https://doi.org/10.7840/kics.2025.50.2.366

액터 모델 기반 5G 네트워크 데이터 분석 기능 시뮬레이터 구현

최 은 혜•, 최 다 영*, 성 지 훈**, 신 명 기**, 박 형 곤°

Implementation of 5G Network Data Analytics Function Simulator Based on Actor Models

Eunhye Choi•, Dayoung Choi*, Jihoon Sung**, Myungki Shin**, Hyunggon Park°

요 약

5G 네트워크에서 NWDAF(Network Data Analytics Function)는 NF(Network Function)들과 사용자 장치(UE, User Equipment)로부터 수집된 다양한 데이터 기반 분석 정보를 NF들에게 제공하는 핵심 기능을 수행한다. 이를 통해 NF는 더 정확한 의사 결정을 수행하고 최적의 네트워크 상태를 유지할 수 있도록 한다. 본 논문에서는 5G 코어 네트워크에서 NWDAF와 NF들 간의 상호작용을 시뮬레이션 할 수 있는 액터(actor) 모델 기반의 시뮬레이터 구현 방법을 제안한다. 각 NF는 독립적인 액터로 구현되며, 이들 간의 상호작용은 메시지를 통해 이루어진다. 이벤트 기반의 메시지는 NF의 통신에 사용되는 SBI(Service Based Interface)를 통해 교환될 수 있다. NF를 액터로 정의하고 이벤트 기반 메시지와 전송 방법을 설계하였으며, 이를 기반으로 NWDAF 분석 정보를 교환하기 위한 시뮬레이터를 구현했다. DDoS(Distributed Denial of Service) 공격 시나리오를 활용하여 NWDAF가 네트워크 트래픽의 이상을 효과적으로 탐지하고 이를 NF에게 실시간으로 통보하는 기능을 검증했다. 검증 결과, 액터로 모델링 된 NWDAF는 분석 결과를 다른 NF에게 정확하게 전달할 수 있으며, NF는 수신한 분석 결과를 이용하여 의사 결정을 내릴 수 있음을 확인했다. 이는 제안된 시뮬레이터를 다양한 다른 네트워크 분석 결과 전달 시나리오에도 손쉽게 확장하여 적용될 수 있음을 보여준다.

키워드 : 네트워크 데이터 분석 기능, 네트워크 기능, 액터 모델, 시뮬레이터
Key Words : NWDAF, NF, actor-model, simulator

ABSTRACT

NWDAF plays a crucial role in 5G networks by providing analytics based on network data collected from other NFs, enabling them to make autonomous decisions and maintain optimal network conditions. In this paper, we propose a method for implementing an actor model-based simulator that simulates interactions between NWDAF and NFs in the 5G core network. Each NF is implemented as an independent actor and

※ 본 연구는 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(2021-0-00739)과 한국연구재단의 지원(NRF-2020R1A2B5B01002528)을 받아 수행되었습니다.
◆ First Author : Agency for Defense Development, Ewha Womans University, Department of Electronic and Electrical Engineering, eunhye.choi@ewha.ac.kr, 정회원
° Corresponding Author : Ewha Womans University, Department of Electronic and Electrical Engineering, hyunggon.park@ewha.ac.kr, 종신회원
* Ewha Womans University, Department of Electronic and Electrical Engineering, dayoung.choi@ewha.ac.kr, 학생회원
** Electronics and Telecommunications Research Institute, jh.sung@etri.re.kr, 정회원; mkshin@etri.re.kr, 정회원
논문번호 : 202407-161-C-RN, Received July 29, 2024; Revised October 2, 2024; Accepted October 21, 2024

interacts with other NFs through messages. Event-based messages can be served as the SBI (Service Based Interface) for inter-NF communications. We define NFs as actors and outline the method for transmitting event-based messages. Based on this, we implemented a simulator to facilitate the exchange of NWDAF analytics. The simulator was validated using a DDoS (Distributed Denial of Service) attack scenario, demonstrating the effectiveness of NWDAF in detecting network traffic anomalies and reporting them to NFs in real-time. We also confirm that the actor-modeled NWDAF can accurately deliver analytics results to other NFs, enabling them to make decisions. This approach demonstrates that it can be easily extended and applied to different scenarios involving the transmission of network analytics.

Ⅰ. 서 론

5G-Advanced 및 6G 네트워크는 현대 사회에서 디지털 변환을 이끄는 주요 기반 기술로 초고속 데이터 전송, 초저지연 통신, 대규모 연결성을 제공하여 자율주행 자동차, 스마트 시티, 산업 자동화 등 기술과 서비스의 범위를 크게 넓혔다. 이런 기술들과 더불어 폭발적으로 증가하는 대용량 데이터 트래픽 처리와 복잡한 네트워크의 효율적인 관리를 위한 네트워크의 자동화 및 지능화는 필수적이다.

이동통신 시스템 표준화 단체인 3GPP(3rd Generation Partnership Project)는 5G 표준에서 네트워크 자동화 및 지능화를 지원하기 위한 기능으로 NWDAF (Network Data Analytics Function)를 정의했다. NWDAF는 코어 네트워크 제어 평면의 NF(Network Function) 중 하나로 네트워크에서 발생하는 다양한 데이터를 수집하고 이를 통해 분석된 결과를 NF에게 제공함으로써 네트워크 자원을 최적화할 수 있는 기능을 제공한다. 최초 정의된 3GPP Release-15에서는 네트워크 슬라이스 선택을 위한 분석 정보를 제공해주는 기능으로만 정의되었으나, 현재 Release-18에서는 서비스 품질 조절, 이동성 관리, NF 부하 조정 등 다양한 기능과 구체적인 사용 사례가 추가되었다[1]. 이러한 표준화의 발전과 함께 AI/ML(Artificial Intelligence /Machine Learning)에 기반하여 데이터 분석 기능을 포함한 NWDAF와 관련된 연구가 진행되고 있다[2-9].

특히 NWDAF 분석 정보를 이용한 이상 탐지에 대한 다양한 연구가 수행되어왔다. 로지스틱 회귀와 XGBoost 같은 ML 알고리즘을 사용하여 네트워크 부하를 예측하고 이상을 탐지하는 방법[3]과 LSTM(Long Short-Term Memory) 오토인코더를 활용하여 비정상 트래픽을 탐지하는 방법[4]이 제안되었으며, NWDAF를 활용한 이상 탐지 시 종합적으로 고려해야 할 사항들을 다룬 연구[5]가 진행됐다. 이 외에도 NWDAF를 활용하여 사용자 이동성을 예측하는 방법[6], 코어 네트워크 시그널링 트래픽 분석을 통해 지능형 MANO (Management and Orchestration) 결정을 수행하는 방법[7], NWDAF를 이용한 네트워크 슬라이싱 자동 관리 방법에 대한 연구 등이 수행됐다[8,9].

NWDAF 테스트베드를 구현하기 위해 Open5GS, Free5GC, OpenAirInterface와 같은 5G 코어 오픈소스 프로젝트에 NWDAF를 통합하는 연구가 진행되었다. 일부 연구에서는 오픈소스 프로젝트를 기반으로 NWDAF를 구현했으나, 3GPP 표준 API를 완전히 지원하지 않거나[6,10] ML 모델 프로비저닝 서비스를 포함하지 않은 경우가 있었다[3]. 이에 따라 3GPP에서 표준화된 분석 서비스와 모델 프로비저닝 서비스를 모두 지원하는 NWDAF 구현에 대한 연구도 이루어졌다[11]. 또한, 오픈소스 기반 환경에서 구현된 단일 NWDAF를 활용하여 네트워크 신호 트래픽을 분석[7]하거나 이상 트래픽을 탐지[4]함으로써, 실제 환경에서의 적용 가능성을 입증하였다. 기존의 NWDAF 연구들은 주로 중앙집중형 구조를 기반으로 여러 NF로부터 데이터를 수집·분석하여 네트워크 성능 향상을 지원하는 기능을 제공해왔다. 그러나 이러한 중앙집중형 구조에서는 대량의 데이터 전송으로 인해 네트워크 부하가 증가하고, NWDAF의 처리 능력이 한계에 도달할 위험이 있었다. 이를 해결하기 위한 다양한 프레임워크가 제안되었다.

