

進捗報告資料

安達智哉

to-adachi@ist.osaka-u.ac.jp

2019 年 3 月 7 日

1 概要

MME および SGW、PGW などの EPC ノードは、主なリソースとして CPU とメモリを持っている。CPU は、アタッチやデタッチなどのシグナリング処理を実行するために必要とされるリソースである。一方メモリは、ベアラなどのセッション情報を保持するために必要とされるリソースである。これらのリソースは、モバイルネットワークにおける通信を可能にするために必須であるため、ネットワーク事業者は、どちらのリソースも枯渇することがないように、CPU とメモリをバランスよく割り当てる必要がある。

その一方で、近年は M2M/IoT 端末の急激な増加が注目されている。M2M/IoT 端末は通信特性において従来の端末 (携帯電話やスマートフォンなどのユーザ端末) とは大きく異なり、データの送信に周期性や間欠性を持つという特徴がある。そのため、データの送信ごとに idel 状態と connected 状態を遷移することが予想される。その結果、端末のネットワーク接続やデータ送信に必要なシグナリングに関する通信や処理を行う、制御プレーンの輻輳が悪化すると考えられる。また、M2M/IoT 端末は消費電力を抑えることが必要とされている。このような問題に対し、RRC Connected Inactive や RRC Suspended と呼ばれる M2M/IoT 端末の新たなステートを導入することによって、消費電力およびシグナリングの削減を目標とする研究が行われている。RRC Connected Inactive や RRC Suspended とは、M2M/IoT 端末の情報が端末及びネットワークに保存され、RAN-CN 間の接続がアクティブ状態で維持されている状態である。図 1、2、3 に Connected および Connected Inactive、Idle 状態における、Access Stratum (AS) レイヤー以下の接続を示す。Connected Inactive 状態においては、一部の情報が保持されているため、connected 状態へ遷移する際に発生するシグナリングは、idel 状態と connected 状態を遷移することによって発生するシグナリングよりも小さくなることが予想される。さらに、シグナリングの削減に伴い、端末の消費電力削減も期待できる。実際、文献 [1] および [2] においては、RRC Connected Inactive を導入することにより、シグナリングオーバーヘッドの削減および消費電力の削減が可能であることを示している。

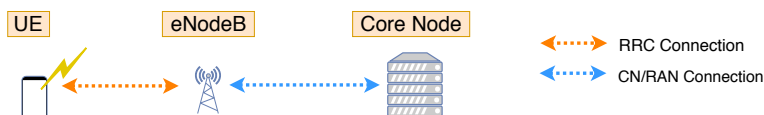


図 1: Connected

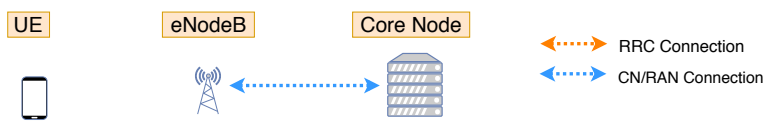


图 2: Connected Inactive

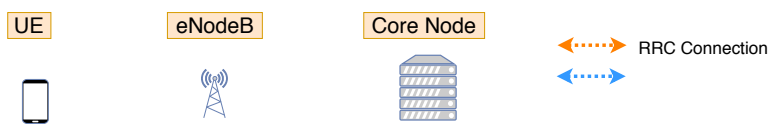


图 3: Idle

上述の RRC Connected Inactive および RRC Suspended を M2M/IoT 端末に適応することは、CPU やメモリなどのサーバリソースの効率的な割り当てを難しくすると考えられる。なぜなら、RRC Connected Inactive や RRC Suspended はその特性から、端末がデータを送信していないタイミングにおいてもその端末情報をネットワークに保持するため、コアネットワークノードのメモリに対してこれまで以上の大きな負荷を発生させるためである。また、M2M/IoT 端末の接続台数の予測が難しいこともリソースの割り当てを難しくする一因である。M2M/IoT 端末は、スマートフォンのようなユーザ端末とは異なり、家電や自動車、電気メーター、センサなど様々な場所、様々な用途で使用される可能性があり、端末の台数およびその分布を予測することは困難であると考えられる。このように、新たな状態の導入や、接続台数の予測が難しい IoT 端末の普及により、今後のネットワーク事業者は、コアネットワークノードへのサーバリソースの割り当てがより難しくなると予想される。

