修士論文

自律分散モバイルシステム環境における AAA の実現

髙田 敦生

主指導教員 宇多 仁

北陸先端科学技術大学院大学
先端科学技術専攻
（情報科学）

令和 8 年 3 月

Abstract

The mission of telecommunications operators is to ensure continuous provision of communication services under any circumstances, including disasters and accidents. While mobile systems serve as fundamental infrastructure that continuously accommodates tens of millions of devices, large-scale failures triggered by physical malfunctions, equipment damage, or operational errors have been increasing in recent years. In mobile systems, client-server-style control operates through the architecture, where the CN acquires and processes subscriber information to generate the necessary contextual data for authentication/authorization, mobility management, and circuit management, with each network function (NF) managing this data in local storage. For communication service continuity, these contexts must remain mutually consistent and accessible. However, disruptions, congestion, or NF failures can cause inconsistencies and require context recreation, potentially triggering increased control communications between NFs that could cascade into service disruptions. Particularly notable failure scenarios include: (a) failure to complete context generation due to signaling storms caused by simultaneous device connections, (b) inability to access contexts due to RAN – CN disconnections, and (c) loss of contexts resulting from NF/RAN downtime.

Particularly, when signaling storms caused by mass device connections prevent context generation completion (a), when RAN – CN disconnections make context access impossible (b), or when NF/RAN failures cause context loss (c), these represent typical failure modes. The Autonomous Distributed Mobile System (ADM) has been proposed as a highly resilient architecture enabling service continuity even under regional disruptions. However, since subscriber information required for AAA operations is typically centralized in databases, distributing information at scale across a wide area would be impractical from both operational and cost perspectives. Therefore, this study first proposes inserting a Replication Layer between UDR and ADM to enhance subscriber information availability. The Replication Layer transparently intercepts requests from ADM destined for UDR, handling both forwarding to UDR and returning responses. Furthermore, it maintains cached copies of subscriber information returned from UDR, allowing the Replication Layer to respond without querying the central UDR when the same information is requested again. This approach minimizes implementation changes on the ADM side while reducing dependence on central systems during disruptions and congestion, thereby improving continuity of AAA-related information retrieval.

Additionally, the traditional ADM architecture's centralized storage of all UE

contexts in each node's local storage (e.g., Redis) presents challenges in terms of scalability and load concentration during failures. Therefore, this study proposes a distributed management approach to improve context availability in ADM environments, based on the requirements of: localized control processing, loss-tolerant context management, secure normalization after recovery, and minimal modification of existing signaling protocols. The proposed method horizontally distributes contexts across a DHT, with each node maintaining only a portion of the key space while enabling context node discovery through internal DHT routing. This ensures continuous context reference even during mobility or disruptions. Moreover, by eliminating context transfer during handovers, it simplifies and reduces control signaling while ensuring scalability to accommodate hundreds of thousands to millions of UEs through additional node additions. This paper presents the design principles and implementation of the above approach and evaluates its effectiveness in terms of service continuity under various failure conditions including disruptions, congestion, and system failures.

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　研究背景

モバイルシステムは社会基盤として常時数千万規模の端末を収容する大規模なシステムである。第五世代モバイルシステム（5GC）では (図 1.1)、機能分割された NF（Network Function）がマイクロサービスのように役割を担い、相互に連携しながら認証認可・移動管理・回線管理などの制御を行う。こうした制御処理は UE–RAN–CN からなるクライアント・サーバ型の構成で動作し、各 NF が生成・保持するコンテキストの整合性がサービス継続の前提となる。一方、近年は物的故障・設備破損・運用ミスなどをきっかけとした大規模障害が増加しており、部分的な障害が制御系の輻輳や不整合を誘発し、通信断へ連鎖するリスクが顕在化している。

通信事業者に求められる要件は、災害や事故を含むいかなる状況下でも通信サービスを継続的に提供し続けることである。実運用では常時 1,000 万オーダーの端末を収容し、24 時間 365 日、通話・データ通信を提供することが求められる。5GC においては NF が機能単位で分割され、各 NF が加入者情報・セッション情報などのコンテキストを扱うことでサービスが成立する。したがって、巨大規模の端末収容と高い可用性を同時に満たすシステム設計が不可欠である。近年、国内外で大規模な通信障害が増加している。例えば、KDDI（2022 年 7 月）や AT&T（2024 年 2 月）などで、数時間規模の通話・データ通信の停止が報告された [2, 3]。こうした障害は物的故障・設備破損・オペレーションミスなど多様なきっかけから発生し、完全に排除することは非現実的である。加えて、災害時には設備損傷や停電、伝送路断といった障害に加え、安否確認や避難に伴う通信集中が同時に発生する可能性がある。 モバイルシステムはきっかけが発生しても大規模障害に波及しづらい制御系の設計や、障害時にもサービスを継続可能なアーキテクチャの検討が重要である。

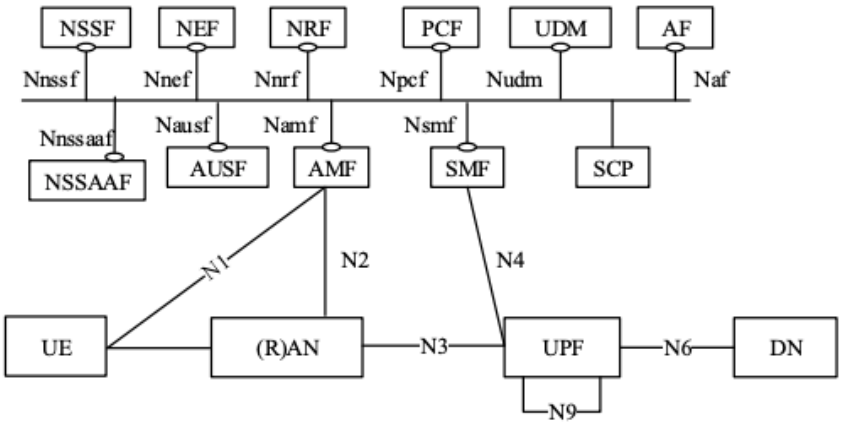

Figure 4.2.3-1: 5G System architecture

図 1.1: 5GC のアーキテクチャ 文献 [1] Fig 4.2.3-1 より抜粋

1.1.1 研究目的

モバイルシステムは常時 1,000 万台規模の端末を収容する社会インフラである一方、近年は物理的故障・設備破損・オペレーションミス等を契機とした大規模障害が増加している。これらの要因を完全に排除することは現実的ではないため、本研究は「障害のきっかけが発生しても大規模障害に発展しにくい」モバイルシステム設計を検討し、サービス継続性の向上を目的とする。

1.1.2 モバイルシステムの一般的挙動とコンテキスト生成

モバイルシステムでは、UE–RAN–CN の連携により各種制御が行われる。UE の接続要求を受けた CN は加入者情報を取得・処理し、認証認可、移動管理、回線管理に必要なコンテキストを生成する。各 NF はコンテキストをローカルストレージで管理し、NF 間のメッセージ交換を通じて整合性を保ちながら利用する。コンテキストの整合性が維持されている場合に限り、UE は正常に通信サービスを利用できる。不整合が発生すると、それを打ち消すための制御が追加で発生し、NF 間通信が増大することでさらなる輻輳や通信断を誘発し得る。この構成は、UE をクライアント、CN をサーバとするクライアント・サーバ型アプリケーションに類似している。RAN と CN はキャリア閉域網（TN: Transport Network, IP 伝送網）を介して接続され、UE からの要求は RAN を経由して CN に到達し、CN が処理・状態更新して応答する。CN は加入者情報を取得・処理しながら「コンテキスト（ステート）」を次々に生成し、認証認可・移動管理・通信回線管理といったあらゆる制御はこのコンテキストに従って実行される。各 NF は NF 間のメッセージ交換

に応じてコンテキストを生成し、各 NF 内のローカルストレージで管理するため、NF 間の整合性が維持されることが正常通信の前提となる。

1.1.3　モバイルシステムにおける AAA の重要性

AAA（Authentication/Authorization/Accounting）は、加入者の正当性確認、利用権限の判断、利用履歴の記録を担う機能であり、モバイルシステムの根幹を成す。どの加入者がどの程度サービスを利用できるか、どの基地局からどの基地局へ移動したか、どの回線が確立されるべきかといった制御は、すべてコンテキストに基づいて行われる。RAN/CN が有効な UE コンテキストを利用可能であれば、UE は通信可能であると言える。従来の標準的な管理では、NF 間のメッセージ交換のたびに各 NF がコンテキストを生成・保持し、相互に整合している場合にのみサービスが成立するため、コンテキストの可用性と整合性はサービス継続性に直結する重要な要素である。モバイルシステムにおけるサービスの継続性とは、(1) 誰がどのくらいサービスを利用していたかという情報が認証・認可・課金によって保証され、(2) 端末がどの基地局からどの基地局へ移動したかが移動体管理によって把握され、(3) それらの情報を基に通信路が確立されてサービス提供が継続できる状態を指す。(図 1.2)

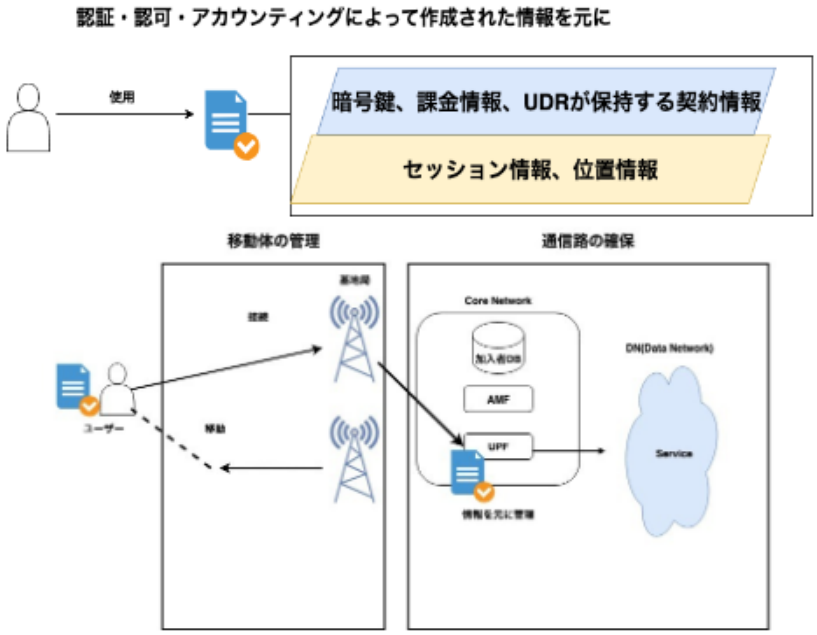

図 1.2: モバイルシステムにおけるサービスの継続性

ここで、モバイルシステムで使用される情報について整理する。加入者情報と

は永続的・静的なユーザ属性情報であり、課金プロファイルや認証鍵などを含む。本来これらは加入者 DB（UDR）に保持され、UDM が管理・参照する。一方、コンテキスト情報とは加入者情報から生成される動的なユーザ属性情報であり、基地局情報やセッション情報などを含む。本来これらは各 NF が保持する。モバイルシステムでは、CN 内の認証系 NF が加入者データベース（UDR）に保存されている加入者情報を用いて認証を実施し、通信サービスを提供する。また、CN は通信サービスの利用状況に基づくアカウンティングも実施する。したがって、加入者情報を受け取れない限り、CN は利用者に対して通信サービスを開始できない。ADM（Autonomous Distributed Mobile System）においても、加入者情報を管理する User Data Repository（UDR）は従来の CN 設計と同様に中央に配置される。1,000 万以上の加入者情報を管理するデータベースシステムを RAN と同様に広域へ分散させることは、運用の利便性とシステムの規模性の観点から現実的ではないと考えられる。しかし、各 ADM ノードから地理的に隔たった中央のみに加入者データベースが存在する状況は、ADM ノードと加入者データベース間の Transport Network（TN）に障害が発生する可能性を十分に考慮できていない。ADM の耐障害設計を満たすためには、中央に UDR を置く前提を維持しつつも、可用性の高い加入者情報取得方式が必要となる。

1.1.4 通信障害として観測される事象の整理

モバイルシステムの制御プレーン（C-plane）は、UE の要求に応じて加入者情報を参照しながら認証・認可、移動管理、セッション確立等の手続きを進め、その過程で複数の Network Function（NF）がコンテキスト（状態）を生成・更新する。これらのコンテキストは NF ごとのローカルストレージに保持され、NF 間の相互整合が成立して初めて「有効な状態」として利用可能になる。ところが、障害や負荷変動が生じると、コンテキストの生成・参照・更新が部分的に失敗し、NF 間で状態が食い違う。この不整合を「補う／打ち消す」ために、再試行・再同期・再生成・ロールバックといった収束処理が必要となり、その結果として収束シグナリングが増加する。特に本研究が対象とする失敗形態は、(a) 生成未完了、(b)(c) 生成済み状態へのアクセス不能、の 2 系統に整理できる。

(a) コンテキスト生成が完了しない（輻輳・シグナリングストーム）端末の一斉接続（メンテナンス明け、切り戻し後、災害直後など）により、登録・認証・セッション確立要求が短時間に集中すると、CN 内の NF（例：AMF/SMF 等）に処理待ちが発生し、タイムアウト・再送・再試行が連鎖してシグナリングストーム（輻輳）へ発展し得る。このときコンテキストは「途中まで生成されたが完了しない」状態が多数発生し、NF 間で不整合な中間状態が残る。結果として、再試行や状態の巻き戻しが多発し、収束のための制御通信がさらに負荷を増幅する。

(b) 生成済みコンテキストにアクセスできない（RAN – CN 分断）災害等により RAN と CN を接続する伝送網（TN）が分断されると、CN 側にコンテキストが

存在していても RAN/UE 側から参照できず、手続きが停止する。また、分断中に UE 側で再登録・再確立が発生すると、CN 側の状態と UE 側の観測が乖離しやすく、分断回復後に一斉再接続が起きると、(a) と同様の輻輳を誘発し得る。ここでは「状態はあるが到達できない」ことが根本原因となり、再試行やフェイルオーバ手続きが収束シグナリングを増加させる。

(c) 生成済みコンテキストが消失・生成機能不全（NF/RAN ダウン）NF や RAN ノードのダウン、あるいはストレージ障害が発生すると、生成済みコンテキストが失われる、または更新ができなくなる。すると、当該 UE に対する手続きは再生成や再認証などの回復処理へ移行し、状態復元のための NF 間通信が増える。さらに、障害前後で「どの状態が正しいか」を確定できない場合、整合性回復のための再同期が必要となり、これが収束シグナリングを増加させる要因となる。

(d) 加入者情報にアクセスできない（UDR 障害）UDR が到達不能または機能停止した場合、認証・認可に必要な加入者情報が取得できず、CN 内の認証系手続きが開始できない。結果として、登録・認証・セッション確立が停止し、UE は通信サービスを開始できない。さらに、再試行が集中すると認証系 NF や UDR 周辺へのシグナリングが増加し、(a) の輻輳と同様に負荷を増幅させる可能性がある。UDR は中央配置が前提となるため、TN 分断や中央障害の影響が直接的に表出しやすい点が本課題の特徴である。

