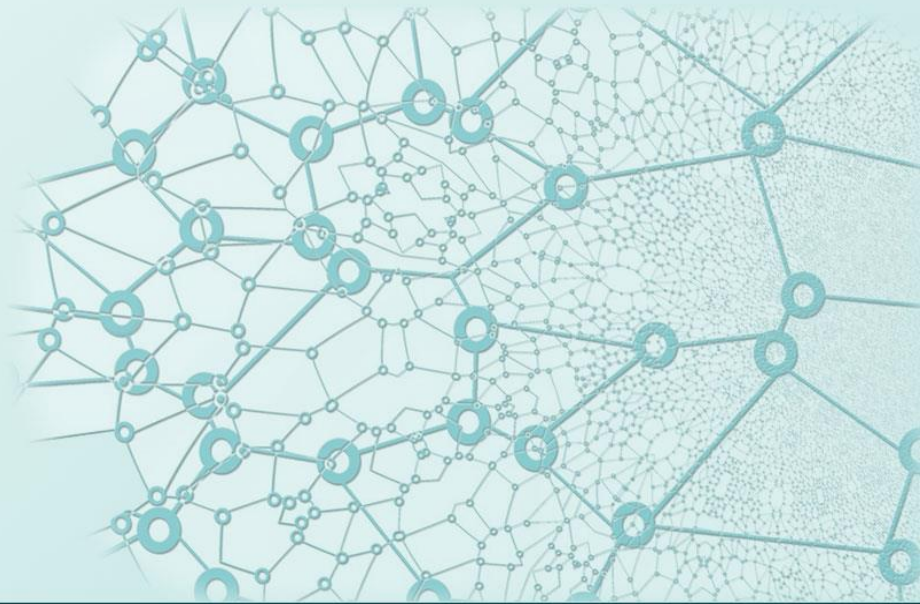


KNOM Review | 28권 2호, 2025년 12월 | ISSN 2287-1543 (Online)

KNOM REVIEW

Network Operations & Management

Vol. 28 No.2 | DEC. 2025



한국통신학회 통신망운용관리연구회

The Committee on Korean Network Operations and Management (KNOM)
Korean Institute of Communication Sciences (KICS)

<http://www.knom.or.kr>

목 차

제28권 제2호, 2025년 12월

NFV 관리를 위한 LLM 기반 Intent Translation 방법 연구 홍지범, 홍원기	1
외부주의와 잔차연결에 기반한 효과적인 네트워크 침입탐지 모델 신창의, 권태웅, 이준	13
볼류메트릭 영상 스트리밍 환경에서 단말 처리 성능이 ABR 알고리즘의 동작에 미치는 영향 분석 유상우, 홍원기	23
세션 및 패킷 번들 단위에서의 WeChat 행위 분석 김지민, 김란아, 장윤성, 백의준, 김명섭	33
원격제어차량 텔레오퍼레이션 및 재해복구를 위한 올인원 휴대용 특화망 장치 구현 최태상, 윤상식, 이준경	41
End-Fog-Cloud 환경에서 데이터 무결성과 접근 제어를 보장하는 데이터 프로비전스 시스템 배태모, 방자원, 황향희, 최미정	51
다중 영역 Kubernetes 환경에서 자체 복구 및 확장을 위한 자율형 PostgreSQL 오퍼레이터 Tahira Amin, Muhammad Asif, Umar Mahmood, 송왕철	67
스테이블코인의 기술적 구조 및 동향 분석 강창훈, 홍원기	78
한국통신학회 통신망운용관리연구회	

NFV 관리를 위한 LLM 기반 Intent Translation 방법 연구

홍 지 범*, 홍 원 기°

A Study of LLM-based Intent Translation Method for NFV Management

Jibum Hong* and James Won-Ki Hong°

요 약

오늘날 네트워크 규모가 커지고 복잡도가 증가하면서, 사용자의 의도 (intent)를 기반으로 네트워크를 효율적으로 관리하기 위한 Intent-based Networking (IBN) 개념이 제안되었다. 최근 대규모 언어 모델 (Large Language Model, LLM)의 발전으로 IBN에 LLM을 적용하여 자연어 intent를 시스템이 해석할 수 있는 구조화된 형태로 변환해 주는 Intent Translation 기술이 연구되고 있다. 본 논문에서는 Intent Translation을 통해 Network Function Virtualization (NFV)를 관리하기 위한 자연어 intent를 추상화 및 구조화하는 템플릿을 생성하는 방법을 제안한다. 제안하는 방법은 프롬프트 엔지니어링 기법 중 Chain-of-Thought (CoT), few-shot learning을 통해 LLM을 최적화하고, 이를 기반으로 Intent Translation을 수행한다. 제안하는 방법을 기존 방법과 여러 LLM 모델을 비교하여 성능을 평가한 결과, 제안하는 방법이 정확도 측면에서 우수한 성능을 보였다.

Key Words : Intent Translation, Large Language Model (LLM), Intent-based Networking (IBN), NFV Management

ABSTRACT

As the scale and complexity of networks increase, the concept of Intent-based Networking (IBN) has been proposed to manage networks based on user intent. With the advent of Large Language Models (LLMs), Intent Translation, which apply LLMs to IBN and transform natural language intents into structured, abstracted forms that can be interpreted by the system, has been attracted. This paper proposes a method for generating templates that abstract and structure natural language intents to manage Network Function Virtualization (NFV) environment through Intent Translation. The proposed method optimizes an LLM using domain adaptation techniques such as Chain-of-Thought (CoT) and few-shot learning. We evaluate the proposed method by comparing it and existing methods. The experimental results showed that the proposed method can significantly enhance the translation accuracy.

※ 이 논문은 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (RS-2024-00392332, 6G 네트워크 통합 지능평면 기술 개발).

• First Author : Pohang University of Science and Technology (POSTECH), Department of Computer Science Engineering, hosewq@postech.ac.kr

° Corresponding Author : POSTECH, Department of Computer Science Engineering, jwkhong@postech.ac.kr

I. 서론

오늘날 네트워크 가상화 (Network Virtualization) 기술의 발전과 cloud-native 인프라의 확산으로, 기존 전용 장비를 통해 폐쇄적으로 운용되던 네트워크 기능 (Network Functions)을 상용 서버에서 소프트웨어 형태로 운용이 가능하게 하는 NFV (Network Function Virtualization)는 오늘날 네트워크 관리 및 운용의 핵심 기술로 활용되고 있다.

이러한 NFV 환경은 네트워크 관리의 확장성과 유연함을 제공하였지만 네트워크 서비스가 다양해지고 수요가 증가함에 따라 네트워크 환경에서 다수의 VNF (Virtual Network Function)와 다양한 서비스를 운용하기 때문에, 네트워크 및 서비스의 배포, 스케일링, 보안/네트워크 정책 설정 등과 같은 관리 작업의 복잡도가 크게 증가한다 [1]. 그러나 여전히 네트워크 관리자 및 서비스 운영자는 이러한 관리 문제를 해결하기 위해 네트워크 관리 정책 또는 구성 (configuration)을 직접 관리하며, 이는 높은 전문성과 많은 수작업을 요구한다.

이러한 문제를 해결하기 위해, 인공지능 (Artificial Intelligence, AI)을 활용한 네트워크 관리 및 최적화 방법에 대한 연구와 intent를 기반으로 네트워크를 자동으로 구성하고 관리하는 Intent-based Networking (IBN) 개념이 제안되었다 [2]. 특히, IBN은 사용자가 요구하는 고수준 의도 (intent)를 기술하고, 시스템이 네트워크 관리 정책과 자원 구성을 자동으로 제공하는 것을 목표로 한다.

그러나 기존 접근 방법은 도메인별로 설계된 제한적인 Intent 언어 또는 고정 템플릿에 의존하거나, 규칙 (rule) 기반으로 파싱하여 저수준 정책 구성으로 변환하는 방식을 취하기 때문에, 다양한 시나리오나 자연어 기반의 사용자 intent를 유연하게 처리하기 어렵다.

한편, 대규모 언어 모델 (Large Language Model, LLM)의 발전으로 AI 분야에서 자연어 처리 능력이 향상되면서, 최근 LLM을 기반으로 프로그래밍 코드를 생성하거나 CLI와 같은 명령어, 또는 설정 파일을 자동으로 생성하는 연구들에 대한 시도가 증가하고 있다 [3]. LLM은 네트워크 관리 분야의 관련 용어와 개념을 이

해하고 예시 구성을 어느 정도 생성할 수 있지만, 자연어 intent로부터 네트워크 관리에 대한 정책이나 구성 생성 이전 단계인 Intent Translation 단계에서도 LLM의 hallucination으로 인해 잘못된 정책이나 구성을 생성한다.

본 논문은 자연어 intent에서 실행 가능한 정책 생성의 이전 단계인, 사용자의 자연어 intent를 추상화, 정형화된 중간 표현으로 변환하는 Intent Translation 단계를 다룬다. Intent Translation은 사용자가 자연어로 기술한 intent를 분석하여, NFV 환경의 관리 작업 유형 (Deployment, Service Function Chaining (SFC), Auto-scaling, Security Configuration, Network Configuration), 대상 인스턴스와 서비스, 자원 요구사항, 네트워크·보안 제약, Service Level Objective (SLO) 정보를 포함하는 JSON 기반 템플릿 (translated intent)을 생성한다. 이 translated intent는 이후 정책 생성 단계에서 실제 실행 가능한 네트워크 관리 정책을 생성할 때, 관리 및 운용하는 서비스가 정의된 Network Service Descriptor (NSD) [4]와 유사하게 intent의 대상 정책 및 자원에 대한 정의를 참조하는 템플릿으로 사용될 수 있으며, 자연어 이해와 정책 생성 과정을 분리함으로써 다양한 NFV 환경에 대한 확장성, 견고성과 재사용성을 높일 수 있다.

논문의 주요 기여는 다음과 같다:

- 네트워크 관리 작업을 표현하기 위해 관리 유형, 대상, 자원 요구, 제약 조건 등을 포함하는 JSON 형식 기반 intent 추상화 및 정형화 템플릿을 정의하는 Intent Translation 방법을 제안한다.
- 자연어 Intent가 정책 생성 및 관리 단계에서 사용될 수 있도록 intent를 추상화 및 정형화하여 LLM 기반 Intent Translation 방법을 설계 및 구현하고, LLM의 도메인 최적화 기법을 사용하여 정확도를 향상시키는 방법을 제시한다.
- 다양한 유형의 네트워크 관리 작업으로 구성된 intent 데이터셋을 구축하고, 제안 방법과 기존 방법의 정확도 및 처리 시간 성능을 비교, 분석하여 제안하는 Intent Translation 방법을 정량적으로 검증한다.

II. 관련 연구

2.1. LLM 및 도메인 최적화 기법

LLM은 대규모 텍스트 코퍼스 (corpus)를 기반으로 사전 학습 (pre-training)되어 자연어 이해와 생성, 요약, 질의응답 등 다양한 자연어 처리 작업을 단일 모델로 수행할 수 있는 범용 모델로 발전해왔다. 최근 LLM은 단순 문장 단위의 응답 생성뿐 아니라, 문서 단위의 텍스트를 이해하고 분석하거나 복잡한 지시를 해석하고, 단계적인 추론을 수행하며, 멀티 모달리티 데이터를 지원하는 기능으로 확장되고 있다. 이에 따라 LLM이 네트워크 관리와 같은 분야에도 활용될 가능성이 높아지며 주목을 받고 있다. 특히, 사용자가 자연어로 표현한 요구를 분석하여 기계가 처리 가능한 형식으로 변환하거나, 로그 분석을 통해 네트워크를 모니터링하고 관리하는 작업은 LLM의 자연어 이해 및 생성 능력이 네트워크 관리 분야에 직접적으로 기여할 수 있는 대표적인 응용 사례로 볼 수 있다 [5].

그러나 LLM은 사전학습된 언어 패턴과 상식과 같은 일반적인 지식에는 강점을 가지는 반면, 네트워크와 같은 특정 도메인에 특화된 용어, 제약 조건, 출력 형식에 대해서는 사전학습된 지식이 부족하거나 기술에 대한 최신 정보 반영이 어렵기 때문에 ChatGPT와 같이 기본적으로 제공되는 LLM을 그대로 네트워크 사용하기 어렵다는 한계가 있다.

이를 보완하기 위해, LLM을 사용 목적에 맞게 도메인 적응 (domain adaptation) 또는 도메인 최적화하는 다양한 기법들이 제안되었다. 대표적으로, 추가적인 학습 없이 프롬프트의 설계를 최적화하는 프롬프트 엔지니어링 (Prompt Engineering) [6]에는 단계적 추론을 위해 프롬프트를 설계하는 Chain-of-Thought (CoT) [7], 입력 프롬프트 내에 관련 예시를 함께 제공하는 few-shot learning [8], 목표로 하는 도메인 데이터로 추가 학습을 통해 LLM의 파라미터를 전체적으로 재조정하는 미세 조정 (fine-tuning) [9], 그리고 추가적인 도메인 지식을 검색하는 Retrieval Augmented Generation (RAG) [10]의 활용, self-refinement, Reinforcement Learning from Human Feedback (RLHF)과 같이 사람의 개입이나 모델이 자체적으로 LLM의 출력을 조정하는 Feedback [11] 매커니즘을 예로 들 수 있다.

프롬프트 엔지니어링 기법은 모델 파라미터를 변경하지 않고도 프롬프트 설계를 통해 도메인 지식을 주입하거나 LLM을 목적에 맞추어

최적화를 할 수 있다는 점에서 별도의 학습 인프라나 자원을 요구하지 않는다. 특히, 도메인에 특화된 작업을 수행하기 위해 프롬프트를 설계하거나 관련 예시를 프롬프트에 포함하면, LLM이 해당 도메인에 맞게 프롬프트를 기반으로 해석 및 추론하고, 그에 따른 출력을 생성하도록 유도할 수 있다. 반면, fine-tuning 기법은 도메인 데이터가 일정 규모 이상 확보되고, 추가 학습 환경을 구성할 수 있을 때 보다 안정적이고 일관된 성능을 기대할 수 있지만, 학습을 위한 초기 비용 등 오버헤드가 크다.

본 논문에서 다루는 Intent Translation은 자연어로 표현된 네트워크 관리 Intent를 JSON 기반의 translated intent라는 중간 표현으로 변환하는 작업이다. 따라서 본 연구에서는 대규모 파라미터 학습이 필요한 미세 조정보다는, CoT 기반 프롬프트 설계와 few-shot learning을 통한 도메인 최적화 기법에 초점을 맞추어 LLM을 Intent Translation 작업에 활용한다.

2.2. 관련 연구

IBN은 사용자의 고수준 요구사항을 기반으로 네트워크 관리 정책을 자동으로 생성하고 적용한다. 네트워크 규모와 복잡도가 증가함에 따라 네트워크 관리자의 부담이 커지고 있기 때문에, 이러한 관리 부담을 줄이기 위한 방법으로 IBN이 큰 관심을 받아왔다. 초기 IBN 연구들은 주로 규칙 기반 접근법을 사용하여 사전에 정의된 템플릿을 기준으로 intent를 해석하고, 네트워크를 구성하였다 [12]. 그러나 이러한 방식은 네트워크 환경의 동적 변화에 유연하게 대응하기 어렵고, 새로운 서비스나 정책을 반영할 때 확장성이 떨어진다는 한계를 가진다.

이를 보완하기 위해, 자연어로 표현된 intent를 정책이나 설정으로 변환하기 위해 Natural Language Processing (NLP)와 LLM을 활용하는 연구들이 제안되었다. 특히, LLM의 등장으로 자연어 기반 intent를 보다 정교하게 해석하고, 이를 정책이나 네트워크 설정 (configuration)의 생성 정확도를 크게 향상시킬 수 있을 것으로 기대된다.

이러한 흐름에서 LLM을 활용한 intent 기반 네트워크 관리 연구가 제안되고 있으며, 특히, Intent Translation에 초점을 두는 연구들이 다수 보고되고 있다. Dzevaroska et al. [13]은

OpenStack 환경에서 사용자의 자연어 intent를 인스턴스 배포 정책으로 변환하는 LLM 기반 시스템을 제안하였다. 이 연구에서는 LLM을 이용해 intent 분류 단계에서 사용자의 요구사항에 내포된 작업(task)과 동작(action)을 식별한 뒤, 이를 OpenStack API에 매핑되는 정책으로 변환한다. 하지만 해당 연구는 인스턴스 배포(deployment)와 상태 점검(health check) 기능에 한정되어 있으며, 오토스케일링, 보안 정책, 네트워크 설정과 같은 관리 기능은 다루지 못한다는 한계를 가진다.

Mekrache et al. [14]은 Code LLaMA와 few-shot learning을 사용하여 서비스를 배포하는 LLM 기반 방법을 제안하였다. 이 연구에서는 자연어 intent를 NSD 템플릿으로 변환하는 Intent Translation을 수행한다. Wang et al. [15]는 B5G/6G 네트워크를 대상으로 LLM을 이용해 자연어 intent를 기계가 읽을 수 있는 정책으로 변환하는 개념적 설계를 제안하였다. 이 연구는 프롬프트 엔지니어링, RAG, fine-tuning을 활용하여 LLM이 실시간으로 네트워크 트래픽과 로그를 분석하고, 네트워크를 표현하는 AI 모델과 결합하여 의사결정을 위한 정책을 생성하는 방안을 제시한다. 그러나 이 연구는 개념 설계 수준에 머물러 구현 및 실험을 통한 검증은 제시하지 않는다.

Tu et al. [16]은 in-context learning, 특히 few-shot learning을 사용하는 LLM 기반 intent translation 방법을 제안하였다. 이 연구에서는 자연어 intent를 NFV 설정 템플릿으로 변환하며, OpenStack 환경에서 사용되는 NFV 설정 템플릿 (인스턴스 deployment, SFC)을 생성한다. Nam et al. [17]은 LLM을 이용한 VNF 생성 및 설치 방법을 제안하였다. 이 연구에서는 Method of Procedure (MOP) 문서를 활용하여 설치 가이드를 분석하고, OpenStack 환경에서 실행 가능한 VNF 설치 명령어 및 설정을 Python 코드 형태로 자동 생성한다.

위와 같은 관련 연구들은 공통적으로 자연어 기반 intent를 형식화된 템플릿이나 설정으로 변환하는 Intent Translation을 다루지만, 그 범위나 환경이 제한적이라는 한계가 있다. 무엇보다 대부분 OpenStack 또는 특정 5G 테스트베드 환경을 대상으로 인스턴스 배포, NSD 템플릿 생성, VNF 설치 명령 생성 등 일부 기능에 초점을 맞추고 있다. 반면, 본 연구는 오토스케일링, 보

안 설정, 네트워크 설정과 같이 NFV 관리에서 빈번하게 요구되는 다양한 관리 작업에 대한 intent translation을 수행한다. 또한, translated intent의 구조를 NFV 관리 전반을 포괄하는 공통 템플릿으로 체계적으로 정의하고, 다양한 관리 작업과 LLM 모델을 대상으로 정확도와 처리 시간을 비교 및 분석한 연구도 상대적으로 부족하다. 본 연구는 NFV 관리를 위한 정책 구성 관련 작업 (서비스 및 인스턴스 deployment, SFC 구성, auto-scaling, security/network configuration)을 포함하는 intent 데이터를 통해 JSON 기반 추상화 및 구조화 템플릿을 생성하는 방법을 설계하고, LLM 기반 Intent Translation 방법을 체계적으로 평가한다.

III. LLM 기반 Intent Translation 설계 및 구현

3.1. 개요

본 장에서는 NFV 환경에서 사용자의 자연어 기반 intent를 실제 실행 가능한 정책으로 변환하기 이전 구조화된 템플릿 형식으로 변환하는 LLM 기반 Intent Translation 방법을 설명한다. Intent Translation은 사용자가 입력한 관리 요구(intent)를 분석하여 관리 작업 유형, 대상 인스턴스, 자원 요구사항, 네트워크 및 보안 정책, SLO (Service Level Objective) 제약 등을 포함하는 JSON 객체 형식 기반의 translated intent로 변환한다. 이를 통해 자연어 수준으로 표현된 추상적인 요구를 정책 생성 단계에서 직접 활용 가능한 중간 표현으로 정규화하고, 이후 정책 생성 단계에서 실제 실행가능한 관리 정책을 생성할 때 참조할 수 있는 정형화된 템플릿을 제공한다.

3.2. Intent Data 및 출력 구조 정의

제안하는 Intent Translation의 입력은 사용자가 자연어로 기술하는 NFV 관리 정책 생성 작업에 대한 요청이다. 본 연구에서는 ETSI NFV IFA 및 SOL 표준 문서, 관련 연구, 실제 NFV 기반 서비스 시나리오를 참고하여 NFV 환경에서 반복적으로 수행되는 관리 작업을 정의한다. 이를 바탕으로 VNF 인스턴스 및 서비스 배포 (Deployment), SFC, Auto-scaling, Security 및 Network Configuration 5가지를 관리 작업으로 정의한다.

각 관리 작업에서 실제 운용 환경에서 사용되는 자연어 기반 intent 표현을 수집 및 생성하여 본 연구에서 사용하는 intent 데이터셋을 구성한다. 이때 특정 NFVI에 종속적인 문법 (Kubernetes YAML manifest, OpenStack HoT 등)은 배제하고, “무엇을 하고 싶은지” 사용자의 의도에 중점을 둔 문장으로 intent 데이터를 구성한다. 예를 들어, “Firewall, IDS, DPI 순서로 SFC를 구성하라”, “CPU 사용률이 90%를 초과하면 web-scale 서버에 Auto-scaling 정책을 적용하라”, “제공 중인 특정 서비스 대해 IP 주소 10.10.10.16 및 포트 80을 차단하라”와 같은 NFV 환경을 전반적으로 관리하기 위한 문장이 이에 해당한다. 이러한 intent 데이터의 생성에는 서로 다른 표현, 동의어, 문장 구조를 일부러 섞어서 다양한 자연어 입력에 대해 일반화 성능을 확보한다.

Intent Translation의 출력인 translated intent는 JSON 객체로 표현되는 템플릿 구조를 갖는다. 이 템플릿은 크게 관리 작업 유형 (management task), 수행 동작 (action), 대상 인스턴스 및 리소스 (targets), 자원 요구사항 (requirements), 운영 및 보안 제약조건 (constraints)으로 구성된다.

Deployment 작업의 경우 대상 인스턴스 이름, 이미지, CPU/메모리 등 자원 요청 및 제한 등의 필드를 포함하고, SFC 작업에서는 체인을 구성하는 VNF 인스턴스의 순서와 각 인스턴스의 역할, 연결 방식, 대상 네임스페이스 등의 정보가 포함된다. Auto-scaling 작업에서는 대상 Deployment, 스케일링 조건, 최소/최대 인스턴스 수 등이 포함되며, Security/Network Configuration에서는 정책이 적용되는 대상 서비스 또는 네트워크와 허용 및 차단할 IP 주소, 포트 등 세부 파라미터가 포함된다.

Translated intent 템플릿은 관리 작업 종류에 상관없이 공통된 상위 필드 구조 (task, action, targets, requirements, constraints)를 유지하면서, 각 작업에 특화된 하위 필드를 계층적으로 포함하도록 설계한다. 누락된 정보는 기본값으로 채우고, 잘못된 단위나 표현은 정규화함으로써 자연어 기반 intent 문장의 모호성을 줄일 수 있다. 이와 같이 설계된 구조화된 템플릿을 기반으로 실제 정책 생성 단계에서 템플릿 구조를 참조하여 대상 NFVI에 대한 관리 정책을 생성할 수 있도록 한다.

기존 IBN 및 intent-driven networking 연구에서는 정형화된 텍스트 입력이나 제한된 슬롯 기반 양식을 사용하여 의도를 파싱하는 규칙 기반 또는 heuristic 접근법이 주로 사용된다. 그러나 이러한 방법은 자연어 표현 방법이 달라져도 인식을 실패하거나, 추상적인 관리 목적을 세밀한 자원 및 정책 요구사항으로 해석하는데 한계가 존재한다. 따라서 본 연구에서는 LLM의 자연어 이해 능력을 기반으로 단계별 추론 기반 프롬프트 설계 (CoT), 예시 기반 학습 (few-shot learning)을 적용한 프롬프트 엔지니어링을 사용하여, 다양한 자연어 표현으로 구성된 NFV 관리 intent를 정형화되고 추상화된 translated intent 템플릿으로 자동 변환하는 방법을 제안한다.

3.3. LLM 기반 Intent Translation 설계

제안하는 Intent Translation 방법은 IETF NMRG RFC 9316 [18]에서 정의한 intent 분류 방법론을 기반으로 설계한다. RFC 9316에서는 intent 기반 관리에서 사용자의 의도를 분석하는 절차를 intent solution, user types, intent type, intent scope, network scope, abstraction 등으로 구분하여 정의한다.

본 연구에서는 이 중 각 관리 요청에 대해 intent type, intent scope, network scope를 식별한 뒤, NFV 관리 도메인에 적합한 수준으로 의도를 추상화 (abstraction)까지의 범위를 대상으로 제안하는 Intent Translation 방법을 설계하고 구현한다.

LLM 기반 Intent Translation은 그림 1과 같이 설계된다. 먼저, 사용자의 자연어 intent를 입력

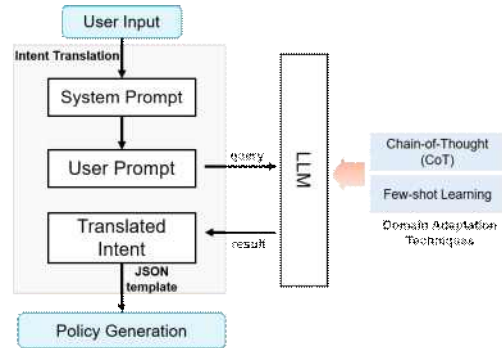


그림 1. NFV 관리를 위한 LLM 기반 Intent Translation 설계
Fig. 1. Design of LLM-based Intent Translation for NFV Management

받으면 CoT 기반 프롬프트 설계를 적용하여 구성된 System Prompt를 통해 LLM이 입력된 intent가 어떤 관리 작업 (tasks)에 해당하는지 (Deployment, SFC, Auto-scaling, Security Configuration, Network Configuration), 어떤 동작 (actions)을 수행해야 하는지 (create, update, delete), 어떤 리소스와 네트워크 범위를 대상으로 하는지 (대상 서비스 이름, 대상 인스턴스 이름 등)를 단계적으로 추론하도록 유도한다. 이때 System Prompt는 RFC 9316에서 정의된 Intent Classification 방법을 참고하여, LLM이 각 관리 intent의 유형과 범위를 명시적으로 분리해서 판단하도록 프롬프트 규칙을 서술한다.

다음으로, User Prompt에는 각 관리 작업별 설계된 Use-case 예시 (few-shot examples)를 포함한다. User Prompt는 각 작업마다 1개의 예시를 정의하고, 각 예시는 자연어로 표현된 intent와 해당 intent에 부합하는 translated intent (JSON 객체) 쌍으로 구성된다. 예를 들어 Deployment 작업은 웹 호스팅 서비스 배포, 특정 VNF 인스턴스 배포와 같은 시나리오를 포함한 JSON 템플릿으로 구성되고, SFC 구성은 SFC 구성 순서, SFC를 구성하는 인스턴스 요구 사항 등이 포함된다.

이와 같은 CoT와 few-shot 기반 프롬프트 엔지니어링 설계의 목적은, LLM이 자연어 intent를 단순히 요약하거나 재서술하는 것이 아니라, NFV 관리 도메인에서 의미 있는 필드들 (작업 유형, 대상, 자원/네트워크/보안 요구사항, 제약 조건 등)을 구조화된 템플릿 형태로 추출하도록 유도하는 데 있다. 또한 intent type과 scope, network scope를 명시적으로 식별하게 함으로써, 향후 실제 정책 생성 관련 연구 단계에서 intent가 목표로하는 네트워크 및 자원에 대한 정책을 생성해야 하는지를 일관성 있게 해석할 수 있도록 한다.

3.4. LLM 기반 Intent Translation 구현

상기 LLM 기반 Intent Translation 설계를 통해 구현된 Intent Translation을 위한 LLM 기반 파이프라인은 그림 2와 같다. 먼저, 프롬프트 엔지니어링 기법을 통해 정의된 System Prompt와 User Prompt를 로드하고, 사용자가 입력한 자연어 intent를 user prompt에 삽입하여 System Prompt와 User Prompt가 결합된 최종 프롬프트

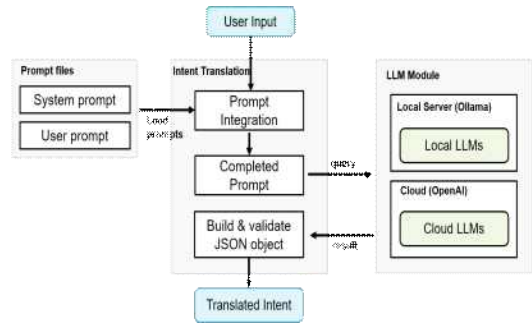


그림 2. LLM 기반 Intent Translation 파이프라인 구현
Fig. 2. Implementation of LLM-based Intent Translation

를 구성한다. System Prompt는 CoT를 기반으로 설계된 정의 및 규칙을 포함하여 LLM이 생성해야 할 JSON 객체의 구조와 단계별 추론 과정을 정의한다. User Prompt에는 관리 작업마다 1개의 few-shot 예시가 포함된다.

프롬프트가 구성되면, Intent Translation은 LLM 모듈에 프롬프트를 기반으로 요청을 전달하여 LLM이 생성한 응답을 받는다. 본 연구에서는 Ollama [19] 플랫폼을 사용하여 로컬 서버에서 실행되는 LLaMA 3.3 (70b) 모델을 사용하였으며, 로컬 환경에서 운용 가능한 LLM 모델과 클라우드 환경에서 운용 가능한 대형 LLM 모델의 성능을 4장에서 비교하기 위해 OpenAI [20] 플랫폼에서 제공하는 API를 사용하여 GPT-4o 모델을 사용한 파이프라인도 구현한다.

LLM 모델의 설정은 일관된 응답 생성을 위해 temperature, 최대 토큰 수, 컨텍스트 길이 등의 값을 설정하였다. 상기 로컬 및 클라우드 기반 LLM 호출 프로세스의 구현에는 LangChain [21]을 사용하여 프롬프트 구성부터 LLM 요청, 응답 수신을 위한 파이프라인을 구성하였고, 해당 파이프라인은 LLM이 생성한 결과물을 기반으로 JSON 객체를 생성하는 과정을 포함한다.

LLM이 생성한 응답은 때때로 설계 의도와는 다르게 자연어 설명 (instruction)과 JSON 객체 코드가 섞인 형태의 응답을 출력할 수 있다. 따라서 본 연구에서 제안하는 Intent Translation의 구현에는 설계 의도에 맞는 intent가 추상화된 JSON 템플릿 출력을 위해 LLM의 출력을 기반으로 Post-Processing를 통해 JSON 객체를 생성하고 검증하는 단계를 거쳐 최종 출력인 translated intent를 정제한다.

이 단계에서는 먼저 LLM의 출력을 기반으로

JSON 객체만을 추출하고, 코드 설명과 같이 JSON 객체 생성에 불필요한 텍스트를 필터링한다. 이후 필수 필드(task, action, targets, requirements, constraints 등)의 존재 여부와 계층적 필드 구조의 유효성을 점검하여 구조적 무결성을 검증한다. 검증에 실패할 경우에는 예외 처리 후 재생성을 요청하고, 디버깅 목적의 로그로 저장한다. 검증을 통과한 JSON 객체는 최종 출력으로 채택되어 JSON 템플릿 파일로 저장된다.

IV. 실험 및 검증

4.1. 실험 환경 및 구성

제안하는 Intent Translation 방법에 대한 검증 및 평가를 위해 실험 환경을 구축한다. 실험 환경은 3장에서 구현한 Intent 데이터셋과 Intent Translation 파이프라인을 사용한다.

Intent Translation은 자연어 intent를 JSON 템플릿으로 변환하는 단계이므로, 변환된 translated intent가 NFVI에서 어떤 자원과 정책으로 매핑되는지는 실제 정책 생성 단계에서 다뤄지기 때문에 OpenStack, Kubernetes와 같은

실제 NFV 환경 구축을 통한 검증 및 평가가 아닌 생성한 Intent 데이터셋을 기반으로 LLM의 생성 정확도를 평가한다. 이를 위해 실험에서는 translated intent가 실제 테스트베드 상의 관리 작업 (Deployment, SFC, Auto-scaling, Security, Network Configuration)으로 자연스럽게 대응되도록 intent 데이터와 템플릿 구조를 설계하여, 실제 네트워크 운용 시나리오와 괴리되지 않도록 데이터셋을 구성하고 Intent Translation 성능을 평가한다.

Intent Translation 성능 측정에는 두 종류의 자연어 기반 intent 데이터셋을 사용하였다. 첫 번째 데이터셋은 이전에 수행된 Intent Translation 선행연구 [16]에서 OpenStack 기반 NFV 환경을 대상으로 사용한 40개의 intent 데이터셋으로, Deployment와 SFC 두 가지 관리 작업만을 포함한다. 이 데이터셋은 제안하는 Intent Translation 방법과 이전 연구, 그리고 성능 평가를 위해 규칙 기반 heuristic 방법을 통해 구현된 Intent Translation 직접 비교하기 위해 사용된다. 두 번째 데이터셋은 본 연구에서 대상으로 하는 NFV 관리 작업을 기반으로 62개로 구성된 intent 데이터셋으로, 해당 데이터셋은 Deployment, SFC, Auto-scaling, Security, Network Configuration

표 1. Intent Translation 실험 평가에 사용된 LLM 모델
Table 1. Detailed LLM models for evaluation of Intent Translation experiment

LLM	Model Names (Parameters)	Used Model Descriptions
Mistral [22]	mistral:7b (7b)	- Mistral AI (2023.09) - Lightweight model
Phi4 [23]	phi4-reasoning:plus (14b)	- Microsoft (2025.04) - Lightweight, reasoning model
GPT-OSS [24]	gpt-oss:20b (20b)	- OpenAI (2025.08) - Open-weight model of GPT, support reasoning
Gemma3 [25]	gemma3:27b (27b)	- Google (2025.03) - Multi-modal/lingual model
Qwen3 [26]	qwen3-coder:30b (30b)	- Alibaba (2025.04) - Optimized for coding tasks, support reasoning
DeepSeek [27]	deepseek-r1:70b (70b)	- DeepSeek (2025.01) - High-capacity open model, support reasoning
LLaMA3.1 [28]	llama3.1:70b (70b)	- Meta (2024.07) - High-capacity open model, support reasoning
LLaMA3.3 [29]	llama3.3:70b (70b)	- Meta (2024.12) - Improved performance compared to llama3.1
GPT-4o [30]	gpt-4o (200b, estimated)	- OpenAI (2024.05) - Closed-weight model - Support full-cot and multi-modal reasoning

5가지 관리 작업마다 10개, combined task 12개로 구성된다. 이 데이터셋은 이전 연구에서 생성된 데이터셋보다 다양한 작업 유형과 난이도 수준에서 Intent Translation의 성능을 평가하기 위한 용도로 사용된다.

실험에 사용된 LLM 모델은 모두 Intent Translation에 동일한 형태의 프롬프트를 입력받도록 구성한다. 먼저, 첫 번째 실험에서는 제안 방법과 이전 연구의 Intent Translation 정확도 및 처리시간의 비교를 수행하며, 이를 위해 LLaMA 계열 모델을 사용한다. 이전 연구에서는 LLaMA 3.1 (70b) 모델을 사용하였고, 본 연구에서는 LLaMA 3.3 (70b)을 기본 LLM 모델로 사용하여 성능을 측정한다. 이후 실험에서는 Mistral, Phi4, GPT-OSS, Gemma3, Qwen3, LLaMA3.3, DeepSeek, GPT-4o 등 다양한 규모와 특성을 갖는 8종의 LLM 모델에 대해 동일한 Intent Translation 프롬프트를 적용하여, 모델 종류와 크기에 따른 성능 변화를 측정한다. 실험에 사용된 모델들의 세부 정보는 표 1과 같다.

성능 검증은 로컬 서버에서 실행이 가능한 LLM 모델은 Ollama 플랫폼을 통해 구축된 로컬 서버에서 수행하고, GPT-4o와 같이 공개되어 있지 않고 로컬 서버에서 실행이 불가능한 클라우드 기반 모델은 OpenAI API 호출을 통해 응답 생성을 요청하여 성능 검증을 수행한다. 각 모델은 최대 컨텍스트 길이를 동일하게 설정하고, 실험의 재현성과 일관된 비교를 위해 temperature는 0.0으로 고정한다.

4.2. 실험 1 - 기존 접근법과의 성능 비교

실험 1에서는 규칙 기반 heuristic 접근법을 사용하여 구현한 Intent Translation, 선행연구,

baseline LLM, 제안하는 방법으로 구현한 Intent Translation의 성능을 비교한다. 성능 비교를 위해 데이터셋은 이전 선행연구에서 사용한 데이터셋과 본 연구에서 생성한 데이터셋을 함께 사용한다. 먼저, 선행연구와 본 연구의 성능 비교를 위해, 이전 연구에서 사용한 40개의 intent 데이터셋 (Deployment 및 SFC 작업)을 그대로 사용한다. 이 데이터셋에 대해서는 규칙 기반 방법, 선행연구 (LLaMA3.1, LLaMA3.3), 제안하는 방법 (LLaMA3.1, LLaMA3.3)의 정확도와 처리 시간을 측정한다. 다음으로, 본 연구에서 생성한 62개의 intent 데이터셋 (5가지 관리 작업)을 대상으로 정확도 및 처리 시간을 측정하여 성능을 평가한다.

실험을 통한 정확도 및 처리시간 성능 측정 결과는 표 2와 나타난 바와 같다. 선행연구의 데이터셋을 통한 성능 측정 결과에서 규칙 기반 방법은 약 62.5% 수준의 정확도와 0.41초의 처리 시간을 보이며, 키워드를 중심으로 intent를 파악하고 해석하는 규칙 기반 접근법의 한계를 보였다. 도메인 최적화 기법을 적용하지 않은 기본 (baseline) LLaMA3.3 모델을 사용하여 Intent Translation을 수행했을 때에는 정확도가 75.5%로 향상되지만, 여전히 상당한 수의 펠드 누락 및 해석 오류가 발생하였다.

선행연구의 실험 결과는 9개의 few-shot 예시를 포함하는 few-shot learning 기법을 사용하여 LLaMA3.1에서 87.5%, LLaMA3.3에서 97.5%의 높은 정확도를 달성하지만, 처리 시간은 각각 6.91초, 7.20초로 증가하였다.

같은 데이터셋에 대해 본 연구에서 제안하는 방법 (CoT + few-shot learning)을 적용한 결과,

표 2. 실험1 - Intent Translation 기존 접근법과의 성능 비교 (accuracy, processing time)
Table 2. Experiment 1 - Performance comparison for existing approaches and proposed method

Methods		Rule-based (Heuristic)	LLM-based				
			Baseline LLM (LLaMA3.3)	Tu et al. [16]		Proposed	
				LLaMA3.1	LLaMA3.3	LLaMA3.1	LLaMA3.3
Tu et al. [16] dataset (Dep. + SFC)	Acc.	62.5	75.5	87.5	97.5	90.0	97.5
	Proc. time	0.41	1.83	6.91	7.20	4.57	4.93
Our dataset (5 tasks)	Acc.	59.7	72.6	-	-	90.3	96.7
	Proc. time	0.48	2.16	-	-	5.09	5.51

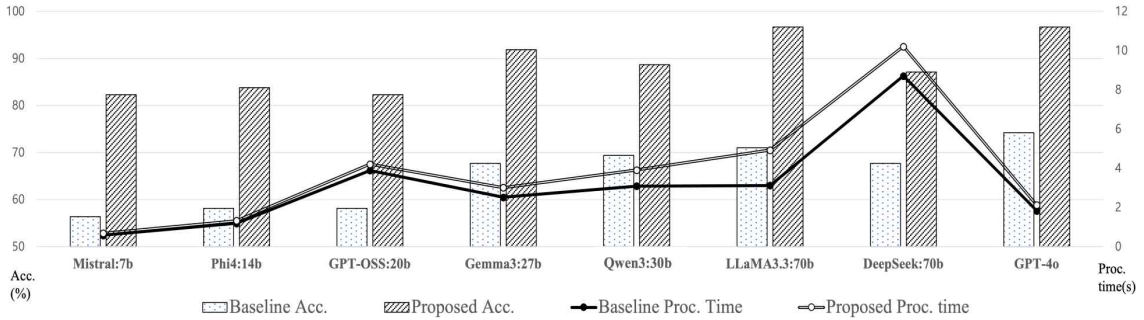


그림 3. LLM 모델별 Intent Translation 성능 측정 결과 (정확도, 처리 시간)

Fig. 3. Performance Comparison of Intent Translation by LLM models (accuracy, processing time)

LLaMA3.1에서는 약 90.0%, LLaMA3.3에서는 97.5%의 정확도를 달성하였다. 즉, 제안하는 방법은 선행연구와 비슷하거나 약간 높은 수준의 정확도를 유지하면서도 처리 시간은 각각 약 4.57초, 4.93초로 감소하였다. 이는 선행연구가 많은 수의 예시를 통해 추론 성능을 향상시킨 반면, 그에 따른 프롬프트 길이 증가로 인해 처리 시간이 크게 증가했다는 것을 의미한다. 반면, 제안하는 방법은 CoT 기반 단계별 추론과 필요한 수준의 예시만을 사용함으로써, 유사한 정확도를 유지하면서도 처리 시간을 줄일 수 있었다.

다음으로, 본 연구에서 생성한 62개의 intent 데이터셋에 대한 결과에서도 각 실험 결과는 유사한 경향을 보였다. 규칙 기반 Intent Translation 방법은 약 59.7%의 정확도와 0.48초의 처리 시간을 보였고, 기본 LLaMA3.3 모델은 약 72.6%의 정확도와 2.16초의 처리 시간을 보였다. 선행연구의 경우에는 Deployment와 SFC만을 대상으로 구현이 되어 있어 본 연구에서 생성한 데이터셋으로는 직접 비교가 불가능하기 때문에 해당 데이터셋에 대해서는 선행연구의 성능 측정이 진행되지 않았다.

제안하는 방법은 LLaMA3.1 기준 약 90.3%, LLaMA3.3 기준 약 96.7%의 정확도가 측정되었다. 이는 다양한 작업과 난이도가 섞여 있는 데이터셋에서도 제안하는 방법은 안정적인 성능 향상을 보였음을 의미한다. 처리 시간 측면에서는 제안하는 방법은 각 모델에서 약 5.09초, 5.51초로 증가했지만, 선행연구 대비하여 여전히 더 적은 처리시간을 보였다.

요약하면, 실험 1에서는 intent를 JSON 템플릿으로

추상화하는 작업에서 제안 방법이 규칙 기반 및 baseline LLM을 사용한 방법보다 높은 정확도를 보였고, 이전 연구와 비교하면 비슷한 수준의 정확도를 유지하면서도 처리 시간을 의미 있게 단축하는 결과를 보여주었다. 이는 NFV 관리 환경에서 다양한 자연어 기반 intent를 실시간에 가깝게 처리해야 하는 상황에서 제안 방법이 보다 실용적인 성능을 제공할 수 있음을 시사한다.

4.3 실험 2 - LLM 종류 및 크기에 따른 성능 비교

실험 2에서는 본 연구에서 제안하는 Intent Translation 방법이 다양한 종류와 크기를 가지는 LLM 모델 전반에서 일관된 성능 향상을 제공하는지 검증한다. 이를 위해 표 1에 설명된 Mistral (7b), Phi4 (14b), GPT-OSS (20b), Gemma3 (27b), Qwen3 (30b), LLaMA3.3 (70b), DeepSeek (70b), GPT-4o (200b 추정) 8종의 LLM 모델을 대상으로, 본 연구에서 생성한 62개 intent 데이터셋에 대해 baseline 방법과 제안하는 방법의 Intent Translation 정확도와 처리 시간을 측정하였다.

먼저, baseline LLM을 통해 Intent Translation의 성능을 측정한 결과, 파라미터 수가 적은 소형/경량화 모델은 Mistral, Phi4, GPT-OSS에서 각각 약 56.4%, 58.1%, 58.1%의 정확도를 보였으며, 약 30b의 파라미터를 사용하는 중형 크기의 모델인 Gemma3, Qwen3는 약 67.7%, 69.4%로 약 10% 높은 정확도를 보였다. 70b의 파라미터를 사용하는 상대적으로 대형 모델인 LLaMA3.3과 DeepSeek는 각각 71.0%, 67.7%의 정확도를 기록하였고, 클라우드 환경에서 API를 통해 수행되는 GPT-4o는 약 74.2%의 가장 높은 정확도를 보였다. 그러나 baseline LLM을 사용한 방법

은 관리 작업의 세부 필드나 제약조건을 반영하지 못하는 사례가 존재하여 단순 baseline LLM 방법만으로는 NFV 관리에 요구되는 수준의 Intent Translation 정확도를 보이지 못했다.

동일한 모델에 대해 제안 방법 (CoT + few-shot)을 적용한 결과, 모든 모델에서 정확도가 크게 향상되었다. 소형 모델의 경우 Mistral은 56.4%에서 82.3%로, Phi4는 58.1%에서 83.8%로 약 20~26%의 정확도 향상을 보였다. 중형 모델인 Gemma3와 Qwen3 역시 각각 67.7%에서 91.9%, 69.4%에서 88.7%로 성능이 크게 개선되었으며, LLaMA3.3은 71.0%에서 96.7%로 약 25%p 이상 향상되었다. DeepSeek는 67.7%에서 87.1%로, GPT-4o는 74.2%에서 96.7%로 향상되어, 클라우드 기반 대형 모델에서도 제안 방법이 높은 정확도 개선 효과를 제공함을 확인할 수 있었다.

처리 시간 측면에서는 프롬프트 복잡도 증가로 인해 모든 모델에서 소폭의 지연이 발생하였다. 예를 들어 Mistral은 평균 0.58초에서 0.68초로, Phi4는 1.19초에서 1.32초로 증가하였다. LLaMA3.3과 DeepSeek는 각각 약 1초 내외의 추가 지연이 발생하였고, GPT-4o는 약 0.3초 정도의 증가를 보였다. 그러나 해당 지연은 대부분 1초 이내 범위에 머물렀으며, 정확도 향상 폭을 고려할 때 Intent Translation 용도에서는 수용 가능한 수준의 오버헤드로 판단된다. 특히 소형·중형 모델의 경우 1초 미만의 처리 시간과 80~90% 이상의 정확도를 동시에 만족하여, 엣지 장비나 자원이 제한된 환경에서도 실용적으로 사용할 수 있는 가능성을 보여준다.