연합 학습(Federated Learning)을 적용한 분산형 NWDAF 구조에서는 네트워크 자원 사용량을 줄이기 위해 개별 NF에 Leaf NWDAF를 위치시켜 로컬 모델을 생성하고, Root NWDAF에서 로컬 모델을 집계하여 글로벌 모델을 구축했다[2]. 계층형 네트워크 데이터 분석 프레임워크(Hierarchical Network Data Analytics Framework, H-NDAF)는 Root NWDAF에서 모델의 학습을 수행하고, 여러 Leaf NWDAF에서 추론 작업을 분산 처리하여 효율성을 높이기 위한 방안으로 제안되었다[12]. 또한, 분산 네트워크의 보안성과 강인성 문제를 개선하기 위해 로컬 차별적 프라이버시(Local Differential Privacy, LDP)와 서버 NWDAF의 피드백

메커니즘이 적용된 프레임워크도 제안되었다[13]. 더 나아가 대규모 이종 환경에서의 데이터 수집 및 ML 모델 훈련을 위해 생성적 적대 신경망(Generative Adversarial Networks, GAN), 전이 학습, 메타 학습 모듈을 포함한 NWDAF 기반의 고수준 아키텍처도 제시되었다[14]. 하지만 제안된 프레임워크 대부분 시뮬레이션을 통한 실질적인 검증 없이 개념적인 수준에서 논의되고 있다[2,14]. H-NDAF 프레임워크에서는 단일 Leaf 노드와 단일 Root 노드를 대상으로 한 제한적인 검증을 수행하였으며[12], LDP와 피드백 메커니즘이 적용된 프레임워크에서는 4개의 가상 머신을 사용하여 시뮬레이션을 수행했다[13].

실제 네트워크 환경에서는 네트워크 규모에 따라 다수의 NWDAF가 존재하며 이들 간의 연합 학습 구조, 분산 시스템에서의 배치 구조 및 협력 방안 등을 검증하기 위해서는 유연한 형태의 시뮬레이터가 필요하다. 오픈 소스 테스트 베드를 활용하면 실제 환경에서의 동작을 확인할 수 있다는 장점이 있지만, 이미 정의된 아키텍처를 따르는 오픈 소스 내에서 다수의 NWDAF가 상호작용하는 네트워크 환경을 실험하는 데 확장성과 유연성에 제약이 존재하며 시스템 구축에 상당한 시간이 요구된다.

본 논문에서는 이와 같은 한계를 극복하기 위해 액터 모델 기반의 NWDAF 시뮬레이터를 제안한다. 액터 모델 기반 시뮬레이터는 각 NWDAF와 NF를 독립적인 액터로 모델링함으로써 복잡한 분산 환경에서의 다양한 NWDAF 구조와 협력 시나리오를 신속하게 구현하고 검증할 수 있다. 액터로 정의된 NF는 이벤트 기반의 메시징을 통해 상호작용하는 구조를 갖는데, 이는 5G 코어 NF들의 서비스 기반 아키텍처(Service Based Architecture, SBA)를 모델링하는 데 적합한 방법이다. 독립적인 NF들은 메시지를 통해서만 상호작용하기 때문에 한 NF에서 에러가 발생하더라도 시스템 전체에 영향을 미치지 않는다. 또한 새로운 액터를 쉽게 추가할 수 있어 시스템 확장이 용이하며, 5G 네트워크의 다양한 서비스 환경을 시뮬레이션하는 데 적합하다는 장점이 있다. 오픈소스 프로젝트와 달리 물리적 시스템 구축 없이도 다양한 시나리오를 신속하게 시뮬레이션할 수 있어 시간과 비용을 절감할 수 있다. 또한 파이썬 기반의 AI/ML 모델도 손쉽게 통합하고 구현할 수 있어 NWDAF의 다양한 학습 및 추론 방안에 대한 검증이 더욱 용이하다는 장점이 있다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 NWDAF의 역할 및 기능 그리고 NF와의 상호작용 메커니즘에 대해서 기술하고, 3장에서는 액터 모델 기반의 NF 시뮬레이터 설계에 대해 설명한다. 4장에서는 DDoS 공격 시나리오에서 NWDAF가 수집된 네트워크 데이터를 분석하여 그 결과를 구독하는 NF들에게 전송하는 시나리오를 통해 제안한 시뮬레이터를 검증하고 5장에서 결론으로 마무리한다.

Ⅱ. NWDAF 및 NF

2.1 NWDAF

NWDAF는 3GPP Release-15에서 도입된 개념으로 네트워크 자동화를 위해 기본적인 분석 정보를 PCF(Policy Control Function)에 제공하는 기능으로 최초 정의되었다[15]. 이어서, Release-16에서는 고도화된 데이터 분석 및 예측기능이 추가되었으며, 다른 NF와의 상호작용을 위한 인터페이스가 추가로 정의되었다[16]. Release-17에서는 네트워크 상태 예측 및 이상 탐지 기능이 강화되었고, 분산처리 및 데이터 수집 구조가 개선됐다[17]. 현재 Release-18에서는 AI/ML 모델 성능 모니터링 기능이 강화되었으며, 연합 학습 지원 등이 포함된 구조 개선이 이루어졌다. 또한, DCCF(Data Collection Coordination Function)를 통한 데이터 수집 효율을 향상시켰다[18]. NWDAF는 각 릴리즈를 통해 지속적으로 발전하며 5G 네트워크의 자율적인 운영 및 관리 기능을 위한 자동화 기술을 위해 다양한 기능을 강화하고 있다.

그림 1은 5G 네트워크 아키텍처로, UE가 gNB (Next generation Node B) 기지국을 통해 코어 네트워크의 다양한 NF와 연결되는 구조를 보여준다. 코어 네트워크에 위치한 NWDAF는 분석 기능을 제공하기 위해 AMF(Access and Mobility Management Function), SMF(Session Management Function), AF(Application

