個別機能制御による5G通信セキュリティ評価環境の構築

山本 優樹[1,a)]　葛野 弘樹[1]　瀧田 愼[1]　白石 善明[1]

概要：第 5 世代移動通信システム（5G）は高速通信・低遅延・多数同時接続を実現する一方で，5G 仕様に内在する設計上の脆弱性や，互換性維持のために継承された 4G 由来の脆弱性が報告されている．これらの脆弱性を実証的に評価するには，実機を用いた検証環境が実環境に即した評価を可能とするが，専用機器の導入コストや環境構築の複雑さが実用上の障壁となる．本研究では，OpenAirInterface 5G を用い，従来の単一ノード構成を Kubernetes ベースのマルチノード環境へ拡張した．各 Network Function（NF）を Pod として独立管理することで，配置制御・リソース制御・通信経路制御を含む個別機能制御を実現した．物理インタフェースを経由する構成により，L2/L3 パケットの安定取得を可能とし，さらに Multus CNI により実環境に近いネットワーク構成を再現した．本論文では，その構築手法について述べ，従来環境で困難であった物理経路上でのパケット観測と，NF の柔軟な分散配置による多様な攻撃シナリオの再現が可能となることを示している．

キーワード：5G セキュリティ，OpenAirInterface，Kubernetes，ネットワーク機能仮想化，脆弱性評価

Building a 5G Security Evaluation Platform through Independent Network Function Management

Yuki Yamamoto[1,a)]　Hiroki Kuzuno[1]　Makoto Takita[1]　Yoshiaki Shiraishi[1]

Abstract: The fifth-generation mobile communication system (5G) realizes high-speed communication, low latency, and massive device connectivity, while vulnerabilities inherent in the 5G specification design and those inherited from 4G for compatibility maintenance have been reported. To empirically evaluate these vulnerabilities, real-device verification environments enable realistic evaluation, but the introduction costs of specialized equipment and the complexity of environment construction pose practical barriers. In this study, we extended the conventional single-node configuration to a Kubernetes-based multi-node environment using OpenAirInterface 5G. By independently managing each Network Function (NF) as a Pod, we realized individual function control including placement control, resource control, and communication path control. The configuration via physical interfaces enables stable acquisition of L2/L3 packets, and Multus CNI reproduces network configurations close to real environments. This paper describes the construction methodology and demonstrates that physical path packet observation, which was difficult in conventional environments, and reproduction of diverse attack scenarios through flexible distributed placement of NFs become possible.

Keywords: 5G security, OpenAirInterface, Kubernetes, Network Function Virtualization, vulnerability assessment

1. はじめに

第 5 世代移動通信システム（5G）は，4G と比較して高速通信・低遅延・多数同時接続という特長を有し，産業分野を含む幅広い領域での活用が進められている．遠隔医療，自動運転，ならびに XR（拡張現実）を含めた応用分野では，リアルタイム性と高信頼性が強く求められており，5G は応用分野技術の実現に向けた基盤技術として期待されている．5G のネットワークアーキテクチャは，従来の専用ハードウェア中心の構成からソフトウェア中心の構成

[1] 神戸大学 大学院工学研究科 電気電子工学専攻
Dept. of Electrical & Electronic Eng., Kobe University
[a)] yamamoto.yuki@gsuite.kobe-u.ac.jp

© 2025 Information Processing Society of Japan

This paper is work in progress and not peer-reviewed.

へと大きく変化している．ソフトウェアを中心とすることにより，NFV（Network Function Virtualization）や SDN（Software Defined Networking）などの仮想化技術が導入され，ネットワーク機能の柔軟かつ効率的な展開が可能になった．こうした技術的進展を背景に，5G は社会基盤として急速に普及しつつある．

しかし，ソフトウェア化による柔軟性の向上は，新たなセキュリティ課題をもたらしている．ソフトウェア実装により攻撃対象面が拡大し，従来のハードウェアベースのセキュリティ対策では対応が困難な脅威が生じている．特に，5G 仕様に内在する設計上の脆弱性や，互換性維持のために 4G から継承された脆弱性が複数報告されている [1], [2], [3]．これらの脆弱性には，認証プロトコルの不備や暗号化の不適切な実装などが含まれ，通信の盗聴・改ざん・なりすましを可能にする深刻なセキュリティリスクとなる．したがって，5G 環境で報告された脆弱性を実証的に観測・評価するための基盤の設計と実装が急務である．

5G セキュリティの評価基盤として，実機環境とシミュレーション環境の 2 つのアプローチが存在する．実機環境は，ソフトウェア無線（SDR: Software Defined Radio）などの専用機器により実際の RF（Radio Frequency）信号を用いた現実的な評価を可能とするが，高い導入コストと構築の複雑さが障壁となる．これに対し，シミュレーション環境はソフトウェアのみで構築でき，低コストかつ再現性の高い評価基盤として注目されている．

