社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

モバイルコアネットワークのノード資源利用の効率化のための シグナリング制御手法

安達　智哉†　阿部　修也†　長谷川　剛††　村田　正幸†

† 大阪大学大学院情報科学研究科　〒 565–0871 大阪府吹田市山田丘 1–5
†† 東北大学電気通信研究所　〒 980–0812 宮城県仙台市青葉区片平二丁目 1 番 1 号
E-mail: †{to-adachi,s-abe,murata}@ist.osaka-u.ac.jp, ††hasegawa@riec.tohoku.ac.jp

あらまし　モバイルネットワーク事業者は，自身が運用するモバイルコアネットワークのノード資源が枯渇しないように，収容端末台数や接続頻度に応じて資源割り当てを行う必要がある．一方，近年増加している IoT 端末は，接続される端末の台数を予測することは困難である．また，端末の通信開始時のシグナリング手順を削減するために，RRC Connected Inactive と呼ばれる状態を導入し，端末情報をメモリに一時的に保存することが検討されている．これらのことから，今後，モバイルコアネットワークノードへの CPU 負荷やメモリ使用量が大きく変動することが予想され，効率的な資源割り当てが難しくなる．そこで本報告では，資源割り当てを動的に行うのではなく，端末の通信時のシグナリング手順のパラメータを調整することにより，割り当てられている資源を効率的に使用し，収容可能な端末台数を増加する手法を提案する．具体的には，端末がアイドル状態に遷移するまでのタイムアウト時間を動的に調整することで，モバイルコアノードの CPU 負荷とメモリ消費量を制御する．シミュレータを用いた性能評価の結果，提案手法を用いない場合には，接続端末が増加した際にシステムの CPU あるいはメモリ資源が不足する場合において，提案手法を用いることで，CPU とメモリの負荷をお互いに調整し，新たな端末を接続できる状態が維持されることを確認した．

キーワード　モバイルコアネットワーク，IoT 通信，資源割り当て，RRC Connected Inactive，PID 制御

1. ま え が き

モバイルネットワーク事業者は，自身が運用するモバイルコアネットワークのノード資源が枯渇しないように，収容端末台数や接続頻度に応じて資源割り当てを行う必要がある．主なノード資源として，CPU およびメモリが挙げられる．CPU は，端末のアタッチやデタッチ等のシグナリング処理やメッセージの送受信のために必要とされる資源である．一方メモリは，ベアラなどの端末のセッション情報を保持するために必要となる．これらのノード資源は，モバイルネットワークサービスを提供するために必須であり，どちらか一方でも枯渇することは許されない．

一方，近年 IoT 端末の急激な増加が注目されている．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，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モバイルコアネットワークの導入時に予測することは困難である．そのため，多数の IoT 端末を収容するためにノード資源を過不足なく割り当てることは難しい．

また，IoT 端末はスマートフォン等の従来の端末とはその通信特性が異なり，データ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，データ送信ごとにアイドル状態と接続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．このような問題に対し，RRC Connected Inactive と呼ばれる新たな状態を導入することによって，特に IoT 端末を対象に，シグナリング手順の削減を目標とする研究が行われている [1,2]．

このように，接続台数の予測が難しい IoT 端末の普及や，モバイルコアネットワークノードに与える負荷を変化させるような新たな状態の導入により，モバイルコアネットワークノードの CPU 負荷やメモリ使用量が時間的に大きく変動することが予想される．そのため，モバイルネットワーク事業者は，これまで以上に効率的に資源割り当てを行う必要がある．

上述のような資源需要の予測が難しく，変動が激しいモバイルコアネットワークにおいて，収容可能な端末台数の増加を目的とした既存研究として，稼働するサーバやインスタンス数を需要に応じて増減させる手法が提案されている [3–7]．しかし，そのような手法では，本来必要とされているよりも多くの資源が供給されることがある．なぜなら，サーバやインスタンス一台あたりの資源構成は固定であることが一般的であり，偏った資源需要に対してサーバやインスタンスを増加すると，本来増強する必要のない資源も供給されるためである．文献 [4] では，IoT 端末を収容している MME のノード台数を単純に増加すると，一部の資源が過剰に供給される可能性があることを述べている．

また，サーバの資源を効率よく活用するためにサーバの資源を分離し，各資源を個別に増強，更新可能とする Server Disaggregation アーキテクチャが提案されている [8,9]．文献 [8] では，Server Disaggregation アーキテクチャをデータセンタに適用することで，CPU やメモリなどの資源を需要に合わせて自由に組み替えることが可能になり，資源の効率的な利用が可能であることが示されている．しかし，Server Disaggregation は，年単位などの長期的なサーバ更新のためのアーキテクチャであり，本報告で対象とするような短期間での負荷変動に対応することを目的とした方式ではない．文献 [9] では，Server Disaggregation アーキテクチャを適用したシステムにおいて，数時間以下の時間スケールで資源制御を行った場合，サーバ資