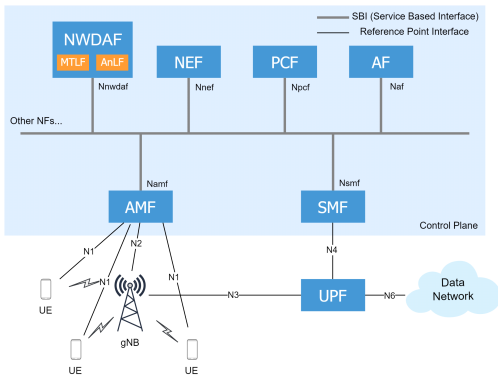

그림 1. 5G 네트워크 아키텍처 및 인터페이스
Fig. 1. 5G network architecture and interfaces

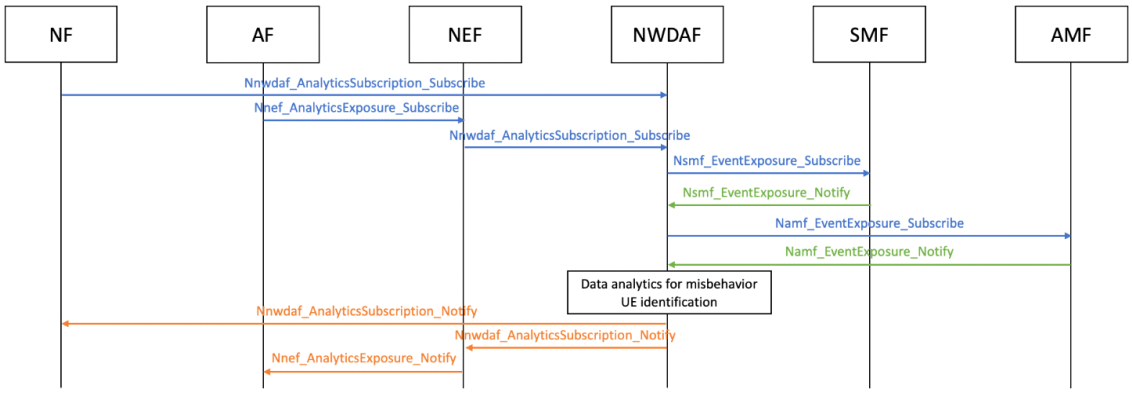

그림 2. NWDAF가 지원하는 오용되거나 하이재킹된 UE의 식별 절차
Fig. 2. Procedure for NWDAF assisted misused or hijacked UEs identification

Function), UPF(User Plane Function), PCF 등 다양한 NF들로부터 데이터를 수집한다. AI/ML 관점에서 NWDAF는 추론을 담당하는 AnLF(Analytics Logical Function)와 학습을 담당하는 MTLF(Model Training Logical Function) 두 가지 기능을 갖는다. 경우에 따라 여러 NWDAF가 존재할 수 있으며, 이들 중 일부는 AnLF나 MTLF 중 하나의 기능만을 보유할 수도 있다. 또한, NWDAF는 계층적인 구조를 통해 다른 NWDAF로부터 수집된 데이터를 분석에 활용할 수 있다. 특히 다수의 NWDAF가 연합 학습을 수행하는 경우, MTLF를 포함하는 NWDAF는 연합 학습에서 서버와 클라이언트 역할을 모두 수행한다. 클라이언트 NWDAF는 로컬 환경에 맞춘 모델을 훈련한 후, 모델의 파라미터만 서버로 전송한다. 서버 역할을 하는 NWDAF는 여러 클라이언트의 업데이트를 통합하여 전역 모델을 업데이트한다. 이러한 방식으로 5G 코어 네트워크는 NWDAF의 분석 기능을 중심으로 다양한 NF들과 상호작용하는 유연한 구성을 제공하며, 이를 통해 네트워크의 효율성을 향상시킬 수 있다.

그림 1에서와 같이 제어 평면 내에서 NWDAF와 다른 NF 간의 상호작용은 SBI(Service Based Interface)를 통해 이루어진다. SBI는 표준화된 통신 방법을 제공하며, HTTP/2와 JSON을 사용하여 RESTful API를 통해 각 NF의 서비스에 접근하고 데이터를 노출할 수 있도록 한다. 그 외의 AMF와 UE/gNB 간의 연결, SMF와 UPF 간의 연결은 레퍼런스 포인트 인터페이스를 통해 이루어진다.

2.2 NF 간 상호작용 메커니즘

NWDAF와 NF 간의 상호작용은 SBI를 활용하여 이벤트 기반 서비스 구독 메커니즘을 활용하여 수행된다. NWDAF는 구독을 통해 다른 NF들로부터 네트워크 데이터를 수집하고, 분석 결과를 구독 중인 NF들에게 통보하는 방식으로 작동한다. NWDAF는 슬라이스 부하 레벨, NF 부하, 네트워크 성능, UE 관련, QoS(Quality of Service) 지속 가능성, 세션 관리, 위치정보 등 다양한 분석 결과를 제공한다[18]. 특히, UE와 관련된 분석에는 이동성 분석, 통신 분석, 네트워크 분석과 관련된 UE 행동 예측 파라미터 분석, 네트워크 관련 이상 행동 감지 분석이 포함된다. 그림 2[16]의 UE 이상 행동 탐지를 위한 데이터 분석 절차를 통해 NWDAF와 NF의 상호작용 메커니즘을 설명한다.

NWDAF의 분석 결과가 필요한 NF는 Nnwdaf_AnalyticsSubscription_Subscribe를 통해 구독 요청을 한다. 이를 위해 NF는 다음과 같은 정보를 포함하여 구독 요청을 전송한다: Analytics ID(예: Analytics ID=“Abnormal behaviour”), 분석 대상(예: Target of Analytics Reporting=SUPI), 그리고 구독에 필요한 상세정보가 포함된 분석 필터 정보(예: Analytics Filter Information = [Exception ID=Suspicion of DDoS attack, Expected UE parameters =(Periodic Time, Communication Duration Time, Scheduled Communication Time, Traffic Profile 등)]). 신뢰할 수 없는 AF의 경우 그림 1과 같이 NEF(Network Exposure Function)를 통해 NWDAF에 구독 요청을 수행한다. 구독을 요청받은 NWDAF는 해당 요청과 관련된 데이터가 없을 경우 해당 데이터를 보유한 NF에게 구독 요청을 통해 데이터를 수집하며, 이와 같은 과정은 SBI를 통해서 이루어진다.

NWDAF는 SMF, AMF와 관련하여 각각 Nsmf_EventExposure_Subscribe, Namf_Eve ntExposure_Subscribe를 통해 UE의 이상 행동에 대한 정보를

구독하고, 각 NF는 Nsmf_EventExposure_Notify, Namf_Even tExposure_Notify를 통해 해당 정보를 제공한다. NWDAF는 수신한 데이터를 이용하여 오용되거나 하이재킹된 UE 식별을 위한 데이터 분석을 수행하고, 분석 결과를 소비자 NF에게 Nnwdaf_Analytics Subscription_Notify를 통해 제공한다. 각 NF가 제공하는 서비스의 상세 내용은 TS23.502[19]에서 확인할 수 있다.

Ⅲ. 액터 모델 기반 NF 시뮬레이터 설계

NWDAF와 NF 간의 상호작용을 모델링할 수 있는 액터 모델 기반의 시뮬레이터를 설계하였다. 액터 모델은 독립적인 액터들로 구성되며, 각 액터들은 서로의 상태에 직접적으로 접근할 수 없고, 비동기 메시지 전달을 통해 통신한다. 이러한 액터 모델의 특징은 5G 코어의 서비스 기반 아키텍처를 모델링하는 데 적합하다.

서비스 기반 아키텍처는 독립적인 NF들로 구분되며, 이들 NF는 SBI를 통해 필요한 정보를 전달받을 수 있다. NF는 특정 서비스를 제공하는 모듈화된 소프트웨어 애플리케이션으로 필요에 따라 독립적으로 배포되고 업그레이드될 수 있어 액터의 특성을 잘 반영한다. 따라서 NWDAF를 포함한 NF들은 독립적인 액터로 구현할 수 있으며, 액터 간의 이벤트 기반 통신 메시지를 SBI로 사용할 수 있다. 이로 인해 독립적으로 구현된 NF는 병렬로 실행될 수 있으며 이는 모델의 확장이 용이하다는 특징을 갖는다. 따라서 액터 모델을 활용하면 NF 간의 복잡한 상호작용을 모델링하는 데 적합하다. 액터 모델을 구현하기 위해 파이썬 언어와 Pykka 라이브러리[20]를 사용했다.

