키움관 및 KNOM 화상회의 시스템

- On-line : 추후 제공
- Off-line : KISTI 본관

시간	구분	발표자
09:30 ~ 10:30	등록	
10:30 ~ 12:30	(전문가 패널) 주제 : 인공지능이 통신망관리에 어떻게 도움을 줄 건인가? 패널 : 주홍택, 최태상, 김명섭, 송왕철, 석승준, 김우태, 옥기상	좌장 : 최미정 교수 (강원대학교)

Technical Sessions 1 – 4월 24일(목)

블록체인

4월 24일(목) 15:00 ~ 16:00,
본관

좌장 : 김명섭 교수(고려대학교)

Technical Sessions 2 – 4월 25일(금)

인공지능기반 관리

4월 25일(금) 10:30 ~ 11:30,
키움관

좌장 : 김경렬 박사(KT)

Technical Sessions 3 – 4월 25일(금)

네트워크 관리 자동화

4월 25일(금) 13:30 ~ 14:30,
키움관

좌장 : 옥기상 박사(KT)

Poster Sessions – 4월 25일(금)

네트워크 관리 자동화

4월 25일(금) 14:30 ~ 15:30,
키움관

좌장 : 김기현 박사 (한국과학기술정보연구원)

코인 믹싱 프로토콜과 믹싱 탐지 방법에 대한 분석

김정헌, 홍원기
포항공과대학교 컴퓨터공학과

{kjheon1118, jwkhong}@postech.ac.kr

Analysis of Coin Mixing Protocols and Mixing Detection Methods

Jeongheon Kim, James Won-Ki Hong
Department of Computer Science and Engineering, POSTECH

요 약

비트코인은 대표적인 암호화폐으로 자리매김하고 있으나, 익명성은 완전하지 않다는 것이 이전의 연구들로부터 밝혀졌다. 코인 믹싱 프로토콜은 이러한 익명성 문제를 해결하기 위해 제안되었으며, 거래의 입력과 출력 간의 관계를 끊거나 모호하게 만듬으로써 익명성을 강화한다. 그러나 익명성을 강화하는 믹싱 프로토콜의 특성 상 불법적인 자금 세탁에 연관되는 경우가 많으며, 몇몇 믹싱 서비스들은 제재를 당한 만큼, 믹싱 프로토콜의 탐지 방법도 연구가 진행되고 있다. 이 논문에서는 비트코인에 대한 다양한 코인 믹싱 프로토콜에 대해서 분석하여 어떠한 방법으로 트랜잭션의 입력과 출력의 관계를 모호하게 만드는지 분석한다. 또한, 믹싱 프로토콜에 대한 다양한 탐지 방법에 대해서도 비트코인 주소를 분류하거나, 트랜잭션을 분류하는 관점으로 나누어 분석한다.

Ⅰ. 서 론

비트코인은 사토시 나카모토가 2008년 발표한 비트코인 백서 [1]에서 제안되었으며, 현재는 암호화폐 시장의 약 60%를 차지하며 대표적인 암호화폐로 자리매김하고 있다 [2]. 중앙화된 기관 없이 신뢰할 수 있는 거래를 가능하게 하는 탈중앙화된 전자 화폐 시스템으로, 모든 거래 내역이 블록체인 상에 투명하게 공개되지만, 거래 당사자의 실제 신원을 특정할 수 없도록 설계되었다.

거래를 진행할 때는 비공개키를 활용하여 디지털 서명을 생성하며, 공개키로부터 비롯된 해시값을 주소로 사용한다. 이렇게 생성된 주소는 비트코인 거래 정보가 블록체인 원장에 공개되어도, 사용자의 실제 신원 정보와 연결되지 않게 하여 가명성(Pseudonymity)을 보장한다 [3]. 그러나 이러한 가명성은 완전한 익명성과는 차이가 있으며, 블록체인 분석 기술이 발전함에 따라 거래 기록이나 패턴을 분석하여 서로 다른 주소들이 동일한 사용자에게 속해있음을 추론하는 것이 가능해졌다 [4].

또한, 많은 암호화폐 거래소들이 자금 세탁 방지 (Anti-Money Laundering) 및 고객 확인 제도 (Know Your Customer)를 준수하기 위해 사용자의 신원 정보를 요구한다 [5]. 이를 통해 주소와 사용자가 직접적으로 연결될 시, 해당 주소와 관련된 모든 거래 내역이 특정될 가능성이 있어 익명성을 침해당할 수 있다.

이러한 비트코인의 익명성 침해 문제를 해결하기 위해 다양한 코인 믹싱 프로토콜들이 제안되었다. 코인 믹싱 프로토콜은 비트코인 거래의 입력과 출력 간의 연결을 끊음으로써, 거래의 보안과 익명성을 강화하는 난독화 메커니즘이다. 대표적으로 2013년에 Gregory Maxwell 이 제안한 CoinJoin [7] 방식이 많이 사용된다. 일반적인 비트코인 트랜잭션의 경우, 1개의 입력과 2개의 출력으로 나뉘어 송신자와 수신자의 식별이 가능하지만, CoinJoin 트랜잭션의 경우 여러 개의 입력과 출력을 하나의 트랜잭션에 포함하여 입력과 출력 간의 연관성을 식별할 수 없도록 한다. 이는 간단하지만 효과적으로 난독화가 가능하도록 하여, 현재 많은 상용 믹싱 서비스들에서 사용하고 있다. CoinJoin 방식을 적용한 Wasabi Wallet [7], JoinMarket [8], Samourai Wallet [9] 등의 오픈소스들도 존재한다.

그러나 믹싱 프로토콜들은 익명성을 보장하기에 자금 세탁, 사이버 범죄 및 다크넷 시장 거래 등의 불법 활동에도 악용될 수 있다 [10]. 예를 들어, 비트코인 믹싱 서비스 중 하나인 Bitcoin Fog 는 운영자가 체포되기 전까지 다크넷 시장을 통해 3억 3천 6백만 달러 이상을 세탁하는데 사용되었다 [11]. 또한 최근에는 북한 해킹 그룹 Lazarus 가 세계 최대 암호화폐 거래소 중 하나인 ByBit 에서 이더리움을 탈취하여, exCh 믹서를 통해 자금 중 일부를 세탁하였다 [12].

이러한 믹싱 프로토콜의 이중적 특성으로 인해 규제 감독이 강화되었으며, 일부 믹싱 서비스들은 정부에 의해 폐쇄당하는 제재를 받게 되었다. Blender.io [13], Tornado Cash [14], Sinbad.io [15]는 모두 Lazarus 그룹의 자금 세탁을 지원한 혐의로, 미국 재무부 해외자산통제국으로부터 제재를 받았다 [16-18].

믹싱 프로토콜이 악용되는 상황을 감지하기 위해 이를 탐지할 수 있는 다양한 방법들이 연구되고 있다. 크게 비트코인의 주소 분류 방법을 사용하여 믹싱 서비스에 연관되어 있는 주소들을 식별하는 방법과, 트랜잭션들을 분석하여 일반적인 트랜잭션과 다른 특징을 가지는 믹싱 트랜잭션을 찾아내는 방법으로 나눌 수 있다. 또한, 이들을 탐지 기준에 따라 휴리스틱 기반, 그래프 분석 기반, 또는 머신러닝에 기반하는 방법 등으로 나눌 수 있다.

본 논문에서는 현존하는 여러 비트코인 기반 믹싱 프로토콜들에 대해서 분석하고, 이러한 믹싱 프로토콜을 탐지하는 연구들의 특징과 한계점들을 분석한다.

Ⅱ. 코인 믹싱 프로토콜

비트코인은 Unspent Transaction Output (UTXO) 블록체인 모델을 사용하므로, 일반적인 트랜잭션은 그림 1 의 좌측처럼 입력 주소 1 개, 출력 주소 2 개로 구성된다. 이는 한 때 블록에 포함된 트랜잭션 중 70% 이상을 차지하였으며, 최근에는 약 40%에 가까운 비중을 차지하고 있다 [19]. 2 개의 출력 주소 중 하나는 수신자에게 전송되며, 나머지 하나는 잔금 주소가 되어 송신자에게 반환된다.

블록체인은 두 출력 주소 중 어떤 게 잔금 주소인지 명시하지 않지만, 휴리스틱 분석을 통해 잔금 주소를 판별하는 방식이 널리 사용된다. 이를 통해 트랜잭션의 송신자와 수신자 사이의 연결성을 파악할 수 있다.

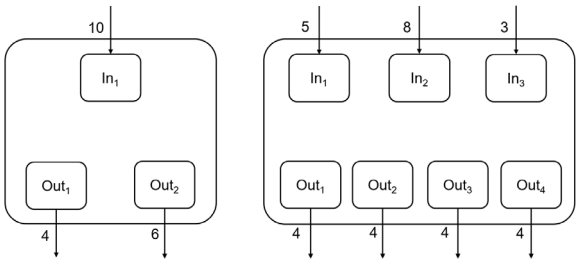

그림 1. 일반적인 트랜잭션 vs CoinJoin 트랜잭션

CoinJoin [6]은 탈중앙화 코인 믹싱 프로토콜로, 트랜잭션 내의 입력과 출력을 쉽게 연결시킬 수 없게 만든다. 그림 1 의 우측에서 볼 수 있듯이, CoinJoin 프로토콜은 비트코인을 전송할 때 혼자 전송하지 않고, 여러 개의 입력과 출력들을 모아 하나의 트랜잭션으로 생성한다. 이때, 출력들을 동일하거나 비슷한 금액으로 출력하여 누가 누구에게 전송하였는지 추적하는 것을 어렵게 만든다.

이러한 CoinJoin 방식은 간단하면서도 비트코인 블록체인 시스템을 변경하지 않고도 완벽히 호환이 가능하여 많은 믹싱 서비스들이 해당 방식을 적용하고 있다. CoinJoin 의 익명성은 해당 프로토콜의 참여자 수에 의존하며, 참여자가 많을수록, 입력과 출력간의 연결 경우의 수가 많아져 익명성이 증가한다.

그러나 CoinJoin 은 탈중앙화 방식이기에 악의적인 사용자가 참여하여 프로토콜을 강제로 중단시키는 등 DoS 공격 및 Sybil 공격에 취약하다. 또한, 해당 프로토콜의 참여자들 간에는 출력과 입력을 연결시킬 가능성이 있기에 내부 익명성을 보장하지 않는다는 단점이 존재한다.

이러한 문제점들을 해결하기 위하여 CoinJoin 을 발전시킨 다양한 믹싱 프로토콜들이 제안되었다. CoinShuffle [20]은 CoinJoin 의 내부 익명성 문제를 해결하기 위해 Decryption Mix Network 구조를 활용한다. 각 참가자는 새로운 출력 주소를 생성하고, 후속 참가자의 공개키를 사용하여 계층별로 암호화한다. 암호화된 주소는 참가자 순으로 무작위로 셔플링되어 출력 주소의 순서를 무작위화하면서도 아무도 입력-출력 매핑 정보를 모르게 함으로써 내부 익명성 문제를 해결한다.

또한, CoinShuffle 의 계층적 암호화와 셔플링을 대체하여 익명 브로드캐스트 프로토콜을 적용하여 효율성을 증가시킨 CoinShuffle++ [21]도 제안되었으며, 각 참여자의 거래 금액 정보까지 Confidential Transaction [22] 방법을 활용하여 숨김으로써, 누가 얼마를 전송했는지조차 완전히 숨길 수 있는 ValueShuffle [23] 프로토콜이 제안되었다. 그러나, 비트코인이 Confidential Transaction 을 지원하지 않기 때문에 ValueShuffle 은 비트코인 시스템과 호환되지 않는다.

해당 프로토콜들은 모두 탈중앙화 방식의 CoinJoin 프로토콜을 기반으로 하고 있으며, 익명성 보장을 위해 오프체인에서 다양한 방식을 적용하였기에 온체인 상에서는 다중 입력-다중 출력의 형식을 띈다.

탈중앙화된 방식의 코인 믹싱 프로토콜들과 달리, 중앙화된 믹서가 믹싱 과정을 주도하는 프로토콜들도 존재한다. 중앙화 믹서는 구현하기 쉽고 일반적으로 탈중앙화 방식에 비해 대기 시간이 짧지만, 믹서가 사용자로부터 받은 코인을 돌려주지 않는 등의 악의적인 행동이 잠재적인 위협이 된다. 중앙화 믹싱 프로토콜들은 이러한 믹서의 악의적인 행동을 탐지하거나 방지할 수 있는 방법들을 포함한다.

Mixcoin [24]은 중앙화 믹싱 프로토콜로, 믹서가 사용자에게 지정된 마감일까지 지정된 출력 주소로 동일한 양의 코인을 반환하겠다고 서명한 보증을 제공하게 한다. 믹서가 이를 지키지 않으면, 사용자는 이 보증을 공개하여 잘못된 동작을 증명하여 믹서에게 페널티를 부과하도록 할 수 있다. 또한, 수수료 패턴 추적을 방지하기 위하여 무작위적으로 수수료를 부과하는 방식을 제안하였다. 그러나, CoinJoin 과 유사하게 믹서가 믹싱에 참여한 사용자들을 식별할 수 있어 내부 익명성이 보장되지 않는다는 단점을 가지고 있다.

Blindcoin [25]은 Mixcoin 에 기반하며, 내부 익명성을 보장하기 위하여 블라인드 서명(Blind Signature) [26]를 활용한다. 사용자가 믹서에게 코인을 전송할 때, 출력 주소를 블라인드 서명을 요구하는 메시지에 포함하여 제출하며, 믹서는 출력 주소를 확인하지 않은 채로 메시지에 서명한다. 이후 이를 기반으로 믹서는 서명 유효성만을 확인하며, 어떤 입력과 연관되었는지 알지 못한 채 출력 주소로 전달하는 방식으로 내부 익명성을 보장한다. 그러나, Mixcoin 과 Blindcoin 은 보증 체계에 기반하기에, 원천적으로 코인 도난을 방지할 수는 없으며, 도난을 감지하고 처벌을 내리는 것만 가능하다.

TumbleBit [27]은 신뢰할 수 있는 중개자에 의존하지 않고도 익명성과 공정한 거래를 제공하도록 설계된 Fair Exchange [28] 방식을 채택하였다. 이는 비트코인 스마트 컨트랙트에 통합되어 믹서의 코인 도난을 방지할 수 있다. 이외에도, 신뢰 실행 환경(Trusted Execution Environment) [29]을 활용하는 Obscuro [30], 믹서를 감시할 수 있는 감시자 시스템을 도입한 CoinLayering [31]과 CoinFA [32], 영지식 증명 [33] 방식을 도입한 SofitMix [34] 프로토콜 등이 존재한다.

Ⅲ. 코인 믹싱 탐지 방법

익명성 강화를 위해 많은 코인 믹싱 프로토콜들이 제안되었으나, 서론에서 설명하였듯이 이를 불법적인 활동에 악용하는 경우가 증가하고 있다. 이러한 악용을 막기 위하여 코인 믹싱을 탐지할 수 있는 방법들도 제안되고 있다. 해당 섹션에서는 다양한 코인 믹싱 탐지 방법들을 믹싱 서비스에 연관된 주소를 식별하여 믹싱을 탐지하는 방법(Address Classification)과 일반적인 트랜잭션과 달리 믹싱 프로토콜을 통해 생성된 믹싱 트랜잭션을 식별하고 분류하는 방법(Transaction Classification)으로 나누어 설명한다.

1. Address Clasffication 기반

비트코인의 주소를 분류하고자 하는 연구는 꾸준히 진행되어 왔다. 불법 거래에 연루된 주소를 구분하는 등의 목적을 위해 패턴 분석, 머신러닝 등의 방법을 활용하는 연구들이 제안되었다. 이에 기반하여 실제 운영되고 있는 믹싱 서비스 또는 커뮤티니를 탐지하고자 하는 다양한 연구들이 제안되었다.

[35] 에서는 믹싱 서비스 주소를 머신러닝의 비지도 방식으로 탐지하기 위해, Deep Autoencoder 기반 임베딩과 커뮤니티 클러스터링(Community Clustering), 이상치 탐지(Outlier Detection)를 결합한 탐지 프레임워크 BGID 를 제안하였다. 비트코인의 트랜잭션들을 그래프로 모델링한 후, 노드의 구조적 관계를 저차원 벡터로 임베딩하고, 클러스터 내부의 이상 행동을 보이는 주소를 믹싱 서비스로 탐지한다. 휴리스틱에 기반하는 커뮤니티 클러스터링 모델과 비교하였으며, 해당 방식보다 높은 F1 점수를 기록하였다. 그러나 제안한 이상치 탐지 계산의 시간 복잡도 문제와 실제 비트코인 거래 그래프에 적용하기에는 시간이 너무 오래 걸린다는 한계점이 있다.

[36] 에서는 비트코인 트랜잭션의 시간적 흐름과 속성 정보를 동시에 반영하는 Attributed Temporal Heterogeneous (ATH) 모티프를 정의하고, 이를 기반으로 비트코인 네트워크를 이중 구조로 구성하여 믹싱 서비스 주소를 탐지하는 머신러닝 모델을 제안하였다. 이때, 비트코인 네트워크는 노드가 비트코인 주소로만 이루어진 Address-Address Interaction Network (AAIN)과, 노드가 트랜잭션과 주소로 이루어진 Transaction-Address Interaction Network (TAIN)으로 구성된다. AAIN 은 주소 간 상호작용을 빠르게 모델링하고, 거래 패턴의 시간 흐름을 파악하는데 유리하며, TAIN 은 거래에 대한 정보들까지 포함하므로 더 정밀한 탐지가 가능하다.

이때, 현실에서 믹싱 서비스의 비트코인 주소는 대부분 알려지지 않아 라벨 수가 부족하다는 문제점이 있다. 이를 극복하기 위해 Positive and Unlabeled learning (PU learning) 기법을 사용하였다. PU learning 은 음성(negative) 데이터를 직접 학습하지 않고도 양성과 비양성을 구분할 수 있는 모델을 학습하는 기법으로, 해당 연구에서는 믹서의 주소인 경우가 양성이 된다. 이때, 나머지 대부분의 주소는 믹서 주소라는 라벨이 없어 불확실한 상황에서도 높은 정확도를 보여주어 라벨 부족 문제를 효과적으로 완화하였다.

[37] 에서는 믹싱 패턴이 점점 고도화됨에 따라 단순한 휴리스틱에 따르거나 단일 머신러닝 모델에 기반하는 것은 한계가 존재한다고 주장하였다. 이러한 문제를 해결하기 위해 여러 개의 개별 모델을 조합하여 최적의 모델로 일반화하는 방법인 앙상블 학습 (Ensemble Learning)을 활용하여 믹싱 서비스 주소를 탐지하는 방법을 제안하였다. 이 모델은 비트코인 주소, 트랜잭션, 네트워크 세 단계의 통계적 특성을 조합하여 설계된 특징들과, 특정 주소를 중심으로 1-hop 이웃까지 포함하는 서브그래프 기반 그래프 커널 분류기를 함께 사용한다. 그래프 커널(graph kernel)은 두 그래프 사이의 유사도를 계산하는 방법으로, 이를 통해 서로 다른 유형의 주소들이 가지는 구조적 차이를 포착할 수 있다. 모델은 AdaBoost, GBDT, XGBoost, LightGBM 과 그래프 커널 분류기의 출력을 랜덤 포레스트(Random Forest)로 통합하여 최종적으로 판별한다.

실제 상용 믹서였던 BitcoinFog 의 믹서 주소를 포함한 데이터셋으로 실험을 진행하였으며, 단일 모델만을 쓰는 이전 연구보다 높은 99.84%의 정확도를 보여주었다. 그러나 믹싱 서비스 주소는 BitcoinFog 에 한정되어 수집되었기 때문에, 모델이 해당 믹서의 패턴에 과적합(overfitting)되었을 가능성이 존재한다. 다양한 믹싱 서비스의 주소에 대해 일반화 가능성을 검증할 필요가 있다. 또한, 그래프 기반 커널 분류기는 서브 그래프 수가 많을수록 계산 복잡도가 증가하기에 대규모인 비트코인 네트워크에서는 시간이 지날수록 비용이 증가한다는 문제점이 있다.

[38] 에서는 CoinJoin 프로토콜을 기반으로 생성되는 CoinJoin 트랜잭션을 기반으로 하여, 트랜잭션과의 거리, 활동 주기, 네트워크 및 사용자 정보들을 반영한 특징들을 설계하였다. 이를 활용하여 클러스터링을 수행하여 불법 사용자 커뮤니티를 식별하였으며, 해킹 등에 연관된 주소들이 일관되게 같은 클러스터로 묶이는 결과를 보여주었다.

2. Transaction Classification 기반

주소에 기반한 분석뿐 아니라, 블록체인에 기록된 트랜잭션들을 분석하여 믹싱 프로토콜을 통해 생성된 믹싱 트랜잭션들을 분류하는 방법들에 대한 다양한 연구들이 제안되었다.

[39] 에서는 비트코인 트랜잭션 그래프를 구축하고, 믹서 주소를 수집하여 해당 믹서 주소의 트랜잭션 패턴을 분석하였다. 해당 그래프 분석을 통하여 대부분의 믹싱 트랜잭션이 여러 입력에서 하나의 출력, 하나의 입력에서 여러 출력, 또는 여러 입력에서 여러 출력의 형태를 가지는 복잡한 분산형 구조를 반복적으로 생성한다고 주장하였다. 이를 기반으로 특징들을 추출하여 C4.5 결정 트리(Decision Tree) 분류기를 활용하는 머신러닝 모델을 제안하였다. 이때, 모델 과적합을 방지하고 트리 크기를 감소시키기 위해 Reduced Error Pruning (REP) 기법을 사용하였으며, 불균형 해소를 위해 믹서의 주소와 믹서의 것이 아닌 주소의 개수를 비슷하게 데이터셋을 구성하였다. 실험 결과로는 97% 이상의 정확도, 정밀도, 재현율을 보여주며, 모델은 55 개의 노드로 경량화하여 실시간 환경에 적용 가능한 수준이라고 주장하였다. 그러나, 해당 연구에서 사용된 8 개의 특징들에는 입출금의

흐름 등 시계열 정보는 반영되지 않았다는 한계점이 있다.

[40] 에서는 위의 연구에서 단일 거래 특징들에만 의존한다는 한계점을 해결할 수 있는 거래 시퀀스 기반의 트랜잭션 트리 정보를 활용한 Long Short-Term Memory (LSTM) 기반 탐지 모델을 제안하였다. 이 모델은 타겟 거래를 중심으로 전후 트랜잭션 트리를 구성하고, 각 레벨의 통계 정보를 시퀀스로 변환하여 시계열 정보를 처리할 수 있는 양방향 LSTM 에 입력한다. 학습을 위해 휴리스틱 기반의 CoinJoin 탐지 규칙을 통해 판별한 트랜잭션들을 라벨링하여 사용하였다. 성능은 Graph Convolutional Network (GCN)을 활용한 이전 연구와 비교하였으며, 정확도는 GCN 을 활용한 방법이 조금 더 높지만, 해당 연구는 높은 재현율을 보여주어, 새로운 믹싱 유형에 대한 일반화 성능이 우수함을 주장하였다. 다만 휴리스틱 기반의 라벨링을 진행하여 라벨에 대한 노이즈가 우려되며, 일부 트랜잭션 트리의 크기가 방대하여 고정 벡터로 압축 시 정보 손실의 가능성이 존재한다.

[41] 에서는 Wasabi Wallet 의 CoinJoin 트랜잭션이 갖는 고유한 구조적 특징을 식별하고 이를 기반으로 한 휴리스틱 탐지 알고리즘을 제안하였다. 다양한 시기별 실거래 데이터를 수집하고 분석하였으며, 입력 및 출력의 개수, 주소 유형, 동일 금액 출력의 정렬 방식 등의 Wasabi Wallet 에서 생성된 CoinJoin 의 고유 거래 패턴을 발견하였으며, 높은 정밀도로 탐지하는 결과를 보여주었다.

[42] 에서는 CoinJoin 거래를 탐지하기 위해 트랜잭션 수준의 feature 들을 활용한 지도학습 머신러닝 기반 탐지 시스템을 제안하였다. Ablation study 를 통해 가장 중요한 feature 만을 선택하였으며, 다양한 모델(Decision Tree, KNN, Logistic Regression, Random Forest, XGBoost)들을 사용하여 성능을 비교하였다. Logistic Regression 모델이 가장 높은 정확도를 보여주었으며, CoinJoin 거래를 미탐하는 결과 없이 성공적으로 탐지하였다. 그러나 데이터셋 구성을 위해 믹싱 트랜잭션을 수집하는데 사용한 방법을 공개하지 않아, 라벨링의 신뢰성이 검증되지 않아 모델의 과적합 가능성이 존재한다는 한계점이 있다.

[43] 에서는 CoinJoin 프로토콜 탐지에 국한되지 않고, 실제 믹싱 서비스들(MixTum, Blender, CryptoMixer)을 대상으로 직접 거래를 수행하여 트랜잭션 및 주소 라벨을 수집하였다. 수집한 데이터를 통해 먼저 트랜잭션 레벨에서의 패턴 분석을 진행하여, 해당 트랜잭션이 믹싱에 관련되었을 여부를 판단한다. 트랜잭션 레벨에서만 분석하였을 땐 거짓 양성(False Positive) 비율이 높아, 체인 레벨의 패턴을 함께 활용한다. 하나의 믹싱 트랜잭션에서 시작해 연속적으로 이어지는 거래 흐름을 추적하여 전체 믹싱 프로세스의 행동 및 시간적 패턴을 파악한다. 평가에서는 세 가지 믹싱 서비스에서 생성된 총 45 개의 믹싱 트랜잭션을 모두 탐지하였고, 오탐이 발생하지 않아 높은 정확도와 재현율을 보여주었다. 그러나 제한된 믹싱 트랜잭션 데이터로 인해 일반화 성능은 검증되지 않았다는 한계점이 있다.

논문	탐지 대상	주요 기법	믹싱 프로토콜
[35]	믹서 주소	Deep Autoencoder	특정되지 않음
[36]	믹서 주소	Hybrid Motif + PU learning	믹싱 서비스(BitcoinFog, BitLaunder, Helix)
[37]	믹서 주소	Dual Ensemble Classifier	BitcoinFog 기반
[38]	의심 사용자 커뮤니티	K-means Clustering	CoinJoin
[39]	믹싱 트랜잭션	Decision Tree	특정되지 않음
[40]	믹싱 트랜잭션	LSTM	CoinJoin
[41]	CoinJoin 트랜잭션	Heuristic Pattern	CoinJoin
[42]	CoinJoin 트랜잭션	Logistic Regression	CoinJoin
[43]	믹싱 트랜잭션	Heuristic Pattern	믹싱 서비스(MixTum, Blender, CryptoMixer)

표 1. 코인 믹싱 탐지 방법

Ⅳ. 결론 및 향후 연구

본 논문에서는 비트코인이 완전한 익명성을 제공하지 못한다는 한계점을 보완하기 위해 제안된 다양한 코인 믹싱 프로토콜들에 대해서 분석하였다. 또한, 이러한 믹싱 프로토콜들을 탐지하는 다양한 연구들을 주소 분류 기반과 트랜잭션 분류 기반으로 나누어 분석하였으며, 각 연구들이 가지는 한계점도 같이 분석하였다. 많은 탐지 방법들이 CoinJoin 프로토콜에 기반하여 믹싱을 탐지하거나, 실제 믹싱 서비스에서 비롯된 믹싱 트랜잭션을 기반으로 믹싱 탐지를 진행하였다.

그러나 CoinJoin 에 기반한 프로토콜이 아닌, Fair Exchange 에 기반하는 여러 프로토콜들이 실제 믹싱 서비스에 사용되는지 현황을 파악하는 연구가 향후에 진행되어야 하며, 이를 탐지할 수 있는 방법들에 대한 연구도 파악되어야 할 것이다.

ACKNOWLEDGMENT

이 논문은 25 년도 정부(경찰청)의 재원으로 과학치안진흥센터 경찰건강 스마트관리 사업의 지원을 받아 수행된 연구임(No. RS-2022-PT000186)

참 고 문 헌

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] CoinMarketCap. "Bitcoin Dominance." CoinMarketCap.

https://coinmarketcap.com/charts/bitcoin-dominance/ (accessed March 12, 2025).

[3] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in bitcoin," in Financial Cryptography and Data Security: 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers 17, 2013: Springer, pp. 34-51.

[4] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," IEEE communications surveys ₩& tutorials, vol. 20, no. 4, pp. 3416-3452, 2018.

[5] S. M. Moreno, J.-M. Seigneur, and G. Gotzev, "A survey of KYC/AML for cryptocurrencies transactions," in Handbook of research on cyber crime and information privacy: IGI Global, 2021, pp. 21-42.

[6] G. Maxwell. "Coinjoin: Bitcoin privacy for the real world." https://bitcointalk.org/?topic=279249 (accessed 02/21, 2025).

[7] Wasabi Wallet, https://wasabiwallet.io/ (accessed 04/03, 2025)

[8] JoinMarket Bitcoin Wiki, https://en.bitcoin.it/wiki/JoinMarket (accessed 04/03, 2025)

[9] Samourai-Wallet github https://github.com/Samourai-Wallet (accessed 04/04, 2025)

[10] R. Van Wegberg, J.-J. Oerlemans, and O. van Deventer, "Bitcoin money laundering: mixed results? An explorative study on money laundering of cybercrime proceeds using bitcoin," Journal of Financial Crime, vol. 25, no. 2, pp. 419-435, 2018.

[11] U.S. Department of Justice. "Bitcoin Fog Operator Sentenced for Money Laundering Conspiracy." https://www.justice.gov/archives/opa/pr/bitcoin-fog-operator-sentenced-money-laundering-conspiracy (accessed March 12, 2025).

[12] C. Rover. "Bybit hacker Moves 5,000 ETH Through eXch Mixer and Chainflip." https://blockchain.news/flashnews/bybit-hacker-moves-5-000-eth-through-exch-mixer-and-chainflip (accessed March 13, 2025).

[13] "Blender.io Mixer." https://blendor.io/ (accessed March 12, 2025).

[14] A. Pertsev, R. Semenov, and R. Storm, "Tornado cash privacy solution version 1.4," Tornado cash privacy solution version, vol. 1, no. 6, 2019.

[15] Chainalysis. "U.S. Sanctions Crypto Mixer Sinbad.io for Role in North Korean Laundering Activities." https://www.chainalysis.com/blog/crypto-mixer-sinbad-sactioned-north-korean-laundering/ (accessed March 11, 2025).

[16] U.S. Department of the Treasury. "U.S. Treasury Issues First-Ever Sanctions on a Virtual Currency Mixer, Targets DPRK Cyber Threats." https://home.treasury.gov/news/press-releases/jy0768 (accessed March 10, 2025).

[17] U.S. Department of the Treasury. "U.S. Treasury Sanctions Notorious Virtual Currency Mixer Tornado Cash." https://home.treasury.gov/news/press-releases/jy0916 (accessed March 10, 2025).

[18] U.S. Department of the Treasury. "Treasury Sanctions Mixer Used by the DPRK to Launder Stolen Virtual Currency." https://home.treasury.gov/news/press-releases/jy1933 (accessed March 10, 2025).

[19] Bitrefill. “One-Input and two-Output Transactions” https://transactionfee.info/charts/transactions-1in-2out/ (accessed April 3, 2025)

[20] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "Coinshuffle: Practical decentralized coin mixing for bitcoin," in Computer Security-ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part II 19, 2014: Springer, pp. 345-364.

[21] M. H. Ibrahim, "Securecoin: a robust secure and efficient protocol for anonymous bitcoin ecosystem," Int. J. Netw. Secur., vol. 19, no. 2, pp. 295-312, 2017.

[22] S. Noether and A. Mackenzie, "Ring confidential transactions," Ledger, vol. 1, pp. 1-18, 2016.

[23] T. Ruffing and P. Moreno-Sanchez, "Valueshuffle: Mixing confidential transactions for comprehensive transaction privacy in bitcoin," in Financial Cryptography

and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers 21, 2017: Springer, pp. 133-154.

[24] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for bitcoin with accountable mixes," in Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers 18, 2014: Springer, pp. 486-504.

[25] L. Valenta and B. Rowan, "Blindcoin: Blinded, accountable mixes for bitcoin," in Financial Cryptography and Data Security: FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, January 30, 2015, Revised Selected Papers, 2015: Springer, pp. 112-126.

[26] D. Chaum, "Blind Signature System," in Crypto, 1983, vol. 83: Springer, p. 153.

[27] E. Heilman, L. Alshenibr, F. Baldimtsi, A. Scafuro, and S. Goldberg, "Tumblebit: An untrusted bitcoin-compatible anonymous payment hub," in Network and distributed system security symposium, 2017.

[28] S. Barber, X. Boyen, E. Shi, and E. Uzun, "Bitter to better—how to make bitcoin a better currency," in Financial cryptography and data security: 16th international conference, FC 2012, Kralendijk, Bonaire, Februray 27-march 2, 2012, revised selected papers 16, 2012: Springer, pp. 399-414.

[29] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," in 2015 IEEE Trustcom/BigDataSE/Ispa, 2015, vol. 1: IEEE, pp. 57-64.

[30] M. Tran, L. Luu, M. S. Kang, I. Bentov, and P. Saxena, "Obscuro: A bitcoin mixer using trusted execution environments," in Proceedings of the 34th Annual Computer Security Applications Conference, 2018, pp. 692-701.

[31] N. Lu, Y. Chang, W. Shi, and K.-K. R. Choo, "CoinLayering: an efficient coin mixing scheme for large scale bitcoin transactions," IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 3, pp. 1974-1987, 2020.

[32] X. Yang, P. Zeng, K.-K. Raymond Choo, C. Li, and Y. Yang, "CoinFA: an efficient coin mixing scheme with flexible amounts," The Computer Journal, vol. 67, no. 12, pp. 3141-3150, 2024.

[33] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems," in Providing sound foundations for cryptography: On the work of shafi goldwasser and silvio micali, 2019, pp. 203-225.

[34] H. Xie, S. Fei, Z. Yan, and Y. Xiao, "SofitMix: A secure offchain-supported bitcoin-compatible mixing protocol," IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 5, pp. 4311-4324, 2022.

[35] NAN, Lihao; TAO, Dacheng. Bitcoin mixing detection using deep autoencoder. In: *2018 IEEE Third international conference on data science in cyberspace (DSC)*. IEEE, 2018. p. 280-287.

[36] WU, Jiajing, et al. Detecting mixing services via mining bitcoin transaction network with hybrid motifs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021, 52.4: 2237-2249.

[37] XU, Chang, et al. How to find a bitcoin mixer: A dual ensemble model for bitcoin mixing service detection. *IEEE Internet of Things Journal*, 2023, 10.19: 17220-17230.

[38] WAHRSTÄTTER, Anton, et al. Improving cryptocurrency crime detection: Coinjoin community detection approach. *IEEE Transactions on Dependable and Secure Computing*, 2023, 20.6: 4946-4956.

[39] Rathore, M. Mazhar, Sushil Chaurasia, and Dhirendra Shukla. "Mixers Detection in bitcoin network: a step towards detecting money laundering in crypto-currencies." *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 2022.

[40] SUN, Xiaowen; YANG, Tan; HU, Bo. LSTM-TC: Bitcoin coin mixing detection method with a high recall. *Applied Intelligence*, 2022, 52.1: 780-793.

[41] TIRONSAKKUL, Tin, et al. The unique dressing of transactions: Wasabi coinjoin transaction detection. In: *Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference*. 2022. p. 21-28.

[42] DEKHIL, Oumayma, et al. Detecting Bitcoin CoinJoin Transactions Using Machine Learning. In: *2024 6th International Conference on Blockchain Computing and Applications (BCCA)*. IEEE, 2024. p. 347-355.

[43] SHOJAEINASAB, Ardeshir; MOTAMED, Amir Pasha; BAHRAK, Behnam. Mixing detection on bitcoin transactions using statistical patterns. *IET Blockchain*, 2023, 3.3: 136-148.

데이터 무결성과 액세스 제어를 위한 DID 기반 IoT 시스템 설계

배태모[1], 방지원[2], 최미정[1,2,3,*]
[1]강원대학교 컴퓨터공학과
[2]강원대학교 빅데이터메디컬융합학과
[3]강원대학교 데이터사이언스학과

{tm3693, jiwonbang, mjchoi}@kangwon.ac.kr

Design of a DID Based IoT System for Data Integrity and Access Control

Taemo Bae[1], Jiwon Bang[2], Mi-jung Choi[1, 2, 3, *]
[1]Dept. of Computer Science and Engineering, Kangwon National Univ.
[2]IGP. of Medical Bigdata Convergence, Kangwon National Univ.
[3]Dept. of Data Science, Kangwon National Univ.

요 약

분산 원장 기술(Distributed Ledger Technology)은 데이터를 중앙 서버가 아닌 여러 노드에 분산하여 저장하는 방식으로, 단일 장애 지점(Single Point of Failure)을 제거하고 데이터 무결성을 보장하는 보안성이 뛰어난 기술이다. IoT(Internet of Things)는 다양한 모듈을 탑재한 사물이 데이터를 수집하고 송수신하는 기술로, 여러 산업 분야에 적용되고 있다. 그러나 IoT 환경에서는 다양한 보안 취약점이 존재하며, 하나의 공격이 전체 시스템에 심각한 영향을 미칠 수 있다. 본 논문에서는 이러한 문제를 해결하기 위해 IOTA(Internet of Things Application)를 활용하여 데이터 무결성을 보장하고, IPFS(InterPlanetary File System)를 이용하여 대용량 데이터 저장의 한계를 극복하는 시스템을 제안한다. 또한, 데이터 액세스 제어를 강화하고 중간자 공격(Man-in-the-Middle Attack)을 방지하기 위해 DID(Decentralized Identifier) 기반의 VC(Verifiable Credential)/VP(Verifiable Presentation) 신원 인증 방식을 적용한다.

Ⅰ. 서 론

분산 원장 기술(Distributed Ledger Technology)은 중앙집중화 데이터베이스가 아닌 여러 노드가 동일한 데이터를 기록하고 검증하는 기술이다. 이를 통해 탈중앙성, 신뢰성, 보안성을 동시에 확보할 수 있다. 분산 원장 기술은 처음에는 암호화폐를 위한 기술이었다. 하지만 시간이 지나면서 보안성과 탈중앙성이 필요한, 헬스케어, 스마트 그리드, 군사·국방 등 다양한 분야로 확대되고 있다[1, 2, 3, 4]. 대표적인 사례로, 에스토니아 정부는 국민의 의료 기록과 세금 정보 등을 중앙 데이터베이스가 아닌 분산 원장에 저장한다. 이로 인해, 높은 보안성과 투명성을 제공하고 있다. 에너지 거래 플랫폼인 Power Ledger는 블록체인을 기반으로 한 소비자 간 전력 거래 기록을 처리하는 시스템을 운영 중이다. 이처럼 분산 원장 기술은 정부, 산업, 공공 분야 전반에서 활용 가능성이 점차 확대되고 있다.

IoT(Internet of Things)는 사물이 센서, 카메라 등에서 수집된 데이터를 통신 모듈을 이용하여 주고받는 기술이다. IoT 기술은 의료, 산업, 가정 등 여러 분야에서 사용되며, 특징으로 중앙집중화 구조, 실시간성, 제한된 성능 등의 특징을 가진다. 그러나 IoT 기술은 구조가 중앙집중화되어 있어, 단일 장애 지점을 가진다. 또한, 제한된 성능으로 인해 다양한 보안 취약점에 쉽게 노출된다. 실제로 MITRE의 조사에 의하면, 2015년부터 2025년까지 IoT 기술에서 발견된 보안 취약점은 총 1,293건에 달한다[5]. IoT 기술에서 보안은 중요하다. IoT 디바이스에서 수집된 데이터가 변조되거나, 망가진 데이터를 수집하게 될 경우 전체 시스템에 장애를 일으킬 수 있다. 또한, 민감한 데이터를 제3자에 의해 유출되는 경우 개인정보 침해로 이어질 수 있다.

본 논문에서는 IoT 기술의 보안 문제를 해결하기 위한 시스템을 제안한다. 분산 원장 기술을 활용하여 데이터 무결성을 보장하고, IPFS(InterPlanetary File System)를 이용함으로써 확장성 및 대용량 데이터 저장의 제약을 극복한다. 또한, DID(Decentralized IDentifier)를 기반으로 한 신원 인증 및 데이터 액세스 제어를 통해 중간자 공격(Man-in-the-Middle Attack) 등을 막는다.

Ⅱ. 배경지식

분산 원장 기술은 데이터를 중앙집중화 저장 장치가 아닌, 지리적, 국가적, 기관적으로 분산된 여러 노드에

동일하게 저장하는 기술이다. 분산 원장 기술은 중앙 관리자 없이도 네트워크가 유지되는 것을 목표로 한다. 이를 실현하기 위해, 해시 함수, 합의 알고리즘, 디지털 서명 등 다양한 기술을 활용한다. 분산 원장 기술의 주요 특징으로 불변성이 있다. 분산 원장에 저장된 데이터는 수정 및 삭제가 불가능해 데이터 무결성을 보장한다. 분산 원장 기술의 다른 특징은 탈중앙화 구조이다. 분산된 노드에 저장되기 때문에 단일 장애 지점이 제거되어, 특정 노드에 장애가 발생해도 시스템 전체에 지장이 없다. 주요 기능으로는 스마트 컨트랙트(Smart Contract)가 있다. 스마트 컨트랙트란 제3자 없이도 특정 조건이 성립하면 자동으로 거래가 진행되는 기술이다. 분산 원장 기술에 한 형태로는 블록체인(Blockchain)이 있다. 블록체인은 데이터를 블록 단위로 저장하며, 각 블록은 이전 블록의 해시 값을 포함하여 체인 형태로 연결된다. 이 외에도 DAG(Directed Acyclic Graph)와 같이 블록체인과 다른 구조를 갖는 분산 원장 기술도 존재한다.

