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.

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

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求められる．そこで本報告ではモバイルコアネットワークノードにおける CPU とメモリ間の負荷のオフロード手法を提案する．具体的には，ネットワークの負荷に合わせて端末の状態を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のバランスを調整し，収容可能な端末台数を最大化する．端末の状態制御は，端末がデータ送受信後にアイドル状態に遷移するまでの時間を制御することで実現する．提案手法により，モバイルコアネットワークノードの資源を増強することなく，収容可能な端末台数が最大で約 150%向上することを示す．

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

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，アタッチやデタッチ等のシグナリングに関する通信や処理を実行する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される資源である．これらのノード資源は，モバイルコアネットワークにおける通信を実現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，近年は 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]．RRC Connected Inactive 状態とは，端末がネットワークから切り離された後も，端末のセッション情報をモバイルコアネットワークノードのメモリに保持している状態である．端末のセッション情報を破棄するアイドル状態とは異なり，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一方で，その端末が再び接続状態へ遷移する際に発生するシグナリング手順の一部を省略できるため，CPU 負荷の削減が可能となり，制御プレーンの輻輳の抑制が期待できる．文献 [1,2] 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ヘッドの削減が可能であることが示されている．

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の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問題がある．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 [8–13]．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突発的な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 [13] では，Server Disaggregation アーキテクチャを適用したシステムにおいて，細かい時間粒度で資源制御を行った場合，サーバ資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，モバイルコアネットワークの端末収容を対象とし，柔軟かつ効率的な，ノード資源の制御手法を提案する．具体的には，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することで，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．さらに，提案手法によるモバイルコアネットワークの性能向上に関する解析的評価を行う．具体的には，端末の通信状態に応じて発生する，モバイルコアネットワークの負荷を数学的解析によって導出し，我々の研究グループの実験結果などを用いて，定量的な評価を行う．

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．第 3. 章では，提案手法について述べる．第 4. 章では，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，定量的な評価を行うためにパラメータを設定し，提案手法がモバイルコアネットワークの収容可能な端末台数に与える影響を評価する．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アーキテクチャ

2.1 ネットワーク構成

図 1 にモバイルコアネットワークを示す．図中の各ノードは，以下に示す機能を持つ．

- **User Equipment (UE):** スマートフォンやタブレット，IoT などの端末．

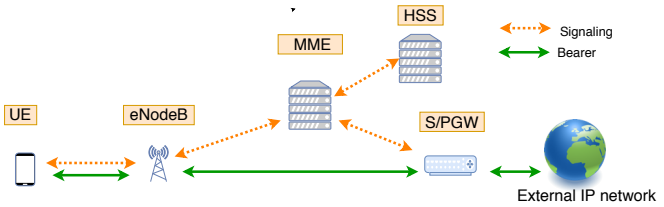

図 1　モバイルコアネットワークモデル

- **evolved NodeB (eNodeB):** UE と無線で通信を行い，MME および S/PGW 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．
- **Mobility Management Entity (MME):** UE やユーザの認証，UE の移動管理およびパケットの経路設定の制御などを行い，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．
- **Serving Gateway / Packet Data Network Gateway (S/PGW):** 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．UE が eNodeB 間を移動した際のアンカーポイントとしても機能する．
- **Home Subscliber Server (HSS):** 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．

2.2 シグナリング手順

図 1 のモバイルコアネットワークにおいて，UE が外部 IP ネットワークと通信する際，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．

