O-RAN SMO 시스템의 실전 배포 및 트러블슈팅에 관한 연구: 사례 분석 및 시사점

신주희, 유현민, 홍인기
경희대학교
{odong3094, yhm1620, ekhong}@khu.ac.kr

Practical Survey on Deploying and Troubleshooting O-RAN SMO Systems: Case Study and Lessons Learned

Juhee Shin, Hyunmin Yoo, Eenkee Hong
Kyunghee University

요 약

본 논문은 개방형 무선접속네트워크 (Open RAN)의 개념과 아키텍처를 체계적으로 고찰하고, 그 핵심 관리 플랫폼인 O-RAN Service Management and Orchestration (SMO)의 실전 배포 과정에서 발생하는 장애 요인과 해결책을 분석한다. Docker와 Helm 환경에서 OSC SMO를 구축하며 발생한 반복적 오류를 체계적으로 식별하고, Pod 아키텍처 분석을 통해 Helm 의존성 충돌, PVC 바인딩 실패, 인증 오류 등에 대한 효과적 대응 방안을 제시했다. 이 연구는 실제 O-RAN SMO 운영을 위한 실용적 모델과 자동화된 운영체계 구축을 위한 기술적 기반을 제공한다.

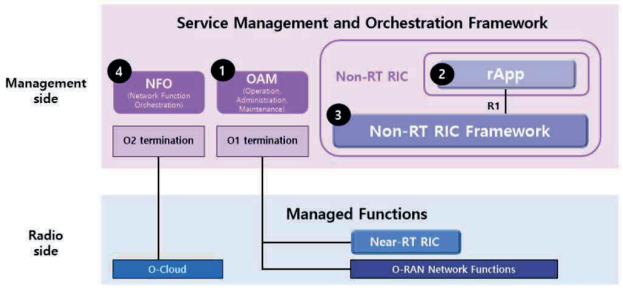
Ⅰ. 서론

Open radio access network (O-RAN)은 개방형 인터페이스와 모듈화 구조를 기반으로, 상호운용성과 유연성을 제공하는 차세대 무선접속망 아키텍처로 주목받고 있다. 이 구조에서 Service Management and Orchestration (SMO)은 O1, O2, A1 인터페이스를 통해 RAN 요소, 클라우드 인프라, 인공지능 기반 제어 계층을 통합 관리하는 핵심 플랫폼의 역할을 수행한다. 그러나 실제 클라우드 네이티브 환경에서 SMO를 설치하고 운영하는 과정에서는 Helm 차트 간의 의존성 충돌, 외부 이미지 인증 실패, PVC 바인딩 오류, 인증 설정 누락 등 다양한 구조적 장애 요소가 반복적으로 발생하고 있으며, 이는 안정적인 서비스 연속성과 자동화된 운용을 저해하는 핵심 요인으로 지적되고 있다[1][2].

본 연구는 O-RAN Software Community (OSC) 기반 SMO 시스템을 Helm과 Docker 환경에 배포하고, 여러 네임스페이스(onap, nonrtric, network, keycloak) 내 핵심 Pod들의 연계 구조를 분석했다. 설치 과정에서 발생하는 구성 오류를 식별하고, Helm 분리 배포, 보안 설정 보완 등을 통한 실용적 트러블슈팅 전략을 도출했다. 이 결과는 O-RAN SMO의 실전 배포 환경 이해와 자동화된 운영 체계 설계를 위한 기초 자료로 활용될 수 있다.

Ⅱ. SMO 구성 및 배포 실증 분석

Open RAN은 다양한 벤더의 장비 간 상호운용성을 보장하는 개방형 무선접속망 (Open Radio Access Network) 구현을 지향하는 표준 기반 아키텍처로서, 네트워크의 유연성 및 기술 혁신 촉진에 있어 주목할 만한 패러다임으로 대두되고 있다[3]. 이러한 O-RAN 아키텍처 내에서 SMO는 중앙 제어 플랫폼으로서 기능하며, RAN 요소와의 통신을 담당하는 O1 인터페이스, O-Cloud 및 Near-RT RIC과의 연계를 위한 O2 인터페이스, 그리고 Non-RT RIC 기반의 xApp 관리 및 정책 주입 기능을 수행하는 A1 인터페이스를 통합적으로 포함한다[4][5]. 그림 1은 SMO의 주요 구성요소와 인터페이스(O1, O2, R1) 간 관계를 한눈에 이해할 수 있게 한다. 상단의 Service Management and Orchestration Framework는 Non-RT RIC, Operation, Administration and Maintenance (OAM), 그리고 이들이 활용하는 O1 Termination 및 O2 Termination 계층으로 구성되며, 각 구성요소는 번호가 매겨진 인터페이스를 통해 상호 연결된다. 특히 Near-RT RIC Framework와 xApp 간의 연계 구조가 명확히 시각화되어 있으며, 하단의 Managed Functions 영역은 Near-RT RIC과 O-RAN Network Functions가 O-Cloud 인프라 위에서 운영됨을 보여준다. 이와 같은 아키텍처 구성은 SMO가 다양한 기능 계층 간의 통합 지점을 형성하고 있음을 시각적으로 보여주며, 특히 O1, O2, A1 인터페이스를 통해 관리 측과 무선 측을 유기적으로 연결한다.