代表的な 5G シミュレーション基盤として，オープンソースの OpenAirInterface 5G（OAI5G） [4] が広く利用されている．OAI5G は，5G コアネットワークと無線アクセスネットワークの主要機能をソフトウェアで実装したプラットフォームである．しかし，OAI5G の標準構成は Docker Compose による単一ノード構成であり，すべての Network Function（NF）が同一ホスト上で稼働する．単一ノード構成は導入が容易である反面，セキュリティ評価に必要な観測・制御において以下の制約がある．

- **通信経路の制約：**NF 間通信が Docker ブリッジ内で完結するため，物理インタフェースを経由したパケット観測や改ざんが不可能である．
- **配置の固定性：**すべての NF が単一ホスト上に配置されるため，実環境のような柔軟な構成変更や攻撃経路の挿入が困難である．

本研究では，これらの制約を解決し，5G 通信の脆弱性を効果的に評価するためのシミュレーション環境を構築する．具体的には，OAI5G をコンテナ化アプリケーションの展開・スケーリング・管理を自動化するオープンソースのコンテナオーケストレーション基盤である Kubernetes [5] 上に展開し，各 NF を Pod として独立管理することでマルチノード構成を実現する．この構成により，物理インタフェースを経由したパケット観測や，NF の柔軟な分散配置が可能となる．さらに，Kubernetes 上で Pod に複数のネットワークインタフェースを割り当てるためのプラグインである Multus CNI [6] を用いて実環境に近いネットワーク構成を再現し，多様な攻撃シナリオの検証を可能にする．本研究の貢献は，従来の単一ノード環境では困難であった個別機能制御を実現し，低コストかつ柔軟なセキュリティ評価基盤を提供することである．これにより，5G 通信のセキュリティ研究における実証的評価の促進が期待される．

2. 5G シミュレーション環境

2.1 概要

5G 通信システムのセキュリティ評価を行うためのアプローチは，実機環境を用いる手法とシミュレーション環境を用いる手法に大別される．

実機環境では，ソフトウェア無線（SDR）や専用の無線機器を用いて実際の RF 信号による通信を行う．この手法は実環境に最も近い条件での評価を可能とする反面，専用機器の導入コストが高く，環境構築に専門的な知識と時間を要するという制約がある．これに対し，シミュレーション環境は，5G 通信システムの各構成要素をソフトウェアで実装し，仮想的な通信環境を構築する手法である．シミュレーション環境の利点は以下の通りである：

- **低コスト：**専用ハードウェアが不要で，汎用的なコンピュータ上で構築可能
- **再現性：**同一条件での実験を容易に再現でき，研究成果の検証が容易
- **柔軟性：**ソフトウェアベースのため，構成変更や機能追加が容易
- **制御性：**各構成要素の動作を詳細に制御・観測可能

特に，セキュリティ評価においては，攻撃シナリオの再現や脆弱性の検証において，制御された環境での実験が重要であり，シミュレーション環境は有効な選択肢となる．

2.2 OpenAirInterface 5G

OpenAirInterface 5G（OAI5G） [4] は，3GPP 標準に準拠した 5G 通信システムのオープンソース実装である．OAI5G は研究・教育目的で開発され，5G コアネットワーク（5GC）と無線アクセスネットワーク（RAN）の主要機能を提供する．

図 1 に実環境とシミュレーション環境における 5G 通信を示す．実環境では物理的な基地局やコアネットワーク機器が使用される一方，OAI5G 環境ではこれらの機能がソフトウェアコンポーネントとして実装されている．

アーキテクチャ

5G コアネットワーク（5GC）は以下の主要な Network Function（NF）から構成される：

AMF（Access and Mobility Management Function）：

© 2025 Information Processing Society of Japan

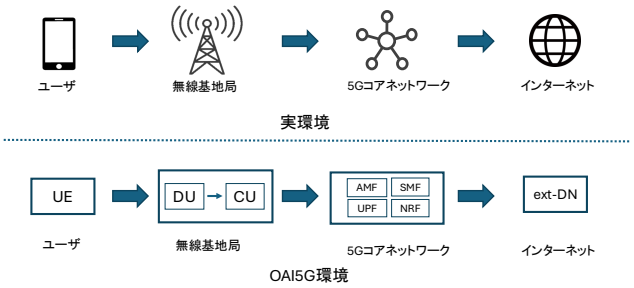

図 1 実環境と OAI5G 環境における 5G 通信
Fig. 1 5G communications in real-world and OAI5G.