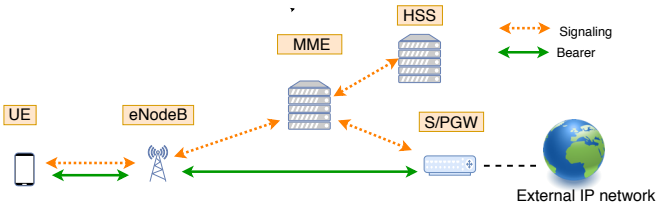

図 1: モバイルコアネットワークモデル

源の再割り当て処理に伴う遅延やコスト面でのオーバーヘッドが大きくなることが示されている．

本報告では，資源割り当てを動的に行うのではなく，割り当てられている資源を効率的に使用することにより，収容可能な端末台数を増加させる手法を提案する．具体的には，時間的に変動するモバイルコアネットワークの負荷に応じて，端末の状態を適応的に制御することにより，モバイルコアネットワークノードの CPU 負荷およびメモリ使用量を調整し，収容可能な端末台数を最大化する．端末の状態の制御は，端末が最後にデータを送信したあと，Connected Inactive 状態からアイドル状態に遷移するまでの時間を適応的に設定することで実現する．そして，提案手法の有効性をシミュレーションにより評価する．具体的には，端末の状態を適応的に制御することによる，CPU 負荷とメモリ使用量の調整や，収容可能な端末台数の増加量を評価する．

本報告の構成は以下の通りである．第 2. 章では，本報告において評価対象となるモバイルコアネットワークの構成，シグナリング手順および端末の状態遷移について述べる．また，我々の過去の研究に基づき，モバイルコアネットワークノードの CPU 負荷とメモリ使用量の関係について述べる．第 3. 章では，提案手法について述べる．第 4. 章では，数値評価を行う際のシミュレータについて述べる．また，数学的解析によってモバイルコアネットワークに発生する CPU 負荷およびメモリ使用量を導出する．第 5. 章では，提案手法を用いた端末の状態制御の有効性に関して評価を行う．最後に第 6. 章で本報告のまとめと今後の課題について述べる．

2. モバイルコアネットワーク

2.1 シグナリング手順

図 1 にモバイルコアネットワークの論理構成を示す．図中の UE (User Equipment) は，スマートフォンやタブレット，IoT などの端末である．eNodeB (evolved NodeB) は，UE と無線で通信を行い，MME (Mobility Management Entity) および S/PGW (Serving Gateway / Packet Data Network Gateway) とシグナリングメッセージやユーザデータを交換する無線基地局である．MME は， UE の認証，UE の移動管理およびパケットの経路設定などを行い，モバイルコアネットワークにおけるシグナリングに関する処理の中核となるノードである．S/PGW は，MME からのシグナリングメッセージに基づいて，モバイルコアネットワークと外部のネットワーク (External IP Network) を接続するノードである．HSS (Home Subscriber Server) は，ユーザごとの契約情報，認証用のキーデータおよび MME のアドレスなどの情報を管理するデータベースノードである．

UE が外部 IP ネットワークと通信する際は，UE と eNodeB 間，eNodeB と S/PGW 間および S/PGW 内にそれぞれベアラと呼ばれる論理的なトンネルを UE 毎に確立する．

本報告においては，UE は接続状態，アイドル状態，および Connected Inactive 状態という 3 つの状態を持つものとする．接続状態とは，全てのベアラが確立されており，ユーザデータの送受信が可能な状態である．アイドル状態とは，ベアラを確立していない状態である．この状態では，UE の消費電力は

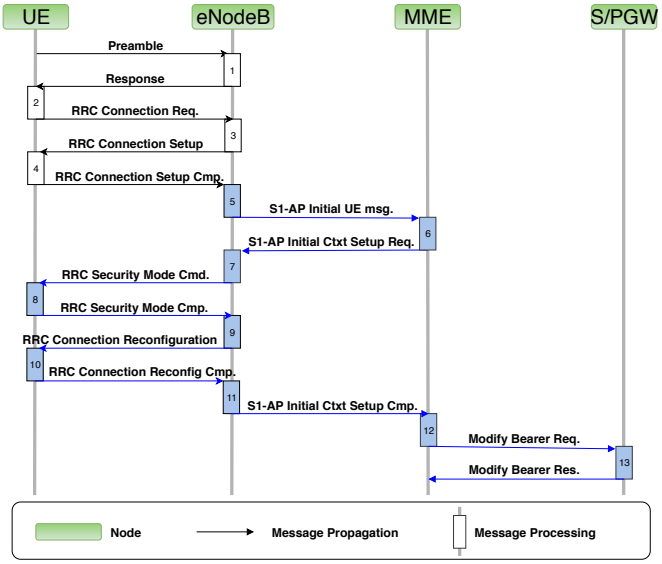

図 2: アイドル状態および Connected Inactive 状態から接続状態へ遷移する際のシグナリング手順 [2, 10]

小さいが，ユーザデータの送受信を行うためには，シグナリング手順を行い，接続状態へ遷移する必要がある．Connected Inactive 状態とは，文献 [1, 2] などで近年検討されている UE の新しい状態であり，UE と eNodeB 間のベアラは解放されているが，eNodeB と S/PGW 間および S/PGW 内のベアラは保持している状態である．この状態では，UE の消費電力は小さく，かつ，接続状態へ遷移するためのシグナリング手順を一部省略することが可能である．一方，UE のセッション情報を保持し続ける必要があり，モバイルコアネットワークノードのメモリ使用量が増加する．本報告では，Connectec Inactive 状態を導入した UE を対象とする．

図 2 に，アイドル状態および Connected Inactive 状態から接続状態に遷移するためのシグナリング手順を示す．図中の req., res., cmp., cmd., msg., ctxt はそれぞれ request, response, complete, command, message, context を意味する．また，図中で青色で示されている，5 番以降のシグナリング処理およびシグナリングメッセージは，Connected Inactive 状態から接続状態に遷移する際には省略される．