[그림 1] Management side에서 바라본 O-RAN Architecture

이러한 구조적 역할에 대해 Polese 등은 SMO가 단순 관리 도구를 넘어 인공지능 기반 네트워크 지능화의 핵심 엔티티로 기능함을 분석하였다[4]. 그러나 실제 클라우드 환경에서는 이론만으로 설명되지 않는 다양한 운영 과제가 발생한다. 주요 장애로는 컨테이너 의존성 충돌로 인한 CrashLoopBackOff, PostgreSQL 연결 장애, ConfigMap 용량 초과, 이미지 인증 실패 등이 확인되었다. 본 연구는 이러한 장애를 분석하고 설정 최적화, 의존성 재설계 등의 해결책을 도출했으며, SDNC-AAF 연동 시 인증서 만료가 심각한 서비스 중단 위험을 초래함을 확인했다.

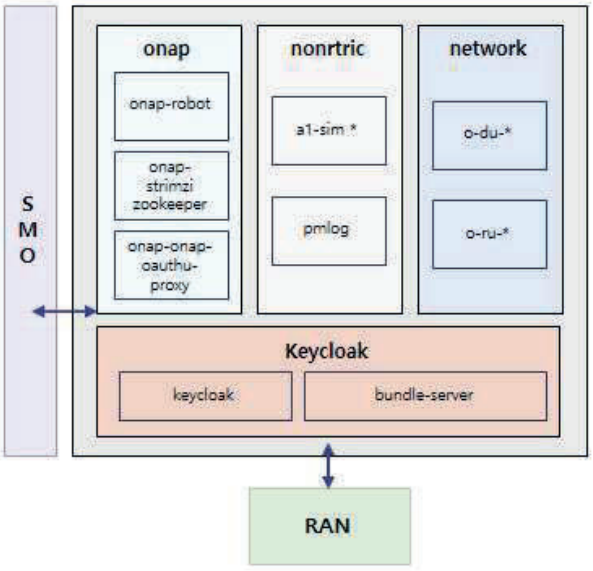
Li 등은 네트워크 슬라이싱 프레임워크를 제안하며 자원 격리와 맞춤형 정책의 중요성을 강조했다[6]. 이러한 관점에서 SMO는 Non-RT RIC과 O2 인터페이스를 통해 슬라이스 수준의 폐쇄 루프 제어(closed-loop control)를 가능하게 하며, 지능형 네트워크 운영의 중심 역할을 수행한다. 본 연구에서는 xApp 조합과 정책 구성으로 슬라이스 간 트래픽 우선순위 조정을 검증했으며, 이 기능의 안정적 운영을 위해 SMO 내부 정합성과 외부 시스템 연동이 필수임을 입증했다.

결론적으로 O-RAN SMO 배포는 단순 설치를 넘어 OAM, RIC, xApp

간 통합과 외부 시스템 연계를 포함하는 복합 프로세스다. ConfigMap 용량 한계, Init Container 오류, 인증서 만료 같은 문제는 표준 문서에서 충분히 다루어지지 않으며, 고급 기능 구현을 위해 SMO 내부 통신 경로에 대한 심층적 이해가 필요하다. 본 연구는 SMO 구축을 위한 실질적 설계 및 운영 가이드라인을 제공한다.

Ⅲ. SMO 클러스터 구성과 실전 장애 분석

그림 2는 본 연구에서 활용한 O-RAN SMO 시스템의 구성요소를 네임스페이스 단위로 시각화한 도식으로, 전체 시스템이 onap, nonrtric, network, keycloak의 네 가지 주요 서브시스템으로 구성되어 있음을 보여준다. 각 네임스페이스는 Kubernetes 기반 클라우드 환경 내에서 기능별로 역할이 분산되어 있으며, Pod 단위로 모듈화되어 상호 연계된다. 예를 들어, onap 네임스페이스에는 Robot Framework 기반 자동화 테스트를 수행하는 onap-robot과 OAuth2 기반 인증 프록시인 onap-onap-oauth2-proxy가 배포되어 있으며, keycloak 네임스페이스에는 사용자 인증 및 토큰 관리를 담당하는 keycloak Pod와 xApp 배포를 위한 bundle-server가 포함된다. nonrtric 네임스페이스에는 A1 인터페이스를 시뮬레이션하는 a1-sim-*, 정책 수립 및 배포 기능을 담당하는 policymanagementservice, 성능 모니터링을 위한 pmlog가 존재하며, network 네임스페이스의 o-du-*및 o-ru-*는 각각 DU 및 RU 기능을 가상화하여 RAN 환경을 시뮬레이션한다. 이러한 구조는 SMO의 마이크로서비스 기반 설계와 높은 확장성을 반영하며, 실제 클라우드 네이티브 환경에서의 유연한 운영을 가능하게 한다.