上述のようなネットワーク (サーバリソース消費の予測が難しく、変動が激しいネットワーク) において、収容可能な端末の増加を目的とした既存研究には、スケールアウトの考え方を採用しているものが多い。これらの研究では主に稼働するサーバやインスタンスの数をリソースの需要に応じて変動させることにより、ネットワークの変動に対応している。しかし、この方法では、本来必要とされているリソース量 (需用量) よりも多くのリソースが供給される、オーバプロビジョニングが発生する問題がある。なぜなら、これらの研究では、サーバやインスタンス一台あたりのリソース量は一定であることを前提とした研究が多く、細かい粒度でリソースを制御できないためである。また、必要とされる CPU とメモリのリソース比があらかじめ分かっていることを前提とした研究が多く、必要とされるリソース比が未知の場合はリソースの効率的な利用ができないからである。例えば、CPU のリソース不足を解消するためにスケールアウトを行った場合、CPU リソースと同時にメモリリソースも増加する。しかし、メモリは元々ボトルネックにはなっていないため、新たに追加されたメモリはオーバプロビジョニングされたことになる。

このような背景から、CPU とメモリのリソース消費の予測が難しような状況や変動が大きいような状況においても、どちらかがボトルネックにならずに、効率的にリソースを活用するアーキテクチャを考えることは重要である。実際、CPU とメモリのリソースを効率よく活用する研究は、データセンターなどの分野では行われている [3]。しかし、モバイルネットワークに特化した研究は行われていない。そこで、私はモバイルネットワークに特化した、CPU とメモリのリソースのオフロードを可能にする仕組みを考案する。この方法により、CPU が過負荷である場合は、メモリの負荷を増加させる代わりに CPU の負荷を削減することが可能である。またその逆に、メモリが過負荷である場合は、CPU の負荷を増加させることによりメモリの負荷を削減できる。この仕組みにより、CPU とメモリのリソース消費の予測が難しい場合や、変動が激しいネットワークであっても、CPU およびメモリ双方のリソース利用率の向上が期待でき、収容可能な端末の増加が期待できる。

2 提案手法

また、UE が最後にデータ送信した後、Idle 状態に移移するまでの時間を Idle timer と定義する。
また、ネットワークが収容できる端末の最大数を、収容可能な端末台数と定義する。

本研究では、CPU とメモリのリソース利用率を最適化し、収容可能な端末台数を最大化するような、Idle timer の制御方法を提案する。

2.1 Idle timer の設定

提案手法では、Idle timer を CPU およびメモリリソースの負荷状況に応じて動的に決定する。Idle timer の設定によってコアノードのリソースの需要が変化する。提案手法では、ユーザ端末と M2M/IoT 端末の存在割合に応じて、収容可能な UE 台数を最大化するような Idle timer を計算によって導出する。処理のフローを図 4 に示す。常に CPU とメモリのリソース負荷を監視し、CPU が過負荷であれば Idle timer を増加させ、反対にメモリが過負荷であれば Idle timer を減少させる。