以上より、モバイルシステムにおける整合性問題と収束シグナリング増大は、(a) 輻輳に起因してコンテキスト生成が未完了のまま残存する場合と、(b) 分断により生成済みコンテキストへ到達できない場合、(c) ノード／ストレージ障害により生成済みコンテキストが消失または更新不能となる場合、という三つの失敗形態に整理できる。いずれのケースでも、NF 間で「どの状態が正しいか」を確定できない、あるいは確定しても状態を参照・更新できないことが引き金となり、再試行・再同期・再生成・ロールバックなどの収束処理が発生する。収束処理は本来、整合性を回復してサービスを正常化するための制御であるが、障害や負荷が継続する状況ではそれ自体が追加の制御通信と状態更新を生み、結果として制御プレーン負荷を増幅し、輻輳や通信断の長期化を招き得る。したがって、サービス継続性を高めるためには、(a)〜(c) の発生要因を前提として、収束処理を「起こさない／増やさない」設計、すなわち制御の局所化、状態の損失耐性、到達性の確保、および復旧後に安全に正常状態へ収束させる仕組みを統合的に検討する必要がある。

本研究の具体的目的は、自律分散モバイルシステム（ADM）において、従来のモバイルシステムが提供するサービスレベルを維持しつつ、部分的なネットワーク分断やノード障害といった過酷な条件下でも、認証・認可・アカウンティング（AAA）機能およびそれに付随するコンテキスト管理を継続的に提供できる方式を設計・実装し、その有効性を検証することである。これにより、分断環境下でハンドオーバー等が発生した場合でも、音声通話やインターネット接続を含むデータ通信サービスの継続を可能にすることを目指す。

1.2　本論文の構成

本論文は、本章を含めて 6 章で構成される。第 2 章では、モバイルシステムにおける AAA/コンテキスト管理の既存研究を整理し、標準構成や UDSF、Proc5GC、SpaceCore、ADM など関連手法の特徴と課題をまとめる。第 3 章では、本研究が提案するアーキテクチャおよび設計方針を示し、障害パターン（a–c）への対応方法を述べる。第 4 章では、提案方式の実装と評価方法を説明し、評価観点・指標および実験環境を示す。第 5 章では、得られた評価結果を基に考察を行い、実運用への適用可能性を議論する。第 6 章では、本研究のまとめと今後の展望を述べる。

第2章　関連研究

2.1　コンテキスト可用性向上に関する既存アプローチ

2.1.1　処理系のスケーラビリティ向上

モバイルシステムにおける処理系のスケーラビリティ向上は、制御処理の集中を避け、負荷や障害の影響を局所化する観点で重要である。代表的には次の二つの方向性がある。

A. 地理的分散処理（分断耐性・負荷分散）　処理機能を地理的に分散配置し、地域ごとのトラフィックや障害を局所化する方式である。RAN 近傍に制御処理を配置することで分断時にも当該領域内でのサービス継続性を確保しやすく、同時に広域の負荷を地域単位で分散できる。ただし、分散された処理間での整合性維持や、広域移動時の状態引継ぎが課題となる。

B. クラウド処理（並列処理による輻輳耐性）　クラウド基盤上で制御処理をスケールアウトし、並列処理によりピーク負荷に耐える方式である。輻輳時に処理リソースを動的に増強できるため、シグナリングストームの影響を緩和しやすい。一方で、状態を持つ NF のスケールアウトにはコンテキスト分散や同期が必要となり、状態管理の複雑化や遅延増加を招き得る。

2.1.2　データ管理の高度化

データ管理の高度化は、コンテキストの可用性を高め、障害時の再生成やシグナリング増加を抑える観点で重要である。代表的には次の二つの方向性がある。

C. 分散管理（一極集中ではない）　コンテキストや関連データを複数のノードに水平分散して保持する方式である。単一ノードへの集中を避けることで負荷分散が可能となり、局所障害時の影響範囲を限定できる。ただし、分散されたデータの一貫性維持や探索コストが課題となる。

D. 冗長管理（同一データの複製）　同一データを複数箇所に複製することで、損失抑制と到達性の向上を図る方式である。障害時にも別拠点からデータを参照できるため、再生成を抑制し、結果として収束シグナリングの増加を抑える効果が期待できる。一方で、更新の伝播や整合性確保のコストが発生するため、複製範囲と更新頻度の設計が重要となる。

2.2　関連システム・研究の整理

2.2.1　標準構成+UDSF

3GPP では、加入者情報を格納する UDR とは別に、標準化されていない非構造データを格納する UDSF（Unstructured Data Storage Function）が定義されている。UDSF は NF が生成する UE コンテキストやセッションデータなどの動的データを外部化し、参照・更新・削除できるストレージ NF として機能する。これにより NF のステートレス化が可能となり、NF 障害時でも他の NF が UDSF から同一データを取得して処理を継続できる。UDSF には Nudsf_DataRepository と Nudsf_Timer の API が定義され、KVS に近いアクセスモデルで運用できるため、コンテキスト喪失に伴う再生成やシグナリング増大の抑制が期待される。

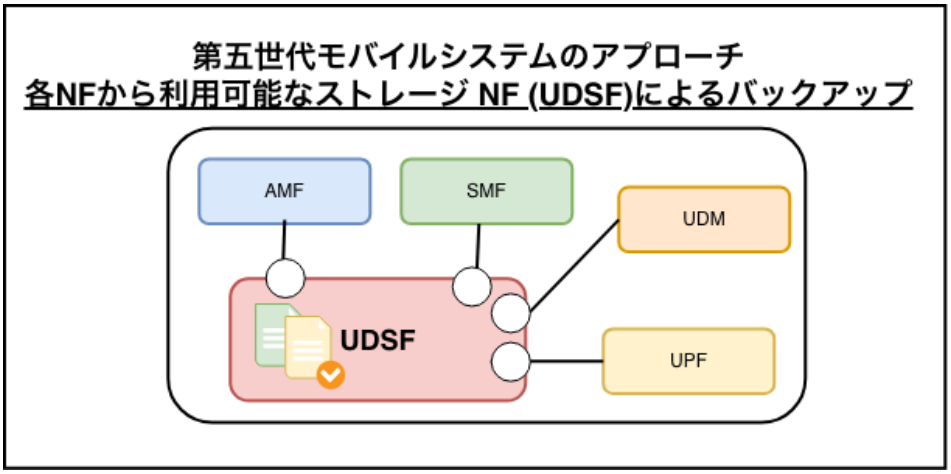

図 2.1: UDSF の概念図

2.2.2　Proc5GC

Proc5GC[4] はプロシージャ型処理により 5GC の制御プレーンをステートレス化する方式である。5G の各プロシージャを個別のソフトウェアとして実装し、UE からの要求ごとにプロセスを起動・実行・終了するリアクティブな実行モデルを採用する。UE コンテキストや状態は外部ストレージ（DB）に集中保存され、CN の C-plane はサーバレスに近い構成となる。これにより、NF ごとの状態維持やコンテキスト移行が不要となり、UE コンテキストの集中管理が可能となる。

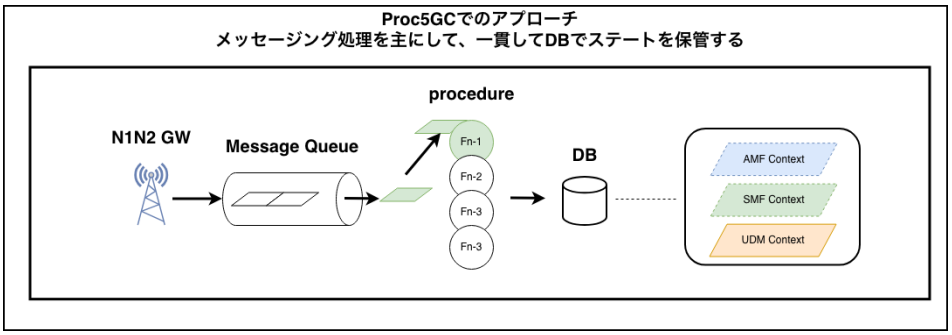

図 2.2: Proc5GC の概念図

2.2.3 SpaceCore

SpaceCore[5] は、LEO 衛星網へモバイルコア機能を押し上げる際に顕在化する「極端な移動性・断続的故障・不信な配置場所」といった条件下で、従来のステートフルなコアが引き起こすシグナリング増大（登録・位置管理・状態移送など）を主要なボトルネックとして捉える。これに対し SpaceCore は、コア機能から状態（コンテキスト）を分離する設計を採用し、軌道上のコア機能自体を可能な限りステートレスに近づける。

コンテキスト管理に関して、SpaceCore は位置に関わる状態を地理空間アドレッシングにより単純化し、衛星の移動に伴う不要な状態移送を地理的サービスエリアへの置き換えで抑制し、状態取得を device-as-the-repository（端末を状態リポジトリとして活用）することで局所化する、という方針をとる。これにより、衛星移動や断続故障が起きても、コンテキスト参照のための制御シグナリングが増えにくい構造を目指す。

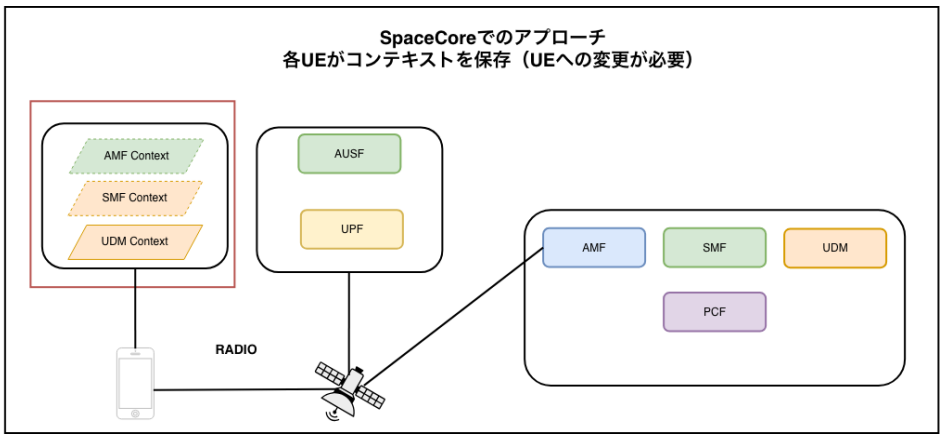

図 2.3: SpaceCore の概念図

2.2.4 ADM(Autonomous Distributed Mobile System)

ADMobile(ADM)[6] は制御プレーン（C-plane）機能を RAN 側へ統合する RAN-Core Convergence を前提とし、UE コンテキストの生成・整合性管理を各 RAN（AD-RAN）内で完結させることで、中央集約型で起きやすい輻輳や分断時の連鎖的な不整合拡大を抑える設計をとる。公開資料では、UE あたりの整合性管理場所が 1 か所（各 RAN）であり、中央に残るのは加入者 DB のみという整理が示されている。

このとき ADM におけるコンテキストは、IMSI 等の識別子に対応づく形で各 AD-RAN により管理され、加入者 DB 参照を除く C-plane 処理が局所化されることで、端末の一斉接続等で生じるシグナリング集中を起こしにくくすることを狙う。また、地域的分断が発生しても RAN 内部で C-plane が完結している限り、分断領域内での通信継続や分断領域内の移動を許容しやすい性質が示されている。

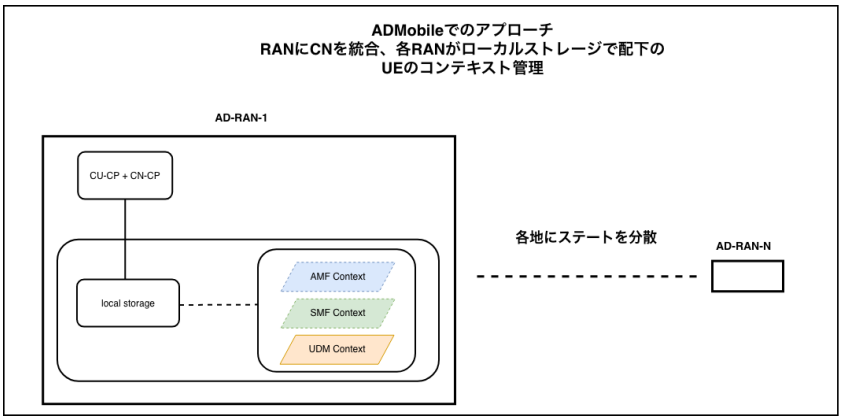

図 2.4: ADM の概念図

2.2.5 比較と対象パターンへの対応関係

各関連研究・システムの特徴と本研究が対象とする障害パターン（a–c）への対応関係を整理する。

表 2.1: 関連研究・システムと対応パターンの整理

研究	概要	アプローチ	対処可能な現象
3GPP 標準	各 NF から利用可能なストレージ NF によるバックアップで C-plane のステートレス化を支援	D	c
Proc5GC/ Cloud5GC	プロシージャ単位で処理を起動・実行・終了し、常駐 NF を排した C-plane 構成	B	a, c（部分）
SpaceCore	各 UE がコンテキストを保持し、端末側を状態リポジトリとして活用（端末変更が必要）	C	a, c（部分）
ADMobile	RAN に CN を統合し、各 RAN が配下 UE のコンテキストをローカル管理	A, C	a, b

表 2.1 より、既存手法は (a) 輻輳、(b) 分断、(c) 消失のいずれかに重点を置く傾向があり、全てを同時に満たすアプローチは限定的である。特に (b) の RAN–CN 分断に対処可能な ADMobile は制御の局所化により分断耐性を備える一方、コンテキスト消失への耐性は冗長化の度合いに依存する。そこで、ADMobile の局所化基盤に (D) データ冗長管理のアプローチを組み合わせることで、(a) 輻輳、(b) 分断、(c) 消失の三つの失敗形態を横断的に緩和できる可能性がある。これが本研究における位置づけである。

2.3 既存手法の課題と本研究の位置づけ

2.3.1 本研究が解く課題の整理

既存手法は、処理系のスケーラビリティ向上（A, B）とデータ管理の高度化（C, D）のいずれかに重心を置き、特定の障害パターンに対して有効性を示している。一方で、(a) 輻輳に起因するコンテキスト生成未完了、(b) 分断による到達不能、(c) 消失・生成機能不全を同時に扱うには、制御の局所化とコンテキストの冗長化を両立する必要がある。本研究が解く課題は、ADM のように分断耐性を備えたアーキテクチャに対し、中央 UDR 依存を低減しつつ、AAA 関連情報とコンテキストの可用性を高める仕組みを設計・実装することである。具体的には、加入者情報の取得継続性を確保し、コンテキストの分散・冗長管理を組み合わせ、(a)〜(c) の障害下でも通信サービスを継続可能な制御基盤を実現する点にある。今回の研究では、ADM の局所化基盤を前提としつつ、加入者情報の冗長管理を組み合わせる

ことで (a) 輻輳，(b) 分断，(c) 消失の三つの失敗形態を横断的に緩和できる可能性を示す。また 5G 基地局（RU: Radio Unit）が 10 万局を超える規模で展開されている一方、CN 機能を統合した AD-RAN を高密度に配置することは難しい．そのため，本研究では AD-RAN の数を 100〜1000 程度と想定し，従来のモバイルシステムが提供するサービスレベルを維持しつつ、AAA 機能およびコンテキスト管理を継続的に提供可能な方式を提案する。

