

# ミーティング資料

安達智哉

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

2019 年 2 月 28 日

## 1 提案手法

### 1.1 ベアラのタイムアウト時間の設定

提案手法では、ベアラのタイムアウト時間を 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。ベアラのタイムアウト時間とは、UE がデタッチ要求を発生させた後、実際にコアノード側でデタッチ処理を実行するまでの待機時間である。この値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ベアラのタイムアウト時間を計算によって導出する。処理のフローを図 1 に示す。常に CPU とメモリのリソース負荷を監視し、CPU が過負荷であればベアラのタイムアウト時間を増加させ、反対にメモリが過負荷であればベアラのタイムアウト時間を減少させる。

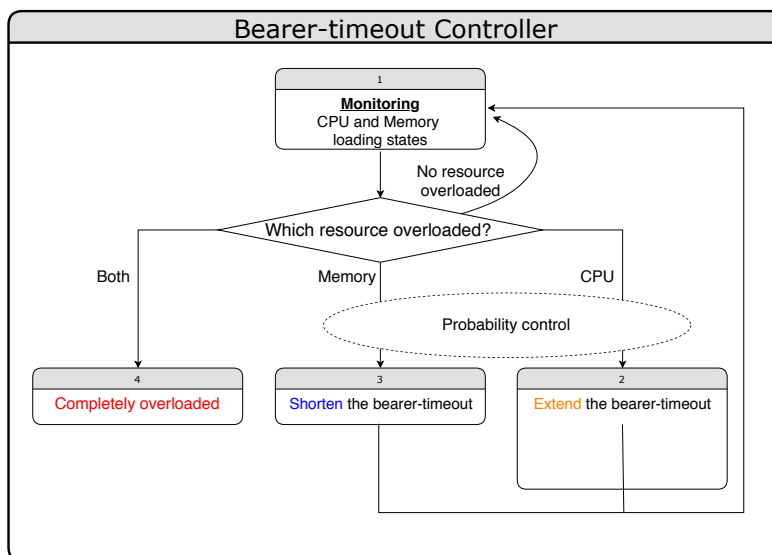


図 1: The proposed method

### 1.2 Probability control

UE の通信特性が鋭いピークを持つ場合 (特定の周期でデータ送信を行う UE が非常に多い場合)、ベアラのタイムアウト時間の設定が不安定になる可能性がある (2)。そのため、場合によってはベ

アラのタイムアウト時間の設定方法を拡張する必要がある。具体的には、ベアラのタイムアウト時間に幅を持たせ、確率的にデタッチを実行する方法がある。

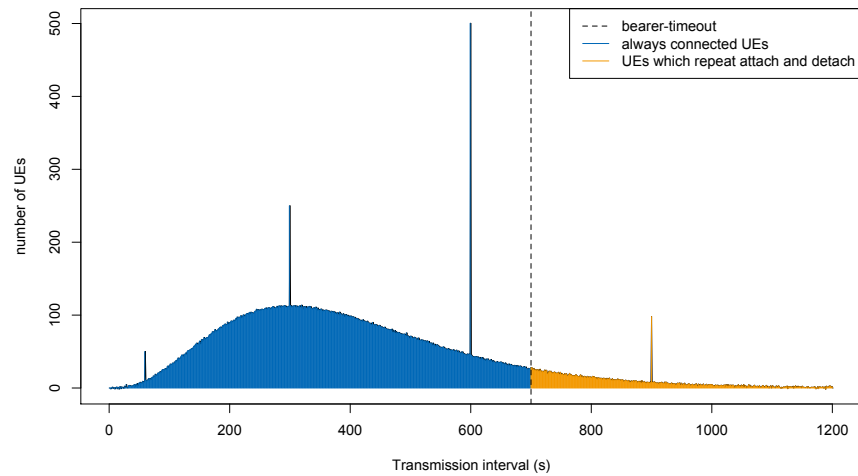


図 2: UE の通信特性が鋭いピークも持つ場合のイメージ

## 2 解析モデル

### 2.1 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 3 に評価対象のネットワークを示す。図 3 は以下のノードから構成される。

- UE
- eNodeB
- MME
- S/PGW
- HSS

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。

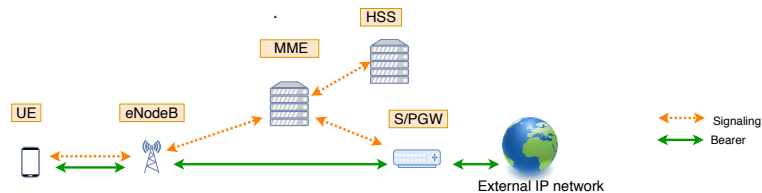


図 3: LTE/EPC ネットワークモデル

### 2.2 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。例えば、10 分や 30 分の周期を持つ UE は、9 分や 29 分の周期を持つ UE と比べて多いと予想される。

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。

### 2.3 サーバ資源の割り当て

今後検討予定である。

## 2.4 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の通信特性とベアラのタイムアウト時間に基づきEPCの負荷を算出する。

EPCの負荷の算出のイメージ図を図4に示す。UEの通信特性とベアラのタイムアウト時間を入力とし、EPCの負荷を出力とする。EPCの負荷の導出のために、統計処理、待ち行列理論および上野さんの実験結果を用いる。