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．

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．

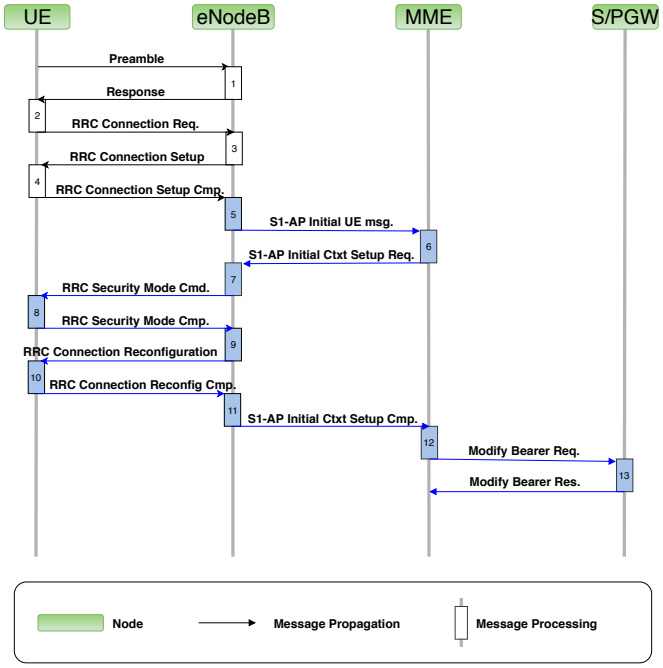

図 2　アイドル状態および Connected Inactive 状態から接続状態へ遷移する際のシグナリング手順 [2,14]

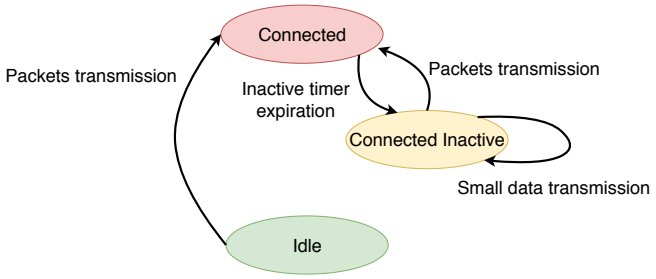

図 3　UE の状態遷移図

2.3　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．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．

3.　提 案 手 法

Connected Inactive 状態を導入することにより，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

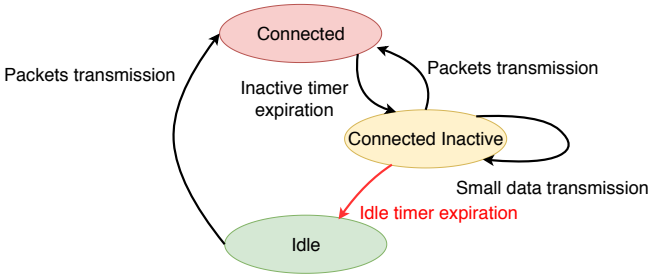

図 4　提案手法適用した場合における UE の状態遷移図

すると予想される 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，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することを提案する．図 4 に，提案手法を適用した場合における，UE の状態遷移図を示す．図 3 と比較して，Connected Inactive 状態からアイドル状態への状態遷移が追加されている．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．その際，UE のセッション情報を解放するため，メモリ使用量が削減される．一方，この遷移はシグナリングを伴うため，モバイルコアネットワークノードの CPU 負荷を増加させる．

以上のことから，Idle タイマを適切に設定することによって，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がわかる．提案手法では，Idle タイマを適応的に設定して CPU とメモリ間で負荷をオフロードすることにより，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を向上する．

4.　性 能 解 析

本章では，提案手法がモバイルコアネットワークの性能に与える影響を定量的に評価することを目的とし，数学的解析に基づくモバイルコアネットワークの負荷の導出を行い，UE の収容可能条件を明らかにする．

本章の解析においては，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する．また，対象とする UE は周期的な通信を行い，その周期は既知であると仮定する．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，制御プレーンの負荷のみに着目する．また，データ送信にかかる時間は UE の通信周期と比べて十分小さいものと仮定する．

4.1　CPU 負荷の導出

CPU 負荷は 1 秒あたりに 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

表 1　UE の状態遷移に伴い，MME が処理するシグナリングメッセージ数

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	$s_{\mathrm{MME}}^{\mathrm{i}\to\mathrm{c}}$

