Osaka Univ.

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

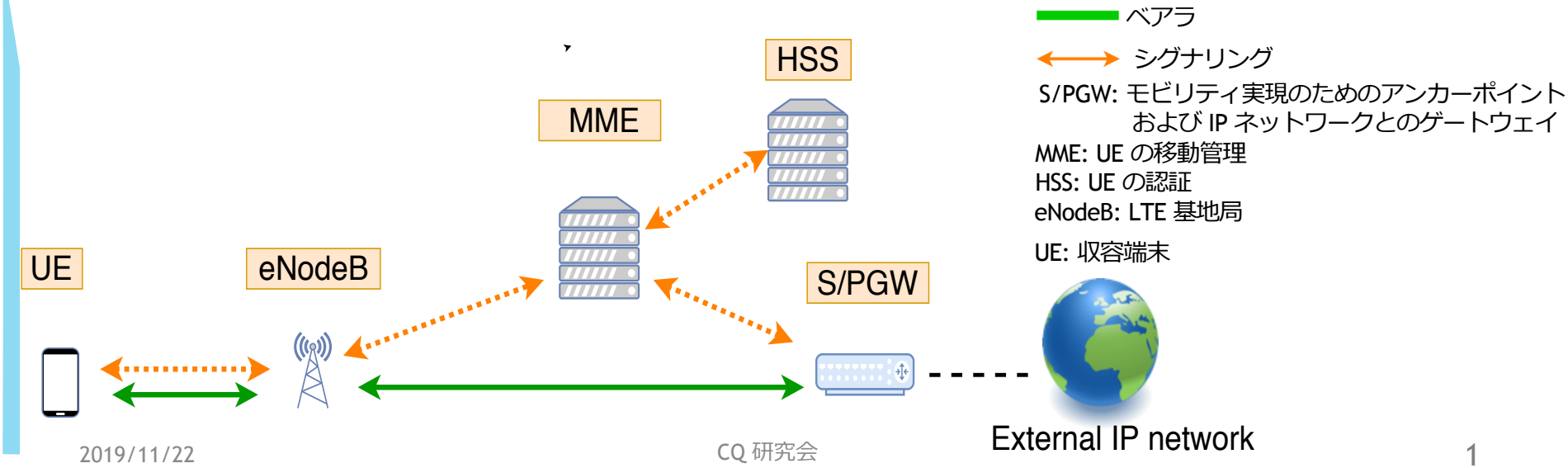
安達智哉† 阿部修也† 長谷川剛†† 村田正幸†

† 大阪大学大学院情報科学研究科

†† 東北大学電気通信研究所

モバイルコアネットワーク

- 今後の移動体通信網の利用形態として M2M/IoT 通信が着目されている
- M2M/IoT 端末を大量に収容することにより、モバイルコアネットワーク内の制御プレーンの負荷が増大
  - 収容端末がデータを送信する前に、ベアラと呼ばれる論理的なデータの伝送路を端末毎に確立する
  - ベアラの確立のために、多数の制御メッセージが伝搬、処理される



研究背景

- モバイルネットワーク事業者は、収容端末台数や接続頻度に応じて、ネットワークに資源を割り当てる必要がある
- しかし、IoT 端末は近年急激に増加しており、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、RRC Connected Inactive と呼ばれる、新たなアーキテクチャが検討されている
  - 端末情報をメモリに一時的に保存することにより、端末の通信開始時のシグナリング手順を削減する
  - CPU およびメモリに与える負荷のバランスが従来とは大きく変化すると考えられる

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

研究目的

- 柔軟かつ効率的なノード資源の制御手法を提案
  - ネットワークの負荷に合わせて端末の状態を制御
  - CPU 負荷およびメモリ使用量のバランスを調整

- 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評価
  - 端末の通信周期が異なる2つのシナリオで提案手法を評価
  - アイドル状態へ遷移するまでの時間(Idle タイマ)と収容可能な端末台数の関係を評価