IPFS는 분산 파일 저장 시스템이다. 사용자가 파일을 저장하면, 해당 파일의 내용으로부터 생성된 고유한 해시 값인 CID(Content IDentifier)가 생성된다. IPFS의 가장 큰 특징은 위치 기반 주소가 아닌 CID를 이용한 컨텐츠 기반 주소를 이용하는 점이다. CID만 있으면 파일의 위치에 무관하게 해당 데이터를 검색할 수 있다. IPFS는 중복 제거(Deduplication) 기능을 제공하며, 파일 변경 시 새로운 CID가 생성되어 버전 관리가 가능하다[6].

DID는 W3C(World Wide Web Consortium)에서 제안한 새로운 유형의 식별자이다. DID는 신뢰 가능한 탈중앙화 저장 공간이 필요로 하는 기술로 블록체인과 같은 분산 원장 기술을 이용한다. 주요 특징으로는 자신의 신원 정보를 스스로 생성하고 제어할 수 있다는 점이다. DID는 그림 1과 같이 구성되며, “did”는 DID 체계를 따른다는 표시이고, “example”은 DID 메서드이다. “123456789abcdefghi”는 해당 DID 메서드 내의 고유한 식별자이다. DID가 생성될 때 그림 2와 같은 DID 문서를 함께 생성한다. DID 문서는 DID를 설명하는 JSON 형태의 파일로, DID 소유자의 인증 방식, 공개키 등을 저장한다[7].

VC(Verifiable Credential)은 W3C에서 제시한 디지털 자격 증명 방식이다. 사용자는 기관, 학교, 회사 등으로부터 VC를 발급받아 소유한다. 자신의 신원을 증명할 때, VC 중 필요한 부분만 선택하여 조합해 VP(Verifiable Presentation)를 생성한다. 사용자는 VP를 제출함으로써 필요 이상의 개인정보를 노출하지 않고 신원 증명이 가능하다[8].

Figure 1. Example of DID


```
{
  "didDocumentMetadata": {},
  "didResolutionMetadata": { "contentType": "application/
  "didDocument": {
    "id": "did:ethr:0x02380544d78bded54485cc58f709e4bfc79
    "verificationMethod": [],
    "authentication": [ ···
    ],
    "assertionMethod": [ ···
    ],
    "@context": [
      "https://www.w3.org/ns/did/v1",
      "https://w3id.org/security/suites/secp256k1recovery
      "https://w3id.org/security/v3-unstable"
    ]
  }
}
```


Figure 2. Example of DID Document

Ⅲ. 관련 연구

IoT 기술의 보안성을 강화하기 위한 방안으로 분산 원장 기술인 블록체인과의 통합이 활발히 연구되고 있다. 이 두 기술을 결합하면, IoT 디바이스에서 수집된 데이터를 블록체인에 기록하여 데이터 무결성을 보장할 수 있다. 특히, 블록체인의 주요 장점인 스마트 컨트랙트를 IoT 기술에 적용할 수 있다. 이를 이용하면, 신원 인증을 통한 데이터 액세스 제어를 구현할 수 있다[9].

블록체인 기반 IoT에 관한 연구들은 주로 Ethereum 또는 허가형(Permissioned) 블록체인인 Hyperledger Fabric을 활용하였다. 허가형 블록체인은 사전 등록된 인증된 사용자만 접근할 수 있도록 제한된 구조를 갖는 블록체인이다. 대부분의 연구에서는 스마트 컨트랙트를 활용하여 신원 인증을 구현하고, 이를 기반으로 데이터 액세스 제어를 수행하였다[10, 11]. 이외에 다른 방법으로, DID와 VC/VP를 활용한 신원 인증 기반 데이터 액세스 제어도 제안되었다. 해당 방식에서는 IoT 디바이스에서 수집된 데이터를 액세스 하기 위해서 DID를 이용한 검증 절차를 통해 액세스 권한을 부여하는 구조를 제안하였다[12, 13]. 그러나 이러한 연구는 여전히 블록체인이 가진 고유한 한계인 확장성 부족과 대용량 데이터 처리의 어려움을 극복하지 못하고 있다.

블록체인의 확장성 한계와 대용량 데이터 처리 문제를 극복하기 위한 방안으로 IPFS(InterPlanetary File System)를 적용한 연구도 진행되었다. 수집된 데이터를 직접 블록체인에 저장하는 대신, IPFS에 저장하고 생성된 CID(Content Identifier)를 블록체인에 기록하는 방식을 통해 확장성 문제와 대용량 데이터 저장의 부담을 완화하는 방식이다[14, 15]. 특히 일부 연구에서는 스마트 컨트랙트를 활용하여 데이터 액세스 제어를 고려하였다. 하지만 전반적으로 신원 인증을 통한 데이터 액세스 제어를 충분히 반영하지 않았다. 이로 인해 중간자 공격 등에 취약하다는 한계가 존재한다.

기존 연구에서는 블록체인 기반 IoT 시스템을 제안하거나, 블록체인의 한계를 극복하기 위해 IPFS 등을 추가했다. 혹은 스마트 컨트랙트 또는 블록체인 기반 DID를 적용하여 신원 인증 등을 넣는 연구도 진행되었다. 그러나 분산 원장 기술, IPFS 그리고 DID를 전부 적용한 연구는 찾아본 바 진행되지 않았다. 본 논문은 위의 세 가지를 결합한 시스템을 제안한다.

Ⅳ. 시스템 설계

본 논문에서 제시하는 시스템은 IOTA(Internet of Things Application)를 기반으로 한다. IOTA는 IoT 디바이스에서 원활하게 사용할 수 있도록 설계된 분산 원장 기술이며, Tangle이라 불리는 DAG를 사용한다. IOTA 네트워크의 참여자는 트랜잭션을 생성할 때, 이전에 발생한 임의의 트랜잭션 두 개를 선택하여 이를 승인한다. 이러한 구조는 거래자와 채굴자의 역할을 모든 참여자가 동시에 수행하므로, 별도의 수수료가 없다는 큰 장점이 있다. 이는 소규모 트랜잭션(Micro-Transaction)이 자주 발생하는 IoT 기술 환경에서 적합하다[16].

MQTT(Message Queueing Telemetry Transport)는 Publish-Subscribe 기반의 경량 메시징 프로토콜이다. 제한된 대역폭과 저전력인 환경에서도 안정적으로 데이터를 전송할 수 있다. MQTT의 구조는 데이터를 받는 Subscriber와 데이터를 보내는 Publisher 그리고 이 둘을 연결해주는 Broker로 구성한다[17].

본 논문에서 제시하는 시스템 설계는 그림 3과 같다. IoT 디바이스 소유자(IoT Device Owner)는 IoT 디바이스를 소유하고 관리하는 사람 혹은 기관이다. 수집된 데이터를 안전하게 저장 및 관리하며, MQTT에서 Broker 역할을 수행한다. IoT 디바이스 소유자는 IoT 디바이스 대신 암호화 알고리즘과 VP 검증 진행한다. User는 IoT 디바이스에서 수집된 데이터를 사용하는 사람 혹은 기관이다. DID와 VC/VP를 이용하여 데이터 액세스 권한을 IoT 디바이스 소유자로부터 얻는다. MQTT 프로토콜에서 Subscriber의 역할이다.

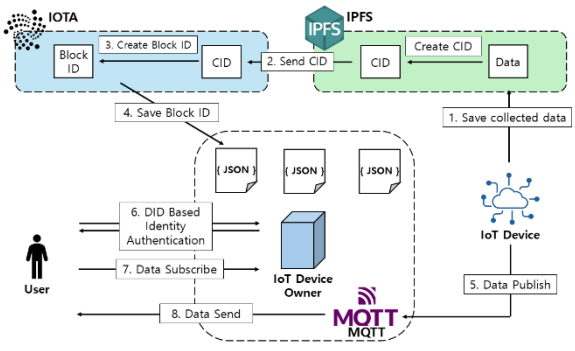

Figure 3. System Overview

제시하는 시스템은 Init-Phase와 Main-Phase, 총 두 단계로 나뉜다. Init-Phase에서는 IoT 디바이스 소유자가 IoT 디바이스에서 수집된 데이터를 저장하는 과정이다. Main-Phase에서는 User의 DID와 VC/VP를 이용한 신원 인증과 MQTT를 이용한 데이터 전송 과정이다.

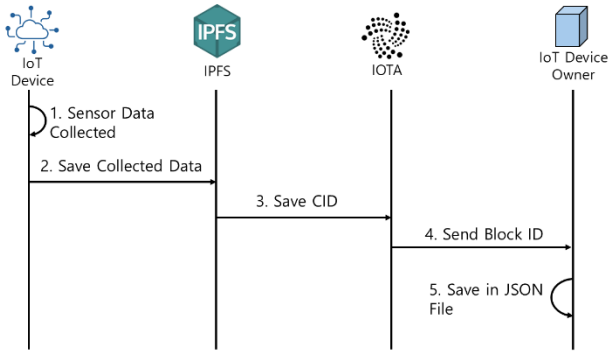

Figure 4. Init-Phase

그림 4는 Init-Phase의 흐름도이다. IoT 디바이스에서 실시간으로 수집된 데이터는 IoT 디바이스의 로컬에 저장된다. 특정 시간이 지날 때마다 IPFS에 업로드하여 CID를 얻어낸다. IoT 디바이스의 로컬에 저장된 데이터는 삭제하여, 작은 저장용량을 가진 IoT 디바이스에서도 적합하도록 설계했다. 이후 CID를 IOTA에 등록하여 Block ID를 얻어온다. 자신의 Device ID와 Block ID를 IoT 디바이스 소유자에게 보낸다. IoT 디바이스 소유자가 이를 받으면 Device ID, Block ID 그리고 현재 시간을 그림 5와 같은 형태로 JSON 파일로 저장한다.

Figure 5. Stored in JSON File

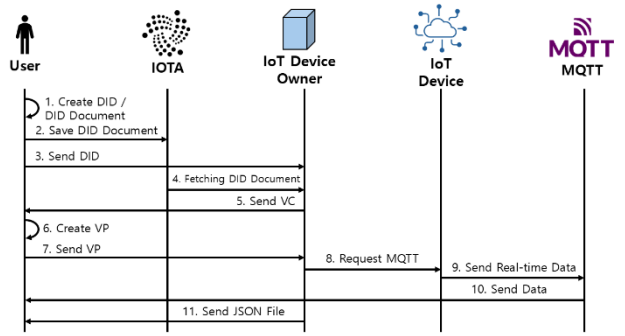

Figure 6. Main-Phase

Main-Phase는 그림 6과 같이 진행된다. 제일 먼저 User는 자신의 DID와 DID 문서를 생성한 후, DID 문서를 IOTA에 저장한다. 이후 DID를 IoT 디바이스 소유자에게 전송한다. IoT 디바이스 소유자는 User의 DID 문서를 참고하여, VC를 생성해 발급한다. User가 IoT 디바이스에서 수집된 데이터가 필요하게 되면 발급받은 VC를 가공하여 VP를 생성한다. 이후에 MQTT Subscriber 메시지와 VP를 IoT 디바이스 소유자에게 전송한다. IoT 디바이스 소유자는 VP 검증으로 신원 인증이 완료되면, 이전에 수집한 데이터를 저장한 JSON 파일을 보낸다. 실시간 데이터를 얻어오기 위해 IoT 디바이스 소유자는 Broker가 되어 IoT 디바이스에게 Publish를 요청한다. IoT 디바이스는

Publish 요청을 받으면 Broker에게 데이터를 보낸다. 이후 Broker는 데이터를 요청한 User에게 전송한다. 이 과정을 통해 IoT에서 중요한 실시간성을 해결할 수 있다.

Ⅴ. 결론 및 향후 연구

본 논문에서는 IOTA, IPFS, DID, 그리고 VC/VP 인증을 통합한 시스템을 제안하였다. 본 시스템은 IoT 디바이스가 수집한 데이터를 IOTA에 저장하여 데이터 무결성을 보장하며, 분산 원장 기술의 한계인 확장성과 대용량 데이터 처리 문제를 해결하기 위해 IPFS를 활용한다. 또한, DID와 VC/VP를 이용한 신원 인증을 통해 데이터 액세스 제어를 가능하게 한다.

본 시스템에서 IoT 디바이스가 수집한 데이터는 주기적으로 IPFS에 저장되며, 생성된 CID는 IOTA Tangle에 기록된다. 이후 IoT 디바이스의 로컬 저장소에서는 해당 데이터를 삭제하여 저장 공간을 효율적으로 관리한다. 또한, 생성된 Block ID는 IoT 디바이스 소유자에게 전송되며, 이는 JSON 파일로 저장된다. User가 DID 기반 인증을 완료하면, IoT 디바이스 소유자는 사전에 저장된 JSON 파일과 함께 MQTT 프로토콜을 이용하여 실시간 데이터를 전송한다. 이를 통해 IoT 환경에서 중요한 요소인 실시간성을 충족할 수 있다.

향후 연구에서는 본 논문에서 제시한 시스템이 실시간 데이터 전송 과정에서 데이터 무결성을 완전히 보장하지 못하는 한계를 해결하는 방안을 연구할 계획이다. 또한, 현재 시스템에서는 IPFS에 저장된 데이터가 암호화되지 않은 상태로 보관되므로, 데이터 보안 강화를 위해 암호화 기법을 적용할 계획이다. 더 나아가, 성능이 제한된 저사양 IoT 디바이스에서도 보안성과 성능을 동시에 만족할 수 있는 방안을 연구할 예정이다.

ACKNOWLEDGMENT

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 지역지능화혁신인재양성사업임(IITP-2024-RS-2023 00260267)

참 고 문 헌

[1] De Aguiar, E. J., Faical, B. S., Krishnamachari, B., Ueyama, J. "A Survey of blockchain-based strategies for healthcare," *ACM Computing Surveys*, Vol. 53, No. 2, pp. 1–27, March. 2020.

[2] Samy, S., Azab, M., Rizk, M., "Towards a secured blockchain-based smart grid," *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC),* NV, USA, pp. 1066–1069, January. 2021.

[3] 박세현, 신인호, 양지원, "국방기술 블록체인 기술의 국방분야 적용 사례와 시사점," *국방과 기술*, Vol. 54, pp. 104–111, March. 2024.

[4] Krichen, M., Ammi, M., Almutiq, M., "Blockchain for modern applications: A survey," *Sensors 2022,* Vol. 22, No. 14, 5274, July. 2022.

[5] "CVE Records for IoT Vulnerabilities," MITRE, [Online]. https://cve.mitre.org/cgi-bin/cvekey.cgi?Keyword=IoT. Accessed: March. 20, 2025.

[6] Benet, J., "IPFS-content addressed, versioned, P2P file system," *arXiv preprint arXiv: 1407.3561*, pp. 1–11, July. 2014.

[7] "Decentralized identifiers (DIDs) v1.0," W3C, July 2022. [Online]. Available: http://www.w3c.org/TR/did-core/. Accessed: March. 19, 2025.

[8] "Verifiable credentials data model v2.0," W3C, Sep. 2024. [Online]. Available: https://www.w3.org/TR/vc-data-model-2.0/. Accessed: March. 20, 2025.

[9] Reyna, A., Martín, C., Chen, J., Soler, E., Díaz, M., "On blockchain and its integration with IoT. Challenges and opportunities," *Future Generation Computer Systems*, Vol. 88, pp. 173–190, November. 2018.

[10] Honar Pajooh, H., Rashid, M., Alam, F., Demidenko, S., "Hyperledger fabric blockchain for securing the edge internet of things," *Sensors 2021,* Vol. 21, No. 2, 359, January. 2021.

[11] Liu, H., Han, D., Li, D., "Fabric-IoT: A Blockchain-based access control system in IoT," *IEEE Access*, Vol. 8, pp. 18207–18218, January. 2020.

[12] Yin, J., "SmartDID: A novel privacy-preserving identity based on blockchain for IoT," *IEEE Internet of Things Journal*, Vol. 10, No. 8, pp. 6718–6732, January. 2022.

[13] Hong, S., "A research on building a smart city model based on DID (Decentralized-Identity) using digital twin," *Advanced Industrial SCIence* 4.1, Vol. 4, No. 1, pp. 34–40, January. 2025.

[14] Sangeeta, N., Nam, S. Y., "Blockchain and interplanetary file system (IPFS)-based data storage system for vehicular networks with keyword search capability," *Electronics 2023*, Vol. 12, No. 7, 1545, March. 2023.

[15] Azbeg, K., Ouchetto, O., Andaloussi, S. J., "BlockMedCare: A healthcare system based on IoT, blockchain and IPFS for data management security," *Egyptian informatics journal*, Vol. 23, No. 2, pp. 329–324, June. 2022.

[16] Popov, S., "The tangle," Oct. 2017. [Online]. Available: https://iota.org/IOTA Whitepaper.pdf Accessed: March. 12, 2025.

[17] Yassein, M. B., Shatnawi, M. Q., Aljwarneh, S., Al-Hatmi, R., "Internet of Things: Survey and open issues of MQTT protocol," *2017 international conference on engineering & MIS (ICEMIS)*, Monastir, Tunisia, pp. 1–6, May. 2017.

GossipSub Protocol 기반 P2P 네트워크에서 통신 지연시간이 브로드캐스트 효율성에 미치는 영향 분석

이성욱, 김형엽, 김승민, 주홍택*
계명대학교, 계명대학교, 계명대학교, *계명대학교
limos1479@gmail.com, ing1821@daum.net, ak47001@naver.com, *juht@kmu.ac.kr

Analysis of the Effects of Communication Delay on Broadcast Efficiency in a P2P Network Based on GossipSub Protocol

Lee Sung-wook, Kim Hyung-yeop, Kim Seung-min, Ju Hong-taek*
Keimyung Univ., Keimyung Univ., Keimyung Univ., *Keimyung Univ.

요 약

본 논문은 Ethereum 2.0, IPFS, Filecoin 등 다양한 P2P 응용에서 활용되는 GossipSub 프로토콜 기반의 P2P 네트워크에서, 물리적인 통신 지연이 브로드캐스트 효율성에 미치는 영향을 분석하였다. 기존 연구들이 주로 브로드캐스트 프로토콜의 설계에 집중한 반면, 본 연구는 통신 지연과 같은 물리적인 네트워크 성능이 GossipSub의 동작에 어떤 영향을 미치는지를 규명하는 데 초점을 두었다. 실험에서는 노드의 송신 지연시간(Outgoing Delay)만을 변수로 설정하고, 브로드캐스트 효율성 지표로 최초 메시지 수신 시각(FRT)과 중복 메시지 수신 수(DRC)를 활용하였다. K-P2PLab 테스트베드를 활용하여 최대 1,000개의 GossipSub 노드를 다양한 지연설정 하에 배치하고 시뮬레이션을 수행한 결과, 통신 지연과 관계없이 FRT는 항상 일정하며, 통신 지연이 증가할수록 DRC는 증가하다가 특정 수치를 기점으로 다시 감소하는 경향을 보였다. 이러한 결과는 Peer Scoring 기능이 통신 지연이라는 물리적 특성과 상호작용하여 네트워크 구성과 전파 효율성에 실질적인 영향을 미친다는 사실을 보여준다.

Ⅰ. 서 론

블록체인은 P2P 네트워크에 기반을 두고 있고, 현재 블록체인의 낮은 TPS는 해결해야 할 주요한 문제이다. [1] 이를 해결하는 방법 중 하나는 블록체인 네트워크의 P2P 브로드캐스트 성능 향상이 있다. 브로드캐스트 성능이 향상되면 트랜잭션과 블록이 더 빠르게 전체 네트워크로 전파될 수 있고, 이는 블록 생성시간을 줄여서 TPS를 증가시킬 수 있다. 이 방법은 네트워크가 점차 확장되더라도 탈중앙화와 보안에 영향을 끼치지 않고 높은 TPS를 확보할 수 있는 방법이다.

P2P에서 피어 간의 전송계층 연결 위에서 메시지를 전파하는 방법에 따라 Tree 기반, Gossip 기반 브로드캐스트 등 다양한 방법이 존재한다. J. Leitao 등은 Eager Push와 Lazy Push를 혼합하여 모든 노드가 가장 빠른 연결 하나만을 유지하는 기법을 통해 일종의 Spanning Tree를 형성하는 Plum Tree 기법을 제안하였고, 자신이 알고 있는 노드 중 일부를 선택해 전파하여, 플러딩보다 중복 메시지의 양을 완화하는 Gossip 기반 브로드캐스트에 관해서도 연구하였다. [2-3] 그 외에도 Eager Push, Lazy Push, Pull 방식들을 혼합하여 브로드캐스트 효율을 개선하기 위한 다양한 형태의 아이디어가 제시되었다. [4]

P2P 브로드캐스트 성능 향상을 위한 다양한 연구가 진행되고 있으나 물리적인 통신 성능과의 연관 관계에 대한 분석은 브로드캐스트 프로토콜 자체의 설계에 관한 연구에 비해 활발히 이루어지지 않고 있다. 전송계층의 통신 성능은 물리적인 네트워크 연결에 영향을 받을 수밖에 없으며, 이는 자연스레 전송계층 상위의 P2P 브로드캐스트 성능에 영향을 줄 수밖에 없다.

물리적인 통신 성능과 P2P 브로드캐스트 성능 사이의 관계가 정확히 규명되어야만 P2P 브로드캐스트 성능 향상을 위한 연구 결과에 대한 실질적인 효과를 수치로 측정할 수 있다. 예를 들어, GossipSub 프로토콜에서 피어를 평가하여 이를 메시지 전달을 위한 브로드캐스트 방법에 적용하는 Scoring Function이라는 방식을 도입하였지만, 점수를 매기는 각각의 매개변수에 가해지는 가중치를 어떻게 조정하는 것이 좋을지는 언급되지 않았다.

우리는 Ethereum 2.0[5], IPFS[6], Filecoin[7] 등 많은 P2P 응용에서 사용되는 GossipSub P2P 브로드캐스트 프로토콜에 대해, 노드의 물리적인 통신 성능이 브로드캐스트 효율성에 어떤 영향을 미치는지 분석하였다. 이를 위해 본 논문에서는 통신의 여러 성능 지표 중 가장 대표적인 노드의 송신 지연시간(Outgoing Delay)만을 고려하였다. 그 외에도 여러 가지 물리적 통신 성능 중 패킷의 Loss 및 Corrupt 등을 고려할 수 있겠지만, 이는 대부분 TCP 단위에서 보완되는 문제이기 때문에 고려하지 않았다. 또한, 브로드캐스트 효율성을 측정하는 지표로 Publish 메시지의 최초 송신 시각(이하 FRT; First message Receiving Time)과 단일 Publish 메시지에 대한 중복 수신량(이하 DRC; Duplicated message Receiving Count)을 고려하였다.

실험은 가장 많이 사용되는 P2P 라이브러리인 Go-libp2p의 GossipSub을 사용하여 구현한 노드를 이용하였고[8-9], P2P 성능 측정을 위한 테스트 플랫폼으로 K-P2PLab[10]을 활용하였다. K-P2PLab에서 Docker Swarm 기반의 노드를 최대 1000개까지 생성하였으며, 다양한 네트워크 성능을 가진 노드로 P2P 네트워크를 구축하고 브로드캐스트 성능을 측정 후 분석하였다.

결론적으로, 노드의 통신 지연 성능이 다르더라도 FRT는 비슷한 시간으로 측정되었고, 노드의 네트워크 지연이 증가함에 따라 DRC는 증가하다 일정 값을 기준으로 다시 감소하였다.

본 논문의 기여는 최초로 물리적인 통신 연결이 GossipSub 기반의 P2P 브로드캐스트 효율성에 어떤 영향을 미치는지 분석한 결과이다. 구체적으로는 한 노드의 네트워크 성능이 GossipSub 기반의 프로토콜에서 다양한 파라미터들과 방법들에 어떤 영향을 미치는지 규명한 것이다.

이어지는 2장에서는 관련 연구들과 GossipSub에 대하여 소개하며, 3장에서는 실험 환경 설정 및 실험 방법에 대해 소개할 것이다. 그리고 마지

막 4장에서는 결과와 분석을 제시하고 5장에서 결론으로 마무리한다.

Ⅱ. 관련 연구

GossipSub 프로토콜은 D. Vyzovitis 등이 제안 및 개발한 Gossip 기반의 PubSub 멀티캐스트 프로토콜이다. [11] 각 노드들은 물리적인 네트워크 연결 위에 전송 프로토콜을 사용해서 상호 연결하여 피어 관계를 형성하고 이런 피어 관계로 P2P 네트워크를 형성한다(GossipSub에서는 해당 네트워크를 Metadata 연결이라는 용어를 사용). GossipSub은 제어 메시지를 통해 피어 관계를 유지하거나 해제한다.

GossipSub은 각 노드가 처음 브로드캐스트를 시작하거나 새로운 메시지를 수신한 노드가 이웃한 피어 노드에게 2가지 방법으로 브로드캐스트 메시지를 전달한다. 하나는 Eager Push이고 다른 하나는 Lazy Pull이다. Eager Push는 이웃 노드에서 새로운 메시지 전체를 곧바로 전달하는 방법이다. 이 방법은 이웃 노드가 이미 이 메시지를 수신하고 있는지와 관계없이 무조건 전달하는 방법이다. Lazy Pull은 이웃 노드에게 새로 수신한 메시지가 있음을 먼저 알리고(I_HAVE 메시지), 피어가 해당 메시지를 아직 수신하지 않은 경우, 해당 메시지를 요청하게 되고(I_WANT 메시지) 그때 전체 메시지를 전달하는 방식이다. 또한, 처음 메시지를 수신하면 이웃 노드들에게 수신 사실을 알려서 더이상 메시지를 보내지 말라고 알리고 있다(I_DONT_WANT 메시지). 이 메시지로 Eager Push나 Lazy Pull을 하지 않도록 막는다.

Eager Push는 메시지 전달을 빨리할 수 있지만, 반면 중복 메시지가 발생할 수 있고 Lazy Pull은 메시지 전달이 늦어지나 중복 메시지 발생이 줄어든다. 기본적으로 노드는 브로드캐스트 메시지를 수신할 때 이웃하는 P2P 피어 중 일부에게 Eager Push를 한다. Eager Push로 전달하는 이웃 노드의 수 D는 GossipSub의 중요 파라미터 중 하나이며 D_{high}, D_{low} 를 사용해서 Eager Push 하는 D 값을 특정 구간에 있도록 조정한다. 이후 매 하트비트마다 Eager Push로 메시지를 전달하는 노드를 포함한 자신과 연결된 모든 노드 중 Gossip Factor만큼 피어를 선택해서 추가로 Lazy Pull 전파를 시도한다.

어떤 피어에게 Eager Push를 할 것인가 Lazy Pull을 할 것인가에 관한 결정은 다음에 설명할 피어의 점수에 의존하고 있다. Lazy Pull 메시지를 전달하던 피어를 Eager Push로 전달하는 피어로 바꾸는 것은 상호 피어간 협의를 통해서(GRAFT 메시지) 연결을 바꾸게 된다(GossipSub 규격에서는 이를 업그레이드로 표현). 반대로 Lazy Pull로 전달하던 피어를 Eager Push 전달 피어로 바꿀 수도 있다(PRUNE 메시지).

Eager Push를 통해 브로드캐스트 메시지는 전체 P2P 네트워크에 빠르게 전파되며, 그렇기에 Eager push의 효율이 브로드캐스트 성능에 큰 영향을 미치므로 노드는 Eager Push를 보낼 대상 노드를 잘 선택하는 것이 매우 중요하다. Lazy Pull은 메시지 전파는 빠르지 않지만, 중복 메시지 수가 줄어들게 되며 메시지 수신을 하지 못하게 되는 노드를 줄이는 효과가 있다.

GossipSub 1.1.0 이상에서는 피어를 평가하는 점수(Scoring) 방법을 도입해 각 노드가 다른 연결된 모든 노드에 대한 평가를 실시간으로 진행하고, 연결 업그레이드 대상을 임의의 노드 대신 자체적인 각 노드에 대한 평가를 바탕으로 점수가 높은 노드와 먼저 업그레이드를 시도하게 된다.

현재 GossipSub에서 점수를 측정하는 지표는 아래와 같다. [12]

- P_1: **Time in Mesh**: 해당 노드가 토픽에 머무른 시간으로, 낮은 양의 가중치와 곱해진다.
- P_2: **First Message Deliveries**: 해당 노드가 브로드캐스트 메시지를 자신에게 최초로 전송한 횟수로, 양의 가중치와 곱해진다.
- P_{3a}: **Mesh Message Delivery Rate**: 해당 노드가 토픽 내에서 자신에게 메시지를 전송한 횟수로, 기댓값보다 크다면 0, 작다면 그 차이만큼의 제곱이 된다. 이는 음의 가중치와 곱해져 전송이 낮은 노드에 벌점을 부여하는 역할을 한다.
- P_{3b}: **Mesh Message Delivery Failures**: 해당 노드가 브로드캐스트 메시지를 전송하는 데 실패한 횟수로, P_{3a}에 의해 낮은 점수를 받아 정리된 피어가 다시 빠르게 네트워크에 합류하지 못하도록 설정되었다. 음의 가중치와 곱해져 역시 벌점을 부여하는 역할을 한다.
- P_4: **Invalid Messages**: 해당 노드가 잘못된 메시지를 전송한 횟수로, 음의 가중치와 곱해진다.
- P_5: **Application-Specific Score**: 개발자가 정의 가능한 애플리케이션 레벨 점수다.
- P_6: **IP Collocation Factor**: 해당 노드와 같은 IP가 네트워크 내에 기댓값보다 적게 있다면 0, 많다면 차이의 제곱이 된다. Sybil 공격을

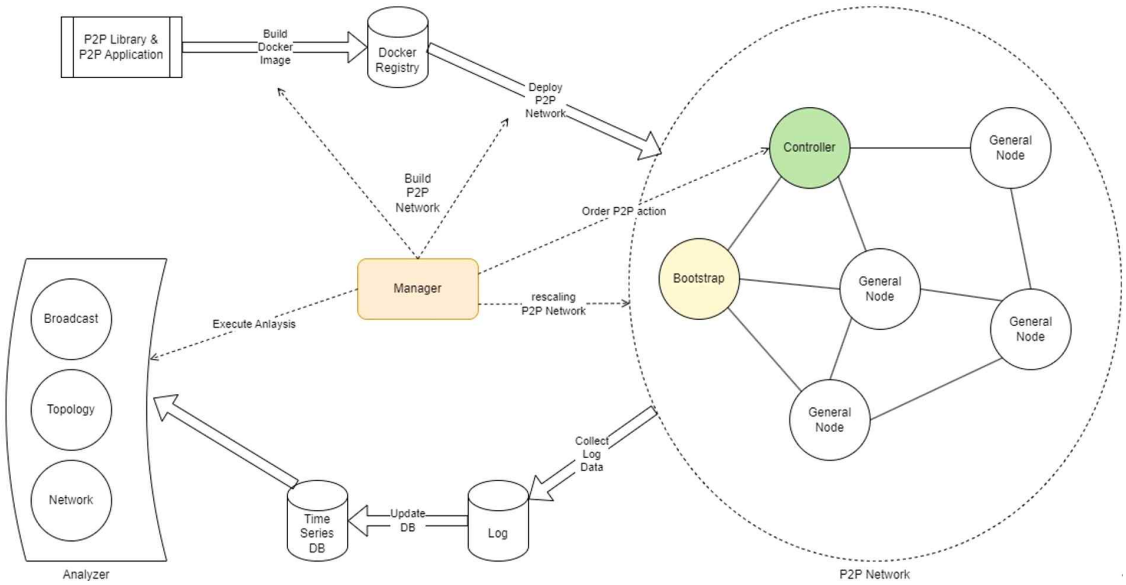

그림 1) K-P2PLab 구조도

진행하기 어렵게 만들기 위한 수치로, 음의 가중치와 곱해진다.

P_7: **Behavioural Penalty**: 해당 노드의 잘못된 행동에 대해 증가하는 Counter로, 제곱하여 음의 가중치와 곱해진다.

이중 P_1부터 P_4까지는 Subscribe 한 토픽 단위로 별도의 매개변수가 지정되고, 나머지는 노드 단위로 매개변수가 지정된다. GossipSub은 토픽별로 피어의 구독에 따라 앞서 설명한 브로드캐스트를 별도로 수행하고 있다. 이러한 매개변수들을 조합하여 수식 1)과 같이 노드의 점수를 평가하며, 이 점수는 앞서 말한 것과 같이 Eager Push를 보낼 다른 노드를 선택하는 지표로 사용하게 된다. 이때 TopicCap은 Topic마다 지정할 수 있는 점수의 상한이다.

$$S = TopicCap(\sum_{i=1}^{4}(w_i P_i)) + w_5 P_5 + w_6 P_6 + w_7 P_7$$
S : 피어의 점수
w_i : i 파라미터에 대한 가중치
P_i : i 파라미터에 대한 점수

수식 1) GossipSub의 Scoring 함수

앞서 언급한 연구에서는 위 매개변수를 기반으로 한 Scoring 시스템을 구축하고 해당 프로토콜이 적용된 네트워크가 Sybil 공격을 비롯한 다양한 공격이 이루어지는 불안정한 네트워크에서 스스로 수복 가능함과 그 퍼포먼스 등을 측정하여 보였다. 이외에도, F. S. De Cristo 등은 최근 Eager Push 네트워크의 다양한 요소를 매개변수로 네트워크를 분석하는 시도의 연구를 진행하였다. [13] 그럼에도 불구하고, 네트워크가 공격받지 않는 정상적인 상황에서의 Scoring 함수가 네트워크 전반에 어떤 영향을 미치는지에 관한 논의는 부족하였다. 안정적인 네트워크의 특성을 분석함으로써 이를 토대로 불안정한 네트워크의 특성을 분석하는 데 기여할 수 있다. 그러므로 우리는 이러한 정상적인 환경에서 Scoring 함수가 미치는 영향에 대해 분석할 것이다.

Ⅲ. 실험 환경

3.1. K-P2PLab

우리는 실험을 위해 K-P2PLab을 활용하여 테스트 환경을 구축하였다. K-P2PLab은 P2P 노드를 Docker 컨테이너로 만들고 노드 간의 연결을 Docker Swarm으로 만든다. 이렇게 Docker 컨테이너와 Docker Swarm으로 P2P 네트워크를 구성하고 각 노드에서 TC 명령어로 물리적 연결의 지연시간을 설정하였다. 이렇게 함으로써 실제 네트워크와 유사한 환경을 같은 라이브러리를 사용하여 모방하면서도, 실제 네트워크에서 환경을 구축하고 실험하는 것보다 간단히 실험 환경을 구축할 수 있었다.

K-P2PLab의 하드웨어 환경은 전체 실험 환경을 관리하는 1개의 매니저 서버와 P2P 노드가 동작하는 확장 가능한 1대 이상의 러너 서버로 구성된다. 우리는 총 10대의 Dell-PowerEdge-R730 서버를 사용하였고, 이는 각각 2개의 Intel Xeon E5-2620 v3 @ 2.40GHz CPU와 32 GiB의 메모리를 갖고 있다. 서버 간 Docker Swarm Network를 구성한 뒤 P2P 노드를 본 하드웨어 구성에서 실행 가능한 최대 1000개까지 노드를 생성하며 실험을 진행하였다.

각 노드는 로그 데이터를 로컬 파일에 기록하고, 기록된 로그 파일은 마운트 되어 실제 물리적 서버에서 한 폴더에 모인다. 서버의 마운트 된 폴더는 다시 매니저 서버의 한 폴더에 마운트 되어 결과적으로 모든 로그가 매니저 서버에 모이게 된다. 이러한 구조를 사용하여 실험이 끝나지 않아도 실시간으로 로그를 파일의 형태로 확인해볼 수 있으며, 이후 데이터 가공을 위해서 로그를 따로 이동할 필요 없이 즉시 처리가 가능하다.

그림 1)에서 사용자는 매니저(Manager)를 사용하여 P2P 라이브러리와 애플리케이션을 Build 하여 사용자 Docker 레지스트리에 저장하고, 이를 실험용 서버에 설치하여 P2P 네트워크를 만든다. P2P 네트워크상의 Controller 노드를 통해 P2P 네트워크의 특정 노드에게 메시지 브로드캐스트 시작과 같은 명령을 전달할 수 있으며, Docker 자체 기능을 활용해 각 설정 별 노드 그룹의 규모를 Rescaling 하는 것 역시 가능하다.

K-P2PLab에는 GossipSub의 모든 이벤트를 추적 가능한 Event Tracer가 포함되어 있으며, 이를 통해 노드의 최초 메시지 수신 시각, 중복 메시지 수신 횟수 등을 수집할 수 있으며, 이 외에도 노드 간의 RPC 메시지를 분석하여 GRAFT와 PRUNE 메시지를 포함한 노드 간의 제어 메시지 역시 확인할 수 있다. 따라서, 이를 통해 모든 노드의 FRT 및 DRC를 확인할 수 있다.

3.2. 실험 순서

K-P2PLab에서 한 Docker 컨테이너당 하나의 P2P 애플리케이션이 동작하도록 만들었다. 각 노드는 Kademlia를 P2P 형성에 사용하여 P2P의 네트워크를 형성하고 그 위에 GossipSub 메시지 브로드캐스트가 동작한다. Kademlia를 사용하므로 각 노드가 네트워크에 진입하기 위해 하나 이상의 Bootstrap 노드가 필요하고, 각 노드에서 GossipSub Publish를 명령하기 위해 Controller 노드 역시 하나가 필요하다. Bootstrap 노드와 제어 노드는 GossipSub에 합류하지 않으므로 브로드캐스트에 전혀 기여하지 않는다.

브로드캐스트 실험을 위해 네트워크 지연이 10, 20, 30, … 1990, 2000ms까지 차례로 10ms씩 차이가 나는 200개의 GossipSub 노드 설정 파일을 제작하여 사용하였다. 노드들의 네트워크 전파 지연을 비롯한 설정은 JSON 파일로 조작 가능하며, 이는 노드가 실행될 때 시스템 매개변수로 입력받게 된다.

네트워크 지연 값에 대해, 설정 가능한 네트워크 지연은 최대 2000ms로, 이를 초과할 시 노드가 P2P 네트워크와의 연결 설정을 완료하지 못해 아예 TCP를 연결할 수조차 없으며, 이에 근접한 값 역시 원활한 통신이 불가능하다. 또한, 각 통신 지연설정의 간격은 10ms로 설정하여 작은 지연시간의 차이에 대한 영향을 분석할 수 있도록 하였다. 그 외에 노드의 파라미터는 수식 2)와 같이 설정되었다. 현재 네트워크 내의 다른 Scoring 파라미터 중 네트워크에 머무른 시간은 고려하지 않으므로 P_4은 0이며, 다른 파라미터는 잘못 전송한 메시지에 대한 처벌이기 때문에 패킷 오류가 발생하지 않는 본 실험에서는 고려하지 않는다.

$$D_{low} = 4, D = 6, D_{high} = 12, GossipFactor = 0.25$$
$$w_1 = 0, w_2 = 1.0, w_{3a} = -0.5, w_{3b} = 0, w_{4...7} = 0$$

수식 2) K-P2PLab 실험에서 사용한 파라미터

처음에는 Bootstrap 노드와 Controller 노드를 생성하고, 이후 각 지연시간 설정 별 GossipSub 노드를 5개씩, 총 1000개 배포하였다. 특히 현재 분석하려는 통신 지연에 의한 점수 차이만을 명확히 측정하기 위해서는, 노드의 진입 순서 등에 의해 연결 구조가 변화하면 안 되므로 모든 노드가 병렬적으로 동시에 네트워크에 진입하도록 했다. 만약 노드의 진입 순서가 고려된다면, 본 연구의 목적과 어긋나게 된다. 1000개의 노드를 동시에 생성한 후에는 각 노드가 P2P 네트워크에서 이탈하거나 새로 참여하는 상황을 만들지 않았다. 이러한 상황은 일반적인 P2P 네트워크에서는 존재하지 않으나, 지연시간에만 메시지 전파가 영향을 받게끔 이러한 상황은 배제하였다.

네트워크에 노드가 충분히 머물러 안정된 후 Controller 노드를 통해

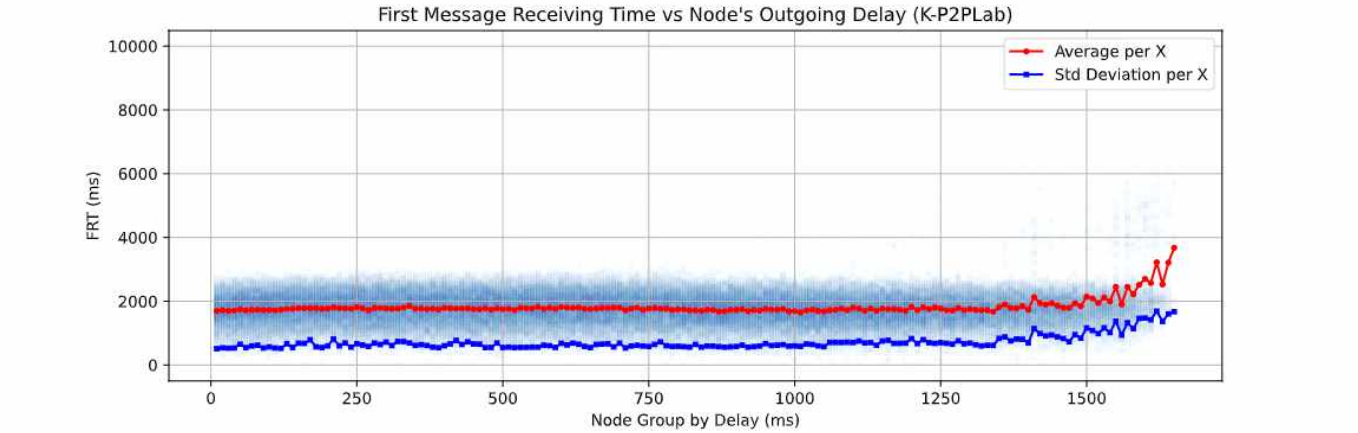

그림 2) 노드의 통신 지연 그룹(X축) 당 FRT(Y축)

1000개의 노드 중 임의의 20개의 노드에서 1KiB 크기의 무작위 메시지를 생성하고, 이를 모든 노드가 공유하는 토픽으로 해당 노드가 1회 Publish 한다. 각 노드는 자신의 로그 파일에 앞서 설명한 로그를 기록하고 이를 분석하여 결과를 얻을 수 있다.