3.1 NF 액터 정의

NF는 이벤트 메시지를 활용하여 구독 기반으로 동작하기 때문에 공통적인 구독 기능과 구독 취소 기능을 포함하는 일반적인 액터로 정의할 수 있다. 그림 3은 일반적인 NF 액터의 정의를 보여준다. 각 NF를 모델링하는 액터는 Threading Actor를 사용하여 독립적인 스레드로 실행된다. NF는 이벤트 구독 기반으로 이루어지기 때문에 구독 요청(subscribe) 및 취소(unsubscribe) 메서드를 기본적으로 포함하고, 자신을 구독하고 있는 다른 소비자 NF에게 notify를 통해 데이터를 전송한다. NF에 따라 각자가 수행하는 기능과 관련한 함수를 추가적으로 포함할 수 있다.

NetworkFunction으로 정의된 일반적인 액터는 사용되는 변수들을 초기화하는 __init__(self) 메서드와 액터가 메시지를 수신할 때 실행되는 on_receive(self, message) 메서드, 그리고 수신된 메시지에 따라 특정한 기능을 수행하는 기타 메서드(subscribe, unsubscribe, response, notify 등)로 구성된다.

```python
class NetworkFunction(pykka.ThreadingActor):
    def __init__(self, actor_id):
        super().__init__()
        self.actor_id = actor_id
        self.subscribers = set()
        self.data_storage = []

    def on_receive(self, message):
        msg_id = message['id']
        msg_type = message['msg']
        data = message.get('data', None)

        if msg_type == 'subscribe':
            self.subscribe(msg_id)
        elif msg_type == 'unsubscribe':
            self.unsubscribe(msg_id)
        elif msg_type == 'response':
            self.response(msg_id, data)
        elif msg_type == 'notify':
            self.receive_data(data)
        else:
            logger.warning(f"Unknown message type
    received: {msg_type}")

    def subscribe(self, msg_id):
        #Add a subscriber and respond to a request

    def unsubscribe(self, msg_id):
        #Delete a subscriber and respond to a
    request

    def response(self, msg_id, data):
        #Print a response to a subscription request

    def notify(self, msg_id, message, data):
        #Send notifications to subscribers

    def receive_data(self, data):
        #Store data and process data according to
    functions
```

그림 3. 일반적인 NF 액터 정의
Fig. 3. Definition of general NF actor

__init__(self) 메서드에서는 이 NF를 구독하는 구독자들을 저장할 subscribers 집합을 초기화하며, 이와 함께 기타 필요한 변수 및 상수들도 초기화할 수 있다.

on_receive(self, message) 메서드는 수신한 메시지에 따라 필요한 메서드를 호출하여 기능을 수행한다. NF는 전송하는 메세지에 각 액터의 고유한 값을 나타내는 id, 요청하는 서비스의 기능을 표현하는 msg, 그리고 전송하는 data를 기본적인 키(key)값으로 포함한다. 메시지의 msg 값에 따라 수행하는 서비스 기능이 달라진다.

- 'subscribe': subscribe(msg_id) 메서드를 호출하여 subscribers에 구독자를 추가한다. 이미 구독 중인 경우에는 추가하지 않는다.
- 'unsubscribe': unsubscribe(msg_id) 메서드를 통해 구독자를 제거하고, 구독자가 존재하지 않으면 무시한다.
- 'response': response(msg_id, data) 메서드를 이용해 구독 요청에 대한 응답을 출력한다.

- ‘notify’: 다른 NF에게 데이터나 결과 분석값을 수신한 경우로 received_data(data) 메서드 호출을 통해 NF 기능을 수행한다. 예를 들어 received_data(data) 메서드에서는 수신된 데이터를 저장하거나, 이상 탐지, 네트워크 성능 평가, 세션 관리 분석 등의 업무를 각 기능별로 정의된 함수를 이용하여 수행하고 결과를 구독자들에게 전달한다.

이 기능들은 모든 NF가 가진 공통적인 메서드들로 필요에 따라 액터를 추가하고 데이터 처리 로직을 구현하여 기능을 확장할 수 있다.

3.2 NF 간의 상호작용

액터로 모델링된 NF는 메시지를 통해 상호작용한다. 각 NF가 액터로 생성될 때 고유 식별자인 URN (Universal Resource Name)이 할당되며, 다른 NF에게 URN을 사용하여 메시지를 보낼 수 있다. 액터는 앞 절에서 정의된 NetworkFunction 액터 클래스를 사용하여 start() 메서드를 호출함으로써 생성하고 실행될 수 있다.

그림 4는 NetworkFunction 액터 클래스를 이용해 actor_id가 NWDAF인 액터와 PCF, SMF인 액터를 생성 및 실행하는 예시를 보여준다. 이 메서드를 통해 각 액터의 ActorRef를 반환 받아 NWDAF, PCF, SMF에 할당하고 ActorRef를 통해 NF에 접근할 수 있다. 이후 각 액터들은 ActorRegistry에 등록되어 관리된다. 정의된 액터들은 ask() 메서드를 통해서 비동기적으로 메시지를 전달한다. 이 방법은 메시지를 보내고 메시지를 수신한 액터의 응답을 기다리는 동안 다른 작업을 수행한다. 메시지가 목적지 액터에 도착하면 해당 액터는 on_receive(message) 메서드를 호출하여 메시지를 처리한다. 그림 5는 PCF 액터와 SMF 액터가 NWDAF 액터에게 구독 요청을 하고 SMF가 구독 요청을 전송하는 메시지를 보여준다. 메시지는 딕셔너리 형태로 id, msg, data 세 가지 키로 구성된다. id는 메시지를 보내는 액터의 ActorRef의 URN 값에 해당하며 구독 요청하는 액터의 고유 식별자를 의미한다. msg는 메시지의 동작 기능으로 구독, 구독 취소, 알림, 반응, 수신 등의 기능을 의미하며 data는 전송하고자 하는 데이터를 의미한다.

```
NWDAF = NetworkFunction.start(actor_id="NWDAF")
PCF = NetworkFunction.start(actor_id="PCF")
SMF = NetworkFunction.start(actor_id="SMF")
```

그림 4. 액터의 생성
Fig. 4. Creation of NF actors

```
# Subscribe NWDAF
NWDAF.ask({'id': PCF.actor_urn, 'msg': 'subscribe',
    'data': None}, block=False)
NWDAF.ask({'id': SMF.actor_urn, 'msg': 'subscribe',
    'data': None}, block=False)
# Transmit data to NWDAF
NWDAF.ask({'id': SMF.actor_urn, 'msg': 'notify', '
    data': data}, block=False)
```