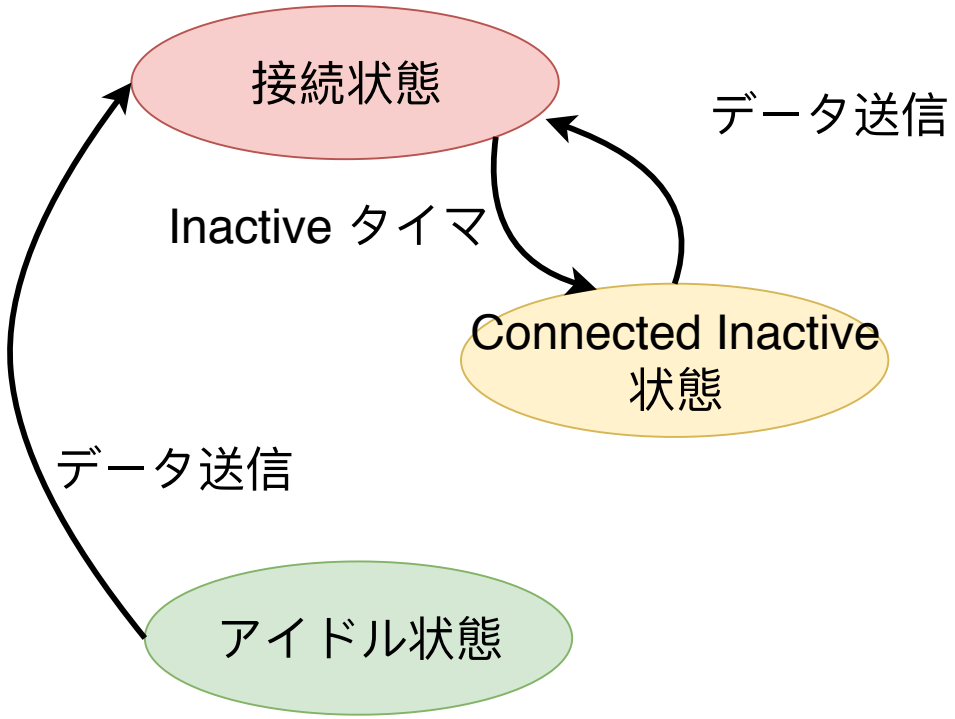
目次

端末の状態

- 端末は接続状態、アイドル状態、および Connected Inactive 状態という3つの状態を持つ
  - 接続状態
    - 全てのベアラが確立されている状態
    - ユーザデータの送受信が可能
  - アイドル状態
    - ベアラを確立していない状態
    - ユーザデータの送受信を行うためには、接続状態へ遷移する必要がある
  - Connected Inactive 状態
    - 端末はネットークから切り離されているが、モバイルコアネットワークは端末のセッション情報を保持し、ベアラを確立している状態
    - ユーザデータの送受信を行うためには、接続状態へ遷移する必要がある

端末の状態遷移

- 端末はデータ送信のタイミングで接続状態へ遷移
- データ送信が完了した端末は、Inactive タイマを起動
  - Inactive タイマが切れるまでに次のデータの送受信が発生しなければ、Connected Inactive 状態へ遷移する