Ⅳ. 결과 및 분석

4.1. 최초 메시지 수신 시간(FRT: First message Receiving Time)

노드의 통신 지연시간별 최초 브로드캐스트 메시지 수신까지 걸린 시간이 그림 2)이다. X축은 노드의 지연시간이고 Y축은 처음 메시지를 수신한 시간이다. 그림 2)에서 음영처럼 표시된 반투명한 각 점은 각 노드가 처음 메시지를 수신한 시간이고 노드 그룹별로 FRT에 대한 평균값과 표준 편차도 표시되어 있다.

지연시간이 1400ms 이상의 노드들은 통신 성능이 이웃하는 노드 간의 메시지 전달 또는 제어 메시지 전달에 영향을 주고 이것이 최초 메시지 전달에 영향을 주는 것으로 판단된다. 그림 2)에서 역시 FRT가 변화하는 구간이 존재함을 확인할 수 있다. 이는 이 이상의 지연시간을 갖는 노드는 메시지 수신에 불이익이 발생한다고 볼 수 있다.

반대로, 지연시간이 1400ms 이하의 노드들은 GossipSub 네트워크 내에서 각 노드의 통신 지연과 무관하게 노드들의 FRT가 같다. 이러한 결과가 나타난 이유는 통신 지연이 낮은 노드가 GossipSub 네트워크 내에서 다른 노드보다 빠르게 응답하므로 더 높은 점수를 받고, 이로 인해 다른 노드들이 해당 노드를 Eager Push 대상 피어로 선택할 가능성이 높기 때문이다. 이러한 선호도는 새로운 Eager Push 노드를 추가할 때 통신 지연이 낮은 노드들이 우선으로 선택되게 만들며, 통신 지연이 낮은 노드들 역시 Eager Push 노드를 추가할 때 통신 지연이 낮은 노드를 우선시하여 연결을 승인한다. 해당 과정이 반복되다 보면 통신 지연이 낮은 노드끼리 서로 연결되게 되어 네트워크의 중심에서 서로 연결되게 되고, 그렇지 않은 노드들은 네트워크의 외곽에서 일부 통신 지연이 낮은 노드와 연결된 형태가 된다. 이러한 노드 간의 연결이 네트워크 내에서 일종의 고속도로와 국도의 역할을 하여 패킷을 빠르게 운반하는 것을 도우며, 따라서 모든 패킷이 어디서 출발하였는지와 관계없이 중앙의 통신 지연이 낮은 노드를 통해 빠르게 네트워크의 구석구석으로 전파되고, 이후 통신 지연이 높은 노드들은 각 그래프의 외곽에서 그러한 패킷을 수신받게 된다.

그 근거로, 그림 3)에서 통신 지연이 낮은 노드의 GRAFT 메시지 수신량이 통신 지연이 높은 노드보다 많았고, PRUNE 메시지의 경우 정확히 그 반대임을 볼 수 있다. GRAFT 메시지의 수신은 다른 노드가 자신을 Eager Push 노드로 포함 시키기를 원할 때 발생하며, PRUNE 메시지의 수신은 다른 노드가 자신을 Eager Push 노드에서 제외하기를 원할 때 발생한다. 이를 통해 노드의 통신 지연이 커질수록 다른 노드가 자신을 Eager Push에 포함 시키려 하지 않는다는 것을 알 수 있으며, 이로 인해 통신 지연이 큰 노드들은 상대적으로 더 낮은 차수를 유지할 가능성이 커짐을 알 수 있다.

결론적으로, 1400ms 이하의 통신 지연을 갖는 노드들은 점수에 의한 피어 평가를 통해 적은 지연시간을 갖는 노드들끼리 서로 연결되는 형태의 네트워크를 형성하고, 이러한 형태의 네트워크가 거의 모든 노드들이 비

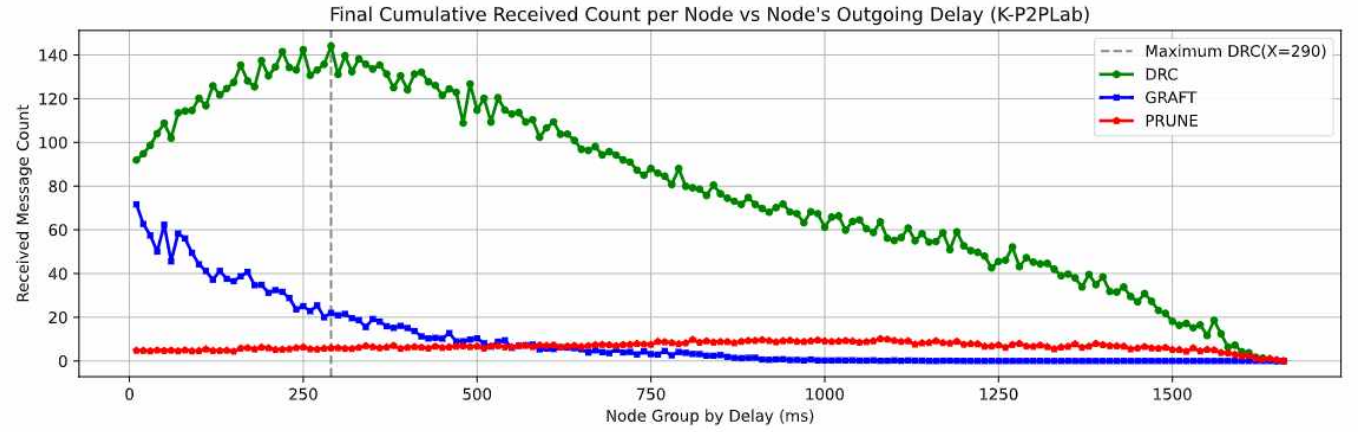

그림 3) 노드의 통신 지연 그룹(X축) 당 GRAFT, PRUNE 메시지 수신량 및 DRC(Y축)

슷한 FRT를 갖게 되는 요인이 된다.

4.2. 중복 메시지 수신 수(DRC: Duplicated message Receiving Count)

DRC는 그림 3)에서 보는 바와 같이 통신 지연이 커질수록 노드의 중복 메시지 수신 수가 늘어나는 경향을 보이다가, 지연시간 300ms 내외에서 최댓값을 갖고, 이후 다시 감소한다.

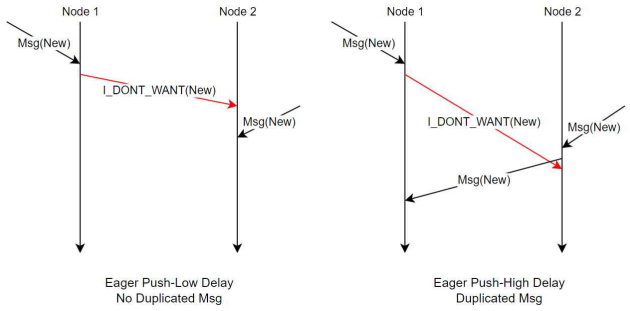

그림 4) 노드의 통신 지연에 따른 Eager Push 모델

그 원인은 다음과 같다. 통신 지연이 빠른 노드 간의 브로드캐스트는 대부분 Eager Push로 이루어지고, 느린 노드 간의 브로드캐스트는 대부분 Lazy Pull로 이루어진다. 이때 그림 4)와 같이 노드의 통신 지연이 커져 전파 속도가 느려지면 자신이 Eager Push를 받았다고 알리기(I_DONT_WANT 메시지) 이전에 다른 노드가 수신한 메시지를 릴레이하여 자신이 받을 가능성이 증가하게 된다. 따라서 노드의 통신 지연이 증가할수록 Eager Push에 의한 중복 메시지의 양이 증가하나, 이후 노드의 통신 지연이 더욱 커져 임계치를 지나게 되면 Lazy Pull이 대부분의 전파를 담당하게 되어 중복 메시지의 수가 다시 감소하게 된다.

이것과 관련하여 모든 노드의 하트비트 간격을 증가시키면, 그렇지 않을 때보다 그래프가 좌측으로 이동하게 된다. 즉, 각 노드의 통신 지연이 실제로 그대로 일 때, 오직 하트비트 간격의 변화만으로도 중복 메시지의 수신량이 변화하게 된다. 그 이유는 하트비트의 간격이 증가하면, 실제로 하트비트마다 방출되는 I_DONT_WANT 메시지의 전파가 평균적으로 지연되는 효과가 발생하게 되고, 위 가정대로 I_DONT_WANT 메시지의 전파가 지연되어 노드가 중복 메시지를 수신할 가능성이 증가한 것으로 해석할 수 있다. 이로 인해 가장 많은 DRC를 갖는 그룹의 노드 통신 지연 수치가 상대적으로 더 낮아지게 된다.

4.3. 종합

우리는 위의 실험을 바탕으로 Scoring 시스템이 정적인 P2P 네트워크 환경에서 노드의 미세한 네트워크 지연 차이를 감지하고 이를 어떻게 활용하는지 파악하였다. 또한, 네트워크의 지연은 단순히 Scoring 시스템과만 상호작용하는 것이 아니라 각종 제어 메시지의 전파와 상호작용하여 노드가 수신하는 제어 메시지의 수에 영향을 미쳐 노드가 더 많은 중복 메시지를 수신하게 할 수 있다는 것 역시 밝혀내었다.

특히 FRT와 DRC를 기반으로 한 분석을 진행하였으며, 이를 통해 Scoring을 포함한 GossipSub 네트워크 내 모든 노드에게 최초로 브로드캐스트 메시지가 도달하는 시각이 거의 유사한 양상을 보임을 밝혀내었고, 그 이유에 대한 근거로 GRAFT와 PRUNE 메시지의 발생량을 제시하였다. DRC 역시 특정 통신 지연 수치를 갖는 노드들이 가장 많은 중복 메시지를 수신함을 보이고, 그 이유를 I_DONT_WANT 메시지의 상호작용으로 서술하였다.

V. 결 론

우리는 이전에 개발한 K-P2PLab 테스트베드를 활용하여 정적인 환경에서의 노드의 네트워크 지연이 GossipSub 프로토콜(특히 Scoring)과 어떻게 상호작용하고, FRT 및 DRC 두 지표에 영향을 미치는지 측정하였다. 이후에는 위의 DRC에서 보인 특정 네트워크 지연 수치에 대한 수치적 해석 및 노드의 네트워크 지연 외에도 Error rate 및 Burst 등 더 많은 특성을 통한 해석이 필요할 것이다.

ACKNOWLEDGMENT

이 논문은 2025년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 이공분야기초연구사업임. (No. NRF-2023R1A2C2006045).

참 고 문 헌

[1] M. Conti, A. Gangwal, and M. Todero, "Blockchain trilemma solver Algorand has dilemma over undecidable messages," in Proc. 14th Int. Conf. Availability, Reliability and Security (ARES), Canterbury, U.K., Aug. 2019, pp. 1 - 10.

[2] J. Leitao, J. Pereira, and L. Rodrigues, "Epidemic broadcast trees," in Proc. 26th IEEE Int. Symp. Reliable Distrib. Syst. (SRDS), Oct. 2007, pp. 301 - 310.

[3] J. Leitão, J. Pereira, and L. Rodrigues, "Gossip-based broadcast," in Handbook of Peer-to-Peer Networking, X. Shen, H. Yu, J. Buford, and M. Akon, Eds. Boston, MA, USA: Springer, 2010, pp. 831 - 860.

[4] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," IEEE Communications Surveys & Tutorials, vol. 7, no. 1-4, pp. 72 - 93, 2005.

[5] libp2p Team, "libp2p and Ethereum," libp2p Blog, Jun. 2020. [Online]. Available: https://blog.libp2p.io/libp2p-and-ethereum/

[6] Protocol Labs, "libp2p - Publish-Subscribe," IPFS Documentation. [Online]. Available: https://docs.ipfs.tech/concepts/libp2p/#publish-subscribe [Accessed: Mar. 2025].

[7] Protocol Labs, "GossipSub," Filecoin Specification. [Online]. Available: https://spec.filecoin.io/algorithms/gossip_sub/ [Accessed: Mar. 2025].

[8] libp2p, "go-libp2p," GitHub repository, [Online]. Available: https://github.com/libp2p/go-libp2p [Accessed: Mar. 2025].

[9] libp2p, "go-libp2p-pubsub," GitHub repository, [Online]. Available: https://github.com/libp2p/go-libp2p-pubsub [Accessed: Mar. 2025].

[10] 이성욱 and 주홍택. "K-P2PLab: P2P 네트워크 토폴로지 분석을 위한 테스트베드 및 분석 플랫폼 개발" KNOM Review 27, no.2 (2024) : 40-48.

[11] D. Vyzovitis, Y. Napora, D. McCormick, D. Dias, and Y. Psaras, "GossipSub: Attack-Resilient Message Propagation in the Filecoin and ETH2.0 Networks," arXiv preprint arXiv:2007.02754, 2020.

[12] libp2p, "The Gossipsub Protocol v1.1," libp2p Specs, GitHub repository, [Online]. Available: https://github.com/libp2p/specs/blob/master/pubsub/gossipsub/gossipsub-v1.1.md [Accessed: Mar. 2025].

[13] F. S. De Cristo, J. A. Meira, J. -P. Eisenbarth and R. State, "A 9-dimensional Analysis of GossipSub over the XRP Ledger Consensus Protocol," NOMS 2024-2024 IEEE Network Operations and Management Symposium, Seoul, Korea, Republic of, 2024, pp. 1-7, doi: 10.1109/NOMS59830.2024.10575688.

개선한 휴리스틱 함수를 사용한 A* 기반 안전한 경로 계획

남승우, 유경민, 박재원, 김성현, 김명섭*

고려대학교

{nam131119@korea.ac.kr, rudals2710, 2018270614, pb1069, *tmskim}@korea.ac.kr

A* based safe path planning with improved heuristic function

Seung-woo Nam, Gyeong-Min Yu, Jae-Won Park, Ui-Jun Baek, Myung-Sup Kim*

Korea Univ.

요 약

경로 계획(Path planning)은 자율주행에서 핵심적인 기술로, 에이전트가 목적지까지 안전하고 효율적으로 이동할 수 있도록 해준다. 경로 계획을 위해 Dijkstra, A*[1], RRT와 같은 다양한 알고리즘들이 제안되어왔다. 빠른 이동을 위한 경로 길이 최소화도 중요하지만, 동시에 목적지까지의 안전한 이동을 보장하는 경로를 고려하는 것도 매우 중요하다. 급격한 회전이 포함되거나 장애물에 너무 가까운 경로는 실제 로봇 내비게이션에서는 적합하지 않을 수 있으며, 로봇이 해당 경로를 따라가기가 어려울 수도 있다. 본 연구에서는 A* 알고리즘의 휴리스틱 함수를 개선하여 곡률을 줄이고 장애물에 가까이 가지 않는 경로를 계획하는 방법을 제안한다. 새로운 곡률 비용 함수와 장애물 비용 함수를 휴리스틱에 도입함으로써, 제안된 방법은 기존의 향상된 A* 알고리즘보다 곡률이 줄어들고 장애물과의 간격이 더 확보된 경로를 생성할 수 있다.

Ⅰ. 서 론

자율 주행 기술 중 경로 계획은 에이전트가 목적지까지 이동하기 위해 필요한 경로를 계획하는 기술이다. 이 기술은 로봇, 공학, 항공 등 다양한 분야에서 사용하고 있으며, 특히 효율성, 시간 절약, 비용 절감 등의 목표를 달성하는 데 필수적인 요소로 작용한다. 경로 계획 문제는 흔히 지도 데이터, 장애물, 거리 등을 고려해 최단 거리를 찾는 문제로 정의하며, 경우에 따라 복잡한 제약 조건과 변수도 고려할 수 있다.

1.1 A* 알고리즘

경로 계획에서는 Dijkstra, A*, RRT 등 다양한 알고리즘이 제안되었다. Dijkstra 알고리즘은 탐욕적 전략을 기반으로 탐색을 진행하기 때문에 실용성을 고려하지 않고 최단 경로만 고려한다. 이는 지도를 완전히 탐색해야 하므로 계산량이 많고, 효율이 낮으며, 충돌 회피 능력이 약하다. Dijkstra 알고리즘의 높은 계산량 문제를 극복하기 위해 제안된 A* 알고리즘은 최단 경로 탐색 문제를 해결하기 위한 대표적인 휴리스틱 기반 알고리즘이다. A* 알고리즘은 시작 노드에서 목표 노드까지의 총 비용을 평가하기 위한 비용 함수 $F(n)$을 기반으로 경로를 계획한다. $F(n)$은 식 (1)과 같다.

$$F(n) = G(n) + H(n) \quad (1)$$

$G(n)$은 시작 노드에서 현재 노드 n까지 이동하는 데 소요된 실제 경로 비용으로, 탐색 과정에서 이미 지나온 경로의 누적 비용을 계산한다. $H(n)$은 현재 노드 n에서 목표 노드까지 이동하는 데 소요될 것으로 예상되는 비용을 추정하는 휴리스틱 함수이다. $H(n)$은 문제에 따라 정의하며, 경로 계획 문제에서는 목표 노드까지의 거리를 예측하기 위해 거리 측도 식을 사용한다. 대표적으로 맨해튼 거리(Manhattan Distance), 대각선 거리(Diagonal Distance), 유클리디안 거리(Euclidean Distance)가 존재한다.

이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(00230661, 하이브리드 양자키분배 방법 및 망 관리 기술 표준개발) 및 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(00235509, ICT융합 공공 서비스• 인프라의 암호화 사이버위협에 대한 네트워크 행위기반 보안관제 기술 개발)을 받아 수행된 연구임.

경로 계획 분야에서 현재까지 A* 알고리즘을 개선한 방법론들이 제안되어왔다. 휴리스틱 함수 개선[2]부터, 탐색 방법[3], 비용 평가 함수 개선[4] 등 다양한 방법론이 제안되었다. 하지만 다양한 환경에서 테스트하지 않았으며, 특정 환경에서는 경로 계획 실패하는 현상까지 발생한다. 이러한 문제를 해결하기 위해, 휴리스틱 함수에 곡률 비용 함수와 새로운 장애물 비용 함수를 추가하여 장애물을 안전하게 회피하며 최소 시간을 가지는 최적 경로를 계획하는 A* 알고리즘을 제안하고자 한다.

Ⅱ. 본론

자율 주행에서 경로 계획은 중요한 도구로 높은 시장 수요, 넓은 응용 가능성, 그리고 큰 개발 잠재력으로 인해 많은 연구자들의 주목을 받고 있다. 그러나 기존 A* 알고리즘은 거리 측도 식만 사용하기 때문에 급격한 회전이 필요하며 장애물에 인접하여 회전하는 경로가 계획될 수 있다. 본 연구에서는 경로 계획 문제를 해결하기 위해 개선된 A* 알고리즘을 제안한다.

2.1 곡률 비용 함수

기존 A* 알고리즘이 생성한 경로는 급격한 회전이 필요할 수 있다. 만약 이 경로를 에이전트가 사용할 경우, 에이전트는 경로를 제대로 따라가지 못할 수 있으며, 이는 이동 비용이 증가할 수 있다. 따라서 경로 계획 시 곡률을 감소시키는 것이 필수적이다. 이러한 곡률 감소를 위해 기존 휴리스틱 함수에서 곡률 비용 함수를 추가한다. 기존 휴리스틱 함수에 곡률 비용 함수를 추가한 식은 식 (2)와 같다.

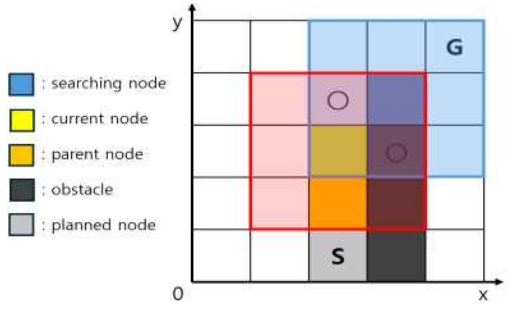

그림 1. 장애물 확인을 위한 각 노드에 대한 인접 노드 확인

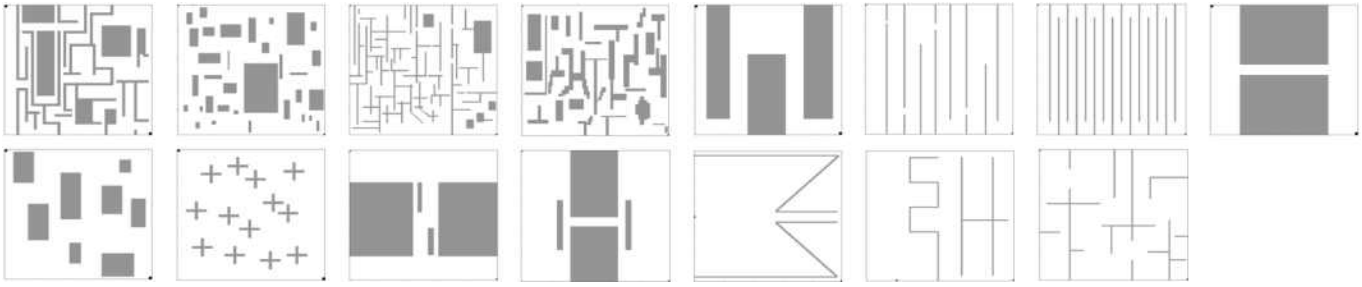

그림 2. 15가지의 맵에 대한 정보.

$$F(n) = G(n) + H(n) + C(n) \quad (2)$$

식 (2)에서 n은 노드이며, $C(n)$이 곡률 비용 함수이다. 곡률 비용 함수 $C(n)$은 탐색 노드와 현재 노드, 현재 노드의 부모 노드 총 3개의 노드를 사용하여 경로 곡률을 계산한다. 헤론 공식을 사용하여 세 노드로 구성되는 삼각형 넓이를 구하고, 구한 넓이를 활용하여 곡률을 계산한다. 자세한 식은 식 (3)~(5)와 같다.

$$s = \frac{Euclidean(n_{i-2},n_{i-1}) + Euclidean(n_{i-1},n_i)}{2} + \frac{Euclidean(n_{i-2},n_i)}{} \quad (3)$$

식 (3)에서 $Euclidean(n_{i-1},n_i)$는 현재 노드 n_{i-1}과 후보 노드 n_i 사이의 유클리디안 거리이다. 현재 노드의 부모 노드 n_{i-2}, 현재 노드 n_{i-1}, 후보 노드 n_i 총 세 노드 각각 사이의 거리를 구하고 이를 기반으로 s값을 구한다.

각 노드 간 유클리드 거리와 식 (3)에서 구한 s를 사용하여 삼각형의 면적 Δ을 계산한다. 삼각형의 면적을 구하는 식은 식 (4)와 같다.

$$\Delta = \sqrt{s \cdot (s - Euclidean(n_{i-2},n_{i-1}) \cdot \sqrt{(s - Euclidean(n_{i-1},n_i) \cdot \sqrt{(s - Euclidean(n_{i-2},n_i)}} \quad (4)$$

삼각형의 면적 Δ을 사용하여 곡률을 계산한다. 계산 식은 식 (5)와 같다.

$$C(n) = \frac{4 \cdot \Delta}{Euclidean(n_{i-2},n_{i-1}) \cdot Euclidean(n_{i-1},n_i) \cdot Euclidean(n_{i-2},n_i)} \quad (5)$$

앞서 언급한 식을 사용하여 구한 곡률 값은 비용 함수에 더해지며, 이는 경로 탐색 시 곡률이 커지는 경로는 비용이 증가하게 되므로, 급격한 회전보다 완만한 경로를 가지는 후보 노드를 선택하며 경로를 계획하게 된다.

본 연구에서는 곡률 비용을 효율적으로 사용하기 위해 곡률 계산에 사용하는 노드 수를 설정할 수 있도록 하였다. 개선한 곡률 함수 식인 $C(n,k)$은 식 (6)과 같다.

$$C(n,k) = \frac{\sum_{j=1}^{k} C(n_{i-j+1})}{k} \quad (k \geq 1) \quad (6)$$

k는 1 이상의 값으로 설정하는 상수로, k가 1일 때 기존 곡률 함수와 동일하며, k가 2일 경우 탐색 노드 n_i 기준 곡률 값과 현재 노드 n_{i-1} 기준 곡률 값을 구하여 구한 곡률들의 평균 값을 사용한다. 이를 통해 현재 노드 기준 이전 노드들의 곡률을 고려하며 탐색을 진행할 수 있다.

2.2 장애물 비용 함수

기존 A* 알고리즘은 장애물에 인접하여 회전하는 경로를 계획한다. 이러한 경로는 로봇이 경로를 따라가면서 회전하는 도중 장애물과 충돌할 수 있으며, 이는 자율 주행 성능에 큰 악영향을 끼친다. 따라서 장애물에 인접한 상태에서 회전하지 않기 위해 장애물 비용 값을 계산한다. 탐색 노드 n_i와 현재 노드 n_{i-1} 각각 8방향의 인접 노드가 서로 겹치는 위치에 장애물이 있으며 노드 n_i, n_{i-1}, n_{i-2} 가 한 직선에 존재하지 않을 때 비용을 추가한다. 그림 1은 두 노드 각각의 인접 노드가 겹치는 위치에 장애물이 존재하는 예시를 보여주는 그림이다. 시작 노드 S에서 목표 노드 G까지 이동하는 경로를 계획할 때, 탐색 노드 (4,4)의 파란색 영인 8방향의 인접 노드와 현재 노드의 인접 노드(3,3)으 빨간색 영역인 8방향의 인접 노드 8개가 겹치는 위치는 동그라미 표시가 되어있는 (3,4), (4,3)이다. 두 위치 중 (4,3) 노드에 장애물이 존재하며 세 노드가 한 직선에 있지 않으므로 비용을 추가한다. 장애물 비용 함수까지 추가한 최종 식은 식 (7)과 같다.

$$F(n) = G(n) + H(n) + C(n,k) + O(n) \quad (7)$$

Ⅲ. 실험

제안한 알고리즘을 테스트하고 비교하기 위해 2차원 그리드 환경의 맵에서 경로를 계획하고자 한다. 2차원 그리드 환경은 경로 계획 알고리즘의 평가를 위한 표준적인 테스트베드로 사용된다. 컴퓨팅 자원은 ubuntu 20.04 Desktop 운영체제, RAM 24GB, Intel® Core™ i5-9600KF CPU @ 3.70GHz이며 Python을 사용하여 경로 계획 알고리즘을 구현하였다. 맵의

그림 3. 6번 맵에서 각 알고리즘에 대한 경로 계획 결과. 왼쪽부터 A*, AdaptA*[1], EBHSA*[2], XiaoA*[3], proposed A*

Map No.	경로 내 점 수					장애물과 가까운 점 수					경로 평균 곡률					경로 곡률 편차					소요 시간				
	A*	AdapA*	EBHSA*	XiaoA*	Proposed	A*	AdaptA*	EBHSA*	XiaoA*	Proposed	A*	AdaptA*	EBHSA*	XiaoA*	Proposed	A*	AdaptA*	EBHSA*	XiaoA*	Proposed	A*	AdaptA*	EBHSA*	XiaoA*	Proposed
1	**135**	**135**	-	138	141	124	122	-	124	82	**0.14**	0.16	-	0.23	0.15	0.21	0.21	-	0.29	**0.19**	**0.01**	0.03	-	0.11	0.06
2	**129**	**129**	153	132	169	67	41	**0**	45	4	0.1	0.12	0.09	0.12	**0.03**	0.15	0.16	0.13	0.19	**0.09**	**0.01**	0.17	0.03	0.78	0.03
3	**222**	226	-	225	238	157	95	-	147	**78**	0.13	**0.12**	-	0.29	0.12	0.21	0.18	-	0.31	**0.18**	**0.07**	0.21	-	0.31	0.35
4	**145**	**145**	166	148	157	93	64	**0**	60	45	0.21	0.15	0.1	0.22	**0.08**	0.15	0.17	0.2	0.24	**0.13**	**0.01**	0.15	0.07	0.74	0.15
5	**128**	**128**	146	128	132	105	85	**0**	99	8	0.06	0.11	0.1	0.12	**0.06**	0.13	0.18	0.2	0.21	**0.11**	**0.01**	0.04	0.04	0.14	0.04
6	**313**	316	-	317	329	234	86	-	161	**12**	0.05	0.07	-	0.26	**0.04**	0.15	0.16	-	0.29	**0.11**	**0.16**	0.28	-	0.89	0.51
7	**1234**	1248	1291	1241	1279	1176	174	**0**	811	27	0.03	0.04	0.05	0.26	**0.03**	0.13	0.12	0.14	0.27	**0.12**	**0.22**	0.32	0.33	0.92	0.58
8	**76**	**76**	81	78	84	43	21	**0**	16	0	0.05	0.14	0.71	0.06	**0.04**	**0.1**	0.17	0.13	0.11	0.11	**0.01**	0.03	0.01	0.11	0.02
9	70	**67**	75	68	80	43	29	**0**	16	10	0.10	0.13	0.16	0.11	**0.04**	0.15	0.16	0.19	0.12	**0.11**	**0.01**	0.06	0.01	0.2	0.01
10	**54**	**54**	61	56	57	14	14	**0**	14	12	**0.07**	0.11	0.19	0.11	0.10	0.14	0.17	0.2	0.14	**0.13**	**0.01**	0.04	0.01	0.21	0.02
11	**140**	146	148	141	155	65	74	**0**	35	1	0.04	0.13	0.04	0.11	**0.02**	0.09	0.19	0.11	0.21	**0.07**	**0.01**	0.15	0.01	0.56	0.04
12	**131**	**131**	138	133	157	62	61	**0**	33	2	0.04	0.08	0.07	0.05	**0.03**	0.09	0.14	0.14	0.12	**0.09**	**0.01**	0.15	0.0	0.54	0.04
13	**143**	**143**	146	144	144	44	48	**0**	2	44	0.01	0.03	0.04	0.15	**0.01**	**0.06**	0.10	0.12	0.25	0.07	**0.06**	0.26	0.11	1.1	0.25
14	**242**	**242**	251	249	254	139	104	**0**	106	2	0.02	0.03	0.04	0.18	**0.01**	0.08	0.1	0.13	0.27	**0.06**	**0.08**	0.31	0.5	1.57	0.33
15	**166**	**166**	-	166	170	34	48	-	16	**15**	0.06	0.05	-	0.12	**0.05**	0.12	0.13	-	0.22	**0.12**	**0.16**	0.28	-	1.04	0.53

표. 1. 실험 결과

크기는 50 x 50 크기와 100 x 100 크기의 맵 두 종류이며 각각 5가지, 10가지의 맵으로 구성하였다. 맵 형태는 그림 2와 같다. 왼쪽 위 맵이 1번 맵이며 오른쪽 방향 순서가 된다. 100 x 100 크기의 그리드 맵은 2-4, 6, 7, 11-15번 맵이며, 50 x 50 크기의 그리드 맵은 1,5,8-10번 맵이다. 출발지와 목적지는 13,14번 맵을 제외하면 각각 왼쪽 위 모서리 끝 점 (0,100)과 오른쪽 아래 모서리 끝 점(100,0)이다. 13,14번 맵의 목적지는 다른 맵과 같으며 13번 맵의 출발점은 (0,50)이고 14번 맵의 출발점은 (20,0)이다. 맵 종류는 크게 세 가지로 나누었으며, 1~4번 맵은 복잡한 환경, 5~7번 맵은 노드 탐색이 많은 환경, 8~15번 맵은 넓은 환경으로 구성하였다.

제안한 A* 알고리즘의 성능 비교를 위해 현재까지 제안된 A* 알고리즘과의 비교 실험을 진행하였다. 비교 알고리즘은 A*, AdaptA*[2], EBHSA*[3], XiaoA*[4]으로 총 4가지이며 제안한 알고리즘인 Proposed A* 알고리즘까지 해서 5가지의 알고리즘을 15가지의 맵에서 경로 계획을 진행하였다. 그림 3은 4번 맵에서 각 알고리즘으로 경로를 계획한 결과이다. 왼쪽부터 차례대로 A*, AdaptA*, EBHSA*, XiaoA*, Proposed A* 알고리즘으로 계획한 결과이다. A*, AdaptA*, XiaoA* 알고리즘은 장애물에 인접한 상태에서 회전을 하지만 EBHSA*, Proposed A* 알고리즘은 장애물에 인접하여 회전하지 않는 경로를 계획한 것을 볼 수 있다. EBHSA* 알고리즘에서는 양방향 탐색으로 인해 특정 구간에서 비효율적인 경로가 계획되었다. 전체 실험 결과는 표 1에 정리하였다.

실험 결과, Proposed A* 알고리즘은 장애물과 가까운 점 수, 경로 평균 곡률, 경로 곡률 편차에서 좋은 성능을 보여주었다. 경로 내 점 수에서는 A* 알고리즘이 좋은 성능을 보여주었다. 기존 A* 알고리즘에서는 급격한 회전을 허용하기 때문에 다른 알고리즘보다 경로 내 점 수가 적은 것을 확인할 수 있다. 장애물과 가까운 점 수에서는 EBHSA* 알고리즘이 확장 거리로 인해 최고의 성능을 보여주었지만, 확장 거리로 인하여 좁은 구역을 이동하지 못하여 1,3,6,15번 맵에서 경로를 계획하지 못하는 현상이 발생하였다. 따라서 EBHSA* 알고리즘을 제외한 알고리즘 중에서는 Proposed A* 알고리즘이 장애물과 가까운 점의 수에서 가장 좋은 성능을 보여주었다. 경로 곡률에서는 모든 환경에서 Proposed A* 알고리즘이 우수한 성능을 보여주었다. 경로 평균 곡률에서는 복잡한 환경인 1, 3번 맵에서 각각 A*, AdaptSearA* 알고리즘이 Proposed A* 알고리즘보다 좋은 성능을 보여주었지만 최대 0.0049 차이밖에 발생하지 않았다. 경로 곡률 편차에서는 복잡한 환경과 노드 탐색이 많은 환경에서 좋은 성능을 보여주었다. 넓은 환경에서도 8,13번 맵을 제외한 다른 맵에서도 좋은 성능을 보여주었다. 이는 대부분의 환경에서 안전한 경로 계획 관점에서 강인한 성능을 보여주는 모습을 보여주었다. 계산 시간에서는 추가 연산을 하지 않는 A* 알고리즘이 가장 좋은 성능을 보여주었지만, 모든 알고리즘이 대부분 1초를 넘기지 않고 경로를 계획하는 것을 확인할 수 있었다.

Ⅳ. 결론

본 논문에서는 A*알고리즘의 휴리스틱 함수에 곡률 비용 함수와 장애물 비용 함수를 추가한 새로운 경로 계획 알고리즘을 제안하였다. 곡률 비용 함수는 경로의 곡률을 최소화하기 위해 세 점을 헤론 공식을 기반으로 새롭게 곡률을 계산하여 회전이 적은 경로를 생성할 수 있도록 하였으며, 장애물 비용 함수는 현재 노드와 탐색 노드의 인접 노드를 활용하여 장애물에 인접하지 않고 회전하는 경로를 계획할 수 있도록 하였다.

기존 A* 알고리즘 및 기타 개선된 A* 알고리즘과의 비교 실험에서 제안한 알고리즘은 다양한 환경에서도 경로의 부드러움, 장애물과의 거리 측면에서 우수한 성능을 보였다. 특히 복잡한 환경과 노드 탐색이 많은 환경에서도 제안한 알고리즘이 안전한 경로를 생성할 수 있음을 확인하였다.

본 연구는 경로 곡률 감소와 장애물 회피를 동시에 고려한 알고리즘의 효과를 입증함으로써 자율 주행, 로봇 공학 등 다양한 응용 분야에서 안전하고 효율적인 경로 계획의 가능성을 제시하였다.

참 고 문 헌

[1] TAN, Chee Sheng; MOHD-MOKHTAR, Rosmiwati; ARSHAD, Mohd Rizal. A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. IEEE Access, 2021, 9: 119310-119342.

[2] ZHANG, Jing, et al. Research on effective path planning algorithm based on improved A* algorithm. In: Journal of Physics: Conference Series. IOP Publishing, 2022. p. 012014.

[3] WANG, Huanwei, et al. An efficient and robust improved A* algorithm for path planning. Symmetry, 2021, 13.11: 2213.

[4] SA, Xiao; HUAIYU, Wu; ZHIHUAN, Chen. Research of mobile robot path planning based on improved A* algorithm. In: 2020 Chinese Automation Congress (CAC). IEEE, 2020. p. 7619-7623.

Non-IID 환경에서 연합학습 성능 향상을 위한 강화학습 및 군집 지능 기반 시스템 설계

장선영[1], 최미정[1,2,*]

강원대학교 데이터사이언스학과[1], 강원대학교 컴퓨터공학과[2]

{jsy0708, *mjchoi}@kangwon.ac.kr

A Reinforcement Learning and Swarm Intelligence-Based Framework for Enhancing Federated Learning Performance in Non-IID Environments

Sun-Young Jang[1], Mi-Jung Choi[1,2,*]

Dept. of Data Science, Kangwon National Univ.[1]
Dept. of Computer Science and Engineering, Kangwon National Univ.[2]

요 약

연합학습(Federated Learning)은 데이터를 공유하지 않고 분산 학습을 수행할 수 있도록 설계된 기법으로, 개인정보를 보호하면서 분산 학습이 가능하다. 그러나 연합학습은 Non-IID(비독립동일 분포) 환경에서 성능 저하와 적대적 클라이언트 공격으로 인한 보안 취약성의 한계를 가진다. 본 논문에서는 이러한 문제를 해결하기 위해 강화학습 기반 클라이언트 선택과 군집 지능 기반 집계 가중치 할당을 통합한 새로운 연합학습 시스템을 제안한다. 본 시스템은 Non-IID 환경에서도 모델 성능과 보안성을 유지하도록 설계되었다. 강화학습 기반 클라이언트 선택 기법은 연합학습에 참여할 클라이언트를 무작위로 선택하는 기존 방식과 달리, 각 클라이언트의 데이터 분포, 과거 학습 가중치, 현재 손실 등의 상태 정보를 반영하여 학습에 적합한 클라이언트를 선별한다. 또한 군집 지능 기반 집계 가중치 할당 기법은 기존의 단순한 데이터 크기 기반 가중 평균 방식을 개선하여 각 클라이언트의 기여도를 평가하고 이를 반영하여 동적으로 가중치를 할당한다. 이를 위해 입자 군집 최적화(Particle Swarm Optimization, PSO) 알고리즘을 적용하여 클라이언트의 기여도를 정량화한다. 본 논문에서 제안하는 시스템은 Non-IID 환경과 적대적 클라이언트가 존재하는 환경에서도 연합학습의 효율성과 견고성을 향상시킬 것으로 기대된다.

Ⅰ. 서 론

컴퓨팅 기술의 발전으로 지난 수십 년간 컴퓨터의 처리 속도는 기하급수적으로 증가하였으며, 이는 인공지능(AI) 기술의 발전에 중요한 기반이 되었다. 또한 사물인터넷(IoT), 스마트폰, 엣지 컴퓨팅 등의 기술들이 급속히 확산되면서 데이터 생성량은 전례 없이 폭발적으로 증가하고 있다. 이러한 데이터의 급증은 인공지능의 학습 가능성과 활용 범위를 크게 확장시켰지만 개인정보 보호 및 데이터 주권과 관련된 새로운 과제를 동시에 제기하고 있다[1].

Google은 이와 같은 문제를 해결하기 위해 연합학습(Federated Learning)[2]을 제안했다. 연합학습은 데이터를 중앙 서버로 전송하지 않고도 분산된 장치들이 협력하여 공통의 글로벌 모델을 학습할 수 있는 방법론이다. 그뿐만 아니라, 프라이버시 보호와 통신 효율성을 동시에 고려한 분산 학습 방식으로 개인정보 보호의 중요성이 부각되고 있는 현대 사회에서 그 필요성과 활용도가 더욱 주목받고 있다. 그러나 초기의 연합학습 방식은 몇 가지 핵심적인 한계점에 직면해 있다. 첫 번째로, Non-IID(Non-Independent and Identically Distributed) 데이터로 인한 모델 성능의 저하이다. 각 클라이언트가 상이한 특성의 로컬 데이터를 기반으로 학습에 참여하기 때문에 글로벌 모델의 수렴 속도와 일반화 성능에 부정적인 영향을 미칠 수 있다[3]. 두 번째로, 적대적 클라이언트의 공격에 취약하다는 점이다. 일반적으로 모델 중독 공격이 있는데, 이는 공격자가 글로벌 모델에 업로드하기 전에 로컬 모델을 수정하여 글로벌 모델에 영향을 끼친다. 따라서 모델 집계 과정에서 보안성을 향상시키기 위한 연구가 필요하다[4].