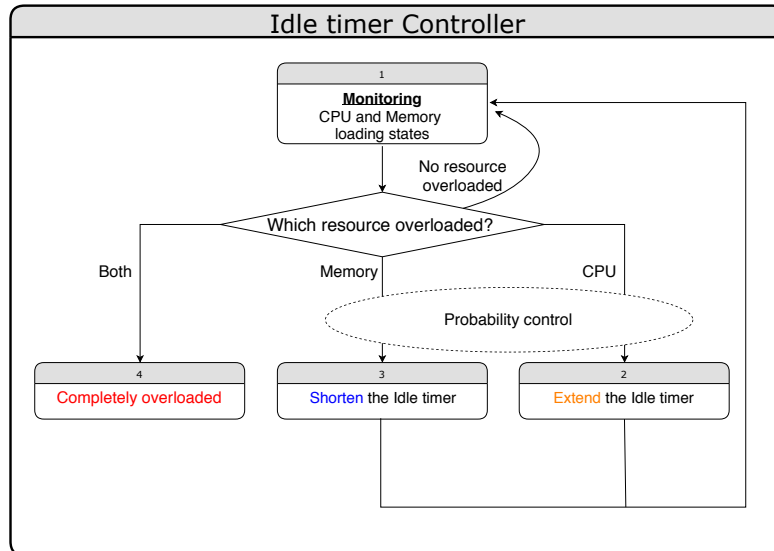


図 4: The proposed method

3 解析モデル

3.1 モバイルネットワーク

単一の EPC で構成されるモバイルネットワークを対象として評価を行う。図 5 に評価対象のネットワークを示す。図 5 は以下のノードから構成される。

- UE
- eNodeB
- MME
- S/PGW

UE および eNodeB は複数台、その他のノードは 1 台ずつ存在する。

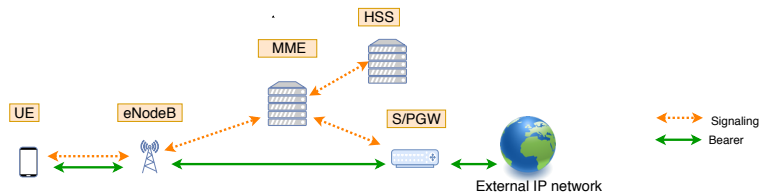


図 5: LTE/EPC ネットワークモデル

3.2 状態遷移モデル

UE が最後にデータを送信した後、Connected Inactive 状態に移行するまでの時間を $T_{connect}$ 、Idle timer を T_{timer} と定義する。ある UE におけるデータの送信間隔を T_h と定義した時、この UE の状態遷移を図 6 に示す。まず、UE はデータを送信すると、Connected 状態へ移行する。データ送信後 $T_{connect}$ 時間 Connected 状態を維持したのち、Connected Inactive 状態へ移行する。そして、データ送信後 T_{timer} 後に Idle 状態へ移行する。

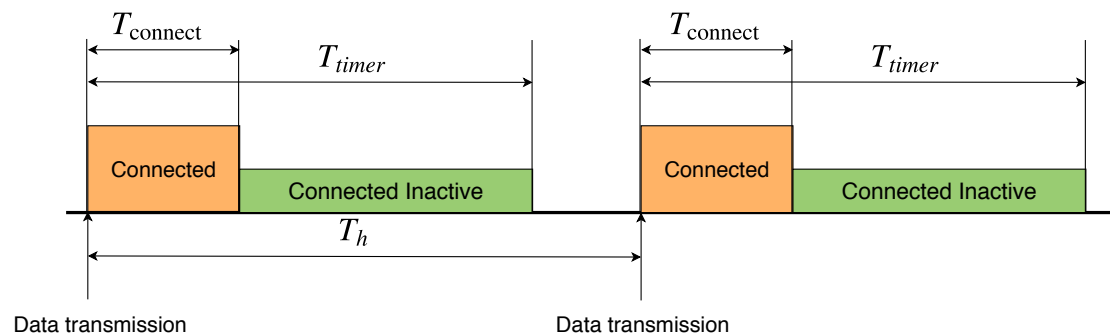


図 6: 状態遷移モデル

3.3 UE の通信特性

UE の通信特性は UE の種類によって異なる。例えばユーザ端末や M2M/IoT 端末には以下のような特徴がある。

ユーザ端末 連続的なデータ通信を行う。通信データ量は多いが、常時ネットワークに接続した状態であるため、シグナリング処理の発生頻度が小さい。

M2M/IoT 端末 間欠的な通信を行う。データ送信の頻度は少ないが、データ送信の度にネットワークへの接続および切断を繰り返すため、シグナリング処理の発生頻度が大きい。また、データの送信周期に偏りがあると考えられる。