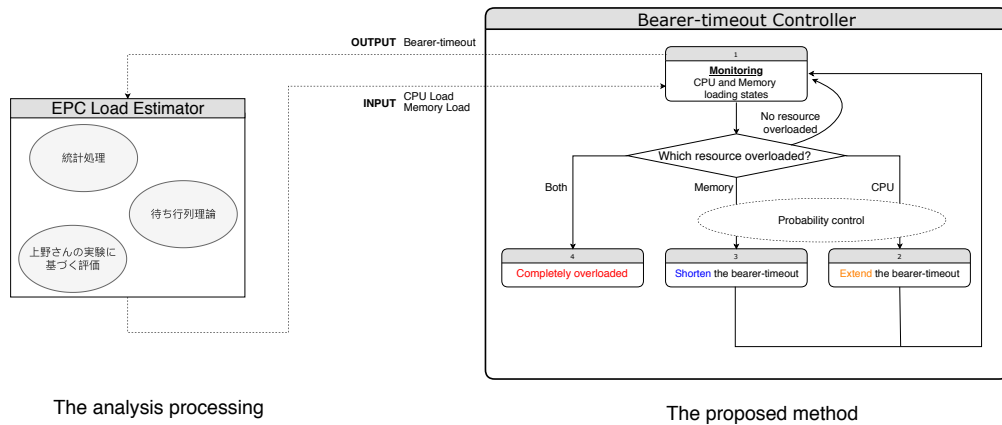


図 4: 解析環境

### 2.4.1 統計処理

この項では、UE の通信特性とベアラのタイムアウト時間から、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。ベアラのタイムアウト時間を  $T_{bearer}$ 、 $u_h$  の通信周期を  $T_h$ 、アタッチ処理およびデタッチ処理をそれぞれ一回実行する際に発生するシグナリングの数を  $n_{attach}$  および  $n_{detach}$  とすると、 $s_h$  は以下の式 (1) で表せる。ここで留意すべきは、 $s_h$  が  $T_{bearer}$  以下であるような UE は、一度アタッチ処理を実行してネットワークに接続すると、常時接続状態となり、アタッチ処理およびデタッチ処理のどちらも発生しないため、 $s_h$  は 0 と定義したことである。

$$s_h = \begin{cases} 0 & \text{if } T_h \leq T_{bearer} \\ \frac{1}{T_h} \cdot (n_{attach} + n_{detach}) & \text{otherwise} \end{cases} \quad (1)$$

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、ベアラの平均接続数を求める。ある時刻において、UE  $u_h$  が接続状態である確率を  $b_h$  と定義する。 $b_h$  は  $T_h$  および  $T_{bearer}$  を用いて以下の式 (3) で表せる。ここでは、 $s_h$  が  $T_{bearer}$  以下であるような UE は、常時接続状態となるため、 $s_h$  は 1 と定義した。

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

ベアラの平均接続数を  $B$  と定義する。 $B$  は  $b_h$  を用いて以下の式 (4) で表せる。

$$B = \sum_{h=1}^{N_{UE}} b_h \quad (4)$$

### 2.4.2 待ち行列理論

CPU にかかる負荷は、OpenAirInterface(OAI)[1] のソースコードの命令文数に基づく待ち行列理論で求める。アタッチ処理およびデタッチ処理、ベアラのサスペンド処理にかかる時間を待ち行列理論によって導出する。

### 2.4.3 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 5、図 6、図 7 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。