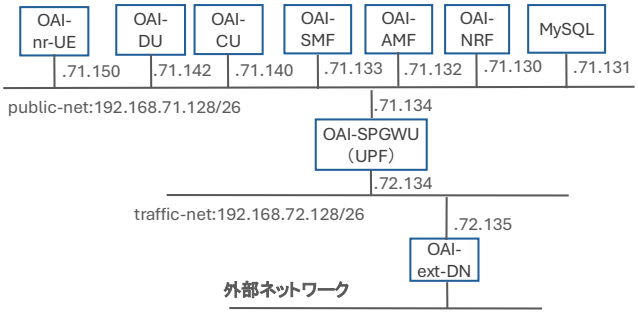

図 2 OAI5G のネットワーク構成
Fig. 2 Network architecture of OAI5G.

接続管理と移動性管理を実行

SMF（Session Management Function）： セッション管理と PDU セッションの制御を実行

UPF（User Plane Function）： ユーザプレーンデータの転送処理を実行

NRF（Network Repository Function）： 各 NF の登録・発見サービスを提供

各 NF は起動時に NRF に自身の情報を登録し，他の NF との通信を動的に確立する Service Based Architecture（SBA）により連携する．

無線アクセスネットワーク（RAN）は以下の要素から構成される：

gNB（gNodeB）： 5G 基地局に相当し，制御処理を行う CU（Central Unit）と無線送受信を行う DU（Distributed Unit）に分離される

UE（User Equipment）： 利用者端末に相当し，gNB との無線通信を行う

図 2 に OAI5G のネットワーク構成を示す．各コンポーネントは定義された通信セグメント（public-net と traffic-net）を介して接続され，実際の 5G 環境に近いネットワーク構成を再現している．

標準構成と制約 OAI5G の標準構成は，Docker Compose を用いた単一ノード構成である．この構成では，すべての NF が同一ホスト上の Docker コンテナとして稼働し，短時間での環境構築が可能である．しかし，セキュリティ評価の観点では以下の制約がある：

通信経路の制約 NF 間通信が Docker の内部ブリッジネットワークで完結するため，物理ネットワークインタフェースを経由した通信の観測や介入が困難である．

構成の固定性 すべての NF が単一ホスト上に配置されるため，実環境のような分散構成や，特定の NF 間通信経路への攻撃者ノードの挿入などの柔軟な構成変更ができない．

処理能力の制約 特に UE-gNB 間通信では大量のパケットが生成されるため，単一ホスト上での処理により負荷が集中し，パケット処理が過負荷により失われる可能性がある．

これらの制約により，標準構成の OAI5G はプロトコルの基本動作確認には適しているが，高度なセキュリティ評価には限界がある．

3. 関連研究

3.1 実機を用いた評価環境

5G 通信システムのセキュリティ評価において，実機を用いた検証環境の構築に関する研究が行われている．

文献 [7] は，USRP-2901 などのソフトウェア無線（SDR）を用いて OAI5G ベースの実験環境を構築した．この研究では，5G のプロトコルスタックをソフトウェアで実装しつつ，無線区間に実際の RF 信号を使用することで，実環境に近い通信試験とシーケンスキャプチャを実現している．その結果，無線区間でのプロトコル挙動やシグナリングを正確に取得でき，高精度な解析と実環境に即した評価が可能となっている．

実機を用いたアプローチの利点は，実際の無線通信環境における物理層からアプリケーション層までの包括的な評価が可能な点である．特に，電波伝搬特性や干渉の影響，実際の端末動作を考慮した現実的な脆弱性評価を実施できる．一方で，この手法にはいくつかの制約がある．第一に，SDR や専用計測機器の導入には高いコストが必要である．第二に，RF 機器の設定や無線環境の構築には専門知識が必要で，環境構築の複雑さが実用上の障壁となる．第三に，特定の機器構成に依存するため，多様な実験条件での再現性の確保が困難である．

3.2 仮想化基盤を用いた展開

3.2.1 Kubernetes 上での OAI5G の展開

AIDY-F2N プロジェクト [8] では，OAI5G を Kubernetes クラスタ上にデプロイし，クラウドネイティブ環境での 5G 通信システムの運用性向上を目指している．この研究では，各 NF を Kubernetes Pod として実行し，Prometheus などの監視ツールを用いたモニタリング機能を実装している．結果として，トラフィック量，遅延，セッション数などの性能指標の可視化と，動的なリソース管理による運用効率の向上を実現している．この手法の利点は，Kubernetes

© 2025 Information Processing Society of Japan

の持つオーケストレーション機能により，NF の自動配置，スケーリング，障害復旧が可能になる点である．また，コンテナ化によりポータビリティが向上し，異なるクラウド環境での再現性が確保される．

3.2.2 その他の仮想化アプローチ

OpenStack や VMware などの仮想化基盤を用いた NFV（Network Function Virtualization）による 5G 環境の構築も研究されている．これらのアプローチは，従来のテレコム環境との親和性が高く，商用環境への適用を想定した研究において有効である．しかし，既存の仮想化基盤を用いた研究の多くは，性能評価やシステム運用に焦点を当てており，セキュリティ評価，特に攻撃シナリオの再現や脆弱性の検証については十分に検討されていない．また，多くの事例では単一ノードでの展開にとどまっており，物理ネットワークを経由した通信の観測や，分散環境での柔軟な構成制御は実現されていない．