4.4 결과 분석

실험 1의 결과를 보면, 제안하는 LLM 기반 Intent Translation은 규칙 기반 방법과 baseline LLM에 비해 translated intent 생성 정확도에서 뚜렷한 우위를 보였다. 규칙 기반 방법은 처리 시간은 매우 짧지만 다양한 자연어 표현과 복잡한 intent로 인해 약 60% 정확도를 보인 반면, 제안하는 방법은 CoT 기반 단계적 추론과 예시를 통해 기존 LLM 기반 연구와 유사하거나 더 높은 정확도를 보였다. 특히, Deployment와 SFC에 한정된 선행연구의 데이터셋뿐 아니라 Auto-scaling, Security, Network Configuration까지 포함하는 62개 intent 데이터셋에서도 90% 이상의 정확도를 안정적으로 달성하여, NFV 관

리 작업이 확장된 환경에서도 높은 성능을 확보했음을 확인하였다.

실험 2에서는 제안하는 방법을 여러 종류의 LLM에 적용했을 때의 성능 변화를 분석하였다. Baseline LLM을 사용할 경우 모델 크기에 따라 약 50~70%의 정확도에 그쳤지만, 제안하는 방법을 적용하면 대부분의 모델에서 정확도 향상이 일관되게 관찰되었다. LLaMA3.3과 GPT-4o와 같은 대형 모델은 translated intent 템플릿의 구조와 제약조건을 비교적 정확하게 재구성하여 90% 중후반대의 정확도를 기록했으며, Mistral, Phi4 등 경량 모델도 80% 이상으로 상승해 자원이 제한된 환경에서도 활용 가능성을 보여주었다. 처리 시간은 프롬프트가 복잡해짐에 따라 baseline 대비 증가했으나, Intent Translation부터 정책 생성, 실제 배포까지 포함한 전체 NFV 관리 파이프라인 관점에서 보면 수십에서 수백초가 소요되는 전체 처리시간을 고려할 때 수용 가능한 수준의 오버헤드로 해석할 수 있다.

요약하면 실험 1과 실험 2의 결과는 Intent Translation 단계의 성능을 결정하는 주요 요인은 모델의 크기 등 파라미터 수가 아니라, CoT 프롬프트 설계 기반 단계별 추론과 few-shot learning과 같은 도메인 최적화 기법의 적용을 통해 LLM을 목표에 맞게 최적화하는 것임을 의미한다. 제안하는 방법은 다양한 모델과 관리 작업에 대해 intent를 추상화 및 구조화하여 재구성하며, 기존 규칙 기반의 접근법이나 baseline LLM을 사용하는 방법보다 처리 시간의 증가는 존재하지만 정확도 측면에서 우수하다. 향후에는 intent 데이터셋을 더욱 다양화하여 다양한 인텐트 유형과 자연어 표현에 대한 평가를 확대하고, 본 연구의 Intent Translation 방법을 기반으로 LLM이 실제 실행 가능한 관리 정책을 자동으로 생성하고 생성된 정책을 적용하는 NFV 정책 생성 및 intent 기반 자동화 관리 연구로 확장할 필요가 있다.

V. 결론

네트워크 가상화 기술이 발전함에 따라 NFV 환경의 네트워크 관리 복잡성이 증가하고 있다. 본 논문에서는 사용자의 자연어 관리 의도 (intent)를 기반으로 intent에 포함된 NFV 관리 정책 관련 작업

을 추상화 및 구조화하여 JSON 형식의 템플릿으로 변환하는 LLM 기반 Intent Translation 방법을 제안한다. 제안하는 방법은 CoT, few-shot learning을 사용하여 LLM 기반 파이프라인을 설계 및 구현하고, 기존 연구 및 여러 LLM 모델과의 성능 비교 실험을 통해 제안하는 방법이 기존 방법보다 높은 정확도를 보이고, 특히 구조가 복잡한 관리 작업에서 두드러진 성능 향상을 보이는 것을 확인하였다.

향후 연구에서는 intent 데이터를 다양하게 구성하여 NFV 서비스 유형과 장애 탐지/복구 [31-32], 자원 최적화 [33]와 같은 운영 시나리오까지 확장하여 제안하는 방법을 더욱 평가할 예정이다. 또한, 본 Intent Translation 연구를 확장하여 intent를 기반으로 Kubernetes [34], OpenStack [35]와 같은 실제 NFV 운용 환경에서 실제 실행 가능한 정책을 생성하고 적용하는 Intent 기반 자동화 관리 프레임워크 개발에 대한 연구를 수행할 예정이다.

References

- [1] R. Boutaba et al., "A comprehensive survey on machine learning for networking: evolution applications and research opportunities", in *Journal of Internet Services and Applications*, vol. 9, issue 1, 2018.
- [2] Saha et al. "Intent-based networks: An industrial perspective", *Proceedings of the 1st international workshop on future industrial communication networks*. 2018.
- [3] Leivadeas et al., "A survey on intent-based networking", *IEEE Communications Surveys & Tutorials* vol. 25(1), pp. 625-655, 2022.
- [4] ETSI, "ETSI GS NFV-SOL 007, Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; Network Service Descriptor File Structure Specification", [Online]. Available at https://www.etsi.org/deliver/etsi_gs/NFV-SOL/001_099/007/03.05.01_60/gs_nfv-sol007v030501p.
- [5] Zhao et al., "A survey of large language models", *arXiv preprint, arXiv:2303.18223*, pp. 1-124, 2023.
- [6] L. Giray, "Prompt engineering with ChatGPT: a guide for academic writers", *Annals of biomedical engineering* vol. 51(12), pp. 2629-2633, 2023.
- [7] Wei et al., "Chain-of-thought prompting elicits reasoning in large language models", *Advances in neural information processing systems* vol. 35 pp. 24824-24837, 2022.
- [8] Song et al., "Llm-planner: Few-shot grounded planning for embodied agents with large language models", *Proceedings of the 2023 IEEE/CVF international conference on computer vision (ICCV 2023)*, pp. 1-12, 2023.
- [9] Lin et al., "Data-efficient Fine-tuning for LLM-based Recommendation", *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 1-10, 2024.
- [10] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive nlp tasks", *Advances in neural information processing systems* 33 (NeurIPS 2020), pp. 9459-9474, 2020.
- [11] Stamper et al., "Enhancing llm-based feedback: Insights from intelligent tutoring systems and the learning sciences", *International Conference on Artificial Intelligence in Education*. Cham: Springer Nature Switzerland, pp. 32-43, 2024.
- [12] J. Hong et al., "A Comprehensive Survey on LLM-Based Network Management and Operations", *International Journal of Network Management (IJNM)*, vol. 35, issue 6, e70029, pp. 1-54, 2025.
- [13] Dzevaroska et al., "LLM-based policy generation for intent-based management of applications", *2023 IEEE 19th International Conference on Network and Service Management (CNSM)*, pp. 1-7, 2023.
- [14] Mekrache and Ksentini, "LLM-enabled intent-driven service configuration for next generation networks", *2024 IEEE 10th International Conference on Network Softwarization (NetSoft)*, pp. 253-257, 2024.
- [15] Wang et al., "Network meets ChatGPT: Intent autonomous management, control and operation", *Journal of Communications and Information Networks* vol. 8(3), pp. 239-255, 2023.

[16] Tu et al., "Intent-Based Network Configuration Using Large Language Models", International Journal of Network Management (IJNM), vol. 35, issue 1, 2024.

[17] S. Nam et al., "LLM-based VNF Deployment Automation in OpenStack Environment", 2025 IEEE/IFIP Network Operations and Management Symposium (NOMS 2025), Honolulu, USA, pp. 1-7, May 2025.

[18] IETF, "RFC 9316; Intent Classification", 2022. [Online]. Available at <https://www.rfc-editor.org/info/rfc9316>.

[19] Ollama, "Ollama Document", [Online]. Available at <https://docs.ollama.com/>.

[20] OpenAI, "OpenAI Documents", [Online]. Available at <https://platform.openai.com/docs/overview/>.

[21] Langchain, "Langchain Document", [Online]. Available at <https://docs.langchain.com/>.

[22] Mistral AI, "Mistral Document", [Online]. Available at <https://ollama.com/library/mistral/>.

[23] Microsoft, "Phi4 Document", [Online]. Available at <https://ollama.com/library/phi4-reasoning/>.

[24] OpenAI, "GPT-OSS Document", [Online]. Available at <https://ollama.com/library/gpt-oss/>.

[25] Google, "Gemma3 Document", [Online]. Available at <https://ollama.com/library/gemma3/>.

[26] Alibaba, "Qwen3 Document", [Online]. Available at <https://ollama.com/library/qwen3-coder/>.

[27] DeepSeek, "DeepSeek Document", [Online]. Available at <https://docs.langchain.com/deepseek-r1>.

[28] Meta, "LLaMA3.1 Document", [Online]. Available at https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_1/.

[29] Meta, "LLaMA3.3 Document", [Online]. Available at https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/.

[30] OpenAI, "GPT-4o Document", [Online]. Available at <https://openai.com/research/gpt-4o/>.

[31] 홍지범, 최민지, 유재형, 홍원기, "딥러닝 기반 네트워크 공격 및 침입 탐지 방법 연구", KNOM Review, Vol.25, No. 2, pp.10-21, Dec. 2022.

[32] 정세연, 홍지범, 유재형, 홍원기, "강화학습을 이용한 엣지 컴퓨팅 환경에서의 Live Migration 정책 제어 연구", KNOM Review, Vol. 25, No. 1, pp. 28-36, Aug. 2022.

[33] 이도영, 유재형, 홍원기, "NFV 환경에서의 Deep Q-Networks 기반 오토 스케일링 기술 연구", KNOM Review Vol. 23, No. 2, pp. 1-10, Dec. 2020.

홍 지 범 (Jibum Hong)



2017 한양대학교 ERICA

컴퓨터공학과 학사

2017 ~ 2020 포항공과대학교

컴퓨터공학과 석사

2020 ~ 현재 포항공과대학교

컴퓨터공학과 박사과정

<관심분야> SDN/NFV, 네트워크 모니터링, 네트워크 관리, 인공지능 응용

홍 원 기 (James Won-Ki Hong)



1983 Univ. of Western Ontario, BSc in Computer Science

1985 Univ. of Western Ontario, MS in Computer Science

1985 ~ 1986 Univ. of Western Ontario, Lecturer

1986 ~ 1991 Univ. of Waterloo, PhD in Computer Science

1991 ~ 1992 Univ. of Waterloo, Post-Doc Fellow

1992 ~ 1995 Univ. of Western Ontario, 연구교수

1995 ~ 현재 포항공과대학교, 컴퓨터공학과 교수

2007 ~ 2011 포항공과대학교, 정보통신대학원장

2007 ~ 2010 포항공과대학교, 정보통신연구소 연구소장

2008 ~ 2010 포항공과대학교, 컴퓨터공학과 주임교수

2008 ~ 2012 포항공과대학교, 정보전자융합공학부장

2012 ~ 2014 KT 종합기술원, 원장

<관심분야> 네트워크 트래픽 모니터링, 네트워크 및 시스템 관리, 블록체인, SDN/NFV

외부주의와 잔차연결에 기반한 효과적인 네트워크 침입탐지 모델

신 창 의*, 권 태 웅*, 이 준°

An Effective Network Intrusion Detection Model based on External Attention and Residual Connections

Chang-Yui Shin*, Tae-Woong Kwon*, Jun Lee°

요 약

컴퓨터 네트워크 및 네트워크 기술에 대한 수요가 증가함에 따라 악성공격에 노출된 표면이 자연스럽게 확대되어 악성공격은 점차 증가하고 있습니다. 따라서 네트워크에서 침입탐지 기술은 보안분야로서의 역할이므로 점점 더 중요해지고 있습니다. 하지만 정보보호 측면에서 트래픽의 페이로드가 암호화되면서 이러한 악성 트래픽에 대한 분석이 네트워크 장비에서 제한적이게 됨에 따라 침입탐지는 더욱 어려워졌습니다. 데이터를 기반으로 하는 네트워크 침입탐지에 딥러닝을 적용하면서부터 딥러닝 모델의 계산 복잡성 또한 지속적으로 제기되었습니다. 본 연구에서는 딥러닝 모델 중 이차적 계산 복잡도를 가진 트랜스포머 모델의 셀프어텐션을 대상으로 그 메커니즘적 발전을 이룬 외부주의 구조와 잔차연결을 순차 활용한 효과적인 침입탐지 모델을 설계하여 제안합니다. 제안하는 모델을 최신의 UKM-IDS20 데이터셋을 활용하여 성능을 실험한 결과로써, 오탐지율 0.24%라는 매우 우수한 결과를 보였습니다.

Key Words : Encrypted Network Traffic, Network Intrusion Detection, External Attention, Residual Connections

ABSTRACT

With the increasing demand for computer networks and networking technologies, the attack surface exposed to malicious threats has naturally expanded, leading to a steady rise in such attacks. Consequently, intrusion detection technologies within networks have become increasingly important as a core component of cybersecurity. However, from the perspective of information security, as the payload of traffic becomes encrypted, the analysis of malicious traffic by network devices has become limited, thereby making intrusion detection even more challenging. Since the application of deep learning to data-driven network intrusion detection, the computational complexity of deep learning models has been continuously raised as a concern. In this study, we propose an effective intrusion detection model that sequentially leverages external attention structures and residual connections —mechanistic advancements targeting the self-attention mechanism of transformer models, which inherently possess quadratic computational complexity. Using the latest UKM-IDS20 dataset to evaluate the performance of the proposed model, we achieved an exceptionally strong result, with a false detection rate of only 0.24%.

※ 이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (RS-2023-00236032, 도로교통 공공 서비스·인프라의 암호화 사이버위협에 대한 네트워크 행위기반 보안관제 기술 개발)

♦ First Author : KISTI(Korea Institute of Science and Technology Information), Science and Technology Security R&D Center, superego99@kisti.re.kr

° Corresponding Author : KISTI(Korea Institute of Science and Technology Information), Science and Technology Security R&D Center, jun.lee@kisti.re.kr

* KISTI(Korea Institute of Science and Technology Information), Science and Technology Security R&D Center, taewoong.kwon@kisti.re.kr

논문번호 : KNOM2025-02-005, Received November 6, 2025; Revised November 26, 2025; Accepted December 18, 2025

I. 서 론

오늘날 컴퓨터 네트워크와 기술에 대한 수요가 급증하면서, 새로운 유형의 공격 발생 위험 또한 높아지고 있습니다. 사이버 위협 탐지는 현대 사이버 보안에서 중요한 방어 기술이 되었습니다. 네트워크 기술이 발전하고 공격 전략이 정교해지면서, 매일 발생하는 공격의 빈도가 증가하고 있습니다 [1]. 전통적인 네트워크 침입탐지 시스템(Network Intrusion Detection Systems, IDS)은 새롭고 더욱 정교한 공격 기술에 적응하는 데 어려움을 겪으며, 이를 업데이트하는 것은 계산 비용이 많이 들고 복잡할 수 있습니다. 최근에는 머신러닝을 활용하게 됨에 따라 네트워크 침입탐지 시스템에 의해 지원되는 네트워크 방어 또한 데이터 기반 접근 방식을 점점 더 많이 채택하여 데이터에 대한 의존도가 높아지고 있습니다.

게다가 기술의 발전과 함께, 사용자들은 최신 기술이 탑재된 네트워크 지원 장치를 점점 더 많이 활용하게 되면서 네트워크 트래픽 소비가 크게 증가하고 있습니다. 더 나아가, 엣지 컴퓨팅이 사물인터넷(IoT)에 점점 더 통합됨에 따라, 네트워크 전반에 걸친 데이터 교환과 증가된 연산 요구로 인해 네트워크 트래픽의 양은 빠르게 확장되고 있습니다 [2]. 이렇게 급속히 증가하는 데이터들 중에 악성 트래픽을 신속정확하게 분류하는 것이 네트워크 침입탐지의 핵심이 될 수 있습니다.

그런데 대부분의 네트워크 장비들은 보안을 강화하기 위해 TLS 및 HTTPS와 같은 암호화 통신 프로토콜을 사용합니다. 그러나 이는 네트워크 트래픽의 내용을 직접 분석하거나 분류하는 것을 점점 더 어렵게 만들 수 있습니다. 다양한 대규모의 네트워크 트래픽으로부터 악성트래픽을 탐지하거나 애플리케이션을 정확히 분류하는 태스크는 현대 통신 시스템에서 보안적 측면, 효과적인 자원 관리 측면, 그리고 서비스 품질(QoS) 향상 측면에서 필수적입니다 [2], [3].

한편, 분류학적인 측면에서 네트워크 침입탐지(Network Intrusion Detection, NID)나 네트워크 트래픽 분류(Network Traffic Classification, NTC)는 네트워크 트래픽 분석(Network Traffic Analysis, NTA)과 분석적 메커니즘의 맥락이 유사하다는 점을 염두해두고 바라볼 필요가 있습니다.

앞서 언급했듯이, 암호화 기술의 광범위한 채택으

로 인해 트래픽관련 태스크는 점점 더 어려워지고 있습니다. 페이로드 기반 분석은 더 이상 가능하지 않으며, 포트 번호나 IP 주소에만 의존하는 방식은 애플리케이션을 정확히 식별하는 데 자주 실패하는 경우가 많습니다. 포트 번호, 통계적 특징, 전문가가 정의한 특성에 기반한 기존의 분류 방법은 낮은 정확도를 보이며 실시간 트래픽 분석에 한계를 드러냅니다. 이에 따라 암호화된 트래픽에서도 침입을 탐지하거나 효과적인 애플리케이션 분류를 위해 의미 있는 특징을 추출하려는 새로운 다양한 접근법들이 등장하였습니다 [5], [6], [7], [8]. 특히, 합성곱 신경망(Convolutional Neural Network, CNN), 순환 신경망(Recurrent Neural Network, RNN), 트랜스포머(Transformer) 기반 모델은 원시 바이트 수준에서 특징을 자동으로 추출할 수 있으며, 암호화된 트래픽에서도 의미 있는 패턴을 학습할 수 있는 잠재력을 보여주었습니다.

트랜스포머 기반 모델은 합성곱 신경망 및 순환 신경망 구조에 비해 우수한 성능을 입증했으나, 모델 복잡성과 계산 비용 측면에서 효율성에 대한 우려가 제기되었습니다 [7]. 트랜스포머 기반 트래픽 분류 모델은 주로 자기주의(self-attention) 메커니즘을 활용하는데, 이는 시퀀스 내 종속성을 포착하기 위해 이차적 계산 복잡도를 가지는 것이 특징입니다. 그 결과, 긴 시퀀스를 처리할 때 상당한 계산 및 메모리 오버헤드가 발생하게 됩니다. 이러한 이유로 자기주의 메커니즘을 사용하는 트랜스포머 모델은 실시간 온라인 트래픽 분류에는 적합하지 않습니다. 따라서 제한된 계산 자원을 가진 네트워크 장치에 배포하기에는 더욱 부적절할 수 있습니다.

이러한 문제를 해결하기 위해 선형 시간 상태 공간 모델(linear-time state space model)이 제안되어 더 빠른 추론과 감소된 메모리 소비를 달성하고자 하였습니다. 그러나 해당 모델 [7] 역시 한계가 존재하며, 초당 분류할 수 있는 샘플 수가 종종 열 개를 넘지 못합니다. 같은 맥락에서, 자기주의의 이차적 계산 복잡도를 완화하기 위해 선형 복잡도를 제공하는 외부주의(External Attention) 메커니즘이 도입되었습니다. 이에 따라, 네트워크 트래픽 분류에 외부주의 메커니즘을 적용하는 가능성을 연구가 시도되었습니다. Yahui Hu 등 [8]은 1차원 합성곱 신경망과 외부주의 메커니즘을 계층적으로 통합하였으나, 그들의 구조에서 합성곱 신경망의 역할은 포괄적인 특징을 추출하기에는 충분하지 않았습니다.

본 연구에서는 외부주의 메커니즘 이후에 보다

포괄적인 특징을 포착하는 데 적합한 발전된 구조를 설계하고, 이에 대한 실험 결과를 제시하였습니다. 따라서 본 논문에서는 외부주의 메커니즘에 1차원 레즈넷 구조를 결합한 모델을 제안합니다. 제안하는 구조는 암호화, 익명화, 구조적 불균형 조건 하에서 네트워크 트래픽 분류를 위한 강건하고 일반화된 해결책입니다. 제안하는 구조는 실제 환경에서 흔히 발생하는 잡음이 많고 이질적인 바이트 수준 입력 시퀀스로부터 전역적 의미적 연관성과 세밀한 1차원적 패턴을 효과적으로 포착하도록 설계되었습니다.

제안하는 구조의 아키텍처는 자연어 처리(Natural Language Processing)와 시계열 분석의 강점을 결합하여 바이트 기반 입력을 효율적으로 처리할 수 있도록 합니다. 구체적으로, 외부주의 메커니즘은 전역적 맥락 인식을 담당하며, 1차원 레즈넷은 세부 특징을 추출하고 이를 더욱 표현력 있는 특징 공간으로 재구성합니다. 외부주의 메커니즘은 입력 시퀀스 전반에서 중요한 영역을 선택적으로 강조하고, 1차원 레즈넷은 이러한 영역을 정제하여 효과적으로 재표현합니다.

본 논문의 구성은 다음과 같습니다. 제 II장은 네트워크 침입탐지에 대한 기존 연구를 개관하고, 본 모델 개발의 동기를 논의합니다. 제 III장은 제안된 프레임워크의 아키텍처 구성 요소와 구현을 상세히 설명합니다. 제 IV장은 포괄적인 실험 설정과 결과를 제시하며, 제안하는 모델의 성능을 기존 다른 방법들과 비교합니다. 마지막으로 제 V장은 본 논문의 결론과 향후 연구방향을 제시합니다.

II. 관련연구

2.1. 네트워크 트래픽 분류 목적 딥러닝 모델

딥러닝은 이미지 처리, 자연어 이해, 시계열 분석 등 다양한 분야에서 탁월한 성능을 입증해왔습니다. 이러한 기술적 발전은 네트워크 트래픽 분석 분야에도 적극적으로 적용되고 있습니다. 초기 연구에서는 2차원 합성곱 신경망을 활용하여 패킷 데이터를 이미지와 유사한 형식으로 변환하여 구조적 특징을 추출하고, 이어서 장단기 메모리(Long Short-Term Memory, LSTM) 네트워크를 통해 흐름 수준의 시계열 종속성을 포착하였습니다. 그 결과로 제안된 2차원 합성곱 신경망-장단기 메모리 아키텍처 [9]는 공간적 및 시계열적 특징을 통합하는 것을 목표로

하였으나, 단일 패킷 데이터를 1차원적 관점에서 다루지는 못하였습니다.

한편, 시계열 데이터 처리에서 매우 효과적인 것으로 입증된 트랜스포머 모델 [10]의 자기주의 메커니즘은 최근 자연어 처리 분야에서 암호화된 네트워크 트래픽 분석 작업에 채택되었습니다 [11]. 그리고 BERT 아키텍처를 기반으로 한 ET-BERT [12] 모델은 패킷 및 흐름 수준에서 분류 성능을 향상시키기 위해 제안되기도 하였습니다. Masked BURST 및 Same-origin BURST Prediction 기법을 도입함으로써, ET-BERT는 ISCXVPN2016 데이터셋 [13]에서 각각 98.90%와 99.37%의 F1-score를 달성하였다. 그러나 강력한 성능에도 불구하고, 이 모델은 암호화된 트래픽의 바이트 수준 특징을 포착하지 못하며, 대규모 사전학습에 의존하기 때문에 막대한 계산 및 자원 비용을 초래하는 실질적 한계를 가지는 단점이 있습니다.

트랜스포머를 활용한 네트워크 트래픽 분석 분야의 또 다른 연구[14]에서는 DistilBERT 기반의 경량 모델이 제안되었으며, 최소 63바이트의 입력 데이터만으로도 높은 성능을 달성하였다. 이 연구에서는 다중 속성 패킷 간의 중복 정보를 제거하고 표현 능력을 강화하기 위해 비교학습(Comparative Learning)을 적용하였다. 그 결과, 해당 모델은 실시간 처리 및 모델 압축 측면에서 ET-BERT보다 더 실용적인 대안을 제공하였습니다. 해당 XENTC 모델은 불필요한 특징을 제거함으로써 네 개의 다중 속성 패킷을 동시에 효과적으로 처리하며, 97.0%에서 98.1% 범위의 F1-score를 달성하였습니다. 학습된 모델을 파인튜닝하여 단순화된 63바이트 표현된 패킷을 데이터로 입력받아 분류하는 데 소요된 시간은 패킷당 약 0.0093초였습니다.

네트워크 트래픽 분석에 특화된 딥러닝 모델들은 성능 향상을 위해 서로 다른 설계 철학에 기반한 독창적인 아키텍처와 데이터 처리 전략을 보여줍니다. Self-Attentive Method(SAM) [15]는 패킷 내 요소들 간의 관계를 학습하기 위해 자기주의 메커니즘을 활용하고, 그 출력을 1차원 합성곱 신경망과 통합했습니다. 저자들은 50 바이트 패킷 입력을 SAM 모델에 적용하여 94.48%의 F1-score를 달성하였습니다. 게다가 해당 연구에서는 패킷당 추론 시간이 2밀리초 미만임을 보고하였으며, 이는 온라인 환경에서 실시간 분류에 적합함을 나타내었습니다. 게다가 각 피처의 중요도를 주의(attention) 관계를 통해 분석한 결과도 제시하였습니다.

2.2 외부주의 모델

자기주의 메커니즘은 트랜스포머 모델의 근본적인 구조로써, 단일 샘플 내에서 장기 종속성을 포착할 수 있는 장점을 제공합니다. 이에 따라 자기주의 메커니즘은 컴퓨터 비전과 자연어 처리 등 다양한 분야에서 널리 채택되어 왔습니다. 해당 메커니즘은 샘플 내 모든 위치 간의 특징 관계를 집계함으로써 각 위치의 표현을 정교화합니다. 그러나 이 과정은 입력 위치의 수에 대해 이차적 계산 복잡도를 초래하기도 합니다. 또한 자기주의 메커니즘은 본질적으로 샘플 내부 관계에 집중하기 때문에, 서로 다른 샘플 간의 잠재적 상관관계는 간과되기도 합니다. 즉, 단일 샘플 내 관계적 표현을 강화하는 데에는 효과적이지만, 샘플 간 관계 모델링에는 기여하지 못합니다.

앞서의 자기주의 메커니즘의 계산 오버헤드와 샘플 간 상관관계 모델링의 부재를 해결하기 위해 외부주의 메커니즘 [16]이 제안되었습니다. 성능비교를 통한 우수성을 제시했습니다. 자기주의 메커니즘은 쿼리(query)와 키(key) 벡터 간의 유사도를 측정하여 어텐션 맵을 계산한 뒤, 이를 값(value) 벡터의 가중 집계와 결합하여 새로운 특징 맵을 생성합니다. 반면, 외부주의 메커니즘은 쿼리 벡터와 외부 학습 가능한 키 메모리 간의 유사도를 계산하고, 그 결과 어텐션 맵을 학습 가능한 값을 가지는 외부 메모리에 곱하여 새로운 특징을 생성합니다. 이 두 외부 메모리는 선형 계층을 통해 구현되며 전체 데이터셋에 걸쳐 공유됨으로써 외부주의 메커니즘의 일반화 능력을 강화합니다. 그 결과, 외부주의 메커니즘은 여러 샘플을 효과적으로 구별할 수 있는 고도로 판별적인 특징을 학습합니다. 또한 입력 크기에 대해 선형적인 계산 복잡도를 가지므로, 실행 시간 측면에서 향상된 효율성을 제공합니다.

Yahui Hu 등 [8]은 네트워크 트래픽 분석 분야에서 패킷 수준의 중요한 바이트 관계를 포착하기 위해 외부주의 메커니즘과 1차원 합성곱 신경망을 통합한 연구를 제시하였습니다. 그들의 실험에서는 패킷 헤더로부터 12바이트 특징을 추출하여 여러 데이터셋에서 96% 이상의 분류 정확도를 달성하였으며, 패킷당 평균 약 0.36밀리초의 추론 시간을 기록하였습니다. 그러나 해당 모델은 외부주의 메커니즘 출력으로부터 파생된 패킷 내 바이트 단위 종속성을 효율적으로 처리하는 데 한계가 있었습니다.

이러한 한계를 해결하기 위해, 본 연구에서는 단

일 샘플 내 중요한 바이트 수준 상호작용을 모델링할 때, 1차원 합성곱 신경망으로 인해 발생하는 관계 정보 손실을 고려하였습니다. 이에 따라, 표준화된 표 형식 입력에 적합하도록 2차원 레즈넷 아키텍처를 최적화하여 1차원 입력에 맞게 재설계해 잔차연결을 1차원 합성곱 신경망에 반영했습니다. 이러한 아키텍처적 개선이 패킷 수준에서 관계적 특성을 포착하는 데 미치는 영향은 제4장에서 제시되는 실험 결과와 분석을 통해 검증하였습니다.

III. 데이터 및 모델 구성

3.1. 데이터셋 선정

UKM-IDS20 [17]은 실제 네트워크 환경 (Universiti Kebangsaan Malaysia)에서 수집된 데이터를 기반으로 생성된 새로운 벤치마킹 데이터셋입니다. UKM-IDS20 데이터셋이 생성된 환경은 다음과 같습니다.

표 1. UKM-IDS20내 정상/공격 데이터 분포

Table 1. Distribution of Normal and Attack Data in the UKM-IDS20 Dataset

Connection Record Type			Training set / Testing set
Sum			10,308 / 2,579
Normal		-	7,140 / 1,769
Abnormal	ARP poisoning	-	476 / 116
	DoS	Mass HTTP requests	461 / 140
		TCP flood	467 / 121
		UDP data flood	445 / 108
	Scans	-	474 / 123
	Exploits	BeEF HTTP exploits	404 / 96
		Metasploits exploits	441 / 106

Hyper-V를 이용한 가상화 환경에서 Office A(서버, IDS, 스위치, 4개 호스트)와 Office B(호스트, 스위치, 웹 서버)로 구성된 네트워크를 모의했습니다. TShark, Kali Linux, Metasploit, Nmap 등의 도구를 사용하여 정상 트래픽과 악성 공격 트래픽을 생성하고 캡처되었습니다.

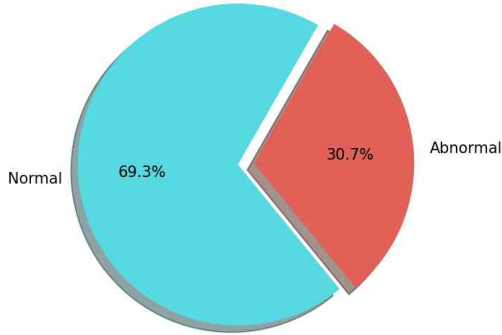


그림 1. 이진분류 데이터 분포
Fig. 1. Binary Class Data Distribution

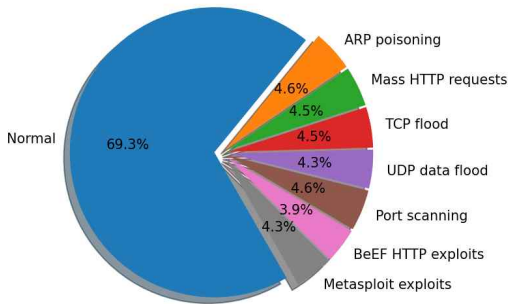


그림 2. 멀티클래스 데이터 분포
Fig. 2. Multi-class Data Distribution

데이터셋의 특징 갯수와 공격유형을 살펴보면, 총 46개의 특징을 포함하며, ARP poisoning, DoS (HTTP 요청, TCP/UDP 플러드), Scans (포트 스캔), Exploits (Metasploit MS17-010, BeEF)의 네 가지 주요 공격 유형과 정상(Normal) 트래픽을 포함합니다. 46개의 특징이 전처리되는 과정은 데이터셋 공개논문[17]에 세부적으로 설명되어 있습니다. 데이터셋의 구성측면에서 총 12,887개의 연결 레코드로 구성되며, 훈련 세트(10,308개)와 테스트 세트(2,579개)로 나뉩니다. 표 1은 데이터셋의 클래스별

세부 현황을 나타냅니다. 정상 레코드 수가 공격 레코드보다 많도록 구성하여 기계 학습 모델의 분류 성능 저하 문제를 완화하였습니다.

한편, Rough-set 이론을 이용한 특징 의존도 (Dependency Ratio, DR) 분석 결과, UKM-IDS20 데이터셋의 특징들이 KDD99 및 UNSW-NB15 데이터셋보다 높은 관련성을 가지는 것이 제시되기도 하였습니다. 특히 기본 연결정보 외에 시간 기반 특징과 ARP 관련 특징이 공격 탐지에 중요한 역할을 하는 것으로 분석되었습니다. 이에 따라 이러한 특징들이 피처에 반영되었다는 점이 해당 데이터셋이 가지는 특징점입니다.

UKM-IDS20 데이터셋은 12,887개의 데이터로 구성되어 있습니다. 다음 그림 1, 2는 그에 대한 수량적 비율 분포도를 파이차트로 나타낸 것인데, 그림 1은 정상 트래픽과 공격트래픽을 기준으로 2진 분류한 것이며, 그림 2는 공격트래픽을 각 7개의 클래스로 세분하고 정상트래픽을 1개 클래스로 구분하여 총 8개 클래스로 분류한 것입니다. 이러한 데이터의 클래스 라벨을 원한 인코딩하고, 숫자 데이터를 일정 구간으로 나눈 뒤, 결측치를 처리해서 PyTorch 텐서 데이터셋으로 변환하는 과정을 거쳐 모델에 입력되도록 했습니다. 이러한 입력 시퀀스에 대해 포지션 임베딩과 바이트 임베딩을 적용하여 각 위치와 바이트 단위 토큰을 128차원 벡터로 매핑하였습니다.

3.2. 모델 설계

제안하는 모델은 입력 데이터를 순차적 아키텍처를 통해 처리하며, 전처리와 임베딩 과정을 거친 후 외부주의 메커니즘과 1차원 레즈넷 모듈을 적용했습니다. 그림 3은 제안하는 아키텍처적 계층 구조를 나타냅니다. 외부주의 메커니즘과 1차원 레즈넷 모두 입력 시퀀스를 처리하도록 설계되었지만, 그 기능적 역할과 초점은 근본적으로 상이합니다. 설계 관점에서 볼 때, 두 모듈은 모델 전체 아키텍처 내에서 상호 보완적인 작업을 수행하도록 순차적으로 설계되었습니다.

외부주의 메커니즘은 전체 입력 시퀀스에서 전역적으로 중요한 영역을 식별하고 선택하는 역할을 주로 담당합니다. 이는 각 요소의 상대적 중요성을 평가하는 메커니즘으로 작동하며, 일반적으로 소프트맥스(soft-max) 기반 가중치 부여 방식을 통해 구현됩니다. 이를 통해 모델은 데이터 내에서 가장 정보의 의미성이 높은 부분에 계산 자원을 집중할 수

있으며, 이러한 선택적 강조는 모델이 고수준 의미적 연관성을 포착하는 능력이 향상되도록 합니다.

반면, 1차원 레즈넷은 외부주의 메커니즘에 의해 강조된 영역으로부터 세밀한 특징을 추출하고 재표현하는 역할을 합니다. 잔차 연결(residual connections) [18], [19], [20]과 1차원 합성곱 연산을 활용하여 선택된 정보를 정제하고, 이를 보다 표현력 있고 판별적인 특징 공간으로 변환합니다. 본질적으로, 외부주의 메커니즘이 어디에 집중할지를 결정한다면, 1차원 레즈넷은 무엇을 추출하고 어떻게 더 정밀하게 표현할지를 결정합니다.

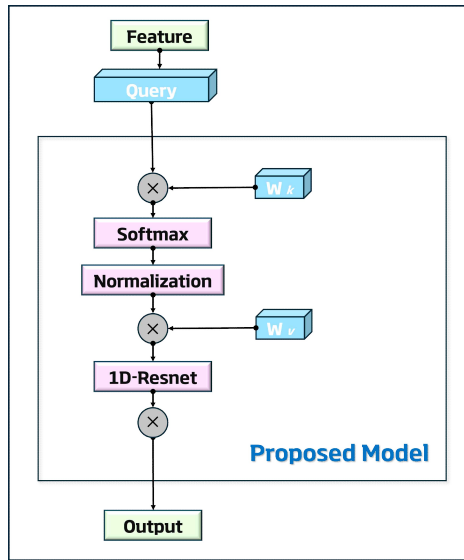


그림 3. 제안하는 모델

Fig. 3. Proposed Model

3.2.1 외부주의 레이어

수식 (1)를 순차적으로 설명하면, $K = QW_k$ 로 쿼리 행렬 Q 와 학습 가능한 가중치 W_k 를 곱해 K 를 얻습니다. 이어서 $A = \text{Softmax}(K/\tau)$ 로 스케일링된 소프트맥스를 적용합니다. 마지막으로 각 행을 정규화하여 $\hat{A} = \frac{A}{\sum_{j=1}^S A_{ij}}$ 를 얻습니다. 즉 쿼리행렬을 선형적으로 변환하고, 소프트맥스를 적용해 각 행단위 정규화 과정을 모두 포함한 하나의 식입니다.

$$\hat{A}_i = \frac{\text{Softmax}(\frac{QW_k}{\tau})_i}{\sum_{j=1}^S \text{Softmax}(\frac{QW_k}{\tau})_{ij}} \quad (1)$$

외부주의 메커니즘은 본 신경망 아키텍처에서 핵심적인 역할을 수행하며, 입력 데이터 내 전역적으로 중요한 위치를 식별하고 선택적으로 강조할 수 있도록 모델을 지원합니다. 이러한 메커니즘은 분류, 생성, 이해와 같은 다운스트림 작업에 필수적인 두드러진 표현을 추출하는 데 기여합니다. 기존의 자기주의 메커니즘이 내부 토큰 간 상호작용에만 의존하는 것과 달리, 외부주의 메커니즘은 추가적인 키키펙 공간을 도입하여 보다 유연하고 문맥 인식적인 주의 점수(attention score) 계산을 가능하게 합니다.

외부주의 메커니즘의 내부 동작 논리는 다음과 같이 설명될 수 있습니다. 먼저, 입력 시퀀스는 사전에 정의된 어텐션 키 공간으로 투영되며, 이는 각 입력 요소의 상대적 중요성을 평가하기 위한 참조 프레임워크로써 기능을 합니다. 이러한 투영은 일반적으로 학습 가능한 선형 변환을 통해 수행되며, 입력 특징을 일련의 어텐션 키로 매핑됩니다. 이후 이 키들은 소프트맥스(soft-max) 함수를 통해 처리되어 값이 정규화되고, 입력 시퀀스 전반에 걸친 중요도 점수 분포가 산출됩니다. 이러한 확률적 가중 메커니즘은 모델이 가장 정보성이 높은 구성 요소에 집중하고, 덜 관련된 요소는 억제할 수 있도록 합니다.

마지막 단계에서는 어텐션으로 강화된 표현을 재구성합니다. 전역적 문맥 강화(global contextual enhancement) 이후, 가중된 입력 특징은 단순하더라도 효율적인 투영 계층을 통해 다시 초기 특징 공간으로 선형 매핑됩니다. 이러한 설계는 입력 데이터의 구조적 무결성을 유지하는 동시에 전역적으로 인식된 의미 정보를 풍부하게 하여, 모델이 보다 정교한 이해와 의사결정을 수행할 수 있는 능력을 강화합니다.

3.2.2 1차원 레즈넷 레이어

수식 (2)는 잔차학습 구조(Residual Learning Framework)를 일반화하여 (1)에서 얻은 결과를 처리하는 과정입니다. 따라서 수식 (2)를 위에서와 마찬가지로 설명하면, $T(Data)$ 는 입력텐서($Data$)의 축을 교환하는 전치연산을 의미합니다. $F_{Conv1D_{max}(2)}$ 는 동일 패딩을 갖는 1차원 합성곱 연산을 2회 적용하는 변환 연산을 적용하는 것을 의미합니다. 이어서 배치 정규화 연산자인 N_{BN} 을 적용하고, σ 는 비선형 활성화 함수인 ReLU 연산이 적용됨을 의미합니다. 마지막으로 이러한 과정에서 얻은 비선형 변환 경로에 $S(T(EA_{out}))$ 이라는 잔차경로를 더함으로써 최종 출력값을 정의하는 형태입니다.

$$y = F_{Conv1D_{same(2)}}(\sigma(N_{BN}(F_{Conv1D_{same(2)}}(S(T(EA_{out})))))) + S(T(EA_{out})) \quad (2)$$

본 모델은 다양한 길이의 입력 시퀀스를 유연하게 수용할 수 있도록 설계된 점이 중요합니다. 서로 다른 입력 크기에서도 구조적 일관성을 보장하기 위해 모든 합성곱 연산은 same padding [18]을 사용하며, 이를 통해 입력과 출력 간의 시퀀스 길이가 유지됩니다. 이러한 설계 선택은 잔차 학습을 안정화시키고, 수동적인 크기 조정(adjustment)이나 절단(truncation)의 필요성을 제거합니다. 그리고 어텐션 가중 입력(attention-weighted input)을 수신함으로써, 1차원 레즈넷 모듈은 목표 지향적 특징 정제를 수행하고, 국소적으로 두드러진 정보를 보다 판별력 있고 표현력이 풍부한 특징 공간으로 변환됩니다. 이는 모델의 전체적인 표현 능력을 향상시키며, 실제 데이터에 내재된 변동성에 대해 강건하게 적응할 수 있도록 합니다.

IV. 성능실험 및 분석

4.1. 실험 환경 및 모델 주요 파라미터

모델은 CUDA 12.9 기반의 Python 3.10.18 및 Pytorch 2.7.1를 기반으로 구현되었습니다. 하드웨어는 20Core Intel(TM) Ultra 7 265K CPU @ 3.90GHz의 CPU와 NVIDIA GeForce RTX(12GB 메모리)의 GPU가 장착된 서버로 구성되었습니다.

앞서 기술된 바와 같이 외부주의모델은 128차원의 벡터를 입력으로 받음에 따라 1차원 레즈넷 모델도 동일한 벡터를 받도록 처리하도록 했습니다. 필터와 커널 사이즈는 각각 256, 16으로 설정하였습니다.

4.2. 실험결과 및 분석

모델을 평가하기 위한 요소로 사용된 지표들은 아래 수식 (3) ~ (8)과 같습니다.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

$$F1-score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

$$Detection Rate(DR) = \frac{TP}{TP + FN} \quad (7)$$

$$False Alarm Rate(FAR) = \frac{FP}{FP + TN} \quad (8)$$

위에서 True Positive(TP)는 실제 True인 정답을 True라고 예측(정답)하는 것이고, False Positive(FP)는 실제 False인 정답을 True라고 예측(오답)하는 것입니다. False Negative(FN)는 실제 True인 정답을 False라고 예측(오답)하는 것이고, True Negative(TN)는 실제 False인 정답을 False라고 예측(정답)하는 경우입니다. 수식(7)은 네트워크 침입 탐지 모델의 성능을 측정할 때 주로 쓰이는 Detection Rate(DR)이며, 분류모델의 성능을 측정할 때 사용되는 수식 (5)의 Recall과 같은 값을 나타냅니다.

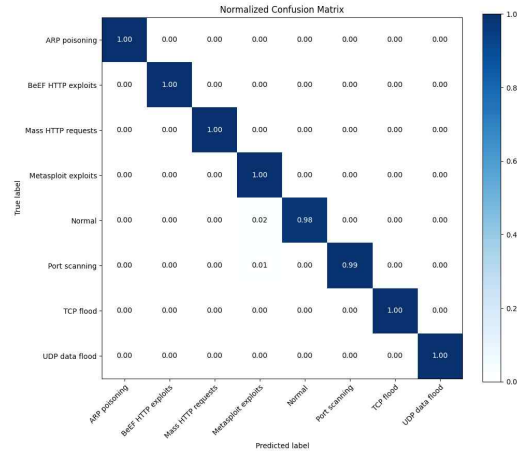


그림 4. 컨퓨전 매트릭스

Fig. 4. Confusion Matrix

그림 4는 제안하는 모델의 학습과정 이후에 예측 성능을 나타낸 컨퓨전 매트릭스입니다. 그림 4를 통해서 모델의 성능을 살펴보자면, 컨퓨전 매트릭스의 좌측 대각선 값은 모델의 정답을 맞춘 결과를 나타냅니다. 따라서 대각선 값들이 0.98~1.00 사이의 값을 가지므로 라벨별로 정답률이 매우 높은 수준임을 확인할 수 있습니다. 그리고 분류결과에 대해 모

델이 예측한 것 중 잘못된 것을 기준으로 모델이 잘못 판단할 수 있는 부분으로, Normal 트래픽이나 Port scanning 트래픽을 일부 Metasploits으로 잘못된 예측을 할 수 있음을 나타냅니다.

표 2. 다른 연구들과의 성능비교

Table 2. Performance comparison with other research

	DANN	HOE-DANN	Proposed
Detection Rate (=Recall)	95.18%	96.91%	98.23%
False Alarm Rate	7.97%	7.57%	0.24%
Accuracy	93.60%	94.66%	99.30%
Precision	-	-	99.51%
F1-score	-	-	98.83%

표 2에서는 제안하는 모델과 다른 모델들이 동일한 UKM-IDS20 데이터셋을 대상으로 성능을 비교한 결과를 확인 할 수 있습니다. 세부적으로 살펴보면 탐지율(Detection Rate)이 98.23%를 보임으로써 제안하는 모델이 HOE-DANN보다 1.32% 높은 성능을 가집니다. 그리고 오경보율(False Alarm Rate)은 0.24%를 보임으로써 HOE-DANN보다 7.33% 낮아, 허위경보비율이 매우 낮아 뛰어난 탐지 성능을 가지고 있는 것을 의미합니다.

표 3. 제안 모델의 총 파라미터 갯수

Table 3. Total Number of Parameters in Our Proposed Model

	Number of parameters	Magnitude (Size)
BERT	109.0 M	440.0 MB
DistilBERT	66.0 M	267.0 MB
ALBERT	11.0 M	49.0 MB
Proposed model	2.7 M	10.9 MB

표 3에서는 모델의 총 파라미터 개수와 모델의 크기를 나타내는데, 학습가능한 파라미터의 수가 3백만개 이하이고 모델의 사이즈도 10MB정도입니다. 따라서 다른 딥러닝 모델보다 매우 간결한 정도이면서도, 앞서의 우수한 성능을 보여주는데서 효과적인 네트워크 침입탐지 모델임을 알 수 있습니다. 46개의 피쳐로된 입력을 처리하는데 걸리는 시간은 초당 4,638개로 매우 빠른 처리 성능을 보여줍니다.

