

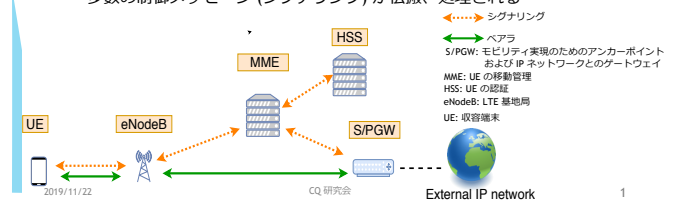
IoT 端末を考慮したシグナリング制御による モバイルコアノードの資源利用の効率化

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の輻輳

- ▶ 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
 - ▶ 端末の状態遷移に伴い、ベアラ (論理的なデータの伝送路) を確立または解放する
 - ▶ ベアラの確立や端末のモビリティ管理などの際に、多数の制御メッセージ (シグナリング) が伝搬、処理される



1

研究背景

- ▶ 収容端末台数や通信頻度などに依存する需要に応じて、モバイルコアノードに資源を割り当てる必要がある
 - ▶ CPU 資源
 - ▶ アタッチやデタッチ等のシグナリングに関するパケット送受信や処理を実行するために必要
 - ▶ メモリ資源
 - ▶ 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
- ▶ 資源の枯渇は接続失敗や遅延の原因となる
- ▶ 大量の IoT 端末を収容することにより、CPU 負荷やメモリ使用量が増大、かつ大きく変動することが予想される
 - ▶ 緊急通信やイベントドリブな通信など、従来の端末とは異なる通信特性を持つ
 - ▶ 資源需要の予測は難しくなっている

2019/11/22

CQ 研究会

2

研究目的

- ▶ M2M/IoT 端末を効率的に収容するための、柔軟なノード資源の制御手法を提案
 - ▶ モバイルコアネットワークの負荷に応じて端末の状態遷移の条件を制御
- ▶ 端末の状態および状態遷移に伴う、CPU 負荷およびメモリ使用量を調整
- ▶ 端末収容能力の拡大効果を数学的解析により評価
 - ▶ 3GPP の標準や我々の過去の実験結果に基づいた解析

2019/11/22

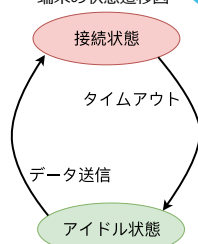
CQ 研究会

3

端末の状態 (現在のアーキテクチャ)

- ▶ 端末は接続状態およびアイドル状態の 2 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
 - ▶ 一部のベアラが解放されている状態
 - ▶ 端末は消費電力を抑えることが可能
- ▶ 端末はデータ送金のタイミングで接続状態へ遷移
 - ▶ データ送金が完了した端末は、一定期間後にアイドル状態へ遷移

端末の状態遷移図



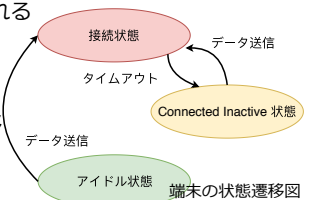
2019/11/22

CQ 研究会

4

端末の状態 (Connected Inactive 状態の導入)

- ▶ 3GPP TR 38.804 等で、Connected Inactive という新たな状態を追加することが検討されている
 - ▶ 端末はネットワ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が、その際の**シグナリング手順が削減される**
- ▶ 送信が完了した端末は、一定期間後にアイドル状態ではなく**Connected Inactive 状態へ遷移**



2019/11/22

CQ 研究会 5

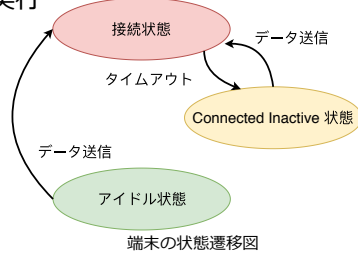
端末の状態遷移とノード負荷

- ▶ 端末が接続状態および Connected Inactive 状態にいる時、その端末のセッション情報を保持

➡ **メモリ使用量に影響**

- ▶ 端末が状態遷移する際には、**シグナリング手順**を実行

➡ **CPU負荷に影響**



2019/11/22 CQ 研究会 6

端末の状態遷移図

シグナリング手順

- ▶ 端末が状態遷移する際に、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

- ▶ 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が省略される

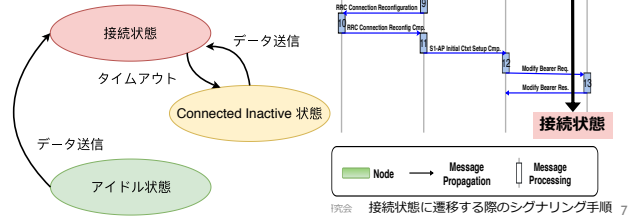


図1 接続状態に遷移する際のシグナリング手順 7

シグナリング手順

- ▶ 端末が状態遷移する際に、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

- ▶ 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が省略される

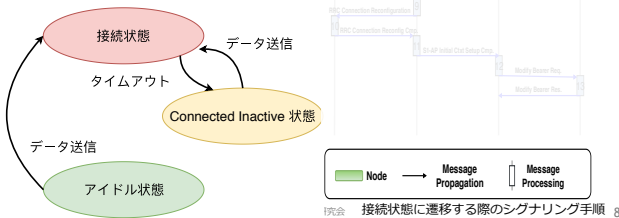


図2 接続状態に遷移する際のシグナリング手順 8

課題

- ▶ Connected Inactive 状態を導入することにより、メモリ使用量が增大

- ▶ 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

特に、今後増加すると予想される IoT 端末には通信周期の大きなものがあり、低頻度のデータ送信に対して長期間 Connected Inactive 状態を維持すると、メモリが浪費される

