

ミーティング資料

安達智哉

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

2019 年 2 月 20 日

1 CPU 負荷とメモリ使用量を考慮したセルラ端末の適応的なタイムアウト制御による、モバイルネットワークの性能向上

1.1 評価方法

シミュレーションを行い、収容可能な UE 台数を導出することで評価を行う。また、UE の通信特性やベアラのタイムアウト時間が与える影響も評価する。

1.1.1 ネットワークモデル

単一の EPC で構成される LTE/EPC ネットワークを対象として評価を行う。図 1 に評価対象のネットワークを示す。図 1 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。

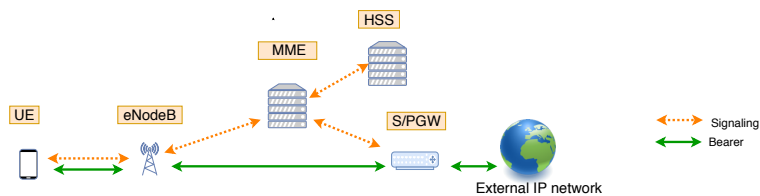


図 1: LTE/EPC ネットワークモデル

1.1.2 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。

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データの送信間隔
が異なる複数種類の UE 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。

1.1.3 ベアラのタイムアウト時間

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。今回の評価では、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。

1.1.4 シグナリング手順

OpenAirInterface(OAI)[1] に基づき、アタッチ処理およびデタッチ処理のシグナリング手順を決
定した。また、シグナリング処理の過程で各ノードにかかる CPU 負荷は、OAI のソースコードの
命令文数を指標として用いる。

1.1.5 サーバ資源の割り当て

今後検討予定である。

1.1.6 CPU 負荷の導出

CPU にかかる負荷はこれまで行ってきた研究と同様に、プログラム行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。

1.1.7 メモリ負荷の導出

上野さんの実験データを基にメモリ負荷を導出する。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。UE 台数が 1 台、2 台、4 台、
8 台、16 台、32 台、64 台、128 台の時の実験データそれぞれ 10 回分を用いて、ベアラ確立の前
後のメモリ使用量の増加量をプロットした結果を図 2、図 3、図 4 に示す。この結果より、全ての
ノードにおいて UE 台数が増加するのに伴い、メモリの使用量が増加していることがわかる。また

この結果を見る限りでは、MME および SPGW におけるメモリ使用量は二次関数的に増加していることがわかる。

このような二次関数的な増加の傾向が見られる理由については、はっきりとした理由は分かっていない。あくまでも仮説ではあるが、OAI の実装では、UE と eNodeB の情報を 2 次元行列の形式で保持しているのではないかと考えている。UE 及び eNodeB をそれぞれ行と列とした 2 次元行列の形式で UE と eNodeB の対応関係やベアラ情報を管理しているのだとすれば、UE 台数の増加に伴い二次関数的にメモリ負荷が増加する理由に説明がつく (上野さんの実験では、UE の増加に伴い eNodeB の数も増加している)。

一方で、UE 台数の増加に伴いメモリの使用量が二次関数的に増加するという結果は、直感的な認識に反するものであると考えられる。なぜなら、メモリ使用量は、メモリに保持させる情報量と同程度でありかつ、UE 1 台あたりのベアラを管理するために必要なメモリに保持させる情報量は、UE 台数と関係なく固定であるはずだからである。そのため今後の予定としては、UE 台数の増加に伴い、線形でメモリ負荷が増加すると仮定して評価を行おうと考えている。UE 1 台あたり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。

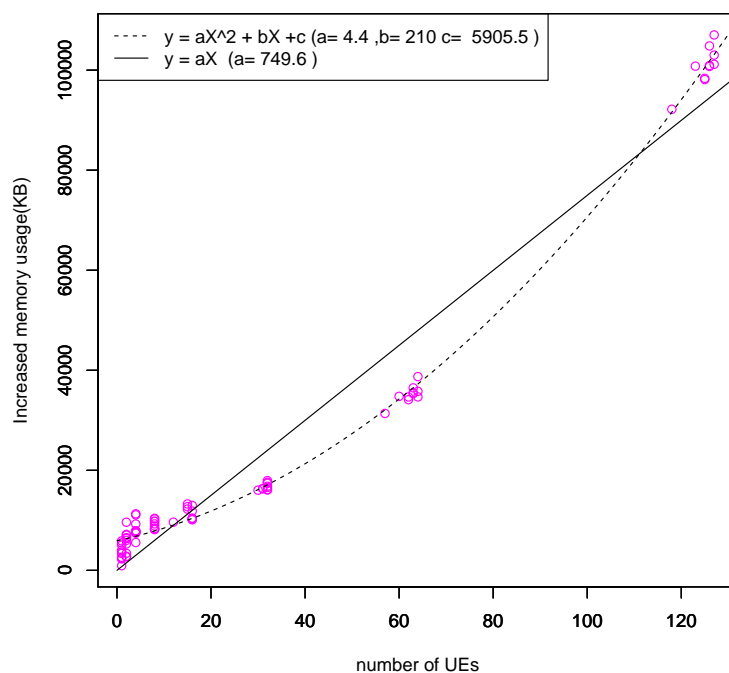


図 2: ベアラの確立によって増加したメモリ使用量 (MME)