する．N_{UE} 台の UE がモバイルコアネットワークに存在するとし，UE の集合 $\boldsymbol{U}$ を，$\boldsymbol{U}=\{u_1,u_2,\ldots,u_{N_{\mathrm{UE}}}\}$ と定義する．Inactive タイマを T^{ci}，Idle タイマを T^{i}，ある UE u_h ($u_h\in\boldsymbol{U}$) の通信周期を T_h とする．Idle タイマが T^{i} である時の，UE u_h が 1 秒あたりに発生させる平均のシグナリングメッセージ数を $c_h(T^{\mathrm{i}})$ とする．また，UE の状態遷移に伴い，MME が処理するシグナリングメッセージ数をそれぞれ表 1 のように定義すると，$c_h(T^{\mathrm{i}})$ は T_h および T^{ci} を用いて以下の式 (1) で表せる．ここで，以下では $T^{\mathrm{i}}\geqq T^{\mathrm{ci}}$ が常に成り立つものとする．

$$c_h(T^{\mathrm{i}})=\begin{cases}\frac{1}{T_h}\cdot s_{\mathrm{MME}}^{\mathrm{c}\to\mathrm{c}} & \text{if } T_h\leqq T^{\mathrm{ci}}\\ \frac{1}{T_h}\cdot(s_{\mathrm{MME}}^{\mathrm{ci}\to\mathrm{c}}+s_{\mathrm{MME}}^{\mathrm{c}\to\mathrm{ci}})\cdot d_h & \\ \qquad+\frac{1}{T_h}\cdot s_{\mathrm{MME}}^{\mathrm{ci}\to\mathrm{ci}}\cdot(1-d_h) & \text{if } T^{\mathrm{ci}}<T_h\leqq T^{\mathrm{i}}\\ \frac{1}{T_h}\cdot(s_{\mathrm{MME}}^{\mathrm{i}\to\mathrm{c}}+s_{\mathrm{MME}}^{\mathrm{c}\to\mathrm{ci}}+s_{\mathrm{MME}}^{\mathrm{ci}\to\mathrm{i}}) & \text{otherwise}\end{cases}\tag{1}$$

ここで，d_h は UE u_h がデータ送信の際に接続状態へ遷移する確率を表す．これは，Connected Inactive 状態の UE は，送信データ量が閾値より小さいとき，接続状態へ遷移せずにデータ送信を行うことを考慮している．

MME のメッセージ処理頻度の平均を $C(T^{\mathrm{i}})$ と定義する．$C(T^{\mathrm{i}})$ は $c_h(T^{\mathrm{i}})$ を用いて以下の式 (2) で表せる．

$$C(T^{\mathrm{i}})=\sum_{h=1}^{N_{\mathrm{UE}}}c_h(T^{\mathrm{i}})\tag{2}$$

4.2　メモリ使用量の導出

Idle タイマが T^{i} である時の，UE u_h が接続状態である時間割合を $\tau_h^{\mathrm{c}}(T^{\mathrm{i}})$，Connected Inactive 状態である時間割合を $\tau_h^{\mathrm{ci}}(T^{\mathrm{i}})$，アイドル状態である時間割合を $\tau_h^{\mathrm{i}}(T^{\mathrm{i}})$ と定義する．これらは，T_h および T^{ci} を用いて以下の式 (3)，(4)，(5) で表せる．

$$\tau_h^{\mathrm{c}}(T^{\mathrm{i}})=\begin{cases}1 & \text{if } T_h\leqq T^{\mathrm{ci}}\\ \frac{T^{\mathrm{ci}}}{T_h}\cdot d_h+\frac{0}{T_h}\cdot(1-d_h) & \text{if } T^{\mathrm{ci}}<T_h\leqq T^{\mathrm{i}}\\ \frac{T^{\mathrm{ci}}}{T_h} & \text{otherwise}\end{cases}\tag{3}$$

$$\tau_h^{\mathrm{ci}}(T^{\mathrm{i}})=\begin{cases}0 & \text{if } T_h\leqq T^{\mathrm{ci}}\\ \frac{T_h-T^{\mathrm{ci}}}{T_h}\cdot d_h+\frac{T_h}{T_h}\cdot(1-d_h) & \text{if } T^{\mathrm{ci}}<T_h\leqq T^{\mathrm{i}}\\ \frac{T^{\mathrm{i}}-T^{\mathrm{ci}}}{T_h} & \text{otherwise}\end{cases}\tag{4}$$

表 2　各状態における UE 1 台当たりの 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	$m_{\mathrm{MME}}^{\mathrm{i}}$

$$\tau_h^{\mathrm{i}}(T^{\mathrm{i}})=\begin{cases}0 & \text{if } T_h\leqq T^{\mathrm{ci}}\\ 0 & \text{if } T^{\mathrm{ci}}<T_h\leqq T^{\mathrm{i}}\\ \frac{T_h-T^{\mathrm{i}}}{T_h} & \text{otherwise}\end{cases}\tag{5}$$