V. 결론 및 향후 연구방향

5.1. 결론

본 논문에서는 효과적인 외부주의 메커니즘과 1차원 레즈넷 모델을 연계하는 레이어 구조를 설계하여 효과적인 침입탐지모델을 설계한 결과를 보여주었습니다. 제안된 모델에서 외부주의 메커니즘은 전체적인 전역적 맥락 인식을 담당하며, 1차원 레즈넷 레이어는 세부 특징을 추출하고 이를 더욱 표현력 있는 특징 공간으로 재구성하는 역할을 담당하도록 설계되었습니다.

제안하는 모델의 성능을 실험하는데 사용한 UKM-IDS20 데이터셋은 기존의 다른 데이터셋 보다 최신 공격 양상을 반영하여 더 높은 특징 복잡성을 가진 데이터셋이기에 본 연구의 성능실험에 의미있는 기여를 한 것으로 판단할 수 있습니다. 제안하는 모델의 구성적 특징을 바탕으로 UKM-IDS20 데이터셋을 대상으로 실험한 결과 오경보율이 0.24% 정도의 매우 낮은 수치를 보여 매우 우수한 IDS 탐지 모델임을 드러냈습니다.

5.2. 향후 연구방향

향후 연구 진전을 통해 발전시키고 싶은 방향은 다음과 같습니다. 첫째, 앞서 제시한 UKM-IDS20 데이터셋은 수집되는 환경에서 정상과 공격 트래픽의 비율을 불균형한 특성을 자연스럽게 반영합니다. 이러한 불균형한 데이터셋에 대한 여러 접근 방법에 대해 살펴보고, 모델의 성능을 보다 향상시킬 수 있는 방법론에 대한 새로운 접근을 시도하려고 합니다. 둘째, 보다 확장된 메커니즘 연구와 함께 다양한 데이터와 모델을 다루어, 추가 실험결과를 통해 성능 일반화 검증을 하려고 합니다. 셋째, 클래스간 오탐 발생원인을 분석하는 것에 XAI를 적용하여 인지가능한 해석을 통해 모델의 실용적 측면에 대한 접근으로 보다 확장하고자 합니다.

References

- [1] E. Kabir, J. Hu, H. Wang, and G. Zhuo, "A novel statistical technique for intrusion detection systems," *Future Generation Computer Systems*, vol. 79, pp. 303–318, Feb. 2018, doi: 10.1016/j.future.2017.01.029.
- [2] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017, doi: 10.1109/IIOT.2017.2683200.
- [3] R. H. Serag, M. S. Abdalzaher, H. A. E. A. Elsayed, and M. Sobh, "Software Defined Network Traffic Classification for QoS Optimization Using Machine Learning," *J. Netw. Syst. Manag.*, vol. 33, no. 2, p. 41, Apr. 2025, doi: 10.1007/s10922-025-09911-6.
- [4] J. H. Kalwar and S. Bhatti, "Deep Learning Approaches for Network Traffic Classification in the Internet of Things (IoT): A Survey," Feb. 01, 2024, arXiv: arXiv:2402.00920. doi: 10.48550/arXiv.2402.00920.
- [5] P. Zhu, G. Wang, J. He, Y. Dong, and Y. Chang, "An encrypted traffic identification method based on multi-scale feature fusion," *Array*, vol. 21, p. 100338, Mar. 2024, doi: 10.1016/j.array.2024.100338.
- [6] S. Xu, J. Han, J. Wang, and Y. Bai, "An encrypted traffic classification method based on autoencoders and convolutional neural networks," *PLOS One*, vol. 20, no. 9, p. e0333276, Sept. 2025, doi: 10.1371/journal.pone.0333276.
- [7] T. Wang, X. Xie, W. Wang, C. Wang, Y. Zhao, and Y. Cui, "NetMamba: Efficient Network Traffic Classification via Pre-training Unidirectional Mamba," Oct. 20, 2024, arXiv: arXiv:2405.11449. doi: 10.48550/arXiv.2405.11449.
- [8] J. Yu, Y. Choi, K. Koo, and D. Moon, "A novel approach for application classification with encrypted traffic using BERT and packet headers," *Comput. Netw.*, vol. 254, p. 110747, Dec. 2024, doi: 10.1016/j.comnet.2024.110747.
- [9] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for Internet of Things," *IEEE Access*, vol. 5, pp. 18042–18050, 2017, doi: 10.1109/ACCESS.2017.2747560.
- [10] A. Vaswani et al., "Attention Is All You Need," Aug. 02, 2023, arXiv: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.
- [11] H. Y. He, Z. Guo Yang, and X. N. Chen, "PERT: Payload Encoding Representation from Transformer for Encrypted Traffic Classification," in *2020 ITU Kaleidoscope: Industry-Driven Digital Transformation (ITU K)*, Feb. 2020, pp. 1–8. doi: 10.23919/ITUK50268.2020.9303204.
- [12] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, and J. Yu, "ET-BERT: A Contextualized Datagram Representation with Pre-training Transformers for Encrypted Traffic Classification," in *Proceedings of the ACM Web Conference 2022*, Apr. 2022, pp. 633–642. doi: 10.1145/3485447.3512217.
- [13] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Encrypted and VPN Traffic using Time-related Features," in *Proceedings of the 2nd International Conference on Information Systems Security and Privacy*, Rome, Italy: SCITEPRESS - Science and Technology Publications, 2016, pp. 407–414. doi: 10.5220/0005740704070414.
- [14] C.-Y. Shin, J.-T. Park, U.-J. Baek, and M.-S. Kim, "A Feasible and Explainable Network Traffic Classifier Utilizing DistilBERT," *IEEE Access*, vol. 11, pp. 70216–70237, 2023, doi: 10.1109/ACCESS.2023.3293105.
- [15] G. Xie, Q. Li, and Y. Jiang, "Self-attentive deep learning method for online traffic classification and its interpretability," *Comput. Netw.*, vol. 196, p. 108267, Sept. 2021, doi: 10.1016/j.comnet.2021.108267.
- [16] M.-H. Guo, Z.-N. Liu, T.-J. Mu, and S.-M. Hu, "Beyond Self-Attention: External Attention

Using Two Linear Layers for Visual Tasks,”
IEEE Trans. Pattern Anal. Mach. Intell., vol. 45,
no. 5, pp. 5436–5447, May 2023, doi:
10.1109/TPAMI.2022.3211006.

- [17] M. S. Al-Daweri, S. Abdullah, and K. A. Zainol Ariffin, “An adaptive method and a new dataset, UKM-IDS20, for the network intrusion detection system,” Computer Communications, vol. 180, pp. 57–76, Dec. 2021, doi: 10.1016/j.comcom.2021.09.007.

- [18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Dec. 10, 2015, arXiv: arXiv:1512.03385. doi: 10.48550/arXiv.1512.03385.

- [19] P. Tchatchoua, G. Graton, M. Ouladsine, J. Muller, A. Traore, and M. Juge, “1D ResNet for Fault Detection and Classification on Sensor Data in Semiconductor Manufacturing,” in 2022 International Conference on Control, Automation and Diagnosis (ICCAD), Lisbon, Portugal: IEEE, July 2022, pp. 1–6. doi: 10.1109/ICCAD55197.2022.9853997.

- [20] C. Szegedy et al., “Going Deeper with Convolutions,” Sept. 17, 2014, arXiv: arXiv:1409.4842. doi: 10.48550/arXiv.1409.4842.

신 창 의 (Chang-Yui Shin)



2003년 2월 : 육군사관학교 운영분석학과 졸업

2007년 2월 : 고려대학교 전자컴퓨터공학과 석사

2025년 2월 : 고려대학교 컴퓨터정보학과 박사

2025년 7월~현재 : 한국과학기술정보연구원 과학기술보안

연구센터 박사후연구원

<관심분야> 네트워크관리, 네트워크 트래픽 분석, Ad-hoc 라우팅, 전술데이터링크, AI, 정보보호

권 태 웅 (Taewoong Kwon)



2012년 2월 : 숭실대학교 컴퓨터학부 졸업

2014년 8월 : 고려대학교 정보보호대학원 정보보호학과 석사

2014년 12월~현재 : 한국과학기술정보연구원 과학기술보안 연구센터 선임연구원

<관심분야> 정보보호, 보안관계, 네트워크 보안, 네트워크 가시화

이 준 (Jun Lee)



2017년 8월 : 한국항공대학교 정보공학 박사

2016년 10월~2017년 11월 : 일본산업기술종합연구소(AIST) 인공지능연구센터(AIRC) Research Assistant

2017년 12월~2019년 12월 : 일본산업기술종합연구소(AIST) 인공지능연구센터(AIRC) 박사후연구원

2020년 1월~2024년 12월 : 한국과학기술정보연구원 과학기술보안연구센터 선임연구원

2025년 1월~현재 : 한국과학기술정보연구원 과학기술보안연구센터 책임연구원

<관심분야> 인공지능, 보안관계, 지식추출, 자연언어처리, 네트워크 보안

볼류메트릭 영상 스트리밍 환경에서 단말 처리 성능이 ABR 알고리즘의 동작에 미치는 영향 분석

유 상 우*, 홍 원 기^o

Analysis of the Impact of Client Computing Capability on ABR Algorithm Behaviors in Volumetric Video Streaming

Sangwoo Ryu*, James Won-Ki Hong^o

요 약

최근 몰입형 미디어 기술의 발전으로 인해 대용량의 볼류메트릭 데이터를 원활하게 제공할 수 있는 효율적인 전송 기술이 중요해지고 있다. 기존 비디오 스트리밍 시스템은 네트워크 변동성에 대응하기 위해 대역폭 또는 버퍼 기반의 적응형 비트레이트 알고리즘을 사용했다. 그러나 볼류메트릭 영상은 디코딩 및 렌더링 과정에서 많은 연산을 요구하며, 네트워크 대역폭이 충분한 상황에서도 단말의 컴퓨팅 성능 한계로 스트리밍 품질이 저하될 수 있다. 본 연구에서는 이러한 단말의 처리 성능이 볼류메트릭 영상 스트리밍 환경에서 ABR 알고리즘의 동작 특성 및 스트리밍 품질에 미치는 영향을 실험적으로 분석한다. 이를 위해 DASH 기반의 볼류메트릭 영상 스트리밍 시스템을 구축하고, 서로 다른 성능을 가진 단말 환경과 정적, 동적인 네트워크 시나리오에서 주요 ABR 알고리즘들의 성능을 평가 및 분석하였다. 이를 통해 볼류메트릭 영상 스트리밍에서 네트워크 지표 뿐만 아니라 단말의 성능, 볼류메트릭 영상 스트리밍의 특성들을 종합적으로 고려하는 적응형 스트리밍의 필요성을 제시하였다.

Key Words : Volumetric Video Streaming, Adaptive Bitrate Streaming, Quality of Experience

ABSTRACT

With the recent advancements in immersive media technology, efficient transmission technology that can smoothly provide large amounts of volumetric data is becoming increasingly important. Conventional video streaming systems have employed bandwidth- or buffer-based adaptive bitrate algorithms to cope with network fluctuations. However, in volumetric video streaming, which involves compute-intensive tasks such as decoding and rendering, streaming quality may degrade due to limited computing capability, even with sufficient network bandwidth. This study analyzes the impact of client computing capability on the Adaptation behavior of ABR algorithms and streaming quality in a volumetric video streaming system. We implemented a DASH-based volumetric video streaming system to evaluate ABR algorithms across diverse device and network conditions. We highlight the necessity for adaptive streaming strategies that consider client performance and volumetric video characteristics along with network metrics.

※ 이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-대학ICT연구센터(ITRC)의 지원(IITP-2025-RS-2024-00437866, 50%)과 삼성전자의 지원(50%)을 받아 수행된 연구임

• First Author : Pohang University of Science and Technology (POSTECH), Graduate School of Artificial Intelligence, rswoo@postech.ac.kr

^o Corresponding Author : POSTECH, Department of Computer Science and Engineering, jwkhong@postech.ac.kr

논문번호 : KNOM2025-02-003, Received November 3, 2025; Revised November 27, 2025; Accepted December 20, 2025

I. 서 론

지속적인 통신 네트워크 기술의 고도화와 함께, 사용자에게 시청 이상의 경험을 제공하는 차세대 실감형 미디어 서비스가 빠르게 확산되고 있다. 특히 볼류메트릭 영상(Volumetric Video)은 포인트 클라우드(Point Cloud), 메쉬(Mesh)와 같은 3차원 데이터로 구성된 영상으로, 현실 세계의 물체, 움직임 등을 가상 세계로 가져와 자유로운 시점으로 몰입형 경험을 제공하는 핵심 기술이다. 하지만 이러한 고품질의 볼류메트릭 영상을 제공하기 위해서는 대규모 데이터 전송 능력 뿐만 아니라 수신된 데이터를 처리하고 렌더링하기 위한 컴퓨팅 자원이 필수적으로 요구된다.

미가공(Raw) 상태의 볼류메트릭 영상은 수 Gbps에서 수십 Gbps 고대역폭을 요구한다 [1]. 데이터 전송 효율을 높이기 위해 Draco [2]와 같은 고효율 압축 알고리즘이나 사용자 시야 영역만을 전송하는 뷰포트 적응(Viewport-adaptive) 기법 등이 적용되고 있지만 [3], 여전히 높은 대역폭을 필요로 한다. 그러나 실제 통신 환경에서 이러한 높은 요구량을 지속적으로 보장하기는 어려우며, 일시적인 대역폭 부족만으로도 재생 끊김(Stalling)과 같은 사용자 경험(Quality of Experience) 저하가 발생할 수 있다. 이러한 문제를 해결하기 위해 현재의 비디오 스트리밍 서비스는 네트워크 상태에 따라 최적의 품질을 동적으로 선택하는 ABR(Adaptive Bitrate) 알고리즘을 사용하며, 볼류메트릭 영상 스트리밍에서도 필수적인 요소이다.

그러나 기존의 ABR 알고리즘들은 주로 2D 비디오를 위해 설계되어, 비트레이트 결정 시 주로 네트워크 대역폭과 버퍼 상태를 고려하고, 단말의 부하는 효과적으로 반영하지 못한다. 볼류메트릭 데이터는 디코딩 및 3D 렌더링 과정에서 많은 연산을 요구하고, 단말의 처리 성능 부족은 시스템의 병목으로 이어진다. 이는 고품질의 데이터를 받을 수 있는 안정적인 네트워크 상황에서도 낮은 프레임 속도(FPS, Frame per second)로 인한 사용자 경험 저하를 초래한다.

볼류메트릭 영상 서비스는 고성능 데스크톱부터 모바일 기기, HMD(Head Mounted Display)까지 다양한 단말 환경을 목표로 하며, 고사양 환경에서는 드러나지 않던 ABR 알고리즘의 비효율성이 자원이 제한된 단말에서는 성능 저하로 이어질 수 있

다. 따라서 범용적인 볼류메트릭 스트리밍 서비스를 위해서는 서로 다른 단말 환경에서 기존 ABR 알고리즘이 실제로 어떻게 반응하고 동작하는지에 대한 성능 평가가 필요하다.

따라서 본 연구는 볼류메트릭 영상 스트리밍 환경에서 단말의 컴퓨팅 성능이 ABR 알고리즘의 동작 및 스트리밍 품질에 미치는 영향을 실험적으로 분석한다. 특히 새로운 ABR 알고리즘을 제안하기 앞서 컴퓨팅 자원 제약이 기존 ABR 알고리즘에 주는 부정적인 영향을 정량적으로 분석하는데 초점을 맞춘다. 이를 위해 서로 다른 성능을 가진 두 가지 단말 환경에서 대표적인 ABR 알고리즘인 대역폭 기반(Throughput-based), 버퍼 기반(Buffer-based), 그리고 하이브리드(Hybrid) 알고리즘의 성능을 비교 평가한다. 이를 통해 향후 볼류메트릭 영상 스트리밍 시스템이 네트워크 상태 뿐만 아니라 단말의 자원 상태까지 인지할(Device-aware) 필요성을 제시한다. 본 연구의 기여는 다음과 같다.

- 1) 안정적 및 동적 네트워크 환경에서 볼류메트릭 영상 적응형 스트리밍을 수행하는 테스트베드를 구축하고 서로 다른 성능의 단말에서 ABR 알고리즘의 성능을 분석
- 2) 비트레이트, FPS, 총 재생 시간 등 성능 지표를 통해 병목 환경에서 기존 ABR 알고리즘들의 비효율성을 확인하고, 향후 단말 인지형 스트리밍 시스템의 필요성 강조

II. 관련 연구

본 장에서는 볼류메트릭 영상 스트리밍 연구 현황과 적응형 비트레이트 알고리즘을 설명한다.

1) 볼류메트릭 영상 스트리밍

볼류메트릭 영상은 3차원 공간 정보 및 색상 정보를 포함하고 있어, 기존 2D 영상 대비 막대한 데이터 전송량을 요구하며, 이를 효율적으로 전송하기 위한 연구들이 진행되었다.

초기 연구인 ViVo [3]는 사용자 시점 및 가려짐 여부, 거리 기반 처리를 통해 대역폭을 절감했으며, GROOT [1]는 사용자 시점 처리에 더해 효율적인 PD 트리 기반 압축 방식 제안으로 대역폭을 절감하였다. Vues [4]는 고효율 코덱을 이용하여, YuZu [5]는 3D 초해상화를 이용해 대역폭을 절감하는 시스템을 제안하였다. 또한 PCC-DASH

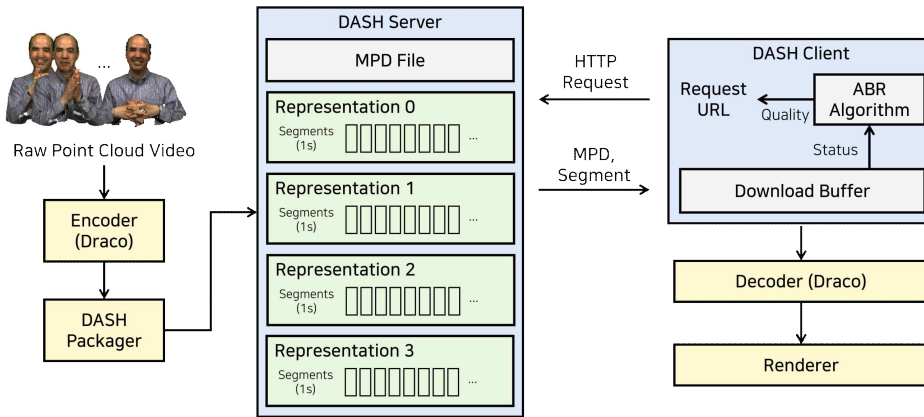


그림 1. DASH 기반 볼류메트릭 영상 스트리밍 시스템 구조
Fig. 1. DASH-based Volumetric Video Streaming System Architecture

[17], DASH-PC [18], VV-DASH [19]와 같이 적응형 스트리밍 기술을 볼류메트릭 영상에 적용하려는 시도들도 있었다. 이 연구들은 DASH 표준을 확장하여 다중 화질을 표현하는 방식에 집중하였다. 이러한 시스템들은 Draco [2], V-PCC [6]와 같이 압축을 위한 인코딩/디코딩 알고리즘을 활용하기 때문에, 클라이언트에서 디코딩 및 렌더링을 위한 연산이 필수적으로 수반된다. 따라서 볼류메트릭 영상 시스템에서 고품질 스트리밍을 보장하기 위해서는 대역폭 절감을 넘어, 단말의 연산 능력까지 고려한 적응형 스트리밍 전략이 요구된다.

2) 적응형 비트레이트(ABR) 알고리즘

변화하는 네트워크 환경에서 끊임 없는 재생을 제공하기 위해, HTTP 적응형 스트리밍 기술인 MPEG-DASH(Dynamic Adaptive Streaming over HTTP) [7]가 널리 사용된다. MPEG-DASH에서는 영상을 수 초 길이의 세그먼트로 분할하고 다양한 화질로 인코딩되며, 클라이언트가 ABR 알고리즘을 이용해 최적의 화질을 선택하여 다운로드 및 재생한다. 기존 ABR 알고리즘은 크게 대역폭 기반(Throughput-based)과 버퍼 기반(Buffer-based) 알고리즘으로 분류된다.

대역폭 기반 알고리즘은 이전 세그먼트의 다운로드 속도를 이용해 대역폭을 추정하여 가능한 대역폭 내에서 화질을 선택한다. 이런 대역폭 기반 알고리즘은 네트워크 자원을 최대한로 활용할 수 있지만, 네트워크 변동에 민감해 불안정성을 보일 수 있다. 버퍼 기반 알고리즘은 재생 버퍼량을 기준으로 화질을 선택하며, 대표적으로 BOLA [8] 알고리즘은 최적화 기법을 이용해 재생 안전성을 확보하였다.

또한 대역폭과 버퍼 상태를 동시에 고려해 사용자 경험(QoE)을 최적화하는 시도들도 있었다. MPC [9]는 대역폭 및 버퍼 상태를 함께 이용하고, QoE 식을 정의해 QoE를 최대화하는 화질을 선택한다. 최근에는 Pensieve [10]와 같이 강화학습을 이용한 방법도 제시되었다.

이러한 기존 알고리즘들은 주로 2D 영상 스트리밍을 위해 설계되었으며, 대부분 네트워크 대역폭 및 버퍼 점유율을 이용해 품질을 결정한다. 볼륨메트릭 영상과 같이 디코딩 및 렌더링 등으로 인해 단말의 부하가 추가적으로 존재할 수 있는 환경에서는 단말의 상태를 충분히 반영하지 못해 사용자 경험을 저해할 수 있다.

따라서 본 연구에서는 단말의 컴퓨팅 성능이 기존 대역폭 및 버퍼 중심 ABR 알고리즘의 동작과 블루메트릭 영상 스트리밍 품질에 어떤 영향을 미치는지에 대한 정량적인 분석을 제시한다.

Ⅲ. 시스템 설계

이 장에서는 ABR 알고리즘의 성능 평가를 위해 설계한 분류메트릭 영상 스트리밍 시스템과 ABR 알고리즘을 서술한다.

1) 볼류메트릭 영상 스트리밍 시스템

그림 1은 볼류메트릭 영상 스트리밍 시스템의 전체 구조를 보여준다. 이 시스템은 HTTP 서버 및 클라이언트로 구성된 DASH 기반의 적응형 스트리밍 구조를 따른다.

서버 측에서는 적응형 스트리밍을 지원하기 위해 원본 포인트 클라우드 영상에서 각 프레임마다 포

인트의 개수를 10%, 40%, 70%, 100%로 샘플링하여 총 4가지의 화질의 영상을 생성하였다. 각 화질의 포인트 클라우드 영상은 Draco 인코더를 이용해 압축되며, 압축된 데이터는 1초 단위의 고정된 길이를 갖는 세그먼트로 패키징된다. 이 과정에서 영상의 비트레이트, 세그먼트 URL 등의 스트리밍에 필요한 메타데이터를 포함하는 MPD(Media Presentation Description) 파일이 함께 작성된다.

클라이언트는 ABR 알고리즘을 이용해 네트워크 상태에 따른 최적의 화질을 선택한다. 이를 위해 서버로부터 MPD 파일을 수신 및 파싱하고, 선택된 화질에 해당하는 세그먼트의 URL로 HTTP 요청을 전송한다. 수신된 세그먼트는 프레임 단위로 분할해 다운로드 버퍼에 저장되며, 비동기적으로 동작하는 디코더 및 렌더러를 거쳐 화면에 시각화한다.

2) 적응형 비트레이트 알고리즘

본 연구에서는 단말의 컴퓨팅 병목이 적응형 스트리밍에 미치는 영향을 분석하기 위해 세 가지 유형의 ABR 알고리즘을 사용하였다.

첫 번째는 대역폭 기반(Throughput-based, TB) 알고리즘이다. 대역폭 기반 알고리즘에서는 이전 세그먼트들의 다운로드 속도를 기반으로 대역폭을 추정하고 비트레이트를 결정한다. 본 연구에서는 추정된 대역폭($T_{measured}$)의 95% 내에서 가장 좋은 화질의 세그먼트를 선택하도록 하였다.

$$B_{TB} = \max\{b \in B_{avail} | b \leq 0.95 \times T_{measured}\}$$

두 번째는 버퍼 기반(Buffer-based, BB) 알고리즘이다. 버퍼 기반 알고리즘에서는 클라이언트의 현재 다운로드 버퍼 점유율(L_{buf})을 기준으로 비트레이트를 결정하며, 본 연구에서는 다운로드 버퍼 점유율의 기준을 20%, 50%, 70%로 설정하고 구간에 따라 화질을 선택하였다.

$$B_{BB} = \begin{cases} \text{Quality 0, when } L_{buf} < 0.2 \\ \text{Quality 1, when } 0.2 \leq L_{buf} < 0.5 \\ \text{Quality 2, when } 0.5 \leq L_{buf} < 0.7 \\ \text{Quality 3, when } 0.7 \leq L_{buf} \end{cases}$$

세 번째는 대역폭 기반 알고리즘과 버퍼 기반 알고리즘을 모두 사용하는 하이브리드(Hybrid, HB) 알고리즘이다. 두 알고리즘의 결과값 중 더 낮은 값을 선택하도록 하여, 각 알고리즘에서의 위험 신호를 모두 받아들이는 안전한 선택을 하도록 하였다.

$$B_{HB} = \min(B_{TB}, B_{BB})$$

IV. 시스템 구현

이 장에서는 볼류메트릭 영상 스트리밍 시스템의 서버, 클라이언트에 대한 구체적인 구현 방식을 설명한다.

1) 서버

볼류메트릭 영상의 적응형 스트리밍을 위한 DASH 서버는 Go 언어 [11]로 구현하였다. 볼류메트릭 영상으로는 MVUB(Microsoft Voxelized Upper Bodies) 데이터셋 [12]의 Phil-9 영상을 사용했으며, 프레임 당 평균 33.4만 개의 포인트로 구성되어 있다. 총 245 프레임(30 FPS에서 약 8초) 길이의 영상을 반복해 총 1800 프레임 길이의 영상으로 수정하였다. 원본 볼류메트릭 영상을 여러 화질로 샘플링하기 위해서 Open3D [13] 라이브러리를 사용했으며, 각 화질에 대한 프레임 당 용량 및 대역폭 정보는 표 1과 같다. 볼류메트릭 영상의 각 프레임을 Draco 코덱을 이용해 압축하기 위해서 DracoPy [14] 라이브러리를 사용하였다. 압축 후 각 화질에 대한 프레임 당 용량 및 대역폭 정보는 표 2와 같다. 적응형 스트리밍을 위한 세그먼트의 길이는 1초(30 프레임)으로 설정하였다.

표 1. 여러 품질로 제작된 포인트 클라우드 정보 (압축 전)
Table 1. Point cloud information with different quality (Before Compression)

Quality Level	Avg. # of Points	Avg. Size (MB)	Mbps @30Hz
0	33,360	0.68	170.36
1	133,443	2.71	681.32
2	233,525	4.74	1192.28
3	333,608	7.09	1783.29

표 2. 여러 품질로 제작된 포인트 클라우드 정보 (압축 후)
Table 2. Point cloud information with different quality (After Compression)

Quality Level	Avg. # of Points	Avg. Size (MB)	Mbps @30Hz
0	33,360	0.20	50.30
1	133,443	0.72	179.98
2	233,525	1.20	301.19
3	333,608	1.66	418.49

2) 클라이언트

볼류메트릭 영상의 적응형 스트리밍을 위한 DASH 클라이언트는 Python으로 구현하였다. 세그먼트 다운로드, 디코딩, 렌더링으로 구성되는 클라이언트의 파이프라인을 효율적으로 수행하기 위해 asyncio(Python 내장), aiohttp [15]와 같은 비동기 라이브러리와 멀티프로세싱을 사용하였다. 다운로드 버퍼의 크기는 15초(450 프레임)로 설정하였다. 프레임 디코딩을 위해서는 DracoPy [14] 라이브러리를, 렌더링을 위해서는 Open3D [13] 라이브러리를 사용하였다.

V. 실험 환경

이 장에서는 볼류메트릭 영상 스트리밍 시스템에서 단말의 성능이 ABR 알고리즘에 미치는 영향을 분석하기 위한 실험 환경을 설명한다.

1) 테스트베드 구성

단말의 컴퓨팅 성능에 따른 ABR 알고리즘의 차이를 분석하기 위해 서로 다른 하드웨어 사양을 가진 두 개의 물리적 단말에 시스템을 설치하였다.

- 단말 A (고성능): AMD Ryzen 9 7900 CPU, RTX 5080 GPU, 64GB RAM
- 단말 B (저성능): Intel Xeon(R) Silver 4125R CPU, Quadro RTX 5000 GPU, 64GB RAM

표 3은 두 단말에서 고정된 화질을 사용하는 경우 파이프라인 전체의 처리 성능(FPS)을 보여준다.

표 3. 각 품질별 단말의 처리 성능 (FPS)
Table 3. Processing performance of each device by quality level (FPS)

Quality Level	Device A (High-end)	Device B (Low-end)
0	29.36	29.06
1	29.64	26.88
2	27.30	19.47
3	26.99	15.00

실험 환경의 일관성을 위해 본 연구에서는 Docker [16] 기반의 가상화 환경을 이용하였다. 각 단말은 서버, 클라이언트, 네트워크 에뮬레이터를 모두 독립된 Docker 컨테이너를 이용해 구동하였다. 볼류메트릭 영상의 렌더링은 헤드리스(Headless) 모

드로 이루어졌으며, 리눅스의 TC (Traffic Control) 유틸리티를 이용해 Docker 네트워크 내에서 클라이언트의 수신 대역폭을 제어하였다.

2) 네트워크 시나리오

단말의 컴퓨팅 성능에 따른 ABR 알고리즘의 차이를 분석하기 위해 두 가지 네트워크 시나리오를 정의하였다.

- 시나리오 1 (정적 네트워크): 대역폭을 1Gbps로 고정하여 모든 품질의 볼류메트릭 영상을 제공할 수 있게 하였다.
- 시나리오 2 (동적 네트워크): 볼류메트릭 영상의 화질 별 대역폭을 고려해, (600Mbps, 400Mbps, 200Mbps, 100Mbps)의 대역폭을 (5초, 5초, 5초, 10초) 주기로 반복하는 시나리오를 구성하였다.

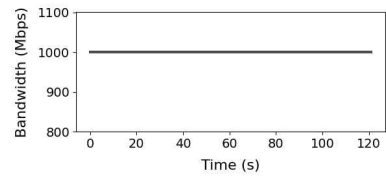


그림 2. 시나리오 1 (정적 네트워크)
Fig. 2. Scenario 1 (Stable Network)

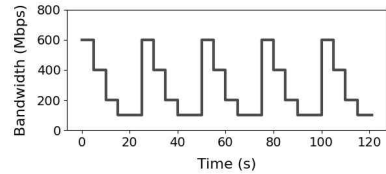


그림 3. 시나리오 2 (동적 네트워크)
Fig. 3. Scenario 2 (Dynamic Network)

3) 실험 과정 및 평가 지표

단말의 성능, 네트워크 시나리오, ABR 알고리즘을 조합하여 총 12가지 경우에 대해 실험을 수행하였다. 각 실험에서는 목표 재생 시간이 60초인(총 1800 프레임) 볼류메트릭 영상을 끝까지 재생하며 평가 지표를 수집하였다. 본 연구에서는 평균 비트레이트(Average Bitrate), 평균 프레임 속도(Average FPS), 총 재생 시간(Total Playback Time))의 총 세 가지 성능 지표를 수집하였다. 평균 비트레이트는 ABR 알고리즘이 선택한 세그먼트의 평균 화질 수준을 나타낸다. 평균 프레임 속도는 실제 렌더링된 초당 프레임 수로, 30 FPS의 원본 영상과 비교했을 때 병목 여부를 보여준다. 총 재생 시간은 1800 프

표 4. 정적 네트워크 환경에서 단말 성능에 따른 스트리밍 품질

Table 4. Streaming quality by device capability under stable network conditions

ABR	Device	Bitrate (Mbps)	FPS	Total Playback Time (s)
Throughput-based	High-end	414.54	27.13	66.38
	Low-end	414.54	16.40	109.82
Buffer-based	High-end	368.24	27.74	64.92
	Low-end	366.28	17.38	103.62
Hybrid	High-end	368.24	27.88	64.60
	Low-end	366.28	17.64	102.11

표 5. 동적 네트워크 환경에서 단말 성능에 따른 스트리밍 품질

Table 5. Streaming quality by device capability under dynamic network conditions

ABR	Device	Bitrate (Mbps)	FPS	Total Playback Time (s)
Throughput-based	High-end	210.48	28.30	63.64
	Low-end	191.00	23.73	75.91
Buffer-based	High-end	253.08	28.37	63.48
	Low-end	299.01	20.35	88.49
Hybrid	High-end	181.33	28.90	62.32
	Low-end	167.39	24.63	73.13

레이م 길이의 영상을 모두 재생하는데 소요된 시간으로, 목표 재생 시간인 60초와 비교했을 때의 지연을 보여준다.

VI. 실험 결과 및 분석

1) 단말 성능이 스트리밍 품질에 미치는 영향

단말의 컴퓨팅 성능 차이가 스트리밍 품질에 미치는 영향을 분석하기 위해 네트워크 대역폭이 충분한 상황(시나리오 1)에서 고성능 단말과 저성능 단말의 성능을 분석하였다. 표 4는 각 ABR 알고리즘에서 고성능, 저성능 단말의 볼류메트릭 영상 스트리밍 성능을 보여준다.

고성능 단말(High-end)의 경우, 세 가지 ABR 알고리즘 모두 최고 화질을 선택하였으며, 약 27 FPS의 부드러운 재생 성능을 보였고, 총 재생 시간 역시 약 64~66초로 측정되어 목표 재생 시간인 60초에 근접한 성능을 보여주어 안정적으로 스트리밍이 진행되었음을 확인하였다.

반면에 저성능 단말(Low-end)의 경우 성능 저하가 관찰되었다. 충분한 네트워크 환경에 따라 세 가지 ABR 알고리즘 모두 최고 화질을 선택하였으나, 단말의 디코딩, 렌더링 과정에서 처리의 한계로 평균 FPS는 16~17를 보였으며, 렌더링 지연으로 인해 총 재생 시간이 100초 이상으로 측정되었다.

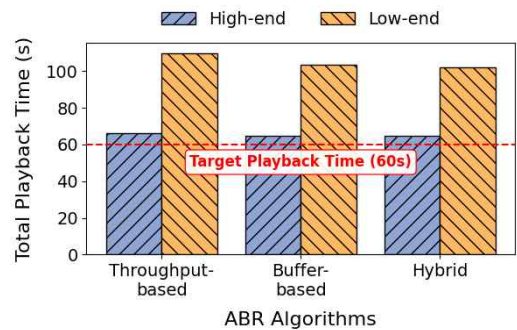


그림 4. 정적인 네트워크 상황에서 단말 성능에 따른 총 재생 시간

Fig. 4. Total playback time by device capability under stable network conditions

그림 4는 이러한 총 재생 시간의 격차를 보여준다.

이는 네트워크 상태 혹은 버퍼 상태를 이용하는 기존 ABR 알고리즘이 볼류메트릭 영상과 같이 연산 부하가 있는 환경에서는 올바르게 동작하지 못하고 사용자 경험을 최적화하지 못한다는 것을 보여준다.

2) 동적 네트워크 상황에서 ABR 알고리즘 분석

동적 네트워크 환경에서 단말의 컴퓨팅 성능이 ABR 알고리즘의 동작에 미치는 영향을 분석하기 위해, 대역폭이 수시로 변동하는 상황(시나리오 2)

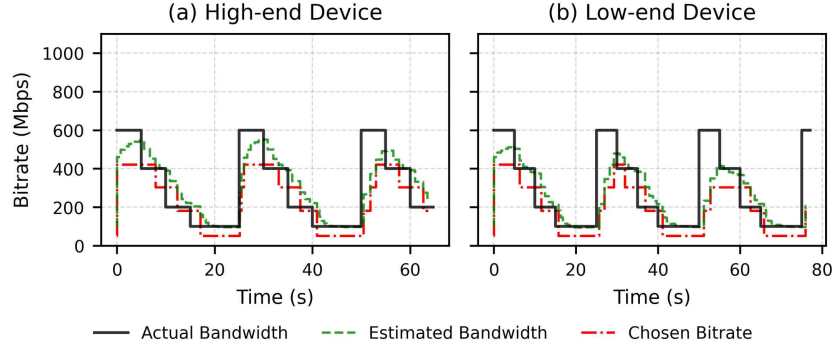


그림 5. 동적인 네트워크 상황에서 대역폭 기반 알고리즘의 단말 성능별 비트레이트 선택
Fig. 5. Bitrate selection of Throughput-based algorithm by device capability under dynamic network conditions

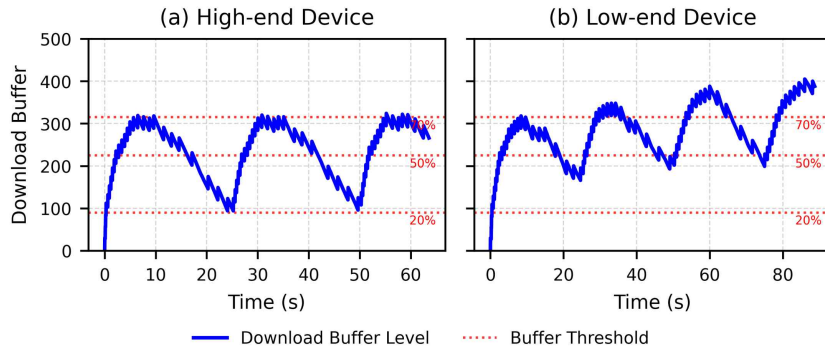


그림 6. 동적인 네트워크 상황에서 버퍼 기반 알고리즘의 단말 성능별 버퍼 점유율
Fig. 6. Download buffer occupancy of Buffer-based algorithm by device capability under dynamic network conditions

에서 고성능 단말과 저성능 단말의 성능을 분석하였다. 표 5는 동적 네트워크 환경에서 각 ABR 알고리즘에 따른 고성능, 저성능 단말의 볼류메트릭 영상 스트리밍 성능을 보여준다. 정적 네트워크 환경과 달리, 동적 네트워크 환경에서는 대역폭 변동으로 인한 비트레이트 조정이 있었다. 전반적으로 낮은 화질의 선택이 저성능 단말의 처리 부담을 줄여서 평균 FPS와 총 재생 시간은 개선되었지만, 세부적인 적응 방식은 ABR 알고리즘마다 다른 특성을 보였다.

먼저 대역폭 기반 알고리즘은 그림 5와 같이 고성능 단말과 저성능 단말 모두 대역폭에 따라 비트레이트를 조절하는 모습을 보였다. 고성능 단말은 평균 28.30 FPS의 재생 성능과 총 재생 시간 약 63초로 안정적인 ABR 동작을 보여주었다. 저성능 단말에서 역시 대역폭에 따라 적절하게 비트레이트

를 조절하였지만, 평균 비트레이트는 약 20Mbps 낮게 측정되었다. 이는 저성능 단말에서 디코딩, 렌더링에서의 처리 부하로 인해 시스템 자원이 포화되어 네트워크 I/O 처리가 지연되었고, 다운로드 속도가 느리게 측정되어 가용 대역폭 또한 낮게 추정되는 구간이 발생했기 때문이다.

버퍼 기반 알고리즘에서는 저성능 단말이 299.01Mbps의 비트레이트로 고성능 단말의 253.08Mbps보다 높은 비트레이트를 보였다. 그림 6은 동적 네트워크 상황에서 버퍼 기반 알고리즘의 버퍼 점유율 추이를 보여주며, 이러한 대조적인 비트레이트 선택에 대한 원인을 보여준다. 고성능 단말(그림6-(a))은 디코딩 및 렌더링으로 인한 부하가 거의 없기 때문에 네트워크 상황의 악화로 다운로드 속도가 느려지는 구간에서는 확보된 프레임을 원활하게 소비하며 유동적인 등락을 보였다. 반면에

저성능 단말(그림6-(b))는 디코딩 및 렌더링으로 인한 부하로, 버퍼에 확보된 고품질 프레임을 원활하게 소비하지 못하였으며, 버퍼 점유율이 점차 상승하는 경향을 보였다. 이러한 버퍼 점유율의 증가는 단말의 컴퓨팅 성능이 부족한 상황에서 버퍼 기반 알고리즘이 고화질 세그먼트의 요청을 지속하게 하는 원인이 되었다.

하이브리드 알고리즘에서는 대역폭과 버퍼 지표를 둘 다 이용하여 안전한 선택을 하기 때문에, 고성능과 저성능 단말 모두 가장 낮은 평균 비트레이트를 보였다. 이에 따라 대역폭 기반 알고리즘과 버퍼 기반 알고리즘보다 높은 FPS와 짧은 총 재생 시간이 측정되었으며, 이는 재생 안정성을 위해서는 단말의 컴퓨팅 성능에 따른 비트레이트 조절이 필요함을 보여준다.

VII. 논 의

본 연구의 실험 결과는 볼류메트릭 영상 스트리밍과 같이 디코딩, 렌더링으로 인해 단말 병목이 발생할 수 있는 상황에서 대역폭, 버퍼 등의 지표를 이용하는 기존 ABR 알고리즘들이 실제 재생 성능을 반영하지 못하고 선택을 내린다는 것을 보여준다. 특히 버퍼 기반 ABR 알고리즘은 디코딩 및 렌더링 부하로 인해 생긴 버퍼 점유율의 상승을 안정 상태로 판단하여 고화질의 세그먼트를 지속적으로 요청하였으며, 대역폭 기반 알고리즘 역시 시스템 부하로 인해 영향을 받는 모습을 보여주었다. 이는 볼류메트릭 영상 스트리밍 같이 많은 연산이 필요한 미디어 서비스에서는 네트워크 지표만으로는 사용자 경험 보장이 어려우며, 향후 스트리밍 시스템에서는 단말의 자원 상태를 모니터링해 비트레이트 결정 과정에 반영하거나 더욱 지능적인 결정을 내리는 알고리즘을 사용할 필요가 있다는 점을 보여준다.

본 연구는 실제 볼류메트릭 영상 스트리밍 시스템 구현을 통해 적응형 스트리밍에서 단말 성능의 영향을 보여주었지만, 다음과 같은 한계점을 가지며, 향후 연구를 통해 보완될 필요가 있다. 첫 번째는 ABR 알고리즘의 범위를 제한했다는 점이다. 본 연구에서는 단말 성능의 영향을 확인하기 위해 기초적인 ABR 알고리즘 세 종류만을 대상으로 분석하

였다. 향후 연구에서는 BOLA, MPC, Pensieve 등 최적화, 기계 학습 기반으로 고도화된 알고리즘들이 어떤 동작 특성을 보이는지 분석할 필요가 있다. 두 번째는 두 종류의 단말 환경만을 다루었다는 점이다. 실제 서비스 환경에서는 무선 네트워크(5G, WiFi 등)의 변동성, 모바일 기기, HMD 기기 등 더욱 다양한 기기와 함께 복잡한 변수들이 존재하기 때문에 더욱 다양한 환경과 기기에서의 분석이 요구된다. 향후 연구에서는 무선 네트워크 환경과 다양한 기기를 포함하는 시나리오에서 시스템의 성능을 분석할 것이다. 또한 자원 사용량 및 모듈별 지연 시간 등 더욱 세분화된 지표를 이용해 정밀한 분석을 수행할 예정이다. 마지막으로 본 연구에서는 기존 ABR 알고리즘들의 동작 특성들을 분석하는데 집중하였고, 이를 해결하기 위한 알고리즘은 제안하지 않았다는 점이다. 향후 연구에서는 기존 ABR 알고리즘들의 판단 지표 뿐만 아니라 단말의 성능, 볼류메트릭 영상 스트리밍의 특성들을 종합적으로 고려하여 사용자 경험을 개선 할 수 있는 적응형 스트리밍 알고리즘을 설계할 예정이다.

VIII. 결 론

본 연구는 볼류메트릭 영상 스트리밍 환경에서 단말의 컴퓨팅 성능이 적응형 스트리밍 알고리즘 동작 및 스트리밍 품질에 미치는 영향을 분석하였다. 이를 위해 DASH 기반 볼류메트릭 영상 적응형 스트리밍 시스템을 구현하고, 서로 다른 성능의 단말에 시스템을 설치하여 비교 실험을 진행하였다. 그 결과 기존 ABR 알고리즘들이 볼류메트릭 영상의 디코딩 및 렌더링으로 생기는 단말의 부하를 효과적으로 반영하지 못하는 것을 확인하였고, 사용자 경험 개선을 위해서는 단말의 성능을 비롯해 볼류메트릭 영상의 특성을 반영한 알고리즘의 필요하다는 점을 시사하였다. 본 연구는 새로운 ABR 알고리즘 제안에 앞서, 컴퓨팅 자원의 제약이 기존 ABR 알고리즘에 미치는 부정적인 영향을 정량적으로 확인하였다. 향후 연구로는 더욱 고도화된 ABR 알고리즘 및 다양한 단말 및 네트워크 환경에서 더욱 세분화된 지표를 활용한 분석과 이러한 분석 결과가 반영된 적응형 스트리밍 알고리즘 설계를 계획하였다.