第3章　加入者情報の可用性向上手法

本章では、加入者情報に関する障害パターン (d) への対応を中心に、提案方式の設計と可用性向上手法を述べる。

3.1　概要

ADM ではコントロールプレーンの局所化により分断耐性を高める一方、加入者情報は従来と同様に中央の UDR で管理する前提に立つ。1,000 万以上の加入者情報を広域に分散配置することは運用・コスト両面で現実的でなく、UDR の中央配置は実務上妥当な選択である。しかし、この前提は UDR 到達性に強く依存するため、TN 分断や UDR 障害が発生すると、局所化された C-plane であっても認証・認可が停滞し得る。従って、中央 UDR を前提としつつも、加入者情報の可用性を高めるシステムが必要となる。

3.2　設計の考慮

自律分散モバイルシステムにおいては、加入者情報を管理する UDR は、従来の CN 設計と同様に中央に配置される。1,000 万以上の加入者情報を管理する UDR を RAN と同様に広域へ分散させることは、運用の利便性とシステムの規模性の観点から現実的ではないと考えられる。しかし、各 AD-RAN から地理的に隔たった中央のみに加入者データベースが存在する状況は、AD-RAN と加入者データベース間の Transport Network（TN）に障害が発生する可能性を十分に考慮できていない。そのため耐障害性を志向する ADM の設計意図を十分に満たしているとは言えない。そこで、中央に UDR を置く前提を維持しつつ、加入者情報の取得継続性を確保する設計が必要となる。そのためには、AD-RAN/UDR の実装を改変しない透過性、TN 分断時でも利用可能なデータの近傍配置、一貫性低下を許容しつつ運用可能な範囲での整合性確保が求められる。透過性は、運用中のモバイルネットワークに対して大規模改修を伴う導入が現実的ではないために必要である。近傍配置は、TN 分断時に中央 UDR への到達性が失われても認証手続きを継続できるようにするためである。整合性確保は、更新の反映遅延や一時的不一致が避けられない中で、運用上許容できる範囲を明確化し、サービスの一貫性と課金の正確性を維持するために必要となる。

3.3 実装概要

本節では Replication Layer の実装方針を示す。

3.4 Replication Layer

Replication Layer は、AD-RAN と UDR の間に挿入される中間層として、加入者情報要求の代理受信と応答を担う。UDR と AD-RAN 双方に対して透過に動作し、既存インタフェースを維持したまま可用性を向上させることを設計方針とする。Replication Layer はリクエスト受付、UDR 転送・応答、キャッシュ管理、監視・統計収集の各機能で構成され、ADM からの要求はキャッシュ照合を経てヒット時は即時応答、ミス時は UDR 転送とキャッシュ格納を行う。このフローにより、同一加入者情報への再要求が集中する状況でも中央 UDR への負荷集中を緩和できる。

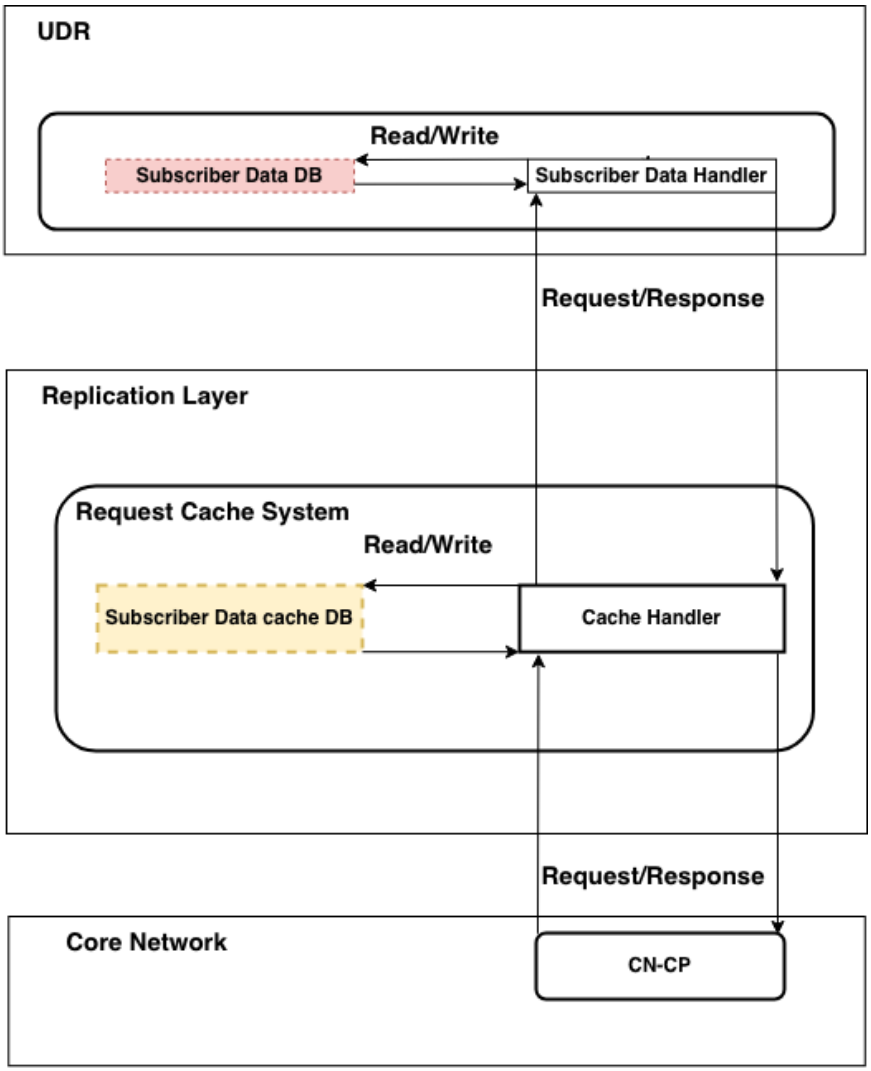

図 3.1: Replication Layer のシステム構成図

3.4.1 障害時の継続性と地理分散配置の効果

図 3.2 のように、UDR との間の通信あるいは UDR 自体に一時的な障害が発生したとしても、Replication Layer 内のキャッシュされたデータを利用することで、AD-RAN から加入者情報へのアクセスが可能となる。これにより、利用者端末のネットワークへの位置登録や回線確立などの手続きを提供でき、通信サービスの継続性を確保できる。また Replication Layer のノードを各地に分散して配置することで、AD-RAN がより近い場所に展開されたノードを利用することも可能となる。AD-RAN からより近い場所に Replication Layer を実装したノードを設置することで、中央に位置する UDR を利用するよりも応答遅延が小さくなる効果や、UDR への負荷集中を分散させる効果が期待できる。以上より、UDR や AD-RAN に対

して透過的に動作する Replication Layer は、加入者情報の可用性を向上させ、モバイルシステム全体の耐障害性の向上に寄与すると考えられる。

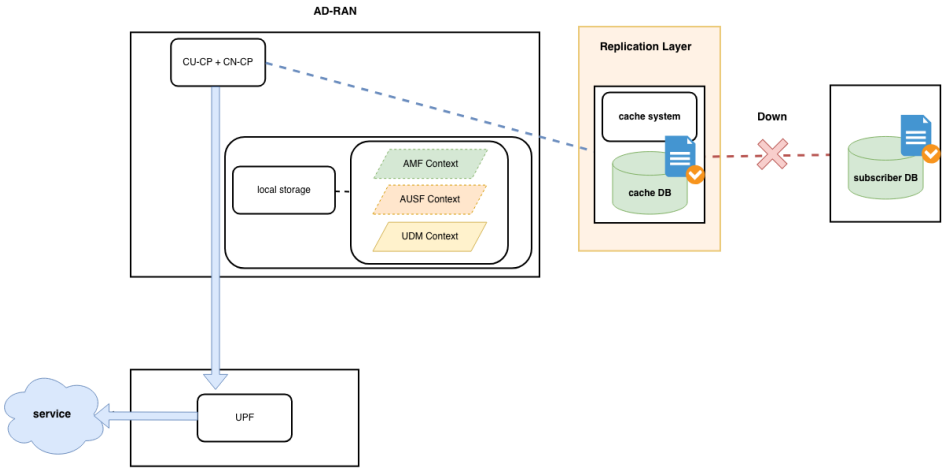

図 3.2: Replication Layer を使用したリンクダウン時の動作例

3.5 障害ケースの対応

障害パターン (d) は、UDR に接続できないことによって認証・認可が停止する事象である。本研究では Replication Layer を介したキャッシュ応答により、UDR 到達不能時でも加入者情報の参照を継続する。(d-1) TN 分断時には、ADM は近傍の Replication Layer にアクセスし、キャッシュされた加入者情報で認証・登録処理を継続する。これにより、中央 UDR との通信が遮断されても局所領域内のサービス継続性を確保できる。(d-2) UDR 障害時には、Replication Layer が最後に取得した加入者情報を利用して応答し、UDR 復旧までの間の手続きを支援する。キャッシュ未保持の加入者については取得不能となるため、キャッシュヒット率を高める運用が重要となる。(d-3) 輻輳時のタイムアウトに対しては、Replication Layer が即時に応答できる範囲を増やすことで再試行を抑制し、認証系 NF と UDR 周辺のシグナリング集中を緩和する。

3.6 まとめ

本章では、加入者情報に関する障害パターン (d) への対応を中心に、提案方式の設計と可用性向上手法を述べた。Replication Layer は、AD-RAN と UDR の間に挿入される中間層として、加入者情報要求の代理受信と応答を担う。UDR と AD-RAN 双方に対して透過に動作し、既存インタフェースを維持したまま可用性を向上させることを設計方針とする。Replication Layer はリクエスト受付、UDR 転送・応答、キャッシュ管理、監視・統計収集の各機能で構成され、ADM からの

要求はキャッシュ照合を経てヒット時は即時応答、ミス時は UDR 転送とキャッシュ格納を行う。このフローにより、同一加入者情報への再要求が集中する状況でも中央 UDR への負荷集中を緩和できる。障害パターン (d) に対しては、TN 分断時には近傍の Replication Layer にアクセスし、キャッシュされた加入者情報で認証・登録処理を継続する。UDR 障害時には、Replication Layer が最後に取得した加入者情報を利用して応答し、UDR 復旧までの間の手続きを支援する。

第4章　コンテキスト情報の可用性向上手法

コンテキスト管理に関する障害パターン（a–c）への対応を中心に、提案方式の設計と可用性向上手法を述べる。

4.1　概要

モバイルコアネットワークにおけるセッション状態とされるUEコンテキストは、従来、NFに集中的に保持される構成が一般的であった。しかし、大規模災害や広域障害、運用ミスなどにより一部のノードやリージョンが利用不能となった場合、コンテキスト参照が集中し、シグナリングストームや輻輳を引き起こす可能性がある。

Context Shareはこの課題に対し、UEコンテキストを透過的に分散KVSへ外部化することで、ネットワーク分断やノードダウンが発生した状況においても、モバイルシステムの基本的な通信サービスを継続可能とする。

本システムでは、コンテキスト管理基盤としてDistributed Hash Table（DHT）を採用する。DHTを用いることで、コンテキストをノード間に分散配置し、シグナリングの局所化と分断時の自立動作を実現する。

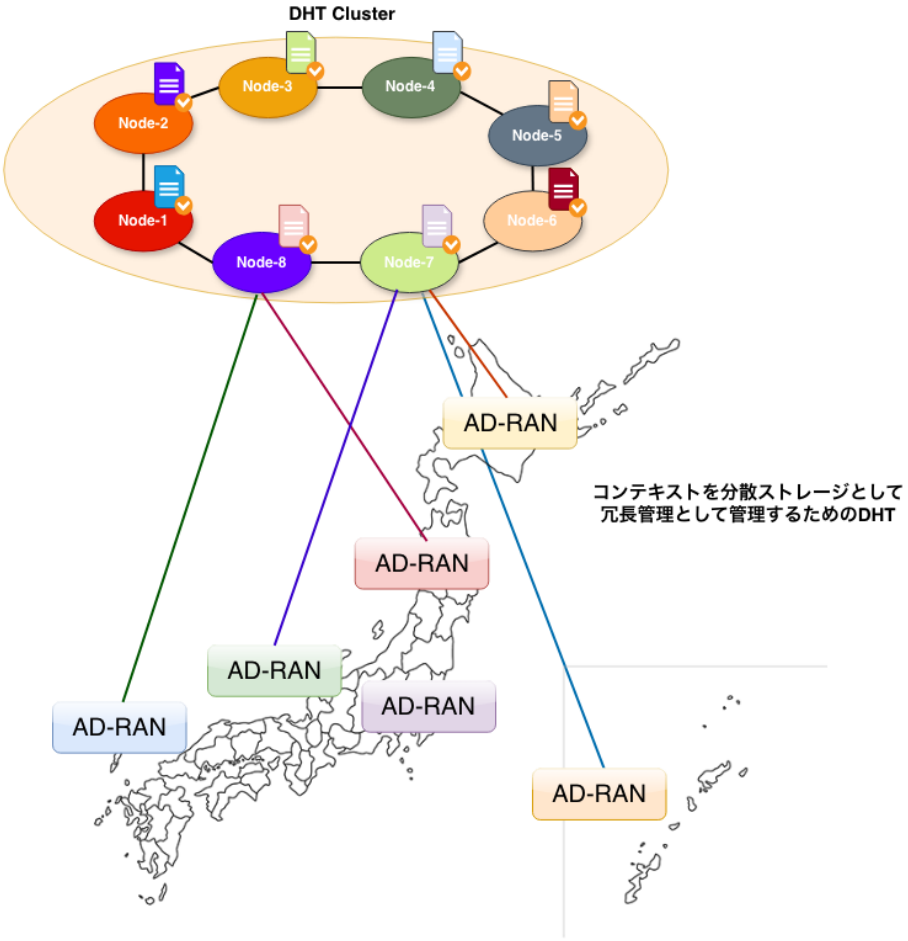

図 4.1: DHT を用いたコンテキスト管理の概念図

4.2 設計の考慮

Context Share の設計にあたり、以下の目的と要件を設定した。

- 地域的な孤立やノードダウンが発生しても通信サービスを継続可能であること
- 標準仕様に基づくシグナリングやアプリケーションの変更を伴わないこと
- ハンドオーバ中においても UE コンテキストを中断なく参照・更新できること

本研究では上記の設計要件を元に Context Share を設計する。DHT を用いることで、UE コンテキストを複数ノードに分散配置し、障害発生時にも局所的にコンテキスト参照が可能となる。また、DHT のルーティング機構により、ハンドオーバ中でも UE コンテキストの探索と更新が継続できる。

4.3 実装概要

4.3.1 Context Share

Context Share は、DHT を用いた UE コンテキストを分散 KVS として管理するミドルウェア層として実装される。既存のコンテキストキーに IMSI を付加し、これを入力としてハッシュ演算を行うことで、UE コンテキストの保存先ノードを一意に決定する。この方式により、特定の UE に対応するコンテキストは常に同一の論理位置に配置され、正しく参照・更新できることを確認した。