그림 5. 액터 간 메시지 전송 방법
Fig. 5. Message transmission between actors

ask() 메서드를 통해 메시지를 수신한 액터는 msg의 값에 따라 on_receive() 메서드에서 처리된다. NF 내에서 처리된 분석 결과나 고유하게 보유한 데이터는 notify(msg_id, message, data) 메서드를 통해 전송되며 그림 6을 통해 이 동작을 확인할 수 있다. NF는 구독자 목록에 존재하는 다른 NF에게 데이터를 전송할 수 있다. 따라서 발신 요청 시 msg_id를 통해 발신 대상의 URN이 레지스트리에 등록되어 있는지 먼저 확인한다. 이 과정을 통해 레지스트리에 등록된 액터의 URN을 반환받고, 이 값을 사용하여 데이터 전송이 가능하다. 액터는 자신의 id, 메시지 기능인 msg, 그리고 분석 결과나 전송할 데이터를 data에 할당하여 다른 액터에게 메시지를 전송한다. 이와 같이 액터들은 ask() 메서드를 활용하여 이벤트 기반의 메시지 전송을 통해 데이터를 교환한다. 만약 액터가 보유하지 않은 URN을 가진 액터의 요청이라면, 해당 액터에게 메시지를 보낼 수 없다는 에러를 출력한다. 또한, 액터 간 비동기적 메시지 전송이 실패한 경우 사전에 정의된 재전송 횟수만큼

```
def notify(self, msg_id, message, data):
    actor = pykka.ActorRegistry.get_by_urn(msg_id)
    if actor is not None:
        for attempt in range(self.retry_limit):
            try:
                # Transmit message
                response = actor.ask({
                    'id': self.actor_id,
                    'msg': message,
                    'data': data}, block=False)
                # Process response from other actor
                if response:
                    logger.info(f"Response received
     from {msg_id}: {response}")
                    return
                else:
                    logger.warning(f"No response from
      {msg_id}. Retrying...")
            # Exception handling
            except pykka.ActorDeadError:
                logger.error(f"Actor {msg_id} is dead
     . Cannot notify.")
                break

            except Exception as error_msg:
                logger.error(f"Failed to notify {
     msg_id} on attempt {attempt + 1}. Error: {str(
     error_msg)}")
            # Wait for retransmission
            time.sleep(self.retry_delay)
        logger.error(f"All retries failed for {
     msg_id}.")
    else:
        logger.error(f"Actor with URN '{msg_id}' not
     found. Failed to send notification.")
```

그림 6. Notify 메서드
Fig. 6. Notify method

메시지 재전송을 수행하며, 해당되는 에러 로그 메시지를 출력한다. 본 논문에서는 재전송 횟수를 3회로 지정했다.

Ⅳ. 시뮬레이션 시나리오 및 결과

4.1 시나리오

PCF가 NWDAF를 통해 UE 이상 행동 탐지(Exception ID=Suspicion of DDoS Attack)에 대한 구독을 수행하고, 수신한 NWDAF의 분석 결과에 따라 의사 결정을 내리는 시나리오를 통해 액터로 모델링된 NF 시뮬레이터에 대한 검증을 수행했다.

PCF는 네트워크 자원과 QoS를 제어하고 관리하며 네트워크의 정책 결정 담당을 통해 사용자의 서비스 경험을 최적화하는 기능을 수행한다. 본 시나리오에서는 PCF는 NWDAF로부터 DDoS 공격에 대한 분석을 구독하고, NWDAF는 구독을 통해 AF, SMF, AMF, UPF로부터 필요한 정보를 수집했다고 가정한다. 그림 7과 같이 NWDAF는 수집된 정보를 바탕으로 오토인코더 기반의 이상 탐지 모델을 활용하여 데이터 분석을 수행하고 데이터 분석 결과를 Nnwdaf_AnalyticsSubscription_Notify를 이용해 통보한다. 이 시나리오에서는 분석을 수행한 트래픽의 세션 정보와 이상 점수를 PCF에 전송한다고 가정한다. PCF는 NWDAF에게 수신한 이상 점수를 자신이 보유한 임계값과 비교하여 이상 탐지를 수행한다. 만약 이상 점수가 PCF가 보유한 임계값보다 작아서 이상이 발견되지 않는다면 PCF는 별다른 조치를 취하지 않는다. 그러나 임계값 보다 높은 이상 점수가 전달되면 PCF는 SMF에 해당 세션의 연결 해제를 요청하는 Nsmf_PDUSession_Release를 전송한다. SMF는 수신된 정보를 바탕으로 해당 세션에 대한 연결을 해제하고 백-오프 타이머(back-off timer)를 구동하여 DDoS 공격에 대응한다. 따라서 본 시나리오를 통해 액터 모델로 구성한 네트워크 코어 간의 정보교환과 수신된 정보를 바탕으로 의사 결정을 수행하는 NF 시뮬레이터를 검증할 수 있다.

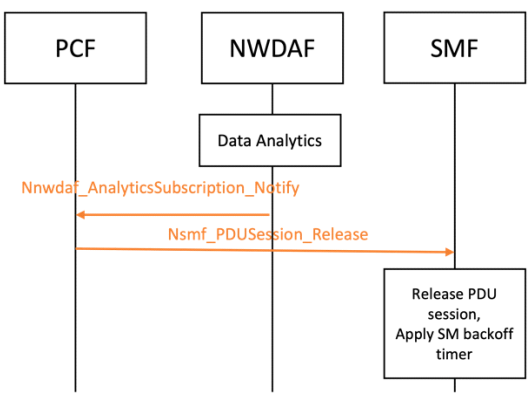

그림 7. DDoS 이상 탐지 시나리오
Fig. 7. Scenario for DDoS attack

4.2 CSE-CIC-IDS 2018 데이터셋

CSE-CIC-IDS 2018 데이터셋[21]은 네트워크 보안 연구를 위해 설계된 데이터셋으로, 다양한 유형의 DDoS 공격을 포함하고 있다. 이 데이터셋은 네트워크 트래픽의 84가지 특징(features)을 포함하고 있으며, 침입 예측을 위한 다양하고 포괄적인 벤치마크 데이터 세트를 제공한다. 본 논문에서는 NWDAF가 UE의 DDoS 공격을 탐지하는 시나리오를 구현하기 위해 CSE-CIC-IDS 2018 데이터셋의 트래픽 플로우의 메타데이터 및 특징 데이터를 코어 네트워크 내의 각 NF들에서 수집된 데이터로 매핑하여 사용했다. CSE-CIC-IDS 2018 데이터셋의 전체 특징은 표 1과 같다.

CSE-CIC-IDS 2018 데이터셋은 공격 인프라에는 50개의 단말이 포함되어 있고 피해 인프라에는 420개의 단말과 30개의 서버가 포함된 환경에서 수집된 데이터로, 이동성 정보는 포함하지 않는다. 따라서 CSE-CIC-IDS 2018 데이터는 세션 관리를 담당하고 세션 관련 트래픽 데이터와 상태정보를 수집하는 SMF 데이터, 사용자 평면 트래픽을 처리하며 패킷의 세부 정보를 수집하는 UPF 데이터[1], 애플리케이션 레벨에서 트래픽과 이벤트 로그를 수집하는 AF 데이터 등으로 매핑할 수 있다.

세부적으로 보면, Source IP와 Destination IP는 UE와 서비스 서버 간의 IP 주소를 나타내며, Source Port와 Destination Port는 UE와 서버 간의 포트 번호를 나타낸다. 이는 SMF, UPF, AF 등에서 수집된 데이터로 볼 수 있다. Flow Duration은 트래픽 플로우의 지속시간으로 세션 지속시간으로 간주할 수 있다. 또한 Flow IAT(Inter Arrival Time) Mean/Std(Standard deviation)/Max/Min /Tot(Total) 같은 특징들은 두 플로우 간의 평균, 표준편차, 최대, 최소, 전체 시간을 의미하며, 이는 SMF, AF 등에서 수집된 세션 데이터로 볼 수 있다. 이때, Fwd(Forward)는 두 플로우 간 순방향을 의미하고, Bwd(Backward)는 역방향을 의미한다. 그 외에도 전송된 총 패킷 수와 수신된 총 패킷 수를 의미하는

1) 현 TS23.288 규격에서는 UPF 관련 데이터들은 직접 수집에 제한적이며, SMF를 통해 간접적으로 수집 가능하다. 추가적으로 UPF 관련 패킷들의 세부 데이터들은 AF 혹은 OAM 등을 통해 수집 가능할 수 있다는 것을 전제로 했다.