References

- [1] Lee, Kyungjin, et al. "GROOT: a real-time streaming system of high-fidelity volumetric videos." . In Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (pp. 1–14).2020.
- [2] Draco 3D Data Compression, Retrieved Nov., 21, 2025, <https://google.github.io/draco>.
- [3] Han, Bo, et al. "ViVo: Visibility-aware mobile volumetric video streaming." . In Proceedings of the 26th annual international conference on mobile computing and networking (pp. 1–13).2020.
- [4] Liu, Yu, et al. "Vues: Practical mobile volumetric video streaming through multiview transcoding." Proceedings of the 28th Annual International Conference on Mobile Computing And Networking. 2022
- [5] Zhang, Anlan, et al. "YuZu:Neural-Enhanced volumetric video streaming." 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22). 2022.
- [6] MPEG V-PCC, Retrieved Nov., 21, 2025, <https://www.iso.org/standard/83535.html>.
- [7] Stockhammer, Thomas. "Dynamic adaptive streaming over HTTP-- standards and design principles." Proceedings of the second annual ACM conference on Multimedia systems. 2011.
- [8] Spiteri, Kevin, Rahul Urgaonkar, and Ramesh K. Sitaraman. "BOLA: Near-optimal bitrate adaptation for online videos." IEEE/ACM transactions on networking 28.4 (2020): 1698-1711.
- [9] Yin, Xiaoqi, et al. "A control-theoretic approach for dynamic adaptive video streaming over HTTP." Proceedings of the 2015 ACM conference on special interest group on data communication. 2015.
- [10] Mao, Hongzi, Ravi Netravali, and Mohammad Alizadeh. "Neural adaptive video streaming with pensieve." Proceedings of the conference of the ACM special interest group on data communication. 2017.
- [11] The Go Programming Language, Retrieved Nov., 21, 2025, <https://go.dev/>
- [12] Loop, Charles, et al. Microsoft Voxelized Upper Bodies - A Voxelized Point Cloud Dataset, Retrieved Nov., 21, 2025, <https://plenodb.jpeg.org/pc/microsoft/>. 2016.
- [13] Zhou, Qian-Yi, et al. "Open3D: A modern library for 3D data processing." arXiv preprint arXiv:1801.09847 (2018).
- [14] DracoPy, Retrieved Nov., 21, 2025, <https://github.com/seung-lab/DracoPy>.
- [15] Async http client/server framework, Retrieved Nov., 21, 2025, <https://github.com/aio-lib/aiohttp>.
- [16] Merkel, Dirk. "Docker: lightweight linux containers for consistent development and deployment." Linux j 239.2 (2014): 2.
- [17] 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.
- [18] Hosseini, Mohammad, and Christian Timmerer. "Dynamic adaptive point cloud streaming." Proceedings of the 23rd Packet Video Workshop. 2018.
- [19] Heidarirad, Hadi, and Mea Wang. "VV-DASH: A Framework for Volumetric Video DASH Streaming." Proceedings of the 16th ACM Multimedia Systems Conference. 2025.

유 상 우 (Sangwoo Ryu)



2020 포항공과대학교, 컴퓨터
공학과 학사

2020 ~ 포항공과대학교, 인공
지능대학원 석/박사 통합과
정

<관심분야> 2D/3D 비디오 스
트리밍, 영상회의, 인공지능

홍 원 기 (James Won-Ki Hong)



2020~현재 포스텍 크립토블록
체인연구센터 공동센터장

2019-2023 포스텍 교육혁신센
터장

2012~2014 KT 종합기술원 원
장

2008~2012 포스텍 정보전자융
합공학부장

2008~2010 포스텍 컴퓨터공학과 주임교수

2007~2010 포스텍 정보통신연구소 연구소장

2007~2011 포스텍 정보통신대학원장

1995~현재 포스텍 컴퓨터공학과 교수

1992~1995 Univ. of Western Ontario, 연구교수

<관심분야> AI 기반 네트워크관리, SDN/NFV, 무
크기반 교육, 블록체인 및 암호화폐, Vmeeting
화상회의시스템

세션 및 패킷 번들 단위에서의 WeChat 행위 분석

김 지 민*, 김 란 아*, 장 윤 성**, 백 의 준***, 김 명 섭^o

WeChat Behavior Analysis with Session and Packet Bundle Level

Ji-min Kim*, Ran-A Kim*, Yoon-Seong Jang**, Ui-Jun Baek***, Myung-Sup Kim^o

요 약

현대 네트워크 환경에서는 암호화로 인해 페이로드 기반 분석이 어려워지면서, 세션과 패킷의 통계적 특성을 활용한 행위 기반 트래픽 분석의 중요성이 커지고 있다. 본 연구는 WeChat 트래픽을 대상으로, 세션 구조와 시간적 패턴을 통합 학습하는 이중 Convolutional Neural Network(CNN) 기반 행위 분석 프레임워크를 제안하였다. 특히 동일 세션 내의 행위 분석이 불가능하다는 세션 단위 분석의 한계를 보완하기 위해 ‘패킷 번들(Packet Bundle)’ 개념을 새롭게 도입하여 세션 내부의 시간 구간별 패턴 변화를 정량적으로 분석하고, Chat과 File Transfer와 같은 동일 세션 내 발생 행위를 효과적으로 구분하였다. 제안된 모델은 세션 단위의 2D CNN과 패킷 번들 단위의 1D CNN을 결합하여 Voice, Video, Chat/File Transfer, Others 세션을 높은 정확도로 분류하였으며, 1D CNN은 평균 97%의 정확도를 기록하였다. 또한 WhatsApp, KakaoTalk 데이터셋에서도 유사한 성능을 보여 방법의 범용성과 실용성을 입증하였다. 본 연구는 세션 및 패킷 번들 기반의 시공간적 특징 학습을 통해 암호화된 환경에서도 세밀한 행위 인식이 가능한 효율적 분석 프레임워크를 제시한다.

Key Words : Network Traffic Classification, WeChat Behavior Analysis, Packet Bundle

ABSTRACT

In modern network environments, encryption has made payload-based analysis increasingly difficult, highlighting the importance of behavioral traffic analysis that leverages the statistical characteristics of sessions and packets. This study proposes a dual CNN-based behavioral analysis framework that jointly learns the structural and temporal patterns of WeChat traffic. To address the limitations of session-level analysis, a new concept called the “Packet Bundle” is introduced to quantitatively analyze temporal variations within sessions and effectively distinguish similar behaviors such as Chat and File Transfer. The proposed model combines a session-level 2D CNN with a packet-bundle-level 1D CNN to classify Voice, Video, Chat/File Transfer, and Others sessions with high accuracy, achieving an average classification accuracy of 97% for the 1D CNN. Furthermore, comparable results obtained on a WhatsApp, KakaoTalk dataset demonstrate the generalizability and practicality of the proposed approach. This study presents an efficient analysis framework capable of fine-grained behavior recognition in encrypted environments through packet-bundle-based spatio-temporal feature learning.

※ 이 논문은 과기정통부·정보통신기획평가원의 정보통신방송표준개발지원(R&D, 정보화)사업으로 수행한 결과(No. RS-2025-02219319, 양자컴퓨터 공격에도 안전한 양자암호기반 제로트러스트 보안 네트워크/서비스 및 제어/관리 기술 표준개발), 2025년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(RS-2025-16067207)

• First Author : Korea University Department of Computer Convergence Software, illiard1209@korea.ac.kr

o Corresponding Author : Korea University Department of Computer Convergence Software, tmskim@korea.ac.kr

* Korea University Department of Mobility Science and Engineering, zufrieden7@korea.ac.kr

** Korea University Department of Computer Convergence Software, brave1094@korea.ac.kr

*** Korea Institute of Science and Technology Information (KISTI), pb1069@kisti.re.kr

I. 서 론

네트워크 기술이 발달함에 따라 이를 체계적으로 관리하고 모니터링하는 기술에 필요성이 늘어나고 있다. 그 중 트래픽 분류는 이상 트래픽을 탐지하거나, 사용자의 행위를 분석하는 등 여러 확장성을 가지고 있어 네트워크 관리 기술에서 중요한 역할을 하고 있다.

네트워크 기술 발전에 따라 트래픽 분류 방법 또한 발전하였는데, 규칙 기반(Rule-based) 접근 방식인 포트 기반[1], DPI(Deep Packet Inspection) 기반[2]과 머신러닝 접근 방식인 통계 기반[3], 그리고 행위 기반[4] 방법이다. 그러나 동적 포트의 등장과, 암호화된 페이로드로 인하여 규칙 기반의 분류 기법들은 사용하는 데에 한계가 있으며, 머신러닝 기법의 경우 특징을 직접 추출하므로 적절한 특징을 설계해야 한다는 점과 개발자의 역량에 의해 분류 정확도가 변화한다는 한계점이 존재한다.

이를 해결하기 위하여 여러 가지 딥러닝 모델들을 트래픽 분야에 접목하여 트래픽 분류를 하는 연구들이 등장하고 있다. 세션을 이미지로 입력하여 네트워크 분류를 하는 CNN 모델 [5]과, 시계열 데이터적 관점에서 처리하는 RNN/LSTM [6]모델 그리고 트래픽을 자연어의 관점에서 바라보는 Transformer 모델 [7]까지 등장하였다. 그러나 이러한 DL 모델들도 기존의 트래픽적 특징을 보존하여 입력으로 사용하는 방법을 채택하고 있지 않기에 네트워크 트래픽 분류 방법론에 발전 가능성을 두고 있다.

네트워크 트래픽 분석 분야는 단순한 패킷 분류를 넘어 여러 세부 업무로 확장되고 있다. 대표적인 예로 응용 서비스 식별, 이상 탐지, 사용자 행위 분석 등이 있다.

이 중 사용자 행위 분석은 특정 응용을 사용한 사용자의 트래픽을 통해 사용자의 행위를 분류하고 분석하는 것이다. 단순히 트래픽의 형태나 클래스에 의존하지 않고, 세션 간의 시간적 상관관계, 패킷 전송의 주기성, 흐름 간 상호작용 등 행위적 특성과 통계적 특징을 추출함으로써 사용자의 목적, 사용 행태, 혹은 비정상적 패턴을 식별할 수 있다.

이러한 행위 분석은 암호화된 환경에서도 페이로드 정보 없이 트래픽의 의미적 변화를 파악할 수 있으며, 단순 분류를 넘어 지능형 네트워크 관리, 보안 위협 탐지 등 다양한 응용 분야로 확장 가능하다는 점에서 그 중요성이 커지고 있다.

본 연구에서는 행위 기반 트래픽 분석의 대상으로 WeChat을 선정하였다. WeChat은 Tencent가 개발한 통합형 모바일 플랫폼으로 채팅, 음성 및 영상 통화, 파일 전송 등 다양한 기능을 하나의 애플리케이션 내에서 제공한다.

본 연구에서는 WeChat 트래픽을 대상으로 사용자의 행위적 특성을 분석하기 위해 세션을 총 네 가지 유형으로 구분하였다.

WeChat 어플리케이션의 경우 mmTLS라는 독자적인 암호화 프로토콜을 사용한다. mmTLS는 전송 구간의 기밀성과 무결성을 보장하기 위해 TLS 기반 구조를 확장한 형태로, 일반 HTTPS 트래픽보다 더 복잡한 세션 구조와 비표준 핸드셰이크 과정을 포함한다. 이러한 특성으로 인해 페이로드 분석이나 세션 복호화가 사실상 불가능하지만, 본 연구에서는 암호화 프로토콜의 내부 동작을 해석하지 않고도 세션 구조와 패킷 통계 정보를 기반으로 행위 식별이 가능함을 제시한다.

WeChat의 주요 기능 중 Chat과 File Transfer는 채팅방 내에서 이루어지는 행위이며, 동일한 세션 내에서 패킷이 발생한다는 특징이 있어 ChatRoom이라는 하나의 클래스로 통합하였으며, Voice, Video, Others 세션은 명확히 구분할 수 있는 행위 패턴을 보여 독립적인 클래스로 정의하였다. 이러한 네 가지 세션 유형을 대상으로 2D CNN 기반 분류 모델을 적용하여 행위적 특징을 학습하고 분류한다. ChatRoom에 존재하는 Chat/File Transfer 클래스는 하나의 세션에 묶여있는 특징이 있어 이를 보완하기 위해 시간 기반의 패킷 번들(Packet Bundle)의 통계적 특징을 추출하여 사용하는 1D CNN로 분류 방법론을 제시한다.

II. 관련 연구

기존 연구에서는 암호화된 트래픽 환경에서 사용자 행위를 구분하기 위한 다양한 시도가 이루어졌다.

[8]은 Sliding Window 기반의 세션 세분화 기법을 제안하였다. 이 방법은 슬라이딩 윈도우의 좌측과 우측 패킷 분포를 비교하여 Kullback-Leibler (KL) Divergence를 계산하고, 이를 통해 트래픽 내 분포적 변화 지점을 탐지함으로써 행위 단위로 데이터를 구분하고자 하였다. 이러한 접근은 원시 데이터에서 직접 행위 구간을 탐색한다는 점에서, 트래픽 내 분포적 차이를 학습하는 새로운 시도를 제시하였다.

[9]는 Netmate 툴을 활용하여 세션 및 플로우 단위에서 44가지의 통계적 특징을 추출한 뒤, 이를 Naïve Bayes, Bayes Net, MLP 등 여섯 가지 머신러닝 분류기에 적용하여 WeChat 트래픽을 분류하였다. 이 연구는 WeChat 트래픽을 대상으로 한 통계적 특징 기반 분류 접근의 타당성과 적용 가능성을 제시하였으며, 추후 비폐이로드 환경에서도 활용 가능한 특징 설계의 기초를 제공하였다.

기존 연구들은 암호화된 트래픽 환경에서 세션 단위로 행위를 구분하기 위한 다양한 시도를 제시하였으나, 여전히 몇 가지 한계가 존재한다.

첫째, WeChat과 같은 통합형 플랫폼에서는 Chat과 File Transfer가 동일한 세션 내에서 혼재되어 발생함에도 불구하고, 두 행위를 구분할 수 있는 명확한 기준이 제시되어 있지 않다. 두 행위가 하나의 세션에 묶여있어 세션이 분류 기준으로 되어있는 방법론이 적용되지 않는다는 한계점이 존재한다.

둘째, 기존 연구 대부분은 패킷의 분포나 시계열적 특성에 기반한 접근으로, 각 행위별 통계적 특징을 충분히 고려하지 못하였다.

이러한 한계를 극복하기 위해서는, WeChat에서 발생하는 행위 단위의 트래픽을 세션 및 패킷 번들 수준에서 정밀하게 분석하고, 이를 효율적으로 분류할 수 있는 새로운 방법론이 요구된다.

따라서 본 연구에서는 WeChat의 주요 행위인 Chat, File Transfer, Voice, Video를 포함한 실제 시나리오 데이터를 수집하여, 각 행위별 트래픽의 통계적·시간적 정보를 프레임워크에서 활용하였다. 이를 바탕으로 행위 구분이 어려운 세션 구조를 보완하고, 암호화된 환경에서도 세션의 행위적 패턴을 정량적으로 인식할 수 있는 새로운 분석 프레임워크를 제안한다.

Ⅲ. 본론

3.1. 데이터셋 수집 및 전처리

표 1. WeChat 데이터셋의 클래스별 세션 및 패킷 수

Table 1. Number of Sessions and Packets per Class in the WeChat Dataset

Class	Session	Packet
ChatRoom	25	7356
Voice	598	43767
Video	516	73719
Other	2628	25902

WeChat 트래픽 데이터는 실제 환경에서 다양한 행위 시나리오를 수행하여 수집하였다. 각 행위는 독립된 시간 구간으로 구성되며, 트래픽 수집 도구를 통해 세션 단위의 pcap 데이터로 저장하였다.

그림 1은 WeChat 트래픽 데이터 수집 과정을 나타낸 것이다. 실험은 실제 사용자 단말 환경에서 총 150초 동안 수행되었으며, WeChat 내의 다양한 기능(Voice, Video, Chat, File Transfer, Others)을 순차적으로 실행하여 트래픽을 캡처하였다. 캡처된 원본 트래픽은 pcap 형식으로 저장되었으며, 이후 세션 단위로 분리하여 분석에 활용하였다.

먼저, 원본 pcap 파일을 5-tuple을 기준으로 세션 단위로 구분하였다. 이후 각 세션은 실제 수행된 기능에 따라 Voice, Video, ChatRoom, Others로 라벨링되었다. 이렇게 라벨링된 세션들은 이후 통합 과정을 거쳐, 모델 입력 형식에 맞게 전처리되었다.

세션 단위 입력을 사용하는 2D CNN 모델의 경우, 각 세션에서 이더넷 헤더를 제거하거나 페이로드 부분만 추출하였다. 그 후 패킷 크기 및 전송 순서를 기반으로 2차원 행렬 형태로 변환하여, 세션의 시계열적 분포 특성이 보존되도록 구성하였다.

1D CNN 모델은 세션 내부의 세부 행위를 구분

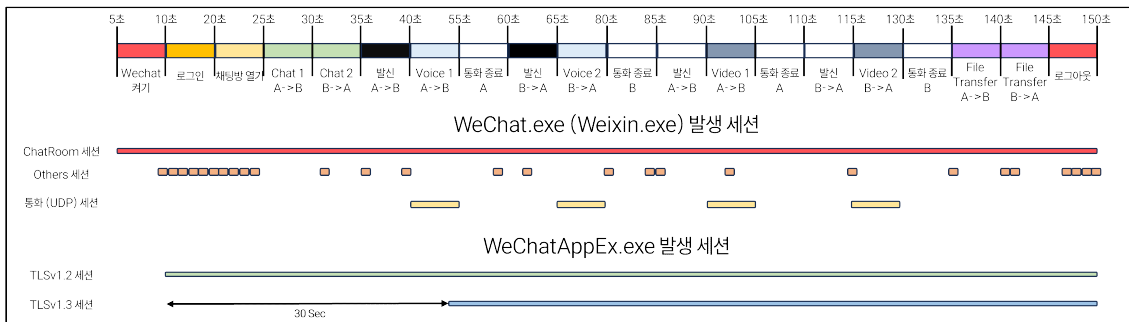


그림 1. WeChat 애플리케이션 사용 시 기능 별 세션 발생 패턴

Fig. 1. Session occurrence pattern by function during WeChat usage

하기 위해 Chatroom 환경에서 수집된 트래픽을 기준으로 하였다. 수집된 시나리오 내 시간 정보를 활용하여, 채팅 메시지 송수신 구간과 파일 전송 구간을 분리하였으며, 이를 각각 독립된 패킷 번들 단위 입력 데이터로 구성하였다. 이러한 방식으로 세션 간 구조적 유사성이 높은 Chat과 File Transfer 구간을 시간적 특성에 따라 세분화함으로써, 모델이 세션 내 행위 단위 패턴을 학습할 수 있도록 전처리를 수행하였다.

3.2. WeChat 발생 트래픽 유형

WeChat 사용 시 WeChat.exe (現Weixin.exe)와 WeChatAppEx.exe 두 가지 프로세스가 발생하는 것을 확인하였다.

WeChat.exe의 경우 크게 네 가지 유형의 세션으로 구분할 수 있었는데, ChatRoom, Voice, Video, mmtls 세션으로 구분하였다. 해당 세션들에 대한 설명은 다음과 같다.

- **ChatRoom** : 채팅방 내에서 발생하는 패킷들을 포함하는 세션이다. 동일한 세션 내에서 Chat과 File Transfer 패킷이 발생한다.
- **Voice** : 음성 통화 시 발생하는 세션으로, 일정한 전송 주기와 양방향 데이터 흐름이 특징이다.
- **Video** : 영상 통화 과정에서 발생하는 세션으로, Voice보다 높은 전송률과 불균형한 데이터 흐름을 보인다.
- **Others** : 인증, 상태 동기화 등 제어 목적의 세션으로, 비교적 짧은 지속 시간과 작은 데이터 단위를 가진다.

3.3. 행위 분석 프레임워크

3.3.1. 세션 단위 2D CNN

세션 단위의 트래픽 특성을 직접 학습하기 위해, 제안된 세션 단위 분류 모델은 각 세션의 패킷 데이터를 2차원 이미지 형태로 변환하여 2차원 합성곱 신경망(2D CNN) 기반으로 학습하도록 설계되었다. 그림 2의 좌측은 세션 단위 분류 과정과 모델의 전체 구조를 보여준다.

먼저, 수집된 pcap 데이터는 세션별로 분할된 후 공통적으로 L2 헤더를 제거하였다. 이후 연구 목적에 따라 L3 및 L4 헤더 제거 여부를 선택적으로 적용하고, 세션 내 첫 번째 패킷부터 최대 784바이트를 추출하였다. 추출된 바이트열은 28×28 크기의 2차원 행렬로 재구성하여 세션 이미지를 생성하였으며, 세션 길이가 부족한 경우에는 Zero-padding을 적용해 입력 크기를 고정하였다. 이 과정을 통해 트래픽의 바이트 시퀀스가 시각적 패턴 형태로 변환되어, CNN이 세션 구조적 특징을 자동으로 학습할 수 있도록 하였다.

제안된 모델은 두 개의 합성곱 계층과 두 개의 완전연결층으로 구성된 기본형 2D CNN 구조를 따른다. 첫 번째 합성곱 계층은 5×5 크기의 필터 32개를 사용하여 입력 이미지($1 \times 28 \times 28$)로부터 지역적 바이트 분포를 추출하고, ReLU 활성화 함수를 통해 비선형성을 부여한다. 이후 2×2 크기의 Max Pooling 연산을 수행하여 특징 맵의 크기를

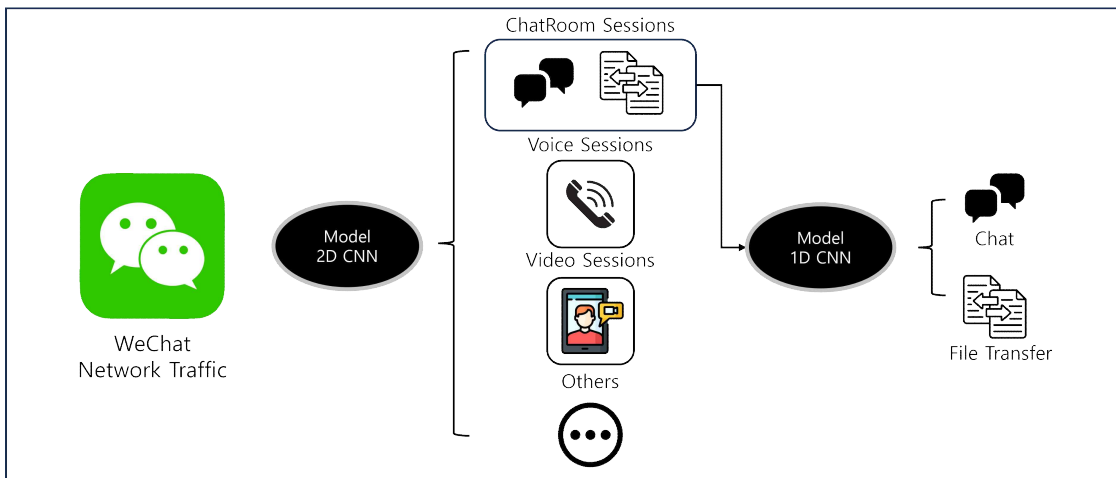


그림 2. 이중 CNN 기반 행위 분석 프레임워크

Fig 2. Dual CNN-based Behavioral Analysis Framework

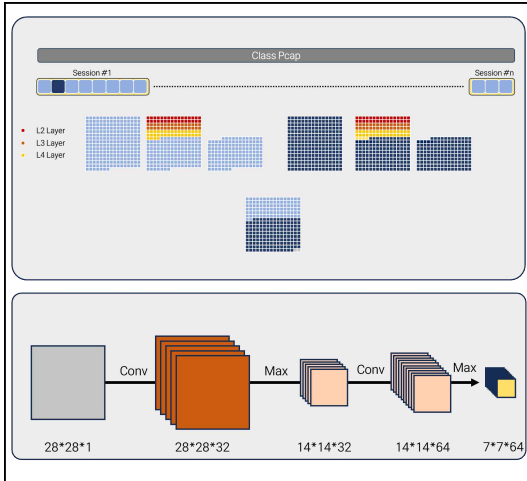


그림 3. 세션 단위 2D CNN 구조도

Fig 3. Session Level 2D CNN Architecture

($32 \times 14 \times 14$)로 축소한다. 두 번째 합성곱 계층은 5×5 크기의 필터 64개를 사용하여 더 추상화된 세션 구조적 특징을 학습하며, 동일한 Max Pooling 연산을 통해 ($64 \times 7 \times 7$)의 출력 특징 맵을 얻는다.

이후 Flatten 과정을 통해 특징 맵을 3,136차원 ($=64 \times 7 \times 7$)의 벡터로 변환하고, 첫 번째 완전연결층 (1,024 뉴런)에서 ReLU 활성화와 0.5의 Dropout을 적용한다. 마지막 완전연결층은 네 개의 뉴런으로 구성되어 Voice, Video, Chat/File Transfer, Others 클래스에 대한 확률을 출력한다. 학습 과정에서는 교차 엔트로피(Cross Entropy) 손실 함수를 사용하고, 학습률 0.001를 적용하여 학습하였다.

제안된 2D CNN 모델은 세션 단위에서 발생하는 평균 패킷 크기, Burst 간격, 전송 방향 비율과 같은 통계적 특성을 별도의 수동 특징 설계 없이 자동으로 학습한다. 또한 세션 이미지를 통해 트래픽의 내부 구조(패킷 분포, 밀도, 주기성 등)를 시각적 패턴으로 인식함으로써, Voice·Video·Others와 같이 구조적 차이가 명확한 트래픽 유형을 분류할 수 있다.

3.3.2. 패킷 번들 단위 1D CNN

본 연구에서는 세션 내부의 시간적 행위 단위를 세밀하게 구분하기 위해 패킷 번들(Packet Bundle) 개념을 도입하였다. 이는 10ms 시간 간격 내에 발생한 연속적인 패킷 묶음을 의미하며, 세션을 시간 기반으로 세분화하여 세션 내 행위 변화(예: 채팅, 파일 전송 등)를 구분하기 위한 분석 단위로 활용된다. 세션 단위 분석만으로는 동일한 세션 내에 존재

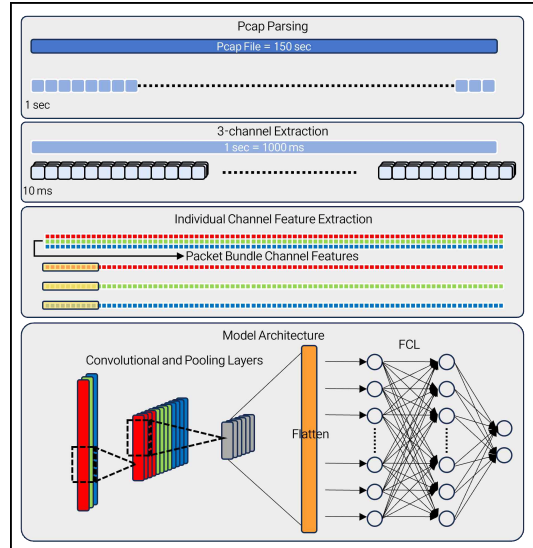


그림 4. 패킷 번들 단위 1D CNN 구조도

Fig 4. Packet Bundle Level 1D CNN Architecture

하는 Chat과 File Transfer 행위의 세밀한 구분이 어려우므로, 세션을 일정 시간 구간으로 분할하고 각 구간을 패킷 번들로 정의하여 1D CNN 기반 시계열 분석을 수행하였다.

수집된 pcap 데이터는 사전에 라벨링된 Chat과 File Transfer 트래픽을 기준으로 1초 단위로 분할하였으며, 각 구간 내에서 10ms 단위의 세부 구간별 (1) 누적 바이트 크기, (2) 패킷 개수, (3) 평균 패킷 간 간격(IAT)을 추출하여 각각 하나의 채널로 구성하였다. 따라서 최종 입력은 (3,L) 형태의 시계열 텐서이며, 이는 세 가지 상이한 통계적 속성을 동시에 표현한다.

제안된 1D CNN은 세 채널의 특성을 개별적으로 학습하기 위해 3개의 독립적인 합성곱 경로(branch)를 둔다. 각 branch는 입력 채널 1개를 받아 Conv1D(1→16, kernel_size=3, stride=1, padding=1), ReLU, MaxPool1D(kernel_size=2) 연산을 수행한다. padding을 1로 설정하여 시간축 길이를 유지한 채, 인접한 약 30ms 범위 내의 지역 패턴을 학습하며, Pooling을 통해 중요 특징만을 추출한다.

이후 세 branch의 출력($3 \times 16 = 48$ 채널)을 채널 축으로 연결(concatenate)하여 통합 시계열 표현을 구성한다. 이후 단계에서는 Conv1D(48→96, kernel_size=3, stride=1, padding=1)과 ReLU, MaxPool1D(kernel_size=2)를 적용하여 채널 간 상관관계와 상위 시계열 패턴을 함께 학습한다.

이러한 두 단계의 합성곱 및 풀링 과정을 거친 출력 텐

서는 Flatten을 통해 1차 벡터로 변환되고, Fully Connected Layer를 통해 Chat과 File Transfer 클래스로 분류된다. 모델은 교차 엔트로피 손실 함수를 사용하고 학습률 0.001를 적용하여 학습하였다.

이 구조는 각 3개의 채널에서 패킷의 개수가 급증하거나, 패킷 번들의 바이트 크기가 증가하거나, 도착 간격이 짧아지는 등의 특징을 독립적으로 추출하고, 이후 각 특징을 결합하여 지역적 특징으로 모델이 행위를 학습하게 한다. 각 채널별 통계적 특징(바이트 크기, 패킷 개수, IAT)을 독립적으로 처리하여 손실 없이 특화된 패턴을 학습하는 동시에, 이후 단계에서 채널 간 상호작용을 통합적으로 고려한다. 그 결과 세션 내부의 미세한 트래픽 변화—예를 들어 메시지 송수신과 파일 업·다운로드—을 정밀하게 구분할 수 있으며, 세션 단위 분석의 한계를 보완하여 전체 행위 인식의 정밀도를 향상시킨다.

IV. 실험 및 결과

본 장에서는 제안한 행위 분석 프레임워크의 성능을 평가하기 위해 WeChat 트래픽을 대상으로 2D CNN과 1D CNN 모델을 각각 학습 및 검증하였다. 모델의 성능 평가는 정확도를 기준으로 수행하였으며, 각 데이터셋은 8:2 비율로 분할하였다.

4.1. 세션 단위 2D CNN 결과

표 2. 세션 단위 2D CNN 분류 정확도 결과

Table 2. Session-level Classification Results

Feature Layer	Accuracy
L3 + L4 + L7	95.22%
L7	93.3%

표 2은 세션 단위 트래픽을 대상으로 2D CNN을 적용한 분류 결과를 나타낸 것이다.

세션 이미지는 28×28 크기로 변환되어 입력되었으며, 모델은 Voice, Video, Others 등 기능별 세션의 구조적 특성을 학습하도록 설계되었다.

그 결과, 2D CNN은 L3와 L4의 헤더 정보를 포함한 경우 약 95%의 분류 정확도를 보였고, 페이로드만 사용하는 경우 93% 수준의 분류 정확도를 보였다.

특히 Voice와 Video 트래픽은 전송 주기, 데이터 방향성, 세션 지속 시간에서 명확한 차이를 보이기 때문에, CNN 기반의 패턴 학습에 유리하게 작용한 것으로 해석된다.

4.2. 패킷 번들 단위 1D CNN 결과

표 3은 Chat/File Transfer 세션의 세부 행위를 구분하기 위해 수행한 1D CNN 기반 분류 결과를 나타낸 것이다. 세션 내 트래픽은 1초 단위의 패킷 번들로 분할되었으며, 각 번들은 송신, 수신, 양방향 트래픽으로 구성된 3채널 입력 구조로 설계되었다.

표 3. 패킷 번들 단위 1D CNN 분류 정확도 및 비교실험

Table 3. Packet-Bundle-level Classification Results

Application	Layers	1D CNN Acc	ActiveTracker Acc
WeChat	L3+L4+L7	97.03	63.25
	L7	95.02	
WhatsApp	L3+L4+L7	90.91	51.05
	L7	93.18	
KakaoTalk	L3+L4+L7	97.67	53.3
	L7	100	

각 채널은 독립적인 Conv1D - MaxPooling 블록을 통해 개별 특징을 추출한 뒤, Flatten 과정을 거쳐 병합(Fusion)된다. 이러한 채널 분리형 구조는 트래픽의 방향성에 따라 서로 다른 전송 패턴을 독립적으로 학습할 수 있도록 하여, 단일 입력 구조 대비 행위 구분에 유리하다. 예를 들어 Chat 세션은 간헐적이고 소규모의 송신 패킷이 주로 발생하는 반면, File Transfer 세션은 일정 시간 동안 대용량의 양방향 데이터가 집중되는 Burst 형태를 보인다. 3채널 구조는 이러한 차이를 각 방향별 특징 공간에서 개별적으로 학습함으로써, 행위 단위의 미세한 전송 패턴을 더욱 정밀하게 포착할 수 있다.

실험 결과, 제안한 1D CNN 모델은 3개의 합성곱 층을 사용할 때 평균 97%의 분류 정확도를 달성하였다. 특히 다채널 입력 구조를 적용하지 않은 단일 채널 모델 대비 약 6~7% 향상된 성능을 보였으며, 이는 제안한 구조가 트래픽의 시간적·방향적 특징을 효과적으로 분리하여 학습함으로써 Chat/File Transfer 행위 인식의 정밀도를 향상시켰음을 의미한다.

또한, 방법론의 일반화 성능을 검증하기 위해 데이터셋 Ablation 실험을 수행하였다. 본 방법론은 WeChat과 유사한 트래픽 발생 경향을 보이는 SNS 어플리케이션에 모두 적용이 가능하다. 따라서, 수집 결과 WeChat과 유사한 트래픽 발생 경향을 보이는 WhatsApp과 KakaoTalk 트래픽에도 동일한 1D CNN 구조를 적용한 결과, 두 어플리케이션 모두 90% 이상의 높은 정확도를 보였다.

V. 결 론

본 연구에서는 암호화된 네트워크 환경에서 사용자의 행위를 효과적으로 식별하기 위해, WeChat 트래픽을 대상으로 한 행위 기반 트래픽 분석 프레임워크를 제안하였다.

WeChat은 메시지, 음성·영상 통화, 파일 전송 등 다양한 기능을 통합적으로 제공하는 대표적인 SNS형 애플리케이션으로, 기능별로 상이한 세션 구조와 전송 패턴을 보인다. 이를 반영하기 위해, 본 연구는 세션 단위(2D CNN)과 패킷 번들 단위(1D CNN)의 이중 구조를 설계하여 행위 구분의 정확도를 향상시켰다. 패킷 번들 단위의 1D CNN은 Chat/File Transfer와 같은 유사 세션 간 미세한 시간적 패턴 차이를 효과적으로 학습하였고, 송신·수신·양방향 3채널 구조를 통해 방향성별 특징을 독립적으로 추출함으로써 행위 분류 성능을 향상시켰다. 추가적으로 WhatsApp과 KakaoTalk 데이터셋에 동일한 모델을 적용한 Ablation 실험 결과, 제안된 구조가 특정 서비스에 종속되지 않고 다양한 메시지 트래픽에 일반화될 수 있음을 확인하였다.

향후 연구에서는 더욱 정교한 행위 인식과 실시간 적용성을 높이기 위한 확장 연구가 요구되고, 패킷 번들이라는 개념을 도입하여 1D CNN이 아닌 Transformer 등 기타 딥러닝 모델의 전처리 방식으로 패킷 정확도 뿐만이 아닌 기타 평가 요소들을 적용하여 모델을 평가하여야 한다.

첫째, WeChat의 파일 전송 과정에서 일부 트래픽 구간이 비동기 채널이나 별도 업로드 절차를 통해 전송되는 것으로 추정되며, 이에 대한 네트워크 계층 수준의 추가 분석이 필요하다.

둘째, 세션 구간 정의를 시간 단위에서 Burst 단위로 확장할 경우 Chat/File Transfer 행위 간의 구분 성능이 향상되는 것으로 확인되었으므로, 향후 연구에서는 이러한 동적 분할 기법을 반영할 예정이다.

셋째, 1D CNN의 3채널 구조를 추가적인 통계 지표(예: 패킷 지연, 전송 비율 등)로 확장함으로써 더 세밀한 행위 구분 및 다중 클래스 분류로 발전시킬 수 있을 것이다.

결론적으로, 본 연구는 암호화된 환경에서도

비페이로드 트래픽의 통계적·시간적 패턴만으로 사용자 행위를 정량적으로 식별할 수 있음을 실험적으로 입증하였다.

향후에는 Burst 기반 세션 정의와 다채널 통계 확장을 포함한 고해상도 행위 분석 모델을 구축하여, 서비스별 트래픽 행위 인식의 범용성과 실시간 분석 가능성을 동시에 확보하는 방향으로 발전시킬 예정이다.

References

- [1] Guang Cheng and Song Wang, "Traffic classification based on port connection pattern," 2011 International Conference on Computer Science and Service System (CSSS), Nanjing, 2011, pp. 914-917, doi: 10.1109/CSSS.2011.5974374.
- [2] M. S. Raza, S. B. A. Kazmi, R. Ali, M. M. Naqvi, H. Fiaz and A. Akram, "High Performance DPI Engine Design for Network Traffic Classification, Metadata Extraction and Data Visualization," 2024 5th International Conference on Advancements in Computational Sciences (ICACS), Lahore, Pakistan, 2024, pp. 1-6, doi: 10.1109/ICACS60934.2024.10473274.
- [3] Moore, Andrew & Zuev, Denis. (2005). Internet traffic classification using Bayesian analysis techniques. *Sigmetrics Performance Evaluation Review - SIGMETRICS*. 33. 50-60. 10.1145/1064212.1064220.
- [4] Lotfollahi, Mohammad & Shirali hossein zade, Ramin & Jafari Siavoshani, Mahdi & Saberian, Mohammadsadegh. (2020). Deep Packet: A Novel Approach For Encrypted Traffic Classification Using Deep Learning. *Soft Computing*. 24. 10.1007/s00500-019-04030-2.
- [5] Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye and Yiqiang Sheng, "Malware traffic classification using convolutional neural network for representation learning," 2017 International Conference on Information Networking (ICOIN), Da Nang, Vietnam, 2017, pp. 712-717, doi: 10.1109/ICOIN.2017.7899588.
- [6] Hwang, R.-H.; Peng, M.-C.; Nguyen, V.-L.; Chang, Y.-L. An LSTM-Based Deep Learning

Approach for Classifying Malicious Traffic at the Packet Level. Appl. Sci. 2019, 9, 3414.

- [7] Lin, Xinjie & Xiong, Gang & Gou, Gaopeng & Li, Zhen & Shi, Junzheng & Yu, Jing. (2022). ET-BERT: A Contextualized Datagram Representation with Pre-training Transformers for Encrypted Traffic Classification. 633-642. 10.1145/3485447.3512217.
- [8] Li, Ding & Li, Wenzhong & Wang, Xiaoliang & Nguyen, Cam-Tu & Lu, Sanglu. (2020). App Trajectory Recognition over Encrypted Internet Traffic based on Deep Neural Network. Computer Networks. 179. 107372. 10.1016/j.comnet.2020.107372.
- [9] Shafiq, Muhammad & Yu, Xiangzhan & Laghari, Asif. (2018). WeChat traffic classification using machine learning algorithms and comparative analysis of datasets. International Journal of Information and Computer Security. 10. 109. 10.1504/IJICS.2018.091467.

김 지 민 (Ji-Min Kim)



2020 ~ 현재 : 고려대학교
컴퓨터융합소프트웨어학과
학사 과정
<관심분야> 네트워크 관리 및
보안, 트래픽 모니터링 및
분석

김 란 아 (Ran-A Kim)



2024 ~ 현재 : 고려대학교
미래모빌리티학과 학사 과
정
<관심분야> 네트워크 관리 및
보안, 트래픽 모니터링 및
분석

장 윤 성 (Yoon-Seong Jang)



2023 : 고려대학교 컴퓨터융
합소프트웨어학과 학사
2023 ~ 현재 : 고려대학교 컴
퓨터정보학과 석/박사 통합
과정
<관심분야> 네트워크 트래
픽 모니터링, SDN

백 의 준 (Ui-Jun Baek)



2018년 : 고려대학교 컴퓨터정
보학과 학사
2025년 : 고려대학교 컴퓨터정
보학과 박사
2025년 ~ 현재 : 한국과학기술
정보연구원(KISTI) 선임연구
원
<관심분야> 블록체인 거래 모
니터링, 네트워크 관리 및 보안, 트래픽 모니터링
및 분석

김 명 섭(Myung-Sup Kim)



1998년 : 포항공과대학교 전자
계산학과 학사
2000년 : 포항공과대학교 전자
계산학과 석사
2004년 : 포항공과대학교 전자
계산학과 박사
2006년 : Dept. of ECS, Univ
of Toronto Canada

2006년 ~ 현재 : 고려대학교 컴퓨터정보학과 교수
<관심분야> 네트워크 관리 및 보안, 트래픽 모
니터링 및 분석, 단일미디어 네트워크

원격제어차량 텔레오퍼레이션 및 재해복구를 위한 올인원 휴대용 특화망 장치 구현

최 태 상*, 윤 상 식°, 이 준 경*,

Implementation of All-in-one Portable Private 5G Network for Remote Control Vehicle Teleoperation & Disaster Recovery

Taasang Choi*, Sangsik Yoon°, Joonkyung Lee*

요 약

본 논문에서는 지구 이상 기후 변화로 인한 지진, 해일, 폭우 등 자연재해 하에서 주요 시설의 기 설치되어 운영 중인 유·무선 통신망이 파괴되었을 때나 원격자율제어차량의 텔레오퍼레이션 환경에서, 기존 통신망과는 별개로 통신 비전문가인 산업시설의 운용자가 수십분 이내에 구축할 수 있는 신속성, 편리, 휴대성을 모두 갖춘 유·무선 올인원 휴대용 5G 특화망 통신 장치 및 멀티모달/데이터 서비스 자율구축 기술을 제안한다. 또한 연구 결과로 개발된 5G 특화망 연구 시제품의 가능 검증 및 시험 결과를 소개한다.

Key Words : All-in-one Private 5G kit, Potable Disaster Recovery Communication, Private 5G Network, Teleoperation

ABSTRACT

This paper introduces our research results of an all-in-one portable private 5G communications system for remote control vehicle (RCA) teleoperation and national important industrial facilities like nuclear power plant under extreme natural disaster due to global abnormal climate change such as earthquake, tsunami and heavy rain incident. In such environment, RCA teleoperation engineers or various industry building operational staffs which do not have technical knowledge of network provisioning and operations must deploy the disaster recovery network. Thus the proposed research prototype provides capabilities to provision a private 5G network autonomically and enable multi-modal (RCA sensor data, control messages, voice/video, etc.) and data services. It also describes our Private 5G disaster recovery communication kit developed based on the proposed core technology and architecture as well as the results of system verification testing.

※ 이 논문은 과학기술정보통신부와 정보통신기획평가원의 지원을 받아 수행된 “극한재난 구조용 지능형 상황인지 초실감 원격협업 및 이동형 자율 네트워킹 핵심기술 개발” 과제 (연구개발과제번호: RS-2023-00263029) 및 산업통상자원부와 한국산업기술진흥원의 지원을 받아 수행된 “AI 기반 다이나믹 B5G/6G 네트워킹 슬라이싱 기술을 이용한 원격제어차량 기술 개발” 과제(연구개발과제번호: P0028063) 연구임.

• First Author : Electronics and Telecommunications Research Institute & (주)모비젠, choits@etri.re.kr & choits@mobigen.co.kr

° Corresponding Author : Electronics and Telecommunications Research Institute, ssyoon90@etri.re.kr

* Author : Electronics and Telecommunications Research Institute, leejk@etri.re.kr

논문번호 : KNOM2025-02-010, Received November 13, 2025; Revised November 28, 2025; Accepted December 21, 2025

I. 서 론

최근 글로벌 기후 변화에 따른 홍수, 지진, 해일, 산불 등의 극한 재난 상황이 전 세계적으로 빈번히 발생하는 상황에서는 최초 재난 발생 시점에서부터 인명 구조가 필요한 재난 골드타임 내에서는 현장 상황에서 절실히 필요한 기본 통신 시스템이 마비되어 통신 서비스가 단절되는 문제가 발생할 수 있다. 이러한 최악의 상황에서는 재난 구조 현장 상황 실과 구조 대원 뿐만 아니라 피구조자와의 통신 부재로 인한 골드타임 내에서 원활한 인명 구조가 이루어지지 못하게 되어 예상치 못한 엄청난 인명 구조 손실이 추가되는 문제가 발생할 수 있다.

따라서 이러한 극한 재난 상황 시에 기본 통신이 단절되는 상황을 대비하여 긴급하게 현장에 이동하여 무정전으로 전원만 켜고 케이블만 연결하고 휴대단말만 켜면 통신이 이루어지는 올인원 휴대용 비상통신 키트 개발이 필요하다.

또한 대형 교통사고 발생시나 일시적인 극심한 교통 체증 등과 같은 비상 상황 발생시 원격자율제어 차량의 제어를 위해 현장에서 긴급 재해복구망 구축·복구 및 신속한 멀티모달·데이터 통신서비스 기능 제공은 추가 피해를 최소화하고 국가 주요 산업시설의 안전을 유지하는데 필수적이다.

따라서 본 연구에서는 다양한 재해 조건이나 원격자율제어 차량 통제 환경에서 통신시설이 와해된 기존 통신망과는 별개로 신속하게 비전문가들도 쉽게 구축할 수 있는 신속성, 편리성, 및 휴대성을 모두 갖춘 유·무선 올인원 휴대용 특화망 비상통신망 장치 및 응급서비스 자율구축 기술을 개발하였다. 본 연구개발 결과물은 정상 운영시에는 정기적인 기능확인을 위한 기본 기능을 유지하다가 재난시, 다음의 핵심 기술을 바탕으로 재해 복구 네트워크를 신속하고 용이하게 구축함으로써 재해 시설 운영자들 및 재해 복구 로봇/드론과 같은 복구 도구들과 원격자율제어차량 및 제어 센터 간에 긴급 통신 서비스를 제공함을 목적으로 한다.

- 포터블 배터리 기반 유·무선 올인원 휴대형 특화망 통신스위치, Private 5G Core/RAN, SDR USRP, 서버 하드웨어 기술
- 원격제어차량 텔레오퍼레이션 및 재난 복구용 유·무선 멀티모달/데이터 통신서비스 자율 제어, 동

적 네트워크 슬라이스 관리 기술

- 또한 심각한 재난상황에 대비하여, 재난 발생 이후 신속하게 재해 복구망을 처음부터 자동 설치 및 구축이 가능하도록 기능을 구현하였다.

상기 목적을 고려하여 본 연구개발에서는 핵심 기술로 SDN (Software Defined Networking)[1], SON (Self-Organizing Network)[2], SIP (Session Initiation Protocol)[3] 기술 기반 자율 운영, 중앙집중형 자율 네트워크 관리, 품질 보장 텔레오퍼레이션을 위한 동적 네트워크 슬라이싱, 동시에 전원과 배터리 지원이 가능한 유·무선 통신 코어 기술을 연구 개발하였다.