[그림 2] SMO 네임스페이스 배치도

이와 같은 구성 환경에서 SMO를 실질적으로 배포하는 과정에서는 다양한 장애가 반복적으로 발생하였다. 표 1은 Docker 및 Helm 기반으로 O-RAN SMO를 배포하는 실험 과정에서 식별된 주요 장애 유형과 이에 대한 트러블슈팅 전략을 요약한 것이다. 대표적인 문제로는 Helm 설치 시 ConfigMap 크기 제한 초과, Chart 의존성 경로 설정 오류, 외부 이미지 레지스트리 인증 실패, PVC 바인딩 지연, 그리고 Pod의 파일시스템 접근 권한 부족 등이 확인되었다. 이에 대한 해결 방안으로는 Helm 템플릿 출력 후 kubectl apply 방식으로 분리 적용하는 전략, 의존성 경로의 file:// 전환, imagePullSecrets 구성, claimRef 수동 패치 및 StorageClass 명시, 그리고 securityContext 및 fsGroup 설정을 적용하였다. 이러한 분석은 Helm 기반 자동화 배포의 구조적 한계와 Kubernetes 연계의 복잡성을 실증적으로 보여주며, 향후 O-RAN 기반 운영환경의 안정성과 장애 대응 체계 고도화를 위한 기초 자료로 활용될 수 있다.

[표 1] 트러블슈팅 전략

번호	문제 요약	원인	해결 방법
1	Helm 설치 용량 초과	Helm이 release 정보를 ConfigMap에 저장하면서 크기 초과	helm template → kubectl apply 방식으로 분리 배포
2	Chart 의존성 경로 오류	@local 경로 인식 불가	file:// 경로로 변경하여 의존성 명시
3	이미지 풀 인증 실패	외부 레지스트리 인증 누락	imagePullSecrets 및 .dockerconfigjson 구성
4	PVC 바인딩 실패	PV에 claimRef 누락, StorageClass 미설정	kubectl patch pv, StorageClass 점검
5	Pod 권한 오류	파일시스템 접근 권한 부족	securityContext, fsGroup 설정 추가

Ⅳ. 결론

본 연구는 O-RAN 아키텍처와 핵심 관리 플랫폼인 SMO를 중심으로 관리 계층과 무선 계층 간 상호연계 메커니즘을 실증적으로 분석하였다. OSC 제공 오픈소스 SMO 구현체를 Docker 및 Helm 기반 클라우드 네이티브 환경에 배포·운영하며 발생하는 구조적 문제점들을 식별하고, 이론적 설계와 실무 적용 간 기술적 괴리를 정량적으로 평가하였다.

배포 과정에서는 다중 네임스페이스 기반 마이크로서비스 아키텍처와 표준 인터페이스 간 연계성이 시스템 초기화 및 기능 통합의 핵심 요소임을 검증하였으며, SMO 플랫폼이 xApp 통합, 네트워크 슬라이싱, 폐쇄 루프 제어 등 고도화된 지능형 네트워크 기능 구현을 위한 중앙 제어 계층으로 기능함을 확인하였다.

결론적으로, 이러한 실증적 접근을 통해 도출된 분석 결과는 O-RAN 아키텍처의 구조적 특성과 실제 배포 환경에서 발생하는 기술적 장애 요인에 대한 체계적 분류와 해결책을 제시함으로써, 차세대 개방형 무선접속망의 자동화된 운영 환경 구축 및 최적화를 위한 이론적 프레임워크로서 중요한 의미를 갖는다.

ACKNOWLEDGMENT

"이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-대학ICT연구센터(ITRC)의 지원(IITP-2025-RS-2021-II212046, 50%)과 과학기술정보통신부 및 정보통신기획평가원의 오픈랜 인력양성 프로그램(연세대) 연구 결과로 수행되었음(IITP-2025-RS-2024-00434743, 50%)"

참 고 문 헌

[1] C. Gabriel, "The impact of 5G and next-generation networks on mobile OPEX spending," Analysis Mason report, 2018. [Online]. Available: https://tinyurl.com/yckndswz

[2] STL Partners, "Neutral host: How open RAN and neutral host paves the way for 5G," 2022. [Online]. Available: https://stlpartners.com/articles/

[3] Li, Q., Wu, G., Papathanassiou, A., & Mukherjee, U. (2016). An end-to-end network slicing framework for 5G wireless communication systems. arXiv preprint arXiv:1608.00572.

[4] Polese, M., Bonati, L., D'Oro, S., Basagni, S., & Melodia, T. (2023). Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges. IEEE Communications Surveys & Tutorials, 25(2), 1376-1411.

[5] O-RAN Working Group 1. (2021). O-RAN architecture description 5.00. O-RAN, Alfter, Germany, document O-RAN.WG1.O-RANArchitecture-Description-v05.00 Technical Specification.

[6] Wypiór, D., Klinkowski, M., & Michalski, I. (2022). Open RAN—Radio access network evolution, benefits and market trends. Applied Sciences, 12(1), 408.