2.2 UE の状態遷移

UE の状態遷移図を図 3 に示す．図中の Connected，Idle，Connected Inactive は UE の状態を表し，それぞれ接続状態，アイドル状態，Connected Inactive 状態に相当する．図中に赤で示された状態遷移は，我々の過去の研究 [11] で提案している状態遷移であり，本報告の第 2.3 節で述べる．アイドル状態の UE は，データ送信要求が発生すると接続状態へ遷移する．その後，Inactive タイマを起動し，そのタイマが切れるまでデータの送受信が発生しなければ，Connected Inactive 状態へ遷移する．Connected Inactive 状態の UE は，データ送信のタイミングで再び接続状態へ遷移する．この時，送信データ量が小さい場合は，接続状態へ遷移することなく，データ送信が行われる．これは，シグナリングメッセージに送信データを含めることで実現される．

2.3 CPU 負荷とメモリ使用量の関係

Connected Inactive 状態を導入することにより，アタッチ後一度でも通信を行った UE は，アイドル状態への遷移が発生しないため，シグナリングメッセージの発生が抑制される．そのため，モバイルコアネットワークノードの CPU 負荷を削減できる．一方，Connected Inactive 状態では，UE のセッション情報を保持するため，モバイルコアネットワークノードのメモリ使用量が増加する．特に，今後増加すると予想される

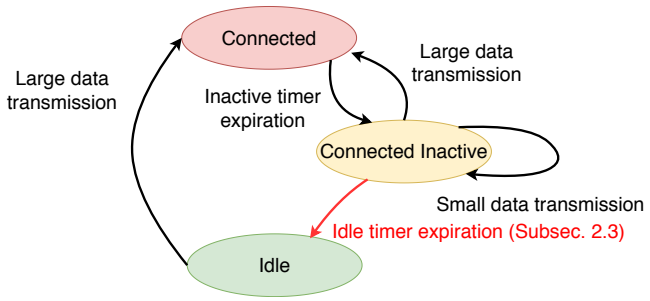

図 3: UE の状態遷移図と [11] における提案

IoT 端末には通信周期の大きなものがあり，そのような UE に対して Connected Inactive 状態を導入すると，低頻度のデータ送信に対して長期間 Connected Inactive 状態を維持するため，メモリが浪費される．

この問題に対し，文献 [11] では，Connected Inactive 状態の UE をアイドル状態へ遷移させる新たな状態遷移 (図 3 中に赤で示した状態遷移) を導入することで，モバイルコアネットワークノードの CPU 負荷およびメモリ使用量を制御できることを示した．この遷移は，Inactive タイマとは別の，Idle タイマによって制御される．UE がデータの送受信を終了したタイミングで Idle タイマが起動し，Idle タイマが切れるまでデータの送受信が発生しなければ，その UE はアイドル状態へ遷移する．文献 [11] では，CPU 負荷とメモリ使用量の間にはトレードオフの関係があることを示した．また，Idle タイマを適切に設定して CPU とメモリ間で負荷を調整することにより，両者の負荷を調整して，モバイルコアネットワークの資源の利用効率が向上することを示した．しかし，実際の環境では端末の通信周期は既知ではなく，適切な Idle タイマの値は自明ではない．さらには，モバイルコアネットワークにおける負荷は，時間と共に変動する．そこで次章では，時間的に変動するモバイルコアネットワークの負荷に応じた，適応的な Idle タイマの制御手法を提案する．

3. 提 案 手 法

3.1 概　　要

本報告では，モバイルコアノードの資源利用を効率化し，収容可能な端末台数を最大化することを目的とした，Idle タイマの適応的制御手法を提案する．収容可能な端末台数とは，現在収容している UE と同じ通信特性を持つ UE がネットワークに参加することを想定し，CPU およびメモリのいずれも過負荷状態にならずに，収容できる最大の UE 台数とする．また，モバイルコアネットワークでは，現在収容している UE 台数，CPU およびメモリ使用量を観測できるものとする．

現在収容している UE 台数を N_{UE} とする．UE 台数が N_{UE}，Idle タイマが T の時に観測される，CPU およびメモリの使用量をそれぞれ $C_{N_{\mathrm{UE}}}(T)$，$M_{N_{\mathrm{UE}}}(T)$ とする．

Idle タイマを T とした時の，収容可能な端末の台数を $N_{\mathrm{UE}}^{\mathrm{cap}}(T)$ とすると $N_{\mathrm{UE}}^{\mathrm{cap}}(T)$ は，$C_{N_{\mathrm{UE}}}(T)$，$M_{N_{\mathrm{UE}}}(T)$，$C^{\max}$ および $M^{\max}$ を用いて，以下の式 (1) で表せる．ここで，$C^{\max}$，$M^{\max}$ はそれぞれシグナリング処理および UE のセッション情報を保持するために使用可能な CPU 資源量およびメモリ資源量である．

$$N_{\mathrm{UE}}^{\mathrm{cap}}(T) = \lfloor N_{\mathrm{UE}} \cdot \min\{\frac{C^{\max}}{C_{N_{\mathrm{UE}}}(T)}, \frac{M^{\max}}{M_{N_{\mathrm{UE}}}(T)}\} \rfloor \quad (1)$$