본 논문에서는 먼저 해외 재해 복구용 유·무선 통신서비스 최신 기술에 대한 차이점을 분석하고, 국내 소방청 및 자율차량 원격제어 환경의 다양한 관련 요구사항을 추가적으로 분석하여 실제 적용 가능한 원격 자율 제어 차량 텔레오퍼레이션 및 극한 자연재해 복구 환경에 적합한 비상 통신망의 최적 구조를 설계하고 구현하기에 적합한 핵심 요소 기술들을 맞춤형으로 개발하여 박스형 일체형 연구 시제품(PoC)을 개발하였다. 본 연구시제품은 한국자동차연구원의 원격자율차량 시험 테스트베드와 운영 중인 원자력 발전소 및 유럽 여러 국가의 소방청에서 운영 중인 재난 모의 현장에 시험 적용하여 현장 적합성을 검증하였다. 본 연구에서는 비상통신망의 제공 서비스로 원격 제어 차량 텔레오퍼레이션에서 발생하는 비디오/오디오 스트리밍 데이터, 각종 차량 센서에서 발생하는 데이터, 차량 제어용 메시지 및 재난 복구에 필요한 유무선 음성 통신과 콘텐츠 스트리밍, 증강/가상현실 스마트안경 및 재난 복구 로봇에서 생성되는 다양한 데이터를 포함한 데이터 통신을 완료하였다.

2절에서는 비상통신 관련 연구에 대한 분석 결과를 제공한다. 3절에는 제안된 원격자율제어차량 텔레오퍼레이션 및 재해 복구용 올인원 휴대형 비상통신망 장치 및 서비스의 구조 및 핵심 기능을 설명한다. 연구시제품 구현, 기능 검증 및 시험 분석 결과는 4절에서 서술한다. 마지막으로 5절은 논문 결과 및 향후 연구에 대해 간략히 소개한다.

주파수와 이기종 단말 기술에 의한 상호연동의 어려움, 광역의 커버리지를 위한 복잡한 무선통신 기술을 주요 특징으로 한다. 3GPP는 이 단점을 보완한 PS-LTE 멀티미디어 동여상 서비스 표준을 정의하였다. 주요 특징으로는 단말들간 직접통신, 그룹통신, 직접 무선통신, 핵심망 연결 붕괴시 독립망 구축 기술 등 비상통신망에 필수 기능을 추가하였다. 2014년에 한국 상향 맞춤형 PS-LTE를 채택하고 2019년에는 국가재난안전통신망을 구축 완료하고 서비스를 중에 있다.

상기에 살펴본 관련 연구들은 각각 적용 대상이 되는 별도의 자체 환경들을 가정하고 있기 때문에 원전과 같이 무선 통신이 차단되는 두꺼운 콘크리트 건물 내벽 및 방화문과 같은 제한조건이 많거나 혹은 재난 상황시에 적용하기에는 다양한 제약점이 있다. 원격제어 환경에서도 동적인 환경 적응형 고품질의 통신서비스가 요구되나 이를 충족하지 못하는 상황이며, 특히 본 연구에서 가정하는 환경 즉 모든 통신 설비가 파괴된 재해 상황에서 신속하게 최소한의 통신 환경을 자동으로 구축하고 통신 서비스를 제공하는 기술로는 많은 변경 없이 적용하기에는 한계가 있다. 따라서 본 논문에서는 관련 핵심 요소 기술들을 기반으로 텔레오퍼레이션 및 다양한 재난 환경의 복구에 적합한 올인원 휴대형 특화망 장치 및 자율 제어/관리 기술을 제안한다.

III. 시스템 요구사항 및 구조

3.1. All-in-One Portable 비상통신망 Kit 구조 요구사항

본 절에서는 제안 시스템의 기능 요구사항을 간략히 정리하였다.

- B5G/6G 환경에서 다이나믹 네트워크 슬라이싱(Dynamic Network Slicing) 기술을 적용하여 차량 원격제어(teleoperation) 서비스의 실시간성과 신뢰성을 향상할 수 있는가?
- 신속한 구조 작업을 위해 재난현장통신시스템을 최대 30분 이내 구축할 수 있는가?
- 긴급 상황에서 신속한 구조작업을 위해 FR(First Responder) 한 사람으로 비상통신망을 구축할 수

있는가?

- FRs의 통신 셋업 시간과 복잡성을 줄이도록 손쉬운 통신망 구축과 구성 프로세스를 제공하는가?
- 이동 휴대성과 운송 편의성을 위해 2개의 손잡이가방으로 되어 있는가?
- 장시간 작업을 위해 최대 14시간 배터리-전원을 제공하는가?
- 이중화 및 백업 솔루션을 제공하는가?
- 자율 네트워크 구성 및 장애 발생 시 자동복구가 가능한가?
- 음성/영상/이미지/텍스트/센서/GIS 및 재난사고 데이터의 상호작용을 보장하는가?
- 협준한 지형을 포함하여 최대 5km2의 재난지역에 비상통신을 제공할 수 있는가?
- 구조장비들(예: OWL드론, 4족보행로봇개, SNAKE로봇, 웨어러블 측위발찌/생체슈트 등)이 여러 네트워크 상에서 채널 할당과 필요 대역폭, 우선 순위 서비스, 라우팅/저장/처리자동화/통신자원 등 네트워크 가용성을 보장하는가?
- +45°C ~ -30°C의 온도 범위 내에서, 강풍/먼지/물기없음에도 내구성/신뢰성을 보장하는가?
- 방수 및 습한 환경에서 작동 가능한가?
- 대기오염, 먼지, 연기, 재, 모래 등 가혹한 재난환경에도 작동 가능한가?
- 100% 습도에서 긴급메시지 우선순위 송·수신, HQ/Commander/FRs/Robots 그룹 분류 통신 가능한가?

다음은 All-in-One Portable 비상통신 장비의 주요 KPI 항목이다.

- 최종목표 비상통신망 Plug&Play Set-up 시간: 30분 이내
- 최대 동시 접속 (VoIP)사용자 수: 200명
- 통신장비 내 배터리만으로 작동 지속시간: 8시간
- 무선 주파수 도달범위: 1,000m2
- Download / Upload 트래픽 성능: 100M/20Mbps

3.2. 비상통신망 Kit 주요 기능 요소

본 절에서는 본 고에서 제안한 All-in-One 비상통신망 시스템 개발에 필요한 핵심 기술을 설명한다.

- One-man 휴대형 장비

네트워크 구축에 전문 지식이 없는 재난 현장 운전

원 한 명이 들고서 최적의 재난 현장으로 쉽게 이동 가능하도록 장비 형태가 단순하고 손잡이나 밴드가 제공되어야 하며, 최대 20kg 미만의 무게로 제작이 되어 현장 운전원 한 명 만으로도 쉽게 구축 가능하도록 설계 제작되었다.

- Power-on/Cable Plug & Play 자동 통신망 구축 통신 비전문가인 재난 현장 운전원들이 매뉴얼에 따라 비상통신망 장비를 지정된 장소로 이동한 후, 뚜껑을 열고 전원 버튼만 on 시키면 자동으로 Private 5G 망이 구축되도록 설계 제작되었다.

- 오픈소스 Private 5G Core/RAN 제안한 비상통신망은 비상시임에도 불구하고 원활한 구조 활동 지원을 위해서 실시간 동영상 화상 통신 및 초실감 영상 전송과 같은 고품질 서비스를 제공하기 위해서는 이들이 요구하는 고대역폭 데이터 전송 및 저지연 데이터 통신 서비스 제공이 가능해야 한다. 이를 위해서 본 제안에서는 오픈소스인 OAI(OpenAirInterface) 5G Core Network와 RAN gNB를 하나의 One-Box 형태로 설계 제작하였으며 좁은 공간의 한계를 해결하기 위해서 초소형 미니 PC 기반으로 내부 서버들을 구성함으로써 공간과 무게, 성능 문제를 동시에 해결하였다.

- 다중 5G 네트워크 슬라이스 서비스 본 비상통신망 장비는 다양한 서비스에 대하여 서로 다른 QoS 제공을 위해서 5G NSSF(Network Slice Selection Function) 서비스를 이용하여 멀티 네트워크 슬라이스 서비스 Private 5G 망을 구축하여 최적의 네트워크 슬라이스 서비스를 제공가능하도록 구축 완료되었다.

- 무전원 Battery powered 극한 재난 상황에서 가장 기본적인 요구사항은 대부분 초기 골든타임 내에서는 전원 공급이 불가능하다는 상황을 가정하고 대응해야 한다는 사실이다. 이 경우 유일한 해결 방안은 배터리를 적용하는 방안이다. 그러나 기존의 네트워크 장비는 전원이 정상 공급된다는 기본 가정 하에 개발되었다. 배터리는 짧은 시간 동안 대체용으로 고려된 경우는 있지만 재난시 골든타임 시간 동안 최소 8시간 이상의 운영 요구사항 설계가 기본으로 필요하다.

- 로컬 데이터 네트워크 및 멀티 WAN 인터넷 인터페이스 제안하는 비상통신망 키트는 전원만 켜면 자동으로 구축되어 작동하지만 필요에 따라서 내부 로컬 데이터 망 서버와의 연결을 위해서 내부망 연결 인터페이스 서비스 또한 제공되어야 한다. 이는 간단히 내부 서버를 위한 데이터 케이블을 서로 연결만 함으로써 자동으로 구축 완료하도록 설계되었다. 또한 외부 WAN 인터넷 망과의 연결을 위해서 UTP 케이블과 WiFi 인터페이스를 동시에 이중화하여 (Active/Stand-by) 5G 특화망의 높은 서비스 가용성을 보장할 수 있는 것이 특징이다.

- SDR(Software Defined Radio) 기능을 포함한 USRP 장비를 이용한 멀티 Band 무선 주파수 설정 제공

본 고에서 제안하는 비상통신망 장비의 활용성을 쉽게 확장하고 여러 재난 현장에서 빠르게 검증하기 위해서, gNB에 SDR(Soft Defined Radio) 주파수 신호 전송/수신이 가능한 범용 USRP(Universal Software Radio Peripheral) 장비를 이용하여 구축 제작함으로써 다양한 5G 주파수 대역폭(예: N41, N77, N78, N79 등) 설정하여 운영 가능한 것 또한 큰 장점이다. 마지막으로 짧은 무선 신호를 확장하기 위해서 유선 IP NAT 라우터를 이용하여 중앙 비상통신망 노드로부터 쉽게 확장 구축 가능하도록 설계되었으며, 뿐만 아니라 신호가 도달하지 않는 빌딩 내부에서도 통신이 가능하도록 5G NAT 라우터를 활용하여 추가 확장 가능하도록 설계된 것도 특징이다. 이러한 NAT 라우터의 인터넷 포트나 WiFi 서비스를 이용하여 5G 모뎀이 들어 있는 단말만이 아니라 일반 통신 장비들도 간단히 비상통신망에 접속하여 5G 통신 서비스를 제공받을 수 있도록 확장 설계되었다.

3.3. All-in-One Portable 비상통신망 Kit 시스템 구조

본 절에서는 3.1절의 사용자/시스템 요구사항 기반의 극한 재난 상황에서 활용 가능한 All-in-One Private 5G 비상통신망 구조에 대하여 서술한다.

그림 2는 All-in-One 비상통신망 Kit 노드의 외부 인터페이스 구조 설계도를 보여주고 있다.

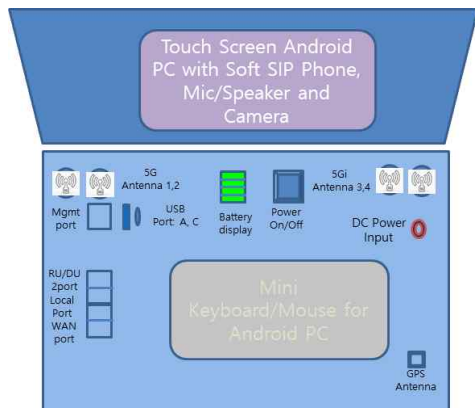


그림 2. All-in-One 비상통신망 Kit 노드의 외부 인터페이스 구조

Fig. 2. External I/F of All-in-One Emergency Communication Kit

비상통신망 KIT 노드는 상단에 기본 음성 서비스 제공 및 내부 시스템 관리/모니터링을 위한 Tap 장비가 장착되고, 노드 하단에는 전원 버튼, 전원 충전 및 상시 전원 입력을 위한 전원 입력 포트, 5G 무선 주파수 신호를 수신/송신할 수 있는 4개의 안테나 연결 포트, 그리고 좌측 하단에는 외부 WAN 연결을 위한 GbE 포트 1개, 내부망 연결을 위한 2개의 이더넷 포트, 그리고 5G 망 추가 확장을 위한 RU/DU 연결 포트 한 개로 각각 구성되어 있다. 마지막으로 5G Core Network 노드에 연결된 관리 이더넷 포트와 USB 포트 2개도 설계되었다.

다음은 All-in-One 5G 비상통신망 노드의 외부 인터페이스에 대한 상세 스펙 내용이다.

- SIP Soft Phone with Receiver/Dial Touch Pad
- Screen, mic/speaker and camera
- RU/DU Data Gigabit Ethernet port (RJ45 UTP):
2 GbE port
- Local Host Edge Gigabit Ethernet port (RJ45 UTP): 1 GbE port
- WAN Internet Gigabit Ethernet port (RJ45 UTP): 1 GbE port
- Management port : 1 GbE (RJ45 UTP)、1 USB 3.0 port
- 4 Wireless 5G Radio Antenna [SDR(B210) TX/RX Radio - SMA extension cable - SMA Female Adapter - External 5G antenna]
- 5-Volt Active GPS Antenna

- Display status/remains of battery
- DC input port for charging battery
- Power On/Off button switch (status light)
- Case Size : L411 * W321 * 165H [Wireline spine node size]
- NANUK 933 Size: 19.9 in x 16.1 in x 10.1 in (505 mm x 409 mm x 257 mm) [Private 5G Case]

다음 그림 3은 외부 인터페이스 디자인 설계 초기 도면을 보여주고 있다. 최종 설계 도면과는 실제 차이가 있으나 전체적인 규격은 동일하게 설계 제작되었다.

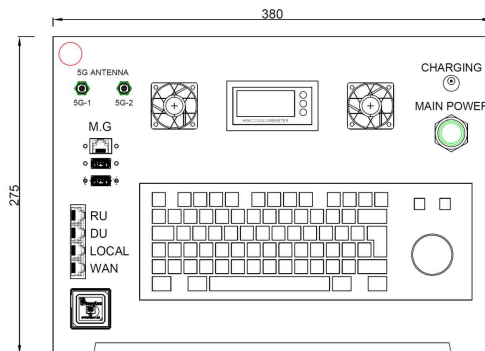


그림 3. All-in-One 비상통신망 외부 인터페이스 상세 설계
도면

Fig. 3. Design of All-in-One Exmergency Communications External I/F

그림 4는 All-in-One 비상통신망 Kit 노드의 내부 장비와 연결 구성도를 보여주고 있다.

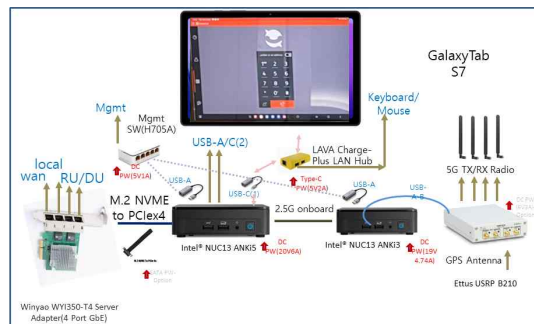


그림 4. All-in-One 비상통신망 Kit 내부 장비 연결 구성도
Fig. 4. Internal Connectivity of All-in-One Emergency Communication Kit

비상통신 노드의 내부 구조는 전체적으로 Intel NUC 기반 미니 PC 2개로 구성되어 있으며 최측은 OAI 5G Core Network을 위해서 NUC13 ANKi5

서버 위에 Ubuntu 22.04 OS를 설치하였고 멀티 기가비트 이더넷 포트 제공을 위하여 M.2 NVME to PCIe4 변환 케이블을 이용하여 PCI NIC 카드와 직접 연결하였다. 다음 우측의 PC는 OAI gNB 서비스를 위한 것으로 NUC13 ANKi3 서버 위에 마찬가지로 Ubuntu 22.04 OS를 설치하였으며 SDR 무선 주파수 송수신을 위한 Ettus B210 USRP 장비를 USB로 직접 연결하였으며 외부로 무선 주파수 안테나와 GPS 케이블을 연결하도록 설계 구축하였다.

IV. 시스템 구현 및 성능 검증

4.1. 비상통신 Kit 시제품 제작 구현

본 절에서는 제안된 All-in-One 비상통신망 Kit 시작품을 제작하고 기본 인터넷 연결 서비스 기능 시험과 실제 화상회의 서비스 시연 기능 검증 결과를 소개한다. 그림 5는 실제 제작한 All-in-One 비



그림 5. All-in-One 비상통신망 Kit 노드 시작품
Fig. 5. All-in-One Emergency Communication Kit PoC System

상통신 Kit 노드 화면이다.

비상통신망 Kit 노드는 개방을 하지 않을 경우 방수 처리가 되는 단단한 케이스(NANUK 933)를 이용하였으며 뚜껑(lid)을 오픈하면 크리스탈 판넬을 이용하여 내부가 보이도록 제작하였다. 하단의 구조는 전체적으로 3단 구조도 되어 있으며 맨 하단에는 대용량 배터리가 내장되었으며 중간 층은 전원 제어/송신부와 USRP가 들어 있고 맨 위층에는 NUC PC 2대와 이터넷 스위치가 장착되도록 제작하였다.

이렇게 제작된 비상통신망 노드에 각각 OAI 5G Core Network(v2.1.0), gNB openairinterface5g (2025.w20) 버전을 각각 설치하여 구축 제작 완료하였다.

기본 Private 5G 망에 추가로 네트워크 슬라이스 서비스 환경 구축을 위하여 3개의 슬라이스를 그림 6과 같이 구축하였다. NS 1은 SST=128, SD=0x08이며 ueransim gNB를 이용하여 연결되었으며 NS 2는 SST=1, SD=0xFFFF로 oai-gnb 시뮬레이터로 연결되었으며 마지막 NS 3은 SST=130, SD=0x82로 gnbnsim 시뮬레이터에 연결되도록 구축 완료하였다.

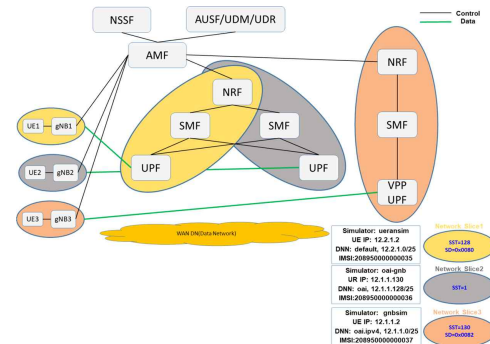


그림 6. Private 5G 망에 구축된 네트워크 슬라이싱 망 구성도

Fig. 6. Network Slicing Architecture of Private 5G Network

4.2. 비상통신 Kit 시제품 성능 시험 및 검증

상기에서 제작된 All-in-One 비상통신망 Kit 노드의 배터리를 제거한 무게는 약 12kg으로서 내장 배터리를 포함한 경우는 약 15kg 무게로 한 사람이 들고 이동하기에 충분한 무게로 측정되었다. 본 논문에서 제작한 비상통신망 Kit의 기본 기능 시험을 위하여 상용 5G 스마트폰에 Private 5G 프로그래머블 USIM (Sysmocom sysmoISIM-SJ2)을 장착하여

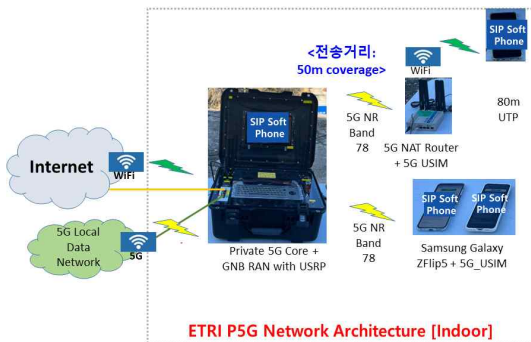
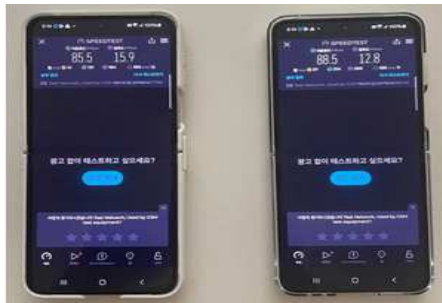


그림 7. Private 5G 비상통신망 구조도 구축 예제
Fig. 7. Example of Private 5G Emergency Communication Network

그림 7의 구조와 같이 인터넷에 연결되는 것을 확인하였다.

5G 스마트폰은 정상적으로 비상통신망에 연결이 되었으며 외부 데이터 네트워크로 상용 인터넷 망에 연동이 되었다. 5G NAT 라우터를 사용하거나 직접 5G 무선으로 연결한 인터넷 성능 속도는 그림 8과 같이 두 개의 스마트폰을 동시에 수행한 결과 DL/UL=85.5/15.9Mbps, 41ms 지연, 12ms 지터의 결과를 보여주었다. 다만 내부 USRP 장비의 무선 신호가 Lab 테스트 수준으로 낮아서 전송거리는 50m 이내로 제한된 결과가 나왔다. 이는 이후에 야외용 고출력 RU를 포함한 gNB 성능을 추가 확장함으로써 해결 가능할 것으로 예상된다.



Speedtest 측정 결과(동시):
Download/Upload=85.5/15.9Mbps

그림 8. 스마트폰 2개 동시 인터넷 성능 측정 결과 화면
Fig. 8. Internet performance testing results of the two Smartphone

고품질 화상회의 서비스 검증 (5G 네트워크 주파수 범위 확장 및 Video conferencing 서비스 현장 기능 검증)

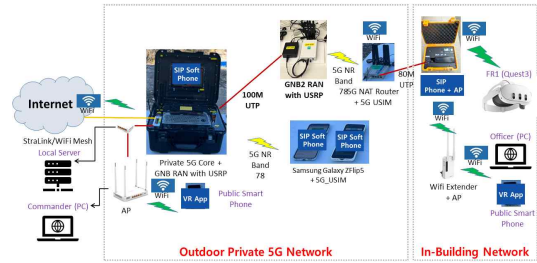


그림 9. Private 5G 비상통신망 추가 확장 구축 구조도 예제
Fig. 9. Extended Private 5G Emergency Communication Network

그림 9는 기존 5G 네트워크 주파수 커버리지 확장을 위한 추가 구조 확장과 Video conferencing 서비스 제공을 위한 WiFi 확장 설계 구조를 보여주고 있으며, 기본 Private 5G 망에 추가 구축 완료하였다. 먼저 거리 확장을 위하여 중앙 5G Core Network 노드에 이더넷 케이블이 연결된 gNB를 추가로 구축하였으며 여기에 5G NAT 라우터를 연결하고 다시 이더넷 케이블을 이용하여 유선 AP를 연결한 후에 마지막으로 WiFi Extender를 이용하여 무선 서비스 범위를 넓게 추가 확장하였다. 이렇게 함으로써 최대 400~500m 범위까지 확장되는 것을 확인하였으며 중앙 HQ(Head Quarter)와 서비스 지연이나 끊김이 없이 매우 만족스러운 고품질의 화상회의 서비스가 원활히 제공되는 것을 확인하였다. 각각의 확장 위치에서의 상세 성능 측정 값은 다음과 같다. 먼저 SIP Soft Phone에서의 인터넷 트래픽 처리 성능은 DL/UL=99.7/18.8Mbps이고, 5G NAT Router의 Wifi에 연결된 경우는 DL/UL=94.3/12.9Mbps, 유선 확장 연결된 AP의 Wifi에 연결된 경우는 DL/UL=69.8/11.6Mbps, 마지막으로 Wi-Fi Extender에 연결된 경우는 DL/UL=63.1/10.7Mbps으로 측정되었다.

V. 결론 및 향후 연구

본 논문에서는 지구 이상 기후 변화로 인한 자연 재해로 파괴된 통신 기능이나 대형 교통사고 발생 시나 일시적인 극심한 교통 체증 등과 같은 비상 상황 발생시 원격자율제어 차량의 제어를 위해 현장에서 긴급 비상통신망 구축·복구 및 신속한 멀티모달·데이터 통신서비스를 네트워크 구급상자를 이동 일체형 박스형 Kit로 제작 및 상시 비치를 통해 재재 발생시 통신 비전문가들에 의해 재난 장소로

이동해 전원만 켜면 유무선 통신이 가능케 하는 재해 복구 통신망이 자율적으로 구축될 수 있는 All-in-One Private 5G 통신 장치 기능 요구사항 도출 및 분석, 구조 설계 및 개발과 연구시제품 기능 검증 및 시험에 대해 기술하였다. 본 제안 기술은 본문에서 예시된 바와 같이 다양한 재해 현장에서 실용적으로 적용될 수 있는 실예를 제시하였는데 의미가 있다고 판단된다.

본 제안 시스템은 실험실과 제한된 현장에서 기술 검증을 위해 제작된 PoC 시스템 기술로 다양한 환경에 노출된 실제 현장 상황에서 적용될 수 있기 위해서는 수백 미터 ~ 수킬로를 지원할 수 있는 RU coverage와 UE ~ CN ~ DN 종단간의 동적 슬라이싱 기능을 지원하기 위한 추가 연구 및 개발이 필요하다. 또한 5G에서 6G로 진화하고 있는 통신 환경을 고려하여 지원할 수 있어야 한다. 이를 위해 네트워크 관점에서의 주요 연구 이슈들은 향후 추가 연구를 통해서 해결 방안을 계속해서 제시할 예정이다.

References

- [1] Grag Cross, Open Networking Forum SDN standards, Jan. 2017. Accessed Nov. 15, 2017. <https://www.opennetworking.org/software-defined-standards/overview>
- [2] 3GPP SON, <https://www.3gpp.org/technologies/keywords-acronyms/105-son>
- [3] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [4] Massimo Tornatore, et. al. (2017), "A Survey on Network Resiliency Methodologies against Weather-based Disruptions", EU-funded RECODIS project
- [5] Science Director, openflow protocol overview, <https://www.sciencedirect.com/topics/computer-science/openflow>
- [6] Jürgen Quittek et. al., "Network Functions Virtualisation (NFV) - Management and Orchestration V.1.1.1", ETSI NFV ISG, Dec. 2014.
- [7] Pedro Neves. et. al. (2017), "Future mode of operations for 5G - The SELFNET approach enabled by SDN/NFV", Computer Standards & Interfaces 54 (2017) 229-8211;246
- [8] ETSI Standard, TERrestrial Trunked RAdio (TETRA), <https://www.etsi.org/technologies/tetra>
- [9] Korea Digital TRS (TETRA) Basic Standard, TTAS.KO-09-0037, April 2005
- [10] 3GPP Public Safety, <https://www.3gpp.org/news-events/3gpp-news/1455-Public-Safety>
- [11] Open Virtual Switch, <https://www.openvswitch.org/>
- [12] Jagadeesan, N.A., Krishnamachari, B: Software-defined networking paradigms in wireless networks: a survey. ACM Comput. Surv. 47(2), 11 p. (2014). Article 27, doi: 10.1145/2655690
- [13] Grag Cross, ONOS open source project, Jan. 2017. Accessed Nov. 15, 2017., <https://www.opennetworking.org/platforms/onos/>.

최 태 상(Taesang Choi)



1988~1991, 컴퓨터통신 석사
1991~1995, 컴퓨터통신 박사
1996~현재, ETRI 책임연구원
<관심분야> SDN/NFV/SON,
네트워크관리, Traffic 관리,
자율관리

윤 상 식(Sangsik Yoon)



1997~1999, 컴퓨터네트워킹 석
사
1999~2003, 컴퓨터네트워킹 박
사
2003~현재, ETRI 책임연구원
2023~현재, (주)이엔소프트 책
임

<관심분야> SDN/NFV, 통신망관리, Traffic 제어,
QoS 보장, 자율관리, Private 5G/6G, Quantum
Network Simulation

이 준 경 (Joonkyung Lee)



1980~1985, 물리학(전산학) 학
사
1985~1987, 컴퓨터학 석사
1991~1997, 컴퓨터학 박사
1987~현재, ETRI (전)실장/책임
연구원, (현)연구전문위원

<관심분야> SDN/NFV/SON/DPI & (B)5G/Edge통
신, AR/XR, xBMI/Wearable, DRS

End - Fog - Cloud 환경에서 데이터 무결성과 접근 제어를 보장하는 데이터 프로비넌스 시스템

배 태 모*, 방 지 원*, 황 향 희**, 최 미 정°

A Data Provenance System Ensuring Data Integrity and Access Control in End - Fog - Cloud Environments

Taemo Bae*, Jiwon Bang*, Hyang-Hee Hwang**, Mi-Jung Choi°

요 약

IoT(Internet of Things)는 제한된 연산과 저장 자원을 보완하기 위해 클라우드 컴퓨팅과 결합하여 발전해왔다. 그러나 물리적 거리로 인한 지연과 실시간 처리 한계는 여전히 문제로 제시된다. 이를 해결하기 위해 Fog 컴퓨팅이 도입되면서 End-Fog-Cloud 구조로 확장되었다. 하지만 기존 보안 기법은 IoT 환경의 제약으로 적용이 어렵고, CSP(Cloud Service Provider)에 대한 신뢰 문제까지 존재해 데이터 무결성과 접근 제어 측면에서 한계를 가진다. 본 논문은 이러한 문제를 해결하기 위해 End-Fog-Cloud 전 구간을 포괄적으로 보호하는 데이터 프로비넌스 시스템을 제안한다. 제안 시스템은 블록체인을 결합한 로그 기반 프로비넌스로 무결성을 검증하고 DID (Decentralized Identifier) 및 VC/VP (Verifiable Credential/Verifiable Presentation) 기반 신원 인증을 통해 계층 간 접근 제어를 강화한다. 마지막으로 STRIDE 위협 모델을 적용해 제안하는 시스템이 주요 보안 위협을 효과적으로 탐지·차단함을 실험적으로 검증한다.

Key Words : IoT, Cloud, Blockchain, Data Provenance, Data Integrity, Access Control

ABSTRACT

The Internet of Things (IoT) has evolved by integrating cloud computing to overcome limited computational and storage resources. However, physical distance still causes network latency and limits real-time processing. To address this, fog computing was introduced as an intermediate layer, forming the End - Fog - Cloud architecture. Yet conventional security mechanisms remain unsuitable because of IoT device constraints, persistent reliance on Cloud Service Providers (CSPs). Consequently, existing approaches fail to guarantee data integrity and access control across all layers. This paper proposes a unified data-provenance-based security system that protects the entire End - Fog - Cloud environment. The system ensures data integrity through blockchain-anchored log provenance and enforces access control using DID (Decentralized Identifier) and VC/VP (Verifiable Credential/Verifiable Presentation)-based identity verification. Finally, we experimentally verify that the proposed system effectively detects and mitigates major security threats using the STRIDE threat model.

* 이 논문 또는 저서는 2024년 대한민국 교육부와 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2024S1A5C3A01043273)

• First Author : Kangwon National University Department of Computer Science and Engineering, tm3693@kangwon.ac.kr

° Corresponding Author : Kangwon National University Department of Computer Science and Kangwon National University IGP. in Medical Bigdata Convergence, Department of Data Science, mjchoi@kangwon.ac.kr

* Kangwon National University IGP. of Medical Bigdata Convergence, jiwonbang@kangwon.ac.kr

** Kangwon National University Department of Sport Science, phyhee@kangwon.ac.kr

논문번호 : KNOM2025-02-011, Received November 13, 2025; Revised December 1, 2025; Accepted December 21, 2025

I. 서 론

IoT (Internet of Things)는 센서와 통신 모듈을 내장한 디바이스를 통해 실시간으로 데이터를 수집하고 전송하는 기술이다. 이러한 IoT 기술은 의료, 스마트 팩토리 등 다양한 분야에서 활용되고 있다 [1][2]. 그러나 IoT 디바이스는 저전력, 저사양이라는 특성 때문에 대규모 데이터를 장기 저장하거나 복잡한 연산을 수행하는 데 한계가 있다 [3][4]. 이러한 제약을 보완하기 위해 IoT 환경은 클라우드 컴퓨팅(Cloud Computing)과 결합되어 발전해왔다. 클라우드 컴퓨팅은 서버, 스토리지 등의 자원을 인터넷을 통해 제공하는 기술이다. 두 기술의 결합으로 IoT 디바이스에서 생성되는 데이터를 대규모 저장 공간과 고성능 환경에서 효율적으로 저장하고 처리할 수 있도록 지원한다 [5]. 하지만 클라우드와 IoT로 구성된 환경은 몇 가지 근본적 문제를 지닌다. IoT 디바이스와 클라우드 간의 물리적 거리로 인해 네트워크 지연이 발생하고, 실시간 처리가 어려워진다. 또한 클라우드에 대한 의존성이 커진다는 문제도 존재한다 [6][7]. 이러한 문제를 해결하기 위해 그림 1과 같이 End (IoT)와 클라우드 사이에 중간 계층을 두는 Fog 컴퓨팅이 도입되었다 [8].

End - Fog - Cloud 환경에서 Fog 계층은 End 계층과 지리적으로 가까운 위치에서 데이터를 처리한다. 이에 따라 네트워크 지연을 줄이고, 연산량이 큰 작업은 클라우드가 담당해 전체 부하를 효과적으로 분산한다. 이러한 구조는 IoT의 성능 제약을 해소하지만, 데이터 무결성과 접근 제어와 같은 보안 문제를 여전히 가진다 [9]. IoT 환경에서는 의료 데이터, 개인정보 등 민감한 데이터가 다루어진다. 이러한 데이터가 제3자에게 노출되거나 공격자에

의해 변조될 경우 심각한 피해로 이어질 수 있다. 변조된 데이터가 상위 서비스나 의사 결정 과정에 반영되면 시스템 오동작이나 장애를 유발할 수 있다 [1][2]. 그러나 IoT는 자원이 제한적이기 때문에 기존의 무거운 보안 기법을 적용하기 어렵다. 또한 수많은 디바이스가 동시에 인증받아야 하므로 중앙 PKI (Public Key Infrastructure) 기반 인증 방식은 적합하지 않다 [10]. 다른 문제로는 클라우드 컴퓨팅의 신뢰 문제도 대두된다. 클라우드 컴퓨팅에 저장된 데이터는 통제권이 데이터 소유자에서 CSP (Cloud Service Provider)로 넘어가게 된다. 만약 CSP가 악의적인 목적으로 데이터를 삭제하거나, 관리의 문제로 데이터가 소멸할 경우, 데이터 소유자는 이를 인지하기 어렵다 [11]. 따라서 End - Fog - Cloud 환경에서는 데이터 무결성을 감시할 수 있는 구조가 필요하다. 동시에 저장된 데이터는 권한을 가진 주체만 접근할 수 있도록 접근 제어 체계를 갖추어야 한다.

본 논문에서는 End - Fog - Cloud 환경에서 데이터 무결성과 접근 제어를 동시에 보장하는 시스템을 제안한다. 제안하는 시스템은 데이터의 생성, 변환, 전달 과정 전체를 추적하기 위해 데이터 프로비넌스(Data Provenance) 기법을 적용한다. 데이터 프로비넌스는 데이터의 출처와 처리 이력을 설명하는 메타데이터이다. 데이터의 전 생애주기를 추적함으로써 무결성 검증과 책임 추적성을 보장하는 기법이다 [12]. 본 연구에서는 로그를 기반으로 데이터 프로비넌스를 구현한다. 기록된 로그는 머클 트리(Merkle Tree) 형태로 구성되며, 루트 노드인 루트 해시를 블록체인에 저장한다. 이를 통해 클라우드에서 발생할 수 있는 데이터 위·변조를 탐지할 수 있다. 또한 DID (Decentralized Identifier), VC (Verifiable Credential)/VP (Verifiable Presentation)를 활용하여 분산 신원 인증과 토큰 기반 접근 제어를 수행함으로써 신뢰성 있는 데이터 관리 환경을 제공한다.

본 논문의 목적은 다음과 같다. 1) 블록체인, 로그 기반 데이터 프로비넌스 그리고 머클 트리를 결합하여 End - Fog - Cloud 환경에서 발생할 수 있는 데이터 위·변조 문제를 해결할 수 있는 시스템을 설계한다. 2) DID와 VC/VP 기반의 분산 신원 인증 체계를 도입하여 중앙화된 인증 서버 없이도 안전한 접근 제어가 가능한 구조를 구현한다. 3) 실험 및 분석을 통해 제안한 시스템이 데이터 무결성과 접근 제어 요구사항을 충족함을 검증한다.

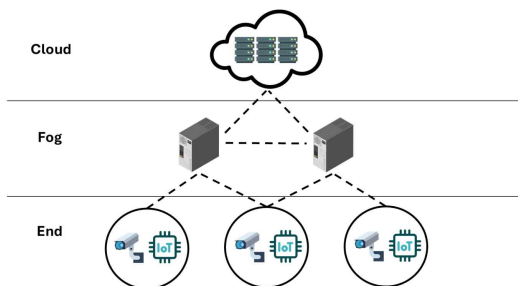


그림 1. End - Fog - Cloud 구조
Fig. 1. End - Fog - Cloud architecture

본 논문의 구성은 다음과 같다. 2장에서는 본 연구에서 활용한 핵심 기술인 분산 원장 기술에 대해 설명한다. 이후 기존 IoT 환경에서 데이터 무결성과 접근 제어를 보장하기 위한 연구들을 소개한다. 3장에서는 제안 시스템의 전체 구조, 동작 방식, 구성 요소에 대해 상세히 기술한다. 4장에서는 제안한 시스템의 데이터 무결성과 접근 제어 보장 능력을 분석한다. 마지막으로 5장에서는 결론과 향후 연구 방향을 제시한다.

II. 배경지식 및 관련 연구

이 장에서는 본 논문에서 제안하는 시스템의 이해를 돕기 위해 사용된 분산 원장 기술과 배경지식을 설명한다. 이후 IoT 환경에서 보안 문제를 해결하기 위한 기존 연구를 소개한다. 마지막에는 제안하는 시스템이 이 기존 연구들과 어떤 차별점을 가지는지 설명한다.

2.1 분산 원장 기술

블록체인은 데이터를 중앙집중형 저장소가 아닌, 지리적, 국가적, 기관적으로 분산된 노드들에 동일하게 저장하는 분산 원장 (Distributed Ledger) 기술이다. 새로운 블록이 생성될 때 합의 알고리즘을 통해 정당성을 검증한다. 각 블록은 이전 블록의 해시값을 포함하는 체인 구조로 연결된다. 만약 저장된 데이터를 수정 및 삭제하려면 모든 블록을 동시에 변경하고 참여자의 합의를 다시 획득해야 한다. 이러한 구조 덕분에 블록체인에 기록된 데이터는 불변성을 가진다. 대표적인 블록체인으로는 Ethereum과 Hyperledger Fabric 등이 있다 [13].

스마트 컨트랙트(Smart Contract)는 제3자의 개입 없이 계약 내용을 자동으로 실행하는 트랜잭션 프로토콜이다. 스마트 컨트랙트는 생성(Creation), 배포(Deployment), 실행(Execution), 완료(Completion)의 4단계 생명주기를 거쳐 동작한다. Ethereum이나 Hyperledger Fabric과 같은 다양한 플랫폼을 통해 구현된다. 스마트 컨트랙트는 블록체인의 불변성과 분산 원장을 통해 계약의 위변조를 방지하고 투명성을 보장함으로써, 기존 방식 대비 관리 비용을 절감하고 비즈니스 프로세스의 효율성을 개선하는 장점을 가진다 [14].

머클 트리는 대규모 집합의 데이터 무결성을 효율적으로 검증하기 위해 고안된 해시 기반 트리 구조이다. 모든 리프 노드에는 원본 데이터의 해시값이 저장되며, 자식 노드 두개의 해시값을 합친 후



그림 2. DID와 DID 문서 예시
Fig. 2. Example of DID, DID document

다시 해시 값을 구해 상위 노드를 구성한다. 최종적으로 완성된 트리에서 루트 노드를 루트 해시라고 부른다. 루트 해시는 전체 데이터 집합의 요약 값 역할을 한다 [15].

DID는 World Wide Web Consortium (W3C)에서 제안한 새로운 유형의 검증 가능한 분산 신원 인증 식별자이다. DID는 신뢰 가능한 저장 공간이 필요하다. 이러한 이유로 DID 기술은 저장된 데이터가 불변성을 가지는 블록체인과 결합되어 사용된다. 사용자가 스스로 DID를 생성하면, DID 문서가 같이 생성되며, 이는 JSON 파일로 DID 문서의 소유자의 DID, 인증 방식, 공개키 등을 저장한다. 그림 2는 DID와 DID 문서의 예시이다. “did”는 DID 스키마로 DID 체계를 따른다는 표시이다. “ethr”는 DID 메서드로 어떻게 구현되는지를 정의한다. ethr는 Ethereum을 뜻하며, web 등 여러 가지가 존재한다. 마지막 부분은 DID를 식별하는 고유의 값이다 [16].

VC는 W3C에서 표준한 데이터 모델로, 신분증, 자격증과 같이 물리적 세계의 자격 증명을 디지털 환경에서 암호화하여 표현한 것이다. VC는 기존의 중앙 집중식 신원 관리 시스템과 달리, 사용자가 자신의 신원 정보를 직접 소유하고 관리한다. VC는 다음과 같은 요소로 구성된다. Claims은 주체에 대한 이름, 나이, 소속 등 정보의 집합이다. Metadata는 자격 증명의 유형, 발급자 정보, 발급 일시, 만료 일시 등을 기록한다. 마지막 Proof는 VC의 발급자의 전자 서명이 포함된다.

VP는 자신의 신원 인증을 받아야 할 때, 하나 이상의 VC로 생성하는 데이터 포맷이다. 주요 장점으로는 사용자는 VC 중 필요한 신원 정보만 제출할 수 있다는 점이다. 신원 인증이 필요할 때, VC 그대로 제출해도 되지만, 검증자가 요구하는 특정 조건에 맞춰 데이터를 가공하여 제출할 수 있다. 이를 통해 모든 정보를 공개하지 않고, 검증에 필요한 최소한의 정보만을 선택적으로 공개할 수 있다 [17].

2.2 관련 연구

End - Fog - Cloud 환경에서는 IoT 환경에서 생성·전송·저장되는 데이터는 여러 계층을 거쳐 처리된다. 또한, 클라우드와 같은 외부 인프라도 사용하기 때문에 데이터 무결성, 접근 제어는 필수적인 보안 요구사항으로 간주된다. 이를 해결하기 위해 기존 연구들은 무결성 검증 기법, 블록체인 기반 데이터 프로비넌스, 분산 접근 제어 모델을 제안해 왔다. 그러나 대부분은 특정 부분에만 집중했으며, 포괄적으로 보호하는 시스템을 제안하는 연구는 드물었다. 본 장에서는 이러한 선행 연구들을 검토하고, 본 논문이 제안하는 시스템과의 차별점을 정리한다.

클라우드 기반 IoT 환경에서 데이터 무결성과 접근 제어를 보장하기 위해 다양한 연구가 진행되었다. Zang et al.은 중앙 집중형 스토리지의 단일 장애 지점, 보안 문제(무단 삭제, 변조, 오염 등)을 지적하며 Edge 환경에 적합한 탈중앙 저장 구조인 DataPond 모델을 제안하였다. 이 모델은 실제 데이터를 여러 Edge 노드에 분산 저장한다. 그 후 블록체인에는 데이터의 버전, 수정 시각, 소유자와 같은 메타데이터만 기록한다. 이러한 과정을 통해 데이터 무결성과 최신성을 검증할 수 있도록 설계되었다. 또한 Kubernetes 기반의 클라우드 네이티브 오케스트레이션을 결합해 노드 장애 대응, 동적 역할 할당, 확장성 향상 등을 지원한다 [18].

Nguyen et al.의 연구는 IoT 데이터를 공유하는 환경에서, 기존 접근 제어 방식이 상황 변화(컨텍스트)를 반영하지 못한다는 문제를 지적한다. IoT 데

이터는 시간, 장소, 사용 목적, 센서 상태 등 주변 상황에 따라 중요도와 민감도가 달라진다. 하지만 기존 시스템은 이런 변화된 상황을 실시간으로 반영하지 못한다. 이에 따라 접근을 과도하게 허용하거나 불필요하게 차단되는 문제가 발생한다. 이를 해결하기 위해 Edge 환경에서 동작하는 EDSaaS (Edge-Driven IoT Data Sharing as a Service) 프레임워크를 제안했다. 이 시스템은 “지금껏 허용된 시간인지”, “이벤트가 발생했는지” 같은 상황 정보를 실시간으로 정책에 반영하며, 해당 조건에 맞게 접근을 자동 조정한다. 정책은 계약 형태로 작성된 후 Edge Hub에서 해석되어 즉시 적용되기 때문에, 사용자의 요청이나 상황 변화에 빠르게 대응할 수 있다 [19].

IoT 환경에서 데이터의 신뢰성을 보장하기 위해 데이터 프로비넌스 적용한 연구도 여러 연구가 진행되었다. Jaigirdar et al.의 연구에서는 IoT 환경에서는 최종 사용자가 데이터의 투명성을 확보하기 위해 전통적인 프로비넌스 기법에 보안 메타데이터를 결합한 모델을 제안하였다. Prov-DM에 보안 메타데이터를 추가하여 Prov-IoT 데이터 프로비넌스 모델을 제안하였다. Prov-DM은 W3C에서 제안한 모델로, entity, activity, agent를 기록하여 데이터 흐름을 표현한다. 데이터가 이동하는 각 단계에서 발생하는 작업과 함께 인증 실패, 미설정된 보안 모듈, 비정상 접근 시도 등 보안 관련 이벤트도 기록한다. 이를 통해 데이터가 어디서 생성되고 어디를 거쳐 전달되었는지 추적 가능하다. 더 나아가, 전달 과정 중 어떤 보안 문제가 발생했고 해당 문제가

표 1. 기존 연구와 제안 시스템 비교

Table 1. Comparison of existing research and proposal systems

Research	Data Integrity Mechanism	Access Control Mechanism	Provenance Support	Considered Layers
[17]	Blockchain-based integrity, IPFS/DHT	Blockchain + relational DB	Data Confidence (Blockchain Ledger)	Edge-Cloud (Fog layer largely not considered)
[18]	Policy violation prevention	Context-sensitive access control	Not supported	End - Fog
[19]	Security metadata - based integrity event detection	Not supported	Prov-DM model	End - Cloud (Fog considerations are limited)
[20]	Zero-watermarking for hop-by-hop integrity verification	Not supported	Watermark-based lightweight provenance	End - Multi-hop - Gateway (Cloud not included)
Ours	Blockchain + Merkle Tree + Data Provenance	DID, VC/VP	Log-based data provenance	End - Fog-Cloud

데이터의 신뢰성과 무결성에 어떤 영향을 미쳤는지를 한눈에 확인할 수 있다 [20].