표 1. CSE-CIC-IDS 2018 데이터셋의 전체 특징
Table 1. Features of CSE-CIC-IDS 2018 dataset

No.	Features	No.	Features	No.	Features	No.	Features	No.	Features
1	Flow ID	18	Bwd Pkt Len Min	35	Bwd IAT Max	52	RST Flag Cnt	69	Subflow Fwd Byts
2	Src IP	19	Bwd Pkt Len Mean	36	Bwd IAT Min	53	PSH Flag Cnt	70	Subflow Bwd Pkts
3	Src Port	20	Bwd Pkt Len Std	37	Fwd PSH Flags	54	ACK Flag Cnt	71	Subflow Bwd Byts
4	Dst IP	21	Flow Byts/s	38	Bwd PSH Flags	55	URG Flag Cnt	72	Init Fwd Win Byts
5	Dst Port	22	Flow Pkts/s	39	Fwd URG Flags	56	CWE Flag Count	73	Init Bwd Win Byts
6	Protocol	23	Flow IAT Mean	40	Bwd URG Flags	57	ECE Flag Cnt	74	Fwd Act Data Pkts
7	Timestamp	24	Flow IAT Std	41	Fwd Header Len	58	Down/Up Ratio	75	Fwd Seg Size Min
8	Flow Duration	25	Flow IAT Max	42	Bwd Header Len	59	Pkt Size Avg	76	Active Mean
9	Tot Fwd Pkts	26	Flow IAT Min	43	Fwd Pkts/s	60	Fwd Seg Size Avg	77	Active Std
10	Tot Bwd Pkts	27	Fwd IAT Tot	44	Bwd Pkts/s	61	Bwd Seg Size Avg	78	Active Max
11	Tot Len Fwd Pkts	28	Fwd IAT Mean	45	Pkt Len Min	62	Fwd Byts/b Avg	79	Active Min
12	Tot Len Bwd Pkts	29	Fwd IAT Std	46	Pkt Len Max	63	Fwd Pkts/b Avg	80	Idle Mean
13	Fwd Pkt Len Max	30	Fwd IAT Max	47	Pkt Len Mean	64	Fwd Blk Rate Avg	81	Idle Std
14	Fwd Pkt Len Min	31	Fwd IAT Min	48	Pkt Len Std	65	Bwd Byts/b Avg	82	Idle Max
15	Fwd Pkt Len Mean	32	Bwd IAT Tot	49	Pkt Len Var	66	Bwd Pkts/b Avg	83	Idle Min
16	Fwd Pkt Len Std	33	Bwd IAT Mean	50	FIN Flag Cnt	67	Bwd Blk Rate Avg	84	Label
17	Bwd Pkt Len Max	34	Bwd IAT Std	51	SYN Flag Cnt	68	Subflow Fwd Pkts		

Total Fwd Pkts/Total Bwd Pkts, 전송된 패킷과 수신된 패킷의 길이와 관련된 통계 특성(평균, 표준편차, 최대값, 최소값), 패킷의 전송 속도, 패킷의 설정 플래그를 의미하는 특징들은 패킷 세부 정보를 수집하는 UPF, AF 등의 데이터로 매핑할 수 있다. Active Mean/Std/Max/Min 및 Idle Mean/Std/Max/Min과 같은 특징들은 주로 네트워크 트래픽의 활동(active) 및 유휴(idle) 상태와 관련된 통계 데이터를 나타내며, 애플리케이션 트래픽을 모니터링하고, 특정 애플리케이션이나 서비스가 얼마나 자주, 얼마나 오랜 기간 동안 활성 상태인지 기록하는 AF 등에서 수집된 데이터로 매핑할 수 있다.

4.3 NWDAF DDoS 이상 탐지 모델

NWDAF에 사용되는 이상 탐지 모델로 오토인코더를 사용했다. 오토인코더는 비지도 학습 기반의 모델로, 정상 네트워크 트래픽 데이터로 학습됐다. 학습 과정에서는 네트워크 트래픽의 통계값만을 반영하기 위해 소켓 정보(Source IP, Destination IP, Source Port, Destination Port)와 Flow ID, Timestamp와 같이 트래픽 이상 탐지와 관련이 없는 특징들과 결측 및 중복 데이터는 전처리 과정을 통해 제외하고 77개의 특징만을 활용했다. 오토인코더 모델은 5개의 레이어로 이루어져 있으며 각 레이어는 77, 57, 38, 57, 77개의 유닛을 포함하고 있다. 학습을 위해 사용된 하이퍼파라미터는 표 2와 같으며, 실험을 통해 이상 탐지 성능을 최대화할 수 있는 값으로 설정하였다.

표 2. 오토인코더의 하이퍼파라미터
Table 2. Hyperparameters for autoencoder

Batch size	256
Learning Rate	0.0001
Epoch	100
Optimizer	Adam
Loss function	L2
Anomaly Score	L1

4.4 검증 결과

독립적으로 모델링된 NWDAF, PCF, SMF 액터는 액터 실행과 동시에 고유한 URN 값을 그림 8과 같이 할당받는다. 각 액터가 발생시키는 메시지의 형태를 구분하기 위해 logger.INFO 출력에는 해당 메시지를 출력하는 액터를 [actor_name] 형태로 표기했다.

```
=============== Check the URN ===============
NWDAF:  urn:uuid:e6f4599f-98cc-400c-b2f2-1670815751dc
PCF   :  urn:uuid:31215ef0-406d-4f15-b840-1affb2510b31
SMF   :  urn:uuid:4728aeb5-c043-4eb1-8fb7-4c17912df6b3
```

그림 8. NF별 할당된 URN
Fig. 8. URN allocation

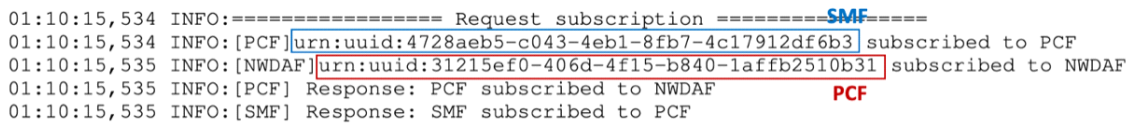

그림 9. SMF와 PCF의 구독 요청과 응답
Fig. 9. Subscription request from SMF and PCF and responses

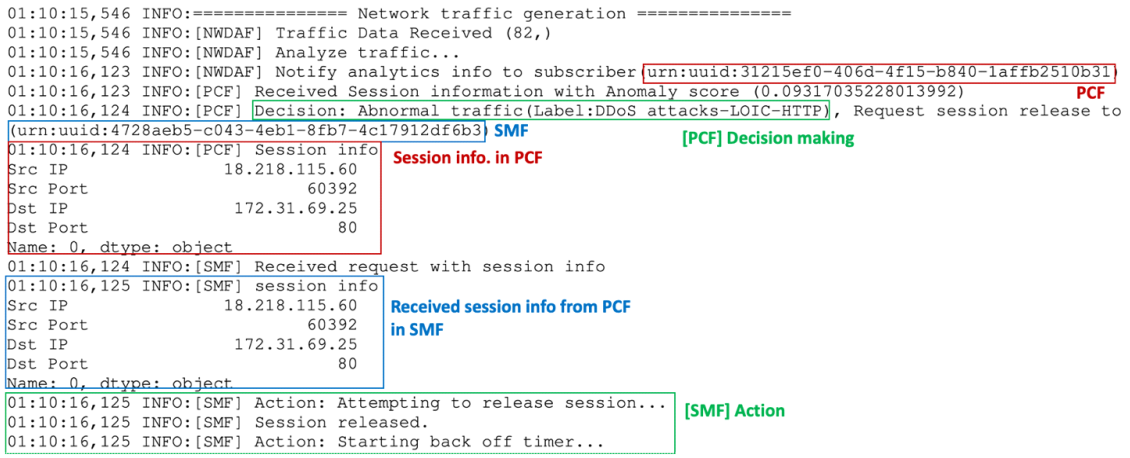

그림 10. 이상 트래픽에 따른 PCF의 의사 결정 및 SMF의 대응 행동
Fig. 10. Decision-making by PCF and response actions by SMF according to abnormal traffic