提案手法では，$N_{\mathrm{UE}}^{\mathrm{cap}}(T)$ を最大化するように，Idle タイマを適応的に制御する．具体的には，一定の時間間隔 (タイムステップ) ごとに，各資源の使用量を観測して，$N_{\mathrm{UE}}^{\mathrm{cap}}(T)$ が大きくなるように Idle タイマを変化させる．この操作を継続的に繰り返すことにより，Idle タイマを制御する．

ここで，1 タイムステップごとの Idle タイマの変化量を小さく設定した場合，Idle タイマが最適な値に到達するまでに大きな時間がかかる可能性がある．一方，Idle タイマの変化量を大きく設定した場合，制御が不安定になる可能性がある．そこで本報告では，動作が単純であり，汎用性が高い PID 制御 [12] を用いて Idle タイマの変化量を制御する．

3.2 PID 制御の適用

我々の研究グループによる過去の研究 [11] より，CPU 使用量は Idle タイマの値に対して広義単調減少であり，かつ，メモリ使用量は Idle タイマの値に対して広義単調増加であることがわかっている．このことから，$\frac{C^{\max}}{C_{N_{\mathrm{UE}}}(T)}$ は Idle タイマの値に対して広義単調増加であり，かつ，$\frac{M^{\max}}{M_{N_{\mathrm{UE}}}(T)}$ は Idle タイマの値に対して広義単調減少である．ここで，以下の式 (2) を満たすような T の集合を $\boldsymbol{T}$ とし，$N_{\mathrm{UE}}^{\mathrm{cap}}(T)$ を最大化するような T の集合を $\boldsymbol{T}_{\mathrm{optimal}}$ とすると，$T \in \boldsymbol{T}$ であることは $T \in \boldsymbol{T}_{\mathrm{optimal}}$ であるための十分条件になる．

$$\frac{C^{\max}}{C_{N_{\mathrm{UE}}}(T)} = \frac{M^{\max}}{M_{N_{\mathrm{UE}}}(T)} \quad (2)$$

そこで，PID 制御における出力値 $y(t)$ および目標値 $r(t)$ を，式 (2) に基づいて，以下の式 (3)，(4) のように定義する．t は時刻を表す変数である．

$$y(t) = \frac{C_{N_{\mathrm{UE}}}(T)}{C^{\max}} - \frac{M_{N_{\mathrm{UE}}}(T)}{M^{\max}} \quad (3)$$

$$r(t) = 0 \quad (4)$$

$e(t)$ を以下の式 (5) ように定義すると，PID 制御における操作量 ($u(t)$) は以下の式 (6) で表せる．ここで，K_p，K_i および K_d はそれぞれ，比例ゲイン，積分ゲインおよび微分ゲインであり，これらの定数はジーグラ・ニコルスの限界感度法 [12] を用いて設定する．

$$e(t) = r(t) - y(t) \quad (5)$$

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)d\tau + K_d \cdot \frac{de(t)}{dt} \quad (6)$$

$u(t)$ に応じて Idle タイマの値を増減させることにより，CPU およびメモリ使用量の変化に対して適応的な Idle タイマの制御を行い，収容可能な端末台数を最大化することが可能になる．

4. シミュレータ

4.1 シミュレーション環境

本報告では，UE およびモバイルコアネットワークをシミュレートし，シミュレーション環境上に提案手法を適用することで，提案手法の有効性を定量的に評価する．

UE は，その状態や状態遷移，データ送信等の動作およびそのタイミングを 1 台ごとにシミュレートする．ここで，対象とする UE は周期的な通信を行うと仮定する．

モバイルコアネットワークは，UE の状態管理や UE に対する Idle タイマの設定等の動作をシミュレートする．また，モバイルコアネットワークに発生する負荷を，各 UE の状態および状態遷移に基づき以下の第 4.2 節および第 4.3 節に示すように，CPU 負荷およびメモリ使用量を算出することによりシミュレートする．本報告では，文献 [4] に基づき，MME がモバイルコアネットワーク内でのボトルネックになると仮定し，MME の CPU 負荷とメモリ使用量に着目し，提案手法を適用する．また，一般的にデータプレーンと制御プレーンの資源は分離されていることから，本報告では制御プレーンの負荷のみに着目する．

提案手法は，上述のシミュレータから MME の負荷を取得し，その値を基に Idle タイマを更新する．そして，更新された Idle タイマを上述のシミュレータへ適用する．

表 1: シグナリングメッセージ数の定義

State Transition		The number of
Source	Destination	signaling messages
Connected	Connected	$s_{\mathrm{MME}}^{\mathrm{c}\to\mathrm{c}}$
Connected Inactive	Connected Inactive	$s_{\mathrm{MME}}^{\mathrm{ci}\to\mathrm{ci}}$
Connected	Connected Inactive	$s_{\mathrm{MME}}^{\mathrm{c}\to\mathrm{ci}}$
Connected Inactive	Connected	$s_{\mathrm{MME}}^{\mathrm{ci}\to\mathrm{c}}$
Connected Inactive	Idle	$s_{\mathrm{MME}}^{\mathrm{ci}\to\mathrm{i}}$
Idle	Connected	$s_{\mathrm{MME}}^{\mathrm{i}\to\mathrm{c}}$