이에 본 논문에서는 기존 연합학습의 구조적 한계를 극복하기 위해 강화학습(Reinforcement Learning) 기반 클라이언트 선택 메커니즘과 군집 지능(Swarm Intelligence) 기반 집계 가중치 할당 메커니즘을 통합한 새로운 연합학습 시스템의 설계를 제안한다. 지금까지 다양한 개선 연구가 진행되었으나 클라이언트 선택과 집계 과정을 동시에 최적화한 연구는 상대적으로 부족하였다. 본 연구는 강화학습을 통해 각 클라이언트의 정보(데이터 분포, 학습 성능 등)를 반영하여 각 클라이언트의 학습 참여 여부를 보다 정교하게 결정하고, 군집 지능의 분산 최적화 기법을 활용하여 Non-IID 환경에서도 각 클라이언트에 적합한 가중치를 부여함으로써 효과적인 학습과 보안성 강화를 동시에 달성하고자 한다. 제안하는 시스템은 클라이언트의 데이터 특성을 고려한 적응형 클라이언트 선택과 군집 지능 기반 가중치 조정을 통해 연합학습의 효율성과 견고성을 향상시키는 것이 목표이다.

본 연구의 주요 기여 내용은 다음과 같다:

(1) 본 연구는 강화학습을 활용하여 클라이언트의 상태 정보(데이터 분포, 이전 학습 가중치, 현재 손실 등)를 종합적으로 고려한 적응형 클라이언트 선택 메커니즘을 제안한다. 이는 기존의 무작위 기반 선택 방식이나 고정된 규칙 기반 방식과 달리 환경 변화에 동적으로 반응하며 최적의 클라이언트를 선별할 수 있다는 점에서 차별화된다.

(2) PSO(Particle Swarm Optimization) 알고리즘을 도입한 가중치 집

계 방식을 적용함으로써 기존의 데이터 크기만을 고려한 단순 가중 평균 방식에 비해 더욱 최적화된 글로벌 모델 갱신이 가능하도록 한다. 클라이언트별 성능(적합도)에 비례하여 가중치를 반영함으로써 글로벌 모델의 학습 성능 향상을 기대할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 연합학습의 한계를 해결하기 위해 수행된 기존 연구들을 검토한다. 3장에서는 본 연구의 기반이 되는 핵심 기술인 연합학습, 강화학습, 군집 지능에 대해 설명하고, 4장에서는 제안하는 시스템의 구조와 세부 메커니즘을 설명한다. 마지막으로 5장에서는 본 연구의 결론과 향후 연구 방향에 대해 논의한다.

Ⅱ. 관련 연구

이 장에서는 Non-IID 환경에서 연합학습의 모델 성능 저하와 적대적 클라이언트에 취약하다는 한계를 해결하기 위한 기존 연구에 대해 검토한다. 연합학습의 문제들을 해결하기 위해 클라이언트 선택(Client Selection)과 클라이언트 집계(Client Aggregation)에 관한 두 가지 접근 방식이 연구되고 있다. 클라이언트 선택 연구는 학습에 참여할 클라이언트를 효율적으로 선정하는 전략이며, 클라이언트 집계 연구는 학습된 로컬 모델을 글로벌 모델로 통합하는 과정에서 적절한 가중치를 부여하는 방식에 관한 연구이다.

2.1 클라이언트 선택(Client Selection)

Zheng 외[5]는 클라이언트의 통계적 이질성과 시스템 자원 차이를 동시에 고려하는 최적화 프레임워크 FedAEB를 제안하였다. 이들은 Soft Actor-Critic(SAC) 기반의 심층 강화학습 기법을 활용하여, 클라이언트 선택과 자원 할당을 동시에 최적화하였다. 저자들이 제안한 방법은 데이터 품질 평가 요소를 도입함으로써 클라이언트의 프라이버시를 보호하면서도 데이터 분포의 질을 간접적으로 평가했다. 또한 모델 정확도, 에너지 소비, 지연 시간 간의 균형을 조절할 수 있는 가중치 메커니즘을 통해 다양한 응용 시나리오에 적응 가능한 유연성을 확보하였다. 저자들은 FedAEB를 평가하기 위해 Fashion-MNIST와 CIFAR-10 데이터셋에 대한 실험을 진행한 결과, FedAEB는 기존의 FedAvg, FedCS, FedCor 등의 기법보다 높은 정확도와 더 낮은 에너지 소비 및 지연 시간을 달성하였으며, 특히 데이터 이질성이 큰 환경에서 뛰어난 성능 향상을 보였다.

Wang 외[6]는 Non-IID 환경에서의 통신 효율성을 개선하기 위한 프레임워크인 FAVOR를 제안하였다. 이 연구는 강화학습, 특히 Deep Q-Network(DQN)[5]을 활용하여 통신 라운드마다 최적의 참여 클라이언트를 선정함으로써 데이터 편향성을 줄이고 모델 수렴 속도를 향상시키는 전략을 제시하였다. 저자들은 MNIST, Fashion-MNIST, CIFAR-10 데이터셋을 활용한 실험에서 각각 최대 49%, 23%, 42%의 통신 라운드 감소 효과를 보였다.

Khan 외[7]는 동적 클라이언트 참여, Non-IID 데이터, 적대적 노이즈 환경에서 9종의 군집 지능(Swarm Intelligence) 알고리즘 성능을 체계적으로 비교·분석하였다. 특히 Grey Wolf Optimization(GWO) 알고리즘이 모든 실험 시나리오에서 가장 높은 정확도, 재현율, F1-score를 기록하며 우수한 성능을 나타냈다. 이 연구는 군집 지능 기반 접근법이 클라이언트 선택의 정확도를 높이고, 오탐률을 줄이며, 자원 사용 최적화와 같은 목표 간 균형을 효과적으로 달성할 수 있음을 입증하였다.

2.2 클라이언트 집계(Client Aggregation)

Shi 외[8]는 연합학습에서 공정성과 효율성을 동시에 달성하기 위한 집계 방식으로 FedFAIM을 제안하였다. 이 방식은 모델 집계 과정에서 클라이언트의 데이터 품질을 고려하여 한계 손실(Marginal Loss) 지표를 기반으로 각 참여자의 업데이트 품질을 평가 후, 설정된 임계값 이상을 만족하는 클라이언트만 선별하여 집계에 반영한다. 또한 코사인 유사도를 바탕으로 각 클라이언트의 기여도를 판별하여 가중치를 부여한다. 이는 기존 FedAvg가 데이터 크기만을 기준으로 가중치를 부여하는 방식의 한계를 극복하며 품질 기반의 공정한 집계를 가능하게 한다. 저자들은 FedFAIM이 Non-IID 데이터 환경에서 높은 모델 정확도와 빠른 수렴 속도를 보였음을 입증하였다.

Jialuo 외[9]는 연합학습에서의 공정성과 견고성을 강화하기 위한 새로운 적응형 집계 알고리즘인 FedAA를 제안하였다. 저자들이 제안한 방법은 데이터 이질성과 적대적 클라이언트의 공격 가능성을 고려하여 심층 강화학습(Deep Reinforcement Learning) 기반으로 클라이언트의 기여도를 동적으로 조정하는 전략을 도입하였다. 심층 결정론적 정책 경사(DDPG; Deep Deterministic Policy Gradient) 알고리즘을 통해 집계 가중치를 연속적으로 제어하고 모델 파라미터 간 거리와 검증 세트 기반 보상 메커니즘을 통합하여 악의적 클라이언트에 대한 견고성과 데이터 불균형 환경에서의 공정성을 동시에 확보하였다. 실험 결과, FedAA는 기존 집계 기법 대비 견고성과 공정성 측면 모두에서 우수한 성능을 보였다.

대다수 기존 연구는 클라이언트 선택과 집계 과정을 별개의 문제로 다루는 경향이 있었으며, 일부 통합적 접근이 존재하더라도 수학적 계산 기반의 방식에 의존하여 실시간 변화에 대한 적응성이 부족한 한계가 있었다. 이에 본 논문에서는 이러한 연구 격차를 해소하고자 강화학습 기반 클라이언트 선택과 군집 지능 기반 집계 가중치 최적화를 결합한 통합 시스템을 제안한다.

Ⅲ. 배경 지식

이 장에서는 본 논문에서 제안한 시스템의 기반이 되는 세 가지 핵심 기술인 연합학습, 강화학습, 군집 지능에 대한 개념을 설명한다.

3.1 연합학습(Federated Learning)

연합학습(Federated Learning)은 McMahan 외[2]에 의해 처음 제안된 개념으로 중앙 서버에 원본 데이터를 전송하지 않고 분산된 엣지 디바이스에서 기계학습 모델을 개별적으로 학습한 후 이를 통합하여 글로벌 모델을 구축하는 분산 학습 방식이다. 이 방식은 데이터 프라이버시를 보호하면서도 효율적인 모델 학습이 가능하다는 점에서 주목받고 있다.

연합학습의 일반적인 프로세스는 다음과 같다: 중앙 서버는 초기 모델을 모든 참여 클라이언트에 배포하며 각 클라이언트는 자신의 로컬 데이터를 기반으로 해당 모델을 학습시킨다. 이후 클라이언트는 학습된 모델의 업데이트 정보(가중치 또는 그래디언트 등)만을 중앙 서버로 전송하고, 서버는 이를 집계하여 글로벌 모델을 갱신한다. 갱신된 글로벌 모델은 다시 클라이언트에 배포되며 이 과정은 사전에 정의된 통신 라운드 수만큼 반복되거나 모델이 수렴할 때까지 지속된다.

3.2 강화학습(Reinforcement Learning)

강화학습(Reinforcement Learning)은 에이전트가 환경과의 상호작용을 통해 시행착오를 거치며 최적의 행동 정책을 학습하는 기계학습의 한 분야이다. 이 방법론은 마르코프 결정 프로세스(Markov Decision Process, MDP)를 기반으로 하며, 일반적으로 상태 공간(State Space), 행동 공간(Action Space), 그리고 보상 함수(Reward Function)로 정의된다.

강화학습 알고리즘은 가치 기반(Value-based) 접근법, 정책 기반

(Policy-based) 접근법, 그리고 이 둘을 결합한 액터-크리틱(Actor-Critic) 접근법으로 분류된다. 가치 기반 접근법의 대표적인 예로 Q-learning이 있으며 이는 상태-행동 쌍의 가치를 추정하는 Q 함수를 학습하여 최적의 정책을 도출한다. 심층 강화학습(Deep Reinforcement Learning)에서는 DQN(Deep Q-Network)이 합성곱 신경망(CNN)을 이용해 고차원 상태 공간에서도 효과적으로 학습을 수행할 수 있음을 보여주었다. 정책 기반 접근법은 명시적인 가치 함수 없이 정책 자체를 직접 학습하며 REINFORCE, Proximal Policy Optimization(PPO) 등의 알고리즘이 이에 속한다. 액터-크리틱 방법은 정책 함수(Actor)와 가치 함수(Critic)를 동시에 학습하여 각각의 장점을 상호 보완적으로 활용하는 접근 방식으로 안정성과 수렴 속도 측면에서 우수한 성능을 보인다.

3.3 군집 지능(Swarm Intelligence)

군집 지능(Swarm Intelligence)은 사회적 곤충과 같아 자연계의 분산 시스템에서 영감을 받은 인공지능의 한 분야로, 단순한 개체 간의 지역적 상호작용을 통해 복잡하고 지능적인 집단 행동이 자발적으로 나타나는 현상을 인공지능에 적용하는 방안을 연구하는 분야이다. 군집 지능의 핵심 특성은 자기 조직화, 분산 제어, 간접적 상호작용, 그리고 집단 의사결정이다. 군집 지능을 대표하는 알고리즘 중 하나인 개미 군집 최적화(Ant Colony Optimization, ACO)는 개미들이 페로몬을 따라 최적 경로를 찾는 자연 현상을 모방한 방식으로, 주로 조합 최적화 문제에 효과적으로 적용된다. 이 외에도 인공 벌 군집 알고리즘(Artificial Bee Colony, ABC), 박쥐 알고리즘(Bat Algorithm), 반딧불이 알고리즘(Firefly Algorithm) 등 다양한 자연 생태 기반의 최적화 기법들이 연구되고 있다.

입자 군집 최적화(Particle Swarm Optimization, PSO)[10] 알고리즘은 Kennedy와 Eberhart에 의해 제안된 대표적인 군집 지능 기반 최적화 알고리즘으로 새 떼나 물고기 무리의 사회적 행동에서 영감을 받아 연속 공간에서의 전역 최적화 문제를 해결하는 데 사용된다. PSO에서 각 입자는 다음과 같은 방식으로 위치를 갱신하며 탐색 공간을 이동한다:

$$v_i^{t+1} = wv_i^t + c_1 r_1 (p_i - x_i^t) + c_2 r_2 (g - x_i^t)$$
$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

여기서 w는 관성 가중치, c_1과 c_2는 인지적 요소와 사회적 요소의 영향력을 조절하는 가속 계수, p_i는 개인 최적 위치, g는 전역 최적 위치이다. 이 과정을 통해 입자들은 점진적으로 최적해를 향해 수렴한다.

본 연구에서 제안하는 시스템은 동적 클라이언트 선택에 액터-크리틱 기반 강화학습과 클라이언트 집계 가중치 할당에 PSO 알고리즘의 적용을 제안한다. 강화학습은 클라이언트의 데이터 품질, 이전 라운드의 학습 결과, 현재 로컬 학습 손실값 등 다양한 요소를 고려하므로 글로벌 모델의 성능 향상을 위해 보다 효과적인 클라이언트를 선택할 수 있다. PSO 알고리즘은 비볼록 최적화 문제 해결 능력, 분산 환경과의 자연스러운 부합, 적대적 공격에 대한 견고성이 강하다. Non-IID 데이터로 인한 목적 함수의 비볼록성은 전통적인 접근 방식의 성능을 저하시킬 수 있으나 PSO는 이러한 환경에서도 효과적으로 해를 찾을 수 있다. 따라서 강화학습과 군집 지능의 결합은 연합학습에서 클라이언트 선택과 클라이언트 집계 가중치 할당 문제를 동시에 최적화함으로써, 보다 효율적이고 강건한 연합학습 시스템을 구축할 수 있다.

Ⅳ. 시스템 설계

본 논문에서는 Non-IID 데이터 환경과 적대적 클라이언트 공격에 효과적으로 대응하기 위한 통합적인 시스템을 제안한다. 제안된 시스템은 강화학습 기반 클라이언트 선택 메커니즘과 군집 지능 기반 집계 가중치 할당 메커니즘으로 구성되며, 두 메커니즘을 통해 연합학습의 효율성과 견고성을 향상시킨다.

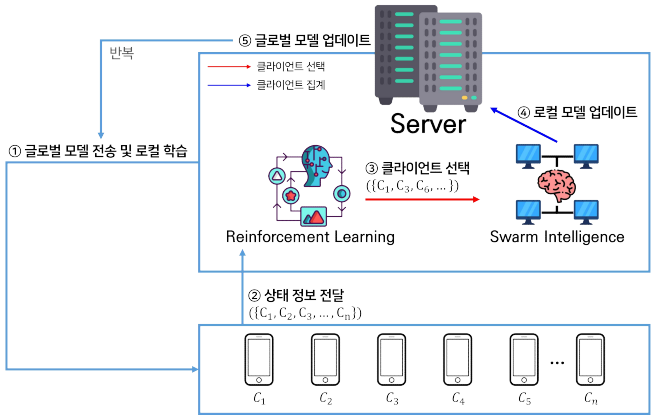

그림 1. 시스템 설계

4.1 프레임워크 개요

제안하는 프레임워크의 전체 구조는 그림 1과 같으며 다음과 같은 단계적 프로세스를 따른다:

1) 중앙 서버는 초기 글로벌 모델을 모든 클라이언트에 배포하고, 각 클라이언트는 로컬 데이터셋을 사용하여 모델을 학습한다.
2) 각 클라이언트는 학습 결과와 함께 상태 정보를 서버에 전송한다.
3) 서버의 강화학습 모듈은 각 클라이언트의 상태 정보를 기반으로 참여 클라이언트를 선택한다.
4) 선택된 클라이언트의 모델 가중치는 군집 지능 알고리즘인 PSO 최적화 알고리즘을 통해 집계되어 글로벌 모델을 갱신한다.
5) 갱신된 글로벌 모델의 성능을 평가하고 이를 기반으로 보상을 계산한다.
6) 위 과정을 지정된 통신 라운드 수 만큼 반복한다.

4.2 강화학습 기반 클라이언트 선택

4.2.1 상태공간

본 연구에서는 클라이언트 i의 상태 s_i를 다음과 같이 정의한다:

$$s_i = [D_i, W_i^{prev}, L_i^{current}]$$

- $D_i \in \mathbb{R}^d$는 클라이언트 i의 데이터 분포를 나타내는 벡터로, 클래스별 데이터 분포 비율, 데이터 샘플 수 등의 정보를 포함한다.
- $W_i^{prev} \in \mathbb{R}$는 클라이언트 i의 이전 학습 가중치의 통계적 요약값으로, 이전 라운드의 학습 결과를 반영한다.
- $L_i^{current} \in \mathbb{R}$는 클라이언트 i의 현재 로컬 학습 손실값으로, 모델의 현재 성능을 나타낸다.

전체 시스템의 상태 S는 N개의 모든 클라이언트의 상태를 포함하는 집합으로 정의된다:

$$S = \{s_1, s_2, \ldots, s_N\}$$

4.2.2 행동공간

강화학습 에이전트의 행동 $a_i \in [0,1]$는 클라이언트 i의 참여 확률을 나타내며, 다음과 같이 이산적 선택으로 변환된다:

$$select_i = \begin{cases} 1 & \text{if } a_i \geq \theta \\ 0 & otherwise \end{cases}$$

여기서 θ는 클라이언트 선택을 위한 임계값이며, 실제 연구 진행에 따라 임계값을 설정할 수 있다.

4.2.3보상함수

보상 함수는 글로벌 모델의 성능을 고려하여 다음과 같이 설계하였다:

$$R = Accuracy(M_{global})$$

4.3 PSO 기반 집계 가중치 할당

선택된 클라이언트의 가중치를 효과적으로 할당하기 위해 PSO 알고리즘을 도입한다. PSO는 지능적 최적화 알고리즘으로 특히 비볼록 최적화 문제에 효과적이며, 지역 최적해에 빠질 위험을 줄인다.

4.3.1 PSO 기반 최적화 프로세스

PSO 기반 집계 가중치 할당은 다음과 같은 단계로 진행된다:

(1) 입자 초기화: 각 클라이언트의 모델 가중치 집합 W_i를 PSO의 입자로 간주한다.

(2) 적합도 평가: 각 입자(클라이언트 가중치 조합)의 적합도를 평가한다.

$$f(W_i) = Accuracy(W_i)$$

(3) 개인 최적 위치 및 글로벌 최적 위치 갱신:

$$P_i = \arg\max_{W_i' \in history(W_i)} f(W_i')$$

$$G = \arg\max_i f(P_i)$$

(4) 속도 및 위치 갱신:

$$v_i = wv_i + c_1 r_1 (P_i - W_i) + c_2 r_2 (G - W_i)$$
$$W_i^{new} = W_i + v_i$$

글로벌 최적 위치와 개인 최적 위치를 동시에 고려하여 글로벌 모델의 성능을 높이는 최적의 모델 가중치 집합을 찾는다. 이러한 적응형 가중치 집계 방식은 성능이 좋은 클라이언트의 기여도를 높이고, 성능이 낮거나 잠재적으로 악의적인 클라이언트의 영향을 줄이는 효과가 있다.

따라서 이러한 통합 알고리즘은 강화학습의 적응적 특성과 군집 지능의 최적화 능력을 결합하여 Non-IID 데이터 환경과 적대적 공격 상황에서 보다 효율적이고 강건한 연합학습을 가능하게 할 것으로 기대된다.

Ⅴ. 결론 및 향후연구

본 논문에서는 Non-IID 환경과 적대적 공격이 존재하는 상황에서도 견고한 연합학습을 실현할 수 있는 새로운 시스템을 제안하였다. 제안된 시스템은 강화학습 기반 클라이언트 선택 메커니즘과 군집 지능 기반 집계 가중치 할당 메커니즘을 결합하여 연합학습의 효율성과 안정성을 동시에 확보하는 것을 목표로 한다. 강화학습 기반 클라이언트 선택 기법은 각 클라이언트의 데이터 분포, 이전 라운드 학습 결과, 현재 손실값을 종합적으로 고려하여 최적의 클라이언트를 선별한다. 또한, 비볼록 함수 최적화에 효과적인 군집 지능 알고리즘을 활용하여 클라이언트의 학습 결과를 글로벌 모델로 집계할 때 최적의 가중치를 동적으로 할당한다.

본 연구에서 제안한 접근 방식은 Non-IID 데이터 및 적대적 클라이언트가 존재하는 환경에서도 기존 연합학습 기법보다 우수한 학습 효율성과 향상된 견고성을 제공할 것으로 기대된다. 향후 연구로는 제안한 시스템을 실제로 구현하고 성능을 분석하여 실험적 검증을 수행할 계획이다. 또한 다양한 강화학습 알고리즘과 PSO 이외의 군집 지능 기반 알고리즘을 적용하여 보다 최적화된 알고리즘 조합을 탐색하는 연구를 진행할 예정이다.

ACKNOWLEDGMENT

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(RS-2023-00242528).

참 고 문 헌

[1] NIPA: 세계 각국 정부의 '디지털 주권' 확보 경쟁 [Online], https://www.globalict.kr/upload_file/kms/2022/06/7629220500473272 5.pdf. Accessed Mar. 24, 2025.

[2] McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B. A. "Communication-efficient learning of deep networks from decentralized data," *Proceedings of the 20th International on Artificial Intelligence and Statistics*, Florida, USA, Vol. 54, pp. 1273 - 1282, April 2017.

[3] Kairouz, P., McMahan, B., et al., "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, Vol. 14, No. 1 - 2, pp. 1 - 210, June 2021.

[4] Qi, P., Chiaro, D., Guzzo, A., Ianni, M., Fortino, G., Piccialli, F., "Model aggregation techniques in federated learning: a comprehensive survey," *Future Generation Computer Systems*, Vol. 150, pp. 272 - 293, Jan. 2024.

[5] Zheng, F., Sun, Y., Ni, B., "FedAEB: Deep reinforcement learning based joint client selection and resource allocation strategy for heterogeneous federated learning," *IEEE Transactions on Vehicular Technology*, Vol. 73, No. 6, pp. 8835 - 8846, June 2024.

[6] Wang, H., Kaplan, Z., Niu, D., Li, B., "Optimizing federated learning on non-IID data with reinforcement learning," *Proceedings of the IEEE Conference on Computer Communications*, Toronto, ON, Canada, pp. 1698 - 1707, July 2020.

[7] Khan, K., Goodridge, W., "Swarm intelligence-driven client selection for federated learning in cybersecurity applications," *arXiv preprint arXiv:2411.18877*, Nov. 2024.

[8] Shi, Z., Zhang, L., Yao, Z., Lyu, L., Chen, C., Wang, L., Li, XY., "FedFAIM: A model performance-based fair incentive mechanism for federated learning," *IEEE Transactions on Big Data*, Vol. 10, No. 6, pp. 1038 - 1050, Dec. 2024.

[9] He, J., Chen, W., Zhang, X., "FedAA: A reinforcement learning perspective on adaptive aggregation for fair and robust federated learning," *arXiv preprint arXiv:2402.05541*, Feb. 2024.

[10] Kennedy, J., Eberhart, R., "Particle swarm optimization," *Proceedings of IEEE International Conference on Neural Networks*, Perth, WA, Austrailia, Vol. 4, pp. 1942 - 1948, Dec. 1995.

하이브리드 클라우드 환경에서의 머신러닝 기반 동적 보안 정책 적용 및 위협 탐지에 관한 연구

김건민[1], 김예진[1], 이은성[1], 장현지[1], 김경백[2]

전남대학교[1], 전남대학교 인공지능융합학과[2]

204869@jnu.ac.kr, ye031010@jnu.ac.kr, 200750@jnu.ac.kr, gka1225@jnu.ac.kr, kyungbaekkim@jnu.ac.kr

A Study on Dynamic Policy Enforcement Using Machine Learning in On-Premise and Cloud Hybrid Environments

Geonmin Kim[1], Yejin Kim[1], Eunseong Lee[1], Hyeonji Jang[1], Kyungbaek Kim[2]

Chonnam National University[1],
Dept. of Artificial Intelligence Convergence, Chonnam National University[2]

요 약

하이브리드 클라우드 환경은 기존의 온프레미스 환경의 견고성과 클라우드의 유연성을 결합함으로써 기업의 운영 효율성을 혁신적으로 향상시키고 있다. 그러나 이러한 이질적인 두 환경의 결합은 전통적인 경계 기반 보안 접근법의 한계를 드러내고 있다. 이에 따라 최근에는 하이브리드 클라우드 환경의 보안 및 성능 최적화에 AI를 기반으로 하는 접근법이 제안되고 있으며, 이를 실증하기 위하여 본 논문에서는 하이브리드 클라우드 환경에서 온프레미스와 퍼블릭 클라우드 간의 보안 격차를 해소하기 위한 머신러닝 기반 동적 보안 정책 적용 통합 보안 아키텍처를 제안하고, 실험을 통하여 그 효과를 논한다. 제안된 아키텍처는 프록시 게이트웨이와 OPA를 연동하여 API 트래픽을 중앙 집중식으로 제어하고, Elasticsearch와 Kibana를 통해 로그 데이터를 수집 및 분석한다. 또한, 머신러닝 모델로 수집된 로그를 학습하고, 학습된 결과를 바탕으로 동적으로 보안 정책을 업데이트하여 OPA 정책에 반영함으로써 정적으로는 탐지하기 어려운 공격 패턴을 효과적으로 식별한다.

Ⅰ. 서 론

최근 빅데이터와 클라우드 컴퓨팅의 급속한 발전으로 인해 전 세계 기업들은 이전과는 다른 방식으로 데이터를 처리하고 운영하게 되었다. 전통적인 온프레미스(On-Premise)[1] 환경은 데이터 저장과 운영을 기업 내부적으로 수행하여 데이터 보안과 물리적 통제에 강점을 지녔으나, 초기 구축 비용이 상당히 높고, 유연성과 확장성 측면에서 많은 제약이 따랐다. 또한, 정기적인 업데이트에 따른 지속적인 하드웨어 및 네트워크 인프라의 유지 관리 비용 또한 급속히 늘어나는 데이터의 양에 대응하기 어렵게 만들었다.

이러한 배경에서 클라우드 컴퓨팅이라는 새로운 패러다임이 등장하며 온프레미스 환경만을 활용하는 방식은 점차 비효율적으로 변하고 있다. 이에 기업들은 보다 유연하고 확장성이 뛰어난 대안을 모색하며 클라우드 기반으로 전환을 시작하였다. 클라우드 컴퓨팅은 초기 투자 비용을 획기적으로 낮추고, 필요에 따라 자원을 탄력적으로 할당할 수 있는 장점을 제공하여 기업의 IT 운영 효율성을 극대화하는 대안으로 부상하였다. 이러한 흐름 속에서 많은 기업들은 클라우드 시스템을 온프레미스 환경과 결합하여 하이브리드 클라우드 환경(Hybrid Cloud Architecure)[2]를 구축하는 방향으로 나아가고 있다. 하이브리드 클라우드 환경은 퍼블릭 클라우드(Public Cloud), 프라이빗 클라우드(Private Cloud) 그리고 온프레미스 인프라가 네트워크를 통해 상호 연결된 환경을 의미하며, 이를 통해 데이터와 애플리케이션이 원활하게 호환될 수 있도록 구성된다. 퍼블릭 클라우드는 아마존 웹 서비스(AWS), 마이크로소프트 에저(Azure), 구글 클라우드(GCP)와 같은 클라우드 서비스 제공업체를 통해 외부에서 인프라를 빌려 사용하는 방식이며, 프라이빗 클라우드는 특정 기업이 내부적으로 운영하는 클라우드 환경을 의미한다. 하이브리드 클라우드 환경은 이러한 퍼블릭 및 프라이빗 클라우드를 온프레미스 인프라와 결합하여, 기업이 필요에 따라 유연하게 자원을 할당하고 데이터 저장 및 처리 방식을 최적화할 수 있도록 지원한다. Flexera 2024 State of the Cloud Report에 의하면, 2024년 기준 73%의 기업이 하이브리드 클라우드 환경을 채택하고 있다.

그러나 온프레미스와 클라우드를 결합하는 하이브리드 환경은 이질적인 환경이 결합하여 네트워크의 경계가 모호해지고, 구조적 복잡성의 증가로 인해 보안 측면에서 새로운 위협 요인들이 대두된다.[3][4] 이러한 문제를 해결하기 위하여 보다 통합적이고 지능적인 보안 체계가 요구되며, 최근에는 인공지능(Artificial Intelligence) 및 생성형 인공지능(Generative AI)를 기반으로 한 새로운 보안 접근법들이 제안되고 있다.[5][6][7][8] 이에 따라 본 논문에서는 Kong Gateway를 기반으로 하는 프록시 게이트웨이 구조에 OPA(Open Policy Agent)를 결합하고, 여기에 머신러닝 기반의 이상 탐지 기법을 추가한 통합 보안 아키텍처를 제안한다. 본 아키텍처는 게이트웨이를 활용한 중앙 집중적인 API 통제를 통해 플랫폼 간 보안 일관성을 확보하고, 머신러닝 모델이 Elasticsearch에 기록된 로그를 바탕으로 실시간 이상 행위를 탐지하여 OPA의 Rego 정책을 동적으로 반영함으로써 지능형 공격에도 신속하게 대응할 수 있도록 설계되었다. 또한, Kibana를 활용하여 실시간 로그 시각화 및 분석을 지원함으로써 정책 실행 결과에 대한 가시성과 관리 편의성을 동시에 달성할 수 있도록 하였다.

Ⅱ. 전통적인 보안 모델의 하이브리드 클라우드 환경에서의 한계

전통적인 보안 모델은 네트워크 경계를 기준으로 내부와 외부를 명확히 구분하고, 외부로부터의 위협을 차단하기 위한 방화벽, 침입 탐지 시스템, 접근 제어 목록 등의 기술을 구축하는 방식으로 설계되었다. 이는 온프레

미스 환경에서 상당 기간 유효하게 기능하였으며, 기업 내부망과 외부망 간 트래픽을 체계적으로 통제함으로써 다수의 보안 위협을 사전에 방어할 수 있었다. 그러나 하이브리드 클라우드 환경은 퍼블릭 클라우드, 프라이빗 클라우드, 온프레미스 환경이 상호 연결된 구조로 네트워크 경계를 물리적으로 명확하게 설정하기 어렵다. 이는 전통적인 방화벽이나 경계 기반 보안 장비들이 효과를 발휘하기 어려운 환경을 만든다.

또한, 이러한 구조적 복잡성은 공격 표면을 크게 넓힌다. 단일 온프레미스 환경에서는 방화벽을 통과하는 지점이 명확하였으나, 하이브리드 클라우드 환경에서는 각기 다른 클라우드 서비스와 온프레미스 시스템을 가로지르는 트래픽이 다방면으로 발생한다. 기업 내부에서 운영되는 서비스가 퍼블릭 클라우드에 배포된 API를 호출하거나, 프라이빗 클라우드의 마이크로서비스가 온프레미스 데이터베이스에 접근하는 등 복합적인 시나리오가 빈번하여 전통적인 방식으로는 모든 상호작용 경로에 대해 일관된 보안 정책을 적용하기가 어려워진다.

더 나아가 하이브리드 클라우드 환경에서는 보안 정책이나 접근 제어에 대한 일관성 확보 또한 어렵다. 각각의 클라우드 서비스 제공 업체가 제공하는 보안 관리 도구나 API가 모두 상이[9]하며, 로깅 및 이벤트 수집 방식도 상이하여 단일한 정책으로 모든 환경을 동기화하기가 어렵다. 결국 하이브리드 클라우드 환경에서는 기존의 경계 기반 기술만으로는 모호한 네트워크 경계와 동적으로 변화하는 자원을 대상으로 일관된 정책을 빠르게 적용하기 힘들며, 새로운 형태의 공격을 효과적으로 탐지하기도 어렵다. 이러한 복잡성과 취약성을 극복하기 위해서는 더욱 지능적인 보안 접근법이 필요하며, 이에 따라 본 연구에서는 하이브리드 클라우드 환경에서 머신러닝 기반 동적 보안 정책 관리 시스템을 제안한다.

Ⅲ. 머신러닝 기반 동적 정책 적용 보안 아키텍처 설계

제안하는 통합 보안 아키텍처는 하이브리드 클라우드 환경 전반의 트래픽을 일관적으로 제어하기 위해 설계되었다. 이를 위해, API를 기반으로 트래픽 흐름을 제어하는 Kong Gateway를 통해 외부 요청이 모든 백엔드 서비스로 전달되는 경로를 단일화하고, 정책 결정 모듈인 OPA(Open Policy Agent)를 별도로 두어 요청 적합성 평가를 수행한 후 허용 또는 차단을 결정한다. 그동안 많은 보안 솔루션은 정적 시그니처를 기반으로 공격을 필터링하는 방식에 의존해 왔으나 이를 극복하기 위하여 본 연구에서는 머신러닝 기반 분석을 통해 새롭게 나타나는 공격 패턴을 실시간으로 학습하고, 그 결과를 정책 엔진에 동적으로 반영함으로써 지능적이고 유연한 방어 체계를 구성하고자 한다.

아키텍처 전반의 동작 흐름은 다음과 같다. 첫째, 다양한 소스(온프레미스와 클라우드 양쪽)에서 발생하는 API 요청들은 API 게이트웨이를 거치며, 게이트웨이는 해당 요청과 관련된 정보(메서드, 경로, 헤더, 사용자 에이전트 등)를 모두 로그로 남긴다. 이 로그들은 모두 Elasticsearch에 저장되어 축적된다. 머신러닝 분석 모듈은 일정 주기마다 축적된 로그를 수집해 정상 및 공격 여부를 분류하고, 공격으로 판결된 로그에서 새로운 키워드나 잠재적 위험 패턴을 추출한다. 이렇게 추출된 패턴은 정규 표현식 형태로 OPA에 전달되어 이후 동일한 패턴이 재등장하면 게이트웨이가 이를 자동으로 차단하게 된다. 이로써 공격자가 난독화나 변형 기법을 동원하더라도, 일정 횟수 이상의 시도가 관측되면 곧바로 정책에 반영되어 더 이상 통하지 않게 된다.

결국 본 아키텍처는 동적 정책 업데이트를 핵심 기제로 삼는다. 이는 정적 시그니처 기반 솔루션 대비 훨씬 민첩하게 환경 변화를 따라갈 수 있게 하며, 다양한 온프레미스와 클라우드 자원을 아우르는 하이브리드 보안 전략의 필수 요건을 충족한다. 게이트웨이와 정책 엔진, 머신러닝 분석 모듈, 그리고 대규모 로그 저장소로 구성된 본 통합 시스템은 고도화되는 데이터 탈취 시도 등의 공격을 보다 지속적이고 지능적으로 방어할 수 있도록 지원한다.

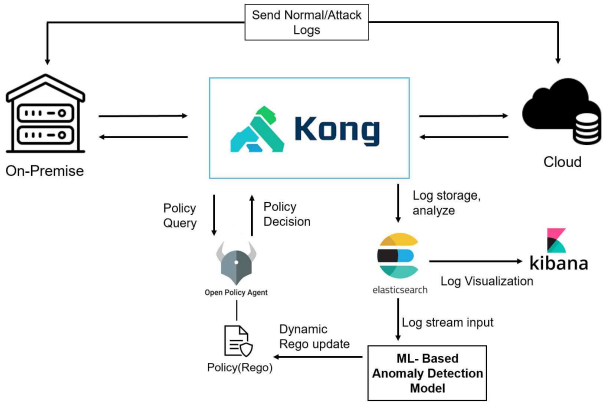

Fig 1. 실험 시나리오 도식도

1. 주요 구성 요소

Kong Gateway는 오픈소스 API 게이트웨이로, 모든 API 트래픽이 반드시 통과해야 하는 관문 역할을 한다. 외부 클라이언트 요청은 게이트웨이를 통해 내부 서비스로 라우팅 되며, 이 과정에서 인증, 인가, 로깅, 로드밸런싱 등의 기능을 중앙에서 수행한다. 특히 보안 측면에서 게이트웨이가 OPA 정책 결정 모듈(정적 규칙 + 머신러닝에서 도출된 동적 규칙)을 호출하여, 요청에 담긴 정보가 사전에 정의된 정책 위배 요소가 있는지를 검사한다. 이를 통해 공격 추적과 차단을 일관적으로 수행할 수 있으며, 온프레미스와 클라우드 모두 동일한 게이트웨이를 거치므로 운영 환경이 여러 개라도 정책 관리가 단순해지는 장점을 갖는다.

정책 결정 모듈로 사용되는 OPA는 제로 트러스트 아키텍처(ZTA, Zero Trust Architecture)를 기초로 텍스트 형태의 규칙(정규 표현식, 사용자 정의 쿼리 등)과 JSON 기반 정책 언어를 사용하며, API 요청이 들어올 때마다 접근 허용 여부를 판별한다. 특히 본 시스템에서 OPA는 Kong Gateway를 이용한 중앙 집중식 접근 제어를 통해 온프레미스 및 클라우드에 맞추어 서비스마다 다르게 보안 정책을 관리할 필요 없이 일괄적인 정책 관리가 가능하다.

Elasticsearch는 오픈소스 분산 검색 엔진으로, 대량의 로그 데이터를 빠르게 저장하고 검색할 수 있는 도구이다. JSON 기반의 문서 데이터를 인덱싱하고, 이를 고속으로 조회할 수 있도록 설계되어 로그 수집, 데이터 분석, 모니터링 등에 널리 사용된다. 특히, 모든 로그는 logs 인덱스로 Elasticserach에 저장되며, 저장된 데이터는 추후에도 보안 이벤트 분석에 사용될 수 있다. Kibana는 Elasticsearch에 저장된 데이터를 시각화하여, 로그를 빠르게 분석할 수 있도록 지원하는 도구로, Kibana 대시보드를 통해 실시간으로 보안 이벤트를 시각화하여 분석할 수 있다.

머신러닝 분석 모듈은 정적 규칙으로는 탐지하기 어려운 공격을 식별하기 위해 사용된다. 본 연구에서는 TF-IDF 기반 벡터화 기법을 통해 로그의 문자열 특징을 추출한 뒤, 이를 랜덤포레스트 분류 모델에 입력하여 정상 및 공격을 판별한다. 공격으로 분류된 로그에서 상위 중요도의 단어를 추출하여, 이를 정규 표현식 형태로 변환하여 정책 결정 모듈에 공유한다. 이렇게 머신러닝 모델의 결과가 정책 결정에 반영되어 공격 패턴이 반복적으로 나타날 시 곧바로 차단 대상이 되어 방어 효율이 크게 향상된다.

2. 실험 로그 구성

실험을 진행하기 위하여 본 연구에서는 로그 생성 과정에서 정상 로그와 공격 로그를 구분하여 설계하였다. 모든 로그는 랜덤 난수(seed)를 통해 생성된다. 정상 로그는 정상 키워드 리스트를 활용하여 일반적인 API 호출, HTTP 요청, 사용자 입력, 데이터베이스 쿼리 등을 무작위로 생성한다. 공격 로그는 매 실험마다 생성되는 전체 로그 중 다른 비율로 다양한 형태의 악성 키워드를 포함하도록 의도적으로 삽입한다. 스크립트 태그를 사용한 XSS 페이로드, 시스템 명령어, Base64 인코딩 후 재인코딩한 난독화된 문자열, SQL Injection 등을 여러 비율로 넣는다. 또한, 공격 페이로드를 단순 텍스트 형태로만 넣는 대신, 난독화된 해시 문자열 사이에 끼워 넣거나, URL 인코딩으로 일부 문자를 변형하는 방식 등을 적용하여 단순한 정규식 패턴으로는 탐지가 어렵도록 구성하였다. 이러한 실험에 사용되는 모든 로그는 Elasticsearch에 저장되며, 추후 머신러닝 분석 모듈의 학습 데이터로 활용된다.

3. 로그 수집 및 분석 인프라 구축

프록시 게이트웨이로 채택된 Kong Gateway는 단순한 API 요청의 전달 역할을 넘어 온프레미스와 클라우드 간의 복잡한 데이터 이동 경로를 중앙 집중식으로 관리하는 핵심 구성 요소이다. Kong Gateway는 API 요청 발생 시 OPA와 연계해 미리 정의된 보안 정책을 적용하고 접근 제어를 수행한다. 보안 정책의 동적 관리 및 세밀한 접근 제어를 위하여 사용되는 OPA와 Kong Gateway와의 연동은 정책 변경 사항이 실시간으로 반영될 수 있도록 지원하며, 발생하는 모든 로그와 이벤트 정보는 모두 저장되어 추후 보안 체계 개선 및 정책 수정에 중요한 자료로 활용된다. 이러한 동적 정책 결정 메커니즘은 기존 정적 규칙에 의존한 보안 시스템이 갖는 한계를 극복하고, 변화하는 위협 환경에 신속하고 유연하게 대응할 수 있는 기반이 된다.