Faraj et al.은 IoT 네트워크에서 데이터 무결성과 프로비넌스를 동시에 확보하기 위한 시스템인 ZIRCON을 제안하였다. 기존 IoT 환경에서는 멀티 홉(Multi-hop) 전송 과정에서 중간 노드가 데이터를 변조, 삭제하거나 공격자가 패킷을 삽입하더라도 이를 탐지하기 어려운 문제가 존재한다. 또한, 프로비넌스 레코드를 패킷에 추가해서 전송하는 구조는 전송할수록 패킷 크기가 선형적으로 증가한다. 이로 인해 IoT 센서에서 과부하를 일으킨다는 한계가 존재한다. ZIRCON은 이러한 문제를 해결하기 위해 패킷 내부에 프로비넌스 정보를 삽입하지 않는 zero-watermarking 방식을 적용한다. 이 기법은 데이터 생성 시 IP 주소, 타임스탬프, 시퀀스 번호, 데이터 해시를 기반으로 고유 watermark를 생성한다. 그 후 tamper-proof 네트워크 데이터베이스에 저장한다. 중간 노드는 패킷을 수신할 때마다 watermark를 계산해 원본 watermark와 비교를 진행한다. 이 과정을 통해 패킷 변조 여부를 실시간으로 판단하여 멀티홉 환경에서 위·변조 탐지가 가능하다 [21].

기존 연구들은 IoT 환경에서 데이터 무결성과 접근 제어를 보장하기 위해 다양한 방법을 제안해왔다. 일부 연구는 블록체인 기반 무결성 검증이나 컨텍스트 기반 접근 제어 등 특정 기능에 집중했다.

다른 연구들은 데이터의 신뢰성을 확보하기 위해 프로비넌스 기법을 적용하였다. 그러나 End, Fog, Cloud 중 특정 계층이나 제한된 환경만을 고려한다. 또한, 데이터 무결성, 데이터 프로비넌스, 접근 제어를 단일 시스템 안에서 통합적으로 제공하는 연구는 부족하다. 이에 본 논문에서는 End-Fog-Cloud 전체 범위에서 세 가지 기능을 모두 지원하는 시스템을 제안하며, 기존 연구와의 차이를 표 1에 정리하였다.

III. 시스템 설계

본 장에서는 End-Fog-Cloud 환경에서 데이터 무결성과 접근 제어를 보장하기 위한 시스템 설계를 제시한다. 제안하는 시스템은 데이터 생성부터 저장, 접근, 변환에 이르는 전 과정을 로그로 기록한다. 이러한 로그를 기반으로 데이터 흐름을 검증·추적하는 로그 기반 데이터 프로비넌스 기법을 사용한다.

그림 3은 제안하는 시스템의 전체 구조를 나타낸다. End 계층은 IoT 디바이스와 게이트웨이로 구성된다. IoT 디바이스는 센서, 카메라 등에서 수집한 데이터를 게이트웨이로 전송한다. 게이트웨이는 수집 데이터를 필터링과 정규화를 진행한다. 이후 암호화를 수행한 뒤 Fog 계층으로 전달한다. 이를 통

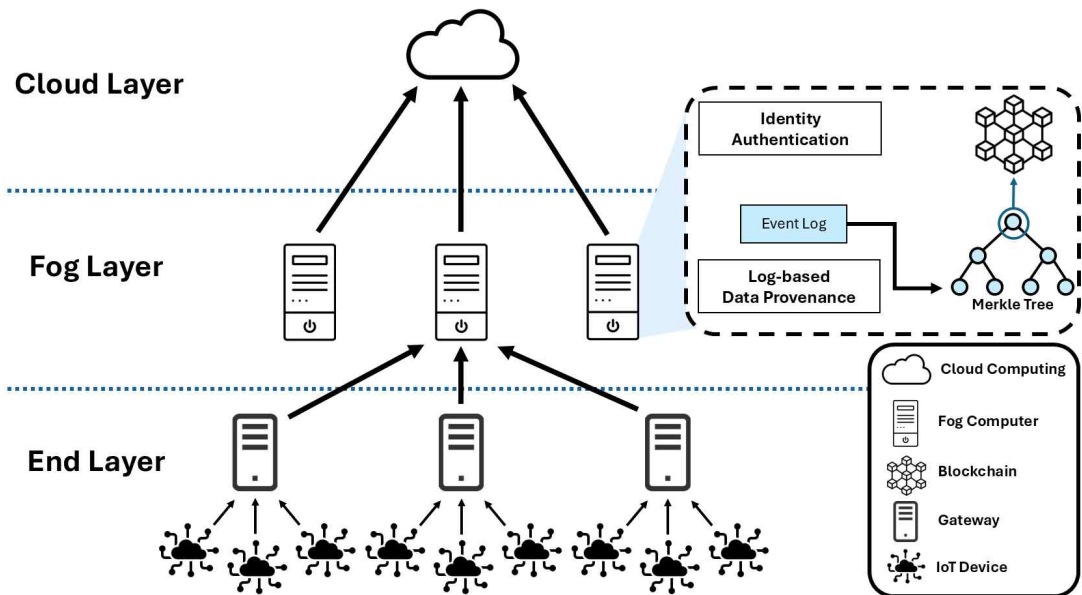


그림 3. 제안 시스템 구조

Fig. 3. Proposed system architecture

해 디바이스의 부족한 연산 능력을 게이트웨이가 보완한다. Fog 계층에는 본 시스템의 핵심 요소인 Fog 노드가 위치한다. Fog 노드는 블록체인 노드를 유지하며, 데이터 무결성을 보장하기 위한 작업을 처리한다. 또한, 이벤트가 발생하면 Fog 노드는 로그를 기록하며, 일정 주기마다 수집된 로그 데이터와 데이터를 로그 머클 트리와 데이터 머클 트리로 생성한다. 계산된 루트 해시를 스마트 컨트랙트를 이용해 블록체인에 업로드한다. DID·VC/VP 기반의 신원 인증과 토큰 기반 접근 제어를 수행한다. 더불어 클라우드 스토리지와의 데이터 송수신을 중계하는 역할도 수행한다. Cloud 계층은 클라우드 스토리지를 통해 원본 데이터를 저장하는 역할을 수행한다. 저장된 데이터는 Fog 노드에서 생성된 로그와 블록체인에 기록된 루트 해시를 이용해 무결성 검증이 가능하다.

제안하는 시스템은 End - Fog - Cloud 전 구간의 데이터 처리 과정에 신원 기반 접근 제어와 로그 기반 데이터 프로비넌스를 적용한다. 이를 통해 데이터 흐름 전반에서 발생할 수 있는 위·변조와 비인가 접근을 차단한다. End 계층은 데이터를 전처리하고 Fog 계층은 인증, 검증, 로그 생성을 담당한다. Cloud 계층은 최종 데이터를 저장한다. 또한 블록체인에 기록된 머클 루트를 활용해 저장된 데이터의 무결성을 검증할 수 있다. 이러한 구조를 통해 제안하는 시스템은 End - Fog - Cloud 환경에서 신뢰할 수 있는 데이터 관리와 접근 제어를 제공한다.

IV. 시스템 구현

표 2. End - Fog - Cloud 계층별 구현 요약
Table 2. End - Fog - Cloud Layer Implementation Summary

Layer	Components	Primary Role	Key Functions
End	IoT Device, Gateway	Data collection, Preprocessing	Data collection, Filtering, Normalization, Encryption
Fog	Fog Node	Authentication, Access control, Data provenance	DID, VC/VP verification, Session token issuance, Log generation, Maintaining an Ethereum node, Merkle root computation & Blockchain anchoring, Cloud interaction
Cloud	ownCloud	Original data storage	WebDAV-based data put/get/remove/modify, Returning data upon Fog requests

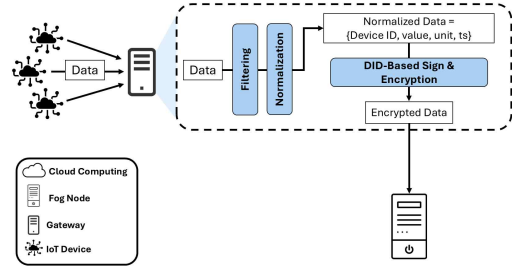


그림 4. End 계층 구현 구조
Fig. 4. End layer Implementation architecture

본 장에서는 제안하는 시스템을 어떻게 구현했는지 설명한다. 표 2는 End - Fog - Cloud 구조에서 계층별로 구현 방법을 정리했다. 이후 절에서는 해당 표를 기준으로 각 계층의 구체적인 구현 방식과 역할을 상세히 설명한다. 마지막으로 시스템 전체가 어떻게 동작하는지 흐름을 설명한다.

4.1 End 계층 구현

End 계층은 데이터의 생성과 전처리의 역할을 가지며, 그림 4와 같이 구현한다. 구성 요소는 IoT 디바이스와 게이트웨이로 구성된다. 두 구성 요소는 DID 기반 인증을 활용해 데이터를 안전하게 처리한다. IoT 디바이스는 센서나 카메라를 통해 데이터를 수집하고 이를 게이트웨이로 전달한다. 게이트웨이는 수신된 데이터를 필터링한다. 이후 디바이스 ID, 측정 값, 단위, 타임스탬프 형태로 정규화를 진행한다. 이러한 전처리 과정을 통해

데이터는 이기종 환경에서도 일관된 구조를 갖는다. 정규화된 데이터는 먼저 IoT 디바이스의 DID에 대응하는 개인키로 서명되어 데이터의 출처와 무결성이 보장된다. 이후 Fog 노드의 DID 문서에 포함된 공개키로 암호화한 뒤 Fog 계층으로 전송된다. Fog 노드는 공개키 기반 서명 검증을 통해 데이터의 무결성과 출처를 확인하고, 자신의 개인키로 데이터를 복호화한다.

4.2 Fog 계층 구현

Fog 계층에서는 제안하는 시스템의 핵심인 Fog 노드가 동작된다. 본 절에서는 머클 트리를 어떻게 구현했는지를 설명한다. 이후, 로그 기반 데이터 프로비넌스의 구조에 대해 설명한다. 마지막으로 Fog 노드가 수행하는 세부 동작을 기술한다.

4.2.1 머클 트리 구현

본 논문에서 제안하는 시스템은 로그 기반 데이터 프로비넌스의 위·변조를 방지하기 위해 머클 트리를 활용한다. Fog 노드의 처리 부하와 블록체인 앵커링 비용을 고려하여, 데이터 무결성 검증 지연을 과도하게 증가시키지 않는 범위에서 30분 주기의 머클 트리 생성을 적용하였다. Fog 노드에서는 30분 주기로 로그 데이터와 수집된 데이터를 각각 로그 머클 트리와 데이터 머클 트리로 계산한다. 두

트리는 동일한 방식으로 생성된다. 각 레코드는 고정된 형식으로 직렬화된 뒤 Keccak-256 해시 알고리즘을 적용하여 리프 노드를 구성한다. 이후 인접한 두 리프를 사전순으로 정렬한다. 그 후 다시 Keccak-256 해시를 적용해 상위 노드를 생성하며, 이를 반복해 최종적으로 하나의 루트 해시만 남도록 한다. 만약 리프 노드가 홀수인 경우, 마지막 리프 노드를 복제하여 짝수 개로 맞추는 방식을 이용한다. 이렇게 얻어낸 머클 트리의 루트 해시를 스마트 컨트랙트를 이용해 Ethereum에 저장한다. 본 구현에서는 Ethereum Sepolia Testnet을 사용한다. 저장할 때, 루트 해시를 찾기 위해 Fog 노드는 Hash ID를 생성한다. Hash ID는 디바이스 ID, 수집 시작 시간, 수집 종료 시간 그리고 루트 해시의 종류로 문자열을 만든다. 루트 해시의 종류는 데이터 머클 트리인 경우, data, 로그 머클 트리인 경우, log로 기록된다. 스마트 컨트랙트를 호출해서 Hash ID를 입력하면 해당 시간에 기록된 디바이스 ID의 머클 트리를 손쉽게 얻어 올 수 있다.

4.2.2 로그 기반 데이터 프로비넌스 구현

Fog 노드는 데이터의 생성, 접근 이력, 변환 과정들을 로그로 기록한다. 로그의 세부 구조는 그림 5와 같다. event_id는 각 로그의 고유 값이다.

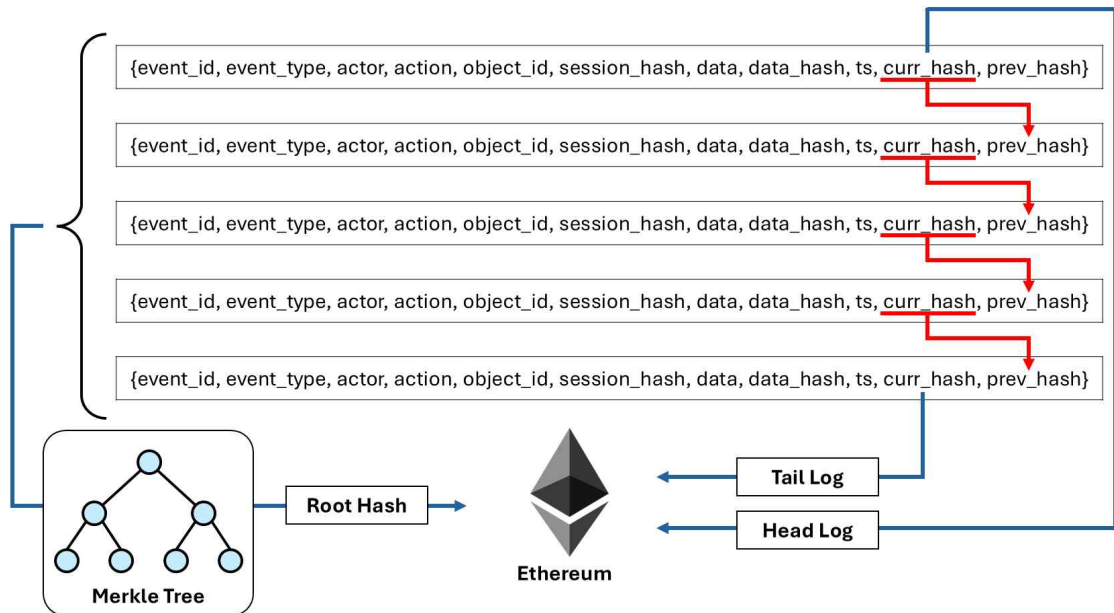


그림 5. 로그 기반 데이터 프로비넌스
Fig. 5. Log-based data provenance

event_type은 메시지의 유형을, actor는 이벤트를 발생시킨 객체를 기록한다. action은 event_type에서의 세부 동작을 명시한다. object_id는 어떤 디바이스에서 어느 시각에 생성된 데이터인지 식별하기 위한 값이다. device_1에서 2025년 11월 1일 13시 5분 30초에 생성된 데이터의 경우 object_id는 device_1:2025-11-01 13:05:30이 된다. session_hash는 접근 제어에 사용된 세션 토큰의 해시 값이다. data는 Fog 노드로 전달된 원본 페이로드이며, data_hash는 data의 무결성을 검증하기 위한 해시 값이다. ts는 로그가 기록된 시각을 기록한다. curr_hash는 해당 로그의 해시 값이며, prev_hash는 직전 로그의 해시 값을 기록한다.

기록된 로그들은 위 변조가 발생해서는 안 되므로, 본 시스템은 로그의 데이터 무결성을 보장하기 위해 다음의 작업을 수행한다. Fog 노드는 수집된 로그들을 30분 간격으로 머클 트리로 계산한다. 이후 계산된 루트 해시를 스마트 컨트랙트를 호출해 블록체인에 저장한다. 제안하는 시스템은 주기적으로 기록된 로그들을 머클 트리로 구성한다. 새롭게 계산된 루트 해시와 블록체인에 기록된 루트 해시를 비교해 로그들이 변조되지 않았는지 지속적으로 검증한다.

제안하는 방법은 주기적으로 생성되는 머클 트리

를 통해 데이터 무결성을 보장할 수 있다. 그러나 머클 트리가 생성되기 전에 로그가 수정되는 경우 이를 탐지할 방법이 없다. 이를 보완하기 위해 본 논문에서는 curr_hash와 prev_hash를 연결한 체인형 구조로 구성한다. 특정 로그가 변조되면 이후 모든 로그의 해시가 연쇄적으로 불일치하게 된다. 이를 통해 변조 탐지가 가능하다. 그러나 이런 구조에서도 문제가 있다. 제일 처음 기록된 Head 로그와 머클 트리가 생성되기 전 마지막에 기록된 Tail 로그가 보호되지 않는다. 이를 해결하기 위해 본 논문에서는 Head 로그와 Tail 로그의 해시 값을 스마트 컨트랙트를 통해 블록체인에 저장한다. 결과적으로 Head와 Tail 로그가 변조되더라도 블록체인에 저장된 해시와 대조함으로써 완전한 검증이 가능해진다.

4.2.3 Fog 노드 동작 구현

Fog 노드의 전체 동작은 그림 6과 같다. 사용자 또는 게이트웨이는 요청 목적을 명시하는 msg_type과 관련 데이터를 포함하여 페이로드를 생성한다. 이후 생성된 페이로드를 TCP를 통해 Fog 노드로 전송한다. Fog 노드는 수신된 페이로드를 파싱(Parsing)하여 msg_type을 식별하고, 이에 따라 다음과 같이 분기 처리를 수행한다.

첫 번째로, msg_type이 data인 경우, Fog 노드는 페이로드 내의 device_id와 타임스탬프를 분석한다.

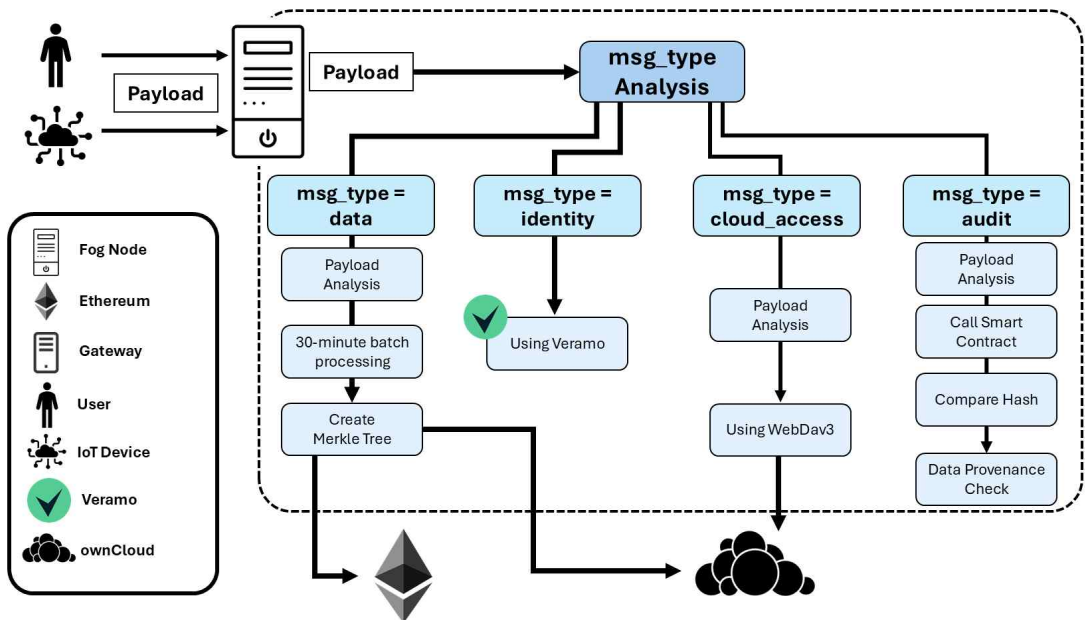


그림 6. Fog 노드 동작
Fig. 6. Fog Node Processing

이후, 데이터의 출처 및 생성 시점을 식별한다. 수집된 데이터는 30분 단위로 배치(Batch) 처리되며, Fog 노드는 해당 주기 동안 축적된 데이터에 대해 머클 트리를 구성한다. 머클 트리에서 루트 해시 얻어, 스마트 컨트랙트를 통해 블록체인에 기록한다.

두 번째로, identity 유형은 Fog 노드가 신원 인증 및 접근 제어 작업을 수행한다. Fog 노드는 페이로드 내 action 필드를 읽고 VP 검증, 세션 토큰 발급 등 인증 관련 작업을 수행한다. 본 시스템은 DID 및 VC/VP 기반 프레임워크인 Veramo [19]를 활용한다. Fog 노드는 Veramo 서버를 구동한다. 이후 HTTP API를 통해 신원 인증 및 접근 제어 관련 작업을 수행한다.

세 번째로, cloud_access 유형은 클라우드에 저장된 데이터에 대한 접근을 제어한다. action 필드에는 세부 동작이 명시된다. 동작은 총 4가지로, get은 데이터 조회, put은 저장, remove는 삭제, modify는 수정을 진행한다. Fog 노드는 해당 요청을 검증한 후, Python 라이브러리인 webdav3를 이용하여 WebDAV API를 통해 ownCloud와 상호작용한다.

마지막으로, msg_type이 audit인 경우는 사용자가 CSP의 데이터 위·변조, 임의 삭제 등 부정행위를 의심하여 데이터 무결성 검증을 요청하는 상황을 처리한다. Fog 노드는 해당 페이로드를 수신하면, 클라우드에 저장된 의심 시간의 데이터를 가져온 뒤 머클 트리를 계산 후 루트 해시를 얻는다. 이후, Ethereum에 기록된 루트 해시를 비교해 불일치가 있는지 확인한다. 만약 불일치가 발생하면, Fog 노드는 감사 요청이 들어온 시간부터 데이터가 생성된 시간까지의 로그 데이터를 역순으로 추적한다. 예를 들어 감사 요청 시각이 2025년 10월 30일 12:00:00이며, 의심되는 데이터가 2025년 10월 28일 12:00부터 13:00까지의 구간이라면, Fog 노드는 감사 요청 시각의 로그 데이터부터 10월 28일 12:00까지 순차적으로 확인한다. 이 과정에서 로그에 포함된 object_id가 핵심적인 식별자로 사용된다. 클라우드에 저장된 원본 데이터들을 이용해서 cloud_object_id를 만든 후, 로그 데이터를 역방향으로 탐색하면서 해당 데이터의 가장 마지막에 기록된 로그를 찾는다. 일치하는 object_id를 발견했고, 로그에 기록된 내용이 맞으면, 해당 데이터는 위·변조가 되지 않은 데이터란 뜻이다. 하지만, cloud_object_id와 일치하는 object_id를 찾을 수 없거나, 찾았음에도 로그의 data 부분과 실제 클라우

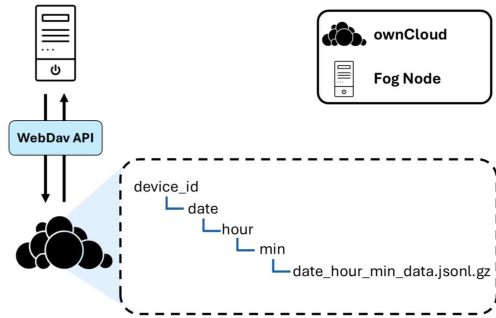


그림 7. Cloud 계층 구현

Fig. 7. Cloud layer Implementation

드에 저장된 데이터가 불일치가 발생한다면 이는 CSP가 데이터를 무단으로 변조했거나 삭제했다는 뜻이다. 이러한 과정을 통해 데이터 무결성 검증이 가능하다.

4.3 Cloud 계층 구현

Cloud 계층은 원본 데이터를 저장하기 위한 클라우드 컴퓨팅이 동작된다. 본 논문에서 제안하는 시스템은 ownCloud를 사용하여 그림 7과 같이 구현하였다. ownCloud는 파일 저장, 사용자 관리, 동기화 기능을 제공하는 오픈소스 클라우드 스토리지 플랫폼이다. ownCloud는 WebDAV 기반의 표준화된 인터페이스를 제공한다. 이를 통해 Fog 노드는 저장소에 대한 조회, 저장, 삭제, 수정 요청을 안전하게 수행할 수 있다. ownCloud는 Docker Compose를 이용해 구성하였다. ownCloud 컨테이너는 MariaDB, Redis, Apache 등과 함께 작동한다. 이를 통해 안정적인 스토리지 운영과 데이터 관리가 가능하다. Fog 노드는 Python의 webdav3 라이브러리를 통해 ownCloud의 WebDAV API와 직접 상호작용하며, 데이터 입출력을 처리한다. 저장되는 데이터는 디바이스별·날짜별·시간대별로 체계적으로 관리된다. 디렉터리 구조는 device_id/날짜/시간/분/파일 형태로 구성된다. 그에 대한 예시로, device_1에서 2025년 11월 1일, 13시 30분에 수집된 경우, device_1/2025-11-01/13/30/2025-11-01_13_data.jsonl.gz와 같이 저장된다. 이러한 구조를 통해 시간 단위로 데이터를 관리한다.

4.4 시스템 동작

본 절에서는 제안하는 시스템의 전체적인 동작을 설명한다. 동작은 총 4가지로, 등록, 저장, 사용

그리고 감사 과정으로 구분된다. 각 동작은 End - Fog - Cloud 환경에서 서로 다른 구성 요소들이 어떤 역할을 수행하며 데이터를 처리하는지를 단계별로 보여준다.

첫 번째로 등록 단계는 그림 8과 같은 흐름을 가진다. 등록 단계는 IoT 디바이스나 사용자가 시스템에 참여하기 위해 Fog 노드로부터 접근 권한을 획득하는 과정이다. 등록 주체는 총 세 가지로, 데이터 생성자인 IoT 디바이스, IoT 디바이스를 소유 중인 IoT 디바이스 소유자 그리고 수집된 데이터를 사용하는 데이터 사용자로 분류된다. 등록 절차는 주체의 유형에 따라 다음과 같이 수행된다. 먼저 IoT 디바이스 소유자와 데이터 사용자는 정부, 기업, 교육 기관 등 신뢰할 수 있는 기관에 자신의 DID를 제출하여 VC를 발급받는다. 이후, 획득한 VC들을 조합하여 자신이 정당한 권한을 가진 사용자임을 증명하는 VP를 생성한다. 반면, IoT 디바이스의 경우, 제조 또는 초기 설정 단계에서 DID와 VP가 ROM에 저장된 상태로 운용된다. 각 주체가 생성된 VP를 포함해 msg_type이 identity인 페이로드를 Fog에 제출하면, Fog 노드는 VP의 유효성을 검증한다. 검증이 성공하면 해당 주체의 역할에 부합하는 세션 토큰을 생성하여 발급한다. 이후 발생하는 모든 페이로드에서 주체는 발급받은 세션 토큰을 포함하여 전송함으로써, 반복적인 DID 검증을 할 필요 없이 자신이 사전에 승인된 사용자임으로 효율적으로 인증한다. 결과적으로 본 시스템은 VP 기반의 신원 인증과 세션 토큰 기반의 접근 제어를 결합하여 IoT 환경에서의 데이터 기밀성을 보장한다.

두 번째는 저장 단계로 IoT 디바이스에서 수집한 데이터가 게이트웨이와 Fog 노드를 거쳐

클라우드에 안전하게 저장되는 과정이다. 흐름은 그림 9와 같다. 먼저 IoT 디바이스는 센서, 카메라 등 물리적 모듈에서 수집한 데이터를 게이트웨이로 전송한다. 게이트웨이는 데이터들을 필터링과 정규화를 거쳐 전처리를 수행한다. 이후 암호화를 수행한 후, IoT 디바이스가 사전에 발급한 토큰을 포함하여 Fog 노드로 전송한다. Fog 노드는 수신된 페이로드의 msg_type이 data인지를 확인하고, 포함된 세션 토큰의 유효성을 검증한다. 검증이 완료되면 전달 받은 데이터를 임시 저장하며, 이러한 내용을 로그로 기록하여 저장한다. Fog 노드는 30분 주기로 수집된 데이터에 대해 머클 트리를 구성하여 루트 해시를 Ethereum에 업로드한다. 이를 통해 데이터 무결성을 확보하고, 암호화된 원본 데이터는 클라우드 스토리지에 저장한다.

세 번째는 사용 단계로, IoT 디바이스 소유자, 데이터 사용자 그리고 다른 IoT 디바이스가 수집된 데이터에 접근하는 과정을 설명한다. 그림 10은 사용 단계의 흐름을 보여준다. 데이터 접근을 희망하는 주체는 msg_type을 cloud_access로 설정하고, 수행하고자 하는 동작, get, put, remove, modify를 action에 명시한다. 이후 접근하고자 하는 데이터의 정보, 디바이스 ID, 수집된 날짜와 시간 등을 명시한 페이로드를 생성하여 세션 토큰과 함께 Fog 노드에 전송한다. 요청을 수신한 Fog 노드는 페이로드와 함께 온 세션 토큰을 검증한다. 요청자의 신원을 확인한 후 역할 기반 접근 제어 정책에 따라 권한을 검증한다. 예를 들어 IoT 디바이스 소유자는 데이터 조회(Get)뿐만 아니라 저장(Put), 삭제(Remove), 수정(Modify)에 대한 모든 관리 권한이 부여된다. 반면, 데이터 사용자는 오직 데이터 조회(Get) 권한만이 허용된다. Fog

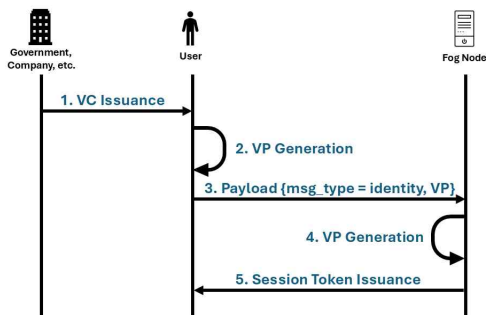


그림 8. 등록 과정 흐름
Fig 8. Registration Flow

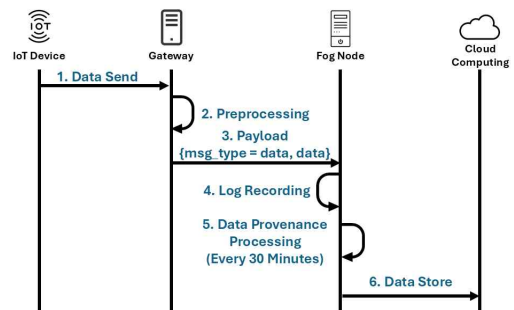


그림 9. 저장 단계 흐름
Fig. 9. Save Flow

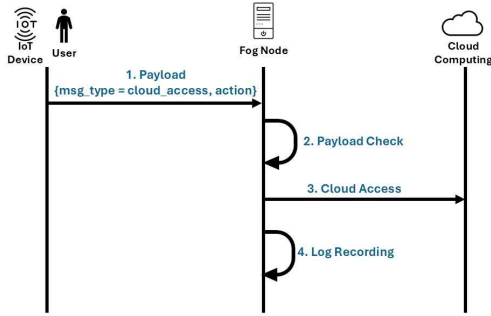


그림 10. 사용 단계 흐름
Fig. 10. Usage Flow

노드는 이러한 권한 검증이 완료된 후 클라우드와 상호작용하여 요청된 작업을 수행하고 그 결과를 반환한다. 마지막으로 이러한 동작에 대한 로그를 생성해서 저장한다.

마지막으로 감사 단계는 IoT 디바이스 소유자가 자신의 디바이스에서 수집된 데이터가 변조되지 않았음을 확인하는 과정으로, 그림 11과 같은 흐름으로 수행된다. 사용자는 발급받은 세션 토큰과 함께 msg_type을 audit으로 설정하여 의심되는 시간 구간을 포함한 페이로드를 Fog 노드로 전송한다. Fog 노드는 세션 토큰 검증을 진행한 후, 페이로드에 명시된 시간대의 원본 데이터를 클라우드에서 불러온다. 이후 동일 구간의 데이터를 기반으로 머클 트리를 재구성하여 루트 해시를 계산한다. Fog 노드는 이 값을 Ethereum에 기록된 해당 시간대의 루트 해시와 비교한다. 두 값이 일치하는 경우 데이

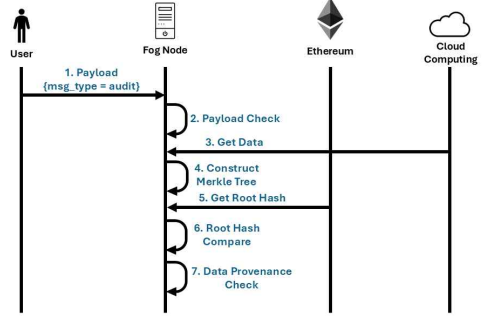


그림 11. 감사 단계 흐름
Fig. 11. Audit Processing

터가 변조되지 않았음을 확인할 수 있다. 반대로 해서 값이 다를 경우, 생성된 데이터 프로비넌스를 조회한다. Fog 노드에 저장된 해당 데이터의 마지막 로그를 찾아서 어떻게 값이 어떻게 변경되었는지 확인한다.

V. 실험 및 검증

본 장에서는 제안하는 시스템이 데이터 무결성과 접근 제어를 보장하는지 분석한다. 보안 분석을 실제 운영을 가정한 환경을 기반으로 진행했다. IoT 디바이스는 Raspberry Pi 3를 사용하였다. 게이트웨이는 Desktop 환경으로 구현했으며, Fog 노드는 다른 Desktop 환경에서 구현하였다. 클라우드 환경은 가상 환경을 이용하였다. 본 논문에서 제안하는 시스템의 보안성을 평가하기

표 3. STRIDE 위협 모델별 제안 시스템 대응

Table 3. Countermeasures of the Proposed System based on the STRIDE Threat Model

STRIDE Category	Threat Definition	Proposed Countermeasures
Spoofing	An attacker attempts unauthorized access using a forged VP.	VP signature verification
Tampering	(1) A CSP arbitrarily modifies or deletes cloud data. (2) An attacker modifies or deletes log data recorded on the Fog node.	(1) Blockchain anchoring and auditing (2) Log chain verification
Repudiation	A legitimate user with modification privileges denies having performed an action.	Data provenance backtracking
Information Disclosure	An attacker attempts to steal data by sniffing network packets in the End - Fog segment.	Public key-based encrypted transmission
Denial of Service	An attacker attempts to flood the system with requests containing invalid signatures.	Preemptive blocking mechanism
Elevation of Privilege	A user with only read permissions attempts to modify or delete data.	Session token-based access control

위해 STRIDE 위협 모델 [22]을 적용한다. STRIDE 위협 모델은 6가지 항목을 기반으로 시스템의 보안 위협을 식별하고 분석하는 방법론이다. 항목은 총 6가지로, 스푸핑(Spoofing), 변조(Tampering), 부인(Repudiation), 정보 노출(Information Disclosure), 서비스 거부(Denial of Service), 그리고 권한 상승(Elevation of Privilege)으로 구성된다. 본 모델을 통해 시스템 설계 상의 취약점과 위험 요소를 식별할 수 있다. 본 논문에서 제안하는 시스템은 STRIDE 위협 모델의 6가지 항목을 기반으로 총 6가지 시나리오를 설정한다. 표 3은 각 항목에 대한 공격 시나리오와 이에 대한 대응을 정리하였다.

4.1 스푸핑 시나리오

첫 번째 시나리오는 스푸핑 공격으로, 공격자가 위조된 VP를 이용해 Fog 노드에 비인가 접근을 시도하는 상황을 가정한다. 본 시스템에서는 VP는 사용자의 DID와 VC를 포함하며, 신원 인증을 수행하는 핵심 요소로 사용된다. 공격자는 과거에 발급된 정당한 VP를 조작하거나, 다른 사용자의 VC를 변형하여 새로운 VP를 구성한 뒤 이를 이용해 접근 권한을 얻으려고 시도한다. 실험에서는 기존에 발급된 정상 VP를 일부 변조한 뒤 Fog 노드에 제출하여 인증 절차를 우회하려는 공격을 시도하였다. Fog 노드는 검증 과정에서 위조된 VP의 서명 불일치를 탐지하였으며, 그림 12와 같이 인증 실패 메시지를 반환하였다. 이를 통해 제안하는 시스템이 위조된 VP 기반의 신원 위조 및 비인가 접근을 효과적으로 차단함을 확인할 수 있다.

4.2 변조 시나리오

두 번째 시나리오는 변조 공격으로, 시스템 내 데이터의 무결성을 훼손하는 위협을 다룬다. 본 논문에서는 변조 공격의 대상을 크게 두 가지로 분류하여 시나리오를 구성하였다. 첫째는 클라우드 계층에서 CSP에 의해 원본 데이터가 조작되는 경우이다. 둘째는 공격자가 Fog 노드 내부

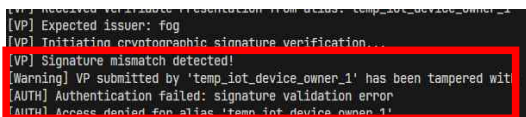


그림 12. VP 검증 결과
Fig. 12. VP Verification Results

시스템에 침투하여, 자신의 악의적인 행위를 숨기기 위해 로그 데이터를 수정하거나 삭제하는 상황이다.

4.2.1 CSP의 악의적인 행위

시나리오 2-1은 악의적인 CSP가 Fog 노드의 통제를 벗어나 클라우드에 저장된 데이터를 임의로 삭제하거나 값을 변경하는 상황을 가정한다. 실험은 CSP의 악의적 행위를 묘사하기 위해 Python 스크립트를 사용하여 11월 11일 10:00부터 14:00 사이에 생성된 데이터 중 무작위로 10개를 선정한다. 그 중 5개는 삭제, 5개는 값을 변조한다. 이후 해당 시간 구간에 대한 Audit을 수행한다. 그 결과, 그림 13과 같이 시스템은 삭제된 데이터와 변조된 데이터의 불일치를 모두 탐지하고 경고를 발생시켰다. 따라서 본 논문에서 제시하는 시스템은 CSP의 악의적인 행동에 대해 데이터 무결성을 검증하는 메커니즘을 제공한다.

4.2.2 로그 데이터 위 변조

시나리오 2-2는 공격자가 Fog 노드 내부 시스템에 침투하여, 자신의 악의적인 행위를 숨기기 위해 로그 데이터를 수정하거나 삭제하는 상황이다. 로그 데이터는 데이터 프로비전스의 핵심이자 변조 탐지의 기준점이므로, 무결성이 훼손되어서는 안 된다. 공격자가 이미 블록체인에 앵커링 된 과거 로그를 수정하거나, 아직 머클 트리가 생성되기 전의 최신 로그를 조작하여 흔적을 지우려는 시도를 할 수 있다. 실험은 두 유형의 변조 시나리오를 직접 구현하였다. 첫 번째 유형은 Ethereum에 루트 해시가 기록된 로그의 내용을 수정한 후, 사용자가 audit 요청을 진행한 상황이다. 두 번째 유형은 머클 트리가 계산되기 이전에 로그를 조작하여, 해당 로그의 curr_hash와 다음 로그의 prev_hash가 불일치가 발생한 상황이다. 두 유형을 재현하기 위해, Python 스크립트로 임의의 로그 데이터를 수정하였다. 그 결과 그림 14와 같이 두 상황에서의 로그 데이터의

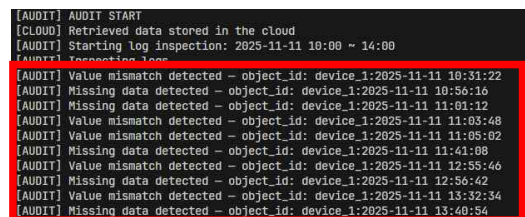


그림 13. 클라우드 데이터 감사
Fig. 13. Cloud Data Audit Results

```
[AUDIT] Experiment started: log tampering scenario simulation
[AUDIT] Detecting tampering in previously finalized logs...
[AUDIT] Loading Merkle root from blockchain...
[AUDIT] Recalculating Merkle root from cloud logs...
[AUDIT] Hash mismatch detected
[Warning] Previously committed log has been modified
[AUDIT] Detecting tampering before Merkle tree generation...
[AUDIT] Verifying log chain hash linkage (prev_hash)...
[AUDIT] Broken hash chain detected - prev hash mismatch
[WARNING] Log modified before Merkle tree creation
[AUDIT] Both tampering scenarios were correctly identified
[AUDIT] System integrity verification completed
```

그림 14. 로그 위·변조 탐지 결과

Fig. 14. Log Tampering Detection Results

위·변조를 탐지했음을 보여준다. 주기적인 감사를 통해, 이전의 로그가 변조됐음을 탐지하고 경고하고 있다. 또한, 머클 트리로 계산되기 전에 로그가 수정됐을 때, 해시 값이 달라져 시스템이 위·변조를 탐지한다.

4.3 부인 시나리오

세 번째 시나리오는 부인으로, 데이터의 수정 권한을 가진 사용자가 데이터를 수정했지만, 이를 부인하는 상황을 가정한다. 정당한 권한을 가진 IoT_Device_Owner_2가 특정 데이터를 Fog 노드를 통해 수정한 후, 해당 행위를 수행하지 않았다고 주장하는 상황을 가정한다. 감사 모듈은 데이터 프로비넌스 로그를 역추적하여, 수정 이벤트가 발생한 시점의 로그를 추출하였다. 이후 로그 내에 기록된 이벤트 발생자를 확인하여 IoT_Device_Owner_2 임을 식별하였다. 그림 15는 감사 결과 화면을 나타내며, 로그에 기록된 내역을 통해 부인 방식을 보장한다.

4.4 정보 노출 시나리오

네 번째 시나리오는 정보 노출 시나리오로, 공격자가 End-Fog 구간의 네트워크 패킷을 스니핑하여 민감한 데이터를 탈취하려는 상황을 가정한다.

```
[AUDIT] Audit Request Received for Object ID: device_1:2025-12-10 19:30:45
[INFO] Target Event Found! Event Type: MODIFY
[INFO] Target Event Actor: IoT_Device_Owner_2
```

그림 15. 부인 방식을 위한 데이터 프로비넌스 감사 결과
Fig. 15. Audit Result of Data Provenance for Non-Repudiation

```
0000 08 bf b8 13 1a 0a b8 27 eb 9a 75 d3 08 00 45 10 .....u...E...
0010 00 78 a2 c7 40 00 40 06 16 3f c8 a8 00 16 c8 a8 ...X...@...?.....
0020 00 03 00 16 d7 e1 94 84 ba 7d d1 2b 53 bd 50 18 .....-)+S-P...
0030 03 8f 03 6c 00 00 e0 99 a2 a8 9b 97 ea b4 e4 c3 .....I.....
0040 ed 48 99 c2 4b 46 35 6b cb 09 b6 16 22 28 73 04 ...H-KF5k.....(s-
0050 ac 53 b0 29 81 75 f3 f7 52 69 c0 f5 0f 6a 42 9e ...S)u...Rk...jB
0060 c8 ee 29 74 2c 0a 49 cb ff de 6b 90 df f0 d5 53 ...-)+...k...S
0070 55 d0 3b 7b 35 55 19 c3 19 b0 03 79 ef 03 93 b9 ...U;{SU...y...
0080 27 6f 5e 98 7e 46 ..o...f
```

그림 16. Wireshark로 캡처한 암호화된 패킷 페이로드
Fig. 16. Encrypted Packet Payload Captured by Wireshark

본 시스템은 전송 구간의 기밀성을 보장하기 위해 게이트웨이가 Fog 노드의 DID 문서에서 획득한 공개키를 사용하여 데이터 페이로드를 암호화한 후 전송하는 방식을 적용한다. 실험에서 네트워크 패킷 분석 도구인 Wireshark [23]를 이용한다. Wireshark를 통해 게이트웨이와 Fog 노드 간의 통신을 가로챌 결과, 그림 16과 같이 패킷 데이터가 암호화된 상태로 전송되는 것을 확인하였다. 이를 통해 제안 시스템은 네트워크 도청 시도에 대해서도 데이터의 기밀성을 보장함을 입증하였다.

4.5 서비스 거부 시나리오

다섯 번째 시나리오는 서비스 거부 시나리오로, 공격자가 대량의 무의미한 요청을 Fog 노드에 전송하여 시스템 자원을 고갈시키고 가용성을 저해하는 상황을 가정한다. 본 시스템은 데이터 처리 로직 수행 전 세션 토큰 및 VP 서명 유효성을 검증한다. 이러한 메커니즘으로 유효하지 않은 패킷을 네트워크 수신 단계 직후 즉시 폐기한다. 실험에서는 Python 스크립트를 통해 초당 1,000건 이상의 유효하지 않은 서명이 포함된 요청을 전송하는 플러딩(Flooding) 공격을 수행하였다. 공격이 수행되면서 동시에 Fog 노드의 CPU 및 메모리 사용량을 측정하였다. 그림 17은 Fog 노드가 인증 실패를 식별하고 악성 요청을 즉각적으로 거부하는 과정을 모니터링한 결과이다. 그림 18과 그림 19는 플러딩 공격당하고 있을 때 측정된 CPU와 메모리 사용량 변화를 나타낸다. 두 그래프에서 붉은색 수직선은 공격이 시작된 시점을, 푸른색 수직선은 공격이 중단된 시점을 의미한다. CPU 사용량은 플러딩 공격을 당하는 중에도 13% 이하로 안정적으로 유지되었다. 또한 메모리 사용량은 공격을 시도한 기간 동안 급격한 증가 없이 48%를 유지하였다. 이는 대량의 악성 패킷이 시스템의 처리 대기열(Queue)이나 메모리에 적재되기 이전에 인증 검증 단계에서 즉시 폐기되었음을 나타낸다. 결과적으로 제안 시스템은 자원 고

```
[+] Access Attempt
[WARN] Session Token Verification Failed
[-] Access Denied
[+] Access Attempt
[WARN] Session Token Verification Failed
[-] Access Exit
[-] Access Denied
[WARN] Session Token Verification Failed
[-] Access Denied
```

그림 17. 서비스 거부 공격 시 모니터링 화면
Fig. 17. Monitoring Screen During a DoS Attack

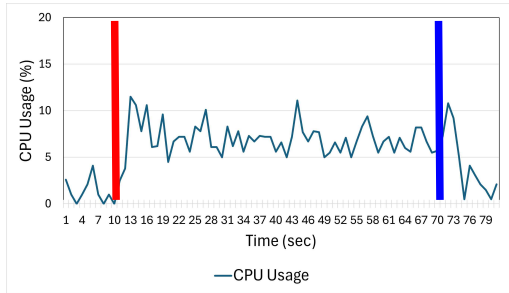


그림 18. 서비스 거부 공격 시 CPU 사용량
Fig. 18. CPU Usage During a DoS Attack

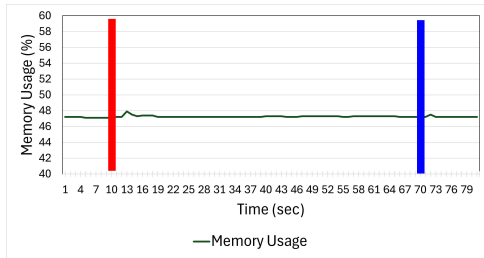


그림 19. 서비스 거부 공격 시 메모리 사용량
Fig. 19. Memory Usage During a DoS Attack

갈을 유도하는 서비스 거부 공격 상황에서도 시스템의 리소스를 효율적으로 보호하여 가용성을 안정적으로 확보함을 확인하였다.