表 2: 時刻 t において各状態遷移を行う UE 台数

State Transition		The number of
Source	Destination	UEs
Connected	Connected	$n^{\mathrm{c}\to\mathrm{c}}(t)$
Connected Inactive	Connected Inactive	$n^{\mathrm{ci}\to\mathrm{ci}}(t)$
Connected	Connected Inactive	$n^{\mathrm{c}\to\mathrm{ci}}(t)$
Connected Inactive	Connected	$n^{\mathrm{ci}\to\mathrm{c}}(t)$
Connected Inactive	Idle	$n^{\mathrm{ci}\to\mathrm{i}}(t)$
Idle	Connected	$n^{\mathrm{i}\to\mathrm{c}}(t)$

表 3: MME のメモリ使用量の定義

State	Memory consumption
Connected	$m_{\mathrm{MME}}^{\mathrm{c}}$
Connected Inactive	$m_{\mathrm{MME}}^{\mathrm{ci}}$
Idle	$m_{\mathrm{MME}}^{\mathrm{i}}$

4.2 CPU 負荷の算出

CPU 負荷は MME が処理するシグナリングメッセージ数 (以下，メッセージ処理頻度と呼ぶ) を基に導出する．ある時刻 t におけるメッセージ処理頻度を $C(t)$ とする．UE の状態遷移に伴い，MME に発生するシグナリングメッセージ数をそれぞれ表 1 のように定義する．また，時刻 t において各状態遷移を行う UE の台数をそれぞれ表 2 のようにおくと，$C(t)$ は式 (7) で表せる．

$$\begin{aligned} C(t) &= s_{\mathrm{MME}}^{\mathrm{c}\to\mathrm{c}} \cdot n^{\mathrm{c}\to\mathrm{c}}(t) + s_{\mathrm{MME}}^{\mathrm{ci}\to\mathrm{ci}} \cdot n^{\mathrm{ci}\to\mathrm{ci}}(t) \\ &+ s_{\mathrm{MME}}^{\mathrm{c}\to\mathrm{ci}} \cdot n^{\mathrm{c}\to\mathrm{ci}}(t) + s_{\mathrm{MME}}^{\mathrm{ci}\to\mathrm{c}} \cdot n^{\mathrm{ci}\to\mathrm{c}}(t) \\ &+ s_{\mathrm{MME}}^{\mathrm{ci}\to\mathrm{i}} \cdot n^{\mathrm{ci}\to\mathrm{i}}(t) + s_{\mathrm{MME}}^{\mathrm{i}\to\mathrm{c}} \cdot n^{\mathrm{i}\to\mathrm{c}}(t) \end{aligned} \tag{7}$$

4.3 メモリ使用量の算出

ある時刻 t において，MME に対してネットワーク全体で発生するメモリ使用量を $M(t)$ と定義する．1 台の UE が各状態にある時に，MME ノードに発生するメモリ使用量を表 3 のように定義する．また，ある時刻 t において，接続状態にいる UE の台数を n^{c}，Connected Inactive 状態にいる UE の台数を n^{ci}，アイドル状態にいる UE の台数を n^{i} とおくと，$M(t)$ は以下の式 (8) で表せる．

$$M(t) = m_{\mathrm{MME}}^{\mathrm{c}} \cdot n^{\mathrm{c}} + m_{\mathrm{MME}}^{\mathrm{ci}} \cdot n^{\mathrm{ci}} + m_{\mathrm{MME}}^{\mathrm{i}} \cdot n^{\mathrm{i}} \tag{8}$$

5. 数 値 評 価

5.1 パラメータ設定

数値評価において用いるパラメータ設定を表 4 に示す．UE の状態遷移に伴うシグナリングメッセージ数は，文献 [2] および [10] に基づき設定した．文献 [2] および [10] に明示されていない一部の状態遷移に関しては，同様の状態遷移に基づいて決定した．UE の状態に応じた MME のメモリ使用量は，モバイルコアネットワーク機能を実装したオープンソースソフトウェアである OpenAirInterface (OAI) [13] のソースコードに基づき設定した．具体的には，MME が保持する情報およびそのサイズを，OAI のソースコードを静的解析することにより導出し，メモリ使用量を決定した．Connected Inactive 状態は OAI で実装されていなため，文献 [2] および [10] に基づくシグナリング手順を踏まえ，接続状態と同等のメモリ使用量と推定した．MME が 1 秒あたりに処理できるシグナリングメッセージ数は文献 [14] を基に設定した．文献 [14] では，OAI を用いた実験により，短時間に多数のアタッチ要求を MME が受け付けた際に，MME の処理遅延時間が急激に増加することを示している．本報告では，文献 [14] において示された，MME の処理遅延時間が急激に増加するメッセージ処理頻度に基づき，MME が 1 秒あたりに処理できるシグナリングメッセージ数を 1,200 とした．また，MME のメモリサイズは 1,000 MB とした．

表 4: パラメータ設定

Parameter	Numerical setting	Parameter	Numerical setting
$s_{\mathrm{MME}}^{\mathrm{c}\to\mathrm{c}}$	0 messages	$M^{\max}$	1000 MB
$s_{\mathrm{MME}}^{\mathrm{ci}\to\mathrm{ci}}$	0 messages	$m_{\mathrm{MME}}^{\mathrm{c}}$	17878 bits
$s_{\mathrm{MME}}^{\mathrm{c}\to\mathrm{ci}}$	0 messages	$m_{\mathrm{MME}}^{\mathrm{ci}}$	17878 bits
$s_{\mathrm{MME}}^{\mathrm{ci}\to\mathrm{c}}$	0 messages	$m_{\mathrm{MME}}^{\mathrm{i}}$	408 bits
$s_{\mathrm{MME}}^{\mathrm{ci}\to\mathrm{i}}$	5 messages	K_p	1.5
$s_{\mathrm{MME}}^{\mathrm{i}\to\mathrm{c}}$	5 messages	K_i	0
$C^{\max}$	1200 messages/s	K_d	0