3.3 本研究の位置づけ

表 1 に既存研究と本研究の比較を示す．本研究の位置づけと独自性は以下の通りである．

セキュリティ評価への特化：既存の仮想化基盤を用いた研究が主に性能評価や運用効率に焦点を当てているのに対し，本研究はセキュリティ評価に特化した環境構築を目指している．具体的には，攻撃経路の挿入，通信の観測・改ざん，脆弱性の再現といったセキュリティ研究に必要な機能を重視している．

物理経路を活用した観測環境：実機環境では無線区間に限定された物理的観測が可能である一方，従来の仮想化環境では物理経路での通信観測は困難であった．本研究では，マルチノード構成により物理ネットワークインタフェースを経由した通信を実現し，L2/L3 レベルでのパケット観測を可能としている．

個別機能制御の実現：従来の単一ノード環境では，すべての NF が同一ホスト上で稼働するため，個別の NF に対する制御や分散配置が困難であった．本研究では，Kubernetes の機能を活用して NF を Pod 単位で独立管理し，配置制御，リソース制御，通信経路制御を含む個別機能制御を実現している．

低コストかつ柔軟な評価基盤：実機環境の高コストと構築の複雑さ，既存仮想化環境の柔軟性不足を解決するため，本研究ではオープンソースソフトウェアのみを用いた低コストな基盤でありながら，柔軟な構成変更と多様な実験条件への対応を可能としている．

4. 提案手法

4.1 概要

従来の OAI5G 環境は，Docker Compose による単一ノード構成のため，セキュリティ評価に必要な観測・制御機能に制約がある．特に，NF 間通信が Docker ブリッジ内で完結するため物理経路での観測は困難であり，全 NF が同一ホスト上に配置されるため柔軟な構成変更ができないという課題がある．

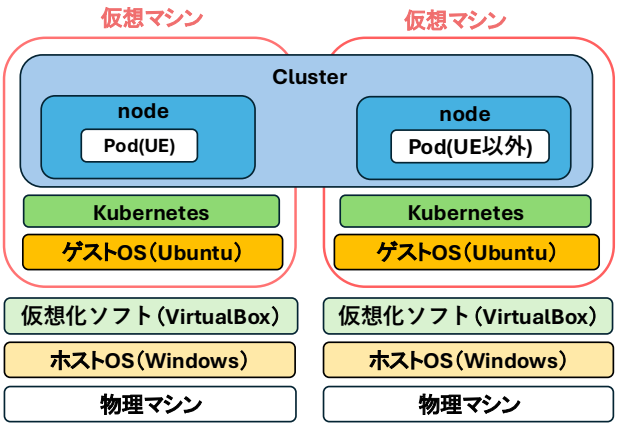

図 3 提案手法の概要
Fig. 3 Overview of the proposed method.

本研究では，これらの制約を解決するため，OAI5G を Kubernetes ベースのマルチノード環境に展開する手法を提案する．図 3 に提案手法の概要を示す．

提案手法では，各 NF を Kubernetes Pod として独立管理し，複数の物理マシンに分散配置する．本構成により，NF 間通信を物理インタフェース経由で実現し，外部からの L2/L3 パケット観測を可能とする．加えて，Pod 管理により，NF の配置制御，リソース制御，通信経路制御を含む個別機能制御を実現する．個別機能制御により，従来環境では困難な，中間者攻撃の再現，通信パケットの改ざん，特定 NF の分離評価など，多様なセキュリティ評価シナリオに対応可能な柔軟な環境を提供する．

4.2 設計要件

提案するセキュリティ評価環境が満たすべき要件を以下の通り定義する．これらの要件は，実環境に近い条件下でのセキュリティ評価と，柔軟な実験環境の構築を両立するために設定した．

要件 A：物理経路での通信観測 NF 間通信を仮想ブリッジのみのパケット転送ではなく，物理インタフェース経由のパケット転送として実現する．これにより，L2/L3 レベルのパケットを外部から取得・解析可能とし，実環境同様の中間者攻撃やパケット改ざんを想定した通信の観測・制御を可能とする．

要件 B：ノード間 L2 ネットワーク共有 UE・gNB などの NF を異なるノードに分散配置した場合においても同一の L2 セグメントを共有可能とする．L2 セグメントを共有することで，ARP・ブロードキャストを含むレイヤ 2 の通信を可能とし，分散環境上でも 5G ネットワークを想定した通信を再現する．

© 2025 Information Processing Society of Japan

表 1　既存研究と本研究の比較
Table 1　Comparison with existing studies.