이 과정에서 발생하는 모든 로그와 이벤트는 Kafka를 통해 스트리밍되어 Elasticsearch에 기록된다. Elasticsearch와 Kibana는 대규모 로그 데이터를 신속하게 인덱싱하고, 검색할 수 있는 기능을 제공하므로, 이를 이용하여 시스템 전반에서 발생하는 보안 이벤트 및 API 호출 기록 등을 관리할 수 있다. 특히, 도중에 Elasticsearch와의 연결이 중단되더라도 로그 손실을 최소화하기 위하여 5회까지 연결을 재시도하는 안전장치를 추가하였다. Kibana는 Elasticsearch에 저장된 데이터를 시각화하여, 로그를 빠르게 분석할 수 있는 도구로, Kong Gateway를 거쳐 발생하는 API 요청 및 이 과정에서 발생하는 이벤트들이 Elasticsearch에 저장되면 Kibana를 이용하여 단순한 데이터 분석을 넘어 시간대별, 사용자별, 이벤트 유형별 등 다양한 차원의 데이터 시각화를 진행할 수 있으며, 이를 통하여 복합적인 보안 위협을 다각도로 분석할 수 있다.

4. 머신러닝 모델을 이용한 로그 분석 및 동적 정책 관리

정적 시그니처나 간단한 정적 정책만으로는 탐지하기 어려운 공격들을 가려내고, 동적으로 변화하는 공격 패턴을 빠르게 학습 및 대응하기 위하여 렌덤포레스트 머신러닝 모델을 도입한다. 머신러닝 모델은 TD-IDF 기반의 벡터화를 통해 URI, 헤더, User-Agent, 요청 본문 등에 포함된 문자열의 빈도 정보를 추출한 뒤, Randomforest 지도 학습 분류기를 사용하여 학습한다. 본 연구에서 사용되는 머신러닝 분석 모듈은 Elasticsearch에 저장된 로그 10,000개씩 나누어 학습하며, 1회차 학습에는 사전에 라벨링 되어 시뮬레이션 된 로그를 통하여 정상 로그와 공격 로그의 패턴을 학습한다. 이후 추가되는 로그들에 대하여 정상 로그인지 공격 로그인지 분석하고 분석된 보안 위협을 바탕으로 OPA의 보안 정책을 자동으로 동적으로 수정 및 추가한다. 이 정보는 정규표현식 기반의 Rego 정책으로 반영되며 OPA에 자동 업데이트됨으로써 보안 정책이 점진적으로 정교화된다. 지속적으로 발생하는 공격 로그들은 추가된 OPA 보안 정책의 패턴에서 식별되면 차단되는 체계로, 이러한 과정은 반복을 통하여 자동화된다. 머신러닝 모듈과 정책 모델 간의 피드백 루프를 형성을 통해 각 학습 라운드마다 탐지 정확도가 향상되었으며, 실험을 통해 그 효과를 정량적으로 입증하였다.

Ⅳ. 실험 결과 및 성능 분석

본 연구에서는 제안한 통합 보안 아키텍처의 효율성을 평가하기 위해 총 10회의 독립적인 실험을 수행하였다. 각 실험은 독립적인 환경에서 서로 다른 초기 조건과 공격 패턴으로 구성되어 있으며, 각 실험에서 10,000개의 로그를 학습 주기로 20회의 반복 학습 및 평가를 진행하였고, 이를 통해 시스템의 학습 능력과 동적 정책 갱신 효과를 평가하였다.

	공격 로그 차단율			
실험 회차	1회 학습	2회 학습	10회 학습	20회 학습
1회차	50.98%	53.36%	73.51%	97.30%
2회차	49.65%	51.93%	71.58%	97.38%
3회차	51.43%	52.46%	71.88%	97.59%
4회차	48.95%	54.13%	72.02%	97.65%
5회차	50.48%	52.56%	72.63%	97.19%
6회차	50.10%	52.90%	72.58%	97.71%
7회차	50.54%	54.30%	73.84%	97.54%
8회차	50.07%	51.68%	71.80%	97.78%
9회차	50.27%	52.87%	72.99%	97.53%
10회차	50.24%	51.84%	72.68%	97.54%
평균 차단율	97.52%			

Table 1. 제안 아키텍처에 대한 10회 실험 결과

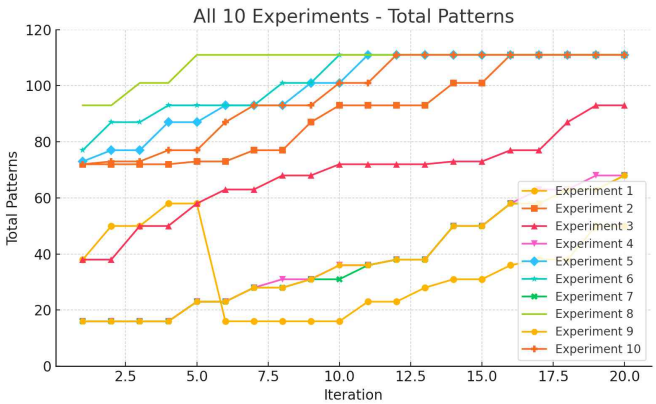

Fig 2. 10회 실험 별 OPA 정책 개수 변화

초기 실험 단계에서는 머신러닝 모델의 학습 부족으로 인해 공격 탐지 및 차단율이 평균 50% 전후로 나타나 다소 낮은 성능을 보였다. 그러나 반복적인 학습이 진행됨에 따라 공격 탐지의 정확도는 지속적으로 개선되어 실험 후반으로 갈수록 신규 공격 패턴에 대한 학습 효과가 두드러져 성능이 급격히 향상되었다. 특히 실험 반복 횟수가 증가함에 따라 시스템은 반복되는 공격 패턴을 정확하게 식별하여 정책 업데이트에 반영할 수 있었으며 최종적으로 모든 실험에서 차단율이 97%를 상회하였다. 이러한 결과는 제안된 아키텍처가 반복 학습과 실시간 정책 업데이트를 통해 매우 높은 정확도로 공격을 탐지하고 차단할 수 있음을 명확히 나타낸다. 전

체 10회 실험 결과 평균 공격 차단율은 97.52%로 이는 하이브리드 클라우드 환경에서 공격 대응을 위한 실질적인 보안 성능 측면에서 안정적인 성능을 보임을 입증한다.

이러한 성능 개선의 주요 원인은 시스템의 동적 정책 관리 기능에서 기인한 것으로 분석된다. 실험이 지속됨에 따라 정책 패턴이 평균적으로 증가하며 중반 이후부터는 최대 111개까지 확대되었다. 결론적으로, 본 연구를 통해 제안한 통합 보안 아키텍처는 반복 학습과 동적 정책 갱신을 통해 매우 높은 정확도의 공격 탐지와 차단 성능을 제공할 수 있음을 입증하였다. 이는 급변하는 하이브리드 클라우드 환경의 다양한 위협 환경에 신속하고 유연하게 대응할 수 있음을 보여준다.

V. 결론 및 향후 연구

본 연구에서는 온프레미스와 클라우드가 결합된 하이브리드 클라우드 환경에서 발생하는 보안 격차를 효과적으로 완화하고, 지능적이고 복합적인 공격에 효율적으로 대응하기 위하여 프록시 게이트웨이 및 OPA 기반 보안 아키텍처와 머신러닝 모듈을 결합한 모델을 제안하였다. 제안된 구조는 프록시 게이트웨이를 중심으로 API 트래픽을 중앙 집중형으로 제어하고, Elasticsearch와 Kibana를 활용하여 로그 분석 환경을 구성하였으며, 이와 동시에 머신 러닝 기반의 로그 분석 및 동적 정책 생성 메커니즘을 통합하여 지능형 공격 및 새로운 공격 패턴에 신속하고 유연하게 대응하고, 높은 정확도로 식별할 수 있음을 실험을 통해 입증하였다. 실험 결과, 프록시 게이트웨이와 동적 정책 관리를 결합한 환경은 97% 이상의 로그 차단율을 기록하였다. 이는 머신러닝 모듈이 시그니처 기반 정적 탐지로는 놓칠 수 있는 이상 행위를 Elasticsearch의 로그 데이터를 통해 비정상적 행위의 특성을 학습함으로써 고도화된 위협에 대해서도 방어력을 제공한다는 점을 보여준다. 이는 정적 규칙에만 의존하는 기존 보안 체계와 달리 데이터 기반 분석을 통하여 오탐과 누락을 줄이고 공격 유형을 세분화하여 더욱 정교한 보안 대응이 가능했음을 시사한다.

향후 연구에서는 더욱 다양한 유형의 공격 기법과 대규모 트래픽 상황을 고려한 추가 실험을 통해 실제 보안 환경에서의 다양한 공격 유형을 반영할 필요가 있다. 이번 연구에서는 SQL Injection, XSS, 난독화된 문자열 공격 등의 유형에 집중했으나 SSRF(Server-Side Request Forgery), CSRF(Cross-Site Request Forgery), 원격 코드 실행(RCE), DNS 바인딩 등의 공격도 고려하여 보다 다양한 공격 시나리오를 체계적으로 반영할 필요가 있다. 또한, 현재는 일정 주기로 로그를 수집하여 일괄적으로 학습하는 방식을 사용하였지만 보다 실시간에 가까운 대응을 구현하기 위하여 온라인 학습 방식이나 모델 경량화, 고속 처리 알고리즘을 도입하여, 보다 높은 트래픽 부하 상황에서도 지연 없이 신속한 대응이 가능하도록 추가적인 최적화 연구가 요구된다. 또한, 실무 적용을 위하여 NGINX Ingress Controller, Envoy Proxy 등 다양한 게이트웨이 활용도 함께 고려될 수 있다.

더 나아가, 제로 트러스트 아키텍처 개념을 본 시스템에 더욱 깊게 연계하는 것도 고려할 수 있다. 사용자 인증, 장치 상태, 네트워크 위치 등 다양한 맥락 정보를 활용하여 보다 정밀한 정책 판단이 가능하도록 OPA 정책을 확장할 수 있다. 반복적인 학습 과정을 통해 만들어지는 보안 정책들 간의 충돌이나 중복 문제도 고려해야한다. 정책 자동화와 충돌 관리를 위한 프레임워크를 구축하여 하이브리드 클라우드 환경 전반에서의 보안 가시성을 유지하면서도, 운영 복잡성을 낮추는 전략이 뒤따라야 한다. 이러한 후속 연구를 통하여 제안된 모델이 하이브리드 클라우드 보안 체계를 획기적으로 강화하는 핵심 요소가 될 수 있을 것으로 기대된다.

ACKNOWLEDGMENT

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-지역지능화혁신인재양성사업의 지원을 받아 수행된 연구임(IITP-2025-RS-2022-00156287, 50%). 본 연구는 한국인터넷진흥원(KISA)-정보보안 특성화대학 지원사업의 지원을 받아 수행된 연구임(50%).

참 고 문 헌

[1] M. Gaianu, "On Premise Data Center vs CLOUD," 2023 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2023, pp. 1068-1071

[2] A. Leff and J. T. Rayfield, "Integrator: An Architecture for an Integrated Cloud/On-Premise Data-Service," 2015 IEEE International Conference on Web Services, New York, NY, USA, 2015, pp. 98-104

[3] S. B. Mallisetty, G. A. Tripuramallu, K. Kamada, P. Devineni, S. Kavitha and A. V. P. Krishna, "A Review on Cloud Security and Its Challenges," 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2023, pp. 798-804

[4] G. Raktate, K. Shelar, P. Parjane, S. Pangavhane, S. More and S. R. Deshmukh, "A Survey on Security Issues and Challenges in Cloud Computing," 2024 International Conference on Decision Aid Sciences and Applications (DASA), Manama, Bahrain, 2024, pp. 1-5

[5] D. K. Seth, K. K. Ratra and A. P. Sundareswaran, "AI and Generative AI-Driven Automation for Multi-Cloud and Hybrid Cloud Architectures: Enhancing Security, Performance, and Operational Efficiency," 2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2025, pp. 00784-00793

[6] C. Anjani, R. M. Balajee, G. Divya, Y. S. Sree, K. Padmanabham and S. S. Srithar, "Evolving Threats and AI Solutions for Modern Hybrid Cloud Architectures," 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2023, pp. 478-484

[7] S. J, K. Kanagasabapathi, K. Mahajan, S. Ahamad, E. Soumya and S. Barthwal, "AI-Enhanced Multi-Cloud Security Management: Ensuring Robust Cybersecurity in Hybrid Cloud Environments," 2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2023, pp. 1-6

[8] Y. Mansouri, V. Prokhorenko, and M. A. Babar, “Anautomated implementation of hybrid cloud for performance evaluation of distributed databases,” J.Netw. Comput. Appl., vol. 167, p. 102740, Oct. 2020

[9] A. Mishra, P. Sarat and R. Afza, "A factual study on hybrid multi cloud cyber security threats and proposed methodologies to enable cyber resilience," 2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2024, pp. 1-6

대규모 언어 모델을 활용한 네트워크 자동화 연구 동향 분석

박지태, 박찬진, 조부승
한국과학기술정보연구원
{pjj5846, pcj0722, bscho}@kisti.re.kr

An Analysis of Research Trends in Network Automation using Large Language Models

Jee-Tae Park, Chanjin Park, Buseung Cho
Korea Institute of Science and Technology Information

요 약

최근 과학 기술의 급속한 발전과 네트워크 환경의 복잡성이 증가함에 따라, 기존의 수작업 기반 네트워크 관리 방식은 한계에 직면하고 있다. 과거에는 관리자가 직접 네트워크를 설정하고 유지보수하는 방식이 일반적이었으나, 네트워크 규모가 커지고 동적으로 변화함에 따라 이러한 접근 방식은 비효율적이며, 높은 비용과 시간이 요구된다. 이를 해결하기 위해 네트워크 자동화에 대한 연구가 오래전부터 진행되어 왔으며, 특히 SDN, NFV와 같은 기술과 ML/DL의 도입으로 자동화된 네트워크 관리가 점차 확산되고 있다. 최근에는 대규모 언어 모델(LLM)의 발전으로 네트워크 자동화의 새로운 가능성이 제시되고 있다. LLM은 자연어 이해와 추론 능력을 갖추고 있어 네트워크 구성, 장애 진단, 보안 정책 적용 등을 자동화하는 데 활용될 수 있다. 본 논문에서는 LLM을 활용한 네트워크 자동화 최신 연구들을 분석하여 연구 동향을 살펴본다. 또한, LLM을 네트워크 분야에 적용하기 위한 기술적 한계점과 향후 연구 방향을 제시함으로써 네트워크 자동화 기술의 발전과 실용적인 활용 가능성을 높이고자 한다.

Ⅰ. 서 론

네트워크는 현대 IT 인프라의 핵심 요소로, 기업 및 데이터 센터뿐만 아니라 통신망, 클라우드 서비스, 사물인터넷(IoT) 등 다양한 산업에서 필수적으로 활용된다. 특히, 디지털 전환이 가속화되면서 단순한 데이터 전송을 넘어 안정적인 서비스 제공, 보안, 성능 최적화의 기반 기술로 자리 잡고 있다 [1, 2].

과거에는 네트워크 엔지니어가 각 장비의 설정을 직접 변경하고, 장애나 보안 침입이 발생하면 수작업으로 대응하는 것이 일반적이었다. 명령줄 인터페이스(CLI) 또는 GUI를 사용해 개별적으로 설정을 조정하고, 로그와 모니터링 도구를 활용해 상태를 확인하며, 장애가 발생하면 원인을 분석해 해결했다. 하지만 네트워크 환경이 복잡해지면서 시간이 많이 소요되고, 많은 오류가 발생한다. 오늘날 네트워크는 대규모 환경에서 수많은 장비가 실시간으로 통신하며, 대용량 트래픽이 동적으로 변화하는 구조를 갖는다. 이에 따라 사람이 모든 변수를 고려해 최적의 설정을 유지하기 어려우며, 기존의 수동 방식은 대응 속도가 느리고 설정 오류나 정책 불일치로 인해 성능 저하 또는 보안 취약점이 발생할 수 있다 [2].

이를 해결하기 위해 네트워크를 자동으로 운영·관리하고, 실시간 최적화 및 장애 대응을 수행할 수 있는 네트워크 자동화 기술에 대한 연구가 활발히 진행되고 있다 [1]. 네트워크 자동화는 설정 변경을 일괄적으로 적용하고, 장비 간의 정책을 동기화하며, 실시간으로 트래픽을 분석하여 최적의 네트워크 상태를 유지하는 것을 목표로 한다.

초기 네트워크 자동화는 주로 반복적인 설정 작업을 간소화하는 데 초점을 맞추었다. 엔지니어들은 Python, Bash, TCL 등의 스크립트 언어를 활용하여 설정을 자동화하고, SNMP 또는 Expect 스크립트를 사용해 장비 상태를 모니터링하는 등의 방법을 도입했다. 이를 통해 일부 작업을 자동화할 수 있었지만, 확장성과 유지보수 측면에서 한계를 보였다 [2, 3].

이러한 한계를 극복하기 위해 최근에는 머신러닝(ML) 및 인공지능(AI)을 활용한 연구가 활발히 이루어지고 있으며, 특히 LLM을 적용한 네트워크 자동화 기술이 주목받고 있다. LLM은 대량의 데이터에서 관계와 의미를 학습하여, 복잡한 질문에 응답하고 자연어 기반 텍스트를 생성하는 인공지능 모델로, 다양한 분야에서 활용되고 있다 [1, 2]. 네트워크 자동화에 LLM을 적용하면 관리자가 자연어로 시스템과 상호작용할 수 있어, CLI나 스크립트 언어에 대한 전문 지식이 부족한 사용자도 쉽게 네트워크를 운영할 수 있다. 또한, 과거 장애 데이터를 분석하여 향후 발생할 수 있는 문제를 예측함으로써 보다 효율적이고 안정적인 네트워크 운영이 가능하다 [2].

LLM을 활용한 네트워크 자동화에 대한 연구는 활발히 진행되고 있지만, 기존 연구를 체계적으로 정리하고 동향을 분석한 사례는 부족하다. 본 논문에서는 LLM을 활용한 네트워크 자동화 최신 연구들을 대상으로 연구 동향을 분석하고, 향후 연구에서 해결해야 할 주요 과제를 제시한다.

본 논문의 구성은 본 장의 서론에 이어, 2장 본론에서 LLM 기반 네트워크 자동화의 최신 연구들을 설명한다. 이후, 3장 결론에서 본 논문에서 소개한 연구들의 동향을 정리하고, 향후 연구 방향을 제시하며 논문을 마친다.

Ⅱ. 본론

본 장에서는 LLM 기반 네트워크 자동화 연구를 다룬다. 네트워크는 다양한 세부 분야로 나뉘며, 각 분야에서 활발한 연구가 이루어지고 있다. 본 논문에서는 그중에서도 특히 연구가 활발한 네트워크 설계 및 구성, 최적화, 보안의 세 가지 분야에 대한 최신 연구를 설명하며, 각 분야의 연구 내용은 표 1에 정리하였다.

1) 네트워크 설계 및 구성(Network Design & Configuration)

네트워크 설계 분야는 안정적이고 효율적인 네트워크 인프라를 구축하기 위한 과정으로, 트래픽 흐름을 고려한 최적의 토폴로지 설계, 장비 선정, IP 주소 할당, 라우팅 및 네트워크 프로토콜 구성 등의 작업을 포함한다.

표 1. LLM을 활용한 네트워크 자동화 연구

분야	참조	게재일	연구 내용	특징
네트워크 설계 및 구성	[3]	2024.04	잘못된 네트워크 구성을 자동으로 정정	- 여러 LLM의 투표 매커니즘 - 8개의 대표적인 LLM을 대상으로 성능 검증
	[4]	2024.05	ChatGPT를 활용한 네트워크 배포 작업 자동화	- 실제 Cisco 라우터를 대상으로 실험
네트워크 최적화	[5]	2024.04	네트워크 알고리즘 개선 코드 자동 생성	- 네트워크 알고리즘에 대한 코드 자동 생성
	[6]	2024.10	비지식 기반 네트워크 관리 및 최적화	- 사전 지식 없이 다양한 네트워크 최적화
네트워크 보안	[7]	2024.08	DDoS 완화 프레임워크 제안	- 여러 가지 트래픽 특징 반영 - 네트워크 특화 프롬프트 엔지니어링 적용
	[8]	2024.07	악성 코드 기법 제안	- Mixtral LLM 모델 활용
	[9]	2025.02	APT 공격 탐지 기법 제안	- LLM 기반 임베딩과 Autoencoder 활용 - 매우 적은 비율의 APT 공격 탐지 가능

[3]에서 저자는 Ciri를 개발하여 네트워크 내 잘못된 구성을 해결 방법을 제시한다. 특히, LLM을 활용할 때, 발생하는 잘못된 대답(Hallucination) 및 비결정성 문제를 해결하는데 초점을 둔다. 실험에서 10개의 오픈소스 시스템과 8개의 LLM을 대상으로 LLM이 도출한 대답의 성능을 비교하였다. [4]에서 저자는 네트워크 환경이 빠르게 변화할 때, 자동으로 구성을 변경할 수 있는 방법을 제안한다. 제안하는 방법은 ChatGPT API를 활용하여 Ansible Playbook을 자동 생성하여 네트워크 배포 작업을 자동화한다. 4대의 Cisco 라우터에서 실험을 수행하였으며, 정확도 99%를 도출하였으며, 기존 방법 대비 효율성이 62% 증대하였다.

2) 네트워크 최적화(Network Optimization)

네트워크 최적화는 가용성, 성능, 보안, 비용 효율성을 극대화하기 위해 트래픽 흐름과 자원 활용을 최적화하는 과정이다. 이 과정에 라우팅 최적화, 대역폭 관리, 장애 대응, 트래픽 최적화 등의 작업을 포함한다.

[5]에서 저자는 NADA 프레임워크를 제안하였다. NADA는 LLM을 통해 기존의 함수 형태의 네트워크 알고리즘을 자동으로 개선하는 코드를 도출한다. 저자는 새로 개선된 ABR 알고리즘을 기존의 알고리즘과 비교하여 성능 측정을 수행하였다. [6]에서는 LLM을 활용하여 비지식 기반 네트워크 관리 프레임워크를 제안한다. 제안하는 방법은 LLM을 통해 방대한 데이터를 학습하고, 최소한의 시스템 정보를 포함한 입력 프롬프트만으로 새로운 작업에서 우수한 추론 성능을 보였다. 이는 사전지식 없이도 다양한 네트워크 최적화 작업에 활용될 수 있기 때문에 비전문가도 쉽게 활용할 수 있다.

3) 네트워크 보안(Network Security)

네트워크 보안은 외부 공격, 내부 위협, 데이터 유출 등을 방지하고 네트워크의 기밀성, 무결성, 가용성을 보호하는 과정이다. 이 과정에 방화벽 관리, 침입 탐지·방지 시스템(IDS/IPS), 접근 제어, 트래픽 모니터링, 취약점 분석 등의 작업을 포함한다.

[7]에서 저자는 LLM을 활용한 종합적인 DDoS 완화 프레임워크인 ShieldGPT를 제안한다. ShieldGPT는 공격 탐지, 트래픽 분석, 보안 지식 적용, 역할 분류의 네 가지 기능을 갖추고 있으며, 트래픽 특징을 반영한 분석 기법과 네트워크에 맞춘 질문 방식을 활용해 쉽게 이해할 수 있는 설명과 대응 방법을 제공한다. [8]에서 저자는 LLM을 활용한 Java 소스 코드 내 악성 코드 탐지 기법을 제안한다. 이 모델은 정상 및 악성 코드로 구성된 다양한 데이터 셋을 학습하여 기존의 정적 분석 도구보다 뛰어난 탐지 성능을 보인다. [9]에서 저자는 지능형 지속 공격(APT)를 탐지하기 위한 프레임워크를 제안한다. 제안하는 방법은 BERT, RoBERTa, DistillBERT을 기반으로 임베딩을 수행하고, AE, VAE, DAE 등의 모델로 탐지 성능을 향상시킨다. 실험을 통해 매우 적은 비율의 APT 공격에 대해서도 효과적으로 탐지할 수 있다.

Ⅲ. 결론

본 논문에서는 네트워크 자동화의 연구 배경과 필요성을 간략히 설명하고, 최근 활발히 진행되고 있는 LLM 기반 네트워크 자동화 연구 동향을 정리하였다. 본 논문에서는 네트워크 분야에서 연구가 활발한 네트워크 설계 및 구성, 최적화, 보안의 세 가지 핵심 분야를 중점적으로 정리하였다.

네트워크 자동화는 다양한 분야에서 활발히 연구되고 있으며, 최근에는 크게 세 가지 방향으로 발전하고 있다. 먼저, 비전문가도 쉽게 네트워크를 설정하고 관리할 수 있도록 지원하는 비지식 기반 자동화 기술이 주목받고 있다. 이를 통해 운영 효율성을 높이고 비용 절감 효과를 기대할 수 있다. 또한, 변화하는 네트워크 환경에 맞춰 사람이 개입하지 않아도 자동으로 구성을 조정하고 최적화하는 기술이 발전하고 있다. LLM을 활용한 트래픽 분석, 장애 예측 및 복구, 자원 할당 최적화 등이 주요 연구 주제로 떠오르고 있다. 마지막으로, 보안 분야에서는 LLM 기반 에이전트를 활용한 실시간 위협 탐지 및 대응 기술이 개발되고 있다. 예를 들어, 악성 코드의 동작 패턴을 자동 분석해 신속히 대응하는 기술이 연구되고 있으며, 네트워크 관리자와 보안 담당자가 직관적으로 이해할 수 있도록 공격 원인과 대응 방안을 자연어로 설명하는 기능도 개발 중이다.

최근 연구들을 살펴보면, 네트워크 기술은 LLM을 활용하여 운영 효율성과 보안성을 더욱 향상하는 방향으로 발전하고 있다. 그러나 몇 가지 해결해야 할 과제가 존재한다. 먼저, 네트워크 데이터, 특히 트래픽 데이터는 대부분 라벨링이 되어 있지 않아 도메인 특화 지식을 효과적으로 학습하는 데 어려움이 있다. 또한, LLM은 잘못된 정보를 생성하는 문제점과 동일한 환경과 입력에서도 일관되지 않은 응답을 생성하는 비결정성 문제가 존재한다. 이와 함께, 기존 네트워크 시스템과의 호환성 문제와 보안 이슈도 해결해야 할 중요한 과제다. 따라서 향후에는 이러한 한계점을 개선하기 위한 연구가 더욱 활발히 진행될 것으로 예상된다.

ACKNOWLEDGMENT

이 논문은 2025년도 한국과학기술정보연구원(KISTI)의 기본사업으로 수행된 연구입니다. (과제번호: K25L5M1C1)

참 고 문 헌

[1] 김태연, 고남석, 양선희, 김선미, “네트워크와 AI 기술 동향,” Electronics and Telecommunications Trends, vol. 35, no. 5, pp. 1 - 13, Oct. 2020.

[2] H. R. Chi, C. K. Wu, N. -F. Huang, K. -F. Tsang and A. Radwan, "A Survey of Network Automation for Industrial Internet-of-Things Toward Industry 5.0," in IEEE Transactions on Industrial Informatics, vol. 19, no. 2, pp. 2065-2077, Feb. 2023

[3] X. Lian, *et al.*, “Configuration Validation with Large Language Models”, arXiv:2310.09690, 2024.

[4] O. Okunaiya, R. Austin and S. Y. Zhu, "ChatGPT-enabled Network Automation using API-based Prompts," NOMS 2024-2024 IEEE Network Operations and Management Symposium, Seoul, Korea, Republic of, 2024, pp. 1-5.

[5] Z. He, *et al.*, “Designing Network Algorithms via Large Language Models”, arXiv:2404.01617, 2024

[6] H. Lee, *et al.,* “Large Language Models for Knowledge-Free Network Management: Feasibility Study and Opportunities”, arXiv:2410.17259, 2024

[7] T. Wang, “ShieldGPT: An LLM-based Framework for DDoS Mitigation”, in Proceedings of the 8th Asia-Pacific Workshop on Networking, 2024, pp. 108-114

[8] A. A. Hossain, M. K. PK, J. Zhang and F. Amsaad, "Malicious Code Detection Using LLM," NAECON 2024 - IEEE National Aerospace and Electronics Conference, Dayton, OH, USA, 2024, pp. 414-416,

[9] S. Benabderrahmane, *et al.*, “APT-LLM: Embedding-Based Anomaly Detection of Cyber Advanced Persistent Threats Using Large Language Models”, arXiv:2502.09385

네트워크 관리 자동화를 위한 LLM 기반 정책 및 네트워크 서비스 디스크립터 생성 연구

홍지범, 홍원기

포항공과대학교 컴퓨터공학과

{hosewq, jwkhong}@postech.ac.kr

A Study of LLM-based Method to Generate Policies and Network Service Descriptors for Automated Network Management

Jibum Hong and James Won-Ki Hong

Department of Computer Science and Engineering, POSTECH

요 약

현대 네트워크 환경은 클라우드 컴퓨팅과 가상화 기술의 발전, 그리고 인터넷 트래픽의 급증으로 인해 점차 복잡해지고 있으며, 이에 따른 효율적인 네트워크 관리 자동화의 필요성이 대두되고 있다. 본 논문에서는 대규모 언어 모델 (Large Language Model, LLM)을 활용하여, 네트워크 관리자 및 사용자의 의도 (Intent)를 효과적으로 반영하는 정책과 네트워크 서비스 디스크립터 (NSD)를 자동으로 생성하는 방법론을 제안한다. 제안하는 방법은 네트워크 도메인 특화 데이터의 수집 및 전처리, 프롬프트 설계와 Retrieval-Augmented Generation (RAG) 기법을 결합하여, JSON 이나 YAML 과 같이 NFVO 등 네트워크 관리 시스템에서 활용할 수 있는 생성 결과를 도출하는 프로세스를 포함한다. 제안하는 방법은 기존 수작업 및 규칙 기반 방식의 한계를 극복하고, 동적인 네트워크 환경에서도 유연하게 대응할 수 있는 관리 자동화를 실현하는 데 기여할 것으로 기대된다. 본 연구는 LLM 의 자연어 처리 능력을 네트워크 관리에 접목시킴으로써, 정책 및 NSD 생성에 대한 개선된 접근법을 제시하며, 향후 실제 구현 및 검증을 통한 실제 네트워크 적용 가능성과 확장성을 모색하기 위한 이론적 방법을 제공한다.

Ⅰ. 서 론

현대 네트워크 환경은 클라우드 컴퓨팅, 가상화 (Virtualization), 인터넷 트래픽 급증 등으로 인해 급격한 변화를 겪고 있으며, 이에 따라 네트워크 관리 복잡성과 서비스 수요가 비약적으로 증가하고 있다. 전통적인 네트워크 관리 방식은 이러한 변화에 효율적으로 대응하기 어렵기 때문에, 안정적이고 신속한 네트워크 운용을 위한 인공지능 기반의 자동화 및 지능형 관리 기법이 요구되고 있다 [1].

네트워크 관리 자동화, 즉 자율 네트워크 오케스트레이션은 네트워크 인프라의 동적인 변화에 실시간으로 대응할 수 있는 핵심 기술로 주목받고 있으며, 특히 네트워크 정책 (policy) 및 서비스 관리는 네트워크를 효율적으로 운용하기 위한 핵심 요소로 자리매김하고 있다. 기존의 정책 수립 방식은 네트워크 관리자 및 전문가의 수작업에 의존하는 경우가 많아, 변화하는 환경에 신속하고 유연하게 대응하지 못한다는 한계가 존재한다 [2].

이러한 문제를 해결하기 위해 인공지능 (AI) 기반 네트워크 관리 자동화 방법이 지속적으로 연구되고 있다. 또한, 네트워크 관리자 및 사용자의 Intent 를 기반으로 네트워크를 관리하는 Intent-based Networking (IBN) 또는 Intent-Driven Networking (IDN)에 대한 연구가 수행되었다. 이와 더불어 최근에는 대규모 언어 모델 (LLM)이 다양한 분야에서 뛰어난 문제 해결 능력을 보여주고 있어 네트워크 관리 영역에도 혁신적인 변화를 가져올 것으로 기대를 받고 있다 [3].

본 논문은 LLM 을 기반으로 Intent 기반 정책 및 네트워크 서비스 디스크립터 (Network Service Descriptor, NSD) 생성 방법을 제안함으로써, 네트워크 관리 자동화의 효율성을 향상시키고자 한다. 제안하는 방법은 네트워크 환경의 복잡성을 고려하여 정책 및 NSD 생성 과정을 자동화하고, 변화하는 네트워크 조건에 효율적으로 대응할 수 있는 방법을 제시한다. 또한, 이를 통해 네트워크 관리 자동화 실현을 위한 정책 및 NSD 생성의 이론적 방법을 제공한다.

Ⅱ. 관련 연구

기존 NSD 및 정책 생성 방법은 크게 2 가지로 구분된다. 먼저, 전통적인 수작업 및 규칙 기반 방식은 네트워크 관리자가 도메인 지식을 바탕으로 NSD 와 정책을 정의한다. 이 방식은 관리자의 전문성에 기반하여 목적에 특화된 정책 생성이 가능하지만 시간 소요가 크고 복잡한 환경에서는 확장성이 떨어진다. 다음으로, AI 기반 방법은 과거 네트워크 데이터를 활용하여 운용 패턴을 학습하고, 이를 통해 정책을 자동으로 추천하거나 생성하는 방식으로, 수작업 기반의 방법보다는 확장성이 개선되었지만 데이터의 품질과 양에 따라 성능이 좌우되는 한계가 있다.

정책 생성 관련 선행 연구 [3]은 관리자나 사용자가 입력한 보안 관련 의도를 규칙 기반 방법을 통해 정형화하여 SDN 컨트롤러가 이해할 수 있는 네트워크 정책으로 자동 변환한다. 다음으로, Intent 기반 네트워크 프로비저닝 선행 연구 [4]는 LSTM 모델을 활용하여 사용자의 Intent 에서 네트워크 요구사항을 추출하고, 이를 정책으로 변환한다. 마지막으로, NSD 생성 관련 선행 연구 [5]는 ETSI NFV 표준 [6]에서 NSD 형식을 기반으로 네트워크 서비스 구성 요소를 정의하여 SDN 컨트롤러 인터페이스에 적합한 JSON 메시지로 변환한다.

Ⅲ. LLM 기반 정책 및 NSD 생성

LLM 기반 정책 및 NSD 생성 과정은 크게 4 가지 단계로 구분된다. 먼저, **1) 도메인 데이터 수집** 단계에서는 기존 네트워크 정책, NSD 사례, 네트워크 인프라 정보 등 도메인 특화 데이터를 수집하고, 이를 LLM 이 이해할 수 있는 형식으로 전처리한다. 다음으로, **2) 프롬프트 및 RAG 설계** 단계에서는 수집된 도메인 데이터를 기반으로 LLM 에 대한 프롬프트 엔지니어링 및 RAG 를 활용한다. 이러한 도메인 최적화 기법을 통해 LLM 이 사용자 및 네트워크 요구사항과 네트워크 상태 정보를 명확히 기술한 자연어 입력과 구조화된 출력 (JSON, YAML 등)을 생성할 수 있도록 한다. 그리고 **3) 정책 및 NSD 생성** 단계에서는 설계된 프롬프트를 이용해 LLM 이 자동으로 정책과 NSD 를 생성하며, 이 과정에서 후보 출력물이 도출된다. 마지막으로, **4) 결과 검증** 단계에서는 전문가 피드백이나 자동화된 검증 절차를 통해 생성된 정책 및 NSD 의 정확성과 실용성을 평가하고, 평가 결과에 따라 생성된 결과물을 수정 및 보완하여 실제 네트워크 관리 시스템에 적용 가능한 결과물을 도출한다 (그림 1).

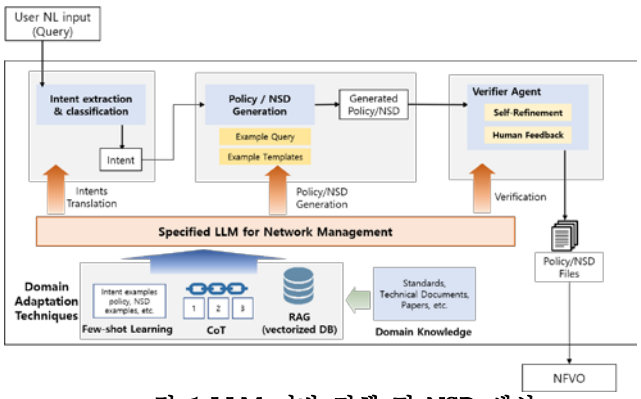

그림 1 LLM 기반 정책 및 NSD 생성

1) 도메인 데이터 수집: 먼저, 네트워크 정책 및 NSD 관련 구성(configuration), 로그 데이터, 문서 등 다양한 도메인 데이터를 수집한다. 이 단계에서는 데이터의 신뢰성과 최신성을 고려하여 수집한다. 수집된 데이터는 비정형 텍스트, 구조화된 문서, 로그 등 다양한 형식을 포함하므로, LLM 의 프롬프트 엔지니어링 및 RAG 에 활용할 수 있도록 텍스트 정제, 불용어 제거 등 전처리 과정을 거친다. 또한, LLM 을 네트워크 도메인에 최적화할 수 있도록 관련 용어 사전이나 템플릿을 구성하여 데이터의 일관성을 높이고, 네트워크 정책 및 NSD 생성에 필요한 핵심 정보를 추출하는 작업도 병행한다.

2) 프롬프트 및 RAG 설계: 전처리된 데이터를 바탕으로 LLM 이 원하는 형식의 정책 및 NSD 를 생성할 수 있도록 명확하고 구체적인 프롬프트 템플릿을 설계한다. 이때, 프롬프트는 네트워크 요구사항, 환경 변수, 제약 조건

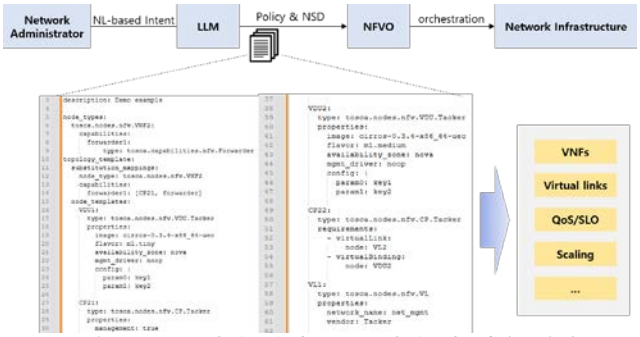

그림 2 LLM 기반 정책/NSD 생성 및 적용 예시

등의 정보를 포함하여, LLM 이 출력으로 JSON, YAML 과 같은 구조화된 데이터를 반환하도록 유도한다. 설계 과정에서는 반복적 테스트와 피드백을 통해 프롬프트를 개선하며, Few-shot Learning, Chain-of-Thought (CoT) 기법 등 최적화 기법을 사용하여 LLM 기반 생성 방법의 성능을 개선한다. 또한, 프롬프트 내에 구체적인 예시나 형식 규칙을 제시하여, LLM 이 도출할 결과물의 일관성과 정확성을 확보할 수 있도록 한다. 그리고 도메인 데이터 수집 과정에서 추출 및 문서화한 데이터를 기반으로 Retrieval-Augmented Generation (RAG)를 활용하여 LLM 의 정책 및 NSD 생성 성능을 보다 높일 수 있다.

3) 정책 및 NSD 생성: 상기 단계에서 프롬프트 엔지니어링 및 RAG 를 통해 최적화된 LLM 을 기반으로 사용자 및 네트워크 관리자의 자연어 기반 Intent 를 입력하면, LLM 은 네트워크 관리 및 운용을 위한 정책 및 NSD 를 자동으로 생성한다. 이 과정에서 설계된 프롬프트 및 RAG 를 기반으로 네트워크 관리자의 Intent 에 부합하는 정책 및 NSD 를 생성하고, 이를 통해 결과 검증 단계에서 네트워크 적용 가능성을 검토한다.

4) 결과 검증: 최종적으로 생성된 정책 및 NSD 후보물은 네트워크 관리자 및 전문가의 검토나 자동화된 스키마 검증, 규칙 기반 검증 등을 통해 정확성과 실용성을 평가한다. 이 단계에서는 생성된 정책 및 NSD 의 구문 오류 (syntax error), 의미적 충돌 등을 점검하고, 필요에 따라 후처리 과정을 거쳐 수정 및 보완한다. 또한, Reinforcement Learning from Human Feedback (RLHF), self-refinement 등 LLM 피드백 기법을 활용하여 프롬프트 템플릿이나 전처리 과정에 반영한다. 이를 통해 반복 실행 시 점진적으로 생성 결과를 개선시키는 피드백 루프를 구축한다.

Ⅳ. 결론 및 향후 연구

본 논문에서는 네트워크 관리 자동화를 위한 LLM 기반 정책 및 NSD 생성 방법을 제안한다. 제안하는 방법은 네트워크 도메인 특화 데이터의 수집 및 전처리, 프롬프트 엔지니어링과 RAG 기법을 통한 자동 정책 및 NSD 생성 과정을 체계적으로 설계한다. 이와 같은 개념적 설계는 LLM 의 자연어 이해 및 처리 능력을 기반으로 복잡한 네트워크 환경에서 사용자의 요구사항을 효과적으로 반영할 수 있는 잠재력을 지니며, Intent 기반 네트워크 관리 자동화 분야에 연구 방향을 제시한다.

향후 연구에서는 제안하는 LLM 기반 정책 및 NSD 생성 방법을 실제로 구현하고, 네트워크 환경 적용 가능성을 높이기 위한 추가 연구를 진행한다. 이후 다양한 네트워크 인프라와 운용 데이터를 대상으로 제안하는 방법의 성능과 신뢰성을 정량적으로 평가하여, 제안하는 방법의 적합성과 활용 가능성을 검증한다.

ACKNOWLEDGMENT

이 논문은 2025 년도 정부(과학기술정보통신부, 경찰청)의 재원으로 정보통신기획평가원 및 과학치안진흥센터의 지원을 받아 수행된 연구임 (RS-2024-00392332, 6G 네트워크 통합 지능평면 기술 개발, RS-2022-PT000186, 경찰건강 스마트관리 사업).

