進捗報告資料

安達智哉
to-adachi@ist.osaka-u.ac.jp

2019 年 12 月 19 日

1　Idle タイマの制御方法

本節では、UE の強制的な状態変化を引き起こさないことを前提にする。つまり、Idle タイマが切れていない UE を強制的に Idle 状態へ遷移させることはないとする。また、Idle タイマの更新は、UE がデータ送信を行うタイミングで実行するものとする。MME は UE を収容するために使用されている CPU およびメモリリソース量を観測できるものとする。つまり、UE の収容とは無関係な処理によって発生する負荷を取り除いた CPU 負荷およびメモリ使用量を知ることができるとする。MME は現在収容されている UE 台数を観測できるものとする。

突発的な負荷の増加に対応するという観点から、現在収容している UE に加え、最も多くの UE を収容できるような Idle タイマの値が最適と考える。具体的には、現在収容している UE と同じ通信周期を持つ UE がネットワークに参加すると仮定し、収容可能な UE 台数が最大となる Idle タイマの値を最適と定義する。また、CPU よびメモリのどちらも過負荷状態でないことは、UE を収容可能であることの必要十分条件であるとする。

まず、UE 一台あたりが各リソースに与える負荷の平均を推定する。現在収容している UE 台数を N_{UE} とする。UE 台数が N_{UE}、Idle タイマが T の時に観測される、CPU 負荷およびメモリ使用量をそれぞれ $C_{N_{\mathrm{UE}}}(T)$、$M_{N_{\mathrm{UE}}}(T)$ とする。この時、UE 一台あたりが与える CPU 負荷およびメモリ使用量の平均 ($C_1(T)$、$M_1(T)$) は以下の式 (1)、(2) で表せる。

$$C_1(T) = \frac{C_{N_{\mathrm{UE}}}(T)}{N_{\mathrm{UE}}} \tag{1}$$

$$M_1(T) = \frac{M_{N_{\mathrm{UE}}}(T)}{N_{\mathrm{UE}}} \tag{2}$$

Idle タイマを T とした時に、収容可能な UE の総数を $N_{\mathrm{UE}}^{\mathrm{capa}}(T)$ とする。$N_{\mathrm{UE}}^{\mathrm{capa}}(T)$ は、$C_1(T)$、$M_1(T)$、$C^{\max}$ および $M^{\max}$ を用いて、以下の式 (3) で表せる。ここで、$C^{\max}$、$M^{\max}$ はそれぞれシグナリング処理および UE のセッション情報を保持するために使用可能な CPU リソース量およびメモリリソース量である。

$$\begin{aligned} N_{\mathrm{UE}}^{\mathrm{capa}}(T) &= \left\lfloor \min\left\{\frac{C^{\max}}{C_1(T)}, \frac{M^{\max}}{M_1(T)}\right\}\right\rfloor \\ &= \left\lfloor N_{\mathrm{UE}} \cdot \min\left\{\frac{C^{\max}}{C_{N_{\mathrm{UE}}}(T)}, \frac{M^{\max}}{M_{N_{\mathrm{UE}}}(T)}\right\}\right\rfloor \end{aligned} \tag{3}$$

Idle タイマを制御する上での目的関数を以下の式 (4) に示す。

$$\text{maximize}: \quad N_{\mathrm{UE}}^{\mathrm{capa}}(T) \tag{4}$$

$N_{\mathrm{UE}}^{\mathrm{capa}}(T)$ を最大化する Idle タイマの値が明らかである場合は、その値を Idle タイマに設定すれば良い。しかし一般的に、UE の台数や通信周期は未知であり時間的に変動するため、$N_{\mathrm{UE}}^{\mathrm{capa}}(T)$

を最大化する Idle タイマの値を知ることは難しい。そのような場合は、$N_{\mathrm{UE}}^{\mathrm{capa}}(T)$ を最大化するように、Idle タイマを適応的に制御する必要がある。具体的には、各リソースの使用量を観測して、$N_{\mathrm{UE}}^{\mathrm{capa}}(T)$ を大きくする向きに Idle タイマを変化させる。このステップを複数回繰り返すことにより、Idle タイマを制御する。