表 5: UE の通信周期の分布

	1 day	2 hours	1 hour	30 minutes
UE 台数の割合	40%	40%	15%	5%

PID 制御の各定数は，ジーグラ・ニコルスの限界感度法に基づき設定した．具体的には，UE 台数を 648,000 台とし，UE の持つ通信周期を文献 [15] に基づき表 5 のように設定した場合において，提案手法を適用して Idle タイマが初期値 (600 s) から最適値 (3280 s) に収束するまでの結果に基づき決定した．

Idle タイマの制御方式として，P 制御，PI 制御および PID 制御の 3 つの制御方式を比較した結果，本報告におけるシナリオでは，P 制御が最も制御が収束するまでの遅延が短く，また，制御が安定していることが分かった．そのため，本報告では，P 制御に基づき Idle タイマを変化させた場合と，Idle タイマを固定した場合の比較を行う．P 制御が最も優れている理由としては，本報告のシナリオでは定常偏差が発生しないため，積分制御が有意に機能しなかったこと，および，離散的な負荷の変動が発生することにより，微分制御を用いると制御が不安定になることがあげられる．

また，Inactive タイマは，文献 [2] を参考に 10 s とし，UE の送信データサイズは，データ送信の際は必ず接続状態に遷移する程度に大きいと仮定した．また，シミュレーションにおける 1 タイムステップの時間幅を 1 s とした．

5.2 評価シナリオ

本報告では，UE の台数が変化する 2 つシナリオに対して，提案手法を適用した場合と，適用しない場合を比較することによって提案手法の性能評価を行う．各シナリオの初期状態における UE 台数は 240,000 台であり，UE の持つ通信周期とそれぞれの通信周期を持つ UE の割合は表 6 の通りである．この状態において，新たに特定の通信周期の UE がネットワークに接続され，一定期間後にネットワークから切り離される場合の評価を行う．

シナリオ 1 では，新たに接続する UE の台数は 320,000 台であり，それらの通信周期は 100 s である．これらの UE は全て，時刻 60,000 s から 70,000 s の間にそれぞれランダムなタイミングでネットワークに接続する．また，新たにネットワークに接続した UE は，時刻 140,000 s から 150,000 s の間にネットワークから切り離される．

シナリオ 2 では，新たに接続する UE の台数は 440,000 台であり，それらの通信周期は 6,000 s である．320,000 台の UE

表 6: UE の通信周期の分布

	10 s	20 s	30 s	···	6000 s	合計
UE 台数	400	400	400	···	400	240,000

が時刻 60,000 s から 72,000 s の間に，120,000 台の UE が時刻 100,000 s から 112,000 s の間に，それぞれランダムなタイミングでネットワークに接続する．また，新たにネットワークに接続した UE は時刻 140,000 s から 152,000 s の間にネットワークから切り離される．

また，提案手法を適用しない場合における Idle タイマは，各シナリオの初期状態において収容可能な端末台数が最大になる値で固定した．

5.3 評価結果と考察

シナリオ 1 における 10 s ごとの平均 CPU 負荷と平均メモリ使用量の変化を図 4 および図 5 にそれぞれ示す．図 4 は，提案手法を適用せず，Idle タイマを固定した場合の評価結果であり，図 5 は提案手法に基づき Idle タイマを変化させた場合の評価結果である．また，提案手法を適用した場合における，Idle タイマの変化を図 6 に示す．図中の破線は $C^{\max}$ および $M^{\max}$ を表す．

図 4 を見ると，UE が増加することで，CPU 負荷が一時的に増加していることが分かる．これは，新たに接続した UE が接続状態へ遷移する際にシグナリングが発生していること，および，新たに接続した UE は，通信周期が Idle タイマと比較して短く，アイドル状態へ遷移しないため，ネットワークへ接続後はシグナリングを発生させないためである．また，UE の増加によりメモリ使用量が 1,044MB に達していることが分かる．これは，通信周期が短い UE は，接続状態あるいは Connected Inacive 状態を維持することにより，メモリ使用量が大きくなるためである．

一方，図 5 に示すように，提案手法を適用した場合は，新たに収容可能な端末台数を可能な限り大きくするように，CPU 負荷を増加させ，メモリ負荷を削減するように制御される．具体的には，Idle タイマを減少させ，接続状態あるいは Connected Inacive 状態を維持する UE 台数を削減する．この制御により，メモリ使用量が削減されるが，その一方で，アイドル状態へ遷移する UE 台数が増加するため，CPU 負荷が増加する．その結果，CPU 負荷が 1,133 まで増加している一方，メモリ使用量を 835MB まで削減できていることを確認できる．図 6 を見ると，提案手法においては，UE の増加に伴い Idle タイマが減少していることを確認できる．