참 고 문 헌

[1] Umoga, Uchenna Joseph, et al. "Exploring the potential of AI-driven optimization in enhancing network performance and efficiency," Magna Scientia Advanced Research and Reviews, vol. 10, no. 1, pp. 368-378, 2024.

[2] R. Boutaba et al. "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," Journal of Internet Services and Applications, vol. 9, no. 1, pp. 1-99, 2018.

[3] A. Chowdhary, A. Sabur, N. Vadnere and D. Huang, "Intent-Driven Security Policy Management for Software-Defined Systems," in IEEE Transactions on Network and Service Management, vol. 19, no. 4, pp. 5208-5223, Dec. 2022.

[4] Mahtout et al. "Using machine learning for intent-based provisioning in high-speed science networks", Proceedings of the 3rd international workshop on systems and network telemetry and analytics, June 2020.

[5] F. Paganelli, F. Paradiso, M. Gherardelli and G. Galletti, "Network service description model for VNF orchestration leveraging intent-based SDN interfaces," 2017 IEEE Conference on Network Softwarization (NetSoft), pp. 1-5, Bologna, Italy, 2017.

[6] ETSI, "Network Functions Virtualisation (NFV); Management and Orchestration; Report on Management and Orchestration Framework," ETSI GR NFV-MAN 001 V1.2.1, 2021.

CRDkit: Kubernetes에서 PostgreSQL 커스텀 리소스의 선언적 관리와 LLM 기반 생성을 위한 Bash 인터페이스

우마르 마흐무드, 송왕철
umarmahmood637@gmail.com, philo@jejunu.ac.kr

CRDkit: A Bash Interface for Declarative Management and LLM-Based Generation of PostgreSQL Custom Resources in Kubernetes

Umar Mahmood, Wang-Cheol Song
Department of Computer Engineering, Jeju National University

Abstract

Custom Resource Definitions (CRDs) have become central to extending Kubernetes functionality, particularly for managing stateful services such as PostgreSQL clusters using the CrunchyData Postgres Operator. However, interacting with CRDs often requires precise YAML authoring and familiarity with complex schemas, making day-to-day management tasks cumbersome for DevOps engineers. To address this challenge, we present *crdkit*, a Bash-based command-line tool designed to simplify the management of PostgreSQL CRDs. The tool provides support for updating replica counts, resizing volumes, deleting clusters, and inspecting resource status—all through a terminal interface with no additional dependencies beyond kubectl and jq. In addition to these declarative operations, CRDkit introduces an LLM-powered mechanism for automatically generating CRD manifests. By integrating with a local Ollama server, users can describe a cluster configuration in simple terms, and the tool generates a valid YAML manifest ready for application to the cluster. We evaluated CRDkit in a local Kubernetes environment, demonstrating its ability to manage CRDs effectively while reducing the manual burden of configuration. Our findings demonstrate that lightweight tooling, combined with local language models, provides a practical and accessible approach to CRD lifecycle management in constrained or offline scenarios.

Ⅰ. Introduction

The adoption of Kubernetes as a platform for deploying and managing complex applications has led to the widespread use of Custom Resource Definitions (CRDs)

[1]. CRDs enable users to define and control domain-specific resources declaratively, allowing Kubernetes to manage components such as databases, caches, and queues with consistency and automation. One widely used example is the CrunchyData Postgres Operator, which exposes a CRD-based interface for provisioning and managing PostgreSQL clusters directly within Kubernetes [2]. While this declarative model offers flexibility and alignment with cloud-native principles, it also introduces challenges. Writing CRDs manually can be error-prone and requires familiarity with detailed YAML schemas [3]. Tasks such as scaling replicas, adjusting storage, or inspecting cluster status often require the use of verbose commands or manual file edits [4]. In many cases, users are left navigating between kubectl commands, documentation, and YAML files just to perform basic cluster operations.

To address these challenges, we present a tool called crdkit. crdkit is a lightweight, Bash-based command-line utility that provides a simplified interface for managing PostgreSQL CRDs in Kubernetes. It enables users to perform core operations, such as updating replica counts, resizing data volumes, deleting clusters, and retrieving cluster status, all through a guided CLI menu with input validation and status feedback. Built with only Bash, kubectl, and jq, the tool is portable and compatible with minimal environments, including air-gapped or offline clusters.

In addition to basic CRD management, crdkit introduces an integration with locally hosted large language models (LLMs) using the Ollama runtime. This enables users to generate new CRD manifests by describing their desired configuration in natural language [5]. The tool constructs a structured prompt, sends it to the local model (e.g., codegemma), and applies the returned YAML directly to the cluster. This approach eliminates the need for manual YAML authoring and reduces the complexity of getting started with PostgreSQL CRDs [6].

This paper describes the motivation behind crdkit, its design and implementation, and the integration of LLMs for manifest generation. We evaluated the tool in a local Kubernetes environment and demonstrated how lightweight scripting, combined with local AI capabilities, can improve the usability and accessibility of CRD-based workflows [7].

Ⅱ. Motivation and Related Works

Managing stateful applications in Kubernetes typically involves the use of operators and Custom Resource Definitions (CRDs), which expose application-specific configuration and lifecycle management through Kubernetes-native interfaces [1]. While this approach offers powerful automation capabilities, it often shifts complexity from imperative scripts to declarative manifests, which can be

challenging to compose and manage, especially for users unfamiliar with the underlying CRD schema or YAML syntax.

This challenge is particularly evident in the case of the CrunchyData Postgres Operator, which defines comprehensive CRDs for PostgreSQL cluster provisioning, scaling, high availability, and backups [2]. Although these resources can be managed declaratively using kubectl, real-world usage often involves editing long YAML manifests, applying patches via complex JSONPath expressions, or consulting operator documentation for schema references. For day-to-day operational tasks such as updating replica counts, changing storage parameters, or deleting clusters, there is a lack of lightweight tools that simplify this workflow without introducing new dependencies or requiring knowledge of Helm or GitOps [4].

Existing solutions, such as Helm and Kustomize, offer templating and configuration management capabilities, but they still rely on templated YAML and predefined chart structures [3]. Tools like Lens provide graphical interfaces for interacting with CRDs, but they require a graphical desktop environment and additional setup. CLI-based alternatives, such as kubectl, krew plugins, and k9s, provide generic interfaces for resource inspection but do not abstract or automate CRD-specific tasks.

Recent advances in large language models (LLMs) have introduced new opportunities for automating infrastructure workflows. LLMs can generate syntactically correct configuration files, explain complex YAML structures, and assist in authoring templates based on natural language input [5], [6]. However, most LLM-based tools are cloud-hosted, depend on internet access, and lack direct integration with local developer tools or Kubernetes environments. Integrating local LLMs such as those powered by Ollama presents an opportunity to combine AI-generated configuration with fully offline, script-based tooling [7]. CRDkit was developed to bridge this gap. It combines a lightweight Bash interface with kubectl and jq to provide direct control over PostgreSQL CRDs while incorporating natural language–driven CRD generation through a locally hosted LLM. This combination enables users to manage and provision PostgreSQL clusters through a single terminal interface, eliminating the need to write YAML files manually, rely on templates, or interact with cloud APIs. The goal is not to replace full-featured operators or GitOps pipelines but to enable a portable and accessible alternative for CRD-centric workflows.

IV. Algorithm and its Implementation

The core functionality of *crdkit* is implemented as a modular Bash script that interacts with Kubernetes using kubectl and manipulates structured data using the jq command. The tool provides an interactive, menu-driven interface that allows users to perform predefined operations on PostgreSQL clusters defined as Kubernetes Custom Resource Definitions (CRDs) via the CrunchyData Postgres Operator. These operations include updating the number of replicas, resizing persistent volumes, deleting CRDs, checking cluster status, and generating new CRD manifests using an LLM backend.

Each menu option corresponds to a specific function in the script, and the user is prompted for required parameters such as cluster name, namespace, and resource values. All user input is validated against regular expressions to ensure syntactic correctness and prevent malformed API requests. For standard CRD operations, crdkit performs the following sequence:

- Validation – The script first checks for the presence of required tools (kubectl, jq) and verifies that the specified Kubernetes namespace exists.
- Existence Check – Before any operation is applied, the script confirms whether the target CRD (postgresclusters) exists in the given namespace.
- CRD Mutation – For patching values (e.g., replicas, volume size), crdkit uses the kubectl patch command with JSON
- Patch syntax to replace the desired field. This allows atomic updates without the need to reapply full manifests.
- Deletion and Status Retrieval – The script supports the safe deletion of a PostgreSQL CRD after user confirmation and can retrieve the full YAML definition of an existing CRD using the' kubectl get' command.

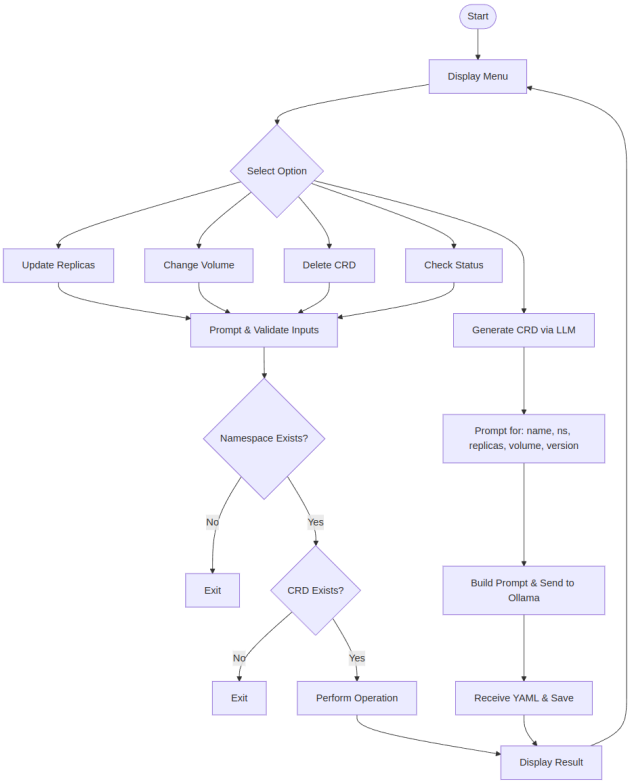

Fig 1. Flowchart illustrating the operational flow of *crdkit*.

In addition to these standard operations, crdkit includes an AI-assisted CRD generation feature using a locally running LLM served via the Ollama HTTP API. This process involves the following steps:

- Prompt Construction – Based on user inputs (cluster name, namespace, replicas, volume size, PostgreSQL version), the script constructs a prompt describing the desired CRD in natural language.
- API Request – The prompt is sent as part of a JSON payload to the Ollama server using curl. The model specified (e.g., codegemma:7b) processes the input and returns a YAML string representing the CRD.
- YAML Handling – The output is parsed using jq, saved to a temporary .yaml file, and then applied directly to the Kubernetes cluster using kubectl apply.
- Error Checking – If the LLM fails to return a valid response or if the CRD fails to apply, appropriate error messages are displayed, and no changes are committed.

The overall flow of crdkit is depicted in Fig. 1. Upon execution, the script initiates a menu-driven interface that allows the user to select from multiple PostgreSQL CRD operations. For standard actions such as updating replicas,

modifying volume size, deleting a cluster, or retrieving status, the tool performs input validation followed by checks for namespace and CRD existence. If all conditions are satisfied, the corresponding kubectl command is executed. Alternatively, when the LLM-based CRD generation option is selected, the tool collects high-level configuration parameters from the user, constructs a prompt, and sends it to a local Ollama API. The returned YAML is saved to a file and applied directly to the Kubernetes cluster. In both cases, the result of the operation is displayed, and control is returned to the main menu loop.

V. Results

We evaluated crdkit in a local development environment using Kind configured with the CrunchyData Postgres Operator and a locally hosted Ollama LLM runtime. The core functions of the tool—including updating replicas, resizing persistent volumes, deleting CRDs, and retrieving cluster status—were tested across multiple PostgreSQL cluster instances deployed via CRDs. All operations were executed correctly using kubectl, and the patching logic functioned as expected via JSON Patch expressions.

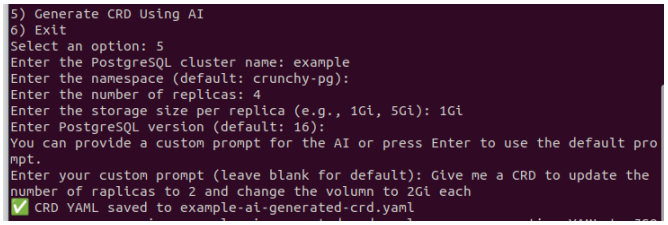

Fig 2. User prompt in CRDkit CLI for LLM-assisted generation of a PostgreSQL CRD manifest.

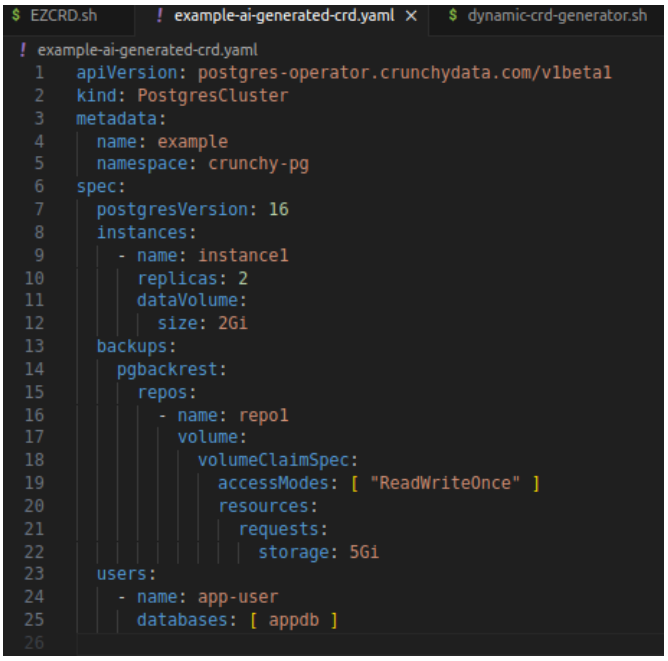

Fig 3. YAML output generated by the local LLM based on the user's input prompt.

To assess the AI-assisted CRD generation capability, we executed a series of tests in which users specified input parameters such as cluster name, namespace, replica count, volume size, and PostgreSQL version. The resulting YAML manifests generated by the LLM were syntactically valid and applied successfully in 9 out of 10 test cases without requiring modification. Minor formatting adjustments (e.g., whitespace or extra metadata) were needed in rare cases, but the overall structure consistently adhered to the expected CRD schema.

Figure 2 illustrates an example input session where the user provides natural language instructions to modify a cluster configuration. Figure 3 shows the corresponding YAML manifest generated by the local LLM in response to the prompt. In a comparison between manual authoring and LLM-assisted generation, the use of crdkit reduced configuration time by 50–70%, depending on the complexity of the manifest. For example, creating a three-replica cluster with a specific storage configuration using the LLM workflow took under 20 seconds end-to-end, whereas manual YAML authoring and validation required over one minute.

These results confirm that crdkit delivers reliable automation for CRD lifecycle operations and effectively integrates local LLMs to simplify resource generation, particularly in scenarios where lightweight, scriptable tooling is preferred over full-fledged operators or UI-based interfaces.

VI. Conclusion

This work introduces crdkit, a lightweight command-line utility designed to simplify the management and creation of PostgreSQL Custom Resources in Kubernetes environments. By leveraging basic shell scripting with kubectl and jq, crdkit provides a focused interface for core lifecycle operations such as scaling, volume resizing, deletion, and status inspection of CRD-managed PostgreSQL clusters. In addition, the integration of local large language models through Ollama enables the generation of syntactically valid CRD manifests from user-specified parameters, reducing the reliance on manual YAML authoring and schema navigation.

The tool is intentionally minimal, requiring no external services or cloud dependencies, and is well-suited for use in constrained environments, offline clusters, and scenarios where users need direct, terminal-based control of Kubernetes resources. Experimental evaluation in a local Kubernetes environment demonstrated that crdkit can reduce operational complexity while remaining fully compatible with existing operator workflows. While crdkit does not aim to replace full-fledged GitOps systems or Kubernetes-native controllers, it provides a practical and extensible foundation for CRD-centric workflows that benefit from local automation and LLM-driven authoring. Future enhancements may include diff-based patching, Git integration, support for additional CRD types, and dynamic schema validation of AI-generated manifests.

References

[1] M. Luksa, Kubernetes in Action, Manning, 2018.
[2] CrunchyData, "Postgres Operator Documentation." [Online]. Available https://access.crunchydata.com/documentation/postgres-operator/latest

[3] Merkel D. Docker: lightweight linux containers for consistent development and deployment. Linux j. 2014 Mar 2;239(2):2.

[4] K. Hightower, B. Burns, and J. Beda, *Kubernetes: Up and Running: Dive into the Future of Infrastructure*, 2nd ed., O'Reilly Media, 2019.

[5] Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A. Language models are few-shot learners. arXiv preprint arXiv:2005.14165. 2020 Jul 24;1:3.

[6] Jiang J, Wang F, Shen J, Kim S, Kim S. A survey on large language models for code generation. arXiv preprint arXiv:2406.00515. 2024 Jun 1.

[7] Li Z, Peng B, He P, Yan X. Evaluating the instruction-following robustness of large language models to prompt injection. arXiv preprint arXiv:2308.10819. 2023 Aug

엣지 컴퓨팅 기반 단안 깊이 추정 모듈과 VSLAM 통합 시스템 설계

유경민, 남승우, 박재원, 백의준, 김지민, 김명섭*

고려대학교, *고려대학교

rudals2710@korea.ac.kr, nam131119@korea.ac.kr, 2018270614@korea.ac.kr, pb1069@korea.ac.kr, illiard1209@korea.ac.kr, *tmskim@korea.ac.kr

Design of an Edge Computing-Based Monocular Depth Estimation Module and VSLAM Integrated System

Yu Gyeong Min, Nam Seung Woo, Park Jae Won, Baek Ui Jun, Kim Ji Min, Kim Myeong Sub*

Korea Univ., Korea Univ., Korea Univ., Korea Univ., Korea Univ.,*Korea Univ.

요 약

본 논문은 엣지 디바이스에서 ORB-SLAM2를 실행하고, 클라우드 서버에서 Depth Anything v2를 병렬로 수행하는 분산형 VSLAM 아키텍처를 제안한다. 제안된 구조는 GPU가 없는 저전력 로봇 환경에서도 고정밀의 단안 깊이 추정을 가능하게 하며, 전체 SLAM 시스템의 정밀도와 효율을 동시에 향상시킨다.TUM RGB-D 데이터셋 기반 실험 결과, ORB-SLAM2 단독 사용 시 평균 절대 궤적 오차(ATE)는 0.442m였으나, 제안한 구조에서는 0.024m로 94.6% 감소하였다. 또한, 저조도 환경(freiburg3_nostructure_texture_far)이나 텍스처가 부족한 장면에서도 ATE가 0.041m에서 0.029m로 감소하여, 다양한 조건에서도 높은 안정성과 정밀도를 유지함을 확인하였다. 본 구조는 자율주행, 실내 내비게이션 등 실시간성과 정밀성이 요구되는 응용 분야에 실용적인 솔루션으로 적용 가능함을 입증한다.

Ⅰ. 서 론

비전 기반 동시 위치 추정 및 지도 작성(VSLAM, Visual Simultaneous Localization and Mapping)은 카메라 센서를 이용하여 주변 환경을 인식하고, 이동체의 위치를 추정하며, 3차원 지도를 구축하는 핵심 기술로, 자율주행, 로봇 내비게이션, 증강현실(AR) 등 다양한 응용 분야에서 활용된다. 특히, 단일 카메라만을 사용하는 단안 VSLAM(Monocular VSLAM)은 하드웨어 구성의 단순성과 비용 효율성 측면에서 장점을 가지며, 소형 로봇이나 저전력 디바이스 기반 플랫폼에 적합한 구조로 주목받고 있다. 그러나 단안 VSLAM은 깊이 정보를 직접적으로 측정할 수 없기 때문에 스케일 모호성(Scale Ambiguity)과 누적 오차(Scale Drift) 문제가 발생하며, 이로 인해 지도 왜곡 및 위치 추정 정확도의 저하(Scale Drift)가 발생하는 한계를 지닌다. ORB-SLAM2 [1]와 같은 대표적인 단안 VSLAM 시스템에서도 이러한 문제가 반복적으로 보고되고 있다.

이를 보완하기 위해 최근에는 단안 이미지 기반의 딥러닝 기반 깊이 추정 기법들이 제안되고 있으나, 대부분의 최신 모델들은 고성능 GPU 자원을 필요로 하며, 메모리 및 연산량이 크기 때문에 엣지 환경에서는 실시간 적용이 어렵다.

본 논문에서는 ORB-SLAM2[1]를 엣지 디바이스에서 실행하고, 선택된 KeyFrame만 클라우드로 전송하여 Depth Anything v2[2] 기반 단안 깊이 추정을 수행하는 분산형 협업 구조를 제안한다. 이 구조는 연산 자원을 분산하고 통신 효율을 극대화하며, 엣지 기반 저전력 로봇 시스템에서도 높은 정확도의 실시간 VSLAM 구현을 가능하게 한다.

Ⅱ. 본론

Table 1. 시스템 구성 및 사양

구성요소	사양	역할
엣지장치	NVIDIA Jetson AGX Orin (32GB), 8-core ARM CPU	VSLAM (ORB-SLAM2) 실행
로봇센서	단안 카메라(30fps)	환경 데이터 수집
클라우드 서버	NVIDIA GeForce RTX 4090 x4 (CUDA 12.2)	단안 깊이 추정 (Depth Anything v2)

2.1 전체 아키텍처 개요

Figure 1은 제안하는 시스템의 전체 구조를 보여준다. 본 시스템은 엣지 기반 ORB-SLAM2[1] 처리 모듈, 클라우드 기반 단안 깊이 추정 모듈, 그리고 양방향 통신 및 통합 최적화 모듈의 세 가지 주요 구성 요소로 이루어져 있다. ORB-SLAM2[1]는 엣지 디바이스에서 실시간으로 동작하며, 시스템 내부에서 선택된 주요 KeyFrame만이 클라우드로 전송된다. 클라우드에서는 Depth Anything v2[2]를 통해 정밀한 깊이 맵을 생성하고, 이를 다시 엣지로 전송하여 SLAM 시스템에 통합하게 된다. 이러한 구조는 연산 부하를 엣지와 클라우드 간에 효과적으로 분산시키고, 통신 대역폭을 절감하는 동시에, 높은 정확도와 실시간성을 모두 확보할 수 있는 장점을 갖는다.

2.2 엣지 기반 ORB-SLAM2 처리 모듈

엣지 단에서는 NVIDIA Jetson AGX Orin과 같은 ARM 기반 장치를 사용하여 ORB 특징점의 추출 및 추적, KeyFrame 선별과 전송, 그리고 초기화 및 로컬 맵 관리 등의 기능을 수행한다. 전체 680프레임 중 약

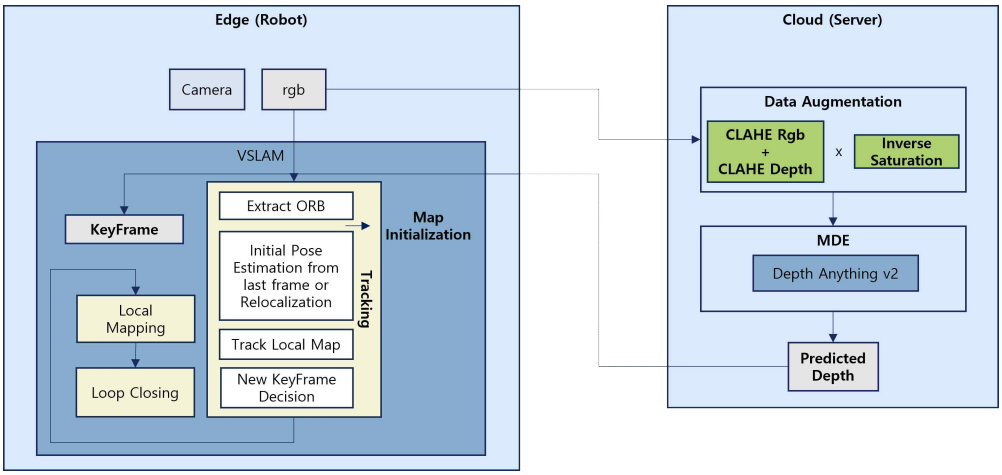

Fig 1. System Architecture

15~20%만이 KeyFrame으로 클라우드에 전송되며, 초기화에 사용되는 프레임은 약 8개로 제한되어 통신량은 평균적으로 95% 이상 절감된다. 클라우드에서 반환된 깊이 정보는 ORB-SLAM2[1]의 초기화 및 로컬 맵 생성 과정에 통합되어 스케일 정합도를 높이고 누적 오차를 줄이는 데 기여한다.

2.3 클라우드 기반 단안 깊이 추정 모듈

클라우드 서버는 NVIDIA RTX 4090 GPU 네 개를 활용하여 고성능 연산을 수행한다. 클라우드는 엣지에서 전송된 KeyFrame을 수신하고 전처리하며, Depth Anything v2[2]를 활용해 단안 영상으로부터 깊이 맵을 정밀하게 추정한 후 이를 다시 엣지로 전송한다. 클라우드 측에서 생성된 깊이 맵은 ORB-SLAM2[1]의 KeyFrame에 통합되며, 이로 인해 맵포인트 초기화 및 최적화 과정에서 더욱 정밀한 깊이 정보를 제공하게 된다.

Table 2 Result of Our System

	Non cloud	Edge Cloud Conmputing			
항목	non	All	Initial	KeyFrame	ours
ATE(m)	0.044	0.19	0.10	0.11	**0.024**
통신 데이터량	0	680	8	57	65
실시간성 (FPS)	34.8	2.5	**28.5**	21.2	18.8
초 기 화 실패율	높음	높음	중간	높음	**낮음**

2.4 양방향 통신 및 최적화 전략

본 시스템은 실시간 처리를 유지하기 위해 다양한 전략을 적용하였다. 첫째, 모든 프레임이 아닌 의미 있는 프레임만을 선택적으로 클라우드로 전송함으로써 통신 지연과 데이터 전송량을 최소화하였다. 둘째, 클라우드에서 반환된 깊이 정보는 ORB-SLAM2[1]의 맵포인트 생성, Bundle Adjustment, 루프 클로징 등 다양한 처리 단계에 통합되어 전체적인 정밀도 향상에 기여한다.

비교 실험 결과는 Table 2에 정리되어 있다. 비교 대상에는 Non-Cloud 방식, 모든 프레임을 클라우드로 전송하는 방식(All), 초기화 프레임만 전송하는 방식(Initial), KeyFrame만 전송하는 방식(KeyFrame), 그리고 제안 방식(Ours)이 포함된다. 절대 궤적 오차(ATE)는 제안 방식이 0.024m로 가장 낮았으며, 통신 데이터량은 전체 프레임 중 약 9.5%인 65프레임만 전송되어 효율적이다. 실시간성(FPS) 측면에서도 제안 방식은 평균 18.8 FPS로, 640×480 해상도 기준 실제 응용 환경에서의 실시간 처리를 만족시킨다. 초기화 실패율 역시 제안 방식이 가장 낮게 나타났다.

2.5 성능 비교 요약

본 시스템의 성능은 TUM RGB-D 벤치마크[3]를 기반으로 기존 ORB-SLAM2[1]와 비교되었다. 이 벤치마크는 휴대용 Kinect 카메라로 촬영된 컬러 영상과 깊이 영상을 포함하며, 30Hz 주기로 수집된 640×480 해상도의 이미지 시퀀스를 제공한다. 카메라의 실제 이동 경로는 고정밀 모션 캡처 시스템을 통해 100Hz로 수집되어, SLAM 시스템의 위치 정확도를 정량적으로 평가할 수 있다. 본 연구에서는 예측 궤적과 실제 궤적 간의 절대 거리 차이를 나타내는 ATE를 주요 평가 지표로 사용하였다. 모든 실험은 VSLAM의 무작위 초기화 및 자세 최적화 특성을 고려하여 5회 반복 수행되었고, 평균값을 기준으로 평가하였다. 오차가 가장 적은 결과는 볼드체로 표기하였다.

실험 결과, 모든 프레임을 클라우드로 전송하는 방식은 ATE 0.19m, FPS 2.5로 실시간성은 매우 낮았고, 초기화 프레임만 처리하는 경우 FPS는 28.5로 높았지만 ATE는 0.10m로 정확도는 낮았다. 반면 제안하는 방식은 ATE 0.024m, FPS 18.8로 정확성과 실시간성을 모두 만족하며, 통신 프레임 수도 전체의 약 9.5%로 가장 효율적인 결과를 나타냈다. 또한 생성된 지도는 Figure 2에서 확인할 수 있듯, 단안 카메라만 사용하는 기존 방식보다 훨씬 정확한 결과를 제공하였다.

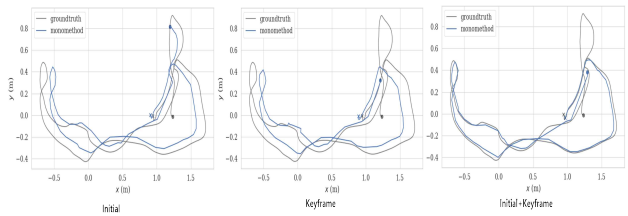

Fig. 2. ORB-SLAM3 MDE with initial,, KeyFrame, Our System(Initial+KeyFrame)

Table 3. 저텍스처 구조에서의 성능

	Mono	
	ORB SLAM [1]	Ours
fr3_nt_t_f	0.853m	**0.044m**
fr3_nt_t_n	0.473m	**0.045m**
fr3_t_nt_f	0.904m	**0.051m**
fr3_t_nt_n	0.693m	**0.056m**

2.6 추가 실험 : 저텍스처 환경 성능 평가

Table 3는 텍스처가 부족하거나 조명이 약한 환경에 대한 데이터셋에서 실험한 결과이다. 기존 ORB-SLAM2[1] 대비 절대 궤적 오차(ATE)가

90% 이상 감소하였으며, 이는 제안 구조가 실환경 적용에서도 높은 견고성과 정밀도를 유지함을 입증한다.

Ⅲ. 결론

본 연구에서는 엣지-클라우드 협업 기반의 분산형 단안 VSLAM 아키텍처를 제안하고, 이에 따른 기술적 기여를 제시하였다. 먼저, ORB-SLAM2[1]와 Depth Anything v2[2]를 각각 엣지와 클라우드에 분산 배치함으로써, 저전력 환경에서도 정밀하고 실시간성이 뛰어난 VSLAM 구조를 구현하였다. 이를 통해 실시간 분산형 VSLAM의 효과적인 구조를 실현할 수 있었으며, 엣지 단에서는 연산 부담을 줄이고, 클라우드에서는 정밀한 깊이 추정 처리를 수행함으로써 각 구성 요소의 장점을 극대화하였다. 또한, 통신 및 연산 최적화를 위해 전체 680프레임 중 초기화 프레임 8개와 키프레임 57개만을 선택적으로 전송하는 방식을 도입하였다. 이 전략을 통해 약 90% 이상의 통신 데이터량을 절감할 수 있었으며, 엣지 디바이스 상에서도 평균 18.8 FPS 이상의 성능을 유지하여 실시간 운용 기준을 만족하였다. 이를 통해 제한된 자원을 가진 환경에서도 안정적이고 효율적인 SLAM 수행이 가능함을 실험적으로 입증하였다. 정밀한 위치 추정 성능 확보 측면에서도 의미 있는 성과를 보였다. TUM RGB-D 데이터셋[3]을 활용한 실험 결과, 기존 ORB-SLAM2[1] 대비 절대 궤적 오차(ATE)가 0.442m에서 0.024m로 약 94.6% 감소하였으며, 조도가 낮거나 텍스처 정보가 부족한 환경에서도 안정적인 동작을 유지하였다. 이는 제안된 구조가 다양한 실내외 환경에서의 정밀한 위치 추정 요구를 충족시킬 수 있음을 보여준다. 종합적으로, 본 논문에서 제안한 분산형 VSLAM 구조는 GPU가 탑재되지 않은 경량 로봇 환경에서도 실질적인 적용이 가능하며, 자율주행 시스템, 모바일 로봇, 실내 내비게이션 등 다양한 응용 분야에서 실용적인 솔루션으로 활용될 수 있다.

향후 연구 방향으로는 세 가지 확장을 고려할 수 있다. 첫째, IMU나 LiDAR와 같은 다양한 센서를 융합한 멀티 센서 기반 구조로의 확장을 통해 더욱 견고하고 정밀한 위치 추정이 가능하도록 할 수 있다. 둘째, 복수의 로봇이 동시에 클라우드와 통신하는 분산 환경을 고려하여 서버 자원의 최적화 및 병렬 처리 구조에 대한 연구가 필요하다. 셋째, 네트워크 상황에 따라 프레임 전송 정책을 동적으로 조정할 수 있는 적응형 통신 알고리즘을 개발함으로써, 다양한 통신 환경에서도 안정적인 실시간성을 확보할 수 있다. 이러한 후속 연구는 본 논문에서 제안한 분산형 VSLAM 구조를 보다 다양한 실제 환경에 효과적으로 확장하고 적용할 수 있는 기반이 될 것으로 기대된다.

ACKNOWLEDGMENT

본 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(No.RS-2023-00230661, 하이브리드 양자키분배 방법 및 망 관리 기술 표준개발)과 2023년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원 (P0024177, 2023년 지역혁신클러스터육성)을 받아 수행된 연구임

참 고 문 헌

[1]MUR-ARTAL, Raul; TARDÓS, Juan D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE transactions on robotics, 2017, 33.5: 1255-1262.

[2]YANG, Lihe, et al. Depth anything v2. Advances in Neural Information Processing Systems, 2025, 37: 21875-21911.

[3]STEINBRÜCKER, Frank; STURM, Jürgen; CREMERS, Daniel. Real-time visual odometry from dense RGB-D images. In: 2011 IEEE international conference on computer vision workshops (ICCV Workshops). IEEE, 2011. p. 719-722.

ROS1과 ROS2의 구조적 차이와 자율주행 시스템 적용 관점에서의 비교 분석

박재원, 유경민, 남승우, 장윤성, 김주성, 백의준, 김명섭*

고려대학교

{2018270614, rudals2710, nam131119, brave1094, jsung0514, pb1069, *tmskim}@korea.ac.kr

Structural Differences Between ROS1 and ROS2 and Comparative Analysis in the Context of Autonomous Driving Systems

Jae-Won Park, Gyeong-Min Yu, Seung-Woo Nam, Ui-Jun Baek, Myung-Sup Kim*

Korea University

요 약

로봇 운영체제(ROS)는 로봇 분야에서 표준적인 프레임워크로 자리 잡았으며, 그 중 ROS1은 빠른 프로토타이핑에 적합한 구조로 널리 활용되었다. 그러나 중앙 마스터 노드에 대한 의존성, 실시간성 부족, 보안 미지원 등의 한계로 인해 자율주행 차량과 같은 안전이 중요한 시스템에는 적용에 제약이 있었다. 이러한 한계를 극복하기 위해 개발된 ROS2는 DDS(Data Distribution Service) 기반의 분산 아키텍처를 채택하여, 실시간성, 결함 내성, QoS 조정, 보안성 등 다양한 측면에서 기능이 향상되었다. 본 논문은 ROS1과 ROS2의 구조적 차이를 통신 구조, 프로토콜, 플랫폼 지원, 보안 측면에서 비교 분석하고, 자율주행 시스템 관점에서 각 버전의 장단점을 평가한다. 또한 Autoware.Auto와 Apex.OS 사례를 통해 ROS2의 산업적 실용성과 안정성을 입증하고, 차세대 로봇 시스템 도입을 고려하는 실무자들에게 ROS2 기반 시스템의 이해와 적용 전략 수립에 실질적인 인사이트를 제공한다.

Ⅰ. 서 론

로봇 운영체제(ROS, Robot Operating System)는 센서 처리, 제어, 시뮬레이션 등 로봇 개발에 필수적인 기능을 제공하는 오픈소스 프레임워크로, 전 세계적으로 학계와 산업계에서 널리 사용되고 있다[1]. 초기 버전인 ROS1은 빠른 프로토타입 개발과 연구에 적합한 환경을 제공하지만, 중앙 마스터 노드에 의존하는 구조적 특성으로 인해 몇 가지 중요한 한계를 지닌다. ROS1의 중앙 집중식 아키텍처는 단일 장애점(Single Point of Failure) 발생 가능성, 실시간성 부족, 보안 미지원 등의 문제를 일으킬 수 있다[1][3]. 이러한 문제들은 자율주행 차량과 같은 고신뢰성 및 실시간 처리 능력이 요구되는 시스템에 ROS1을 적용하는 데 큰 어려움을 초래한다.

이러한 한계를 극복하기 위해 개발된 ROS2는 DDS(Data Distribution Service)를 기반으로 한 분산형 통신 아키텍처를 채택하여, 시스템의 실시간성, 결함 내성, QoS(서비스 품질) 조정, 보안성 등에서 크게 개선된 기능을 제공한다[2][3]. ROS2는 중앙 마스터 노드의 의존성을 제거하고, 다양한 운영체제 및 임베디드 환경을 지원함으로써 산업 응용에 적합한 구조를 갖추고 있다. 이로 인해, ROS2는 자율주행 시스템을 포함한 고신뢰성, 고성능을 요구하는 로봇 시스템 설계에서 중요한 역할을 할 수 있게 되었다.

본 논문에서는 ROS1과 ROS2의 구조적 차이를 통신 구조, 프로토콜, 실시간성, 플랫폼 호환성, 보안성 측면에서 비교하고, 자율주행 시스템에서의 적용 사례를 통해 각 버전의 장단점을 분석한다. 이를 통해 ROS2의 기술적 우수성과 실용 가능성을 입증하고, 자율주행 시스템을 포함한 차세대 로봇 시스템 설계에 필요한 기초 자료를 제공하고자 한다.

Ⅱ. 본론

2.1.1 통신 아키텍처

ROS1은 중앙집중식 아키텍처를 채택하고 있으며, 모든 노드는 ROS 마스터(ROS Master) 노드에 등록한 후 다른 노드와 통신을 수행한다. 마스터 노드는 각 노드의 토픽, 서비스 정보 등을 관리하고, 연결을 설정하는 역할을 한다. 또한, XML-RPC 기반의 프로토콜을 통해 이름 해석(name resolution)과 통신 경로 설정을 지원한다[1]. 그러나 이러한 구조는 마스터 노드의 장애[1] 시 전체 시스템이 동작을 멈추는 단일 장애점(Single Point of Failure)의 문제를 초래하며, 시스템의 확장성에 제한을 두어 분산 노드의 확장 시 통신 지연이나 병목현상이 발생할 수 있다[1].

반면, ROS2는 DDS(Data Distribution Service)를 기반으로 한 분산형 통신 아키텍처를 채택하여, 별도의 마스터 노드 없이 각 노드가 네트워크 상에서 동적으로 발견되고 통신을 설정하는 구조를 구현하였다[1]. DDS의 discovery 메커니즘은 네트워크 내에서 노드들이 서로를 자동으로 인식하고 연결을 설정할 수 있도록 한다. 각 노드는 퍼블리셔와 서브스크라이버 관계를 설정하여 데이터를 주고받으며, 이를 통해 시스템의 확장성과 결함 내성을 크게 향상시킬 수 있다. 또한, 네트워크 구성 변화에도 유연하게 대응할 수 있어, 시스템 안정성을 높이고 분산 환경에서 더 효과적으로 운영될 수 있다[2].

이러한 구조적 차이는 로봇 시스템이 소형 단일 장비에서 동작할 때보다, 자율주행 차량과 같은 복잡한 분산 시스템에서 더욱 중요한 영향을 미친다. ROS2는 여러 ECU(Electronic Control Unit) 간의 분산 통신을 지원하며, 자율주행 시스템의 안정성과 유연성을 동시에 확보할 수 있도록 설계되었다. 이는 자율주행 차량의 다양한 센서와 제어 시스템들이 실시간

으로 데이터를 주고받고, 통합된 환경에서 원활하게 동작할 수 있도록 도와준다.

2.1.2 통신 프로토콜 및 QoS

ROS1은 기본적으로 TCPROS 및 UDPROS라는 자체 정의 프로토콜을 사용하여 통신을 수행한다[1]. TCPROS는 신뢰성 있는 통신을 보장하지만, 대역폭이 크거나 지연에 민감한 환경에서는 성능 저하가 발생할 수 있다. UDPROS는 신뢰성을 포기하고 멀티캐스트 전송을 가능하게 하지만, 실질적으로 활용되는 빈도는 상대적으로 낮다. 또한, ROS1에서는 QoS(서비스 품질)를 세부적으로 설정할 수 없어, 센서 데이터와 제어 명령 간의 우선순위를 반영하거나 지연 허용 범위를 조절하는 데 제한이 있다. 이러한 점은 실시간성과 우선순위 제어가 중요한 자율주행 시스템에서 큰 제약으로 작용할 수 있다.

반면, ROS2는 DDS의 표준 통신 프로토콜인 RTPS(Real-Time Publish-Subscribe)를 기반으로 하여, 다양한 QoS 정책을 세밀하게 설정할 수 있다[2][3]. DDS는 Reliability(신뢰성), Durability(지속성), Deadline(데드라인), History(히스토리) 등 다양한 QoS 항목을 제공하여, 애플리케이션의 요구 사항에 맞춰 통신 조건을 정밀하게 조정할 수 있다. 예를 들어, 자율주행 시스템에서 차량 제어 명령은 반드시 손실 없이 정확히 전달되어야 하므로, '높은 신뢰성 + 짧은 데드라인' QoS 설정이 가능하다. 반면, 일부 센서 데이터 스트림은 네트워크 상태에 따라 샘플 손실을 허용할 수 있도록 '베스트 에포트(Best Effort)' 방식으로 전송할 수 있다.