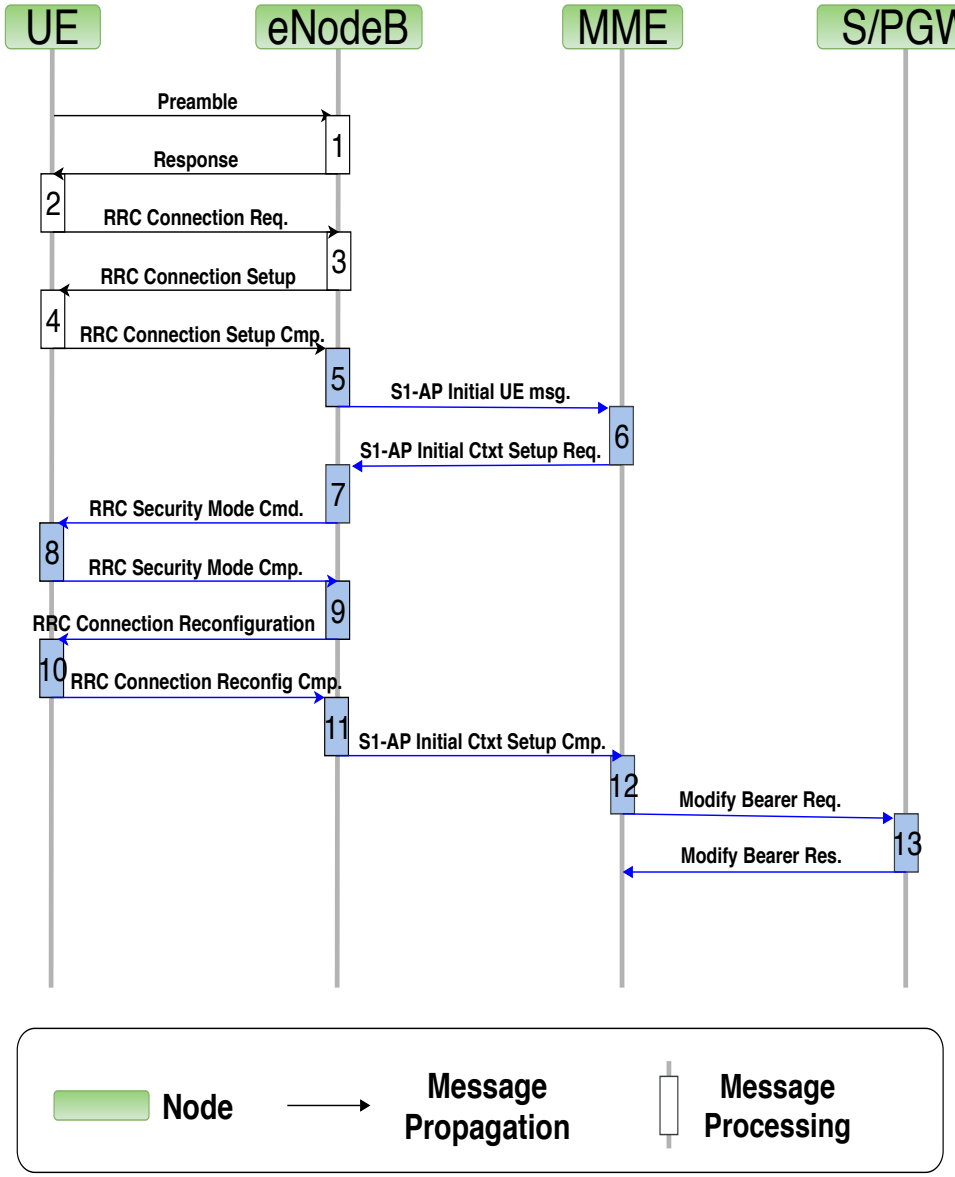


端末の状態遷移図

シグナリング手順

- ベアラを確立するために、モバイルコアネットワークの各ノードで実行される一連の処理
- 端末がデータ送信を行うためには、右図に示したシグナリング手順を実行し、接続状態へ遷移する必要がある
- Connected Inactive 状態から接続状態へ遷移する際は、右図の青色で示したシグナリング手順を省略することが可能