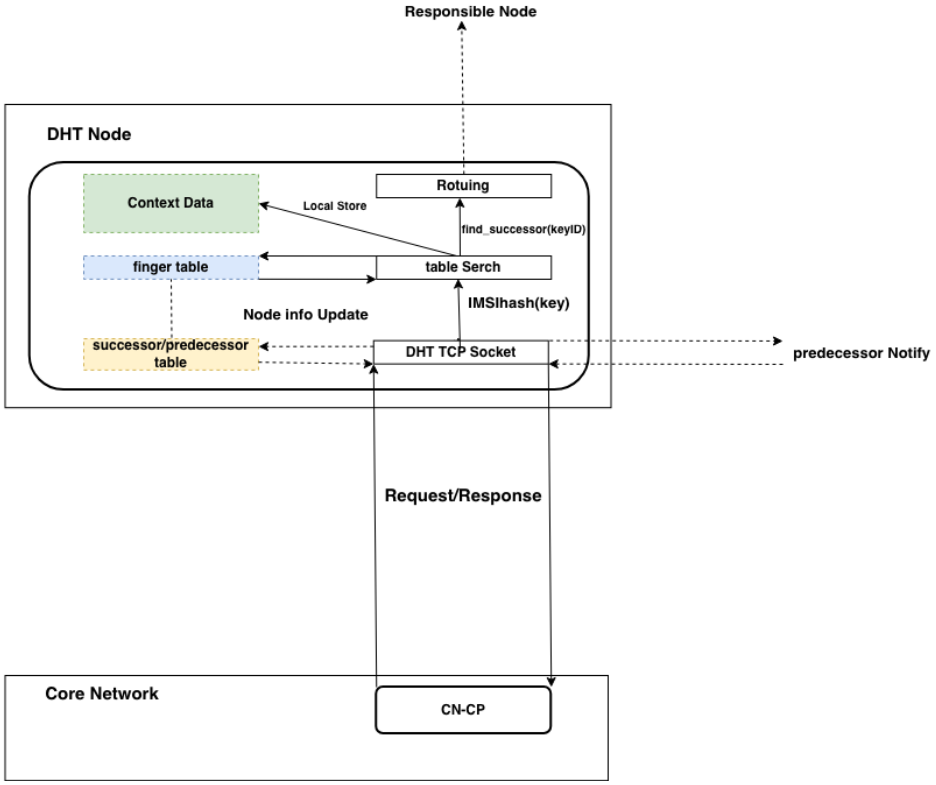

図 4.2: DHT を用いたコンテキスト管理のシステム構成図

コンテキストの要求が開始されると、Context Share は IMSI を用いて DHT 内の保存先ノードを特定する。DHT 内で使用されているアルゴリズムは Chord であり、ノード間のルーティングは Chord のルーティング機構に従って行われる。IMSIHash は、UE を一意に識別する IMSI を入力としてハッシュ演算を行い、Chord の識別子空間上のキー ID へ変換を行う関数である。これにより、UE ごとのコンテキストは DHT リング上の一意な論理位置に対応付けられる。Context Share は、算出されたキー ID に対して Chord の `find_successor(keyID)` を実行し、当該キー ID を担当するノード（Responsible Node）を探索する。

Chord の探索処理では、各ノードが保持する finger table を参照しながら次ホップを決定し、問い合わせを逐次転送する。探索が完了すると、担当ノードが保持するローカルストレージ（Context Data）から該当する UE コンテキストを取得し、要求元へ応答を返す。書き込み要求の場合も同様に担当ノードを決定した上で、担当ノードのローカルストレージにコンテキストを格納する。この仕組みにより、要求がどの DHT ノードに到達したとしても、最終的に同一の担当ノードへ到達し、一貫した参照・更新が可能となる。

また、Chord リングの維持のために、各ノードは successor/predecessor 情報を保持し、ノード参加・離脱に伴う更新を行う。ノード情報の更新は、近接ノード間の通知（predecessor notify）や定期的な更新処理を通じて実施される。これにより、ノード構成の変化が発生した場合でも、DHT の探索経路が維持され、担当ノードの探索が継続可能となる。

本実装では、Core Network C-Plane(CN-CP) からの要求を Context Share が受け付け、ノードが開放している TCP socket を介して DHT ノード群に要求を伝搬する。このとき CN-CP は保存先ノードを意識する必要がなく、IMSI およびコンテキストキーを与えるのみで保存・取得処理を実行できる。この透過性により、既存のモバイルコア処理に対して最小限の変更で分散コンテキスト管理を導入可能である。

4.3.2 Ho の際のコンテキスト管理

従来のアーキテクチャでは、各 AD-RAN が配下の全 UE コンテキストをローカルストレージ（例：Redis）で保持する「集中型管理モデル」を採用していた。この構成において拠点間ハンドオーバ（N2 ベースのハンドオーバを想定）が発生した場合、移動元 AD-RAN から移動先 AD-RAN に対して、UE コンテキストを明示的に転送する処理が必要となる。

従来方式では移動に伴う「状態の移送（Context Transfer）」が必須であり (図 4.3)、本研究で提案する Context Share は、このコンテキスト転送を不要とすることで、よりステートレスで迅速なハンドオーバを実現するものである。

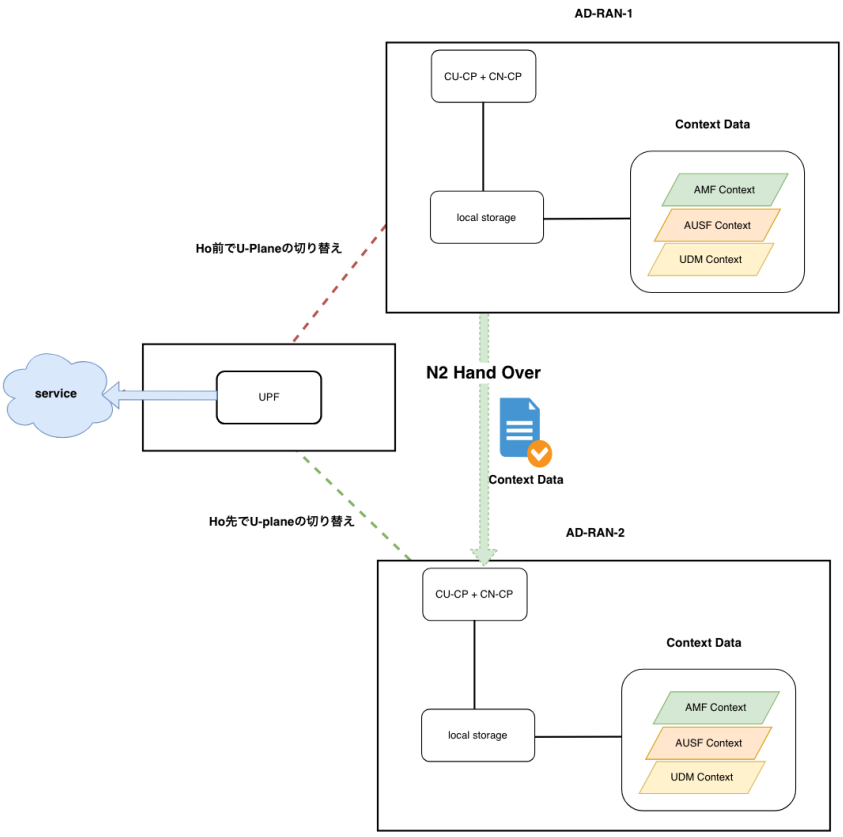

図 4.3: 従来の ADM における N2 Ho

一方、Context Share では、複数ノードが UE ごとのキー空間を分担し、各ノードが UE コンテキストを保持する水平分散型モデルを定義する。(図 4.4) libctx は AD-RAN 内でコンテキスト管理を行うライブラリとして機能しており、UE コンテキストの保存・取得をライブラリ内の関数を用いて DHT ノード群に対して透過的に実行する。DHT 内部のルーティング機構により、コンテキストがどのノードに存在するかを探索可能であり、ハンドオーバ時にコンテキストを明示的に転送する必要はない。これにより、移動時の耐障害性向上と柔軟なハンドオーバを実現と UE のステートレス化を両立する。

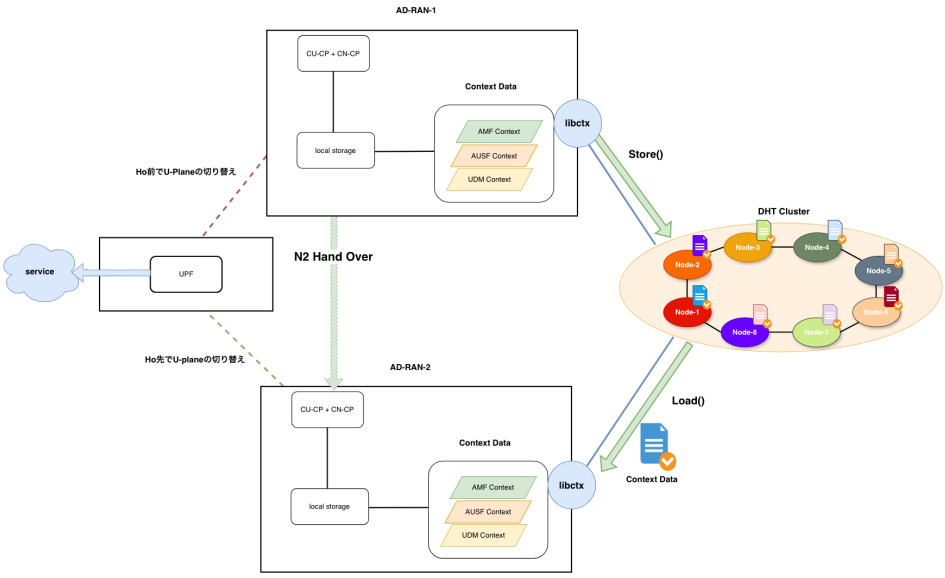

図 4.4: DHT を用いたコンテキスト管理のシステムにおける N2 Ho

4.3.3 局所性に対する考慮

単純に DHT を導入した場合、キーのハッシュ結果に基づいて UE コンテキストの担当ノードが決定されるため、物理的に遠隔なノードへコンテキストが割り当てられる可能性がある。本研究ではコンテキストのキーとして IMSI を利用し、`IMSIHash(IMSI)` によって DHT のキー ID 空間へ変換する方式を採用しているが、このときキーの分布は原理的に一様であり、地理的な配置とは独立に担当ノードが決定される。その結果、例えば沖縄地域のユーザのコンテキストが北海道地域の DHT ノードに割り当てられるケースが一定割合で発生し、コンテキスト参照・更新のたびに遠距離通信が発生する。これはハンドオーバ時の処理遅延増大や、分断時のコンテキスト参照不能といった問題を引き起こす可能性がある。

局所性を持つシステムとして、ハンドオーバ中でも UE コンテキストを中断なく参照・更新できることを示すことであり、コンテキストは分散 KVS（DHT）へ外部化しつつ、網側のシグナリング手順は標準に沿ったまま変更しない方針を取る。以上より、DHT を用いたコンテキスト管理方式には、少なくとも次の要件が求められる。

(a) **透過性**：標準に沿ったシグナリングに従い、かつコアネットワーク側アプリケーションの大幅な変更を伴わないこと。

(b) **局所性・低遅延**：ハンドオーバによって参照先が変化する状況を想定し、UE コンテキストを近接領域に配置することで低遅延に参照可能であること。

この局所性の課題に対しては、既存の DHT を使用したシステムに階層構造を導入し、地理的な近接性を反映したコンテキスト配置を行う方式が有効であると考

えられる。本研究では、地域単位で局所性を維持する方法として、階層化 DHT や複数 ID 空間を用いた設計を検討する。

4.3.4 局所性に特化した DHT の関連システム

Crescendo

局所性とスケーラビリティを両立する階層化 DHT に関する代表的な先行研究として、Canon in G Major[7] が挙げられる。この論文では、階層型 DHT を構成するための設計原理である Canon を提案し、その具体例として Chord を階層化した方式 Crescendo を示している。Crescendo は DHT に DNS のような階層構造を導入する研究であり、複数の下位リング（ドメイン）を上位層が連結することで、地理・組織の階層を反映した DHT を構築可能とする。(図 4.5) この方式では、各階層が独自のアルゴリズムで DHT リングを形成し、問い合わせは階層を横断してルーティングされる。また、アプリケーションが持つ名前（URL、識別子など）をハッシュしてキー ID とする点は従来の DHT と同様であり、Chord と同程度の探索計算量オーダを維持できることが特徴である。一方で、階層を統合するためのアルゴリズム設計が複雑であり、子リングをマージしたとしても各ノードのリンク数とルーティング効率を DHT 並みに保つ必要があるなど、実装面での難しさが考慮される。また、階層内での自ノードの位置関係を考慮してリングを構成する必要があり、運用上の制約も増える。

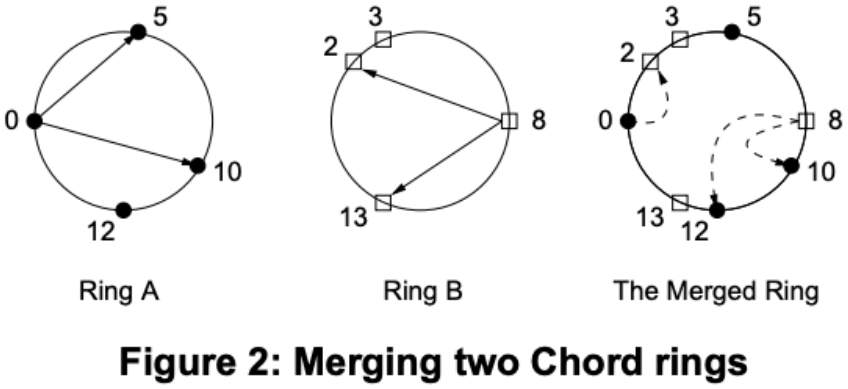

図 4.5: 文献 [7] Fig 2 より抜粋

MoHiD

移動体通信における局所性を意識した階層 DHT の応用として、MoHiD が挙げられる [8]。MoHiD は階層 DHT（HDHT）により端末の位置情報を管理するモビ

リティプラットフォームであり、問い合わせ処理では近い階層での解決を優先し、不足する場合のみ上位階層へ問い合わせをエスカレートする。(図 4.6) さらに、各エントリが保存レベル（どの階層に登録するか）を制御できるため、必要以上に上位階層へ依存せず遅延を抑制できる。このように、物理・運用ドメインごとに独立した DHT を構成し、それらを階層的に接続する設計は、管理や障害影響範囲を局所化しつつ、他領域からの探索を可能にする性質を持つ。また、値の本体は原則として所属クラスタに保存し、上位層にはポインタ（ロケータ）または縮約エントリのみを登録するポリシーを取りやすい点から、局所性と低遅延を要求するモバイルネットワークの要件 (b) と相性が良い。

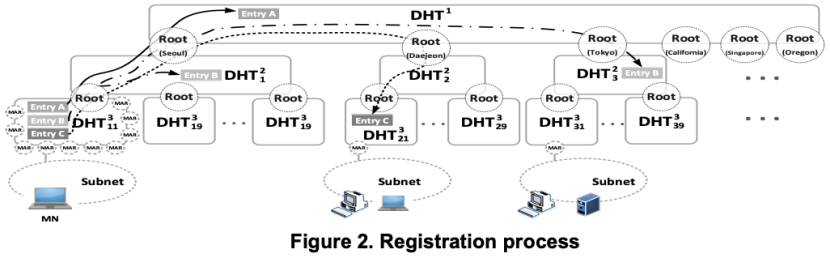

Figure 2. Registration process

図 4.6: 文献 [8] Fig 2 より抜粋

以上の先行研究を踏まえると、局所性を確保しつつ透過性 (a) を損なわない設計としては、従来の Chord や Crescendo のような単一 ID 空間を維持して全ノードへ値を一様分散する方式よりも、地域・運用ドメイン単位で独立した ID 空間を持つ階層型 DHT の方が有利である。単一 ID 空間型では値の配置がリング全体に分散されるため、ハンドオーバ時に近接領域へコンテキストを寄せたいという要件との整合が難しい。一方、複数 ID 空間型では物理・運用ドメインごとに独立した DHT を構成し、上位層にはポインタのみを保持することで、局所性と低遅延を満たしやすい。