各状態におけるメモリ使用量をそれぞれ表 2 のように定義すると，UE u_h が MME に与えるメモリ使用量の平均 $m_h(T^{\mathrm{i}})$ は以下の式 (6) で表せる．

$$m_h(T^{\mathrm{i}})=m_{\mathrm{MME}}^{\mathrm{c}}\cdot\tau_h^{\mathrm{c}}(T^{\mathrm{i}})+m_{\mathrm{MME}}^{\mathrm{ci}}\cdot\tau_h^{\mathrm{ci}}(T^{\mathrm{i}})+m_{\mathrm{MME}}^{\mathrm{i}}\cdot\tau_h^{\mathrm{i}}(T^{\mathrm{i}})\tag{6}$$

MME に対してネットワーク全体で発生する平均的なメモリ使用量を $M(T^{\mathrm{i}})$ と定義する．$M(T^{\mathrm{i}})$ は $m_h(T^{\mathrm{i}})$ を用いて以下の式 (7) で表せる．

$$M(T^{\mathrm{i}})=\sum_{h=1}^{N_{\mathrm{UE}}}m_h(T^{\mathrm{i}})\tag{7}$$

4.3　UE の収容可能条件

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の最大値およびメモリのサイズをそれぞれ $C^{\max}$，$M^{\max}$ とする．この時，以下の式 (8) に示した 2 つの条件を同時に満たすような T^{i} が存在するならば，与えられた UE は全て収容可能である．

$$\begin{aligned}&C(T^{\mathrm{i}})\leqq C^{\max}\\ &M(T^{\mathrm{i}})\leqq M^{\max}\end{aligned}\tag{8}$$

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 3 に示す．Inactive タイマは，文献 [2] を参考に，10 s に設定した．UE の状態遷移に伴うシグナリングメッセージ数は，文献 [14] および [2] に基づき設定した．文献 [14] および [2] 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) [15] 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 [14] および [2] 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 [16] を基に設定した．文献 [16] では，OAI を用い，アタッチ要求の頻度と MME の処理時間との関係を実

表 3　パラメータ設定

Parameter	Numerical setting
T^{ci}	10 s
$s^{\mathrm{c}\to\mathrm{c}}_{\mathrm{MME}}$	0 messages
$s^{\mathrm{ci}\to\mathrm{ci}}_{\mathrm{MME}}$	0 messages
$s^{\mathrm{c}\to\mathrm{ci}}_{\mathrm{MME}}$	0 messages
$s^{\mathrm{ci}\to\mathrm{c}}_{\mathrm{MME}}$	0 messages
$s^{\mathrm{ci}\to\mathrm{i}}_{\mathrm{MME}}$	5 messages
$s^{\mathrm{i}\to\mathrm{c}}_{\mathrm{MME}}$	5 messages
$m^{\mathrm{c}}_{\mathrm{MME}}$	17878 bits
$m^{\mathrm{ci}}_{\mathrm{MME}}$	17878 bits
$m^{\mathrm{i}}_{\mathrm{MME}}$	408 bits
$C^{\max}$	1200 messages/s
$M^{\max}$	1,000 MB
d_h	1

表 4　シナリオ 2 における UE の通信周期

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

験により評価し，短時間に多数のアタッチ要求を MME が受け付けた場合，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 [16] において 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．また，UE の送信データサイズは十分大きいものとし，データ送信の際は必ず Connected 状態に遷移すると考え，$d_h = 1$ とした．

5.2　評価シナリオ

本報告では，UE の通信周期が異なる 2 つのシナリオに対して提案手法を適用した際の性能評価を行う．シナリオ 1 では，UE 台数が 500,000 台であり，UE ごとの通信周期は 10 s から 6,000 s の範囲で一様分布とする．このシナリオでは，一般的に UE の通信周期は明らかでないことを考慮して，UE の通信周期は一様分布であるという簡単な仮定をおいた場合の評価を行う．シナリオ 2 では，UE 台数が 500,000 台であり，UE ごとの通信周期は文献 [17] に基づき設定した．具体的には，UE の持つ通信周期は 1 day，2 hours，1 hour，30 minutes のいずれかであり，それぞれの通信周期を持つ UE の割合は表 4 の通りである．このシナリオでは，現実的なアプリケーションを想定した評価を行う．