この時、1 ステップごとの Idle タイマの変化量を考える必要がある。この値を小さく設定すると、最適な値に到達するまでに大きな時間がかかってしまう場合がある。逆に Idle タイマの変化量を大きく設定すると、Idle タイマが発振する可能性もあり、制御が不安定になる。また、UE の通信周期によって、Idle タイマが変化した時に各リソースの負荷の変化量が異なる点も考慮する必要がある。つまり、ネットワークの変化に短い時間スケールで対応しつつ、安定した制御を実現するためには、ネットワークの環境に応じて Idle タイマの変化量を制御する仕組みが必要である。このような制御には様々な手法が考えられるが、本報告では動作がシンプルであり、汎用性が高い PID 制御を用いる。T および $N_{\mathrm{UE}}^{\mathrm{capa}}(T)$ をそれぞれ、PID 制御における入力値および出力値として捉えることで、Idle タイマの変化量を調整しつつ、最適値に近づけることができる。

まず、PID 制御における出力値 $y(t)$ および目標値 $r(t)$ を設定する。以前の評価より、UE 台数を固定した時、CPU 負荷は Idle タイマの値に対して広義単調減少でありかつ、メモリ使用量は Idle タイマの値に対して広義単調増加であることがわかっている。このことから、式 (3) を確認すると、$\frac{C^{\max}}{C_{N_{\mathrm{UE}}}(T)}$ は広義単調増加でありかつ、$\frac{M^{\max}}{M_{N_{\mathrm{UE}}}(T)}$ は広義単調減少であることがわかる。ここで、$\frac{C^{\max}}{C_{N_{\mathrm{UE}}}(T)}$ と $\frac{M^{\max}}{M_{N_{\mathrm{UE}}}(T)}$ の差分を最小化するような T の集合を $\boldsymbol{T}$ とする。また、$N_{\mathrm{UE}}^{\mathrm{capa}}(T)$ を最大化するような T の集合を $\boldsymbol{T}_{\mathrm{optimal}}$ とする。すると、$T \in \boldsymbol{T}$ であることは $T \in \boldsymbol{T}_{\mathrm{optimal}}$ であるための十分条件になる。

このことを踏まえ、PID 制御における出力値 $y(t)$ および目標値 $r(t)$ を以下の式 (5)、(6) のように定義する。t は時刻を表す変数である。

$$y(t) = \frac{C^{\max}}{C_{N_{\mathrm{UE}}}(T)} - \frac{M^{\max}}{M_{N_{\mathrm{UE}}}(T)} \tag{5}$$

$$r(t) = 0 \tag{6}$$

式 (5) の問題点と修正案

式 (5) では、CPU の余力を表す指標として $\frac{C^{\max}}{C_{N_{\mathrm{UE}}}(T)}$、メモリの余力を表す指標として $\frac{M^{\max}}{M_{N_{\mathrm{UE}}}(T)}$ を用いている。ここでの余力とは、現状収容している UE 台数に対する、追加で収容可能な UE 台数の割合である。

式 (5) の問題点は、各指標の分母が小さい値をとると、制御が著しく不安定になることである。例えば、あるタイムステップで CPU 使用率が 0 になったとすると、$\frac{C^{\max}}{C_{N_{\mathrm{UE}}}(T)}$ が無限大に発散するため、Idle タイマを減少させる向きに無限の力で PID 制御が働くことになる。このような問題を解消するために、式 7 に示すように $y(t)$ の式を変更することを考えている。

$$\begin{aligned} y(t) &= (CPU \text{ 使用率}) - (\text{メモリ使用率}) \\ &= \frac{C_{N_{\mathrm{UE}}}(T)}{C^{\max}} - \frac{M_{N_{\mathrm{UE}}}(T)}{M^{\max}} \end{aligned} \tag{7}$$

時刻 t における $y(t)$ と $r(t)$ の差を $e(t)$ として以下の式 (8) ように定義すると、PID 制御における操作量 ($u(t)$) は以下の式 (9) で表せる。

$$e(t) = r(t) - y(t) \tag{8}$$

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{de(t)}{dt} \tag{9}$$

ここで、K_p、K_i および K_d はそれぞれ、比例ゲイン、積分ゲインおよび微分ゲインと呼ばれる定数である。これらの定数は、$e(t)$ およびその積分値、微分値が $u(t)$ にどの程度寄与するのかを決定する。

2 今後の予定

- PID 制御の出力値 ($y(t)$) に関する式を変更した上で、これまで行っていた一連の評価結果を差し替える (次回の報告まで)
- 制御の安定性と制御方式の関係について整理する
- リソースが不足した際の制御を考える

参考文献