4.3.5 Regional & Core Cluster 型 DHT の採用

局所性と検索可能性を両立するために、地域単位で独立した Regional Cluster を構成し、その上位に Core Cluster を配置する「Regional & Core Cluster 型 DHT」を採用する。(図 4.7)Regional Cluster では地域ごとに独立した Chord リングを構成し、UE コンテキスト本体を保存する。このとき Node ID は物理ノード ID から地域ごとに独立に生成し、地理情報そのものは Node ID へ直接埋め込まない。これにより、本体コンテキストが所属 Regional DHT から他地域の DHT へ再配置されることはなく、地域単位の局所性が保たれる。

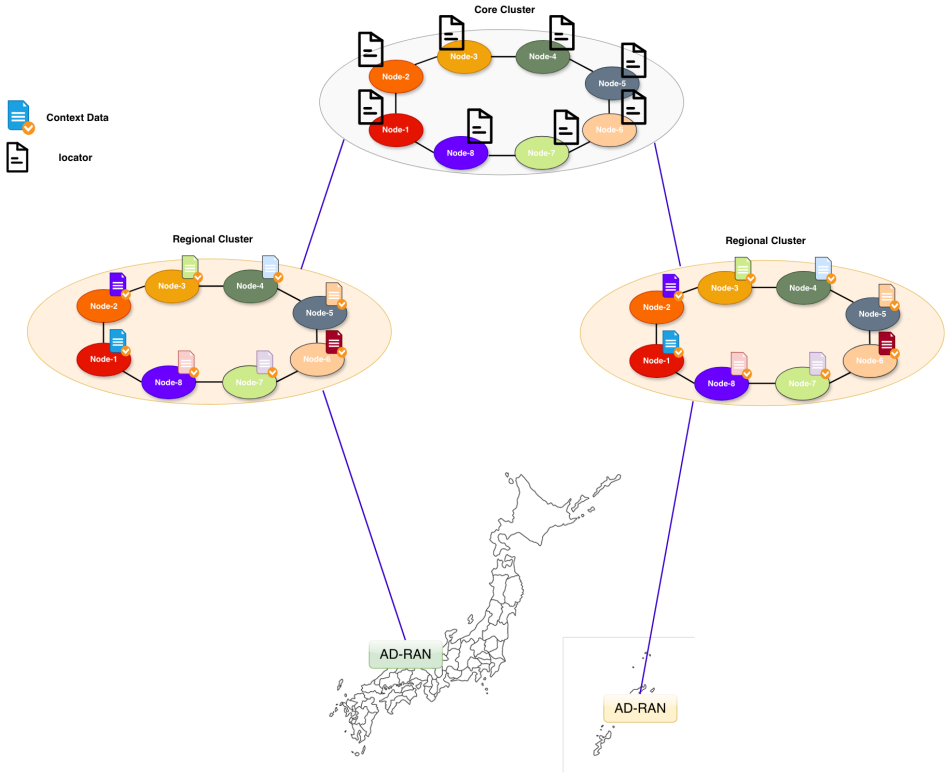

図 4.7: Regional & Core Cluster 型 DHT の概念図

4.3.6 Regional & Core Cluster 型 DHT の実装概要

図 4.8 のシステム構成図は Region Cluster としての側面を持ちつつ Core Cluster としても動く DHT ノードの仕様である。Core Cluster は各 Regional Cluster のスーパーノードのみが参加する別インスタンスの Chord リングであり、UE コンテキスト本体は保持せず、ロケータ情報 (該当コンテキストが存在する Regional Cluster のスーパーノード ID) を保存する。具体的には、Core Cluster に対して、次のようなロケータ情報を値として保持する。

```
{
  core_node_id:  // 対象 Regional Cluster のスーパーノード ID
}
```

このとき DHT ノードは Regional Node ID と Core Node ID の両方を持つ場合があり、Core 上で解決された `core_node_id` を宛先としてルーティングすることで、対象 Regional Cluster のスーパーノードへ到達可能となる。以降、対象スーパーノードは自リージョンの DHT を参照し、UE コンテキスト本体を取得する。すなわち、本方式は以下の二段階ルーティングにより他地域からの探索を実現する。

1. Core DHT に対して `Key` を問い合わせ、対象 Regional Cluster のスーパーノード（ロケータ）を解決する。

2. 解決したロケータ情報を用いて対象 Regional Cluster へ到達し、Regional DHT から UE コンテキスト本体を取得する。

この構成により、UE コンテキスト本体は常に所属地域内に配置され、近接領域での低遅延アクセスを実現できる。さらに、Core Cluster を介したロケータ解決により、他地域からの検索も可能である。以上より、本研究の要件 (a) 透過性および (b) 局所性・低遅延を両立する設計として、Regional & Core Cluster 型 DHT を採用する。

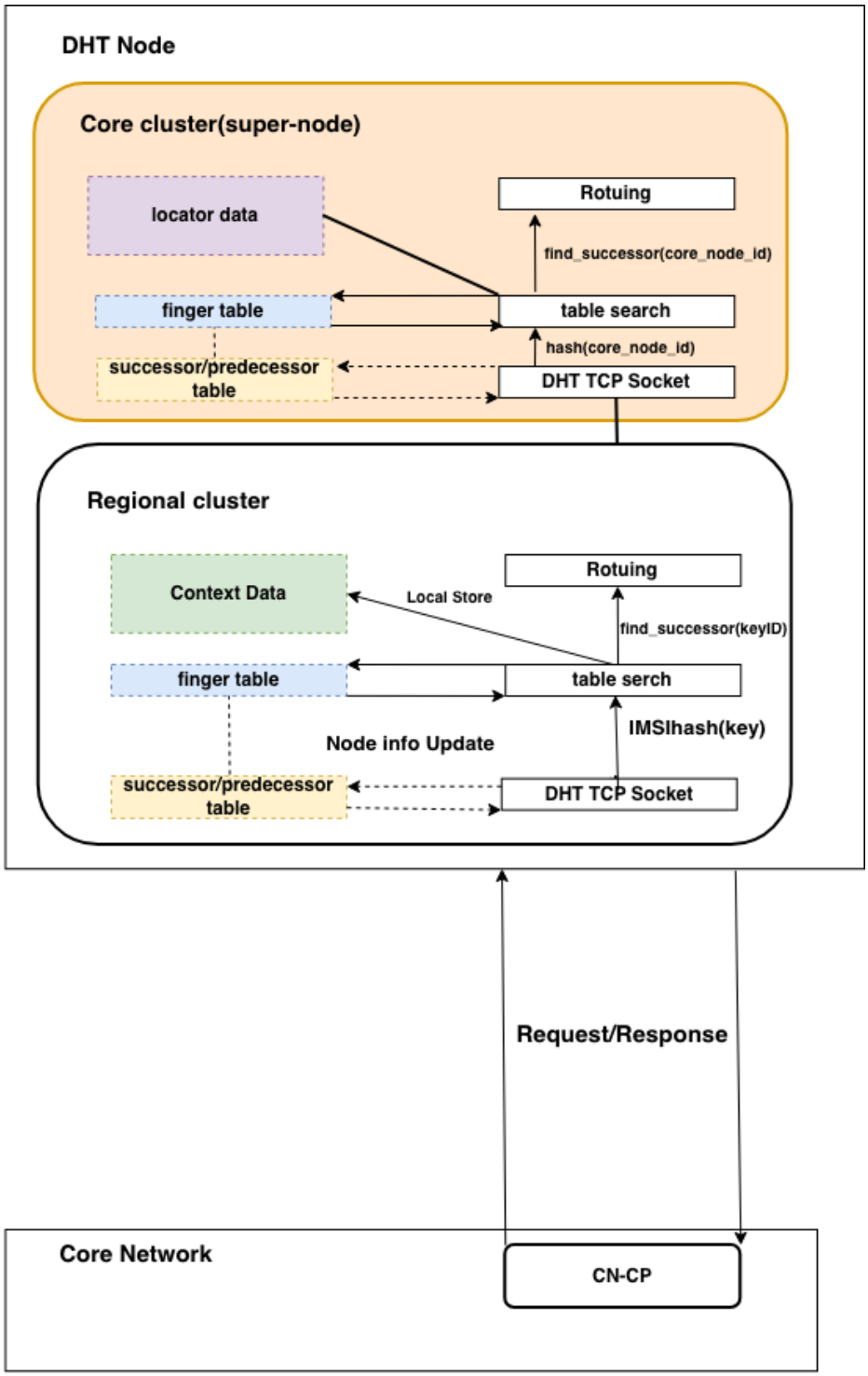

図 4.8: Regional & Core Cluster 型 DHT のシステム構成図

4.3.7 N2 Based Inter AD-RAN Handover における Context Share の動作

図 4.9 は、従来の N2 Handover を対象として、提案する Context Share を組み込んだ場合の処理シーケンスを示す。本シーケンスは、Region-A（Source）から Region-B（Target）への拠点間ハンドオーバを想定しており、制御プレーンは AD-RAN 単位で独立して動作する一方、ユーザプレーンは同一 UPF を継続利用する構成である。

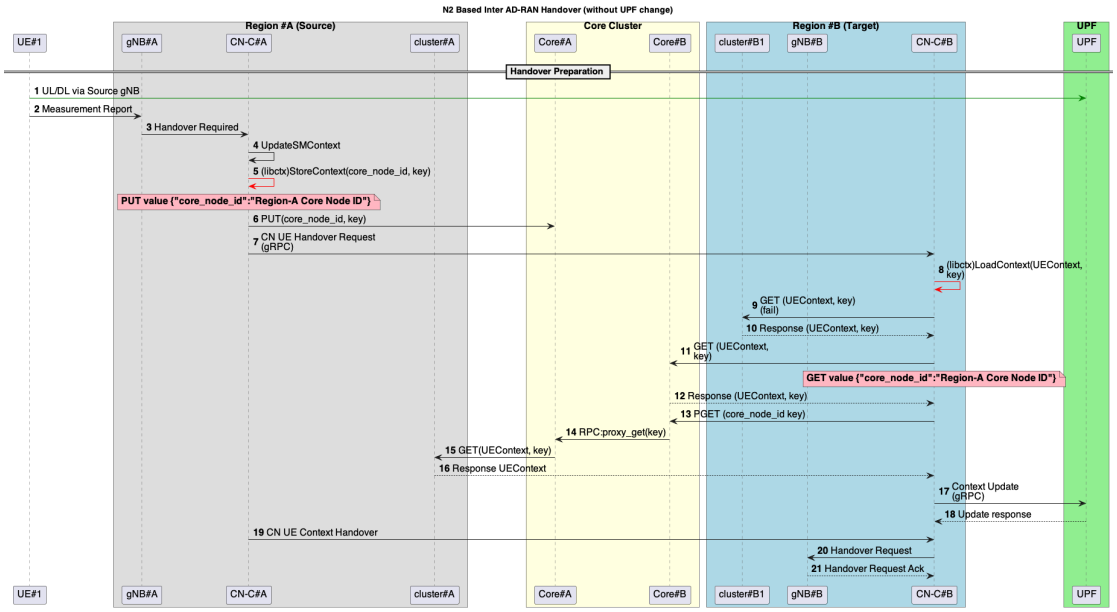

図 4.9: N2 Based Inter AD-RAN Handover における Context Share の動作シーケンス

4.3.8 Handover Preparation フェーズにおけるコンテキスト外部化

Handover Preparation の開始時点では、UE は Source gNB を介して Uplink/-Downlink 通信を継続している（図中 1,2）。その後、gNB が UE に関する測定レポートを送信し（3）、N2 を介したリロケーションの実行が決定される（4）。

ハンドオーバ準備に伴い、Source 側の CN-C（CN-C#A）は SMF に対して PDU Session に関する更新要求を送信する（5）。続いて Source 側の CN-C は、ハンドオーバに必要な UE 関連コンテキストを収集する（6）。

階層化方式では、UE を識別する IMSI に基づくキーを用い、Regional Cluster に対してコンテキスト本体の Store 処理を実行する（7）。さらに、異なる Regional Cluster からの探索を可能とするため、Core Cluster に対して `key=IMSI` を入力とし

たロケータ情報の登録（value={core_node_id}）を行う（8）。この Core Cluster 登録は、Target 側がどの地域クラスタに UE コンテキストが存在するかを解決するための索引として機能する。

その後、Source 側 CN-C は標準の N2 Handover 手続きに従い、UE コンテキストハンドオーバに必要な情報を Target 側へ転送する。本研究では、網側シグナリングの手順を変更せず、Context Share をあくまで外部化ストレージとして動作させる点を重視している。

Target 側におけるコンテキスト復元処理

Target 側（Region-B）では、Handover に必要な UE 状態を復元するため、Context Share による Load 処理を実行する。このとき、Target 側は直接 UE コンテキスト本体を探索するのではなく、まず Core Cluster を参照して Key=IMSI に対応するロケータ情報（core_node_id）を取得する。取得したロケータ情報を用いて、対象 Regional Cluster へ到達し、Regional Cluster 内に保存された UE コンテキスト本体を取得する。

この二段階ルーティングにより、UE コンテキストは所属地域に局所配置されたまま維持され、他地域からは Core Cluster を介して探索可能となる。これにより、単一 DHT におけるハッシュ分散に起因する地理的非局所性（例：遠隔地域へのコンテキスト配置）を抑制しつつ、拠点間ハンドオーバで必要となる参照性を確保できる。

UE コンテキストの取得後、Target 側 CN-C（CN-C#B）は当該 UE に対応する状態を復元し、U-Plane セッションに関連する制御情報（例えば GTP トンネル終端情報）を更新する。本シーケンスは UPF 変更を伴わないため、PDU Session Anchor は維持されるが、Target gNB に向けた経路更新が必要となる。これに伴い SMF は UPF に対して PFCP Session Modification を行い、Target 側 gNB に対する GTP トンネル情報を更新する。この結果、ユーザプレーンはハンドオーバ後に Target gNB 経由で継続される。

Handover Execution フェーズと通信確立

図 4.10Handover Execution では、Source 側から UE へ Handover Command が送信され、UE はターゲットセルへ同期した後に Handover Confirm を返す。この間、Downlink Data の転送やデータフォワーディングが行われる。その後、PathSwitch 処理が開始され、Target 側で Uplink Data が利用可能となり、ハンドオーバが完了する。

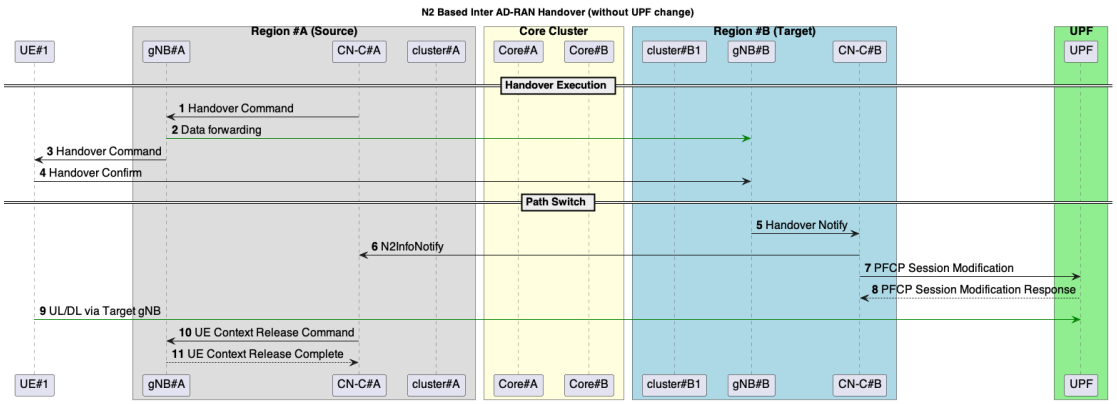

図 4.10: コンテキスト復元後の Handover Execution

完了後、Target 側 CN-C は N2 の通知手続きを実行し、End marker 送信を経て、Target gNB 経由で Downlink Data が再開される。最終的に Source 側は UE Context Release 手続きを実行し、Source 側に残る UE コンテキストを解放する。