また，シナリオ 2 においては，UE 台数を変化させた場合の評価も行う．

5.3　評価結果と考察

シナリオ 1 において，Idle タイマを 10 s から 6,000 s まで変化させた場合における，メッセージ処理頻度とメモリ使用量の関係を図 5 に示す．横軸がメッセージ処理頻度，縦軸がメモリ使用量である．図中の縦の破線および横の破線はそれぞれ，$C^{\max}$ および $M^{\max}$ を表す．また，グラフの点の色は Idle タイマの値に対応している．図 5 より，Idle タイマの変化に

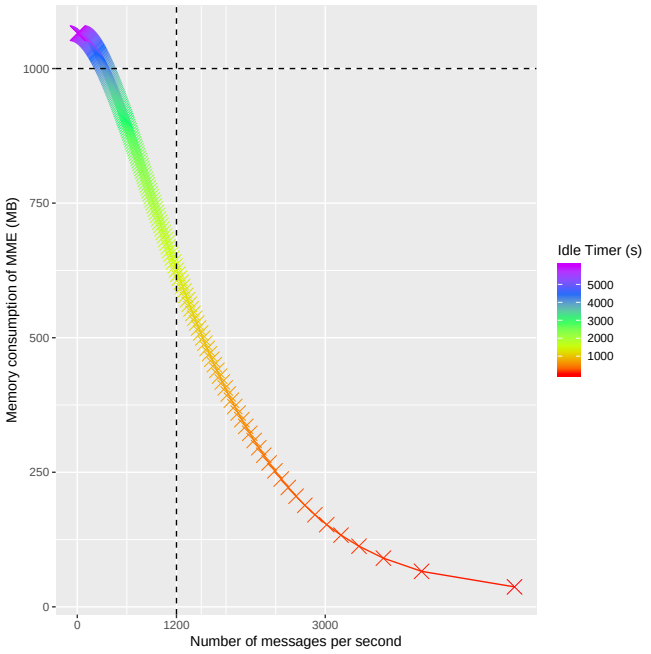

図 5　Idle タイマに対する，メッセージ処理頻度およびメモリ使用量の関係 (シナリオ 1)

応じて，メッセージ処理頻度およびメモリ使用量が変化することがわかる．具体的には，Idle タイマが増加すると，メッセージ処理頻度は減少し，メモリ使用量は増加する．これは Idle タイマの増加に伴い，データ送信後に Idle 状態へと遷移する UE が減少し，シグナリングメッセージが削減される一方，Connected Inactive 状態を維持する UE が増加し，MME が保持する情報が増加するためである．以上より，Idle タイマの適応的な設定により，CPU 負荷とメモリ使用量を互いにオフロードできることが確認できる．また，Idle タイマが 1,422 s より小さい時は，メッセージ処理頻度が $C^{\max}$ を超え，Idle タイマが 4,000 s より大きい時は，メモリ使用量が $M^{\max}$ を超える．そのため，本シナリオにおいて全 UE を収容するためには，Idle タイマを 1,422 s 以上，4,000 s 以下に設定する必要がある．

シナリオ 2 において，Idle タイマを 10 s から 864,000 s まで変化させた場合における，メッセージ処理頻度とメモリ使用量の関係を図 6 に示す．シナリオ 1 と同様に Idle タイマが大きくなるとメモリ使用量は増加している．一方，メッセージ処理頻度は段階的に減少している．これは表 4 に示すように UE の通信周期の分布が離散的であるためである．また，Idle タイマが 72,789 s より大きい時は，メモリ使用量が $M^{\max}$ を超える．そのため，本シナリオにおいて全 UE を収容するためには，Idle タイマを 72,789 s 以下に設定する必要がある．

シナリオ 1 とシナリオ 2 を比較すると，UE 台数が同じであっても，UE の通信周期の分布が異なると，MME に発生する負荷が異なり，全ての UE を収容するための Idle タイマの設定条件が異なることが確認できる．これは，IoT 端末のように通信周期の予測が難しい端末を収容する際には，適応的な Idle タイマの設定を行うことによって，MME の資源制御を行