CPU 負荷が削減される



接続状態に遷移する際のシグナリング手順

提案手法

先行研究

Connected Inactive 状態を導入することにより、
CPU とメモリの負荷が変化する

- シグナリング手順が削減されるため、 CPU 負荷が減少
- 端末のセッション情報を保持するため、メモリ使用量が増加

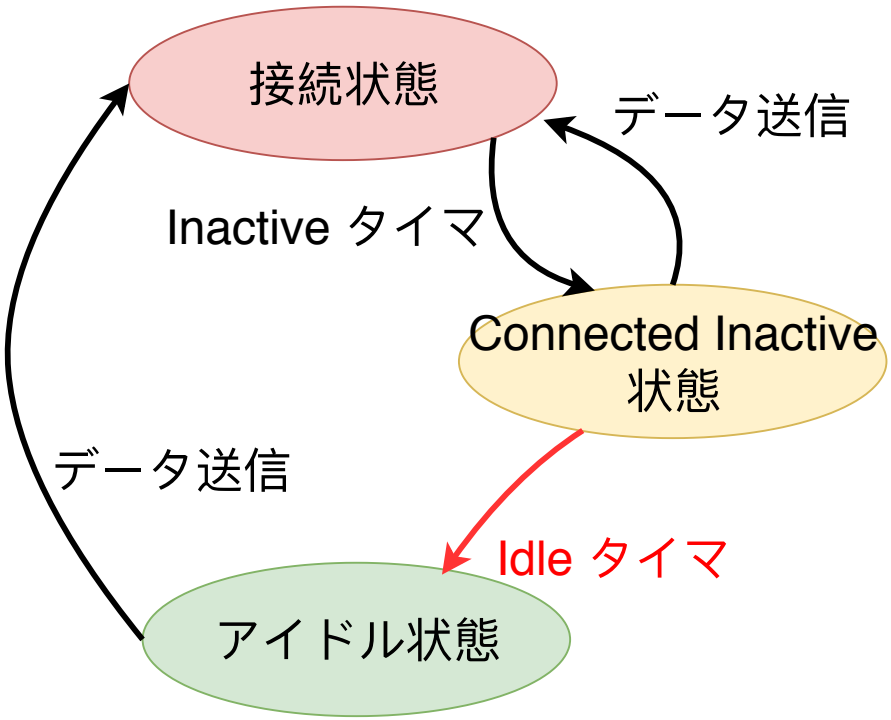
提案手法

Connected Inactive 状態の端末をアイドル状態へ
遷移させる新たな状態遷移を導入

- 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

提案手法

- Connected Inactive 状態からアイドル状態への状態遷移を Idle タイマによって制御する
- データ送信が完了した端末は、Idle タイマを起動
  - Idle タイマが切れるまでに次のデータの送信が発生しなければ、アイドル状態へ遷移する
  - Idle タイマが長い場合
    - メモリ使用量が減少
  - Idle タイマが短い場合
    - CPU 負荷が減少
- Idle タイマに適切な値を設定することにより CPU とメモリ負荷を調整可能



端末の状態遷移図 (提案手法)

性能解析

- 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に発生する負荷に着目する
- 評価指標
  - CPU 負荷　　：1 秒あたりに MME が処理するシグナリング数(メッセージ処理頻度)
  - メモリ使用量　　：MME が保持する端末のセッション情報
  - 収容可能な端末台数

パラメータ設定

- 端末の状態遷移に伴うシグナリングメッセージ数は
  先行研究に基づき、表1ように設定した
- 端末の状態に応じた MME のメモリ使用量は、OpenAirInterface の
  ソースコードに基づき、表2のように設定した
- MME が1秒間に処理可能なシグナリング数は、先行研究に基づき
  1,200 とし、MME のメモリサイズは 1,000 MB とした

表 1 : 端末の状態遷移に伴うシグナリング発生数

状態遷移		シグナリング
遷移元	遷移先	メッセージ数
接続状態	Connected Inactive 状態	0
Connected Inactive 状態	接続状態	0
Connected Inactive 状態	アイドル状態	5
アイドル状態	接続状態	5

表 2 : 各状態におけるメモリ使用量

状態	メモリ消費量 (bit)
接続状態	17878
Connected Inactive 状態	17878
アイドル状態	408

評価シナリオ

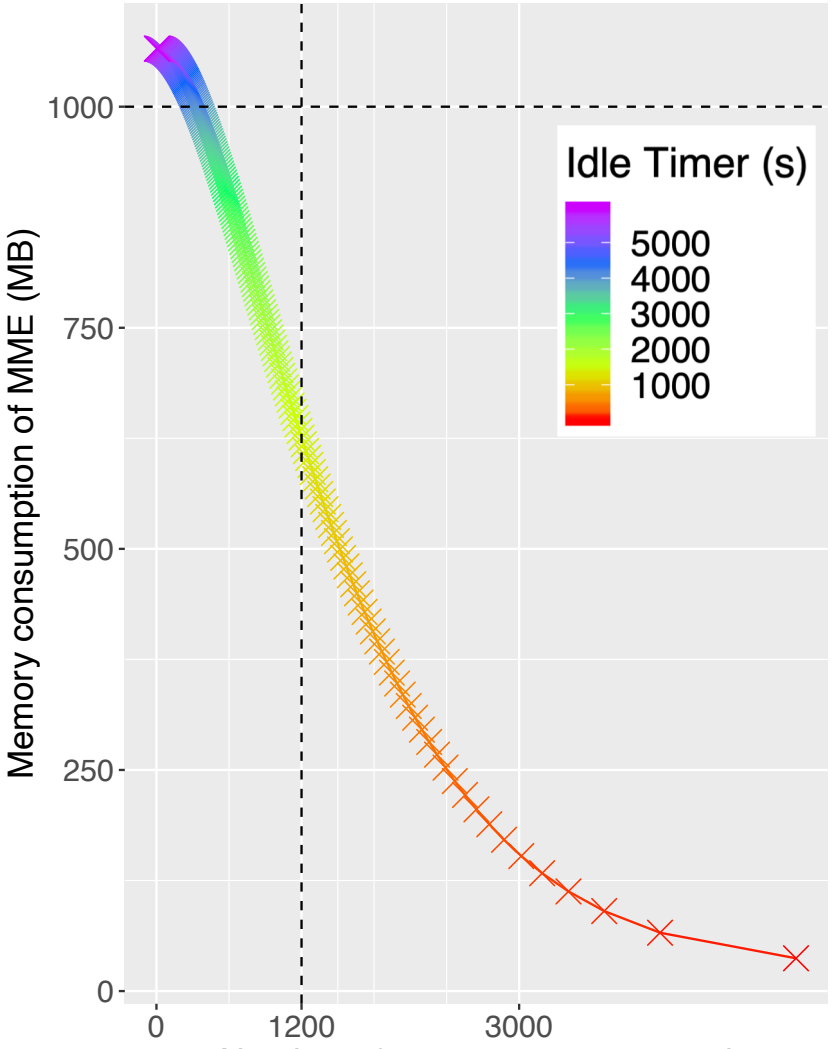
- 以下の2つのシナリオに対して評価を行う
  - シナリオ 1
    - 端末台数 (N_{UE}) ：500,000 台
    - 端末ごとの通信周期は10 s から 6,000 s の範囲で一様分布に従う
  - シナリオ 2
    - 端末台数 (N_{UE}) ：500,000 台
    - 端末ごとの通信周期は以下の表に従う