- ▶ 稼働するサーバやインスタンスの台数を増やすと、オーバプロビジョニングが発生する

- ▶ サーバやインスタンス一台あたりの資源構成は固定
- ▶ 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が供給される

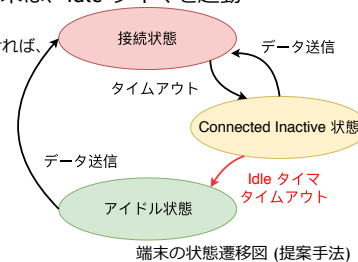
2019/11/22

CQ 研究会

9

提案手法 (1)

- ▶ 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
- ▶ 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
- ▶ 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送信が発生しなければ、アイドル状態へ遷移



2019/11/22 CQ 研究会 10

端末の状態遷移図 (提案手法)

提案手法 (2)

- ▶ Idle タイマの大きさを変化させ、Connected Inactive 状態とアイドル状態の端末台数を調整

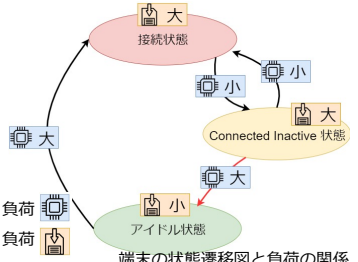
➡ **CPU とメモリにかかる負荷が調整され、収容可能な端末台数の増加が期待できる**

- ▶ Idle タイマが大きい場合、Connected Inactive 状態を維持する端末が増加

- ▶ CPU 負荷が減少
- ▶ メモリ使用量が増加

- ▶ Idle タイマが小さい場合、アイドル状態へ維持する端末が増加

- ▶ CPU 負荷が増加
- ▶ メモリ使用量が減少



2019/11/22

CQ 研究会

11

端末の状態遷移図と負荷の関係

性能解析

▶ 提案手法の有効性を定量的に評価

- ▶ 現実的なスケールで負荷の調整が可能か？

▶ 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のサイズ
- ▶ 収容可能な端末台数

モバイルコアネットワーク内のボトルネックはMME
 以下では、MME に発生する負荷に着目

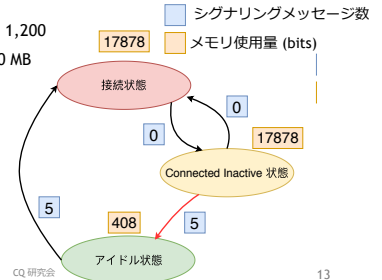
2019/11/22

CQ 研究会

12

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 3GPP 標準および我々の過去の実験結果に基づき設定
- ▶ 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設定
- ▶ MME ノードの性能は、我々の過去の実験結果に基づき設定
 - ▶ 1秒間に処理可能なシグナリングメッセージ数: 1,200
 - ▶ MME のメモリサイズ: 1,000 MB



2019/11/22

CQ 研究会

13

評価シナリオ

▶ 以下の2つのシナリオに対して評価

▶ 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

提案手法の基本的な特性を確認

▶ シナリオ 2

- ▶ 端末台数 (N_{UE}): 469,200 台、925,700 台、1,178,100 台
- ▶ 端末ごとの通信周期

端末の通信周期の分布 (3GPP TR45.820 より)

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

現実的なモデルにおける提案手法の有効性を確認

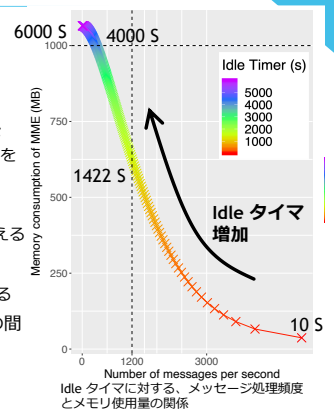
2019/11/22

CQ 研究会

14

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、全端末を収容可能



2019/11/22

CQ 研究会

15

評価結果 (シナリオ 2)

- ▶ Idle タイマを**最大**に設定し、メモリに負荷をオフロードした場合、**469,200**台の端末を収容可能
- ▶ Idle タイマを**最小**に設定し、CPU に負荷をオフロードした場合、**925,700**台の端末を収容可能
- ▶ Idle タイマに**適切な値**を設定し、CPU とメモリの負荷を調整することにより、**1,178,100**台の端末を収容可能

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

2019/11/22

CQ 研究会

16

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可能
 - ▶ 資源利用が効率化される
- ▶ 特定のシナリオにおいて、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、収容端末台数を最大化する設定値は変化
 - ▶ 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

2019/11/22

CQ 研究会

17

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
 - ▶ 未知かつ時間的に変動する端末台数および通信頻度に応じた Idle タイマの動的制御
- ▶ 既存の資源管理手法と提案手法の組み合わせの検討
 - ▶ 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た制御