Idle タイマを固定した場合は，メモリ使用量が $M^{\max}$ を超過しているため，これ以上の UE を収容することは難しい．一方，提案手法を用いた場合は，CPU 負荷とメモリ使用量のどちらも $C^{\max}$ および $M^{\max}$ に達していないため，さらに多くの UE を問題なく収容できる．このことから，提案手法を用いることにより，CPU とメモリ間の負荷を調整し，収容可能な端末台数を増加させる効果があることを確認できる．

また，UE がネットワークから切り離されることで，Idle タイマが UE の増加前の水準に戻っていることも確認できる．

シナリオ 2 における 10 s ごとの平均 CPU 負荷と平均メモリ使用量の変化を図 7 および図 8 にそれぞれ示す．図 7 は，提案手法を適用せず，Idle タイマを固定した場合の評価結果であり，図 8 は提案手法に基づき Idle タイマを変化させた場合の評価結果である．また，提案手法を適用した場合における，Idle タイマの変化を図 9 に示す．図中の破線は $C^{\max}$ および $M^{\max}$ を表す．

図 7 および図 8 を見ると，UE の増加に伴い，CPU 負荷およびメモリ使用量が共に増加していることが分かる．CPU 負荷は，新たに接続した UE の通信周期が Idle タイマと比較して長く，データ送信毎にアイドル状態への状態遷移が発生するため増加する．メモリ使用量は，UE の増加に伴い，接続状態あるいは，Connected Inactive 状態の UE 台数が増加するため増加する．図 7 を見ると，UE が 320,000 台接続することにより，CPU 負荷が 1,041，メモリ使用量が 602MB まで増加し，その後，さらに UE が 120,000 台接続することにより，CPU 負荷が 1,253，メモリ使用量が 692MB に達している．

一方，図 8 に示すように，提案手法を適用した場合は，新たに収容可能な端末台数を可能な限り大きくするように，Idle タイマを増加させることにより，シナリオ 1 の場合とは逆に，メモリ負荷を増加させ，CPU 負荷を削減するように制御される．その結果，UE が 320,000 台接続した際の CPU 負荷が 965，メモリ使用量が 719MB であり，さらに UE が 120,000 台追加した際の，CPU 負荷が 1,166，メモリ使用量が 858MB に達していることが分かる．図 9 を見ると，提案手法においては，UE の増加に伴い Idle タイマが増加していることを確認できる．

Idle タイマを固定した場合は，UE が 440,000 台接続することで，CPU 負荷が $C^{\max}$ を超過しているため，これ以上の UE を収容することは難しい．一方，提案手法を用いた場合は，CPU 負荷とメモリ使用量のどちらも $C^{\max}$ および $M^{\max}$ に達していないため，さらに多くの UE を問題なく収容できる．このことから，シナリオ 1 と同様に，シナリオ 2 においても，提案手法を用いることにより，収容可能な端末台数を増加させる効果があることを確認できる．

6. まとめと今後の課題

本報告では，モバイルコアネットワークノードの資源利用の効率化を目的とした，端末の状態遷移に関するパラメータの適応的制御手法を提案した．具体的には，端末をアイドル状態へ遷移させる条件となる Idle タイマの値を，モバイルコアネットワークに発生する負荷に応じて，PID 制御を用いて適応的に制御する手法である．次に，UE およびモバイルコアネットワークのシミュレータを用いることで，提案手法の有効性を定量的に評価した．その結果，提案手法を用いることにより，モバイルコアネットワークの負荷に応じて，CPU 負荷とメモリ使用量を適応的に調整することが可能であることを示した．また，提案手法を用いることにより，そうでない場合は収容することが難しい台数の UE を収容可能であることを示した．具体的には，提案手法を用いることにより，シナリオ 1 においては 560,000 台，シナリオ 2 においては 680,000 台の UE 全てを収容可能であることを示した．

提案手法は，PID 制御に基づく制御手法を用いており，収束に一定の時間を要する．そのため，それよりも短い時間スケールで負荷が変動する場合には，制御が収束しない．その結果，ノードが一時的に過負荷になることが考えられる．このような課題に対して，Server Disaggregation アーキテクチャやスケールアウト/スケールイン等を用いた資源の増強を組み合わせた制御を行うことで，より効率的な資源制御を行うことを検討したい．

References

[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. S ä ily, 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. Shimizu, H. Nakazato, and H. Seshake, “Scale-Out Architecture for Service Order Processing Systems,” in *Proceedings of 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, May 2013, pp. 880–883.

[4] P. C. Amogh, G. Veeramachaneni, A. K. Rangisetti, B. R. Tamma, and A. A. Franklin, “A Cloud Native Solution

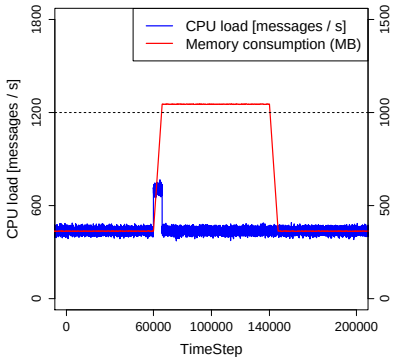

図 4: シナリオ 1 - CPU 負荷とメモリ使用量の変化 (制御なし)

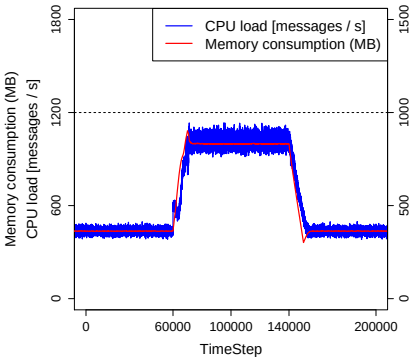

図 5: シナリオ 1 - CPU 負荷とメモリ使用量の変化 (提案手法を適用した場合)