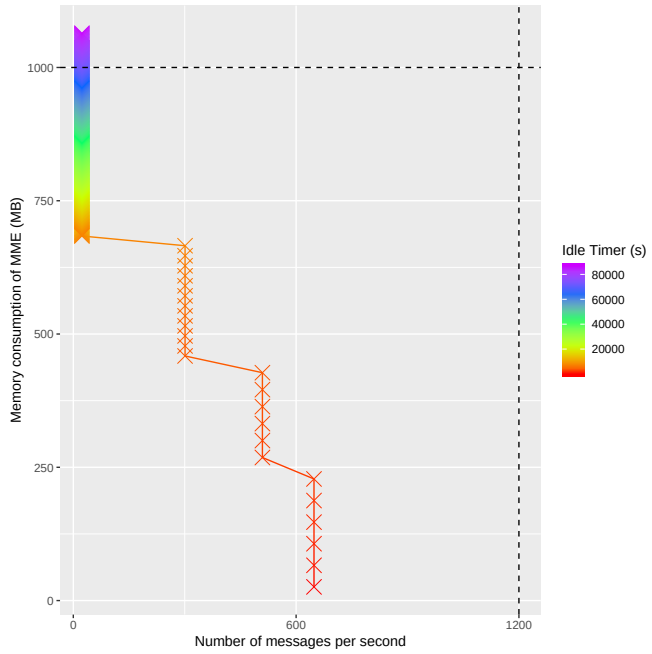

図 6　Idle タイマに対する，メッセージ処理頻度およびメモリ使用量の関係 (シナリオ 2)

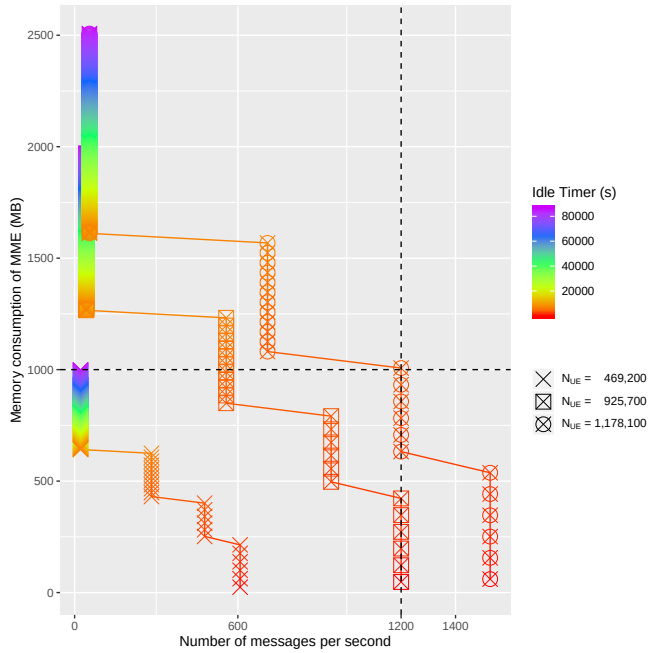

図 7　Idle タイマに対する，メッセージ処理頻度およびメモリ使用量の関係 (シナリオ 2，$N_{\mathrm{UE}} = 469,200$，$925,700$，$1,178,100$)

う必要があることを示している．

最後に，Idle タイマの設定値が収容可能な UE 台数に与える影響を評価する．UE の持つ通信周期およびその UE の割合はシナリオ 2 と同じとする．その上で，UE 台数を 469,200 台，925,700 台，および 1,178,100 台と変化させた時の評価結果を図 7 に示す．UE 台数が 469,200 台のグラフを見ると，Idle タイマが小さい場合はメッセージ処理頻度およびメモリ使用量は共にシステム容量 ($C^{\max}$，$M^{\max}$) を下回っているが，Idle タイマが 80,000 s 以上の場合，メモリ使用量が $M^{\max}$ に達している．これは，Idle タイマに 80,000 s 以上の値を設定する場合には，収容可能な UE 台数が 469,200 台であることを意味する．UE 台数が 925,700 台のグラフを見ると，Idle タイマが 1,800 s 未満の場合，メッセージ処理頻度が $C^{\max}$ に達している．これは，Idle タイマに 1,800 s 未満の値を設定する場合には，収容可能な UE 台数が 925,700 台であることを意味する．さらに，UE 台数が 1,178,100 台のグラフを見ると，Idle タイマが 1,800 s 以上 3,281 s 以下の時，メッセージ処理頻度およびメモリ使用量が $C^{\max}$ および $M^{\max}$ にそれぞれ達している．これは，Idle タイマを 1,800 s 以上 3,281 s 以下に設定する場合には，収容可能な UE 台数が 1,178,100 台であることを意味する．