上述のように UE の用途に応じて通信特性は変化する。そのため今回の評価では、ユーザ端末およびデータの送信間隔が異なる複数種類の IoT 端末が存在することを想定して評価を行う。また、それぞれの UE の存在割合をパラメータとして変化させる。

3.4 サーバ資源の割り当て

今後検討予定である。

3.5 解析環境

今回の研究はシミュレーションを用いてネットワークの性能評価を行う。シミュレーションのタイムステップは 1 分から 5 分程度を想定している。各タイムステップでは、UE の通信特性と Idle timer に基づき EPC の負荷を算出する。

EPC の負荷の算出のイメージ図を図 7 に示す。UE の通信特性と Idle timer を入力とし、EPC の負荷を出力とする。EPC のメモリ負荷の導出のために上野さんの実験結果を用いる。

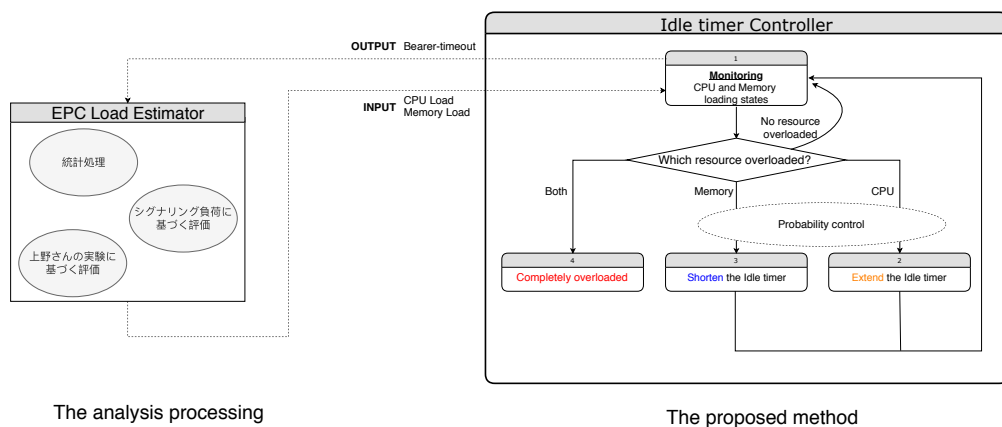


図 7: 解析環境

3.5.1 統計処理

この項では、UE の通信特性と Idle timer から、モバイルネットワークにおけるシグナリングの発生レートおよびベアラの平均接続数を導出する。

まず、ネットワーク全体におけるシグナリングの発生レートを求める。 N_{UE} 台の UE がネットワークに存在すると仮定した時の UE の集合 \mathbf{U} を、 $\mathbf{U} = \{u_1, u_2, \dots, u_{N_{UE}}\}$ と定義する。ある UE u_h ($u_h \in \mathbf{U}$) が 1 秒あたりに発生させるシグナリングの数を s_h と定義する。最後のデータ送信から Connected Inactive 状態へ遷移するまでの時間を $T_{connect}$ 、Idle timer を T_{timer} 、 u_h の通信周期を T_h とする。また、状態遷移に伴うシグナリングの発生回数をそれぞれ表 1 のように定義すると、 s_h は以下の式 (1) で表せる。ここで留意すべきは、 T_h が $T_{connect}$ 以下であるような UE は、一度 Connected 状態に遷移すると、常時 Connected 状態を維持し、状態遷移によるシグナリングが発生しないため、 s_h は 0 と定義したことである。