PCF는 NWDAF의 이상 트래픽 분석을 구독하며, SMF는 이상 트래픽에 대한 PCF의 판단 결과를 구독한다. 그림 9는 SMF와 PCF의 구독 요청 결과를 보여준다. SMF의 구독 요청을 받은 PCF는 정상적으로 구독 메시지를 처리한 후, 응답 메시지를 SMF에 전송한다. SMF는 PCF에 성공적으로 구독되었음을 알리는 메시지를 수신한 후 이를 출력한다. PCF도 NWDAF에 성공적으로 구독되었다는 사실을 출력 메시지를 통해 확인할 수 있다.

각 NF가 구독을 완료한 후 정상 및 비정상 형태의 네트워크 트래픽이 NWDAF로 수집되며, NWDAF는 데이터 분석 결과를 PCF에게 전달한다. 그림 10은 NWDAF가 비정상인 네트워크 플로우를 분석하고 그 결과를 구독자인 PCF에 통보할 때의 과정을 보여준다. NWDAF는 데이터를 수신하면 수신한 데이터의 정보를 콘솔창에 출력한 후 네트워크 트래픽에 대한 이상 탐지 분석을 수행한다. 이 과정에서 NWDAF는 보유한 오토인코더 모델을 사용하여 네트워크 데이터를 분석하며, 입력 데이터의 세션 정보(Source IP/Port, Destination IP/Port)와 이상 점수를 PCF에게 전송한다.

PCF는 NWDAF로부터 수신한 세션 정보와 이상 점수를 콘솔창에 출력한 후, 자신의 이상 탐지 임계값과 이상 점수를 비교하여 네트워크 데이터의 이상 유무를 판단한다. 본 연구에서는 실험을 통해 이상 탐지 정확도를 최대화하는 값인 0.06으로 임계값을 설정하였다. 첫 번째 네트워크 플로우 데이터는 이상 점수가 0.09로 임계값인 0.06보다 크기 때문에 비정상(DDoS attacks)으로 판단된다. 비정상 데이터로 판명된 경우, PCF는 구독자인 SMF에게 해당하는 세션의 릴리즈 요청을 세션 정보와 함께 전송한다. SMF는 PCF로부터 이상 트래픽 데이터로 판명된 데이터의 세션 정보를 수신하고 그 정보를 콘솔창에 출력한다. 그림 10에서 PCF가 전송한 세션 정보가 정확히 SMF로 전달되었음을 확인할 수 있다. 이후 SMF는 수신한 세션 정보에 대해 릴리즈 동작을 수행하고, 백-오프 타이머를 구동한다.

그림 11은 정상인 네트워크 플로우가 연속으로 들어온 경우의 결과를 보여준다. 두 번째로 수신된 트래픽 데이터의 분석 결과는 NWDAF에서 PCF로 정상적으로 전송된다. 그러나 전송된 이상 점수가 0.02로 PCF가 보유한 이상 탐지 임계값 보다 작기 때문에 정상 데이터로 판별되며, SMF에 별도의 메시지를 전송하지 않는다. 세 번째로 수신된 트래픽 데이터도 두 번째와 동일하게 정상 데이터로 처리됨을 확인할 수 있다.

본 시나리오를 통해 NWDAF와 PCF, SMF 간의 비동기적 데이터 전송 및 응답이 정상적이고 일관되게 이루어짐을 검증하였다.

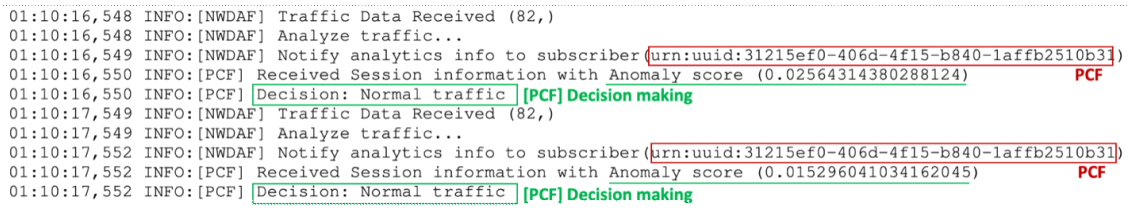

그림 11. 정상 트래픽에 따른 PCF의 의사 결정
Fig. 11. Decision-making by PCF according to normal traffic

V. 결 론

본 논문에서는 액터 모델 기반의 5G 코어의 NF 시뮬레이터를 설계하고 구현했다. 제안된 시뮬레이터는 NF를 기본적인 액터 클래스를 통해 다양한 기능을 가진 독립적인 액터로 생성할 수 있는 기능을 제공한다. 각 NF는 이벤트 기반의 메시징 시스템을 통해 다른 액터와 데이터를 교환하고 상호작용을 수행하며, 이를 통해 5G 네트워크 환경에서의 복잡한 상호작용을 효과적으로 모사할 수 있다. 본 연구에서는 CSE-CIC-IDS 2018 데이터셋을 활용하여 NWDAF가 오토인코더를 통해 네트워크 트래픽 이상 탐지 분석을 수행하고, 그 결과를 PCF에게 전송하는 시나리오를 통해 제안된 액터 기반 시뮬레이터의 동작을 검증했다. 시뮬레이션 결과, 제안된 시뮬레이터가 5G 코어 네트워크의 다양한 NF 기능을 효과적으로 구현할 수 있음을 확인하였다. 특히, 이 시뮬레이터는 NF 간의 상호작용을 분석하는 데 유용한 도구로써 가능성을 보여주었다. 액터 모델 기반의 NF들은 공통적인 구독, 구독 취소, 알림 외에 각자 기능에 따른 메서드를 보유할 수 있어 필요한 기능을 유연하게 정의할 수 있으며, 독립적인 액터를 생성하여 시스템을 확장할 수 있는 능력을 갖추고 있다.

따라서 제안된 액터 기반 NF 시뮬레이터의 유연성과 확장성, 시뮬레이션 용이성을 통해 네트워크 구성 요소 간의 상호작용을 명확히 이해하고 분석하는 데 사용될 수 있을 것으로 기대된다. 향후 연구에서는 더 많은 NF 기능을 포함한 확장된 시나리오를 통해 시뮬레이터의 범위를 넓히고, 실험 결과를 바탕으로 네트워크 설계 및 관리 전략을 개발하는 데 활용할 예정이다.