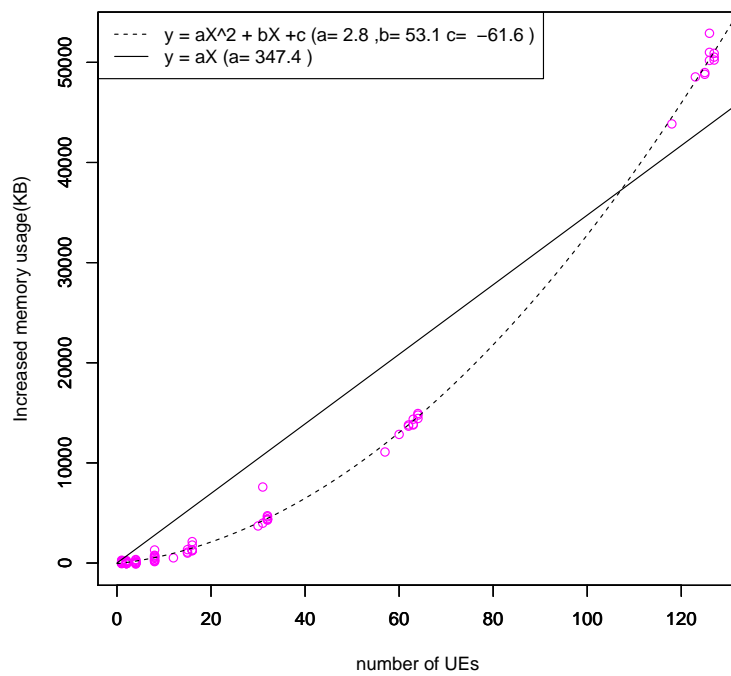


図 3: ペアラの確立によって増加したメモリ使用量 (SPGW)

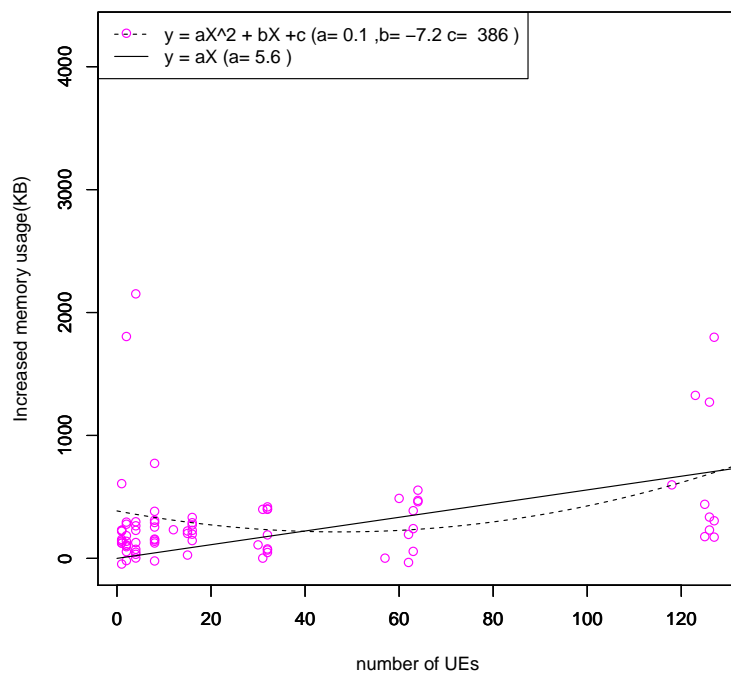


図 4: ベアラの確立によって増加したメモリ使用量 (HSS)

1.2 シミュレーション方法

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。そしてその負荷に基づき、次のタイムステップにおけるベアラのタイムアウト時間を設定する。シミュレートを実行するために必要な処理を以下に示す。

1. EPCの負荷の算出
2. ベアラのタイムアウト時間の設定

EPCの負荷の算出のイメージ図を図5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、待ち行列理論、上野さんの実験結果および統計処理を用いる。