4.6 권한 상승 시나리오

마지막 여섯 번째 시나리오는 권한 상승으로, 조희 권한만 가진 일반 데이터 사용자가 관리자 권한이 필요한 데이터 삭제나 수정을 시도하는 상황을 가정한다. 본 시스템은 이러한 비인가 행위를 방지하기 위해, 사용자 인증 시 발급된 세션 토큰에 기반한 접근 제어를 수행한다. 실험 결과, 그림 20과 같이 일반 사용자의 토큰을 포함한 데이터 삭제 요청이 토큰에 명시된 권한 범위를 초과함에 따라 시스템 정책에 의해 즉시 차단됨을 확인하였다.

VI. 결론 및 향후 연구

본 논문에서는 End - Fog - Cloud 환경에서 발생하는 데이터 무결성 훼손과 비인가 접근 문제를 해결하기 위한 보안 시스템을 제안한다. 제안 시스템은 블록체인, 로그 기반 데이터 프로비넌스 그리고 DID·VC/VP 기반 인증 및 토큰 기반 접근 제어를 결합한 구조이다. IoT 디바이스, IoT 디바이스 소유자, 데이터 사용자를 대상으로 DID 기반 등록과 VC/VP 검증을 수행함으로써

```
[+] Access Attempt
[WARN] Permission Mismatch
[-] Access Exit
```

그림 20. 권한 상승 시도에 대한 접근 차단
Fig. 20. Results of Blocking Privilege Escalation Attempts

신원 위조를 방지한다. 또한 Fog 노드는 수집된 데이터와 로그를 30분 주기로 머클 트리로 구성하고, 루트 해시를 블록체인에 저장하여 데이터와 로그의 무결성을 보장한다. 데이터 접근 과정에서 세션 토큰 기반 검증을 적용해 안전한 접근 통제를 제공한다.

제안하는 시스템의 보안성을 검증하기 위해 STRIDE 위험 모델을 기반으로 평가 시나리오를 구성하였다. 시나리오는 총 6개로, 스푸핑, 변조, 부인, 정보 유출, 서비스 거부, 권한 상승 상황으로 구성하였다. 실험 결과, 6가지의 보안 위험을 효과적으로 탐지하고 방어함으로써 시스템의 안전성과 데이터 무결성을 보장함을 확인하였다.

향후 연구에서는 로그를 역순으로 탐색하는 현재의 감사 구조가 데이터 축적에 따라 검증 시간이 증가하는 한계를 가진다는 점을 고려하여, 로그 인덱싱 기법 또는 구간 단위 요약 구조를 도입함으로써 감사 효율을 개선할 계획이다. 또한 머클 트리 생성 과정과 DID 기반 인증 절차에서 발생하는 지연 시간을 기존 시스템과 비교·분석할 예정이다. 이를 통해 데이터 무결성과 접근 제어를 강화함으로써 발생하는 오버헤드가 어느 수준인지 정량적으로 평가하고, 제안 시스템이 기존 시스템 대비 갖는 효율성을 객관적으로 제시하고자 한다.

References

- [1] C. Li, J. Wang, S. Wang, and Y. Zhang, "A review of IoT applications in healthcare," *Neurocomputing* 2024, Vol.565, pp. 1 - 12, Jan. 2024.
- [2] X. Mu, M. F. Antwi-Afari, "The applications of Internet of Things (IoT) in industrial management: a science mapping review," *International Journal of Production Research*, Vol. 62, No. 5, pp. 1928 - 1952, Mar. 2024.
- [3] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, and W. Lv, "Edge computing security: State of the art and challenges," *Proceedings of the IEEE*,

- Vol. 107, Issue. 8, pp. 1608 – 1631, Jun. 2019.
- [4] N. Sharma, P. Dhiman, “A survey on IoT Security: Challenges and their solutions using machine learning and blockchain technology,” *Cluster Computing 2025*, Vol. 28, No. 313, pp. 1 – 40, Apr. 2025.
 - [5] A. Al-Dulaimy, M. Jansen, B. Johansson, A. Trivedi, A. Losup, M. Ashjaei, and A. V. Papadopoulos, “The computing continuum: From IoT to the cloud,” *Internet of Things*, Vol. 27, pp. 1 – 33, Jun. 2024.
 - [6] M. Almutairi, T. F. Sheldon, “IoT – Cloud Integration Security: A Survey of Challenges, Solutions, and Directions,” *Electronics 2025*, Vol. 14, No. 7, pp. 1 – 28, Mar. 2025.
 - [7] L. Kong, J. Tan, J. Huang, G. Chen, S. Wang, X. Jin, and S. K. Das, “Edge-computing-driven Internet of Things: A Survey,” *ACM Computing Surveys*, Vol. 55, Issue 8, No. 174, pp. 1 – 41, Dec. 2022.
 - [8] S. Yi, C. Li, and Q. Li, “A survey of fog computing: concepts, applications and issues,” in *Proc. Mobidata ‘15: Proceedings of the 2015 Workshop on Mobile Big Data*, pp. 37 – 42, New York, USA, June 2015.
 - [9] S. N. Srirama, “A decade of research in fog computing: Relevance, challenges, and future directions,” *Practice and Experience*, Vol. 54, Issue 1, pp. 3 – 23, Jan. 2024.
 - [10] P. Williams, I. K. Dutta, H. Daoud, and M. Bayoumi, “A survey on security in internet of things with a focus on the impact of emerging technologies,” *Internet of Things*, Vol. 19, pp. 1 – 24, Jul. 2022.
 - [11] P. Goswami, N. Faujdar, S. Debnath, A. K. Khan, and G. Singh, “Investigation on storage level data integrity strategies in cloud computing classification, security obstructions, challenges and vulnerability,” *Journal of Cloud Computing*, Vol. 13, No. 45, pp. 1 – 23, Feb. 2024.
 - [12] Y. L. Simmhan, B. Plale and D. Gannon, “A survey of data provenance in e-Science,” *ACM Sigmod Record*, Vol. 32, No. 3, pp. 31 – 36, Sep. 2005.
 - [13] M. Di Pierro, “What is the blockchain,” *Computer in Science & Engineering*, Vol. 19, No. 5, pp. 92 – 95, Sep. 2017.
 - [14] Z. Zheng, S. Xie, H. Dai, W. Chen, X. Chen, J. Weng, and M. Imran, “An overview on smart contracts: Challenges, advances and platforms,” *Future Generation Computer Systems*, Vol. 105, pp. 475 – 491, Apr. 2020.
 - [15] R. C. Merkle, “A digital signature based on a conventional encryption function,” in *Proc. Advances in Cryptology – CRYPTO ‘87*, California, USA, pp. 369 – 378, Aug. 1987.
 - [16] W3C DID Working Group, Decentralized Identifiers (DIDs) v1.0(2022), Retrieved Nov. 18, 2025, from <https://www.w3.org/TR/did-1.0/>.
 - [17] W3C, Verifiable Credentials Data Model v2.0(2024), Retrieved Nov. 10, 2025, from <https://www.w3.org/TR/vc-data-model-2.0/>.
 - [18] H. Zang, H. Kim, and J. Kim, “Blockchain-based decentralized storage design for data confidence over cloud-native edge infrastructure,” *IEEE Access*, Vol. 12, pp. 50083 – 50099, Apr. 2024.
 - [19] P. Nguyen, H. H. Nguyen, P. Phung, H. L. Truong, and T. Cheung, “Advanced context-sensitive access management for edge-driven IoT data sharing as a service,” *ACM Transactions on Internet Technology*, Vol. 25, Issue 2, No. 9, pp. 1 – 31, Apr. 2025.
 - [20] F. T. Jaigirdar, B. Tan, C. Rudolph, and C. Bain, “Security-aware provenance for transparency in IoT data propagation,” *IEEE Access*, Vol. 11, pp. 55677 – 55691, May. 2023.
 - [21] O. Faraj, D. Megías, and J. Garcia-Alfaro, “ZIRCON: Zero-watermarking-based approach for data integrity and secure provenance in IoT networks,” *Journal of Information Security and Applications*, Vol. 85, pp. 1 – 25, Sep. 2024.

- [22] M. Howard and S. Lipner, The Security Development Lifecycle, Microsoft Press, 2006.
- [23] Wireshark Foundation, Wireshark User's Guide (2024), Retrieved Dec., 9, 2025, from <https://www.wireshark.org/docs/>.

배 태 모 (Taemo Bae)



2020년 3월~현재 : 강원대학교
컴퓨터공학과 학사

<관심분야> 네트워크 관리, 블록체인, 정보보안

최 미 정 (Mi-Jung Choi)



1998년 2월 : 이화여자대학교
공학사

2000년 2월 : 포항공과대학교
공학석사

2004년 2월 : 포항공과대학교
공학박사

2004년~2005년: 프랑스 INRIA

연구소 박사 후 연구원

2005년~2006년 : 캐나다 워털루대학 박사 후 연구

2006년~2008년 : 포항공과대학 컴퓨터공학과 연구
조교수

2008년~현재 : 강원대학교 컴퓨터학부 컴퓨터공학전
공 교수

<관심분야> 네트워크 관리, 블록체인, 정보보안

방 지 원 (Jiwon Bang)



2018년 2월 : 강원대학교 컴퓨
터과학과 졸업

2020년 2월 : 강원대학교 컴퓨
터과학과 석사

2020년 3월~현재 : 강원대학교
빅데이터메디컬융합학과 박
사

<관심분야> 네트워크 관리, 블록체인, 정보보안

황 향 희 (Hyang-Hee Hwang)



1994년 2월 : 강원대학교 체육
학사

1997년 2월 : 강원대학교 교육
학석사

2003년 2월 : 한국체육대학교
이학박사

2005년~현재 : 강원대학교 스

포츠과학과 교수

<관심분야> 스포츠여가사회학, 스포츠여가복지, 노
인체육

다중 영역 Kubernetes 환경에서 자체 복구 및 확장을 위한 자율형 PostgreSQL 오퍼레이터

타히라 아민*, 무하마드 아시프*, 우마르 마흐무드*, 송 왕 철°

Autonomous PostgreSQL Operator for Self-healing and Scaling in Multi-Domain Kubernetes Environments

Tahira Amin*, Muhammad Asif*, Umar Mahmood*, Wang-Cheol Song°

ABSTRACT

The increasing adoption of Kubernetes for cloud-native applications has intensified the need for reliable and autonomous management of stateful database services, particularly in multi-zone environments where failures and topology changes are expected. Although Kubernetes Operators offer declarative automation for managing complex workloads, their effectiveness in handling production-grade database lifecycles remains insufficiently explored. This paper presents an experimental evaluation of an autonomous PostgreSQL Operator deployed in a multi-zone Kubernetes cluster, focusing on its ability to manage key operational scenarios without manual intervention. The study investigates horizontal scaling, self-healing and automated failover, configuration-triggered zone relocation, and rolling upgrades through controlled configuration updates and fault injection. Operator behavior is analyzed using Kubernetes events, pod lifecycle transitions, and cluster state observations. Results show that scaling operations are executed in a consistency-preserving manner through synchronized replica initialization, primary failures are resolved autonomously via safe replica promotion, topology changes correctly trigger zone-aware re-orchestration, and rolling upgrades are performed sequentially while maintaining service availability. Across all scenarios, the Operator relies solely on Kubernetes-native declarative control loops to reconcile observed and desired system state. These findings demonstrate that PostgreSQL Operators can function as effective autonomous control systems for managing stateful databases in multi-zone Kubernetes environments, reducing operational complexity and supporting resilient, production-grade cloud-native deployments.

Key Words : Kubernetes Operators, Autonomous Database Management, Multi-Zone Kubernetes, PostgreSQL, Cloud-Native.

♦ First Author : Jeju National University Department of Electronic Engineering, tahiraamin95@stu.jejunu.ac.kr

° Corresponding Author : Jeju National University Department of Computer Engineering, philo@jejunu.ac.kr

* Jeju National University Department of Computer and Electronic Engineering, asif07@office.jejunu.ac.kr, umarmahmood637@gmail.com

I. Introduction

Modern cloud-native applications increasingly depend on stateful services to support reliable data management, transactional integrity, and real-time analytics in distributed environments. Despite the widespread adoption of microservices and Kubernetes-based orchestration, relational databases remain fundamental to enterprise and scientific workloads. Operating such databases in Kubernetes, however, introduces significant challenges related to availability, fault tolerance, and elastic scalability, particularly when deployments span geographically or topologically separated availability zones. In multi-zone Kubernetes environments, failures are expected operational conditions rather than rare events, ranging from individual pod or node failures to partial or complete zone outages. Manual handling of such events is slow, error-prone, and incompatible with the scale and agility required by modern cloud platforms.

Numerous approaches have attempted to address these challenges. Traditional high-availability database solutions often rely on external clustering software, static replication configurations, or infrastructure assumptions that conflict with Kubernetes' declarative and dynamic design principles. Early Kubernetes deployments frequently treated databases as fixed entities, relying on manual failover procedures and tight coupling to specific nodes. Although Kubernetes Operators introduced a more structured automation model—by encapsulating operational knowledge into controllers that continuously reconcile a desired system state with the observed runtime state—many existing solutions still focus primarily on basic lifecycle tasks such as deployment and backup. As a result, support for autonomous self-healing, zone-aware failover, and elastic scaling under realistic failure scenarios remains limited, and significant manual oversight is often required in multi-zone deployments with complex failure modes.

This gap motivates the central research question of this study: can a PostgreSQL Operator deployed in a multi-zone Kubernetes environment autonomously deliver reliable self-healing and scalable behavior suitable for production-grade stateful workloads? Beyond basic feasibility, this question also concerns whether such automation is observable, reproducible, and robust under controlled fault, scaling, relocation, and upgrade scenarios. This paper advances the hypothesis that a well-designed PostgreSQL Operator—built on Kubernetes-native primitives, declarative control loops, and zone-aware scheduling mechanisms—can function as an autonomous control system for managing stateful databases in dynamic, failure-prone environments.

In this context, a declarative control loop refers to the continuous process by which the Operator monitors cluster state through Kubernetes events and resource status, and automatically triggers corrective actions when deviations from the desired configuration are detected. Zone-aware scheduling enables database instances to be placed across distinct failure domains, such as availability zones, to improve resilience against localized disruptions. Together, these mechanisms form the foundation of the autonomous behaviors evaluated in this study.

The main contributions of this work include: (i) deployment of a PostgreSQL Operator - managed cluster in a multi-node, multi-zone Kubernetes environment; (ii) experimental evaluation of autonomous self-healing and failover behavior; (iii) analysis of horizontal scaling operations; (iv) investigation of configuration-triggered pod relocation across zones; and (v) examination of rolling update behavior during PostgreSQL upgrades. Rather than proposing a new database operator or control algorithm, this work focuses on a systematic, scenario-driven evaluation of autonomous operator behavior in a multi-zone Kubernetes environment. The contribution lies in the structured validation of declarative control loops under realistic operational conditions—such as failure recovery, topology-aware scheduling,

and rolling lifecycle events—which are often underexplored in existing operator evaluations.

The remainder of the paper is organized as follows. Section 2 reviews related work, Section 3 presents the methodology and system architecture. Section 4 reports experimental results, and Section 5 concludes the paper.

II. Related Work

The orchestration of stateful services in Kubernetes has emerged as a central research challenge as cloud-native platforms evolve toward production-grade data management. Although Kubernetes was initially designed for stateless workloads, recent studies show that managing databases introduces additional complexity related to scheduling, availability, fault tolerance, and lifecycle automation. Surveys and taxonomies of Kubernetes scheduling mechanisms highlight limitations of native schedulers in handling stateful workloads, topology awareness, and multi-objective constraints [1], [13].

Broader cloud-native research emphasizes that stateful services require orchestration mechanisms beyond basic container placement and restart semantics. Service-centric surveys underline the growing need for intelligent lifecycle management, resilience, and automation in distributed systems,

motivating the adoption of higher-level abstractions such as Kubernetes Operators and custom controllers[2]. At the same time, reliability-focused studies reveal that declarative infrastructure and control-loop-based management are prone to misconfigurations and subtle bugs, which can propagate into large-scale failures if not carefully evaluated [4], [14],[15]. Self-healing and disaster recovery have been explored through automated recovery frameworks and proactive lifecycle management strategies for stateful microservices.

These works highlight the importance of coordinated recovery and replica management, particularly in multi-cluster and failure-prone environments [6], [7]. Scaling of stateful workloads has also attracted attention, with experimental studies showing that naive horizontal scaling fails to account for replication overhead and consistency constraints [11]. Subsequent research proposes hybrid and rule-based scaling approaches to improve resilience and burst tolerance under dynamic workloads [15], [9].

Topology-aware orchestration, migration, and relocation represent another important research direction. Scheduling studies propose SLA-aware and multi-objective placement strategies, while migration frameworks enable runtime relocation of workloads across nodes or clusters using

Table 1. Comparative analysis of representative works, methodologies, and limitations in Kubernetes-based orchestration of stateful services

Ref.	Research Focus	Methodology	Key Limitation
(Carrión, 2022)	Kubernetes scheduling limitations	Survey and taxonomy	No stateful database-specific evaluation
(Deng et al., 2024)	Cloud-native service orchestration	Comprehensive survey	Broad scope, non-operator-specific
(Drosos et al., 2024)	Reliability of declarative infrastructure	Empirical fault analysis	Does not address runtime recovery
(Kim et al., 2024)	Automated disaster recovery in Kubernetes	System design and experimental evaluation	Focused on recovery only
(Meliani et al., 2025)	Proactive resiliency for stateful microservices	Experimental lifecycle management	Multi-cluster focus, not zone-specific
(Tran & Kim, 2024)	Hybrid auto-scaling for cloud/edge systems	Experimental evaluation	Serverless-oriented, not database-focused
(Dinh-Tuan & Jiang, 2025)	Stateful microservice migration	Checkpoint-based experimental study	Migration-focused, not full lifecycle

checkpointing and dynamic scheduling techniques [5], [10], [3]. Finally, rolling updates and zero-downtime deployment strategies have been studied in the context of cloud applications, highlighting trade-offs between availability and operational complexity and motivating controlled, sequential update mechanisms [12], [8]. Table 1 summarizes representative approaches to Kubernetes-based orchestration of stateful services, highlighting differences in methodology and evaluation scope.

III. Methodology and System Architecture

This section presents the methodology and system architecture of the autonomous PostgreSQL Operator evaluated in this study. The system is

designed as a Kubernetes-native control framework for managing PostgreSQL clusters deployed across multiple availability zones. The architecture emphasizes autonomous operation, observability, and reproducible evaluation under failure, scaling, and topology change scenarios. Figure 1 illustrates the overall architecture and interactions among system components.

3.1 Architectural Overview

The proposed architecture follows a control-loop-driven design, a common paradigm in Kubernetes-based systems, in which a PostgreSQL Operator continuously reconciles the desired cluster configuration with the observed runtime state of the system. In this context, the desired state is specified declaratively through custom resources, while the observed state is derived

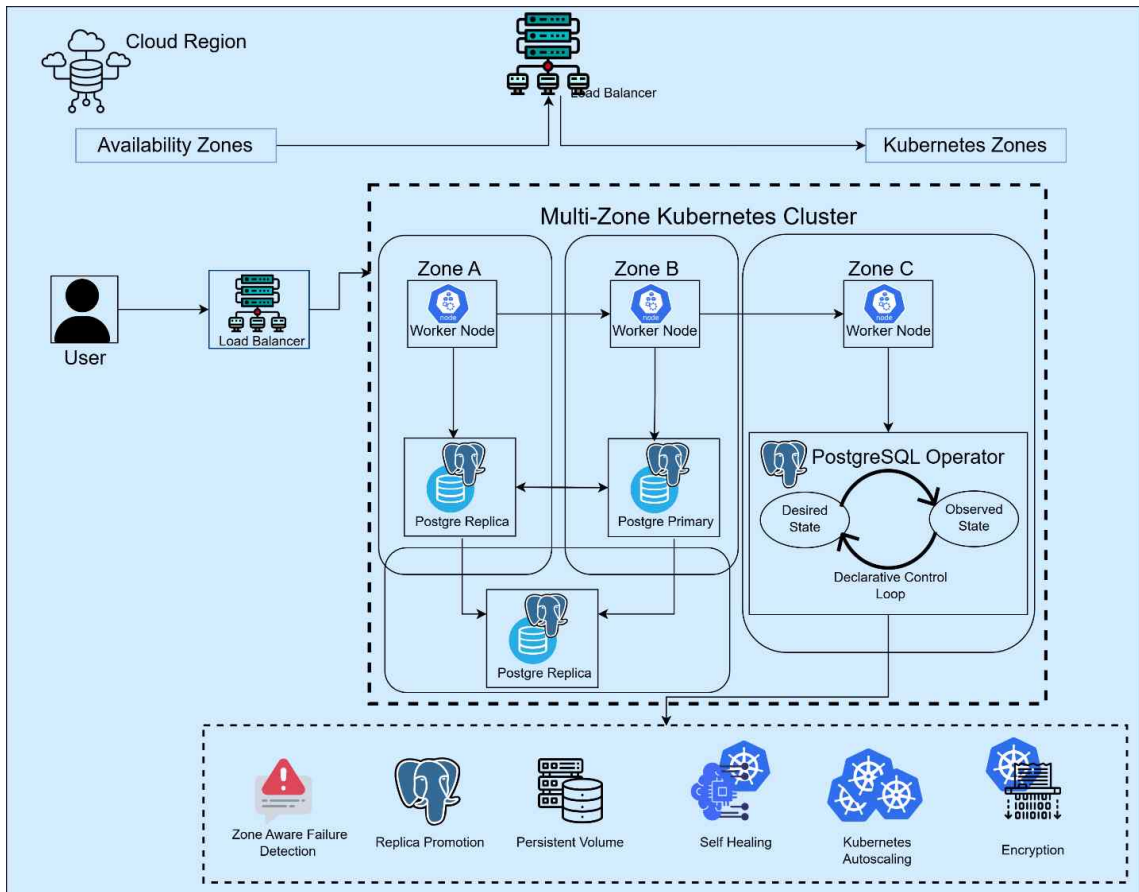


Fig. 1. Autonomous PostgreSQL Operator architecture deployed in a multi-zone Kubernetes environment

from Kubernetes events, pod status, and database health information. Any deviation between the two triggers corrective actions initiated by the Operator.

The system is deployed on a multi-zone Kubernetes cluster, where worker nodes are distributed across distinct zones to represent realistic cloud failure domains. Client access to the PostgreSQL service is provided through Kubernetes Services and an external load balancer, which transparently routes requests to the active primary instance and maintains connectivity during failover events.

At a high level, the architecture consists of four interacting layers: (i) the Kubernetes infrastructure layer, which provides scheduling and resource management; (ii) the PostgreSQL cluster deployment layer, which defines database instances and storage; (iii) the Operator control layer, which encodes autonomous management logic; and (iv) the observability and evaluation layer, which collects runtime information for analysis. These layers communicate exclusively through Kubernetes-native APIs, events, and declarative specifications, forming a closed-loop system capable of responding autonomously to failures, scaling requests, and topology changes. PostgreSQL instances are deployed using StatefulSets in a primary-replica configuration, with durable storage provided by Persistent Volumes to preserve data across restarts and role transitions.

3.2 Multi-Zone Infrastructure and Cluster Deployment

The Kubernetes cluster is configured with explicit zone labels to enable zone-aware scheduling, which allows the placement of database instances across distinct failure domains. This approach improves resilience by reducing the impact of localized node or zone failures. Persistent Volumes are provisioned using zone-scoped storage classes to ensure data locality and durability while avoiding unintended cross-zone dependencies.

Cluster properties, including replica count, resource allocation, and zone constraints are defined declaratively through custom resource definitions. These specifications serve as the authoritative input to the Operator's reconciliation logic, allowing configuration changes to be applied consistently and reproducibly without manual intervention.

3.3 PostgreSQL Operator Control Loop

The PostgreSQL Operator implements a continuous reconciliation loop that monitors pod status, replication health, storage bindings, and Kubernetes events. Failure detection is zone-aware, enabling the Operator to distinguish between localized pod or node failures and broader zone-level disruptions. When deviations from the desired state are detected, the Operator autonomously initiates corrective actions such as replica promotion, pod re-creation, or zone-aware redistribution of database instances.

Scaling operations are handled through controlled replica provisioning and deprovisioning, with synchronization and readiness checks enforced to preserve data consistency. This control-loop-based approach enables the Operator to manage complex stateful behaviors without relying on external orchestration logic or manual oversight.

3.4 Observability and Methodological Workflow

Observability components collect Kubernetes events, Operator logs, and PostgreSQL metrics to support evaluation of system behavior. These data sources provide visibility into recovery time, failover latency, replica readiness, and scaling stabilization behavior, while remaining strictly separated from the control logic to avoid influencing operational decisions.

The evaluation workflow begins with declarative cluster deployment and stabilization, followed by controlled fault injection and scaling events. System behavior is monitored until the cluster returns to a stable state, enabling repeatable and

objective assessment of autonomous self-healing and scaling behavior in a multi-zone Kubernetes environment.

3.5 Observability and Evaluation

The architecture integrates observability mechanisms to collect runtime information required for experimental analysis. Kubernetes events, Operator logs, and database-level metrics are used to evaluate recovery behavior, role transitions, and stabilization times. By maintaining a clear separation between observability and control, the experimental results reflect the inherent behavior of the Operator rather than external instrumentation effects, supporting reproducible evaluation.

3.6 Methodological Workflow

The methodological workflow begins with the declarative deployment of a PostgreSQL cluster across multiple zones, followed by stabilization and baseline metric collection. Controlled events including pod failures, primary termination, and replica scaling are then introduced to trigger autonomous responses by the Operator. System behavior is continuously observed until the cluster returns to a stable state, enabling objective assessment of self-healing effectiveness, scaling behavior, and topology-aware re-orchestration.

IV. Experimental Results

This section evaluates the autonomous behavior of the PostgreSQL Operator under controlled experimental scenarios, focusing on horizontal scaling, self-healing and failover, configuration-triggered zone relocation, and rolling upgrades. Experiments were conducted in a reproducible multi-node Kubernetes environment using declarative configuration and Kubernetes-native control mechanisms. The goal is not performance benchmarking, but systematic observation of operator-driven lifecycle management under realistic operational events. In addition to qualitative verification of state

transitions, quantitative timing indicators were extracted from Kubernetes pod lifecycle timestamps and Operator event logs to summarize the responsiveness of autonomous operations.

4.1 Experimental Environment and Baseline

Experiments were performed on a Kubernetes cluster created using `kind`, consisting of one control-plane node and two worker nodes. To emulate a multi-zone deployment, worker nodes were labeled as belonging to separate zones, enabling evaluation of zone-aware scheduling without geographically distributed infrastructure. The CloudNativePG PostgreSQL Operator was deployed to manage a three-instance PostgreSQL cluster defined declaratively through a custom resource. Each instance was provisioned with persistent storage and node affinity constraints. After deployment, the cluster stabilized with all pods in a running and ready state, forming the baseline for subsequent experiments.

4.2 Horizontal Scaling

Horizontal scaling was evaluated by increasing the desired replica count from three to five through a declarative configuration update. Upon applying the updated cluster specification, the PostgreSQL Operator automatically initiated the scaling process without manual intervention. The operator first created temporary join pods to synchronize data between the existing cluster members and the newly provisioned replicas. During this phase, newly created pods transitioned through initialization and synchronization states before becoming active members of the cluster. After the data synchronization phase completed, all PostgreSQL pods reached the Running and Ready state, resulting in a stable five-instance cluster configuration. Based on Kubernetes pod lifecycle timestamps and event ordering, the data synchronization phase required approximately 25 - 30 seconds, while the full scaling stabilization time measured from configuration update to all replicas becoming ready was approximately 40 -

45 seconds. This behavior demonstrates that the operator performs controlled and consistency-preserving scaling by enforcing synchronization and readiness checks before exposing new replicas to the cluster.

Table 2 summarizes the observed outcomes and stabilization time of the horizontal scaling experiment, as derived from Kubernetes pod lifecycle events and Operator logs.

Table 2. Observed outcomes for the horizontal scaling experiment

Scaling Attribute	Observed Outcome
Initial replica count	3
Target replica count	5
Scaling trigger	Declarative configuration update
Temporary join pods	Created for data synchronization
Data synchronization duration	~25 - 30 s
Full stabilization time	~40 - 45 s
Manual intervention required	No

4.3 Self-Healing and Failover

Self-healing behavior was examined by deliberately deleting the active primary PostgreSQL pod identified through cluster status inspection. Upon failure injection, the PostgreSQL Operator detected the primary unavailability and autonomously promoted a healthy replica to the primary role while recreating the failed pod, without requiring manual intervention.

Based on Kubernetes event ordering and pod lifecycle timestamps, failure detection occurred within approximately 5 - 7 seconds, and replica promotion was completed within 10 - 12 seconds after detection. Full cluster recovery, including recreation of the failed pod and readiness verification, was achieved within approximately 30 - 35 seconds. The cluster returned to a healthy state with all pods running and ready, confirming the Operator's ability to maintain service availability through autonomous failover.

Table 3 summarizes the observed outcomes and recovery characteristics of the self-healing and

Table 3. Observed outcomes for the self-healing and failover experiment

Failover Attribute	Observed Outcome
Failure type	Primary pod deletion
Failure detection time	~5 - 7 s
Replica promotion time	~10 - 12 s
Failed pod re-creation	Automatically initiated
Full recovery time	~30 - 35 s
Manual intervention required	No
Service availability	Maintained during failover

failover experiment.

4.4 Zone Relocation

To evaluate topology-aware relocation, node affinity constraints were modified to target a different availability zone through a declarative configuration update. After applying the updated configuration, the PostgreSQL Operator autonomously terminated existing pods and re-created PostgreSQL instances on nodes matching the new zone label. During this re-orchestration process, temporary initialization and join pods were created to ensure correct data synchronization and role assignment.

Based on Kubernetes event ordering and pod lifecycle timestamps, pod re-creation and relocation required approximately 20 - 25 seconds, while full cluster stabilization in the target zone—measured from configuration update to all pods reaching a running and ready state—was achieved within approximately 35 - 40 seconds. These observations demonstrate correct enforcement of declarative topology constraints and confirm the Operator's ability to perform autonomous, zone-aware re-orchestration without manual intervention.

Table 4 summarizes the observed outcomes and stabilization characteristics of the zone relocation experiment.

4.5 Rolling Upgrades

Rolling upgrade behavior was evaluated by updating the PostgreSQL container image from

Table 4. Observed outcomes for the zone relocation experiment

Relocation Attribute	Observed Outcome
Relocation trigger	Node affinity configuration update
Source zone	Original deployment zone
Target zone	Updated affinity zone
Pod re-creation initiated	Yes
Temporary initialization/join pods	Observed during relocation
Pod relocation duration	~20 - 25 s
Full stabilization time	~35 - 40 s
Manual intervention required	No

version 15 to version 16 through a declarative configuration change. Upon applying the updated specification, the PostgreSQL Operator autonomously initiated a rolling upgrade procedure, sequentially restarting individual pods. Each PostgreSQL instance was upgraded only after the previously updated pod reached a running and ready state, ensuring that service availability was maintained throughout the process. The upgrade completed successfully with all PostgreSQL pods running the updated version, which was verified through in-database version checks. No manual intervention was required during the upgrade, and no service interruption was observed. These results confirm that the Operator correctly enforces ordered pod restarts and readiness checks to support safe and autonomous rolling upgrades.

Table 5 summarizes the observed outcomes of the rolling upgrade experiment.

Table 5. Observed outcomes for the rolling upgrade experiment

Upgrade Attribute	Observed Outcome
Upgrade trigger	PostgreSQL image update (v15 → v16)
Upgrade strategy	Sequential pod restarts
Readiness enforcement	Yes (one pod at a time)
Service availability	Maintained throughout upgrade
Manual intervention required	No
Upgrade verification	In-database version check
Final cluster state	All pods running PostgreSQL v16

Across all experiments, the PostgreSQL Operator exhibited consistent autonomous behavior aligned

with its declarative control model. Scaling, failover, relocation, and upgrades were handled without manual intervention and within bounded stabilization times, validating the Operator's capability to manage stateful database lifecycles in a multi-zone Kubernetes environment using Kubernetes-native mechanisms.

Table 6 summarizes the observed stabilization times for autonomous scaling, failover, relocation, and upgrade operations extracted from Kubernetes event logs and pod lifecycle transitions.

Table 6. Summary of observed stabilization times for autonomous PostgreSQL Operator operations

Scenario	Trigger Event	Observed Stabilization Time
Horizontal scaling	Replica count update (3 → 5)	~40 - 45 s
Self-healing / failover	Primary pod deletion	~30 - 35 s
Zone relocation	Node affinity update	~35 - 40 s
Rolling upgrade	PostgreSQL image update (v15 → v16)	Sequential, no service interruption

V. Conclusion and Future Work

This paper presented a systematic evaluation of an autonomous PostgreSQL Operator deployed in a multi zone Kubernetes environment, with the goal of understanding how declarative, Kubernetes native control mechanisms can be used to manage stateful database lifecycles. A range of operational scenarios was examined, including horizontal scaling, self healing and automated failover, configuration driven zone relocation, and rolling upgrades, in order to observe how the Operator responds to dynamic conditions without manual intervention.

The experimental results show that scaling operations progress in a controlled and consistency preserving manner, primary failures are handled autonomously through safe replica promotion, and topology changes correctly trigger zone aware relocation of database instances. In addition, rolling upgrades were carried out sequentially

while maintaining service availability, indicating that routine version transitions can be supported without requiring full service disruption.

Overall, these findings suggest that PostgreSQL Operators can function as effective autonomous control systems for managing stateful databases in Kubernetes, particularly in multi zone environments where reconciliation logic, topology awareness, and storage integration must work together correctly. The experiments were conducted using a kind based Kubernetes cluster with a limited number of nodes, which allowed for controlled execution and clear observation of operator behavior. While this environment is sufficient for validating reconciliation logic and operational correctness, it does not fully reflect the scale, infrastructure diversity, or network characteristics typically found in production cloud deployments.

When deployed in larger clusters or managed Kubernetes platforms such as GKE or EKS, additional factors including inter zone network latency, cloud managed storage behavior, autoscaling interactions, and higher replica counts may influence failover and recovery characteristics. Even so, because the PostgreSQL Operator relies on Kubernetes native declarative reconciliation mechanisms, the qualitative behaviors observed in this study are expected to carry over to real world environments. Future work will focus on evaluating performance under load, comparing different operator implementations, and extending the analysis to larger or geographically distributed deployments.

References

- [1]. Carrión, C. (2022). Kubernetes Scheduling: Taxonomy, Ongoing Issues and Challenges. *ACM Comput. Surv.*, 55(7), 138:1-138:37. <https://doi.org/10.1145/3539606>
- [2]. Deng, S., Zhao, H., Huang, B., Zhang, C., Chen, F., Deng, Y., Yin, J., Dustdar, S., & Zomaya, A. Y. (2024). Cloud-Native Computing: A Survey From the Perspective of Services. *Proceedings of the IEEE*, 112(1), 12 - 4 6 . <https://doi.org/10.1109/JPROC.2024.3353855>
- [3]. Dinh-Tuan, H., & Jiang, J. (2025). Optimizing Stateful Microservice Migration in Kubernetes with MS2M and Forensic Checkpointing. 2025 28th Conference on Innovation in Clouds, Internet and Networks (ICIN), 83 - 90. <https://doi.org/10.1109/ICIN64016.2025.10942720>
- [4]. Drosos, G.-P., Sotiropoulos, T., Alexopoulos, G., Mitropoulos, D., & Su, Z. (2024). When Your Infrastructure Is a Buggy Program: Understanding Faults in Infrastructure as Code Ecosystems. *Reproduction Package for Article: "When Your Infrastructure Is a Buggy Program: Understanding Faults in Infrastructure as Code Ecosystems*, 8(OOPSLA2), 3 5 9 : 2 4 9 0 - 3 5 9 : 2 5 2 0 . <https://doi.org/10.1145/3689799>
- [5]. Farid, M., Lim, H. S., Lee, C. P., Zarakovitis, C. C., & Chien, S. F. (2025). Optimizing Kubernetes with Multi-Objective Scheduling Algorithms: A 5G Perspective. *Computers*, 14(9), 390. <https://doi.org/10.3390/computers14090390>
- [6]. Kim, J.-B., Choi, J.-B., & Jung, E.-S. (2024). Design and Implementation of an Automated Disaster-Recovery System for a Kubernetes Cluster Using LSTM. *Applied Sciences*, 14(9), 3914. <https://doi.org/10.3390/app14093914>
- [7]. Meliani, A. E., Mekki, M., & Ksentini, A. (2025). Resiliency focused proactive lifecycle management for stateful microservices in multi-cluster containerized environments. *Computer Communications*, 236, 108111. <https://doi.org/10.1016/j.comcom.2025.108111>
- [7]. Mondal, S. K., Zheng, Z., & Cheng, Y. (2024). On the Optimization of Kubernetes toward the Enhancement of Cloud Computing. *Mathematics*, 12(16), 2476. <https://doi.org/10.3390/math12162476>

-
- [8]. Pakrijauskas, K., & Mažeika, D. (2025). A Rule-Based Method for Enhancing Burst Tolerance in Stateful Microservices. *Electronics*, 14(14), 2752. <https://doi.org/10.3390/electronics14142752>
- [9]. Pashaeheir, A., Shariati, S., Shafaghi, S., Moghimi, M., & Momtazpour, M. (2025). KubeDSM: A Kubernetes-based Dynamic Scheduling and Migration Framework for Cloud-Assisted Edge Clusters (No. arXiv:2501.07130). *arXiv*. <https://doi.org/10.48550/arXiv.2501.07130>
- [10]. Perera, H. C. S., De Silva, T. S. D., Wasala, W. M. D. C., Rajapakshe, R. M. P. R. L., Kodagoda, N., Samararatunge, U. S. S., & Jayanandana, H. H. N. C. (2021). Database Scaling on Kubernetes. 2021 3rd International Conference on Advancements in Computing (ICAC), 258 - 263. <https://doi.org/10.1109/ICAC54203.2021.9671185>
- [11]. Rudrabhatla, C. K. (2020). Comparison of zero downtime based deployment techniques in public cloud infrastructure. 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 1082 - 1086. <https://doi.org/10.1109/I-SMAC49090.2020.9243605>
- [12]. Senjab, K., Abbas, S., Ahmed, N., & Khan, A. ur R. (2023). A survey of Kubernetes scheduling algorithms. *Journal of Cloud Computing*, 12(1), 87. <https://doi.org/10.1186/s13677-023-00471-1>
- [13]. Sun, X., Luo, W., Gu, J. T., Ganesan, A., Alagappan, R., Gasch, M., Suresh, L., & Xu, T. (2022). Automatic Reliability Testing For Cluster Management Controllers. 143 - 159. <https://www.usenix.org/conference/osdi22/presentation/sun>
- [14]. Tran, M.-N., & Kim, Y. (2024). Optimized resource usage with hybrid auto-scaling system for knative serverless edge computing. *Future Generation Computer Systems*, 152, 304 - 316. <https://doi.org/10.1016/j.future.2023.11.010>
- [15]. Xu, Q., Gao, Y., & Wei, J. (2024). An Empirical Study on Kubernetes Operator Bugs. *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, 1746 - 1758. <https://doi.org/10.1145/3650212.36>

Tahira Amin



2018 September : B.S in
Computer Science,
COMSATS University
Islamabad, Pakistan
2024 March : M.S in
Computer Science,
COMSATS University

Islamabad, Pakistan

2025 September - Present: Ph.D. in Electronic Engineering, Jeju National University, Republic of Korea.

<Field of Interest> Natural Language Processing (NLP), Large Language Models (LLMs), Kubernetes and Container Orchestration, Kubernetes Operators

Muhammad Asif



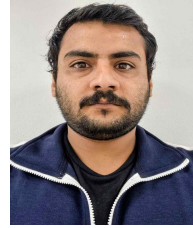
2015 September : B.S in
Computer Software
Engineering, Abasyn
University Peshawar,
Pakistan
2020 September: M.S in
Computer Software
Engineering, University of

Engineering and Technology Peshawar, Pakistan

2014 January - Present: Ph.D. in Computer Engineering, Jeju National University, Republic of Korea.

<Field of Interest> Agentic AI-driven autonomous and intent-based systems, LLMs, NLP, STGNNs, zkML, cloud-native computing with Kubernetes, SDN, secure and privacy-preserving network automation, and Space-Air-Ground Integrated Networks.

Umar Mahmood



2022 July: B.E, in Computer
Software Engineering.
National University of
Science and Technology,
Pakistan
2024 September - Present:
Ph.D. in Electronics

Engineering, Jeju National University, Republic of Korea

<Field of Interest> Field of interest: computer network, cloud networking, IBN, network automation

Wang-cheol Song



1986 February: B.S in Food
Engineering, Yonsei
University, Republic of
Korea

1989 February: B.S in
Electronic
Engineering,
Yonsei University,
Republic of Korea.

1991 February: M.S in Electronic Engineering, Yonsei University, Republic of Korea.

1995 August: Ph.D. in Electronic Engineering, Yonsei University, Republic of Korea.

<Field of Interest> VANETs and MANETs, SDN/NFV, Intent-based networking, and Network management.

스테이블코인의 기술적 구조 및 동향 분석

강 창 훈*, 홍 원 기°

Analysis of Technical Structure and Trends of Stablecoins

Changhoon Kang*, James Won-Ki Hong°

요 약

블록체인 기술을 기반으로 한 스테이블코인은 디지털 자산의 변동성 문제를 해결하고, 탈중앙화 금융과 전통 금융을 연결하는 핵심 인프라로 자리 잡았다. 최근 스테이블코인은 활용 범위가 결제, 송금, 금융 상품 등으로 급격히 확대되고 있다. 그러나 기존 연구들은 주로 거시경제적 영향이나 통화 정책적 함의를 분석하는 데 편중되어 있어, 스테이블코인의 안정성과 신뢰성을 보장하는 기술적 실체에 대한 체계적인 분석은 부족한 실정이다. 본 논문은 스테이블코인을 복합적인 시스템으로 정의하고, 이를 기술적 관점에서 해부하기 위한 새로운 분석 프레임워크를 제안한다. 제안하는 프레임워크는 시스템의 정적 구조를 정의하는 4계층 아키텍처와 동적 운영 흐름을 설명하는 5단계 생애주기로 구성된다. 본 연구는 이 프레임워크를 적용하여 스테이블코인 시스템 전반에서 발생하는 핵심적인 기술적 트레이드오프를 분석하는 예시를 제공한다. 나아가 스테이블코인이 단순한 가치 저장 수단을 넘어, 국경 간 송금, B2B 결제에서의 활용, 원화 기반 모델을 통한 특화 산업의 발전, 그리고 AI 에이전트 기반의 머신 이코노미를 실현하는 프로그래밍 가능한 화폐로서의 잠재력을 설명한다.

Key Words : Stablecoin, Blockchain, Smart Contract, Digital Asset

ABSTRACT

Stablecoins based on blockchain technology have established themselves as a core infrastructure that resolves the volatility issues of digital assets and bridges decentralized finance (DeFi) with traditional finance. Recently, the utility of stablecoins has rapidly expanded across payments, remittances, and financial products. However, as existing studies have predominantly focused on macroeconomic impacts or monetary policy implications, there remains a lack of systematic analysis regarding the technical entities that guarantee the stability and reliability of stablecoins. This paper defines stablecoins as complex systems and proposes a new analytical framework to dissect them from a technical perspective. The proposed framework consists of a 4-Layer Architecture defining the system's static structure and a 5-Stage Lifecycle describing its dynamic operational flow. By applying this framework, this study provides examples of analyzing key technical trade-offs that occur throughout the stablecoin ecosystem. Furthermore, this paper explores the potential of stablecoins as a programmable currency that goes beyond a simple store of value. It discusses their applications in cross-border remittances and B2B payments, the advancement of specialized industries through KRW-based models, and the realization of the Machine Economy driven by AI agents.

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

* First Author : POSTECH, Department of Computer Science and Engineering, chkang@postech.ac.kr

° Corresponding Author : POSTECH, Department of Computer Science and Engineering, jwkhong@postech.ac.kr

논문번호 : KNOM2025-02-015, Received November 16, 2025; Revised December 3, 2025; Accepted December 21, 2025

I. 서론

최근 4차 산업혁명의 핵심 기술로 꼽히는 블록체인은 데이터의 무결성을 보장하는 분산 원장 기술로서, 중앙 관리자 없이도 참여자 간의 신뢰를 형성할 수 있는 혁신적인 기반을 제공한다. 이 기술을 바탕으로 등장한 비트코인 (Bitcoin) [1]이나 이더리움 (Ethereum) [2]과 같은 암호화폐는 국경과 시간의 제약 없는 자유로운 가치 이전을 가능하게 했으나, 극심한 가격 변동성으로 인해 실질적인 교환 수단이나 가치 저장 수단으로 활용되는 데에는 근본적인 한계가 있었다. 이러한 배경에서 등장한 스테이블코인 (Stablecoin)은 블록체인의 기술적 이점인 투명성과 보안성을 유지하면서도, 달러 (USD)와 같은 법정화폐 (Fiat Currency)에 가치를 연동하여 가격 안정성을 확보함으로써 디지털 자산과 전통 금융을 잇는 가교 역할을 수행하고 있다.

스테이블코인은 높은 변동성을 가진 디지털 자산 시장에서 가치 저장 및 교환의 매개체 역할을 수행하며, 오늘날 글로벌 금융 시스템의 핵심 인프라로 자리 잡았다. 2024년 기준 스테이블코인의 총 거래량은 27.6조 달러에 달하며 [3], 이는 비자 (Visa)와 마스터카드 (Mastercard)의 합산 처리량을 능가하는 경제적 규모이다. 초기에는 단순히 암호화폐 거래소 내의 기축 통화로 사용되었으나, 현재는 탈중앙화 금융 (DeFi) 생태계의 유동성 공급부터 국경 간 송금 및 결제에 이르기까지 그 활용 범위가 지속적으로 확대되고 있다.

그러나 이러한 양적 성장에도 불구하고, 스테이블코인의 실제 활용 실태를 살펴보면 대부분의 거래가 차익거래나 자금 이동 등 금융 자동화 활동에 편중되어 있다는 특징을 보인다. 이에 따라 스테이블코인에 대한 기존 연구들은 주로 거시경제적 영향이나 금융 안정성, 그리고 통화 정책적 함의에 집중되어 왔다. 반면, 실제 스테이블코인의 작동을 가능하게 하는 프로토콜 아키텍처, 스마트 컨트랙트의 구현 방식, 그리고 온체인 거버넌스 모델 등 기술적 근간에 대한 기술적 분석은 상대적으로 부족한 실정이다.