標準手順の維持と外部化の役割分担

本シーケンスより、Context Share は標準の N2 Based Handover 手順を変更することなく、UE コンテキストを分散 KVS へ外部化し、Target 側で復元する機構として機能することが分かる。特に、Core Cluster を索引として用いることで、UE コンテキスト本体を Regional Cluster に局所保存したまま、拠点間での探索・取得を実現している。これにより、ハンドオーバにおける「参照先の変更」と「低遅延での状態復元」を両立し、ユーザプレーン通信を継続可能にする設計である。

4.4 障害パターン（a–c）への対応

最後に、提案方式である Context Share が想定する代表的な障害パターン（a）輻輳・ストーム、（b）分断、（c）NF/RAN ダウン、に対して、どのように影響を緩和し、手続き継続性を高めるかを整理する。特に本研究では、UE コンテキストを DHT に外部化し、Regional Cluster 単位での自律動作を可能とする点が特徴である。以下では、各障害において問題となる要因と、Context Share による抑制効果を説明する。

4.4.1 輻輳・ストーム時（a）

輻輳やシグナリングストームが発生した場合、集中型ストレージではアクセスが特定ノードに集中する。その結果、ストレージ層がボトルネックとなり、認証・

登録・セッション確立といった複数の手続きが同時に遅延、あるいはタイムアウトすることで、障害の波及が生じやすい。また、処理遅延によって NF 側の再試行が増えると、追加シグナリングが発生し、さらなる輻輳を引き起こす「再試行ループ」が生じる可能性がある。

Context Share では、UE コンテキストが複数ノードに分散配置されるため、特定ノードへの負荷集中が緩和される。さらに、手続き処理の対象となる UE 群が DHT のキー空間上で分散されることで、アクセスは自然に水平分散される。これにより、輻輳時であっても処理性能低下が局所化され、手続き全体の成功率を維持できる可能性が高まる。加えて、障害発生時の影響範囲が局所に限定されるため、障害が収束しやすく、復旧フェーズでの負荷集中を抑制できる。

4.4.2 分断時（b）

地域間のネットワーク分断が発生した場合、集中型アーキテクチャでは制御プレーンへの参照が成立せず、結果として当該地域全体で手続きが停止する恐れがある。

Context Share では、UE コンテキスト本体が Regional Cluster 内の DHT に保存されているため、地域間分断が発生した状況でも、当該地域内での通信サービスを継続可能である。すなわち、Core Cluster へのアクセスが不能であっても、Regional Cluster 内で完結する形で必要なコンテキスト参照が成立し、既存 UE の通信継続や、限定的な手続き処理が可能となる。この構成により、「Core Cluster に依存しない自立動作」という性質を実現し、分断時のサービス提供範囲を維持できる。また、分断が解消された後には、Regional Cluster 側で継続的に更新された状態を Core 側へ再同期することで、システム全体として整合性を回復できる。

4.4.3 NF/RAN ダウン時（c）

特定の AD-RAN がダウンした場合、メモリ上に保持されていた UE コンテキストが消失し、再接続時に再生成が必要となる。標準構成では、復旧直後に多数の UE が一斉に再登録や再確立を行うことで、制御プレーンが瞬間的に過負荷となり、復旧遅延が生じる可能性がある。この問題は「復旧ストーム」として現れ、障害そのものが解消してもサービスが回復しにくい要因となる。

Context Share では、UE コンテキストが DHT 上に保持されているため、再起動後に必要な状態を外部から取得し、手続きを継続できる。加えて、復旧フェーズにおける再同期処理を DHT ノード群へ分散して実行できるため、特定ノードに負荷が集中しにくい。これにより、復旧直後の一時的な輻輳を抑制し、迅速なサービス再開を可能とする。また、NF 側の状態が失われた場合でも UE のコンテキストが残存するため、UE 側から見た再確立時間を短縮でき、結果として通信断時間を抑えられる可能性がある。

第5章　評価

5.1　評価方針

本章では、提案手法である ADMobile（Local Storage/Context Share）の有効性を、標準 5GC 構成と比較することで定量的に評価する。提案手法は「モバイルシステムの通信継続性を高めること」を目的としており、その効果は平常時の性能だけでなく、障害や分断が発生した状況で評価する必要がある。一方で、提案方式はコンテキストの外部化や問い合わせを伴うため、標準構成と比較して手続き時間やシグナリングが増加する可能性がある。そこで本章では、(i) 平常時の処理オーバーヘッド、(ii) 分断・障害下での継続性、(iii) DHT 規模が与える理論的影響、の 3 つの観点から評価を行う。

5.2　評価環境

Ubuntu 24.04.1 LTS Intel(R) Xeon(R) Gold 6338 CPU @ 2.00GHz Memory 16GB

5.3　評価方法

5.3.1　評価指標

評価指標として、手続き完了時間を用いる。これは、各手続き開始から終了までに要した時間を計測し、標準 5GC 構成と比較することで、提案方式が導入する追加処理の影響を明確化するためである。また、障害下の通信継続性を確認するため、U-Plane のスループット（iperf3 によるスループット推移）を補助指標として用いる。

5.3.2　評価シナリオ

評価は、目的が異なる 3 種類の実験から構成する。

(i) ベースライン評価（平常時オーバーヘッド）　提案方式が導入する追加処理の影響を確認するため、平常時における主要プロシージャの処理時間を比較する。対象手続きは、A. Registration、B. PDU Session Establishment、C. ハンドオーバ（県内）、D. ハンドオーバ（県間）とする。

(ii) 分断環境における通信継続性の評価　提案方式の本質的な狙いである「分断や障害があっても通信継続できること」を示すため、県内／県間のハンドオーバを実行する。評価指標として U-Plane の通信が途切れないことをスループット推移により確認する。

(iii) 規模の影響（理論評価）　提案方式（Context Share）において、DHT ノード数が負荷分散および探索経路長に与える影響を理論モデルとして評価する。これは、実験環境の規模制約では評価しにくい大規模性を扱うための補助評価である。

5.4　実験環境

標準 5GC 環境におけるベースライン評価および分断環境における通信継続性の評価では、図 5.1 に示す実験環境を用いる。コンポーネントの構成としては、UE/gNB エミュレーターとして PacketRusher[10] を、5GC コアネットワークとして Free5GC[9] を使用する。PacketRusher[10] は UE/gNB の機能を統合的に提供するエミュレーターであり、Free5GC はオープンソースの 5GC 実装である。想定としては、N2 Based Handover を対象とし、UPF の切替を伴わない構成とする。

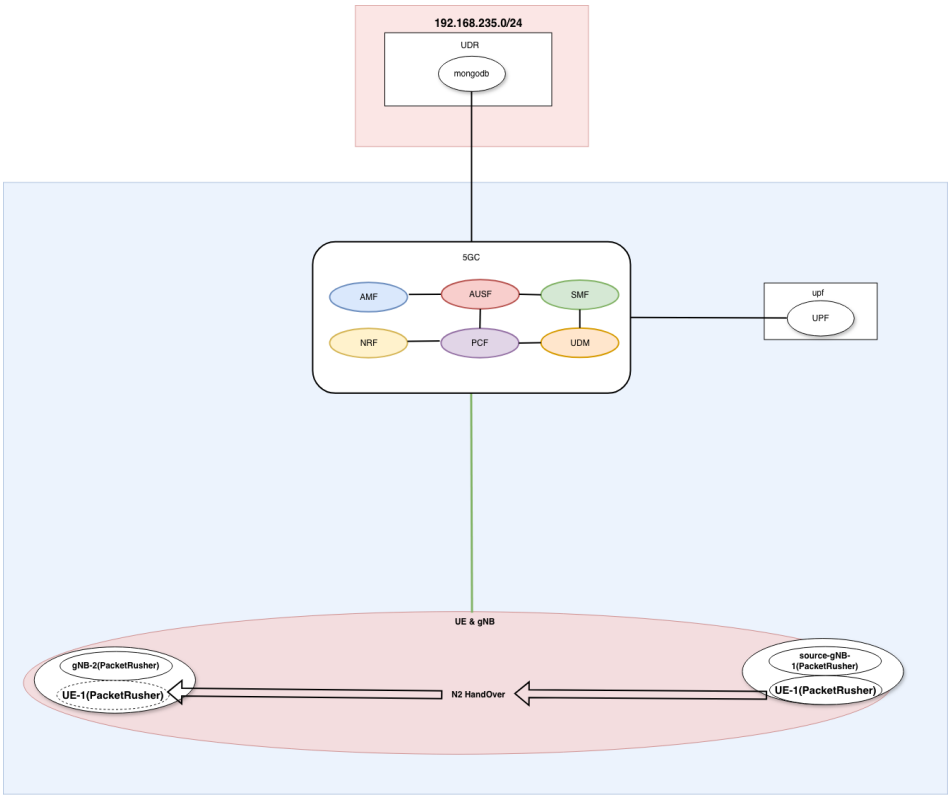

図 5.1: 標準 5GC における評価環境の構成図

次に ADM 環境では、Local Storage 構成として図 5.2 に示す実験環境を用いる。標準 5GC と同じく、UE/gNB エミュレーターとして PacketRusher を、コアネットワーク部分を Proc5GC に置き換えた構成である。この現環境が ADMobile の基本構成となる。その上で ADMobile における Handover である N2 Based Inter AD-RAN Handover を評価する。

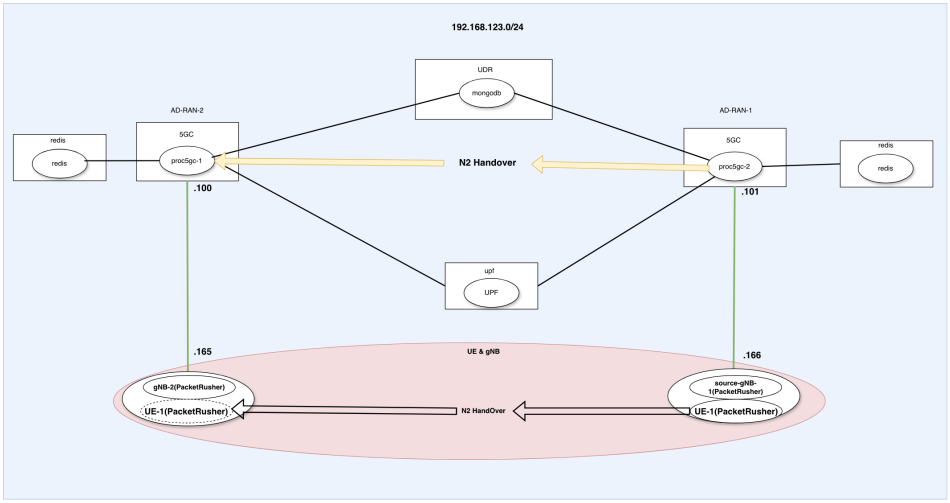

図 5.2: ADM における評価環境の構成図

Context Share の構成として図 5.4 に示す DHT の機能を実装したノード 1 台の構成を用いる。UE/gNB エミュレーターとして PacketRusher を、コアネットワーク部分を ADMobile に置き換えた構成である。これは KVS ストアとしての DHT ノードの性能を評価する上で 1 台としている。N2 Based Inter AD-RAN Handover を評価する。

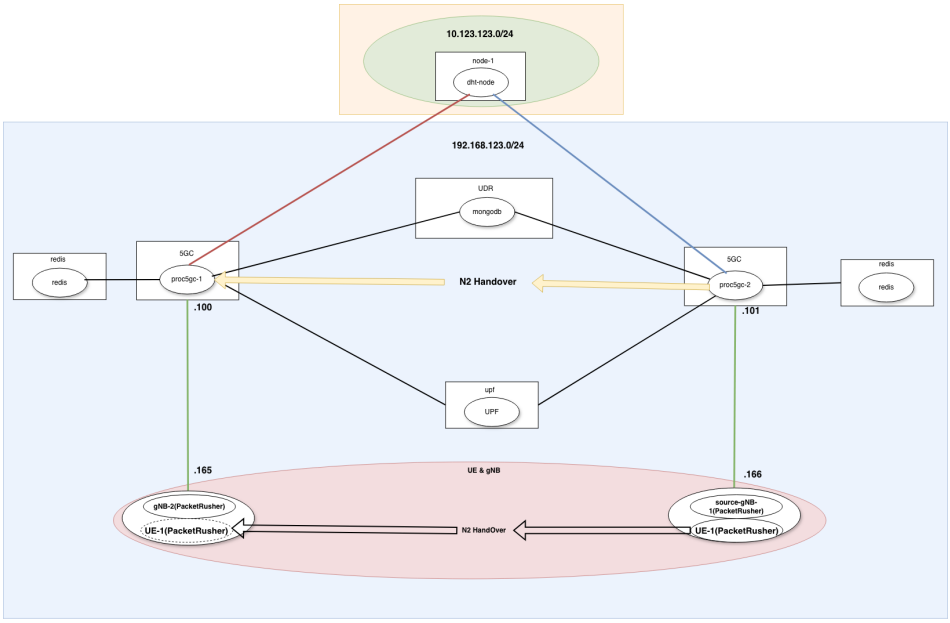

図 5.3: ADM + Regional Cluster(ノード一台) における評価環境の構成図

同じく Context Share の構成として図 5.4 に示す DHT の機能を実装したノード 4 台の構成を用いる。Regional Cluster として想定した上で N2 Based Inter AD-RAN Handover を評価する。