이러한 QoS 설정의 유연성은 자율주행 시스템과 같은 복잡한 환경에서 통신 요구 사항이 다양한 경우 강력한 확장성을 발휘한다. 예를 들어, 자율주행 차량에서는 다양한 센서들이 실시간으로 데이터를 송수신하는데, 각 데이터의 중요도와 긴급성에 맞춰 QoS를 조정함으로써 통신의 효율성을 극대화하고, 시스템의 안정성을 보장할 수 있다. 또한, ROS2의 QoS 설정은 시스템의 요구 사항에 맞는 최적화된 통신을 가능하게 하여, 통신 지연을 최소화하고, 데이터 손실을 방지하는 데 중요한 역할을 한다.

2.1.3 실시간성과 성능

ROS1은 일반적인 리눅스 환경에서 실행되며, 실시간 스케줄링과 디터미니즘을 보장하는데 어려움이 있다. 노드 간 통신 시 불필요한 데이터 복사 및 컨텍스트 전환 오버헤드가 발생하며, 시스템 부하가 증가할수록 메시지 처리 지연이 발생할 수 있다[3]. 특히 실시간성이 중요한 제어 루프에서 이러한 문제는 시스템의 정확한 동작을 방해하고, 제어 시스템의 신뢰성에 큰 영향을 미칠 수 있다.

ROS2는 실시간 임베디드 시스템의 요구를 반영하여 설계되었으며, RTOS(Real-Time Operating System)와의 호환성을 확보하였다. DDS 미들웨어는 항공, 산업 자동화 등 고신뢰성 시스템에 널리 활용되어 왔으며, 이를 바탕으로 ROS2는 낮은 레이턴시와 예측 가능한 응답 지연을 실현할 수 있다[2][3]. 또한, ROS2는 intra-process communication 기능을 활용하여 퍼블리셔와 서브스크라이버 간의 데이터 복사를 제거함으로써 성능을 최적화하고, 시스템의 처리 속도를 개선할 수 있다. 추가적으로, Lifecycle Management 기능을 통해 노드의 상태를 체계적으로 관리함으로써 시스템 초기화 시간을 단축시키고, 자원의 효율적 활용을 가능하게 한다. 이러한 특성들은 자율주행 차량처럼 실시간 제어가 중요한 시스템에서 성능을 최적화하는 데 중요한 역할을 한다.

2.1.4 멀티플랫폼 및 보안

ROS1은 주로 Ubuntu Linux에서만 안정적으로 지원되며, Windows나 기타 운영체제에서의 지원은 제한적이다. 보안 측면에서도 기본적으로 암호화, 인증, 접근 제어 기능이 제공되지 않아, 네트워크 보안에 취약한 구조를 가지고 있다[3]. SROS1이라는 실험적인 보안 계층이 존재하지만, 이는 안정성이나 적용 범위에 한계가 있어 실제 운영 환경에서의 활용에 제약이 있었다.

반면, ROS2는 초기 설계 단계부터 멀티플랫폼 지원을 고려하여, Linux 뿐만 아니라 Windows, macOS, RTOS 등 다양한 운영체제에서 실행할 수 있다. 이러한 특성 덕분에 임베디드 시스템이나 소형 보드에서도 유연하게 배포가 가능하다. 또한, ROS2는 DDS 보안 스펙에 기반한 SROS2를 통해 메시지 암호화, 노드 인증, 권한 제어 등의 보안 기능을 제공한다. 이를 통해 자율주행 차량과 같이 보안이 중요한 시스템에서 안전하고 신뢰할 수 있는 환경을 구축할 수 있다[2][6]. 특히 SROS2는 ROS2의 보안 기능을 강화하여, 민감한 데이터와 명령들이 안전하게 전달될 수 있도록 보장하며, 보안성이 중요한 산업 및 자율주행 시스템에 적합한 솔루션을 제공한다.

Table. 1은 ROS1과 ROS2의 주요 구조적 차이점을 요약 비교한 것이다.

항목	ROS1 (1세대 ROS)	ROS2 (차세대 ROS)
아키텍처	ROS 마스터 노드 필요 (중앙 집중식)	DDS 기반 분산 아키텍처 (마스터 노드 없음)
통신 프로토콜	XML-RPC 기반 TCPROS/UDPROS(전용 프로토콜)	DDS/RTPS (표준 미들웨어 프로토콜 활용)
실시간 지원	비실시간 (일반 Linux 커널, 실시간 보장 어려움)	실시간성 고려 (RTOS 지원, DDS QoS 통한 튜닝)
플랫폼 지원	주로 Ubuntu Linux	Linux, Windows, Mac, RTOS 등 멀티플랫폼
보안	기본 보안 미제공 (SROS1 실험적)	DDS 보안 지원 (SROS2 공식 지원)
기타 특징	Nodelet으로 프로세스 내 통신 일부 지원	Lifecycle 등 노드 관리 기능, 다중 DDS 벤더 지원

Table. 1. Comparison of Key Features between ROS1 and ROS2

2.2 자율주행 시스템 적용 관점에서의 비교 분석

ROS의 구조적 특성은 자율주행 시스템의 설계와 운영에 중요한 영향을 미친다. 자율주행 차량은 카메라, 라이다(LiDAR), 레이더, IMU 등 다양한 센서로부터 초당 수백 MB 이상의 데이터를 수집하며, 이를 실시간으로 분석하여 경로 계획 및 제어에 반영해야 한다. 이 과정에서 통신 지연, 데이터 손실, 시스템 다운은 곧 안전 문제로 이어질 수 있기 때문에, 자율주행 시스템에서는 신뢰성과 실시간성이 특히 중요한 요소로 작용한다.

ROS1 기반 자율주행 플랫폼인 Autoware.AI는 연구 및 시제품 개발에 적합한 환경을 제공했으나, 시스템 확장 시 발생하는 여러 문제에 직면했다. 특히 노드 수가 증가하면서 마스터 노드의 과부하, 센서 데이터 동기화 지연, 통신 병목 현상 등으로 인한 성능 저하가 문제가 되었다[5]. 예를 들어, ROS1에서는 토픽 충돌이나 노드 재연결 실패 등이 종종 발생하며, 초기화 과정에서 전체 시스템의 지연을 초래하기도 한다.

이러한 한계를 해결하기 위해 Autoware 프로젝트는 ROS2 기반의 차세대 플랫폼인 Autoware.Auto를 개발하였다. ROS2는 DDS 기반의 분산 통신 구조를 채택하여, 실시간 분산 통신 환경을 구축하고, 노드 간 의존성을 줄여 모듈화를 강화하였다[5]. 또한, ROS2에서는 QoS 설정을 통해 센서 종류별로 데이터 우선순위를 조절함으로써, 차량 제어에 필수적인 핵심 정보의 지연 및 손실을 최소화할 수 있었다. 이로 인해 자율주행 시스템에서 발생할 수 있는 위험 요소를 줄이며, 실시간 처리 성능을 향상시

켰다.

또한, 미국의 Apex.ai는 ROS2를 기반으로 한 상용 자율주행 소프트웨어 프레임워크인 Apex.OS를 개발하고, ISO 26262 ASIL-D 등급의 기능 안전 인증을 목표로 하여 ROS2의 안정성과 실시간성을 산업적으로 입증하고 있다[4]. Apex.OS는 ROS2 API와 완벽하게 호환되며, 하드 실시간성 보장을 통해 고안정성 자율주행 플랫폼을 구축하고 있다. 이 시스템은 이미 일부 자동차 제조사에 채택되어 실제 자율주행 시스템에서 활용되고 있다.

이러한 사례들은 ROS2가 ROS1의 구조적 한계를 극복하고, 자율주행 시스템에 요구되는 성능, 안정성, 보안성을 충족할 수 있는 기술적 기반을 제공하고 있음을 잘 보여준다. 특히 DDS 기반의 QoS 설정, 분산 discovery 메커니즘, 암호화 통신 등의 기능은 실질적인 시스템 구현에서 중요한 역할을 하며, ROS2의 확산과 산업 적용을 가속화하는 데 중요한 요소로 작용할 것이다[2][6].

Ⅲ. 결론

본 논문에서는 ROS1과 ROS2의 구조적 차이를 비교하고, 자율주행 시스템 적용 관점에서 두 프레임워크의 장단점을 심도 있게 분석하였다. ROS1은 중앙 마스터 노드에 의존하는 구조로 인해 실시간성 부족, 단일 장애점 발생 가능성, 보안 미지원 등의 문제를 가지고 있으며, 이러한 문제들은 안전성과 확장성이 중요한 자율주행 시스템에 적용하는 데 큰 제약을 주었다.

반면, ROS2는 DDS 기반의 분산 아키텍처를 채택하여 노드 간 통신 구조를 근본적으로 개선하고, QoS 조절, 실시간성 보장, 멀티플랫폼 지원, 보안 강화 등 다양한 측면에서 기술적 진보를 이루었다. 특히 Autoware.Auto와 Apex.OS와 같은 사례들은 ROS2가 자율주행 시스템에서 요구되는 실시간성, 안정성, 기능 안전성 등을 충족시키며 실제로 활용되고 있음을 증명한다. ROS2는 자율주행 분야뿐만 아니라 다양한 로봇 시스템에 실질적인 기술적 기반을 제공하고 있으며, 이는 ROS1이 해결하지 못한 문제들을 효과적으로 개선한 결과이다.

다만, ROS2는 아직 완전한 실시간성 확보나 DDS 설정의 복잡성 등의 문제를 안고 있으며, 이러한 과제를 해결하기 위한 추가 연구가 필요하다. 그러나 ROS2는 ROS1의 한계를 극복하며 로보틱스 산업, 특히 자율주행 시스템의 요구를 충족하는 방향으로 발전하고 있다. 향후 ROS2의 성능 최적화, 사용자 친화적인 개발 환경 개선, 그리고 ROS1에서 ROS2로의 전환을 위한 마이그레이션 도구 및 전략 개발이 중요한 과제가 될 것이다. 본 연구는 ROS를 활용하는 로봇 개발자 및 자율주행 시스템 설계자들에게 실질적인 기술적 통찰을 제공하며, ROS2의 도입 및 적용 전략 수립에 중요한 참고자료가 될 것으로 기대된다. 또한, ROS2의 지속적인 발전과 보급이 로봇 기술과 자율주행 분야에서 중요한 혁신을 이끌어갈 것으로 전망된다.

ACKNOWLEDGMENT

이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(00235509, ICT융합 공공 서비스● 인프라의 암호화 사이버위협에 대한 네트워크 행위기반 보안관제 기술 개발)과 2023년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행된 연구임 (P0024177, 2023년 지역혁신클러스터육성)의 지원을 받아 수행된 연구임.

참 고 문 헌

[1] ROS.org, “Robot Operating System,” https://www.ros.org

[2] OMG, “Data Distribution Service (DDS),” https://www.dds-foundation.org

[3] M. Macenski et al., “Robot Operating System 2: Design, Architecture, and Uses in the Wild,” Science Robotics, 2022.

[4] Apex.AI, “ROS 2 on Self-Driving Cars,” Apex.ai Blog, 2020. [Online]. Available: https://www.apex.ai/post/ros-2-on-self-driving-cars

[5] K. Tokuda, et al., “Autoware on ROS 2: Design, Architecture, and Lessons Learned,” Autoware Foundation, 2021.

[6] OMG, “DDS Security Specification,” Version 1.1, Object Management Group (OMG), 2020.

볼류메트릭 영상에서의 적응형 스트리밍 기술 활용을 위한 동향 연구

유상우 [1], 홍원기 [2]

포항공과대학교 인공지능대학원 [1]
포항공과대학교 컴퓨터공학과 [2]

{rswoo, jwkhong}@postech.ac.kr

A Survey of Adopting Adaptive Bitrate Streaming for Volumetric Video

Sangwoo Ryu[1], James Won-Ki Hong[2]
Graduate School of Artificial Intelligence, POSTECH[1]
Department of Computer Science and Engineering, POSTECH[2]

요 약

적응형 스트리밍은 미디어 스트리밍에서 효율적인 네트워크 자원 사용과 안정적인 미디어 서비스 제공을 위해 필수적인 기술이다. 실감형 미디어 콘텐츠인 볼류메트릭 영상은 2D 영상과 다른 특성을 가지기 때문에 적응형 스트리밍 기술 적용을 위해서는 추가로 고려해야 하는 요소들이 생긴다. 본 논문에서는 볼류메트릭 영상 스트리밍에서 적응형 스트리밍 기술을 적용하기 위해, 2D 영상과의 대표적인 차이점인 뷰포트 적응, 압축 코덱, 전송 프로토콜 측면을 중심으로 관련 연구를 소개한다. 이를 통해 해당 분야 연구의 주요 과제와 현재 해결 방향에 대한 통찰을 제공하며, 향후 볼류메트릭 영상의 적응형 스트리밍 기술의 발전을 위한 이해를 돕고자 한다.

Ⅰ. 서 론

적응형 스트리밍(Adaptive Bitrate Streaming, ABR) 기술은 네트워크 대역폭과 사용자 기기 성능에 맞춰 미디어의 전송 품질을 동적으로 조절하는 기술이다. 대표적으로는 HTTP 기반의 HLS(HTTP Live Streaming) [1]과 MPEG-DASH(Dynamic Adaptive Streaming over HTTP) [2]가 있다. 이는 비디오를 여러 비트 전송률로 인코딩한 후, 세그먼트 단위로 전송하는 방식으로 작동하며, 동적인 네트워크 상황에서도 버퍼링을 최소화하고 사용자 경험을 향상시킬 수 있어 대부분의 동영상 서비스에서 이를 활용하고 있다.

최근 확장 현실(Extended Reality, XR) 기술의 발전으로, 확장 현실 공간에서 활용되는 볼류메트릭 영상(Volumetric Video) 콘텐츠가 주목을 받고 있다. 볼류메트릭 영상은 사용자가 6-DoF(Degree of Freedom)로 콘텐츠를 관람할 수 있어 사용자에게 높은 몰입감을 줄 수 있고, 엔터테인먼트, 교육 등 다양한 분야에서 활용되고 있다. 하지만 볼류메트릭 영상은 2D 영상과 다르게 3 차원 공간 정보를 담아야 하기 때문에 대용량의 데이터를 처리해야 하며, 가공되지 않은 볼류메트릭 데이터는 수 Gbps 에서 수십 Gbps 까지 사용한다 [3]. 따라서 안정적인 서비스를 위해서는 이를 효율적으로 전송 및 처리하기 위한 기술 연구가 필수적이다.

볼류메트릭 영상을 적응형 스트리밍하는 경우에는, 2D 영상과는 다른 고려 사항들이 생긴다. 대표적으로 2D 영상에서는 단일 시점에서의 데이터만 제공한다면, 볼류메트릭 영상에서는 3 차원 공간에서 사용자가 원하는 시점과 위치의 데이터를 제공해야 한다는 것이다. 두 번째로는 코덱의 차이가 있다. 2D 영상은 이미 많은 연구 및 서비스가 이루어지고 있어 원하는 화질 및 비트 전송률로의 인코딩이 용이하고, 단일 스트림으로 여러 화질을 표현하는 코덱도 존재한다 [4]. 반면 볼류메트릭 영상은 상대적으로 연구가 덜 이루어졌으며, 기존에 3D 데이터 표현을 위해 주로 사용되는 포인트 클라우드(Point Cloud), 메쉬(Mesh)를 비롯해, 최근에는 NeRF(Neural Radiance Field) [5], 3D/4D Gaussian Splatting (3D/4DGS) [6] 등 새로운 표현 방식도 제안되고, 각 표현 방식마다 특성이 다르기 때문에 적응형 스트리밍 기술 적용의 복잡도를 높이고 있다.

따라서 볼류메트릭 영상에 적응형 스트리밍을 성공적으로 적용하기 위해서는 2D 영상에서의 경우와 다른 부분들을 이해하고, 관련한 연구 현황을 파악하는 것이 중요하다. 본 논문에서는 볼류메트릭 영상에서의 적응형 스트리밍에 대한 통찰을 제공하기 위해 2D 영상의 경우와 차이점을 중심으로 대표적인 연구들과 최신 동향을 제시한다. 그림 1 은 볼류메트릭 영상을 위한 적응형 스트리밍에서 본 논문에서 다루는 범위를 보여준다.

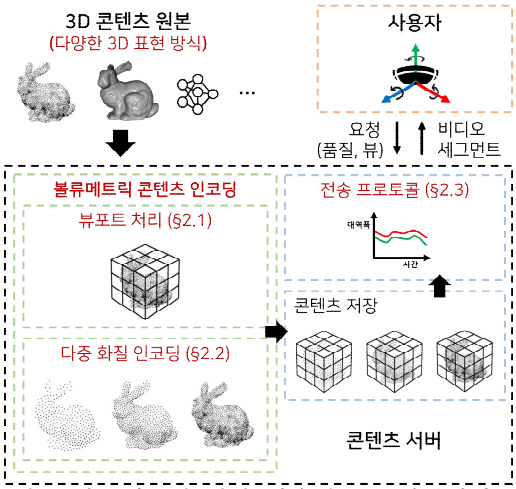

그림 1 볼류메트릭 영상 적응형 스트리밍에서 본 연구의 분석 범위

Ⅱ. 본문

본문에서는 볼류메트릭 영상에서 2D 영상과 주로 다른 점이 생기는 뷰포트 적응, 압축 코덱, 전송 프로토콜이라는 세 가지 특성으로 나누어 적응형 스트리밍과 관련된 연구를 설명한다.

1. 뷰포트 적응

뷰포트 적응 (Viewport Adaptation)은 그림 2 와 같이 볼류메트릭 영상에서 사용자 시점에 따라 콘텐츠를 최적화하는 기술이다. 360 도 영상 (3 DoF)에서와 다르게 사용자의 움직임도 처리해야 한다는 차이점이 있으며 (6 DoF), 360 도 영상처럼 2D 영상을 기반으로 하지 않기 때문에 3D 콘텐츠를 동적으로 처리할 필요가 생긴다.

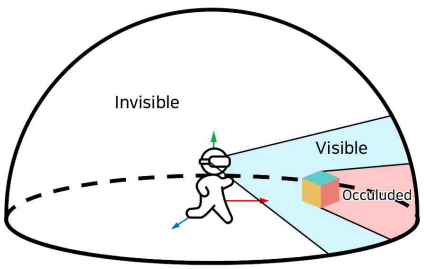

그림 2 볼류메트릭 영상에서의 뷰포트 적응

표 1 은 뷰포트 적응을 다룬 혹은 관련 연구들을 분류한 기준이다. 명시적 표현 방식은 3D 공간에서 데이터의 위치를 직접적으로 표현하며, 포인트 클라우드, 메쉬와 같은 전통적인 3D 데이터 표현 방식 등이 있다. 기존 그래픽스 기술들을 이용하면 시점에 따른 처리가 용이하다는 장점이 있다. 암시적 표현 방식은 함수, 신경망을 통해 데이터를 간접적으로 표현하며, NeRF[5]가 대표적이고, 실제와 유사한 높은 품질을 보여준다는 장점이 있다.

뷰포트 적응 연구 분류	
	명시적 (Explicit) 3D 표현 방식 [3, 7, 8]
	암시적 (Implicit) 3D 표현 방식 [9-11]

표 1 뷰포트 적응 연구 분류

명시적 표현 방식을 타겟으로 하는 볼류메트릭 영상 스트리밍에서 뷰포트 적응을 다룬 대표적인 연구는 다음과 같다. ViVo [7]에서는 포인트 클라우드 콘텐츠를 대상으로 총 3 가지 뷰포트 적응 방법을 사용했으며, 사용자의 위치와 시점에 따른 최적화, 시점 내에 있지만 다른 물체에 가려진 부분에 대한 최적화, 거리에 따른 최적화를 적용하였고, 각각에 대한 방식을 제안하였다. 또한 선형 회귀를 이용한 시점 예측으로 콘텐츠를 미리 로드할 수 있도록 하였다. GROOT [3]에서 역시 포인트 클라우드를 대상으로 ViVo 와 유사한 뷰포트 적응 방식을 소개하였으나, 3D 데이터 저장을 위한 구조를 함께 제안하며, 이를 이용해 효율적이고 연속적으로 하기 위한 방법을 제안하였다. 이외에도 콘텐츠 선호도 및 클라이언트에서의 캐시를 이용해 반복적인 전송을 감소하는 적응 방식 [8] 등이 제안되었다.

암시적 표현 방식은 명시적 표현 방식과 달리 3D 데이터의 위치를 직접적으로 알기 어렵기 때문에 시점에 따른 처리를 어렵게 한다. 이를 해결하기 위해서는 KiloNeRF [9]와 같이 3D 콘텐츠 전체를 하나의 신경망으로 표현하는 대신, 여러 개의 신경망으로 표현하는 방식이 활용될 수 있으며, NeRVo[10] 시스템에서는 메쉬 구조와 신경망 기반 색 표현을 동시에 활용하여 뷰포트 적응을 위한 타일화를 수행하였고, FSVFG [11]에서는 Feature grid 를 사용하여 공간 분리를 수행해 뷰포트 적응에 활용하였다.

뷰포트 적응 방식은 서비스 제공자 측면의 개발이지만, 다양한 플랫폼에서의 적응형 스트리밍을 위해서는 공통적인 저장 구조가 필요할 수 있다. 또한 이는 전송 프로토콜과도 관련이 있어 다른 부분과도 연계하여 연구 및 논의가 필요하다.

2. 압축 코덱

코덱은 미가공 상태의 데이터를 인코딩 및 디코딩을 할 수 있는 하드웨어/소프트웨어를 의미한다. 볼류메트릭 영상의 3D 데이터를 인코딩/압축하면 파일의 크기 및 사용 대역폭을 크게 줄일 수 있어 볼류메트릭 영상 스트리밍에서 필수적인 요소라고 볼 수 있다.

현재 대표적으로 사용되는 압축 코덱 기술로는 구글의 Draco [12]와 MPEG 에서 표준화가 진행중인 MPEG G-PCC [13], V-PCC [14] 등이 있다. MPEG V-PCC 는 3D 물체를 2D 로 투영하기 때문에 2D 영상에서 사용하는 코덱을 그대로 사용할 수 있다는 장점이 있지만, 계산 복잡도가 높아 인코딩 지연시간이 높고, 고정밀 콘텐츠는 표현하기 어려울 수 있다는 단점이 있다 [3]. Draco, MPEG G-PCC 는 3D 공간상에서의 기하, 속성 정보를 옥트리 (octree) 등의 구조를 이용하여 압축을 하는 형태로, 3D 공간의 정보를 그대로 이용하기 때문에 고정밀 데이터를 압축하는데 유리하며, 특히 Draco 는 빠른 인코딩/디코딩 속도를 보여준다.

적응형 스트리밍을 위한 다중 화질 인코딩 연구 분류	
	인코더 인자 조절을 통한 압축률 조절
	포인트 클라우드, 메쉬 [12], 3DGS [15], NeRF [23]
	다중 디테일 수준 (Level of Detail)
	포인트 클라우드 [16-19], Feature Grid [11], Neural SDF [21], 3DGS [22]

표 2 다중 화질 인코딩 연구 분류

품질을 유지하며 높은 압축률을 얻기 위해 많은 연구들이 제안되었지만, 적응형 스트리밍을 위해서는 여러 비트레이트로 인코딩을 할 수 있는 능력이 필요하다. 표 2 는 다중 화질 인코딩을 다루거나 관련된 연구들을 분류한 기준이다.

먼저, 현재 2D 영상의 코덱과 유사하게 인코딩 인자를 조절하여 다중 화질로 만드는 방법이 가능하다 (그림 3). Draco 는 포인트 클라우드 및 메쉬에서 저장하는 기하 정보 등의 양자화 및 압축 정도를 조절하여 여러 비트레이트로의 인코딩을 지원한다. Sun et al. [15]의 논문에서는 3DGS (3D Gaussian Splatting)이 학습 기반의 3D 표현 방식이지만, 포인트 클라우드와 유사하게 기하 및 속성 정보를 가진다는 것에 착안하여, Draco 와 같이 인자를 조절해 압축률을 조절할 수 있는 경우, 대역폭 등 네트워크 상황을 고려해 최적의 인코딩 인자를 찾는 연구를 수행하였다. 신경망 기반 표현 방식의 경우 양자화와 같은 모델 압축 방식들을 사용할 수 있으며, 이 또한 양자화 정도를 조정해 여러 비트레이트로 만드는 연구가 제안되었다 [23].

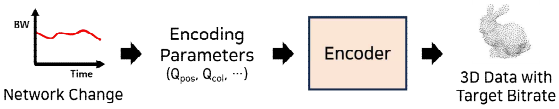

그림 3 인코더 인자 조절을 통한 압축률 조절

위와 같이 단일 3D 콘텐츠에 대해 인코딩 과정에서 압축률을 조정해 여러 비트레이트로 인코딩하는 방식이 있는 반면, 여러 품질의 3D 콘텐츠를 미리 만들거나, SVC (Scalable Video Coding) [4]와 유사하게 여러 품질의 콘텐츠로 쉽게 변환이 가능하도록 만드는 방식도 존재한다. 3D 콘텐츠는 Level-of-Detail (LOD)으로 품질 및 디테일의 차이를 표현하며, 여러 LOD 를 가지는 구조로 설계하여 효율적인 품질 변환을 지원한다.

포인트 클라우드를 이용하는 볼류메트릭 영상에서는 대부분 포인트 샘플링을 통해 여러 품질을 만들어내며, 단순히 여러 품질의 버전으로 만들거나 [16, 17], 혹은 여러 품질을 조합할 수 있는 MDC (Multi Description Coding) [29] 형태로 만드는 등 [18, 19] 구체적인 방식은 연구마다 상이하다. 3D 메쉬에서 역시 다운샘플링 혹은 메쉬 단순화 등의 방법으로 여러 품질로 변환하며, Petrangeli et al. [20]의 논문에서는 여러 버전의 메쉬 LOD 를 가지고 있을 때 최적의 버전을 선택하여 스트리밍을 하는 방식을 제안하였다.

위와 같은 기존 표현방식들은 여러 품질 지원 및 스트리밍에서의 활용에 대해 이미 다양한 연구가 이루어진 반면, 최근 제안되는 학습 기반 표현 방식에서의 다중 품질 표현은 단순 샘플링으로 해결할 수 없어, 이를 해결하기 위한 연구가 최근에 많이 이루어지고 있다. FSVFG [11]에서는 InstantNGP [20]의 다중 화질 인코딩을 이용해 Feature grid 를 여러 품질로 나누어 다중 LOD 를 지원하였고, Takikawa et al. [21]의 논문에서는 Neural SDF 표현 방식을 실시간으로 여러 LOD 로 만드는 방법을 제안하였다. LapisGS [22]는 3DGS 표현 방식에서 데이터의 구조를 SVC 와 같이 여러 레이어로 이루어지도록 하여 적응형 스트리밍에서 레이어 조절을 통한 LOD 변환을 할 수 있도록 하는 학습 과정을 제안하였다.

이처럼 볼류메트릭 영상의 기존 및 새로운 3D 표현 방식들에 대해 다중 품질을 지원하기 위한 다양한 연구가 존재한다. 기존 표현 방식의 코덱은 표준화가 진행 중으로, 적응형 스트리밍에서의 활용이 가속화될 것으로 보이지만, 새로운 표현 방식들은 아직 명확한 표준이 정해지지 않아 다양한 플랫폼에서의 적응형 스트리밍을 위한 실용적인 활용은 어려운 상황이며, 앞으로의 지속적인 연구 및 논의가 필요하다.

3. 전송 프로토콜

현재 적응형 스트리밍을 위해서는 MPEG-DASH 나 HLS 와 같은 프로토콜을 활용한다. 이들은 HTTP 를 기반으로 하여 대부분의 디바이스 및 플랫폼에서 지원을 하고 있어 높은 호환성을 가진다는 장점이 있다. 하지만 현재 해당 프로토콜들은 2D 영상에 대한 인코딩 형식들만 지원하기 때문에, 볼류메트릭 영상에 바로 사용하기는 어렵다. 따라서 볼류메트릭 영상의 적응형 스트리밍을 위해서는 기존 프로토콜의 확장 또는 새로운 프로토콜의 제안이 필요하다. 표 3 은 전송 프로토콜을 다루거나 관련된 연구들을 분류한 기준이다.

전송 프로토콜 연구 분류	
	MPEG-DASH 확장 프로토콜 [24-28]
	기타 프로토콜
	TCP 기반 [7], UDP 기반 [18, 19]

표 3 전송 프로토콜 연구 분류

기존 볼류메트릭 영상 스트리밍 연구에서는 많은 경우에 MPEG-DASH 를 확장하여 이용하였다. MPEG-DASH 는 그림 4 와 같이 서버가 조각화된 여러 품질의 콘텐츠를 가지고 있으며, 클라이언트가 어떤 콘텐츠에 대한 메타데이터 정보를 담고 있는 MPD 파일을 이용해 원하는 품질을 요청하고 받는 구조이다. Hosseini et al. [24]와 Hooft et al. [25]의 논문은 포인트 클라우드에 대한 DASH 확장을 제안하였고, Farrugia et al. [26]의 논문에서는 3D 메쉬에 대한 DASH 기반 스트리밍 시스템을 제안하였다. YuZu [27]와 VoLUT [28]는 3D 초해상화 기반의 대역폭 절약 시스템을 제안하며, 저품질 및 중간 품질 콘텐츠 전송을 위해 DASH 를 확장해 사용하였다. 이와 같이 MPEG-DASH 는 새로운 코덱을 사용하더라도 MPD 파일과 여러 품질의 세그먼트 등의 요구사항을 만족하고, 원하는 코덱에 대한 인코딩 및 디코딩 과정을 추가한다면 다양하게 확장이 가능하다.

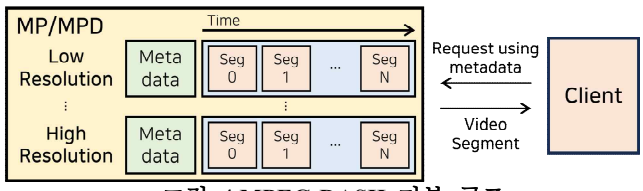

그림 4 MPEG-DASH 기본 구조

일부 연구에서는 MPEG-DASH 외의 다른 프로토콜을 사용하였다. ViVo [7]는 TCP 기반의 자체 프로토콜의 사용을 언급했지만, 구체적인 내용은 설명하지 않았다. Matthias et al. [18, 19]은 WebRTC 를 통해 UDP 기반 프로토콜인 RTP 를 확장해 사용하였다. MPEG-DASH 의 기존 플랫폼에서 호환성과 확장 가능성으로 볼류메트릭 영상을 위한 새로운 적응형 스트리밍 프로토콜은 거의 제안이 되지 않고 있다. 하지만 2D 영상과 달리 뷰포트 적응 등의 차이점이 있기 때문에, 저지연, 고품질 미디어 제공과 효율적 자원 사용이 가능한 최적화된 기술을 위해서는 지속적인 연구가 필요하다.

Ⅲ. 결론

본 논문에서는 볼류메트릭 영상에 적응형 스트리밍을 적용하기 위한 관련 연구들을 소개하였다. 특히 2D 영상의 경우와 대표적인 차이점인 뷰포트 적응, 압축 코덱, 전송 프로토콜을 중심으로 각각에 대한 대표적인 연구들을 설명하였다. 이를 통해 적응형 스트리밍 적용을 위해 볼류메트릭 영상과 2D 영상 스트리밍이 서로 다른 요구사항을 가진다는 점을 강조하고, 이를 성공적으로 적용하기 위해 서비스 혹은 네트워크 관리 측면에서 핵심 문제 및 방향성에 대한 통찰과 연구의 필요성을 제시하였다. 향후 연구로는 본 연구를 바탕으로 여러 형태의 3D 콘텐츠에 대해 통합적으로 적용될 수 있는 실용적인 적응형 스트리밍 기술 연구를 목표로 한다.

ACKNOWLEDGMENT

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-대학 ICT 연구센터(ITRC)의 지원을 받아 수행된 연구임(IITP-2025-RS-2024-00437866)

참 고 문 헌

[1] Pantos, R., and W. May. HTTP Live Streaming. RFC 8216, Internet Engineering Task Force, August 2017, doi:10.17487/RFC8216.

[2] MPEG-DASH, Retrieved Mar., 28, 2025, https://www.iso.org/standard/83314.html

[3] Lee, Kyungjin, et al. "GROOT: A real-time streaming system of high-fidelity volumetric videos." Proceedings of the 26th Annual International Conference on Mobile Computing and Networking. 2020.

[4] Schwarz, Heiko, Detlev Marpe, and Thomas Wiegand. "Overview of the scalable video coding extension of the H. 264/AVC standard." IEEE Transactions on circuits and systems for video technology 17.9 (2007): 1103-1120.

[5] Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." Communications of the ACM 65.1 (2021): 99-106.

[6] Kerbl, Bernhard, et al. "3d gaussian splatting for real-time radiance field rendering." ACM Trans. Graph. 42.4 (2023): 139-1.

[7] Han, Bo, et al. "ViVo: Visibility-aware mobile volumetric video streaming." Proceedings of the 26th annual international conference on mobile computing and networking. 2020.

[8] Liu, Junhua, et al. "Cav3: Cache-assisted viewport adaptive volumetric video streaming." 2023 IEEE Conference Virtual Reality and 3D User Interfaces. 2023.

[9] Reiser, Christian, et al. "Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps." Proceedings of the IEEE/CVF international conference on computer vision. 2021.

[10] Liu, Junhua, et al. "Mobile volumetric video streaming system through implicit neural representation." Proceedings of the 2023 Workshop on Emerging Multimedia Systems. 2023.

[11] Yin, Daheng, et al. "FSVFG: Towards Immersive Full-Scene Volumetric Video Streaming with Adaptive Feature Grid." Proceedings of the 32nd ACM International Conference on Multimedia. 2024.

[12] Draco 3D Data Compression, Retrieved Mar., 28, 2025, https://google.github.io/draco.

[13] MPEG G-PCC, Retrieved Mar., 28, 2025, https://www.iso.org/standard/78990.html.

[14] MPEG V-PCC, Retrieved Mar., 28, 2025, https://www.iso.org/standard/83535.html.

[15] Sun, Yuan-Chun, et al. "Multi-frame bitrate allocation of dynamic 3D Gaussian splatting streaming over dynamic networks." Proceedings of the 2024 SIGCOMM Workshop on Emerging Multimedia Systems. 2024.

[16] Nguyen, Quang Long, et al. "Toward optimal real-time dynamic point cloud streaming over bandwidth-constrained networks." Proceedings of the 5th ACM International Conference on Multimedia in Asia. 2023.

[17] Hu, Kaiyuan, et al. "LiveVV: Human-Centered Live Volumetric Video Streaming System." IEEE Internet of Things Journal (2025).

[18] De Fré, Matthias, et al. "Scalable mdc-based volumetric video delivery for real-time one-to-many webrtc conferencing." Proceedings of the 15th ACM multimedia systems conference. 2024.

[19] De Fré, Matthias, et al. "Demonstrating Adaptive Many-to-Many Immersive Teleconferencing for Volumetric Video." Proceedings of the 15th ACM Multimedia Systems Conference. 2024.

[20] Petrangeli, Stefano, et al. "Dynamic adaptive streaming for augmented reality applications." 2019 IEEE International Symposium on Multimedia (ISM). IEEE, 2019.

[21] Takikawa, Towaki, et al. "Neural geometric level of detail: Real-time rendering with implicit 3d shapes." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[22] Shi, Yuang, et al. "LapisGS: Layered Progressive 3D Gaussian Splatting for Adaptive Streaming." arXiv preprint arXiv:2408.14823 (2024).

[23] Takikawa, Towaki, et al. "Variable bitrate neural fields." ACM SIGGRAPH 2022 Conference Proceedings. 2022.

[24] Van Der Hooft, Jeroen, et al. "Towards 6dof http adaptive streaming through point cloud compression." Proceedings of the 27th ACM International Conference on Multimedia. 2019.

[25] Hosseini, Mohammad, and Christian Timmerer. "Dynamic adaptive point cloud streaming." Proceedings of the 23rd Packet Video Workshop. 2018.

[26] Farrugia, Jean-Philippe, et al. "Adaptive streaming of 3D content for web-based virtual reality: an open-source prototype including several metrics and strategies." Proceedings of the 14th Conference on ACM Multimedia Systems. 2023.

[27] Zhang, Anlan, et al. "YuZu:Neural-Enhanced volumetric video streaming." 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22). 2022.

[28] Wang, Chendong, et al. "VoLUT: Efficient Volumetric streaming enhanced by LUT-based super-resolution." arXiv preprint arXiv:2502.12151 (2025).

[29] Goyal, Vivek K. "Multiple description coding: Compression meets the network." *IEEE Signal processing magazine* 18.5 (2002): 74-93.

중앙화 암호화폐 거래소의 준비금 증명: 현황, 한계점, 규제

강창훈, 홍원기
포항공과대학교 컴퓨터공학과

{chkang, jwkhong}@postech.ac.kr

Proof-of-Reserves in Centralized Crypto Exchanges: Status, Challenges, and Regulations

Changhoon Kang, James Won-Ki Hong
Department of Computer Science and Engineering, POSTECH

요 약

본 논문은 중앙화 암호화폐 거래소 (Centralized Exchange, CEX)의 사용자 자산 신뢰성 확보를 위한 기술적 접근 방식인 준비금 증명 (Proof-of-Reserves, PoR)에 대해 다룬다. 2022년 FTX 거래소의 파산은 사용자 자산의 실제 보관 여부를 외부에서 확인할 수 없는 중앙화 거래소 구조의 근본적인 취약성을 드러냈으며, 이를 계기로 거래소의 지급 능력과 자산 투명성을 입증할 수 있는 기술적 시스템에 대한 관심이 급증하였다. 본 논문에서는 준비금 증명의 개념과 주요 국내외 거래소의 도입 현황 및 기술적 방식의 차이를 비교한다. 준비금 증명의 구성요소에 대해 설명하고, 거래소별로 현재 어떤 방법을 통해 준비금 증명을 이행하고 있는지 자세히 알아본다. 또한, 현재 시스템이 갖는 기술적 한계에 대해 논의하고, 미국, EU, 대한민국의 규제 환경에 대해 살펴본다. 본 연구는 향후 준비금 증명 시스템의 실효성과 지속 가능성을 높이기 위한 기술적, 제도적 개선 방향을 제안하며, 준비금 증명 관련 기술개발 및 제도 마련을 위한 참고 자료로 활용될 수 있을 것이다.

Ⅰ. 서론

암호화폐 (Cryptocurrency)는 탈중앙화된 디지털 자산으로서 블록체인 기술을 기반으로 발행 및 운용되며, 가치 저장, 송금 등 다양한 기능을 수행한다. 2009년 비트코인 [1]을 시작으로 수많은 암호화폐들이 다양한 기반 기술과 목적을 갖고 발행되어 왔다. 또한, 이들 암호화폐를 기반으로 선물, 옵션, ETF 등 여러 파생상품들까지 만들어지고 있으며, 전세계에서 수많은 자금이 유입되며 시가총액이 꾸준히 증가해왔다. 이러한 암호화폐의 거래는 일반적으로 암호화폐 거래소를 통해 이루어지며, 이들 거래소는 사용자에게 자산 보관, 매매, 출금 등의 서비스를 제공한다.

암호화폐 거래소는 크게 중앙화 거래소 (Centralized Exchange, CEX)와 탈중앙화 거래소 (Decentralized Exchange, DEX)로 나뉜다 [2]. 중앙화 거래소는 기업 또는 단체가 운영하는 구조로, 사용자 자산을 거래소 내부 암호화폐 지갑에 예치하고 거래를 중개하는 방식이다. 대부분 거래소가 자체 관리하는 지갑에서 모든 사용자들의 자산이 관리되며, 각 사용자별 잔고나 거래소 내 매매 내역은 거래소 서버에서 전산상으로만 기록되는 구조로 운영된다. 이에 반해 탈중앙화 거래소는 스마트 컨트랙트를 기반으로 운영되며, 사용자 간 직접 거래를 지향한다. 대부분 Automated Market Maker (AMM) [3] 라는 개념을 적용해 중개자 없이 유동성 풀을 기반으로 사용자 간 직접적인 거래를 지원한다.

중앙화 거래소는 높은 유동성과 사용자 친화적인 인터페이스, 빠르고 다양한 거래 기능으로 인해 전 세계 암호화폐 거래의 대부분을 차지하고 있다. 탈중앙화 거래소들에서의 거래량이 점차 증가하는 추세이나, 여전히 많은 사람들이 중앙화 거래소를 이용하고 있다. 그러나 이의 중앙화된 구조 특성상, 사용자 자산이 실제로 어떻게 보관되고 있는지 외부에서 검증하기 어렵고, 거래소의 내부 통제나 재무 구조에 대한 투명성이 낮다는 구조적 한계를 갖는다. 즉, 사용자는 이를 운영하는 기업이나 단체에 대한 신뢰를 기반으로 자신의 디지털 자산을 맡기고 관리할 수밖에 없다. 이러한 문제는 2022년 11월 발생한 FTX 거래소의 파산 사태 [4]로 더욱 극명하게 드러났다. FTX는 고객 자산을 계열사에 무단으로 이전하고, 회계 정보를 조작하며, 준비금 없이 거래를 지속해온 것으로 밝혀졌다. 회사 자산 대부분을 자사 발행 토큰인 FTT 및 특정 암호화폐로만 보유하고 있어 자금 유동성이 거의 없는 상태였고, 관련 기사로 인해 많은 사용자들이 자신의 자산을 출금하였고 FTT의 가치 역시 폭락하여 결국 파산으로 이어지게 되었다. 이로 인한 자산 인출 중단으로 수십억 달러 규모의 사용자 자산이 손실되었고, 암호화폐 생태계 전반에 걸쳐 신뢰 위기가 발생했다.