観点	実機環境 [7]	単一ノード Kubernetes 環境 [8]	本研究
評価対象	セキュリティ評価	性能評価・運用性	セキュリティ評価
コスト	高（専用機器）	低	低
構成の柔軟性	機器依存で限定的	単一ノードで固定的	マルチノードで柔軟
ノード構成	複数ノード	単一ノード	マルチノード
物理経路での観測	無線区間のみ部分的	不可	可
NF 分散配置	不可	不可	可
攻撃シナリオ再現	部分的	不可	可
再現性	環境依存で困難	高	高

要件 C：NF の個別機能制御　NF を個別の Pod として管理し，任意のノードへの配置・隔離，リソース制限，通信経路制御を可能とする．通信経路制御を柔軟とし，評価目的に応じた NF の分離や，通信経路上への観測点・制御点の挿入を容易とし，多様な攻撃シナリオや実験条件を再現可能とする．

要件 D：OAI5G シーケンスの正常動作　上記 A〜C の要件を満たす分散構成においても，OAI5G が提供する登録（Registration），セッション確立（PDU Session Establishment），切断を含む一連の 5G シーケンスが正常に動作することを保証する．要件 D により，攻撃シナリオの検証や脆弱性評価に適用可能な実用的環境を実現する．

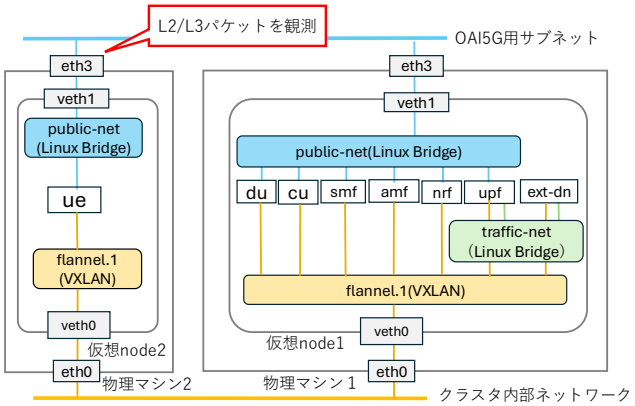

図 4　提案手法のネットワーク構成
Fig. 4　Network configuration of the proposed method.

4.3　システム構成

提案手法のシステムは 2 台の物理マシン上に構築された仮想ノードからなる Kubernetes クラスタで構成される．各ノードの仕様を表 2 に示す．UE を node2 に配置し，その他の主要 NF（DU，CU，SMF，AMF，NRF，UPF，および ext-dn）を node1 に配置し，UE-gNB 間通信を物理インタフェース経由で強制的に通過させる構成とした．

図 4 にネットワーク構成を示す．本システムは「クラスタ内部ネットワーク」と「OAI5G 用サブネット」の二層構成を採用する．

クラスタ内部ネットワークでは，Flannel VXLAN により Pod 間通信を提供する．Flannel VXLAN は Kubernetes クラスタ用の プラグインであり，L2 通信を L3 ネットワーク上で透過的に延長する技術である VXLAN を介して複数ノード間でも仮想的に単一のネットワークセグメントであるかのように通信させる構成を可能とする．

OAI5G 用サブネットでは，public-net と traffic-net を bridge プラグインである Multus CNI で実現する．Multus CNI は Kubernetes 上で Pod に複数のネットワークインタフェースを割り当てるためのプラグインである．各セグメントはノード上の Linux Bridge（public-net, traffic-net）に接続される．このうち public-net は物理インタフェースに対してブリッジし， NF 間通信を物理インタフェース経由で実現する．

Multus CNI を用いて，UPF や ext-dn など必要な NF に対して複数の IP アドレスを静的に割り当てる．Service および FQDN による名前解決と組み合わせ，分散環境においても 5G シーケンスの安定動作を担保する．

本構成により，両ノード間の Pod 通信は Flannel VXLAN トンネルで確立され，各 NF には public-net ブリッジ上の静的 IP が付与される．UPF と ext-dn は traffic-net にも並行して接続し，外部ネットワークとユーザプレーンの疎通を確保する．

5.　機能評価

本章では，4.2 節で定義した設計要件 A〜D に対する提案手法の達成度を検証し，既存研究との比較を通じて本研究の優位性を評価する．

5.1　提案手法の評価

- **要件 A：物理経路での通信観測**
  ゲスト OS 上に Linux Bridge を作成し，仮想インタフェースをブリッジアダプタとしてホスト OS の物理インタフェースに接続した．Linux Bridge の物理インタフェースへの接続により，NF 間通信は物理インタ

© 2025 Information Processing Society of Japan

表 2 提案手法におけるノード仕様
Table 2 Node specifications in the proposed method.