본 논문은 이러한 배경 하에, 스테이블코인의 기술적 구조와 최신 동향을 체계적으로 분석하고자 한다. 이를 위해 본 연구는 스테이블코인의 기술적 메커니즘을 해부하는 새로운 분석 프레임워크를 제안한다. 이 프레임워크는 시스템을 L1: 인프라 (보

안/데이터), L2: 프로토콜 (발행/안정화), L3: 통합 (상호운용성), L4: 감독 및 거버넌스 (규제/감사)의 네 가지 핵심 계층으로 구분하며, 본 논문은 이 계층들이 설계, 발행 및 운영, 위험 관리, 규제 준수, 업그레이드라는 5단계 생애주기 전반에서 어떻게 유기적으로 상호작용하는지 기술적 관점에서 규명한 다.

나아가 본 연구는 이러한 기술적 아키텍처가 실제 스테이블코인 활용의 성패를 결정짓는 핵심 변수임을 제안한다. 국경 간 송금이나 B2B 결제와 같이 이미 알려진 활용 사례들조차, 블록체인 네트워크의 확장성이나 상호운용성 표준과 같은 구체적인 기술적 요건이 충족되지 않을 경우 실효성을 갖기 어렵다. 특히, 이러한 기술적 완성도는 최근 많은 관심이 집중되고 있는 머신 이코노미, 즉 AI 에이전트 기반의 자율 경제 환경에서 더욱 중요해진다. 본 논문은 스테이블코인이 단순한 가치 저장 수단을 넘어, 수많은 온오프라인 활용처에서 프로그래밍 가능한 금융 인프라로서 기능할 수 있음을 설명한다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 제2장에서는 연구 배경과 관련 연구를 기술한다. 제3장에서는 본 연구의 핵심 분석 프레임워크를 제안하고 제4장에서 이 프레임워크를 적용하여 시스템의 신뢰성, 확장성, 규제 준수 관점에서 발생하는 기술적 트레이드오프를 일부 분석한다. 제5장에서는 스테이블코인의 실질적 활용 방안을 논의하고, 마지막으로 제6장에서 결론을 맺는다.

II. 연구 배경 및 관련 연구

2.1. 연구 배경

스테이블코인이라는 개념이 최근에 갑자기 등장한 것은 아니다. 2014년 테더 (Tether, USDT) [4]의 등장을 시작으로, 암호화폐 시장의 높은 변동성을 회피하기 위한 온체인 기축통화로서 지난 10여년간 꾸준히 존재해 왔다. 초기 스테이블코인은 주로 중앙화된 암호화폐 거래소 내에서 법정화폐 입출금의 불편함을 해소하거나, 트레이딩을 위한 유동성 공급 수단으로 활용되는 등 그 역할이 제한적이었다.

그러나 최근 2~3년 사이 스테이블코인은 단순한 거래소용 칩을 넘어 글로벌 금융 결제망의 핵심 인프라로 급부상하고 있다. 이러한 변화의 기폭제로는

미국의 정책적 기조 변화를 꼽을 수 있다. 트럼프 행정부의 출범과 함께 디지털 자산에 대한 규제 명확성 확보와 친화적인 산업 육성 정책이 예고되면서, 달러 패권을 디지털 영역으로 확장하려는 미국의 전략적 움직임이 가시화되었다. 이는 페이팔 (PayPal) [5], 스트라이프 (Stripe) [6]와 같은 글로벌 핀테크 거대 기업들이 자체 스테이블코인을 발행하거나 결제 수단으로 도입하게 만드는 촉매제가 되었으며, 전통 금융 기관들이 블록체인의 결제 및 자산 시스템에 진입하는 진입 장벽을 낮추는 결과를 낳았다.

이에 발맞추어 전 세계적으로 스테이블코인 도입 경쟁이 가속화되고 있다. 유럽, 싱가포르, 일본 등 주요 금융 선진국들은 자국 통화 기반의 스테이블코인 발행을 위한 법적 기틀을 마련하고 있으며, 국내에서도 토큰 증권 (STO) [7]과 연계된 디지털 화폐 논의가 활발히 진행 중이다. 결과적으로 스테이블코인은 이제 틈새시장의 실험적 도구가 아니라, 국가 간 무역 결제, 송금, 그리고 기업 간 (B2B) 정산에 활용되는 실질적인 금융 상품으로 진화하고 있다.

2.2. 글로벌 규제 환경 및 국가별 도입 현황

스테이블코인의 시장 영향력이 확대됨에 따라, 전 세계 주요국은 금융 안정성과 사용자 보호를 위한 법적 프레임워크를 마련하고 있다. 각국의 규제는 기술적 요구사항 (상환 주기, 준비금 구성 등)과 거버넌스 모델 (발행 주체 제한)에 직접적인 영향을 미치며, 이는 스테이블코인 아키텍처 설계 시 반드시 고려해야 할 제약 조건으로 작용한다. 주요국의 규제 현황은 다음과 같다.

- 1) 미국: 미국은 달러 패권 유지를 목적으로 가장 적극적인 입법 활동을 펼치고 있다. 2025년 7월 서명된 GENIUS Act (Governing the Emission of Nationally Issued Utility Stablecoins Act) [8]는 연방 차원의 명확한 법적 지위를 확립하였다. 해당 법안은 알고리즘 기반 스테이블코인을 금지하며, 발행 자격을 은행의 자회사 또는 연방/주 정부의 승인을 받은 비은행 기관으로 제한한다. 특히 빅테크 기업의 시장 독점을 방지하기 위해 기술 기업의 발행에는 스테이블코인 인증 심의 위원회 (SCRC)의 만장일치 승인을 요구하는 등 높은 진입 장벽을 설정하고 있다.
- 2) 유럽연합 (EU): EU는 2024년부터 포괄적 암

호자산 규제법인 MiCA (Markets in Crypto Assets) [9]를 시행 중이다. MiCA는 스테이블코인을 단일 통화로 연동된 전자화폐 토큰 (EMT)과 자산 바스켓에 연동된 자산준거 토큰 (ART)으로 명확히 분류한다. 기술적 요구사항으로는 발행사가 EU 내에 법인을 설립해야 하며, 발행량과 1:1로 매칭되는 현금성 준비금을 보유하고 이에 대한 정기적인 보고를 의무화하고 있다.

- 3) 일본: 일본은 2023년 개정 자금결제법 (Payment Services Act) [10]을 통해 세계 최초로 스테이블코인 발행을 합법화하였다. 발행 주체는 은행, 신탁회사, 자금이체업자로 엄격히 제한되며, 발행자, 중개자, 사용자 간의 책임 구조를 법적으로 명확히 규정하였다. 알고리즘 스테이블코인은 별도의 암호자산 규제를 적용받는다.
- 4) 홍콩: 홍콩통화청 (HKMA)은 2024년 스테이블코인에 대한 규제 프레임워크를 확정하고 2026년부터 라이선스 제도를 의무화할 예정이다. [11] 발행사는 홍콩 내 법인을 설립해야 하며, 정기적인 준비금 감사와 대중 공시가 필수적이다. 홍콩 역시 알고리즘 기반 스테이블코인의 발행을 금지하고 있다.
- 5) 싱가포르: 싱가포르 통화청 (MAS)은 2024년 하반기부터 싱가포르 달러 (SGD) 또는 G10 통화로 연동된 스테이블코인에 대한 규제를 시행하였다. [12] 핵심적인 기술적 요구사항은 발행사가 5영업일 이내에 액면가로 상환 (Par-value redemption) 할 수 있는 유동성을 확보해야 한다는 점이다. 준비금은 현금 또는 단기 국채로 100% 구성되어야 한다.
- 6) 베트남: 베트남은 2026년 발효 예정인 디지털 기술 산업법 (Law on Digital Technology Industry) [13]을 통해 가상자산을 법제화할 예정이다. 현재는 결제 수단으로 금지되어 있으나, 2026년부터 다낭 등 특정 지역에서 샌드박스 프로그램을 통해 스테이블코인 결제 시범 사업을 추진하는 등 제도권 도입을 모색하고 있다.
- 7) 대한민국: 한국은 2024년 가상자산 이용자 보호법 [14]을 시행하였으나, 스테이블코인에 대한 직접적인 규제는 아직 입법 논의 단계이다. 현재 국회에는 디지털자산기본법 [15]을 비롯하여 금융 안정성, 산업 진흥 등 다양한

목적은 둔 여러 법안들이 발의되어 논의 중이다. 한국은행은 자본 유출과 통화 정책 약화를 우려하여 기업이 아닌 은행 중심의 발행 모델을 선호하는 등 신중한 입장을 취하고 있다.

주요국들의 스테이블코인 규제 현황과 주요 기술적 제약사항을 비교하면 아래 표 1과 같다.

표 1. 주요국 스테이블코인 규제 프레임워크 비교
Table 1. Comparison of Stablecoin Regulatory Frameworks in Major Countries

국가/지역	관련 법안	발행 주체 요건
미국	GENIUS Act (2025)	은행, 자회사, 인가 비은행 기관
EU	MiCA (2024)	EU 내 설립 법인
일본	자금결제법 (2023)	은행, 신탁회사, 자금이체업자
홍콩	HKMA 프레임워크 (2024)	홍콩 내 법인 (라이선스 필요)
싱가포르	MAS 규제안 (2024)	MAS 인가 발행사
베트남	디지털 기술 산업법 (2026 예정)	(샌드박스 참여 기업)
대한민국	(관련 법안 논의 중)	(은행 vs. 민간 기업 논의 중)

이와 같이 각국은 자국의 금융 환경에 맞춰 상이한 규제를 적용하고 있다. 따라서 글로벌 호환성을 갖춘 스테이블코인 아키텍처를 설계하기 위해서는 이러한 규제적 파편화를 기술적으로 수용할 수 있는 유연한 구조가 필수적이다.

2.3. 관련 연구

스테이블코인에 관한 기존 학술 연구는 대부분 경제학적 분석이나 금융 정책적 관점에 집중되어 있다. 선행 연구들은 주로 스테이블코인의 가격 유지 메커니즘이 거시경제 지표에 미치는 영향, 통화 정책의 유효성 위협, 그리고 뱅크런 시나리오에서의 시스템 리스크 분석 등을 다루었다 [16-23]. 특히 테라-루나 (Terra-Luna) [24] 사태 이후에는 알고리즘 기반 스테이블코인의 실패 원인을 게임 이론이나 금융 기술적으로 분석한 연구들이 다수 발표되

었다.

반면, 스테이블코인의 실체를 기술적 관점에서 분석한 연구는 상대적으로 부족한 실정이다. 기존의 기술적 연구들은 주로 스마트 컨트랙트의 취약점 탐지나 블록체인 네트워크의 성능 측정과 같은 단편적인 요소 기술에 국한되어 있으며, 스테이블코인 시스템 전체를 아우르는 프로토콜 아키텍처, 오픈체인 자산과 온체인 토큰 간의 동기화 메커니즘, 규제 준수를 위한 기술적 제어 구조 등을 통합적으로 분석한 시도는 찾아보기 어렵다. 기술적 구성요소 하나의 차이더라도 스테이블코인 시스템에서는 안정성, 효율성 등이 크게 달라질 수 있게 때문에 이를 통합적으로 관리하는 것은 매우 중요하다.

따라서 본 연구는 경제학적 논의에 편중된 기존 연구의 한계를 넘어, 법정화폐 기반 스테이블코인을 하나의 거대한 소프트웨어 아키텍처로 정의하고, 그 내부의 기술적 구조와 설계 결정사항을 체계적으로 분석함으로써 이러한 연구 공백을 메우고자 한다.

III. 스테이블코인의 기술적 분석

본 장에서는 스테이블코인의 기술적 정의와 시장 현황을 살펴보고, 이들의 복잡한 작동 원리를 기술적으로 해부하기 위한 새로운 분석 프레임워크를 제안한다.

3.1. 스테이블코인의 정의 및 시장 현황

3.1.1. 정의 및 핵심 기술 요소

스테이블코인은 암호화폐의 본질적인 문제인 높은 가격 변동성을 해결하기 위해 고안된 디지털 자산으로, 법정화폐나 금과 같은 안정적인 기준 자산에 가치를 연동하도록 설계되었다. 스테이블코인은 가치를 유지하는 담보에 따라 크게 세 가지 유형으로 분류된다.

- 1) 법정화폐 기반 (Fiat-backed): 미국 달러와 같은 법정화폐를 1:1 비율로 준비금으로 보유하여 가치를 보증하는 방식이다. USDT (Tether)와 USDC (Circle) [25]가 대표적이며, 가장 직관적이고 안정적이나 발행 주체에 대한 신뢰가 필요하다.
- 2) 암호화폐 기반 (Crypto-backed): 이더리움 (ETH) 등 변동성 있는 암호화폐를 담보로 맡기고 스마트 컨트랙트를 통해 스테이블코인을 발행하는 방식이다. 담보 가치 하락에 대비해

일반적으로 초과 담보를 설정하며, DAI (MakerDAO) [26]가 대표적인 사례다.

- 3) 알고리즘 기반 (Algorithmic): 별도의 담보 자산 없이, 알고리즘이 시장 수요에 따라 토큰 공급량을 자동으로 조절하여 가격을 유지하는 방식이다. 탈중앙화된 구조를 지향하나, 테라루나 (Terra-Luna) 사태에서 보듯 신뢰 붕괴 시 죽음의 소용돌이 (Death Spiral)로 인해 가치가 급락할 위험이 있다.

스테이블코인은 데이터 저장을 담당하는 블록체인 네트워크 위에서 발행 및 소각 로직을 수행하는 스마트 컨트랙트를 통해 토큰의 형태로 발행된다. 이러한 스테이블코인 토큰의 소유권, 전송 정보가 블록체인에 기록되는 방식으로 동작하는 것이다. 동일한 스테이블코인이라도 다양한 종류의 블록체인 상에서 동시에 발행하여 운영되는 경우가 대부분이다.

이러한 온체인 토큰과 오프체인 준비금을 비교하는 예치금 증명 시스템 (Proof-of-Reserves) 역시 스테이블코인 관련 논의에 항상 고려되는 핵심요소이다. 발행자가 스테이블코인을 다시 준비 자산으로 교환해줄 수 있는 준비금을 발행된 가치 이상으로 갖고 있는지 항상 확인이 필요하며, 현재까지 대부분 회계 법인을 통한 외부 감사로 이 과정이 진행되고 있다. 최근 ChainLink [27]에서는 이 회사가 가진 오라클 기술을 통해 준비금 정보를 온체인 상으로 보내 연동하는 서비스를 공개하기도 했다.

3.1.2. 스테이블코인 시장 점유율

스테이블코인 시장은 담보 유형에 따라 다양하게 분류되나, 현재 시장은 법정화폐 기반 스테이블코인이 압도적인 우위를 점하고 있다. 법정화폐 기반 모델은 현재 전체 스테이블코인 시장 점유율의 90% 이상을 차지한다. 2025년 2월 기준, 법정화폐 기반 스테이블코인의 총 공급량은 약 2,250억 달러에 달하며, 이는 시장의 표준 모델로 자리 잡았음을 시사한다. [28]

특히, 이러한 법정화폐 기반 스테이블코인 중에서도 미국 달러 (USD)에 가치가 고정된 스테이블코인의 비중은 절대적이다. 시가총액 기준으로 전체 시장의 99% 이상이 미국 달러에 페깅되어 있으며, 전체 시장 가치인 약 2,000억 달러 중 95% 이상이 달러를 앵커 (Anchor)로 삼고 있다. [29] 이는 스테이블코인이 사실상 달러의 온체인 확장 역할을 수행하고 있음을 보여준다.

법정화폐 기반 스테이블코인 시장은 테더 (Tether)의 USDT와 서클 (Circle)의 USDC라는 두 가지 거대 프로젝트에 의해 주도되고 있다. 2025년 2월 기준, 두 코인의 합산 점유율은 약 88.5%에 이른다. USDT는 2014년 출시 이후 시장을 선점해왔으며, 2025년 5월 기준 유통량이 1,500억 달러를 초과하는 등 약 64%의 점유율로 시장 1위를 유지하고 있다. 가장 풍부한 유동성을 바탕으로 주로 암호화폐 거래소의 기축 통화로 활용된다. USDC는 약 24.5%의 점유율을 차지하며, 정기적인 준비금 증명 공시와 규제 당국의 감시 하에 투명성을 강조하는 전략을 취한다. 이러한 특성 덕분에 탈중앙화 금융 (DeFi) 프로토콜이나 기업 간 결제 (B2B) 등 신뢰가 중요한 비즈니스 영역에서 선호된다.

이러한 통계는 스테이블코인 생태계가 다양한 블록체인 (이더리움, 트론, 솔라나 등) 위에서 발행되지만, 그 가치 안정성의 근간은 중앙화된 달러 준비금과 소수의 발행사에 집중되어 있다는 구조적 중앙화를 명확히 보여준다.

3.2. 제안하는 계층형 분석 프레임워크

기존 연구들은 스테이블코인을 단순히 경제적 가치가 고정된 토큰으로만 취급하는 경향이 있었다. 그러나 기술적 관점에서 스테이블코인은 블록체인 인프라, 스마트 컨트랙트 로직, 외부 시스템 연동, 그리고 거버넌스 프로세스가 복합적으로 상호작용하는 거대 소프트웨어 시스템이다. 따라서 본 논문은 스테이블코인의 기술적 실체를 명확히 규명하기 위해 스테이블코인 생애주기별 계층형 분석 프레임워크를 제안한다. (그림 1)

이 프레임워크는 시스템의 정적 구조를 정의하는 4계층 아키텍처와 동적 운영 흐름을 설명하는 5단계 생애주기 모델로 구성되며, 이 둘의 상호작용을 통해 스테이블코인 시스템의 복잡성을 체계적으로 분석한다.

3.2.1. 4계층 아키텍처

스테이블코인 시스템의 기술 스택은 기능적 역할과 추상화 수준에 따라 다음과 같이 네 개의 논리적 계층으로 구분된다.

- 1) L1 - 인프라 레이어 (Infrastructure Layer): 시스템의 최하단에 위치하며, 트랜잭션의 최종 확정 (Finality), 데이터 가용성 (Data Availability), 그리고 보안 합의 (Consensus)를 담당하는 기반 블록체인 네트워크 계층이

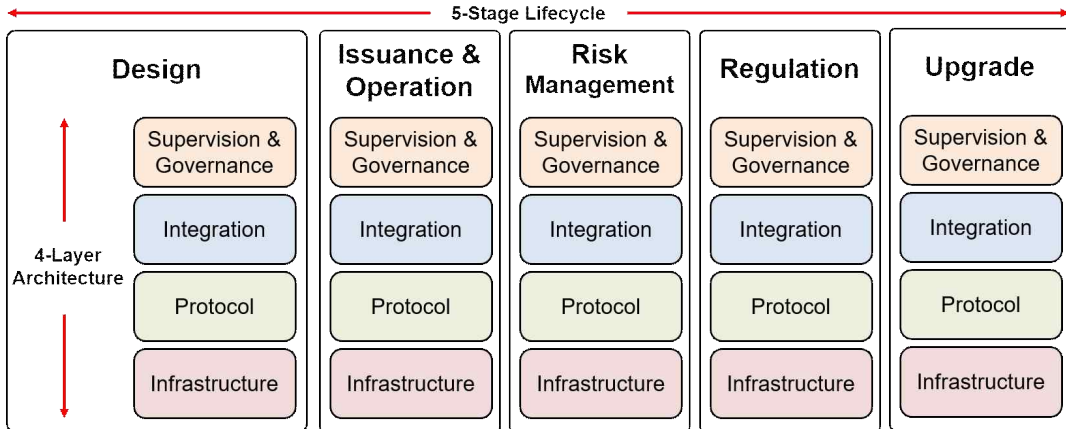


그림 1. 스테이블코인 생애주기별 계층형 분석 프레임워크
Fig. 1. A Layered Analytical Framework Across the Stablecoin Lifecycle

다. 이더리움 (Ethereum), 솔라나 (Solana) 등 메인넷의 선택은 스테이블코인의 확장성 (Scalability), 탈중앙성 (Decentralization), 보안성 (Security)이라는 블록체인 트릴레마 속성을 결정짓는다.

- 2) L2 - 프로토콜 레이어 (Protocol Layer): 스테이블코인의 핵심 비즈니스 로직이 구현되는 스마트 컨트랙트 계층이다. 담보 자산의 정의, 토큰의 발행 (Minting) 및 소각 (Burning) 메커니즘, 가격 안정화 모듈 (PSM), 그리고 청산 엔진 (Liquidation Engine) 등 가치 유지를 위한 핵심 알고리즘이 이곳에 존재한다.
- 3) L3 - 통합 레이어 (Integration Layer): 서로 다른 블록체인 네트워크 간의 자산 이동 등의 상호운용성 (Interoperability)을 보장하는 계층이다. 크로스체인 브릿지 (Cross-chain Bridges), 래핑된 자산 (Wrapped Assets) 표준, 그리고 체인 간 메시징 프로토콜이 포함되며, 유동성 파편화 문제를 해결하는 역할을 수행한다.
- 4) L4 - 감독 및 거버넌스 레이어 (Supervision & Governance Layer): 기술적 시스템과 현실 세계의 규제 요구사항을 연결하는 최상위 계층이다. 준비금 증명 그리고 비상 상황 시 프로토콜을 제어하는 거버넌스 (DAO 또는 중앙화된 관리자) 모듈이 포함된다.

3.2.2. 5단계 생애주기 모델

스테이블코인 시스템은 정지된 상태가 아니라, 시

간의 흐름에 따라 설계, 운영, 위기 대응, 규제 준수, 그리고 진화의 과정을 거친다. 본 연구는 이를 다음과 같이 5단계의 생애주기 (Lifecycle)로 정의하고 각 단계별 기술적 특성을 분석한다.

- 1) 설계 (Design): 시스템의 방향성을 결정하는 초기 단계로, 향후 수정하기 어려운 핵심 아키텍처가 확정된다. L1에서는 퍼블릭 대 프라이빗 체인 선택과 합의 알고리즘에 따른 보안 모델이 정의되며 이에 따라 거래 처리 속도, 송금 수수료 등이 결정된다. L2에서는 담보 유형 (법정화폐, 암호화폐, 알고리즘)과 가격 유지 메커니즘이 설계되고, L3에서는 멀티체인 확장 전략이, L4에서는 규제 준수 요건을 반영한 거버넌스 모델이 수립된다.
- 2) 발행 및 운영 (Issuance & Operation): 시스템이 실제 가동되어 사용자와 상호작용하는 단계이다. L2의 스마트 컨트랙트를 통해 사용자의 담보 예치에 따른 토큰 발행과 소각이 실행되며, L1 인프라는 트랜잭션 처리와 노드 운영을 담당한다. L3에서는 브릿지를 통한 체인 간 유동성 라우팅이 일어나며, L4에서는 자동화된 감사 로그 생성과 같은 일상적인 감독 활동이 수행된다.
- 3) 위험 관리 (Risk Management): 시장 변동성 확대, 해킹 시도, 디페깅 (De-pegging) 등 시스템 위협에 대응하는 단계이다. L2의 청산 엔진이 가동되어 부실 담보를 정리하고 페깅을 유지하며, L1에서는 DDoS 공격 방어 및

노드 복구가 이루어진다. L3에서는 브릿지 해킹 등 보안사고 시 자산 이동을 차단하는 비상조치가, L4에서는 위기 대응 거버넌스 투표 및 감독 기관 통보 절차가 실행된다.

- 4) 규제 준수 (Compliance): 외부 규제 기관의 요구사항을 기술적으로 이행하는 단계이다. L4 레이어가 주도적으로 작동하며, 실시간 준비금 증명 데이터 생성, 자금세탁방지를 위한 특정 주소 동결, 그리고 규제 리포팅을 위한 감사 데이터 추출이 수행된다. L1과 L2는 이러한 규제 조치가 위변조 불가능하게 기록되도록 데이터 무결성을 보장하는 역할을 한다.
- 5) 업그레이드 (Upgrade): 기술 발전과 시장, 규제의 변화에 맞춰 시스템을 진화시키는 단계이다. L2 스마트 컨트랙트의 로직 개선이나 새로운 담보 자산 추가, L1 네트워크의 하드 포크 (Hard Fork) 대응, L3의 새로운 크로스체인 표준 도입 등이 포함된다. L4에서는 변경된 규제 환경에 맞춰 거버넌스 정책과 규칙 셋 (Rule Set)을 갱신한다.

본 논문은 이어지는 장에서 이 4계층 아키텍처와 5단계 생애주기가 교차하는 지점에서 발생하는 핵심적인 설계 결정사항과 기술적 트레이드오프를 분석한다.

IV. 프레임워크 기반 분석

앞서 제안한 4계층 아키텍처와 5단계 생애주기 모델을 교차 분석하면, 스테이블코인 시스템이 해결해야 할 기술적 문제들이 드러난다. 본 장에서는 시스템의 성패를 좌우하는 가장 결정적인 세 가지 기술적 요소인 신뢰성, 확장성, 그리고 규제 준수를 중심으로 핵심 설계 결정사항과 기술적 트레이드오프를 분석한다.

4.1. 신뢰성: 담보 모델과 준비금 증명

스테이블코인의 가장 본질적인 요구사항은 가격 안정성이나, 이를 구현하는 과정에서 자본 효율성과 신뢰성 사이의 상충 관계가 발생한다. L2 프로토콜 레이어의 설계 단계에서 발행사는 담보 모델을 결정해야 하는데, 법정화폐 담보 모델은 1:1 폐경의 신뢰도가 가장 높다는 장점이 있으나 오프체인 은행에 대한 신뢰 비용이 발생한다는 한계가 있다. 반면 암호화폐 담보 모델은 온체인 투명성을 보장하지만, 담보 가치 변동에 따른 청산 리스크를 방지하

기 위해 과도한 초과 담보를 요구하여 자본 효율성이 떨어진다. (L2 - 설계, 발행 및 운영)

또한 운영 및 규제 준수 단계에서는 실시간 투명성 확보가 핵심 과제로 대두된다. 전통적인 월간 회계 감사는 데이터의 시차로 인해 급변하는 시장 상황에서 완전한 신뢰를 보장하기 어렵다. 이에 대한 기술적 해법으로 도입되고 있는 준비금 증명 기술은 오라클을 통해 오프체인 은행 계좌의 잔고를 실시간으로 온체인에 증계하고, 스마트 컨트랙트가 온체인 발행량이 오프체인 준비금보다 작거나 같다는 조건을 매 트랜잭션마다 검증하도록 강제한다. 이는 사후 감사 중심의 신뢰 모델을 실시간 기술적 검증 모델로 전환시키는 핵심적인 진보이다. (L4 - 발행 및 운영, 규제 준수)

4.2. 확장성: 멀티체인 전략과 브릿지 리스크

스테이블코인 발행사는 발행한 스테이블코인이 널리 사용되어 수요가 증가하고 더 많이 발행할수록 쌓인 준비금 운용을 통해 높은 수익을 얻을 수 있다. 이 때문에 다양한 블록체인 상에서 높은 활용도를 통해 많이 사용되도록 하는 것은 필연적인 도전과제이다. 하지만 수많은 블록체인들 중 특히 초기에 어떤 체인 상에서의 발행과 활용에 집중할지 선택하는 것은 스테이블코인의 성공 여부를 결정하는 중요한 전략이 된다. 블록체인 네트워크마다 다양한 측면의 장단점이 있고 이에 따른 핵심 애플리케이션들이 다르기 때문에 스테이블코인 프로젝트가 추구하는 목적에 따라 적절한 체인을 선택해야 한다. 예를 들어, 이더리움과 같이 탈중앙화 및 보안성이 검증된 네트워크는 높은 신뢰성과 규제 친화성을 제공하지만, 느린 거래 처리 속도와 높은 수수료 측면에서 단점이 있다. 반면 솔라나와 같은 고성능 네트워크는 높은 TPS와 낮은 비용이라는 장점을 가지나, 검증 노드 구성의 상대적 중앙화 리스크가 존재한다. 따라서 스테이블코인 발행사는 대상 시장이 기관 간 대규모 정산인지 소매 결제 중심인지에 따라 인프라 레이어 선택이 달라질 수 있다. (L1 - 설계)

운영 단계에서 크로스체인 상호운용성 전략 역시 중요한 설계 변수이다. 둘 이상의 블록체인에서 동시에 발행 또는 유통하고자 할 때 이를 어떻게 기술적으로 구현할 것인지에 관한 전략을 의미한다. 래핑 (Wrapped) 자산 방식은 소스 체인에서 자산을 락업하고 목적 체인에서 대응 토큰을 발행하는 Lock&Mint 구조로 구현이 용이하고 초기에 유동성

을 빠르게 조성할 수 있으나, 브리지 스마트컨트랙트, 검증자 등 특정 컴포넌트에 의존하기 때문에 단일 실패점으로 이어질 수 있으며 과거 해킹 사례의 주요 공격 대상이 되었다. 반면 CCTP와 같은 네이티브 Burn&Mint 방식은 소스 체인에서 자산을 소각하고 목적 체인에서 동일한 양을 발행하는 구조로, 체인 간 유동성을 통합하고 자산 파편화를 해소하며 자본 효율성과 보안성이 높다는 평가를 받는다. 최근 기술적 흐름은 보안성과 규제 대응, 유동성 일원화를 중시하며 네이티브 방식 채택이 확대되는 방향으로 이동하고 있다. (L3 - 발행 및 운영, 위험 관리)

4.3. 규제 준수: 통제 메커니즘

L4 감독 레이어의 규제 준수 단계는 블록체인의 핵심 이념인 검열 저항성과 가장 극명하게 충돌하는 지점이다. 미국 GENIUS Act나 EU MiCA와 같은 규제 프레임워크를 준수하기 위해 법정화폐 기반 스테이블코인은 스마트 컨트랙트 내에 블랙리스트 및 자산 동결 함수를 필수적으로 포함해야 한다. 이는 발행사가 특정 주소의 자산 이동을 원천 차단하거나 강제로 소각할 수 있는 권한을 부여하는 것으로, 탈중앙화된 네트워크 위에서 고도로 중앙화된 통제를 가능케 한다. (L2, L4 - 규제 준수)

이러한 백도어 기능은 규제 당국의 요구사항을 충족시키지만, 사용자 입장에서 자산 동결 리스크라는 새로운 위협이 된다. 이를 완화하기 위해 최근에는 다중 서명 관리자 기능을 도입하거나, 자산 동결의 조건과 절차를 스마트 컨트랙트에 명시하여 자의적인 권한 행사를 방지하는 프로그래밍 가능한 거버넌스 모델이 연구되고 있다. 결국 스테이블코인의 아키텍처는 기술적 탈중앙성과 법적 규제 준수 사이에서 최적의 균형점을 찾아가는 과정에 있다.

V. 스테이블코인의 활용 방안

앞서 제안한 기술적 아키텍처와 생애주기 모델은 결국 스테이블코인이 실물 경제와 디지털 경제 내에서 실질적인 효용을 창출하기 위한 기반이다. 본 장에서는 스테이블코인이 현재 글로벌 금융 시장에서 활용될 수 있는 일반적인 사례를 분석하고, 나아가 원화 (KRW) 기반 스테이블코인이 도입될 경우의 특화된 활용 시나리오와 미래 머신 이코노미 환경에서의 역할을 제안한다.

5.1. 글로벌 금융 인프라로서의 일반적 활용

스테이블코인의 가장 본질적인 활용 가치는 블록체인의 효율성과 법정화폐의 안정성을 결합하여 전통 금융 시스템의 비효율성을 해소하는 데 있다. 첫째, 국경 간 결제 및 송금 분야에서의 혁신이다. 기존 SWIFT망을 통한 해외 송금은 중계 은행을 거치며 2~5일의 시간과 높은 수수료가 소요되나, 스테이블코인은 24시간 연중무휴로 즉각적인 정산이 가능하며 수수료를 획기적으로 절감시킨다. 이는 비자와 같은 전통 결제 기업이 스테이블코인을 정산 통화로 채택하여 은행 간 정산 프로세스를 효율화하는 사례로 증명되고 있다.

둘째, 탈중앙화 금융 생태계와 전통 금융의 연결 고리 역할이다. 스테이블코인은 높은 변동성을 가진 암호화폐 시장에서 신뢰할 수 있는 교환 매개이자 가치 저장 수단으로 기능하며, 거래소 유동성의 60% 이상을 책임지는 기축 통화 역할을 수행한다. 또한 최근에는 미국 단기 국채와 같은 실물 자산 (RWA)을 담보로 하여 보유자에게 직접 이자 수익을 제공하는 수익형 스테이블코인이 등장함에 따라, 온체인 자본이 전통 자산 시장으로 유입되는 핵심 경로로 자리 잡고 있다.

셋째, 경제적 불안정성에 대한 헤지 수단이다. 베네수엘라나 아르헨티나와 같이 자국 통화의 변동성이 극심한 고인플레이션 국가에서 스테이블코인은 누구나 접근 가능한 디지털 달러로서 기능하며 자산 가치를 보존하는 역할을 한다. 이는 인터넷 연결만으로 금융 서비스에 접근할 수 있게 하여 금융 포용성을 확대하는 사회적 효용을 창출한다.

5.2. 원화 기반 스테이블코인의 특화 활용

현재 시장을 지배하는 달러 기반 스테이블코인과 달리, 원화 기반 스테이블코인은 한국의 산업적 특성과 결합하여 고유한 부가가치를 창출할 수 있다. 우선 도매 및 무역 금융의 효율화이다. 동대문이나 남대문 시장과 같은 대형 도매 시장에서 활동하는 외국인 상인들은 환전과 현금 처리의 불편함을 겪고 있다. 원화 스테이블코인은 이러한 B2B 거래에서 환전 비용 없는 실시간 정산을 가능케 하며, 수출입 기업 간의 무역 대금 결제 프로세스를 투명하고 신속하게 개선할 수 있다. 현재 이미 우리나라를 방문하는 외국인들이 전통 시장에서 스테이블코인을 통해 값을 지불하거나, 스테이블코인을 원화 현금으로 바꿀 수 있는 ATM이 설치되는 등의 실제 사례

를 찾아볼 수 있다.

또한 K-콘텐츠 및 관광 산업과의 시너지 효과를 기대할 수 있다. 전 세계적으로 확산된 K-Pop 팬덤이나 웹툰 독자들은 국가별 결제 시스템의 장벽으로 인해 콘텐츠 구매나 굿즈 결제에 어려움을 겪는 경우가 많다. 원화 스테이블코인은 이러한 국경 간 소액 결제를 원활하게 지원하는 글로벌 결제 수단 이 될 수 있다. 이미 해외의 일부 이커머스들은 스테이블코인을 통한 지불을 도입하고 있고, 관련 기업들의 인수합병과 빠른 성장이 진행되고 있다. 의료 관광 분야에서도 외국인 환자들이 성형외과나 피부과 시술 비용을 원화 스테이블코인으로 결제함으로써 환율 변동 위험을 줄이고, 세금 환급 절차를 스마트 컨트랙트로 자동화하여 투명성을 높이는 등 행정 비용을 절감할 수 있다.

5.3. 머신 이코노미와 AI 에이전트의 자율 결제
최근 AI 기술의 고도화로 인간의 개입 없이 기계 간 상호작용이 이루어지는 머신 이코노미가 부상함에 따라, 스테이블코인은 AI 에이전트를 위한 핵심적인 금융 인프라로 주목받고 있다. 기존의 신용카드나 은행 계좌 시스템은 신원 인증과 복잡한 승인 절차를 요구하므로, 자율적으로 판단하고 행동하는 AI 에이전트가 사용하기에는 기술적 장벽이 존재한다. 또한 AI가 수행하는 API 호출이나 데이터 구매와 같은 초소액 결제를 처리하기에는 기존 결제망의 수수료 구조가 비효율적이다.

스테이블코인은 이러한 한계를 극복하고 웹의 초기 비전이었던 HTTP 402 (Payment Required) 프로토콜 [30]을 실현할 수 있는 최적의 수단이다. 스테이블코인이 스마트 컨트랙트를 기반으로 발행되는 프로그래밍 가능한 디지털 화폐라는 특성 덕분에, AI 에이전트는 별도의 은행 방문 없이 자체적으로 지갑을 생성하여 스테이블코인을 보유할 수 있으며, 스마트 컨트랙트를 통해 사전 정의된 조건이 충족되면 즉시 자금을 전송하는 자율적인 경제 활동이 가능하다. 이는 향후 자율주행차의 충전 요금 자동 결제, AI 비서의 서비스 구독 관리, IoT 기기 간의 데이터 거래 등 기계가 주체가 되는 경제 생태계에서 스테이블코인이 기축 통화로서 필수적인 역할을 수행할 것임을 시사한다. 실제로 코인베이스 (Coinbase)와 클라우드플레어 (CloudFlare)는 협력

을 통해 이를 실현할 수 있는 x402 표준을 발표하였고, 이를 활용한 거래 역시 점차 증가하고 있다.

VI. 결 론

본 논문에서는 스테이블코인의 기술적 구조와 동향에 대한 포괄적인 분석을 제공하였다. 이를 통해 스테이블코인이 단순한 암호화폐 토큰을 넘어 블록체인 인프라와 스마트 컨트랙트, 그리고 외부 규제 시스템이 결합된 복합적인 시스템을 보였다. 기존의 경제학적 측면에만 머물러 있던 논의를 기술적 설계 관점으로 초점을 변경하여 4계층 아키텍처와 5단계 생애주기 모델을 포함하는 새로운 분석 프레임워크를 제안하였다.

나아가 본 논문은 이러한 기술적 기반 위에서 스테이블코인이 국경 간 송금과 B2B 결제의 효율성을 혁신할 뿐만 아니라, 원화 기반 모델을 통한 특화 산업의 발전, 그리고 AI 에이전트 기반의 머신 이코노미를 실현하는 프로그래밍 가능한 화폐로서의 잠재력을 설명하였다.

향후 연구에서는 본 논문에서 정립한 프레임워크를 심화 확장하여, 4계층 아키텍처와 5단계 생애주기가 교차하는 각 지점에서의 핵심 기술 동향과 현존하는 한계점에 대해 자세히 분석하고자 한다. 단순히 구조를 도식화하는 것을 넘어, 각 계층과 생애주기 단계가 맞물릴 때 발생하는 구체적인 기술적 요건을 식별하고, 이를 해결하기 위해 시도되고 있는 최신 기법들을 심도 있게 다룰 것이다. 아울러, 현재 글로벌 시장에서 널리 사용되고 있는 USDT, USDC, DAI 등 실제 스테이블코인 프로젝트들을 제안한 프레임워크를 통해 분석하는 실증적 연구를 진행할 계획이다. 또한, 대표적인 실패 사례로 뽑히는 테라-루나 사태의 UST 프로젝트 역시 본 프레임워크를 통한 모델링을 통해 프로젝트 내 산재하고 있던 위험성에 대해 분석을 진행할 예정이다. 주요 프로젝트들의 기술적 설계 결정과 운영 메커니즘을 본 프레임워크에 대입하여 비교 분석함으로써, 각 프로젝트의 기술적 특성을 명확히 규명하고 제안된 프레임워크가 실제 시스템을 진단하고 설계하는 데 있어 유효한 도구임을 보일 것이다.

References

- [1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
<https://bitcoin.org/bitcoin.pdf>
- [2] V. Buterin, "A next-generation smart contract and decentralized application platform," Ethereum White Paper, 2014.
<https://ethereum.org/en/whitepaper/>
- [3] S. Feingold, "Stablecoin surge: Reserve-backed cryptocurrencies are on the rise," World Economic Forum, Mar. 26, 2025.
<https://www.weforum.org/stories/2025/03/stable-coins-cryptocurrency-on-rise-financial-systems/>
- [4] Tether Operations Limited, "Tether (USDT) Official Website," [Online]. Available: <https://tether.to>
- [5] PayPal Holdings, Inc., "PayPal USD (PYUSD): A U.S. dollar-denominated stablecoin," Product documentation, 2023. [Online]. Available: <https://www.paypal.com/>
- [6] Stripe, Inc., "Stripe Crypto and stablecoin payments," Product documentation, 2024. [Online]. Available: <https://stripe.com/>
- [7] Financial Services Commission (FSC), Republic of Korea, "Regulatory guidelines for security token offerings and tokenized securities," Policy Statement, Feb. 2023. [Online]. Available: <https://www.fsc.go.kr/>
- [8] 119th U.S. Congress, "S.394 – GENIUS Act of 2025," Public Law, 2025. [Online]. Available: <https://www.congress.gov/bill/119th-congress/senate-bill/394/text>
- [9] European Union, "Regulation (EU) 2023/1114 of the European Parliament and of the Council on Markets in Crypto-assets (MiCA)," Official Journal of the European Union, 2023; stablecoin rules effective June 2024. [Online]. Summarized at: <https://www.eba.europa.eu/>
- [10] Financial Services Agency (FSA), Japan, "Amendments to the Payment Services Act and related cabinet orders concerning stablecoins," 2023. [Online]. Available: <https://www.fsa.go.jp/>
- [11] Hong Kong Monetary Authority, "Stablecoins Ordinance and licensing and regulatory framework for stablecoins," first gazetted Dec. 2024. [Online]. Available: <https://www.hkma.gov.hk/>
- [12] Monetary Authority of Singapore, "MAS finalises stablecoin regulatory framework for single-currency stablecoins pegged to SGD or G10 currencies," Framework announced Aug. 2023. [Online]. Available: <https://www.mas.gov.sg/>
- [13] National Assembly of Vietnam, "Law on Digital Technology Industry," Law No. XX/2024/QH15. [Online]. Overview: <https://www.vietnam-briefing.com/>
- [14] National Assembly of the Republic of Korea, "Act on the Protection of Virtual Asset Users," Act No. 19564, Jul. 18, 2023. [Online]. Available: https://elaw.klri.re.kr/eng_mobile/viewer.do?hseq=63752&type=part&key=23
- [15] B. D. Min et al., "Basic Act on Digital Assets," Bill No. 2210736, 22nd National Assembly, Republic of Korea, Jun. 2025. [Online]. Available: https://watch.peoplepower21.org/index.php?mid=LawInfo&bill_no=2210736
- [16] D. Bullmann, J. Klemm, and A. Pinna, "In search for stability in crypto-assets: are stablecoins the solution?," ECB Occasional Paper, no. 230, 2019.
- [17] R. K. Lyons and G. Viswanath-Natraj, "What keeps stablecoins stable?," The Journal of Finance, vol. 78, no. 4, pp. 2091 – 2135, 2023.
- [18] D. W. Arner, R. Auer, and J. Frost, "Stablecoins: risks, potential and regulation," Bank for International Settlements (BIS) Working Papers, no. 905, 2020.
- [19] G. B. Gorton and J. Zhang, "Taming wildcat stablecoins," The University of Chicago Law Review, vol. 90, pp. 905 – 958, 2021.
- [20] B. Eichengreen, "From commodity to fiat and now to crypto: what does history tell us?,"

NBER Working Paper, no. w25426, 2019.

- [21] A. Klages-Mundt, D. Harz, L. Gudgeon, A. R. Minca, and W. J. Knottenbelt, "Stablecoins 2.0: Economic foundations and risk-based models," in Proceedings of the 2nd ACM Conference on Advances in Financial Technologies (AFT), 2020, pp. 59 - 79.
- [22] A. Briola, D. Vidal-Tomás, Y. Wang, and T. Aste, "Anatomy of a stablecoin's failure: The Terra-Luna case," Finance Research Letters, vol. 51, p. 103358, 2023.
- [23] I. Aldasoro, A. Cornelli, M. Ferrari Minesso, L. Gambacorta, and M. Habib, "Stablecoins, money market funds and monetary policy," BIS Bulletin, no. 87, 2024.
- [24] H. Liu, I. Makarov, and A. Schoar, "Anatomy of a run: The Terra Luna crash," NBER Working Paper, no. w31160, 2023.
- [25] Circle Internet Financial, "USD Coin (USDC)," [Online]. Available: <https://www.circle.com/en/usdc>
- [26] MakerDAO, "MakerDAO Official Website," [Online]. Available: <https://makerdao.com>
- [27] Chainlink, "Chainlink Official Website," [Online]. Available: <https://chain.link>
- [28] The Block, "Total Stablecoin Supply," The Block Data Dashboard, Feb. 2025. [Online]. Available: <https://www.theblock.co/data/decentralized-finance/stablecoins>
- [29] DefiLlama, "Stablecoin Pegs and Market Share Breakdown," DefiLlama Data Dashboard, Feb. 2025. [Online]. Available: <https://defillama.com/stablecoins>
- [30] R. Fielding, M. Nottingham, and J. Reschke, Eds., "HTTP Semantics," Internet Engineering Task Force (IETF), RFC 9110, Jun. 2022. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc9110.html#name-402-payment-required>

강 창 훈 (Changhoon Kang)



2019 포항공과대학교, 컴퓨터 공학과 학사
 2019 ~ 현재 포항공과대학교, 컴퓨터공학과 석박통합과정
 <관심분야> 블록체인, 암호화폐, 결제 시스템, 스테이블 코인, 에이전틱 AI

홍 원 기 (James Won-Ki Hong)



1983 Univ. of Western Ontario, BSc in Computer Science
 1985 Univ. of Western Ontario, MS in Computer Science
 1985 ~ 1986 Univ. of Western Ontario, Lecturer
 1986 ~ 1991 Univ. of Waterloo, PhD in Computer Science
 1991 ~ 1992 Univ. of Waterloo, Post-Doc Fellow
 1992 ~ 1995 Univ. of Western Ontario, 연구교수
 1995 ~ 현재 포항공과대학교, 컴퓨터공학과 교수
 2007 ~ 2011 포항공과대학교, 정보통신대학원장
 2007 ~ 2010 포항공과대학교, 정보통신연구소 연구 소장
 2008 ~ 2010 포항공과대학교, 컴퓨터공학과 주임교수
 2008 ~ 2012 포항공과대학교, 정보전자융합공학부장
 2012 ~ 2014 KT 종합기술원, 원장
 <관심분야> 네트워크 및 시스템 관리, 생성형 AI, 블록체인 및 암호화폐

한국통신학회 통신망운용관리연구회

KNOM Review Vol.28/ No.2

ISSN : 2287-1543(Online)

2025년 12월 31일 인쇄

2025년 12월 31일 발행

발행인/ 김우태

편집인/ 김명섭 편집위원장

발행처/ 한국통신학회 통신망운용관리연구회
서울시 서초구 서초동 1330-8 현대기림오피스텔 1504동 6호
전화 : 02-3453-5555
홈페이지 : www.knom.or.kr/knom-review/