이 사건 이후, 중앙화 거래소에 대한 신뢰를 회복하고 사용자 자산의 실재성과 보관 상태를 검증하기 위한 기술적 시도로 준비금 증명 (Proof-of-Reserves, PoR) [5] 시스템이 주목받기 시작했다. 준비금 증명은 다양한 방법을 활용하여 거래소가 보유한 자산이 실제 사용자 예치금과 일치하거나 더 많음을 외부에서 검증할 수 있도록 하는 체계다. 일부 주요 거래소들은 머클 트리 (Merkle Tree) [6], 영지식 증명 (Zero-Knowledge Proof) [7] 등 여러 암호 기술 기반의 준비금 증명 시스템을 개발 및 도입하고 있으며, 이 중 일부는 오픈소스 도구를 제공하여 사용자 스스로 자신의 자산이 준비금 내에 포함되어 있는지를 확인할 수 있도록 하고 있다.

본 논문은 이러한 배경 하에, 중앙화 거래소의 준비

금 증명에 관한 현황과 한계점, 그리고 세계 각국의 관련 규제에 대해 살펴보고자 한다. 구체적으로는 준비금 증명의 개념과 구현 방식에 대해 정리하고, 해외 주요 거래소들의 기술 도입 사례를 분석하여 현재 사용되는 방법들의 한계점을 검토한다. 또한, 국내외 관련 규제 환경을 비교 및 정리하여, 향후 중앙화 거래소에서 준비금 증명이 보다 효과적으로 안전하게 작동할 수 있도록 하는 방안을 논의한다.

Ⅱ. 배경

1. 중앙화 암호화폐 거래소

중앙화 거래소는 기업이나 단체 등 특정 운영 주체가 중앙 서버를 통해 거래를 중개하고 자산을 관리하는 형태의 암호화폐 거래소이다. 사용자들은 암호화폐나 원화, 달러와 같은 법정화폐 (Fiat Money)를 해당 거래소에 예치한 뒤, 거래소가 제공하는 웹 또는 모바일 인터페이스를 통해 암호화폐 매매, 보관, 입출금 등의 서비스를 이용한다. 이 과정에서 사용자 자산은 블록체인 상에 개별 주소로 분산 저장되는 것이 아니라, 거래소가 보유한 하나 또는 여러 개의 공용 지갑 주소 (Wallet Address)에 함께 보관된다 [8]. 예를 들어, 수천 명의 사용자가 각각 1 비트코인을 예치하더라도, 이 비트코인은 사용자별 고유 지갑이 아닌 거래소가 운영하는 통합 지갑에 함께 저장되며, 사용자별 잔고는 거래소 내부 데이터베이스 상에서만 구분된다.

이처럼 사용자 자산은 거래소의 온체인 지갑에 실제 존재하지만, 각 사용자의 보유 내역은 블록체인 상에서 확인할 수 없고, 전적으로 거래소 내부 전산 시스템 (오프체인 장부)에 의존하여 기록된다. 사용자는 로그인 후 자신의 계정에서 보유 자산 현황을 확인할 수 있으나, 이는 블록체인에 기록된 정보가 아니라 중앙 서버의 데이터베이스에서 조회되는 값이다. 이는 거래소가 높은 거래 처리 속도와 유동성 확보를 위해 사용하고 있는 구조이며, 다양한 보안 정책과 백업 체계를 갖추어 자산을 관리한다. 그러나 이런 방식은 블록체인의 핵심 속성인 투명성과 불변성으로부터 벗어나 있으며, 사용자는 거래소가 실제로 자신이 보유한 자산을 안전하게 관리하고 있다는 점을 신뢰에 기반하여 받아들여야만 한다.

이러한 구조적 특성은 평상시에는 문제없이 작동하지만, 거래소의 내부 운영에 문제가 발생하거나 자산 유동성이 부족해질 경우, 사용자는 본인의 자산이 실제로 존재하는지조차 확인할 수 없다는 위험성을 내포하고 있다. FTX 파산 사태는 이러한 위험이 실제로 현실화된 대표적인 사례로, 거래소의 자산 보유 실태와 부채 구조가 외부에서 확인되지 않는 상황에서 신뢰 기반 모델이 어떻게 붕괴할 수 있는지를 보여주었다. 이 사건 이후 거래소의 자산 운용 실태에 대한 기술적 투명성 확보 수단의 필요성이 본격적으로 제기되었으며, 그 해결책으로 준비금 증명 시스템이 부상하게 되었다.

2. 준비금 증명 (Proof-of-Reserves)

준비금 증명은 중앙화 거래소가 사용자들이 예치해 둔 자산에 상응하는 암호화폐를 실제로 보유하고 있다는 사실을 다양한 방법을 통해 외부에서 검증 가능하게 증명하는 체계이다. 이는 외부 회계감사 결과일수도 있고, 누구나 접근 가능한 공개 데이터와 수학적 구조를 활용하여 사용자와 제 3 자가 직접 확인할 수 있는 투명한 증명 방식이 되기도 한다. 준비금 증명은 아래와 같이 크게 두 가지 핵심 요소로 구성된다.

i. 자산 증명 (Proof-of-Assets)

자산 증명은 거래소가 실제 보유 중인 암호화폐 자산의 총량을 블록체인 상에서 입증하는 절차이다. 사용자들이 예치한 암호화폐 자산을 포함해 자산이 실제 온체인 지갑 상에서 갖고 있는 암호화폐의 종류와 각 잔고를 증명한다. 이를 위해 일부 거래소는 가장 간단한 방법으로 자체 관리하는 지갑 주소를 모두 공개하고, 해당 주소의 잔액을 누구나 확인할 수 있도록 한다. 이러한 공개는 보통 블록체인 탐색기 (Explorer) [9]를 통해 실시간으로 검증 가능하며, 블록체인의 불변성과 공개성을 기반으로 데이터의 위변조 가능성이 낮다.

그러나 단순히 지갑 주소를 공개하는 것만으로는 해당 지갑이 진정으로 거래소의 자산인지 여부를 확인하기 어렵기 때문에, 거래소는 해당 지갑의 개인 키 [10]를 이용하여 특정 메시지를 암호학적으로 서명하거나, 서명된 트랜잭션을 제출함으로써 해당 지갑에 대한 소유권을 입증한다. 이 과정을 통해 거래소는 단순히 자산의 존재뿐만 아니라 자산에 대한 소유까지 증명할 수 있다.

ii. 부채 증명 (Proof-of-Liabilities)

부채 증명은 거래소 서버 내 데이터베이스에서 기록된 사용자들의 암호화폐 잔고 총량을 증명하는 절차이다. 이를 위해 거래소는 모든 사용자 계정의 잔고 정보를 집계하고, 그 총합을 외부에서 검증 가능하도록 만든다.

단, 사용자 잔고 데이터는 프라이버시와 보안상의 이유로 공개할 수 없기 때문에, 거래소는 전체 사용자 데이터를 머클 트리 구조로 요약하거나 암호화된 형태로 구조화하여 외부에 제공한다. 각 사용자는 거래소가 제공하는 검증 도구를 이용해 자신의 계정 정보가 해당 부채 증명에 포함되어 있는지 확인할 수 있으며, 이 과정을 통해 거래소가 실제로 사용자 자산을 부채 총합에 반영하고 있는지를 검증할 수 있다.

최종적으로 위 자산 증명과 부채 증명이 함께 수행되면, 외부에서는 거래소가 보유한 암호화폐 자산이 사용자들의 잔고 총합 이상이며, 개별 사용자도 자신의 자산이 시스템 상에 존재함을 확인할 수 있게 된다. 이처럼 준비금 증명 시스템은 거래소의 지급 능력을 입증하는 동시에 사용자 신뢰를 제고하고, 생태계 전반의 건전성과 투명성을 향상시키는 역할을 수행한다.

Ⅲ. 주요 거래소 준비금 증명 도입 현황

2022 년 FTX 거래소 파산 이후, 다수의 중앙화 거래소는 사용자 자산의 안전성과 투명성을 확보하기 위해 준비금 증명을 도입하거나 검토하고 있다. 주요 글로벌 거래소들은 각각의 기술적 방식과 운영 모델을 바탕으로 준비금 증명 시스템을 구축하였으며, 일부는 이를 정기적으로 외부에 보고하거나 자체 검증 도구를 제공하고 있다. 이 절에서는 주요 거래소들의 준비금 증명 도입 현황을 비교하고, 각 거래소가 활용하고 있는 기술 메커니즘에 대해서도 간략히 살펴본다.

표 1 은 주요 국내외 거래소들의 준비금 증명 실시 여부 및 기술 도입 현황을 정리한 것이다. 자산 증명과 부채 증명의 실시 여부를 분리하여 표시하고, 각 거래소가 활용 중인 기술적 메커니즘과 특징을 함께 제시하였다. 현재까지 준비금 증명을 위해 가장 많은 기술적 시도를 하고 있는 곳은 해외 주요 거래소인 Binance [11]와 OKX [12]이다. 두 거래소는 자산 증명을 위해 거래소 소유의 암호화폐 지갑 주소를 모두 공개하고 있다. 하지만 지갑 주소를 공개한다는 것이 꼭 해당 지갑을 소유하고 있음까지 증명하는 것은 아니기 때문에, OKX 의 경우 "I am an OKX address" 라는 메시지를 개인 키로 서명하여 함께 공개하고 있다. 부채 증명을 위해서는 둘 다 머클 트리 구조 및 영지식 증명을 활용하는 방법을 사용하고 있다. 거래소는 사용자별 잔고 정보를 머클 트리의 리프 (Leaf)로 두고 머클 루트 (Root)를 생성하여 이를 제공한다. 그러면 거래소는 자신이 제공하거나 오픈 소스로 공개된 검증기를 통해 각 사용자의 계정 잔고가 머클

거래소	기술적 준비금 증명 방법		주요 특징
	자산 증명	부채 증명	
Binance	거래소 지갑 주소 단순 공개	머클 트리 및 영지식 증명 기반 검증 지원	zk-SNARK 적용하여 머클 트리 유효성 증명, 자체 검증 도구 제공 (매월 1 회)
OKX	지갑 주소 및 서명 공개	머클 트리 및 영지식 증명 기반 검증 지원	zk-STARK 적용하여 머클 트리 유효성 증명, CLI 기반 자체 검증 도구 제공 (매월 1 회)
Coinbase	미실시	미실시	자체 발행한 Wrapped BTC 인 cbBTC 에 대해서만 지갑 주소 공개
Kraken	미실시	머클 트리 기반 검증 지원	외부 회계법인이 머클 트리 및 실사보고서 제작하여 공개 (6 개월~1 년 주기)
업비트	미실시	미실시	분기별 실사보고서 공개 (총량 제외)
빗썸	미실시	미실시	분기별 실사보고서 공개 (총량 제외)
코인원	미실시	미실시	분기별 실사보고서 공개 (총량 제외)
코빗	거래소 지갑 주소 단순 공개	자산별 부채 총량 단순 공개	분기별 실사보고서와 병행

표 1. 주요 거래소 준비금 증명 도입 현황

트리에 포함되었는지, 부채를 실제보다 적게 발표하기 위해 마이너스 잔고를 가진 가짜 계정을 포함시키지는 않았는지 등을 사용자 계정 정보와 잔고를 공개하지 않으면서 영지식 증명을 통해 외부에 증명하는 것이 가능하다.

이외 대부분의 주요 거래소들은 외부 회계법인을 통해 정기적인 감사를 받고 보고서를 공개하는 것을 통해 준비금 증명을 진행하고 있다. 국내 거래소인 업비트 [13], 빗썸 [14], 코인원 [15] 역시 외부 감사를 통해 실사보고서를 분기별로 공개하고 있다. 다만 이들은 부채 대비 자산 보유 비율만을 공개할뿐, 구체적인 규모에 대해서는 공개하지 않고 있다. 코빗 [16]의 경우에는 분기별 보고서와 함께 실시간으로 보유 비율 및 구체적인 수량과 지갑 주소 목록까지 공개하고 있다. 하지만 위 해외 사례처럼 공개된 정보가 사실인지 외부에서 검증할 수 있는 방법이 없다는 일부 한계가 있다.

Kraken [17]의 경우, 위 해외 거래소 사례처럼 머클 트리 구조로 부채 증명을 만들고 있으나 이는 외부 회계법인을 통해 이루어진다. 자산 증명 과정 역시 외부 회계법인에 의해 거래소 보유 지갑 및 잔고 확인이 진행된다. 머클 루트가 공개되어 사용자들이 직접 자신의 자산 포함 여부를 검증할 수 있고, 구체적인 거래소의 자산 및 부채 규모도 공개된다는 점에서 국내 거래소들의 준비금 증명과 차이점이 있다.

Coinbase [18]는 현재 거래소 자체적으로 발행한 Wrapped BTC 인 cbBTC [19]에 대해서만 준비금 증명을 실시하고 있다. 거래소에 비트코인을 비축해두고 이와 동일한 가치를 가지는 cbBTC 를 발행한 것이다. 이 경우 자산과 부채 모두 온체인 상에서 확인 가능하기 때문에 양쪽에 해당하는 지갑 주소를 공개하는 것으로 준비금을 증명하고 있다.

Ⅳ. 현 준비금 증명의 기술적 한계점

현재 거래소들의 준비금 증명 시스템은 자산 투명성과 사용자 신뢰 회복을 위한 유효한 방안으로 주목받고 있으나, 아직 기술적으로 여러 가지 한계점이 존재한다. 우선 기본적으로 외부 회계법인을 통해 준비금 증명을 하는 경우, 사용자들이 해당 회계법인을 믿어야 한다는 신뢰 비용이 발생한다. 앞선 절에서 설명한 머클 트리 구조와 영지식 증명을 활용한 방법은 이러한 신뢰 비용 문제를 해결했지만 여전히 아래와 같은 한계를 갖고 있다.

먼저, 자산 및 부채 증명 과정이 비효율적이며 사용자 친화적이지 못하다. 자산 증명을 위해 거래소가 보유한 수많은 지갑 주소를 단순히 공개한다면 투명성을 제공할 수는 있겠지만, 사용자가 이것이 정확한 정보인지 온체인 데이터와 비교 검증하고 거래소의 소유 여부까지 확인하기는 많은 시간과 노력이 필요하다. 관련 작업을 돕기 위해 일부 거래소는 관련 툴을 제공하고 있으나 일반 사용자가 활용하기에는 어려움이 있다. 또한, 부채 증명 역시 관련 검증 도구 사용에 대한 기술적 장벽이 존재하며, 관련 증명 과정 자체에 대해 사용자들의 기술적 이해를 높이기 위한 시도도 부족하다.

다음으로, 준비금 증명의 지속성 부족 및 긴 검증 주기이다. 거래소가 한번 증명을 생성하여 공개되고 나면 이는 생성 순간의 스냅샷 정보를 바탕으로 만들어진 것이므로 그 시점에서만 유효한 증명이 된다. 따라서 증명 생성 이후 거래소가 자금을 임의로 이동시키더라도 다음 증명이 발행되기 전까지는 외부에서 이를 알기 어렵다. 따라서 실시간 또는 매우 짧은 주기로 반복적인 증명을 제공할수록 조금 더 높은 신뢰도를 얻을 수 있지만, 거래소 내 수많은 사용자 계정들의 잔고를 모두 포함하여 머클 트리를 만들기 위해서는 많은 시간과 비용이 소요된다. 이러한 이유로 현재 해당 방식을 도입한 거래소들은 최소 한달 또는 분기를 주기로만 증명을 공개하고 있다.

Ⅴ. 준비금 증명 관련 국내외 규제 현황

거래소의 준비금 증명 시스템과 관련된 규제 환경은 국가별로 상이하며, 아직까지 명확한 국제적 표준은 부재한 상태이다. 다만, 최근 글로벌 규제 기관들은 가상자산 시장에 대한 감독을 강화하는 방향으로 움직이고 있으며, 준비금 증명 또는 이에 준하는 자산 보유 증명 체계를 제도권에 편입하려는 움직임이 나타나고 있다.

미국의 경우, 연방 차원에서 중앙화 거래소에 대한 포괄적인 가상자산 규제 법안은 아직 제정되지 않았으나, 각 주의 금융 라이선스 제도와 기존 금융법을 통해 간접적인 감독이 이루어지고 있다. 예를 들어, 뉴욕주의

NYDFS (New York State Department of Financial Services)는 BitLicense 제도 [20]를 통해 거래소에 일정 수준의 재무 건전성, 내부 통제 및 사용자 자산 보호 체계를 요구하고 있다. 이에 따라 거래소들은 준비금 증명 시스템을 자율적으로 운영하고 있다. EU 는 2024 년부터 MiCA (Markets in Crypto-Assets) 규제 [21]를 통해 가상자산사업자에 대한 종합적인 규제 체계를 도입하고 있다. MiCA 는 거래소의 자본금 요건, 유동성 요건, 내부 통제 기준, 사용자 자산 보호 등 다양한 항목을 포함하고 있으며, 특히 자산 및 부채 구조의 정기 보고를 의무화하여 간접적으로 준비금 증명 도입을 유도하고 있다.

대한민국은 2024 년부터 시행된 "가상자산 이용자 보호 등에 관한 법률" [22]을 통해 가상자산사업자에 대한 준비금 보유 의무를 명시하였다. 해당 법률에 따르면, 가상자산사업자는 해킹, 전산장애 등 사고에 대비해 이용자 가상자산 보유량의 5% 또는 원화 마켓을 운영하는 경우에는 30 억 원 이상에 해당하는 금액을 보상한도로 하는 보험이나 공제에 가입해야 한다. 또한, 이용자 자산의 80% 이상을 콜드월렛에 보관하고, 이용자 자산과 사업자 고유 자산을 분리 보관해야 하며, 외부감사, 재무제표 공시, 사고 발생 시 보고 의무 등도 함께 규정되어 있다.

요약하면, 아직까지 준비금 증명 시스템에 대한 국제적 표준은 부재하지만, 각국은 자산 보유 및 유동성 확보를 법제화하거나 지침 형태로 도입함으로써, 실질적으로 준비금 증명의 도입을 유도하고 있다. 향후에는 외부 회계감사 기관이 아닌, 기술 기반의 준비금 증명 방식을 제도권에서 공식적으로 인정할지 여부가 주요 쟁점이 될 것으로 보인다.

Ⅵ. 결론 및 향후 연구

본 논문에서는 중앙화 암호화폐 거래소의 준비금 증명에 대해서 개념, 주요 거래소의 도입 현황 및 기술적 구현 방식, 그리고 현재 기술의 한계와 규제 환경을 중심으로 분석하였다. 준비금 증명은 블록체인의 특성을 활용하여 사용자 자산의 실재성과 거래소의 지급 능력을 외부에서 검증할 수 있도록 하는 기술로, 특히 FTX 사태 이후 거래소에 대한 신뢰 회복을 위한 핵심 도구로 부상하였다. 현재의 준비금 증명 시스템은 머클 트리, 영지식 증명 등 다양한 암호학적 기법을 활용하고 있으나 사용자 친화성 부족, 증명 주기의 비효율성, 실시간 검증의 어려움 등의 문제는 여전히 기술적 한계로 남아 있다. 규제 측면에서는 아직 국제적인 표준이 존재하지 않으며, 각국이 상이한 법적 틀 안에서 자산 보호 규정을 마련하고 있는 상황이다. 현재까지 대부분 기술적 방식의 준비금 증명 시스템을 직접적으로 의무화하거나 효력을 공식적으로 인정하고 있지는 않고 있으며, 향후 이러한 기술 기반 증명 수단에 대한 법적 인정 여부가 중요한 논의 주제가 될 것으로 보인다.

향후 연구로는 실시간 또는 고빈도 준비금 시스템 구현을 위한 기술적 구조와 성능 분석이 필요하다. 또한, 사용자 친화적 검증 도구의 개발 및 표준화 방안에 대한 연구가 준비금 증명 확산의 실효성을 높이는 데 기여할 수 있다. 마지막으로, 기존의 방식에서 벗어나 고빈도가 아닌 일정 주기로만 준비금 증명을 수행하더라도 기존의 지속성 문제를 해결할 수 있는 방법이 있을지에 대한 연구도 필요하다.

기술적 준비금 증명 시스템은 아직까지 거래소의 상환 능력 검증을 위한 보조적인 수단으로 도입되어 사용되고 있다. 여러 측면에서 바라봤을 때, 외부 회계감사를 통한 방식과 비교하여 서로 장단점이 분명히 존재하고, 따라서 결국 거래소가 두 가지 방식을 상호보완적으로 함께 사용하는 것이 사용자의 자금을 더욱 안전하게 보호하는 일이라 생각된다.

ACKNOWLEDGMENT

이 논문은 25 년도 정부(경찰청)의 재원으로 과학치안진흥센터 경찰건강 스마트관리 사업의 지원을 받아 수행된 연구임(No. RS-2022-PT000186)

참 고 문 헌

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.
[2] S. Hägele, “Centralized exchanges vs. decentralized exchanges in cryptocurrency markets: A systematic literature review,” Electronic Markets, vol. 34, no. 1, p. 33, 2024.
[3] A. Capponi and R. Jia, “The adoption of blockchain-based decentralized exchanges,” arXiv preprint arXiv:2103.08842, 2021.
[4] E. Akyildirim et al., “Understanding the FTX exchange collapse: A dynamic connectedness approach,” Finance Research Letters, vol. 53, p. 103643, 2023.
[5] A. Dutta and S. Vijayakumaran, “MProve: A proof of reserves protocol for Monero exchanges,” in 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2019.
[6] H. Liu et al., “Merkle tree: A fundamental component of blockchains,” in 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS), IEEE, 2021.
[7] X. Sun et al., “A survey on zero-knowledge proof in blockchain,” IEEE Network, vol. 35, no. 4, pp. 198–205, 2021.
[8] S. Nummelin, “Risks and benefits of centralized and decentralized cryptocurrency exchanges and services,” 2022.
[9] C. Lee et al., “Blockchain explorer based on RPC-based monitoring system,” in 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), IEEE, 2019.
[10] R. Stephen and A. Alex, “A review on blockchain security,” IOP Conference Series: Materials Science and Engineering, vol. 396, no. 1, IOP Publishing, 2018.
[11] Binance. “Proof of Reserves.” Available at https://www.binance.com/en/proof-of-reserves, Accessed: 2025-04-01.
[12] OKX. “Proof of Reserves.” Available at https://www.okx.com/proof-of-reserves, Accessed: 2025-04-01.
[13] Upbit. “업비트, 이용자 자산 보호 및 투명성 강화를 위한 '예치금 및 실시간 보유 자산' 공개.” Available at https://www.upbit.com/service_center/notice?id=4825, Accessed: 2025-04-01.
[14] Bithumb. “회사보고서.” Available at https://www.bithumbcorp.com/ko/company/report.php, Accessed: 2025-04-01.
[15] Coinone. “코인원, 예치금 및 보유코인 현황 공개.” Available at https://coinone.co.kr/info/notice/3734, Accessed: 2025-04-01.
[16] Korbit. “Proof of Reserve – Korbit Insights.” Available at https://insights.korbit.co.kr/reserve, Accessed: 2025-04-01.
[17] Kraken. “Proof of Reserves.” Available at https://www.kraken.com/proof-of-reserves, Accessed: 2025-04-01.
[18] Coinbase. “Proof of Reserves – Coinbase.” Available at https://www.coinbase.com/cbbtc/proof-of-reserves, Accessed: 2025-04-01.
[19] Coinbase. “Coinbase Bitcoin Transparency Report.” Available at https://www.coinbase.com/cbbtc, Accessed: 2025-04-01.
[20] U. W. Chohan, “Oversight and regulation of cryptocurrencies: BitLicense,” Cryptofinance: A new currency for a new economy, pp. 105–120, 2022.
[21] F. Annunziata, “An Overview of the Markets in Crypto-Assets Regulation (MiCAR),” 2023.
[22] 강윤희, “가상자산 이용자 보호 등에 관한 법률상 이용자 보호 규정의 검토,” 저스티스, no. 205, pp. 463- 490, 2024.

SDN 기반의 에지/코어 스토리지 및 네트워크 관리 시스템 개발

김기현, 김동균, 김기욱, 조부승
한국과학기술정보연구원
{kkh1258, mirr, wowook, bscho}@kisti.re.kr

Development of SDN-based Edge/Core Storage Management System

Ki-Hyeon Kim, Dongkyun Kim, Kiwook Kim, Buseung Cho
Dept.of KREONET Center, Korea Institute of Scinece and Technology Information

요 약

최근 과학기술분야의 거대연구장비를 운영하고 있는 출연연구소에서는 연구장비로부터 수집된 데이터를 기반으로 컴퓨팅, 스토리지, 네트워크등 다양한 연구인프라를 활용하여 시뮬레이션, 가시화, 추론 등의 연구활동을 통해 연구 결과를 도출한다. 하지만 출연연구소에서는 절차적 그리고 기술적 이슈로 거대연구장비와 연구인프라를 동시에 활용하는 것은 매우 어려운 환경에서, 연구자는 파편화된 연구인프라를 사용하기 때문에 연구장비의 공동 활용을 저하는 물론 연구 생산성 저하의 문제가 발생하고 있다. 이를 해결하기 위해 KISTI에서는 SDN 기반의 에지/코어 스토리지를 활용한 스토리지/네트워크 자동화 관리 시스템을 개발하였다. 이를 통해 출연연구소의 연구장비와 연계하여 연구자는 연구장비로부터 수집된 데이터를 빠르게 전송하고, 전송된 데이터를 에지/코어 스토리지를 통해 효율적으로 활용할 수 있는 환경을 제공하여 출연연구소의 연구생산성은 높이고자 한다.

Ⅰ. 서 론

현재 과학기술연구분야에서 거대 연구장비를 운영하고 있는 다양한 출연연구소에서는 거대 연구장비로부터 발생하는 데이터를 수집 및 저장하여 이를 활용하고 있다. 하지만 저장된 데이터를 연구자가 사용하기 위해서는 저장된 스토리지로부터 자신의 연구환경으로 전송할 수 있는 시스템이 필요하고, 이를 빠르게 전송하기 위한 네트워크 또한 필요하다. 하지만 이와 같은 거대 연구장비를 운영하고 있는 출연연구소에서 파편화된 연구인프라(컴퓨팅, 스토리지, 네트워크)를 거대 연구장비와 연동하여 사용하기 위해서는 매우 많은 노력이 필요하다. 이는 기관의 보안적인 절차 또한 복잡하고, 이런 연구인프라와 연구장비를 연동하는 것 또한 매우 복잡하여 길게는 1년 이상 짧게는 6개월이 걸리는 것이 현실이다. 이와 같이 연구 인프라와 연구장비를 자동화할 수 있는 기술적인 한계가 있으며, 수동적이고 파편화된 연구 환경을 사용하다 보니, 연구 생산성 저하의 문제가 계속적으로 발생하고 있다. 이를 해결하기 위해 KISTI에서는 SDN 기반의 에지/코어 스토리지를 활용한 데이터 전송 자동화 시스템을 개발하였다.

KISTI에서는 2015년부터 KREONET[1]의 소프트웨어화를 위한 KREONET-S[2] 프로젝트를 계속적으로 진행해왔다. KREONET-S는 국내 최초로 구축된 소프트웨어 정의 광역연구망으로, 첨단과학기술연구와 응용연구를 위해 요구되는 연구협업 적시성을 제공하고자 종단간 온디맨드 가상화/지능화 프로그래머블 네트워크를 구현하기 위해 설계 및 구축되었다. KREONET-S 네트워크 인프라는 현재 국내 대전, 서울, 부산, 광주, 창원 지역망센터에 구축되어 있고, 국제적으로는 시카고, 시애틀, 홍콩에 구축되어 네트워크를 구성하고 있다. KREONET-S는 개방형 SDN(Sofrware Defined Network) 컨트롤러인 ONOS[3] 컨트롤러를 클러스터화 하여 컨트롤 플레인을 제공하며, SDN 장비는 표준 Openflow 프로토콜을 지원하는 장비들로 구성하여 데이터 플레인을 구성하였다. 그리고 KREONET-S 인프라를 기반으로 제어 플레인에서 실행중인 제어 어플리케이션을 통해 네트워크를 프로그램할 수 있으며, 이를 통해 어플리케이션 계층을 구성하였다. 개발된 어플리케이션 중에 가상전용망(Virtual Dedicate Network, VDN)[4]이라는 유무선 네트워크 슬라이싱 기술을 개발하여 사용자들에게 전용의 회선을 제공하고 이를 통해 보안적으로 안전한 네트워크 환경을 제공하고 있다.

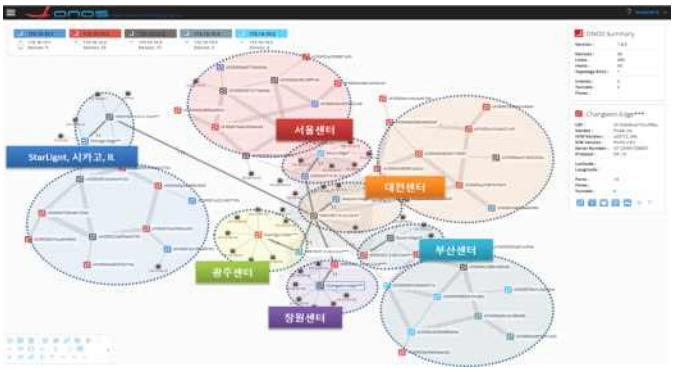

그림 1. KREONET-S 네트워크 구성 현황

본 논문에서는 KREONET-S 인프라 기반의 코어/에지 스토리지와 네트워크를 관리하기 위한 시스템을 개발하였으며, 이를 통해 기존의 출연연연구소에서 관리하는 연구장비들과 연계하여 연구자들의 연구생산성 향상에 도움을 줄 수 있다. 2장에서는 KISTI에서 개발한 SDN 기반의 코어/에지 스토리지 관리 시스템에 대해 더 자세하게 설명하고자 한다.

Ⅱ. 본론

SDN 기반 코어/에지 스토리지 및 네트워크 관리 시스템은 총 5가지의 기능을 가지고 있다. 5가지의 기능은 권한 체계 기능, 코어 스토리지 관리 기능, 에지 스토리지 관리 기능, 거리 기반의 데이터 자동 전송 기능, 데이터 전송 시 VDN 자동 생성 기능을 가지고 있다. 본 장에서는 5가지의 기능을 자세하게 설명하고자 한다.

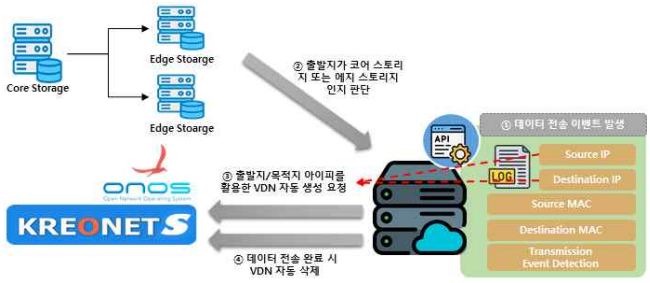

그림 2. 코어 에지 스토리지 및 네트워크 관리 시스템 구성도

첫 번째 기능인 권한 체계 기능은 로그인 기능과 유사하며, 본 시스템에서는 권한별로 나누어 권한을 부여하도록 구성한다. 사용자가 로그인을 수행하면, KREONET 가입기관의 연구자인지 관리자가 확인을 먼저 수행한다. 그 후 검증된 사용자인 경우 3가지 권한으로 나누어진다. 모든 권한을 가지고 있는 시스템 관리자, 기관에서 지정한 기관 스토리지 관리자, 스토리지를 사용하는 사용자로 나누어지며, 관리자는 기관에서 지정한 스토리지 관리자를 지정하여 코어 스토리지를 관리할 수 있는 권한을 부여한다. 스토리지 관리자의 경우 자신의 기관의 스토리지를 사용할 수 있는 사용자의 권한을 부여할 수 있으며, 코어 스토리지에 데이터를 저장, 삭제, 업데이트 등을 수행할 수 있는 권한을 갖이 가지고 있다. 따라서 로그인한 사용자라고 모든 스토리지를 사용할 수는 없다. 기관의 스토리지 관리자의 승인을 받아 사용할 수 있는 비공개적인 서비스로 구성하였다.

두 번째 기능은 코어 스토리지를 관리하는 기능을 제공한다. 코어 스토리지 관리기능은 현재 기관에서 사용하고 있는 물리 스토리지 서버를 소프트웨어 서버로 구성해야하는 과정이 필요하다. Ceph[5], Rustre[6], MiniO[7] 3가지 중 기관에서 원하는 스토리지로 구성하면 된다. 만약 기관에서 스토리지로 구성을 완료했다면, 스토리지 관리 시스템에서 현재 사용하는 스토리지의 기관이름, Access Key ID, Secret Access Key, 스토리지의 도메인 주소를 입력하여 시스템에 등록하는 절차가 필요하다. 등록을 완료할 시 사용자들은 코어 스토리지에 업로드된 데이터를 내 컴퓨터로 다운로드 받을 수 있는 환경이 제공된다.

세 번째 기능은 에지 스토리지 관리 기능을 제공한다. 에지 스토리지는 기관에 있는 스토리지는 아니고, 외부에서 데이터를 다운로드 받는 사용자를 위한 스토리지로써, 연구원에서 분원을 가지고 있거나, 협업을 진행하고 있는 시관이 있는 경우 사용할 시 매우 빠르게 데이터를 다운로드 받을 수 있다. 에지 스토리지를 구성하는 경우 외부 기관에 에지 스토리지를 코어 스토리지와 유사하게 Ceph, Rustre, MiniO 3가지 중 기관에서 원하는 스토리지로 구성하면 된다. 구성을 하게 되면, 시스템 관리자가 에지 스토리지를 등록하는 절차가 필요하다. 코어 스토리지와 유사하게 노드의 이름, Access Key ID, Secret Access Key, 스토리지의 도메인 주소, 스토리지의 용량, 위도, 경도를 입력하여 에지 스토리지를 생성할 수 있다. 생성된 에지 스토리지는 코어 스토리지 생성 시 포함하여 같이 생성되도록 설계하였다.

네 번째 기능은 거리 기반의 데이터 자동 전송 기능이다. 에지 스토리지를 구성할 시 당연히 코어 스토리지의 용량과 에지 스토리지의 용량은 당연히 차이가 존재할 것이다. 따라서 코어 스토리지의 모든 데이터를 에지 스토리지로 전송할 수 없기 때문에 알고리즘을 개발하여 사용자가 자주 사용하는 데이터를 기반으로 데이터를 전송하도록 설계하였다. 알고리즘은 그림 3과 같이 설계하여 개발하였다. 코어 스토리지의 데이터가 존재하고 에지 스토리지에서 존재하지 않는 경우를 판단하여 데이터를 전송한다. 이 때 본 시스템에서는 사용자의 위치 정보 수집하도록 구성되어 있다. 따라서 로그인한 사용자의 위치정보를 활용하여 데이터를 사용자가 데이터를 가까운 스토리지로부터 다운로드 받을 수 있도록 개발하였다. 이에 따라 사용자의 위치를 기반으로 코어 스토리지와의 거리를 계산하고, 에지 스토리지와의 거리를 계산하여 더 가까운 스토리지로부터 데이터를 다운로드 받을 수 있도록 구성하였다.

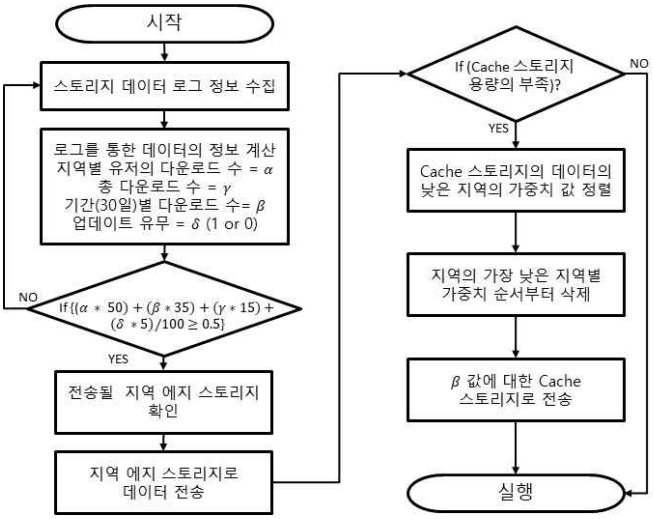

그림 3. 에지 스토리지 데이터 자동 전송 알고리즘

다섯 번째 기능은 데이터 전송 시 VDN 자동 생성 기능이다. 사용자가 데이터를 전송하기 위해서는 스토리지와 사용자 간 데이터를 전송할 수 있는 VDN을 생성해주어야 한다. 따라서 사용자가 코어 스토리지에서 자원을 다운로드 받기 위해 전송 버튼을 누를 시 발생하는 이벤트를 기반으로 사용자의 아이피를 수집하고, 수집된 아이피를 기반으로 출발지는 코어 또는 에지 스토리지로 설정하고, 목적지는 사용자의 아이피를 기반으로 목적지를 설정하여 VDN을 생성한다. 기본적으로 VDN은 한번 생성하여 자원을 선점하고 있으면, 추후에 사용자가 증가할 경우 VDN의 자원 부족으로 네트워크 자원을 할당 받지 못하는 경우가 발생할 수 있다. 이를 위해 데이터를 전송할 때만 VDN을 생성하고, 데이터가 모두 전송된 후에는 VDN을 자동으로 삭제하여 필요시에만 사용할 수 있도록 개발하였다.

에지/코어 스토리지 및 네트워크 관리 시스템은 대형 장비로부터 데이터를 수집하는 출연연구소에 적용하여 데이터 전송에 도움을 줄 수 있을 뿐만 아니라 네트워크를 내부적으로 관리하지 않아도 네트워크 자원을 자동으로 할당 및 삭제하기 때문에 관리 측면에서도 매우 편리하다. 또한 데이터 전송 시에만 네트워크를 할당하기 때문에 데이터를 전송 후에는 네트워크가 자동으로 막혀 보안적으로도 안전한 네트워크를 제공할 수 있다. 또한 위의 기능들은 모두 API 형태로 구성되어 추후 네트워크 자동화 기술로 활용될 수 있도록 개발하였다.

Ⅲ. 결론

최근 거대 연구장비를 운영하고 있는 다양한 출연연구소에서는 연구 인프라를 연동하여 자동화된 연구 환경을 구성하기 위해 노력하고 있다. 하지만 파편화된 연구 인프라와 수동적인 연구 환경은 연구자들의 연구 생산성의 저하 문제를 가지고 있다. 이러한 문제를 해결하기 위해 KISTI에서는 SDN 기반의 에지/코어 스토리지 및 네트워크 관리 시스템을 개발하였다. 본 시스템은 총 5가지의 기능으로 구성되어 있다. 권한체계 기능(로그인 기능), 코어 스토리지 관리 기능, 에지 스토리지 관리 기능, 거리 기반의 데이터 자동 전송 기능, 데이터 전송 시 VDN 자동 생성 기능으로 구성되어 있다. 에지/코어 스토리지 및 네트워크 관리 시스템은 SDN 네트워크 기반으로 구성되어 스토리지에서 데이터 전송 시 네트워크 자원을

자동으로 할당하며, 코어 스토리지에 저장되어 있는 데이터를 에지 스토리지로 전달하여 사용자의 지역적으로 가까운 스토리지에서 데이터를 다운로드 받기 때문에 빠른 데이터 전송 환경을 제공하는 시스템이다. 추후 본 시스템을 출연연구소에 적용하여 연구 장비와 연동하여 네트워크 자원을 자동으로 할당하고 빠른 데이터 전송 환경을 제공하고자 한다.

ACKNOWLEDGMENT

이 논문은 2025년도 한국과학기술정보연구원(KISTI)의 기본사업으로 수행된 연구입니다. (과제번호: K25L5M1C1)

참 고 문 헌

[1] KREONET, https://www.kreonet.net/, 03. 2025

[2] 김동균, 김용환, and 김기현. "KREONET-S 의 SDN 기반 지능기술 구축개발현황 및 계획." OSIA Standards & Technology Review Journal 31.4 ,31-38, 2018.

[3] ONOS, https://opennetworking.org/onos/, 03, 2025.

[4] 김용환, 김기현, 김동균. "SD-WAN 기반 첨단연구망에서의 가상전용망 서비스 설계 및 구현." 한국통신학회논문지 42.10, 2050-2064, 2017.

[5] Ceph, https://ceph.io/, 03, 2025.

[6] Lustre, https://www.lustre.org/, 03, 2025.

[7] MiniO, https://min.io/, 03, 2025.

한국통신학회 통신망운용관리연구회
2025년 통신망운용관리 학술대회 논문집
Proceedings of KNOM Conference 2025

ISSN : 2586-0232(Online)
2025년 4월 24일 인쇄
2025년 4월 24일 발행

발행인/ 김우태 운영위원장
편집인/ 김명섭 출판위원
발행처/ 한국통신학회 통신망운용관리연구회
서울시 서초구 서초동 1330-8 현대기림오피스텔 1504동 6호
전 화 : 02-3453-5555
홈페이지 : www.knom.or.kr
디자인 및 편집 : 고려대학교 NM Lab

한국통신학회 통신망운영관리연구회