項目	物理マシン 1	物理マシン 2	仮想ノード 1，2（共通）
CPU	Intel Core i5-12500	Intel Core i5-6500	4 vCPU
メモリ	32 GB	32 GB	16 GB
ストレージ	250 GB SSD	120 GB SSD	50 GB
GPU	UHD Graphics 770	HD Graphics 530	—
OS	Windows 10	Windows 10	Ubuntu 22.04 LTS
ネットワーク	—	—	NAT (Flannel VXLAN 用) + ブリッジ (Multus CNI 用)

フェースを介してノード間に転送され，L2/L3 パケットの取得が確認できた．したがって，要件 A は達成した．

- **要件 B：ノード間 L2 ネットワーク共有**
  Multus CNI を用いて UE や DU などの異なるノード上の NF に追加ネットワーク（public-net）を付与した．異なるノード上にある Pod 間で同一セグメント上の IP による疎通を確認した．したがって，要件 B は達成した．
- **要件 C：NF の個別機能制御**
  nodeSelector を用いることで任意の NF の任意ノードへの配置制御や，Resource Quotas を用いたリソース制御機能を確認した．また，public-net 経由の通信経路制御は可能であることを確認した．一方，traffic-net がノード間で未共有のため，UPF と ext-dn を分離配置した場合に動作制約が生じた．traffic-net の L2 ネットワーク共有の適用により，制約は解消される．したがって，要件 C は部分的に達成した．
- **要件 D：OAI5G シーケンスの正常動作**
  Multus CNI による静的 IP 付与と CoreDNS の編集により，OAI5G の NF 間通信は正しく確立した．各 NF は FQDN で相互到達し，標準的な NF 登録・接続手順が正常動作することを確認した．したがって，要件 D は達成した．

5.2 既存研究との比較

OAI5G の標準構成である Docker Compose による単一ノード構成および，3.1 節で紹介した実機環境 [7]，ならびに，単一ノード Kubernetes 事例 [8]，本研究の提案手法を比較した．結果を表 3 に示す．

実機環境 [7] では要件 A は無線区間に限定されるため「部分的に可」，要件 B と D は「可」，要件 C は UE の位置固定により「不可」とした．単一ノード Kubernetes 環境 [8] では要件 A〜C が成立せず「不可」，要件 D は「可」とした．提案手法では要件 A・B・D が「可」，要件 C は一部制約が残るため「部分的に可」とした．比較結果より，提案手法は従来環境で困難な要件 A・B を満たし，要件 C に対しても柔軟性を拡大可能である．

6. 性能評価

本章では，提案手法の実用性を検証するため，環境構築時間と通信パケットの観測性能について評価を行う．

6.1 環境構築時間の評価

6.1.1 評価条件

環境構築時間の評価において，Kubernetes クラスタは事前構築済み，コンテナイメージはローカルキャッシュ済みとした．計測対象は，Kubernetes マニフェスト適用から全 NF（AMF，SMF，UPF，gNB，および UE 等）の起動完了までとし，OAI5G の Docker Compose を用いた単一ノード構成と Kubernetes を用いた提案手法を比較した．

6.1.2 評価結果

表 4 に NF 単位の起動時間，および全体の構築時間を示す．評価結果より，提案手法では NF ごとの起動時間は Docker Compose に比べて長く，特に 5GC のうち AMF，SMF，SPGWU の起動時間は約 35 秒ほど増加している．しかし，全体では約 2〜3 分で収束し，再現性を持って安定して構築可能であることを確認した．

6.2 通信パケットの観測

6.2.1 キャプチャ環境・条件

キャプチャは，UE から gNB へ送信されるパケットが通過する物理マシン 2 の物理インタフェース（eth3）上で実施した．図 4 にキャプチャした場所を示している．キャプチャ環境および条件は表 5 の通りである．

6.2.2 取得結果

取得結果を表 6 に示す．OAI5G の Docker Compose を用いた単一ノード構成では，UE-gNB 間のパケットは起動直後に大量のパケットが発生し，パケットキャプチャ処理が応答不能状態に陥ったため，安定した観測は困難である．一方，提案手法では約 409 万パケットを安定して取得できた．取得パケットの大部分は TCP 関連であり，上位層では TLSv1.2 の通信が継続的に確認された．通信元・宛先の組み合わせは，物理マシン 1 と物理マシン 2，および図 2 で示した DU（`192.168.70.142`）と UE（`192.168.70.150`）の IP アドレスが観測され，本環境におけるノード間通信

© 2025 Information Processing Society of Japan

表 3　環境別比較（要件 A〜D）
Table 3　Comparison across environments (Requirements A–D).

要件	OAI5G 環境 [4]	実機環境 [7]	単一ノード Kubernetes 環境 [8]	提案手法
A：物理経路での通信観測	不可	部分的に可	不可	可
B：ノード間の L2 ネットワーク共有	不可	可	不可	可
C：NF の個別機能制御	不可	不可	不可	部分的に可
D：OAI5G シーケンスの正常動作	可	可	可	可

表 4　NF ごとの起動時間の比較
Table 4　Per-NF startup times.