References

[1] M. K. Shin, S. H. Lee, and J. H. Yi, “Trends of 5G network automation and intelligence technologies standardization,” *Electr. and Telecommun. Trends*, vol. 34, no. 2, pp. 92-100, Apr. 2019. (https://dx.doi.org/10.22648/ETRI.2019.J.340210)

[2] Y. Jeon, H. Jeong, S. Seo, T. Kim, H. Ko, and S. Pack, “A distributed NWDAF architecture for federated Learning in 5G,” in *Proc. IEEE ICCE*, pp. 1-2, Las Vegas, NV, USA, Jan. 2022. (https://doi.org/10.1109/ICCE53296.2022.9730220)

[3] S. Sevgican, M. Turan, K. Gökarslan, H. B. Yilmaz, and T. Tugcu, “Intelligent network data analytics function in 5G cellular networks using machine learning,” *J. Commun. Netw.*, vol. 22, no. 3, pp. 269-280, 2020. (https://doi.org/10.1109/JCN.2020.000019)

[4] A. Mekrache, K. Boutiba, and A. Ksentini, “Combining network data analytics function and machine learning for abnormal traffic detection in beyond 5G,” in *Proc. IEEE GLOBECOM*, pp. 1204-1209, Kuala Lumpur, Malaysia, Dec. 2023. (https://doi.org/10.1109/GLOBECOM54140.2023.10436766)

[5] Y. Yuan, C. Gehrmann, J. Sternby, and L. Barriga, “Insight of anomaly detection with NWDAF in 5G,” in *Proc. IEEE Int Conf. CITS*, pp. 1-6, Athens, Greece, Jul. 2022. (https://doi.org/10.1109/CITS55221.2022.9832914)

[6] A. Chouman, D. M. Manias, and A. Shami, “Towards supporting intelligence in 5G/6G core networks: NWDAF implementation and initial analysis,” in *Proc. IEEE IWCMC*, pp. 324-329, Dubrovnik, Croatia, May 2022. (https://doi.org/10.1109/IWCMC55113.2022.98

24403)

[7] D. M. Manias, A. Chouman, and A. Shami, "An NWDAF approach to 5G core network signaling traffic: Analysis and characterization," in *Proc. IEEE GLOBECOM*, pp. 6001-6006, Kuala Lumpur, Malaysia, Dec. 2023. (https://doi.org/10.1109/GLOBECOM48099.2022.10000989)

[8] K. Abbas, T. A. Khan, M. Afaq, J. J. Diaz Rivera, and W.-C. Song, "Network data analytics function for IBN-based network slice lifecycle management," in *Proc. APNOMS*, pp. 148-153, Tainan, Taiwan, Sep. 2021. (https://doi.org/10.23919/APNOMS52696.2021.9562662)

[9] K. Abbas, T. A. Khan, M. Afaq, and W.-C. Song, "Ensemble learning-based network data analytics for network slice orchestration and management: An intent-based networking mechanism," in *Proc. IEEE/IFIP NOMS*, pp. 1-5, Budapest, Hungary, Apr. 2022. (https://doi.org/10.1109/NOMS54207.2022.9789 706)

[10] T. Kim, J. Kim, H. Ko, S. Seo, Y. Jeon, H. Jeong, S. Lee, and S. Pack, "An implementation study of network data analytic function in 5G," in *Proc. IEEE ICCE*, pp. 1-3, Virtual Online Conf., Jan. 2022. (https://doi.org/10.1109/ICCE53296.2022.9730290)

[11] S. Lee, J. Lee, T. Kim, D. Jung, I. Cha, H. Ko, and S. Pack, "Design and implementation of network data analytics function in 5G," in *Proc. Inter. Conf. ICTC*, pp. 757-759, Jeju, Korea, Oct. 2022. (https://doi.org/10.1109/ICTC55196.2022.9952559)

[12] Y. Jeon and S. Pack, "Hierarchical network data analytics framework for 6G network automation: Design and implementation," *IEEE Internet Comput.,* vol. 28, no. 2, pp. 38-46, Mar.-Apr. 2024. (https://doi.org/10.1109/MIC.2024.3369939)

[13] P. Rajabzadeh and A. Outtagarts, "Federated learning for distributed NWDAF architecture," in *Proc. Conf. ICIN*, pp. 24-26, Paris, France, Mar. 2023. (https://doi.org/10.1109/ICIN56760.2023.10073493.

[14] P. Gkonis, N. Nomikos, P. Trakadas, L. Sarakis, G. Xylouris, X. Masip-Bruin, and J. Martrat, "Leveraging network data analytics function and machine learning for data collection, resource optimization, security and privacy in 6G networks," *IEEE Access*, vol. 12, pp. 21320-21336, 2024. (https://doi.org/10.1109/ACCESS.2024.3359992)

[15] 3GPP TS23.501, "*System architecture for the 5G system* (Release 15)," v2.0.1, Dec. 2017.

[16] 3GPP TS23.288, "*Architecture enhancements for 5G system (5GS) to support network data analytics services* (Release 16)," v16.3.0, Mar. 2021.

[17] 3GPP TS23.288, "*Architecture enhancements for 5G system (5GS) to support network data analytics services* (Release 17)," v17.4.0, Mar. 2022.

[18] 3GPP TS23.288, "*Architecture enhancements for 5G system (5GS) to support network data analytics services* (Release 18)," v18.0.0, Mar. 2024.

[19] 3GPP TS23.502, "*Procedures for the 5G system (5GS)* (Release 18)," v18.5.0, Mar. 2024.

[20] J. Jodal, *Pykka* v4.0.2, Retrieved Jul., 2, 2024, from https://www.pykka.org

[21] C. I. for Cybersecurity, *CICIDS 2018 dataset(2018)*, Retrieved Jun. 15, 2024, from https://www.unb.ca/cic/datasets/malmem-2020.htm

최 은 혜 (Eunhye Choi)

2012년 2월 : 이화여자대학교 전자공학과 졸업
2014년 2월 : 이화여자대학교 전자공학과 석사
2014년 2월~현재 : 국방과학연구소 선임연구원
2022년 3월~현재 : 이화여자대학교 전자공학과 박사과정
<관심분야> 이상 탐지, 인공지능, 머신러닝, 멀티에이전트 시스템
[ORCID:0000-0002-6208-4030]

최 다 영 (Dayoung Choi)

2023년 2월 : 이화여자대학교 전자전기공학전공 졸업
2023년 3월~현재 : 이화여자대학교 전자전기공학과 석/박사 통합과정
<관심분야> 딥러닝, 이상탐지, 데이터 분석
[ORCID:0009-0008-9270-0528]

성 지 훈 (Jihoon Sung)

2008년 2월 : 충남대학교 전기정보통신공학부 공학사
2010년 2월 : 한국과학기술원 전기 및 전자공학과 공학석사
2016년 8월 : 한국과학기술원 전기 및 전자공학과 공학박사
2016년 9월~2020년 8월 : 삼성전자 무선사업부 책임연구원
2020년 9월~현재 : 한국전자통신연구원 표준연구본부 선임연구원
<관심분야> 5G/6G, 네트워크 지능화

신 명 기 (Myungki Shin)

2000년 8월 : 충남대학교 대학원 컴퓨터공학과 박사
1994년 2월~현재 : 한국전자통신연구원(ETRI) 표준연구본부 책임연구원/PL
2008년 3월~현재 : 과학기술연합대학원대학교(UST) 정보통신공학 교수
<관심분야> 5G/6G 네트워크, 인공지능 기반 네트워크 자동화
[ORCID:0000-0002-2575-9916]

박 형 곤 (Hyunggon Park)

2004년 2월 : 포항공과대학교 전자전기공학과 졸업
2006년 3월 : University of California, Los Angeles (UCLA) M.S.
2008년 12월 : University of California, Los Angeles (UCLA) Ph.D.
2010년~현재 : 이화여자대학교 전자전기공학과 교수
<관심분야> 멀티에이전트 시스템 최적화, 머신러닝, 인공지능, 게임이론
[ORCID:0000-0002-5079-1504]