表 1: state signaling

Source	Destination	The number of signaling occurrences
Connected	Connected Inactive	$n_{connect \rightarrow inactive}$
Connected Inactive	Connected	$n_{inactive \rightarrow connect}$
Connected Inactive	Idle	$n_{inactive \rightarrow idle}$
Idle	Connected	$n_{idle \rightarrow connect}$

$$s_h = \begin{cases} 0 & \text{if } T_h \leq T_{connect} \\ \frac{1}{T_h} \cdot (n_{inactive \rightarrow connect} + n_{connect \rightarrow inactive}) & \text{if } T_{connect} \leq T_h \leq T_{timer} \\ \frac{1}{T_h} \cdot (n_{idle \rightarrow connect} + n_{connect \rightarrow inactive} + n_{inactive \rightarrow idle}) & \text{otherwise} \end{cases} \quad (1)$$

1 秒毎にネットワーク全体で発生するシグナリングの合計を S と定義する。 S は s_h を用いて以下の式 (2) で表せる。

$$S = \sum_{h=1}^{N_{UE}} s_h \quad (2)$$

次に、各状態ごとの UE 数を求める。ある時刻において、UE u_h が Connected 状態である確率を $b_{connected}$ 、Connected Inactive 状態である確率を $b_{inactive}$ 、Idle 状態である確率を b_{idle} と定義する。するとこれらの値は、 T_h および T_{timer} 、 $T_{connect}$ を用いて以下の式 (3)、(4)、(5) で表せる。

$$b_{connected} = \begin{cases} 1 & \text{if } T_h \leq T_{connect} \\ \frac{T_{connect}}{T_h} & \text{otherwise} \end{cases} \quad (3)$$

$$b_{inactive} = \begin{cases} 0 & \text{if } T_h \leq T_{connect} \\ \frac{T_h - T_{connect}}{T_h} & \text{if } T_{connect} \leq T_h \leq T_{timer} \\ \frac{T_{timer}}{T_h} & \text{otherwise} \end{cases} \quad (4)$$

$$b_{idle} = \begin{cases} 0 & \text{if } T_h \leq T_{connect} \\ 0 & \text{if } T_{connect} \leq T_h \leq T_{timer} \\ \frac{T_h - T_{timer}}{T_h} & \text{otherwise} \end{cases} \quad (5)$$

Connected 状態である UE の平均数および Connected Inactive 状態である UE の平均数、Idle 状態である UE の平均数を $B_{connected}$ 、 $B_{inactive}$ 、 B_{idle} と定義すると、それらは、 $b_{connected}$ および $b_{inactive}$ 、 b_{idle} を用いて以下の式 (6)、(7)、(8) で表せる。

$$B_{connected} = \sum_{h=1}^{N_{UE}} b_{connected} \quad (6)$$

$$B_{inactive} = \sum_{h=1}^{N_{UE}} b_{inactive} \quad (7)$$

$$B_{idle} = \sum_{h=1}^{N_{UE}} b_{idle} \quad (8)$$

3.5.2 シグナリング負荷に基づく評価

ネットワーク全体におけるシグナリングの発生レートから、CPU にかかる負荷を算出する。

3.5.3 上野さんの実験に基づく評価

上野さんの実験データを基にメモリ負荷を導出する。図 8、図 9、図 10 に示す実験データから、ベアラの確立が EPC の各ノード (MME、SPGW、HSS) のメモリに与える影響を測定した。その結果、UE 1 台あたりのベアラの確立がメモリに与える負荷は MME、SPGW、HSS でそれぞれ 750KB、350KB、6KB であることが分かった。この結果を参考にしてメモリ負荷を算出することを考えている。