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

端末の通信周期の分布 (3GPPより)

評価結果 (シナリオ１)

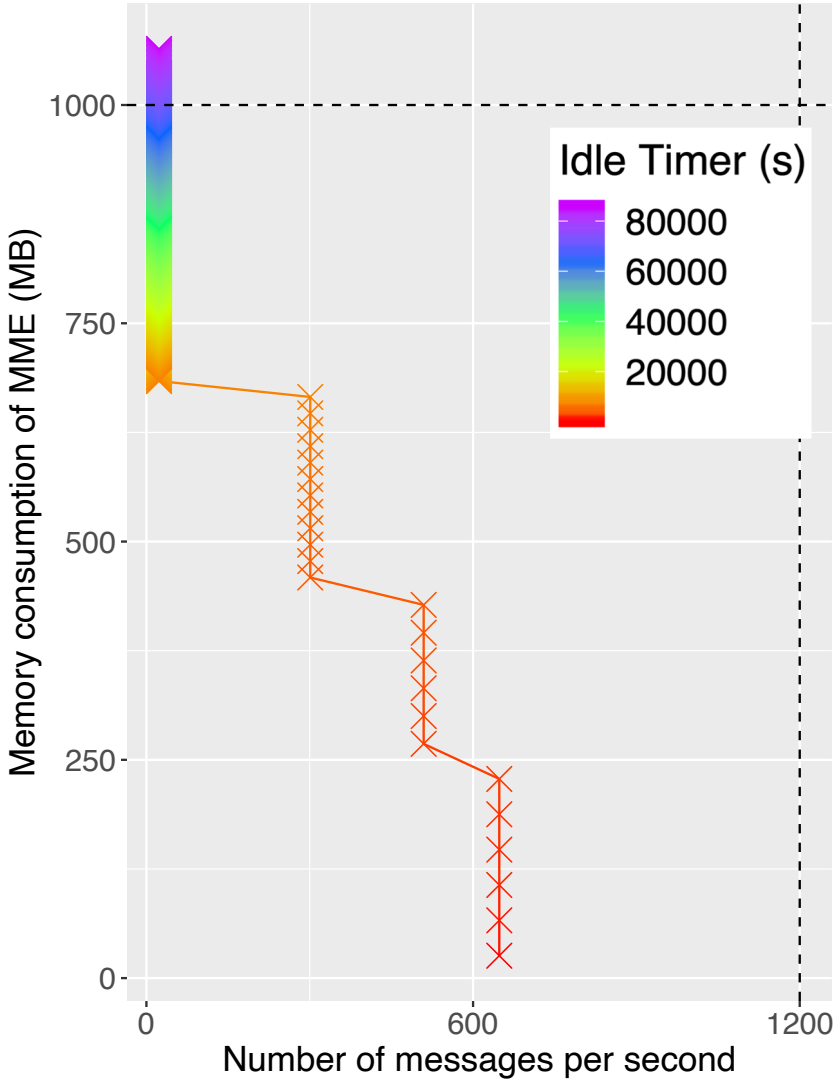
- Idle タイマの増加に伴い、
  メッセージ処理頻度は減少し、
  メモリ使用量は増加
  - Idle タ イマの増加に伴い、
    アイドル状態へと遷移する端末が
    減少し、Connected Inactive 状態を
    維持する端末が増加するため
- Idle タイマが1,422 s 以下の場合
  - メッセージ処理頻度が1,200を超える
- Idle タ イマが4,000 s 以上の場合
  - メモリ使用量が 1,000 MB を超える
- Idle タイマを1,422 s ～ 4,000 s の間
  に設定することにより、
  全端末を収容できる