NF	Docker Compose 環境	提案手法
MySQL	約 5〜10 秒	約 5〜10 秒
NRF	約 10 秒	約 10 秒
AMF	約 10 秒	約 45 秒
SMF	約 10 秒	約 45 秒
SPGWU	約 10 秒	約 45 秒
CU	約 10 秒	約 10〜15 秒
DU	約 10 秒	約 10〜15 秒
UE	約 10 秒	約 10〜15 秒
全体	約 1〜2 分	約 2〜3 分

表 5　キャプチャ環境および条件
Table 5　Capture conditions.

項目	設定内容
キャプチャ場所	物理マシン 2
対象インタフェース	`eth3`（物理インタフェース）
ツール	Wireshark
開始タイミング	UE 起動直前

および UE–DU 間通信に対応していた．

7. 考察

7.1 機能評価に関する考察

本研究で構築した環境は，既存の実機を用いた評価環境 [7] や単一ノード Kubernetes 環境 [8] と比較して，観測可能性と柔軟性を兼ね備えている点に特徴を持つ．具体的には，物理インタフェースを経由した L2/L3 パケットの取得および，複数ノード間における L2 ネットワークを共有可能にすることで，従来環境で困難な UE–gNB 間の通信パケットを安定的に取得・解析可能な基盤を確立した．

一方，NF の個別機能制御は，traffic-net の未共有による通信制御の制約が残り，「部分的達成」である．traffic-net の未共有の課題は，public-net と同様に traffic-net に Multus CNI によるノード間の L2 ネットワーク共有を適用することで解消可能であり，環境拡張の余地を示している．

機能評価より，提案手法は実機環境と比較して柔軟で構築容易性が高く，単一ノード環境に比べて観測可能性が広い点において柔軟性と可観測性の両立を実現した評価環境と位置づけられる．したがって，本環境は既存の研究環境に対する有効な代替手段として，5G 通信の脆弱性評価に貢献できる．

7.2 構築時間に関する考察

本研究で提案した Kubernetes によるマルチノード環境の構築時間は表 4 で示したように，OAI5G の Docker Compose による単一ノード環境と比較して約 1 分長く，全体で 2〜3 分を要した．この差異の主因は，AMF，SMF，SPGWU において導入した Kubernetes の機能である initContainer による初期化処理であり，各 NF 起動に約 35 秒を追加したことによる．一方，MySQL，NRF，CU，DU，UE などは Docker Compose 環境とほぼ同等である．

initContainer の導入は，NRF を除く 5GC（AMF, SMF, SPGWU）の NF 起動初期に NRF 登録要求が失敗する事象が発生したことによる．登録要求の失敗は，Pod 起動時に Multus CNI を含む CNI によるネットワーク設定の完了前に NRF を除く 5GC の NF の起動や，NRF 内部の HTTP サーバの listen 遅延が影響すると考えられる．

構築時間の評価より，起動時間増加は NF 実装ではなく，Kubernetes のネットワーク初期化や依存関係制御に起因すると考えられる．なお，全体で 2〜3 分程度の時間は研究利用に十分許容可能であり，観測可能性や柔軟性の利点を損なうものではない．

7.3 通信パケットの観測に関する考察

提案手法による環境では，表 6 に示したように 409 万パケット（約 6.2 GB）を取得でき，1 秒あたり約 52,000 パケットに相当するレートを処理可能である．一方，OAI5G の Docker Compose による単一ノード環境では，大量のパケット処理によりゲスト OS 上で負荷が増大し，安定したキャプチャは実現困難である．両者の差異は，Docker Compose による単一ノード環境が VM の割り当てリソースに依存するのに対し，本研究の Kubernetes 環境ではホスト OS の性能を直接活用可能なため，処理能力が向上したことに起因すると考えられる．したがって，本環境は 5G ネットワークの観測において，従来よりも安定したキャプチャを可能にする有効な基盤である．

また，キャプチャパケットの内容解析より，パケットの多くは TCP セッションの確立パケットおよび Wireshark 上で TLSv1.2 の Application Data と表示されるパケットであった．TCP パケットが多くを占めたのは，OAI5G の

© 2025 Information Processing Society of Japan

表 6　キャプチャ結果の比較
Table 6　Packet capture results.

環境	総パケット数	キャプチャ時間	データサイズ	平均パケットレート	備考
提案手法（UE 隔離構成）	約 4,090,000	77.5 秒	約 6.2 GB	約 52 kpps	安定してキャプチャ可能
Docker Compose（公式構成）	測定不能	—	—	—	負荷増大による処理停止

rfsimulator の実装特性により，UE–gNB 間で交換される無線リソース制御（RRC: Radio Resource Control）や非アクセス層シグナリング（NAS: Non-Access Stratum）などのメッセージが TCP プロトコルにカプセル化されるためである．Wireshark 上では通信の存在やトラフィック規模は観測可能だが，RRC/NAS メッセージを平文で，5G の通信シーケンスとして観測不可であると確認した．