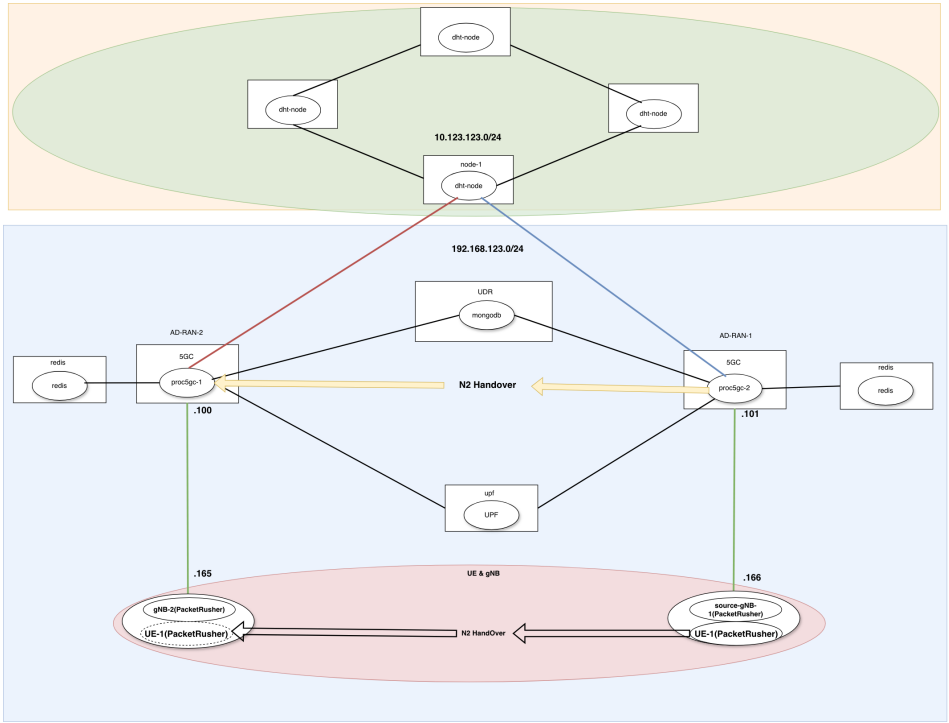

図 5.4: ADM + Regional Cluster(ノード 4 台) における評価環境の構成図

本格的な実運用を考えた構成として図 5.5 に示す Regional Cluster 想定のノードと Regional Cluster と Core Cluster として機能するスーパーノードの計 8 台の構成を用いる。DHT の Node ID 空間としては、RegionA と RegionB で独立した ID 空間を持ち、Core Cluster は両地域のスーパーノードで構成される。県間 HandOver を想定した上で N2 Based Inter AD-RAN Handover を評価する。

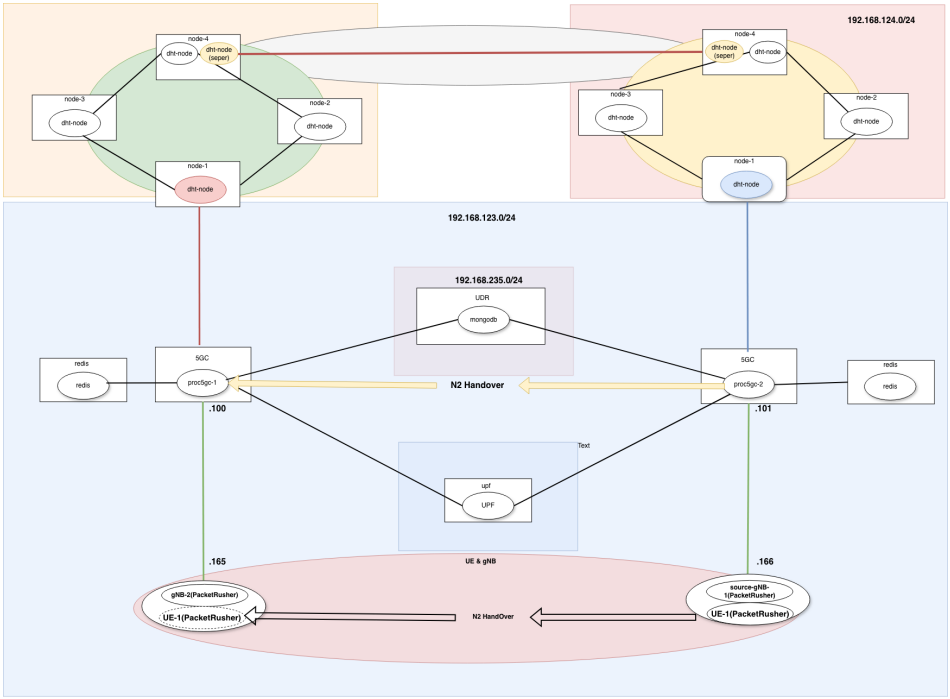

図 5.5: ADM + Regional Cluster + Core Cluster 評価環境の構成図

5.4.1 評価アーキテクチャ

ベースライン評価、また分断環境における通信継続性の評価では、以下の 3 構成を比較する。

- **標準 5GC:** 加入者情報および各種状態は NF を中心に管理される。
- **ADMobile(local-storage):** コンテキストをローカルストレージに外部化し、同一 AD-RAN 内で取得可能とする。
- **(提案手法)ADMobile（Context Share）:** コンテキストを DHT に外部化し、分断環境でも担当ノード経由で取得可能とする。

5.5 ベースライン評価：平常時オーバーヘッド

5.5.1 目的

本評価の目的は、提案方式の導入が主要手続きに与える処理オーバーヘッドを明確化することである。手続き時間および制御信号量の増減を測定し、実運用上許容可能な範囲であるかを確認する。

5.5.2　手法

ベースラインの評価を行うにあたっては評価アーキテクチャを指標として比較する。またシグナリングとして、Registration、PDU Session 確立、ハンドオーバ（県内）、特殊なケースとしてのハンドオーバ（県間）をそれぞれ実行し、手続き完了時間を計測する。

5.5.3　ベースラインの評価結果

本節では、各手続きにおける処理時間の計測結果を示し、提案構成の差分を比較する。また、各バー上部の数字は評価ケース番号を表し、ケース A が標準 5GC、ケース B が ADMobile（Local Storage）、ケース C が ADMobile（DHT、N=1）、ケース D が ADMobile（Context Share、N=4）である。

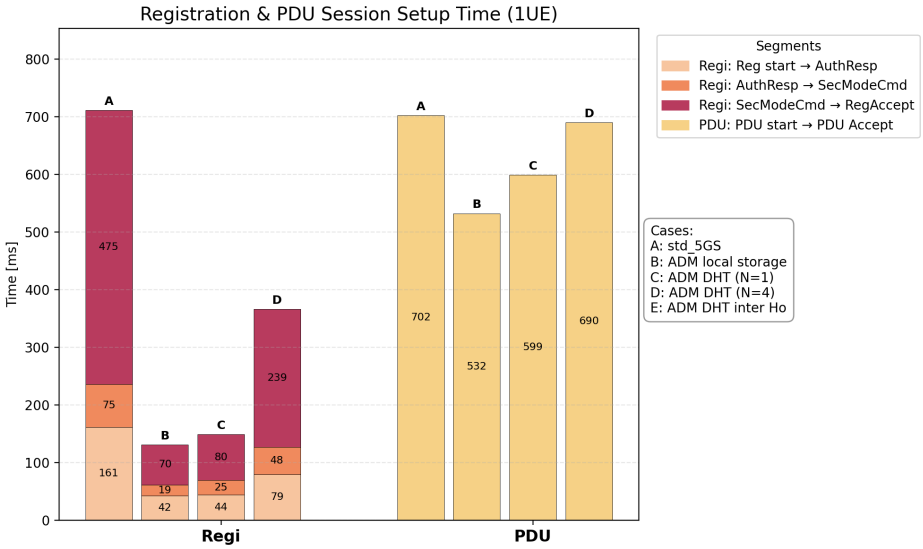

図 5.6: Registration/PDU Session 確立に伴うベースライン計測の結果

Registration の結果

図 5.6 は、Registration（Regi）、PDU セッション確立（PDU）複数の構成における手続き処理時間を積み上げで示したものである。各バーは手続きの一連の処理を複数区間に分割し、各区間の所要時間を積み上げた値として表している。Registration 手続きに関して、標準構成（ケース A）は全体で約 710 ms 程度となり、他の構成と比較して最も大きい値を示した。内訳を見ると、「SecModeCmd 送信から RegAccept 受信まで」の区間が約 475 ms と支配的であり、Registration 全体の遅延の主要因となっている。全体的に標準構成においては NF 間でのコンテキスト管理が集中型

に行われるため、HTTP/2 通信や内部処理の待ち時間が発生しやすいことが影響していると考えられる。この区間のオーバヘッドについては全体的に大きく、全体的に処理時間が増加している。

一方、ADMobile 構成では、Local Storage（ケース B）および Context Share を用いた構成（ケース C、D）において、Registration の処理時間が大きく短縮され、全体として約 130〜370 ms の範囲に収まっている。これは、提案構成により UE コンテキストの参照が局所化され、集中型処理による待ち時間が抑制されたことを示唆している。

また、DHT ノード数の影響に着目すると、DHT（N=1、ケース C）と DHT（N=4、ケース D）の間で、Registration 手続き時間の大幅な増加は観測されなかった。この結果から、DHT の導入が Registration 手続きの遅延を決定的に悪化させないことが分かる。ただし、ケース D では一部区間においてケース C よりも処理時間が増加しており、DHT ノード数の増加に伴う探索経路長や制御メッセージ数の増加が、一部の処理区間に影響を与えている可能性がある。

また、DHT のノード数の違いに着目すると、DHT（N=1, ケース C）と DHT（N=4, ケース D）の間で Registration の大幅な増加は観測されず、DHT の導入が Registration の手続き時間を決定的に悪化させないことが分かる。

PDU セッション確立（PDU）の結果

PDU セッション確立手続きに関しては、標準構成（ケース A）が約 702 ms と最も大きな値を示した。これに対し、ADMobile（Local Storage、ケース B）では約 532 ms に短縮され、DHT を用いた構成（ケース C、D）においても、それぞれ約 599 ms および 690 ms 程度であった。

この結果から、PDU セッション確立においても、提案構成は標準構成と比較して処理時間を短縮できることが確認できる。一方で、DHT（N=1、ケース C）と DHT（N=4、ケース D）を比較すると、N=4 の方が処理時間がやや増加している。

ただし、いずれの DHT 構成においても、標準構成と比較した場合には依然として短い時間で手続きが完了しており、提案構成の利点は維持されているといえる。

ハンドオーバ（Ho）の結果

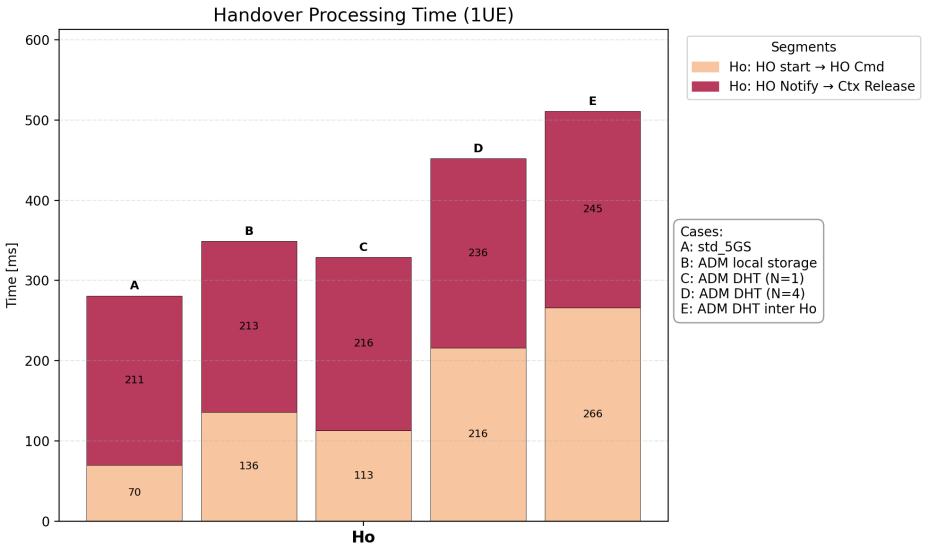

図 5.7: Handover に伴うベースライン計測の結果

図 5.7 は、Handover（Ho）に対して、複数の構成における手続き処理時間を積み上げで示したものである。ハンドオーバ全体の処理時間は、ケース A からケース D においておよそ 320〜450 ms の範囲で推移している。

ケース D およびケース E では、前半区間である HO start から HO Command 受信までの処理時間が増加している。特にケース E では、県間ハンドオーバを想定して Core Cluster を介した UE コンテキスト取得および転送が発生するため、全体の処理時間が約 510 ms 程度まで増加している。この増加は、Inter-Region 通信に伴う遅延や Core cluster への制御処理の増加が影響していると考えられる。

ただし、ケース E においてもハンドオーバ手続きは安定して完了しており、Context Share が県間ハンドオーバのような条件下においても実用的な処理時間を維持できることが確認できた。

5.6 分断環境における通信継続性の評価：Ho による通信継続性

5.6.1 目的

提案方式の本質的な狙いである「分断や障害があっても通信継続できること」を示すため、県内、県間のハンドオーバを実行する。評価指標として手続きの成功率に加えて、U-Plane の通信が途切れないことをスループット推移により確認する。

5.6.2　手法

本節では、分断や障害が発生した状況下においても UE の通信が継続可能であることを、Ho を使用した U-Plane スループット推移により確認する。図 5.8 は、iperf3 により計測したビットレートの時系列変化を示したものであり、比較対象として標準 5GC（std_5GC）に加えて、提案方式である ADMobile の 3 構成（Local Storage、Regional Cluster、Regional + Core Cluster）を含めている。

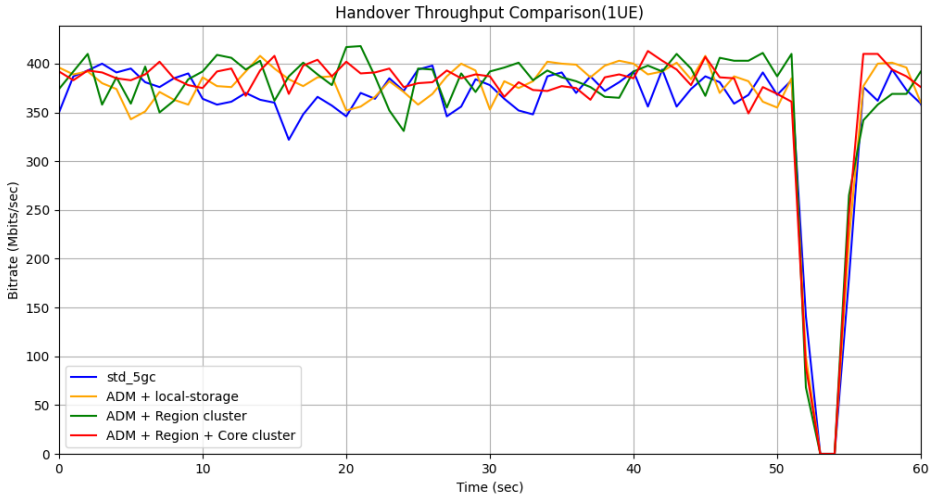

図 5.8: iperf3 によるスループット推移の比較

Ho の発生

分断環境における継続性の観点では、ビットレートが低下している区間が通信断を示す。図では、複数構成において 50 – 55 秒付近で一時的に 0Mbit/sec 近傍へ低下する区間が確認できる。これは、ハンドオーバに伴う経路切替の影響を反映している。

しかし、その後の区間でスループットは再度回復しており、各構成は 55 – 60 秒付近でビットレートが高くなっている。特に、ADM + Regional cluster および ADM + Region + Core cluster は回復後に平常時まで戻っており、通信が再確立されていることが確認できる。この結果は、障害や分断が発生しても、コンテキストを外部化しておくことで通信再開が可能であることを示唆している。

5.7 規模の影響評価：理論モデル

5.7.1 目的

提案方式（Context Share）では、DHT を用いて UE コンテキストを分散管理する。このとき、DHT ノード数の増加は 1 ノードあたりの担当範囲を縮小させるため負荷分散に寄与する一方で、探索経路長や転送回数が増加し、転送遅延が増える可能性がある。そこで、DHT 規模が与える影響を理論的に整理し、負荷分散と転送コストのトレードオフを明確にする。