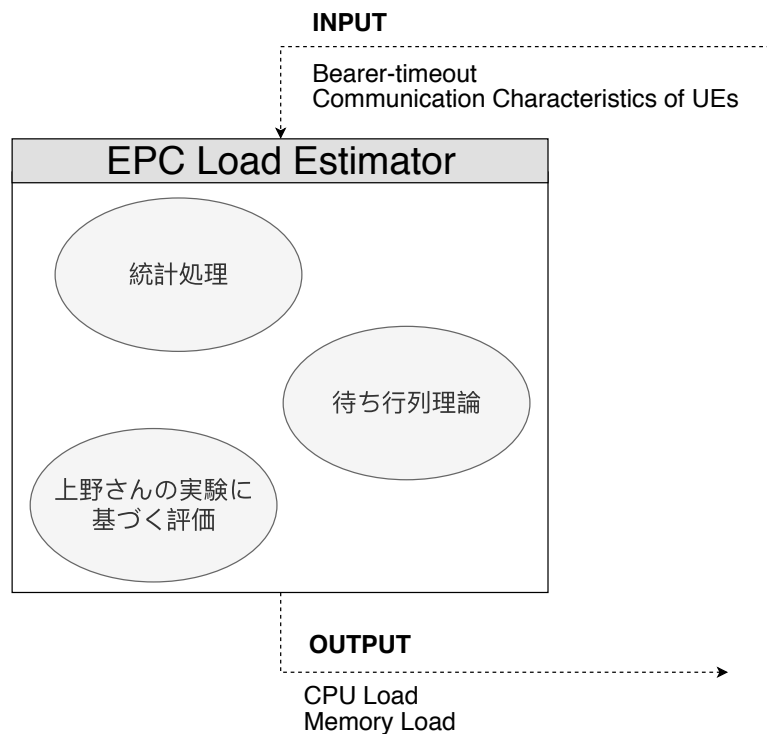


図 5: EPC 負荷の算出イメージ図

1.2.1 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回数およびベアラの接続数を統計的に導出する方法について述べる。

まず、図6に示すイメージ図のように、UEを常にベアラを確立しているUE(u)とアタッチおよびデタッチを繰り返すUE(\bar{u})に分類し、それぞれの台数を n_u 、 $n_{\bar{u}}$ とする。それぞれのUEのデータの平均送信間隔を $t_u(s)$ 、 $t_{\bar{u}}(s)$ とすると、シグナリングの発生レート $S(\text{回数}/s)$ およびベアラの接続数 B は以下の式(1)、(2)で表される。

$$S = (n_{attach} + n_{detach}) \cdot \frac{n_{\bar{u}}}{t_{\bar{u}}} \quad (1)$$

$$B = U_n + n_{\bar{u}} \cdot \frac{t_{out}}{t_{\bar{u}}} \quad (2)$$

ここで、 n_{attach} および n_{detach} はそれぞれアタッチ処理およびデタッチ処理に伴い発生するシグナリングの個数を示す。また、ベアラのタイムアウト時間を t_{out} とする。

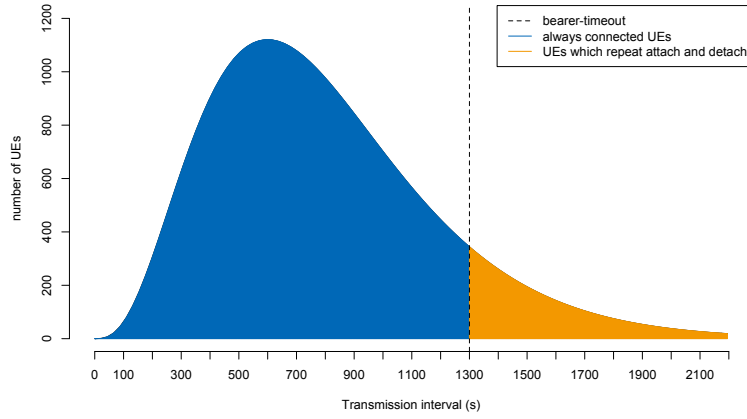


図 6: イメージ図

ベアラのタイムアウト時間の設定は、図 7 に示すように、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される。この処理では、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。なお、各リソースの負荷状況は図 5 によって計算された値を用いる。

一方で、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。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。現在考えている拡張案をいかに示す。

- 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。
- 移動平均を用いてベアラのタイムアウト時間を更新することで、動作を安定させる。

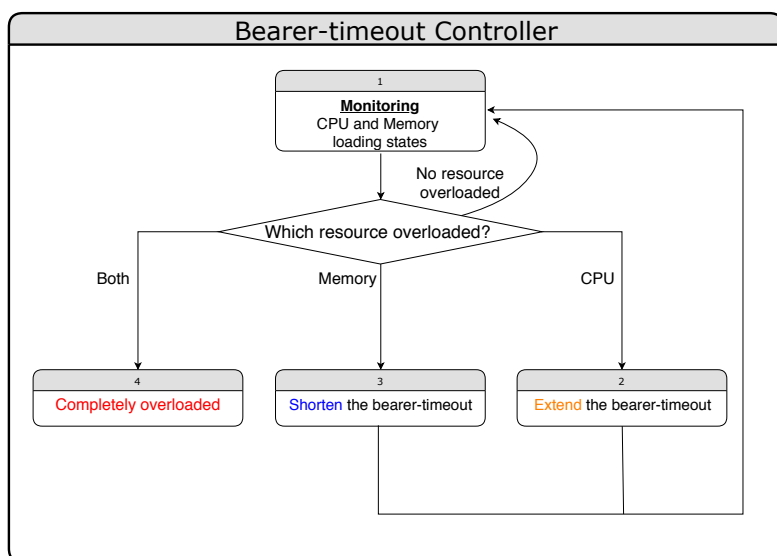


図 7: bearer-timeout 制御フロー

2 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。

3 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。

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.