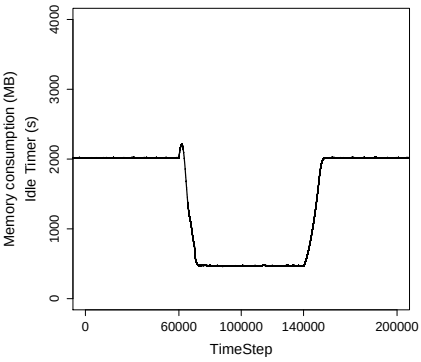

図 6: シナリオ 1 - Idle タイマの変化 (提案手法)

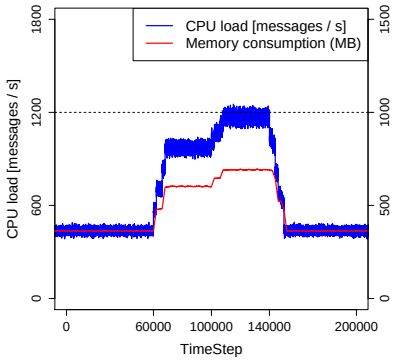

図 7: シナリオ 2 - CPU 負荷とメモリ使用量の変化 (制御なし)

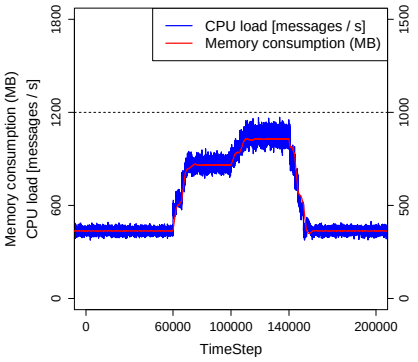

図 8: シナリオ 2 - CPU 負荷とメモリ使用量の変化 (提案手法を適用した場合)

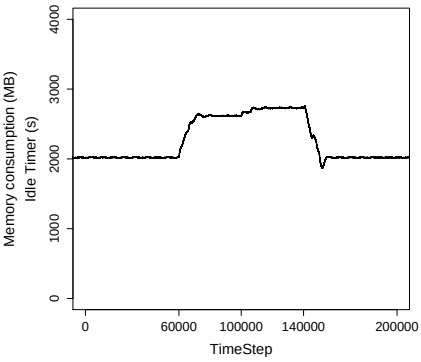

図 9: シナリオ 2 - Idle タイマの変化 (提案手法)

for Dynamic Auto Scaling of MME in LTE," in *Proceedings of 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct. 2017, pp. 1–7.

[5] I. Alawe, Y. Hadjadj-Aoul, A. Ksentini, P. Bertin, and D. Darche, "On the Scalability of 5G Core Network: The AMF Case," in *Proceedings of 2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan. 2018, pp. 1–6.

[6] Y. Ren, T. Phung-Duc, J. Chen, and Z. Yu, "Dynamic Auto Scaling Algorithm (DASA) for 5G Mobile Networks," in *Proceedings of 2016 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2016, pp. 1–6.

[7] C. H. T. Arteaga, F. Rissoi, and O. M. C. Rendon, "An Adaptive Scaling Mechanism for Managing Performance Variations in Network Functions Virtualization: A Case Study in an NFV-Based EPC," in *Proceedings of 2017 13th International Conference on Network and Service Management (CNSM)*, Nov. 2017, pp. 1–7.

[8] 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.

[9] S. Legtchenko, H. Williams, K. Razavi, A. Donnelly, R. Black, A. Douglas, N. Cheriere, D. Fryer, K. Mast, A. D. Brown, A. Klimovic, A. Slowey, and A. Rowstron, "Understanding Rack-Scale Disaggregated Storage," in *Proceedings of 9th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 17)*. Santa Clara, CA: USENIX Association, 2017. [Online]. Available: https://www.usenix.org/conference/hotstorage17/program/presentation/legtchenko

[10] 3GPP, "Study on architecture enhancements for Cellular Internet of Things (CIoT)," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 23.720, Mar. 2016, version 13.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2894

[11] 安達 智哉, 阿部 修也, 長谷川 剛, 村田 正幸, "IoT 端末を考慮したシグナリング制御によるモバイルコアノードの資源利用の効率化," **電子情報通信学会技術研究報告**, vol. 119, no. 298, pp. 47–52, 2019 年 11 月.

[12] 山本重彦, 加藤尚武, *PID* **制御の基礎と応用**. 朝倉書店, 1997. [Online]. Available: https://iss.ndl.go.jp/books/R100000002-I000002584429-00

[13] "OpenAirInterface." [Online]. Available: http://www.openairinterface.org/

[14] M. Ueno, G. Hasegawa, and M. Murata, "Experimental Evaluation of Mobile Core Networks on Simultaneous Access from M2M/IoT Terminals," in *Proceedings of 2019 International Conference on Information Networking (ICOIN)*, Jan. 2019, pp. 13–18.

[15] 3GPP, "Cellular System Support for Ultra-low Complexity and Low Throughput Internet of Things (CIoT)," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 45.820, Dec. 2015, version 13.1.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2719