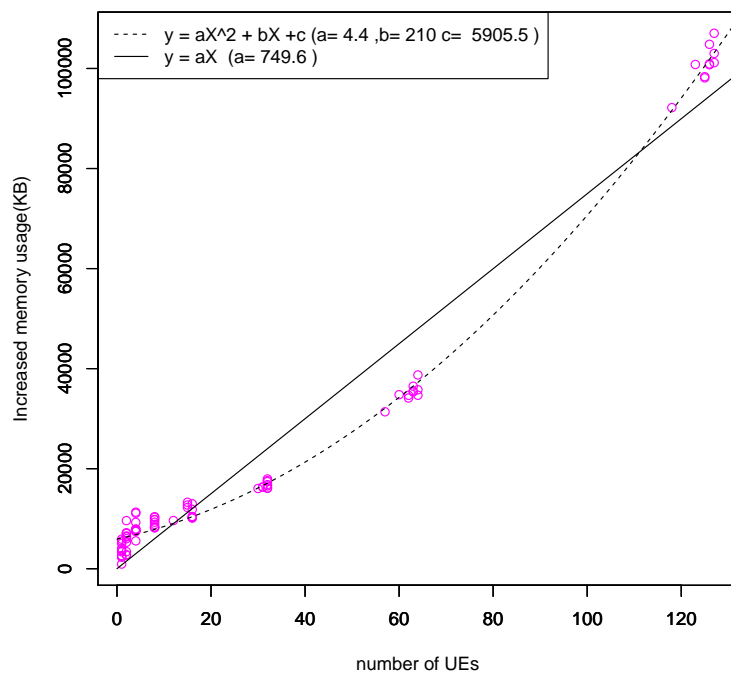


図 5: ベアラの確立によって増加したメモリ使用量 (MME)

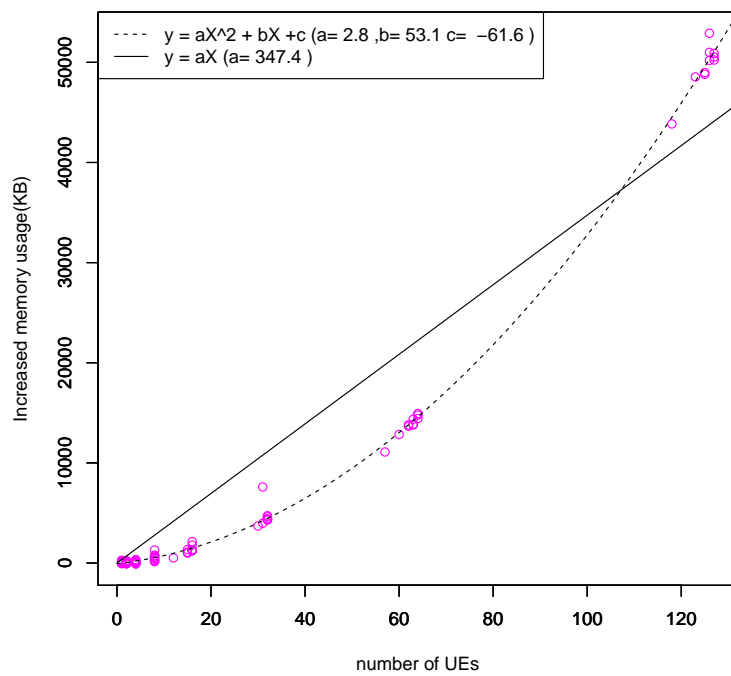


図 6: ベアラの確立によって増加したメモリ使用量 (SPGW)

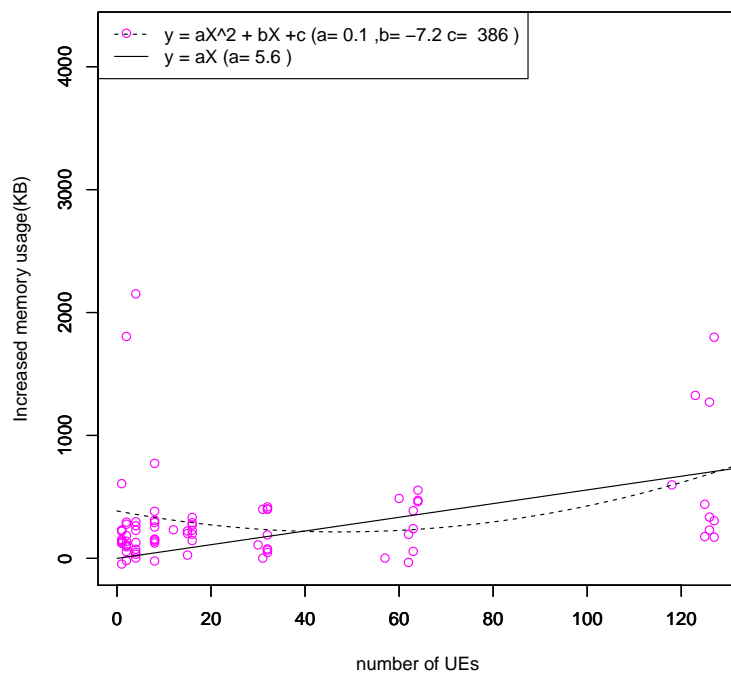


図 7: ベアラの確立によって増加したメモリ使用量 (HSS)



### 3 先行研究調査

サーバのリソース分離に関する文献 [2] 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 [3] 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 [4] 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 [5] 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。

### 4 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] “OpenAirInterface.” [Online]. Available: <http://www.openairinterface.org/>
- [2] 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.
- [3] 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.

- [4] 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.
- [5] S. Hailu, M. Saily, and O. Tirkkonen, “RRC State Handling for 5G,” *IEEE Communications Magazine*, vol. 57, no. 1, pp. 106–113, Jan. 2019.