一方， CU–DU 間においては，暗号化の影響を受けずに 5G の通信シーケンスを平文としてキャプチャが可能であった．TCP のカプセル化の影響は UE–gNB 間に限られるため，研究目的に応じて観測対象とするネットワーク機能間を選定することが重要である．

さらに，Wireshark 標準の TLS ディセクタでは Application Data の復号は不可能であることから，OAI 特有の NAS を解析可能な 専用ディセクタやログ解析ツールが求められる．これらのツールにより，シミュレータ特有の TCP のカプセル化に対応し，より詳細な 5G の通信シーケンスの解析が可能となり，本環境を用いた 5G 脆弱性評価の精度向上につながる．

8. おわりに

本研究では，OAI5G を Docker Compose 単一ノード構成から Kubernetes マルチノード構成へ拡張し，5G 通信のセキュリティ脆弱性評価基盤を構築した．提案手法により，従来は困難な UE-gNB 間および NF 間通信の L2/L3 パケットを物理インタフェース経由で取得可能とした．また，各 NF を Pod 単位で独立管理し，配置制御・リソース制御・通信経路制御を含む個別機能制御を実現した．この制御により，Docker Compose による単一ノード構成では困難な，中間者攻撃やパケット改ざんを含む多様な攻撃シナリオの再現に加えて，特定 NF の分離評価を可能にし，セキュリティ評価に適した柔軟性と拡張性を備えた基盤を確立した．機能評価では，設定した 4 つの設計要件のうち 3 つを完全達成し，1 つを部分達成した．性能評価では，約 52,000 パケット/秒の安定した観測性能を実現し，研究利用に十分な実用性を有することを実証した．既存研究との比較においても，セキュリティ評価に必要な観測可能性と制御柔軟性を両立した環境であることが確認された．

一方，本研究で提案した環境には解決すべき課題が残されている．第一に，攻撃シナリオの実装である．5G の脆弱性として報告されている暗号化アルゴリズムのダウングレード攻撃や，攻撃者ノードの挿入によるメッセージの改ざんを再現し，実際の攻撃が通信シーケンスに及ぼす影響の評価が必要である．第二に，プロトコル解析機能の強化である．UE-gNB 間通信が TCP トンネルにカプセル化されるため，Wireshark では TLS Application Data となり，RRC や NAS メッセージの平文解析が困難である．今後は OAI 固有の実装に対応したディセクタやログ解析ツールを導入し，RRC や NAS の 5G シーケンスを平文として解析することが必要である．今後は，攻撃シナリオの実装と解析機能の強化により，本研究で提案した基盤の発展性を実証し，5G ネットワークにおけるセキュリティ評価の実効性を高めることを予定している．

謝辞　本研究の一部は，JSPS 科研費 JP25K07749，JP23K03847 の助成を受けたものです．

参考文献

[1] Jover, R.P. and Marojevic, V.：Security and Protocol Exploit Analysis of the 5G Specifications, *Proc. IEEE Wireless Communications and Networking Conference* (WCNC 2019), IEEE, pp.1-6（2019），DOI:10.1109/WCNC.2019.8641117.

[2] Basin, D., Dreier, J., Hirschi, L., Radomirović, S., Sasse, R. and Stettler, V.：A Formal Analysis of 5G Authentication, *Proc. ACM Conference on Computer and Communications Security* (CCS 2018), ACM, pp.1383-1396（2018），DOI:10.1145/3243734.3243846.

[3] Zhang, R., Zhou, W. and Hu, H.：Towards 5G Security Analysis against Null Security Algorithms Used in Normal Communication, Security and Communication Networks, Vol.2021, Article ID 4498324,（online），DOI:10.1155/2021/4498324（2021）.

[4] OpenAirInterface Project：OpenAirInterface 5G Repository（online），available from 〈https://gitlab.eurecom.fr/oai/openairinterface5g〉（accessed 2025-08-17）.

[5] The Kubernetes Authors：Kubernetes Official Documentation（online），available from 〈https://kubernetes.io/ja/〉（accessed 2025-08-17）.

[6] Kubernetes Network Plumbing Working Group：Multus CNI GitHub Repository（online），available from 〈https://github.com/k8snetworkplumbingwg/multus-cni〉（accessed 2025-08-17）.

[7] Sawamoto, T., Suzuki, M., Osako, Y., Kasama, T., Inoue, D. and Nakao, K.：Inherited Threat Reproduction on Open Source 5G Testbed, *Proc. IEEE Conference on Dependable and Secure Computing* (DSC), IEEE Computer Society, pp.9-16（2024），DOI:10.1109/DSC63325.2024.00014.

[8] AIDY-F2N：OAI-UERANSIM GitHub Repository（online），available from 〈https://github.com/AIDY-F2N/OAI-UERANSIM〉（accessed 2025-08-17）.

© 2025 Information Processing Society of Japan