5.7.2 前提とモデル化

本節では、Context Share を階層型 DHT としてモデル化し、DHT 規模が応答遅延に与える影響を理論的に整理する。システム全体は複数の Regional Cluster から構成され、各 Regional Cluster を代表する Super Node によって Core Cluster が形成されるものとする。

ここで、モデル化に用いるパラメータを以下に定義する。

- N: システム全体の DHT ノード数（Super Node は除く）
- C: Regional Cluster の数
- N/C: 1 つの Regional Cluster に含まれる DHT ノード数
- d: Region 内ノード間の伝搬遅延（Intra-Region delay）
- D: Region 間ノード間の伝搬遅延（Inter-Region delay）
- p: 1 hop あたりの処理時間（転送・ルックアップ処理を含む）

Chord 型 DHT では、キー探索に要する平均 hop 数はノード数 n に対して $O(\log n)$ に比例する。したがって、Regional Cluster 内の探索 hop 数を

$$h_1 = \log\left(\frac{N}{C}\right) \tag{5.1}$$

Core Cluster 内の探索 hop 数を

$$h_2 = \log(C) \tag{5.2}$$

と表す。

5.7.3 Intra-Region における応答遅延 T_1

同一 Region 内で UE コンテキストを取得する場合、探索は Regional Cluster 内で完結する。このとき、探索に必要な hop 数は h_1 である。

探索はリクエストとレスポンスの往復を伴うため、1 hop あたりの遅延を $(d+p)$ とすると、往復探索の総遅延は $2h_1(d+p)$ となる。さらに、最終的なコンテキスト取得に要する処理時間を p として加えることで、Intra-Region の応答遅延 T_1 は次式で表される。

$$T_1 = 2h_1(d+p) + p \tag{5.3}$$

$$T_1 = 2\log\left(\frac{N}{C}\right)(d+p) + p \tag{5.4}$$

この式より、クラスタ数 C を増やす（すなわち 1 クラスタ当たりのノード数 N/C を減らす）ことで $h_1 = \log(N/C)$ が減少し、Intra-Region の探索遅延が短縮されることが分かる。

5.7.4 Inter-Region における応答遅延 T_2

次に、異なる Region に存在する UE コンテキストを取得する場合を考える。Inter-Region では、Regional Cluster 内探索に加えて、Core Cluster を介した転送が必要となる。

Inter-Region の応答遅延は、大まかに以下の処理から構成される。

1. 送信元 Region 内で担当ノードを探索（Regional Cluster 内探索）
2. 送信元 Region の Super Node へ転送
3. Core Cluster 内で目的 Region の Super Node を探索
4. 目的 Region の Regional Cluster 内で担当ノードを探索
5. 応答を送信元へ返送

Region 内での 1 hop は $(d+p)$、Region 間での 1 hop は $(D+p)$ のコストを持つと仮定する。このとき Inter-Region の応答遅延 T_2 は、Regional Cluster 内探索と Core Cluster 内探索を合算して表現できる。

本モデルでは、Inter-Region の取得処理を整理すると次式で表される。

$$T_2 = 5p + 8(d+p)\log\left(\frac{N}{C}\right) + 2(D+p)\log(C) + 2D \tag{5.5}$$

ここで、$8(d+p)\log(N/C)$ は Regional Cluster 内の探索・転送に起因する遅延成分であり、$2(D+p)\log(C)$ は Core Cluster 内探索に起因する遅延成分である。また、$2D$ は Region 間の物理距離による往復遅延を表す。

5.7.5　計算例

図 5.9 は、Regional Cluster 数 C を変化させたときの応答遅延を示したものである．ここでは、Intra-Region の応答遅延 T_1 と Inter-Region の応答遅延 T_2 を比較する．パラメータは，$N = 1000$、$p = 1$ms、$d = 5$ms、$D = 10$ms とした．

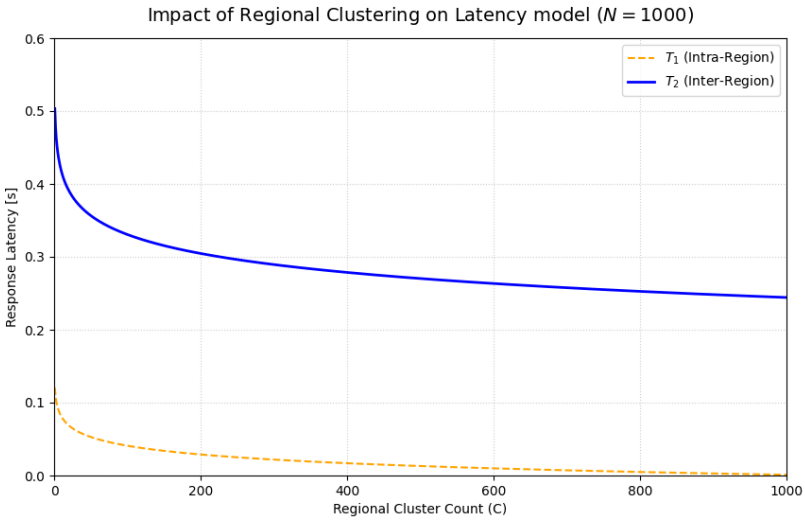

図 5.9: Regional Cluster 数の変化に対する Intra-Region 遅延 T1 と Inter-Region 遅延 T2 の比較

Intra-Region の応答遅延 T_1 は単調に減少する傾向が確認できる。これは、各 Regional Cluster に含まれるノード数が N/C に分割され、DHT 探索に必要な hop 数が減少するためである。すなわち、クラスタを細分化するほど Region 内での探索コストは低下する。

同じく Inter-Region の応答遅延 T_2 は、クラスタ数 C の増加に伴い単調に減少する傾向が確認できる。このモデルから、Regional Cluster 数に対する T_1 と T_2 の差分は、Context Share の物理配置を設計する上での有用な評価指標となり得る。また，本理論モデルで用いた各種パラメータは実運用環境に応じて調整可能であり、ネットワーク遅延や処理時間の特性を反映したモデル化が可能である。これらのパラメータを考慮した上で、最適なクラスタ数を決定することが今後の課題である．

第6章　今後の展望

本章では、本研究で提案した Replication Layer および Context Share を実運用へ近づける上で残る課題と、それに対する今後の発展方向を述べる。特に、本研究は「障害のきっかけが発生しても大規模障害へ波及しにくい」制御系を目指しており、より大規模かつ現実的な条件での評価が今後の主要課題となる。

6.1　Replication Layer に伴う加入者情報の運用ポリシーの検討

中央に位置する UDR のみが加入者情報を扱う場合、常に一貫性のある加入者情報が運用されている前提を敷くことができる。一方、UDR に加えて Replication Layer も加入者情報を扱う場合は、一貫性をどの程度保つべきかという議論が生じる。加入者情報は読み出しだけではなく更新を伴う書き出しの対象にもなるため、各地に分散配置される Replication Layer 上の加入者情報が更新されたとき、データの親元である UDR との間に不一致が発生する可能性がある。運用上は UDR を加入者データベースとして扱うため、端末の状態（Replication Layer に保存されている状態）と UDR の状態に差があることをある程度許容する必要がある。また、Replication Layer は本来 UDR にしか存在しない加入者情報を複製して網内で流通させる点において、従来の加入者情報の運用ポリシーと大きく異なる。このように、モバイルシステムの耐障害性向上は、加入者情報の運用ポリシーにも変更を与える影響の大きいテーマである。

6.1.1　Context Share における複製制御の高度化

本研究の Context Share は、UE コンテキストを DHT へ外部化することで、輻輳、分断時の影響を局所化し、サービス継続性を高めることを狙った。一方で、現状の実装はコンテキストを基本的に単一担当ノードに保持するため、担当ノード自体の障害やリージョン全体の喪失に対しては、耐性が限定的である。この課題に対して、今後は複製（レプリケーション）を導入し、「到達可能な場所に状態を残す」設計へ発展させる必要がある。

複製制御を導入する際には、単に複製数を増やすだけでなく、障害形態と通信形態に応じて複製範囲を調整することが重要である。例えば、(a) 輻輳対策として

はホットキー（特定 IMSI への集中）を複数ノードへ分散させる局所複製が有効である一方で、(b) 分断対策としては地域内に必ず複製が残るような地域内複製が有効である。さらに、(c) ノード消失に備えるためには、異なるラック、異なる拠点へ複製を分散する必要がある。このように、障害パターン（a から c）を前提として「どの単位（ノード、地域、上位層）で複製を持つべきか」を整理し、運用上許容可能な複製コストと遅延増加の範囲で設計することが課題となる。

また、複製を導入すると整合性制御が不可避となる。UE コンテキストは移動やセッション更新により頻繁に更新されるため、強い一貫性を常時要求すると更新遅延が増大し、ハンドオーバ時の遅延や輻輳を招き得る。一方で、整合性が弱すぎると、認証、課金、セッション制御に影響し得る。そのため、更新頻度や重要度に応じて整合性レベルを切り替える方式を用いた設計が検討課題となる。

6.1.2 実運用に近い障害ケースの評価

本研究の評価は、実験環境の制約の中でベースライン性能と、分断時の通信継続性（スループット推移）を中心に示した。しかし、実運用では端末数が 1,000 万規模であり、障害形態も複合的（分断と輻輳、復旧直後の再接続集中など）に発生する。したがって今後は Replication Layer と Context Share を踏まえたより大規模な条件を想定した評価が必要となる。

評価の方向性としては、(i) エミュレーション規模の拡大（多数 UE、多数 ADRAN、多数 DHT ノード）、(ii) 障害注入の体系化（リンクダウン、ノード停止、遅延注入、パケットロス注入の組合せ）、(iii) 復旧過程の評価（復旧ストーム時の収束時間、成功率、通信断時間の分布）が重要となる。特に、通信断時間や復旧時間の分布を評価することで、「大規模障害へ波及しにくい」設計かどうかをより説得的に示せる。また理論モデルはパラメータとして、遅延などを用いているがこの遅延は実際の大規模環境下でどのように変化するかを評価することも重要である。大規模な Context Share においては、DHT ノード数の増加に伴い探索遅延がどのように変化するかを実験的に評価し、理論モデルの妥当性を確認することも今後の課題である。

6.2 まとめ

本研究では、ADM 環境における通信サービス継続性の向上を目的として、加入者情報の可用性を高める Replication Layer と、コンテキスト可用性を高める Context Share を提案した。Replication Layer は UDR への依存を低減し、UDR 障害や TN 分断時でも加入者情報の取得を継続できる可能性を示した。Context Share は UE コンテキストを分散 KVS 上に配置し、輻輳、分断時の影響を局所化することで、

制御負荷の増幅を抑制できることを示した。さらに、評価を通じて平常時オーバーヘッドと通信継続性の観点から提案方式の有効性を整理した。

今後は、複製制御と整合性設計の高度化、運用ポリシーの明確化、および大規模条件、複合障害下での評価を進めることで、「障害のきっかけが発生しても大規模障害へ波及しにくい」モバイルシステム設計としての実運用適用可能性を高めることが課題である。

謝辞

本研究を行うにあたり、多くの方から多大なご助言やご助力をいただきました。それらの方々のご協力がなければ、本研究は成り立ちませんでした。ここに深く感謝し、心から厚く御礼申し上げます。本研究を進めるにあたり、主指導教員である宇多仁准教授には適切な助言とご指導を賜りました。また、共同研究先であるSoftBank 渡邊大記氏には研究の進め方に加え、設計や評価に関する助言やモバイルネットワークに関する知識の提供など、様々なご指導をいただきました。自律分散モバイルシステムに携わった 2 年間にわたり、多大なご支援を賜り、誠にありがとうございました。深く感謝いたします。また同じく共同研究先である SoftBank 張 亮氏には、研究の技術的な側面に関して多くのご助言をいただきました。深く感謝いたします。また、インターンシップ指導教員である丹康雄教授には、中間発表と修士論文審査会において、モバイルネットワークの観点から適切なご意見をいただきました。深く感謝いたします。本研究室の元指導教員である篠田陽一氏、修了生の中川颯馬氏、本研究室の中村一貴氏、西村優典氏、Lu Xingchen 氏には研究に関する様々な議論や研究生活を送る上での多大なご助力をいただきました。深く感謝いたします。最後に、学生生活を支えていただいた家族へ心から感謝いたします。修士論文の提出に至るまで多くの方からご支援をいただきました。ありがとうございました。

参考文献

[1] 3GPP, “System architecture for the 5G System (5GS),” TS 23.501, v16.6.0, 2020. Accessed: Sep. 06, 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/16.06.00_60/ts_123501v160600p.pdf.

[2] KDDI, “通信障害に関するお知らせ,” 2022. Accessed: Sep. 06, 2024. [Online]. Available: https://news.kddi.com/kddi/corporate/newsrelease/2022/07/29/6183.html.

[3] Fierce Network, “FCC reports on AT&T’s nationwide outage in February,” 2024. Accessed: Sep. 06, 2024. [Online]. Available: https://www.fierce-network.com/wireless/fcc-reports-atts-nationwide-outage-february.

[4] H. Watanabe, K. Akashi, K. Shima, Y. Sekiya, and K. Horiba, “A Design of Stateless 5G Core Network with Procedural Processing,” 2023 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10299772.

[5] Y. Li, H. Li, W. Liu, et al., “A case for stateless mobile core network functions in space,” SIGCOMM ’22: Proceedings of the ACM SIGCOMM 2022 Conference, pp. 298-313, 2022. [Online]. Available: https://doi.org/10.1145/3544216.3544233.

[6] H. Watanabe, K. Shima, and K. Horiba, “Autonomous Decentralized Mobile System: C-Plane RAN Core Convergence for Stable Network,” Proceedings of the 20th Asian Internet Engineering Conference (AINTEC ’25), pp. 52-60, 2025. [Online]. Available: https://doi.org/10.1145/3763400.3763449.

[7] P. Ganesan, K. P. Gummadi, and H. Garcia-Molina. Canon in g major: Designing dhts with hierarchical struc-ture. In Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS) pages 263 – 272, 2004.[Online]. Available: https://people.mpi-sws.org/~gummadi/papers/hierarchical_dhts.pdf.

[8] H. Kim, J. Park, S. Han, and M. Kim, “MoHiD: A Scalable Mobility Platform based on Hierarchical Distributed Hash Table,” Hawaii International Conference on System Sciences (HICSS), 2018. [Online]. Available: https://aisel.aisnet.org/hicss-51/st/wireless_networks/2/.

[9] free5GC Project. free5GC: Open source 5G core network https://free5gc.org/. accessed 2026-03-22.

[10] Hewlett Packard Enterprise. PacketRusher: High performance 5G UE/gNB simulator and CP/UP load tester https://github.com/HewlettPackard/PacketRusher.

本研究に関する対外発表

- Takata Atsuki, Hiroki Watanabe, Keiichi Shima, Katsuhiro Horiba, Satoshi Uda, Yoichi Shinoda. “Distributed AAA in an Autonomous Distributed Mobile System Environment,” IEICE Technical Report, vol. 123, no. 397, NS2023-209, pp. 220-223, 2024. (査読なし, 発表済み)
- 髙田 敦生 自律分散モバイルシステムにおける AAA WIDE Project ポスターセッション, Dec, 2025.
- 髙田 敦生、渡邊 大記，張 亮（ソフトバンク株式会社)、宇多 仁 自律分散モバイルシステムにおける分散コンテキスト管理 第 206 回マルチメディア通信と分散処理・第 112 回コンピュータセキュリティ合同研究発表会, Mar, 2026. (査読なし, 発表済み)