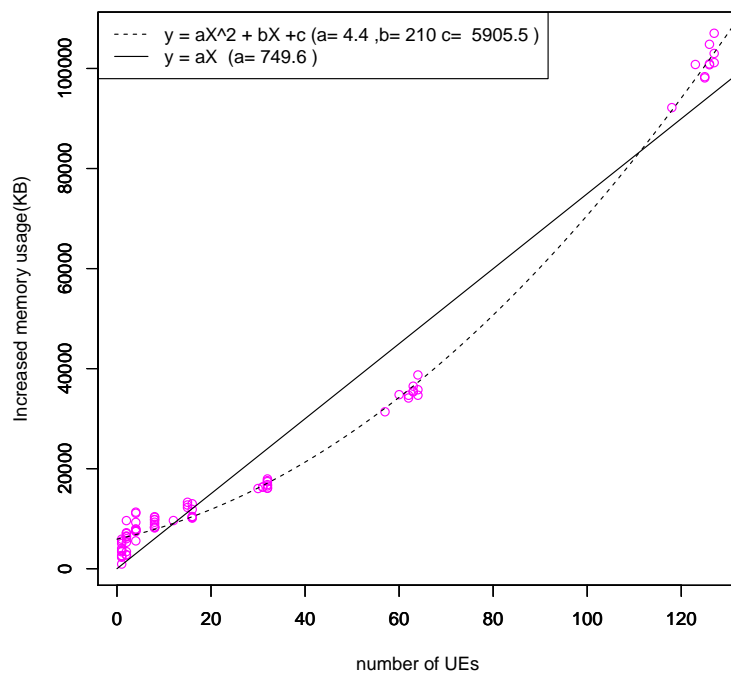


図 8: ベアラの確立によって増加したメモリ使用量 (MME)

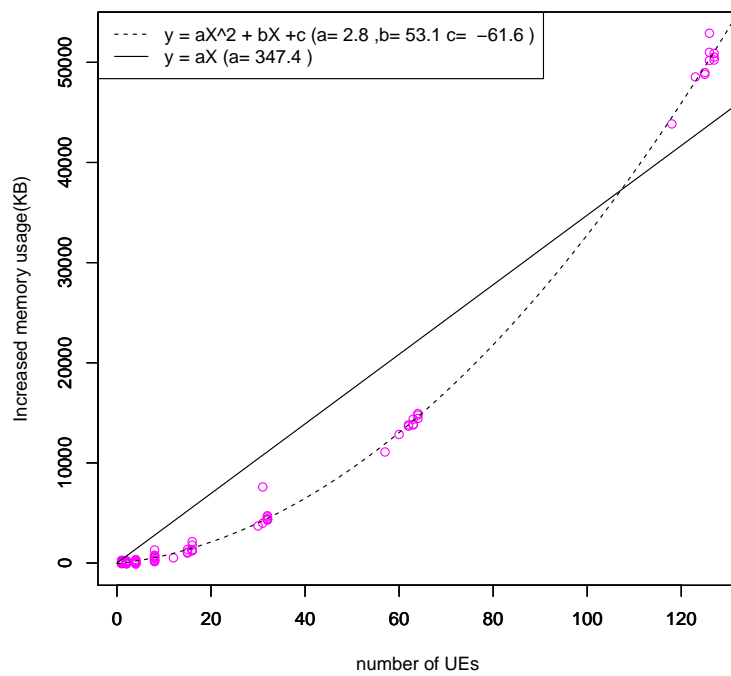


図 9: ベアラの確立によって増加したメモリ使用量 (SPGW)