以上の評価より，Idle タイマの値によって収容可能な UE 台数が大きく変化することがわかる．具体的には，シナリオ 2 においては，Idle タイマを適切に設定することにより，Idle タイマを 80,000 s 以上に設定した場合および 1,800 s 未満に設定した場合と比較して，収容可能な UE 台数がそれぞれ 27%および 151%向上する．

提案手法を用いない場合においては，Connected Inactive 状態からアイドル状態へ状態遷移が発生しない．これは，Idle タイマを無限大に設定した場合と透過であり，上述の結果より，収容可能な UE 台数は 469,200 台である．したがって，シナリオ 2 においては，提案手法を用いて Idle タイマを適切に設定することにより，提案手法を用いない場合と比較して，収容可能な UE 台数が 151%向上するといえる．

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，端末の状態遷移に関するパラメータ制御による，CPU およびメモリ間の負荷のオフロード手法を提案した．具体的には，Connected Inactive 状態の端末がアイドル状態へ遷移する条件として Idle タイマを導入し，端末の状態および状態遷移の発生頻度を制御する．次に提案手法の評価を行うために，数学的解析に基づいて CPU 負荷およびメモリ使用量を導出した．最後に，端末台数や通信周期が異なるいくつかのシナリオにおいて提案手法を適用した場合の評価を行い，提案手法が CPU 負荷およびメモリ使用量に与える影響を明らかにした．その結果，提案手法を用いることにより，CPU 負荷とメモリ使用量を互いにオフロードすることができ，資源利用が効率化されることを確認した．さらに，特定のシナリオにおいては，提案手法を用いることにより，収容可能な端末台数が最大 151%向上することを確認した．

今後の課題として，Idle タイマの動的かつ適応的な制御手法の検討が挙げられる．本報告においては，端末台数および通信周期が明らかかつ不変であり，適切な Idle タイマの値が導出できるという前提の下で提案手法の評価を行った．しかし，一般的に，端末台数および通信周期は未知であり時間的に変動する．そのような状況においても，Idle タイマを適切に制御できる手法を検討することにより，より現実的なシナリオにおける評価が可能になる．また，Server Disaggregation アー

キテクチャやスケールアウト/スケールイン等と提案手法を組み合わせて，より効率的な資源制御を行うことも検討したい．

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 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] “Intel’s Disaggregated Server Rack,” Moor Insights Strategy, Technical Report (TR), Aug. 2013.

[10] C. Devaki and L. Rainer, “Enhanced Back-off Timer Solution for GTP-C Overload Control,” Feb. 2016. [Online]. Available: http://www.freepatentsonline.com/y2016/0057652.html

[11] B. Abali, R. J. Eickemeyer, H. Franke, C. Li, and M. Taubenblatt, “Disaggregated and Optically Interconnected Memory: When will it be cost effective?” *CoRR*, vol. abs/1503.01416, 2015. [Online]. Available: http://arxiv.org/abs/1503.01416

[12] “Disaggregated Servers Drive Data Center Efficiency and Innovation,” Intel Corporation, Technical Report (TR), Jun. 2017.

[13] 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

[14] 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

[15] “OpenAirInterface.” [Online]. Available: http://www.openairinterface.org/

[16] 上野真生, 長谷川 剛, 村田正幸, “多数の M2M/IoT 端末からの集中アクセスを考慮したモバイルコアネットワークの実験評価,” in *Proceedings of* 電子情報通信学会技術研究報告 *(NS2018-226)*, Mar. 2019.

[17] 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