Idle タイマに対する、メッセージ処理頻度
とメモリ使用量の関係

評価結果 (シナリオ 2)①

- Idle タイマの増加に伴い、
  メッセージ処理頻度は段階的に減少
  - 端末の通信周期の分布が
    離散的であるため
- Idle タイマが 72,789 s 以上の場合
  - メモリ使用量が 1,000 MB を超える
- Idle タイマを 72,789 s より小さく
  定すると、全端末を収容できる



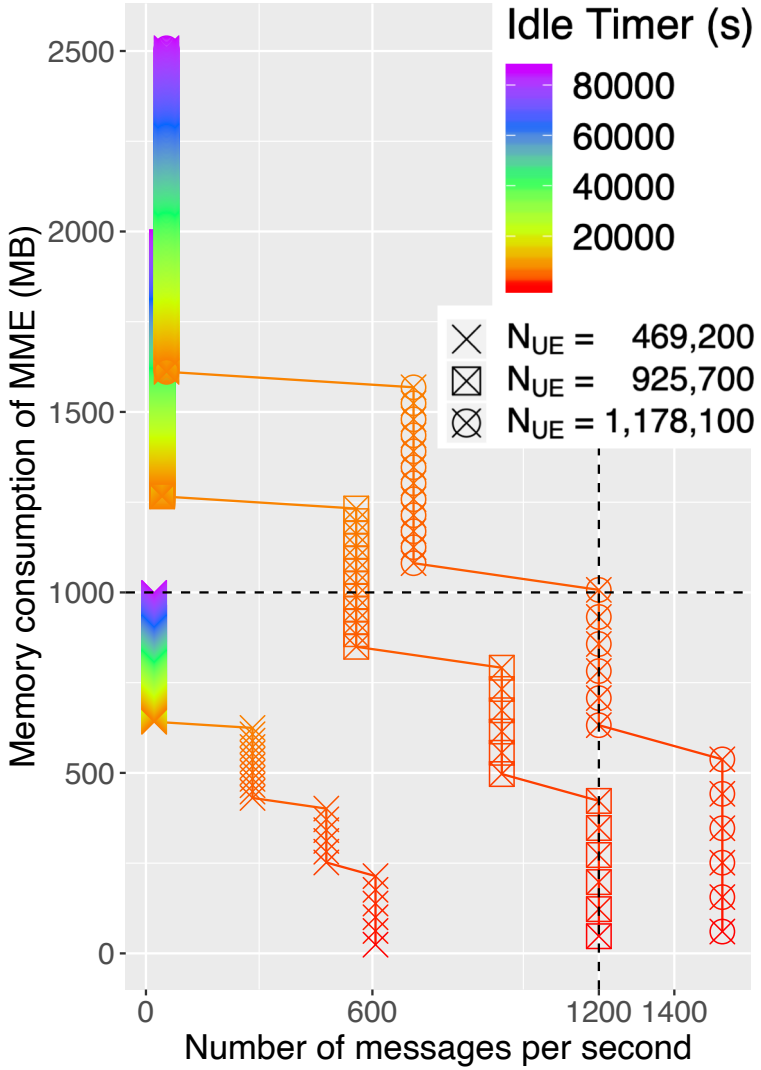
Idle タイマに対する、メッセージ処理頻度
とメモリ使用量の関係

評価結果 (シナリオ 2)②

- 端末台数を 469,200 台、925,700 台、1,178,100 台と変化させた場合の評価結果を右図に示す
- 右図より、Idle タイマの設定値と収容可能な端末台数に関する以下の結果を得る

Idle タイマ	収容可能な端末台数
80,000 s 以上	469,200 台
1,800 s 以下	925,700 台
1,800 s 以上 & 3,281 s 以下	1,178,100 台

Idle タイマの設定値と収容可能な端末台数の関係



Idle タイマに対する、メッセージ処理頻度とメモリ使用量の関係

まとめ

- Idle タイマを制御することにより、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る
  - 資源利用が効率化される
- 特定のシナリオにおいては、Idleタイマに適切な値を設定することにより、収容可能な端末台数が最大151%向上
- 端末の通信周期の分布が異なると、
  Idle タイマの設定条件が変化する
  - 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 タイマの制御を行う必要がある

今後の課題

- Idle タイマの動的かつ適応的な制御手法の検討
  - 一般に、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
  - このような状況においても、Idle タイマを適切に制御きる手法を検討し、より現実的なシナリオにおける評価を行う
- 既存の資源管理手法と提案手法の組み合わせを検討
  - 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る
  - より効率的な資源制御が可能であると考えられる