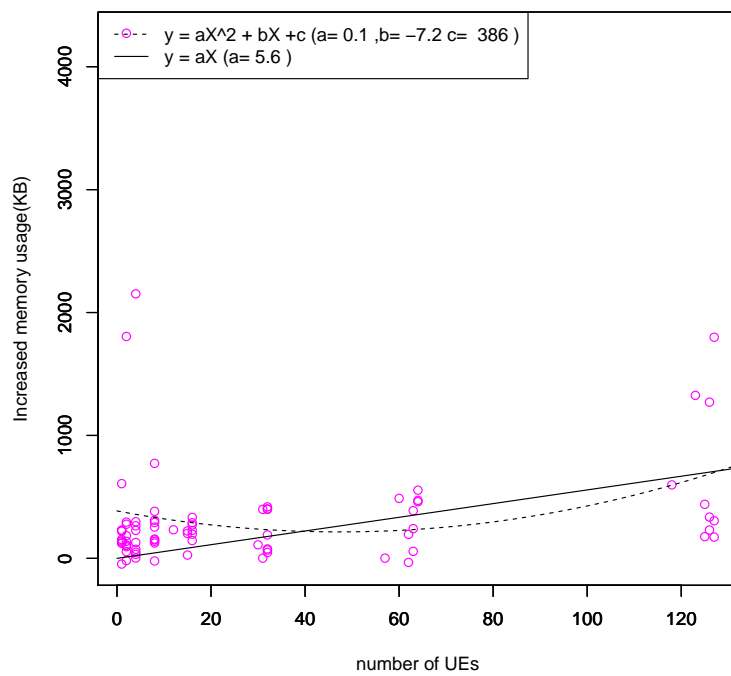


図 10: ベアラの確立によって増加したメモリ使用量 (HSS)

4 先行研究調査

サーバのリソース分離に関する文献 [3] を調査した。この文献では、データセンタのコストを削減することを目的し、サーバリソースの分離を提案している。そして、サーバごとにリソース構成が固定されている従来のデータセンタと、CPU とメモリのリソース構成をサーバから分離したデータセンタとの比較を行っている。評価の結果、リソースを分離することによって、CPU とメモリのオーバープロビジョニングを削減できることを示している。そして、コスト面では（アプリケーションにも依存するが）最大 40% の削減が期待できると結論づけている。

文献 [4] では、モバイルネットワークにおいてハンドオーバーに関わるシグナリングの発生回数を減らすことを目的とし、Delay Time Algorithm を提案している。このアルゴリズムでは、従来の仕組みであればハンドオーバー処理を行うような状況であっても、あえてハンドオーバー処理を行うタイミングを遅らせることによって、シグナリングの発生を削減することを可能としている。

文献 [5] では、UE が Connected 状態から idle 状態へ遷移するまでの時間 (inactive timer) を制御することによって、シグナリング発生回数と UE の消費電力を最適化する研究を行っている。inactive timer を大きくすることによって、UE の状態遷移に伴うシグナリングの発生回数を抑えることが可能である一方、UE が長い時間 Connected 状態に止まるために消費電力が増加することを明らかにしている。

文献 [1] では、RRC Connected Inactive state と呼ばれる状態を新しく導入することによって、シグナリングの発生回数及び UE の消費電力を削減できることを示している。

5 今後の課題

- デタッチの処理負荷を求めるため、OAI に基づくデタッチ処理のプログラム行数を調査する。
- UE の通信特性の分布を決定する上で、根拠となるデータを見つける。
- メモリの負荷が二次関数的な増加を示している理由を調査する。調査方法としては、UE 数および eNodeB 数のどちらか一方を固定し、もう一方を変化させつつベアラ確立の実験を行い、メモリ負荷を観測する。もしくは、OAI のソースコードを読みデータの保持に関する実装を調査する。

参考文献

- [1] S. Hailu, M. Saily, and O. Tirkkonen, “RRC State Handling for 5G,” *IEEE Communications Magazine*, vol. 57, no. 1, pp. 106–113, Jan. 2019.
- [2] I. L. Da Silva, G. Mildh, M. Saily, and S. Hailu, “A Novel State Model for 5G Radio Access Networks,” in *Proceedings of 2016 IEEE International Conference on Communications Workshops (ICC)*, May 2016, pp. 632–637.
- [3] M. Mahloo, J. M. Soares, and A. Roozbeh, “Techno-Economic Framework for Cloud Infrastructure: A Cost Study of Resource Disaggregation,” in *Proceedings of 2017 Federated*

Conference on Computer Science and Information Systems (FedCSIS), Sep. 2017, pp. 733–742.

- [4] C. Lee and P. Lin, “Modeling Delay Timer Algorithm for Handover Reduction in Heterogeneous Radio Access Networks,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 1144–1156, Feb. 2017.
- [5] Q. Liao and D. Aziz, “Proceedings of Modeling of Mobility-Aware RRC State Transition for Energy-Constrained Signaling Reduction,” in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2016, pp. 1–7.