

# 進捗報告資料

安達智哉

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

2019 年 6 月 18 日

## 1 MME におけるメモリ負荷の算出

### 1.1 UE の状態遷移に伴うメモリ負荷

文献 [1] に示されているコネクション確立に伴うシグナリング図を図 1 に示す。UE が Idle 状態から Connected 状態へ遷移する際に各ノードのメモリが保持する情報について OAI のソースコード (OpenairinterfaceCN-develop) を元に調査を行っている。具体的には、各シグナリングを処理する際に各ノードがメモリに格納する情報をリストアップし、それらの情報量を足し合わせることでメモリ負荷を推定する。今回は、MME が関与している以下のシグナリングを処理する際に MME が保持する情報を調査した。

- S1-AP Initial UE msg
- S1-AP Initial Ctxt Setup Request
- S1-AP Initial Ctxt Setup Compl
- Modify Bearer Request
- Modify Bearer Response
- S1-AP UE Ctxt Release Req
- Release Access Bearers Req
- Release Access Bearers Resp
- S1-AP UE Ctxt Release Cmd
- S1-AP UE Ctxt Release Compl

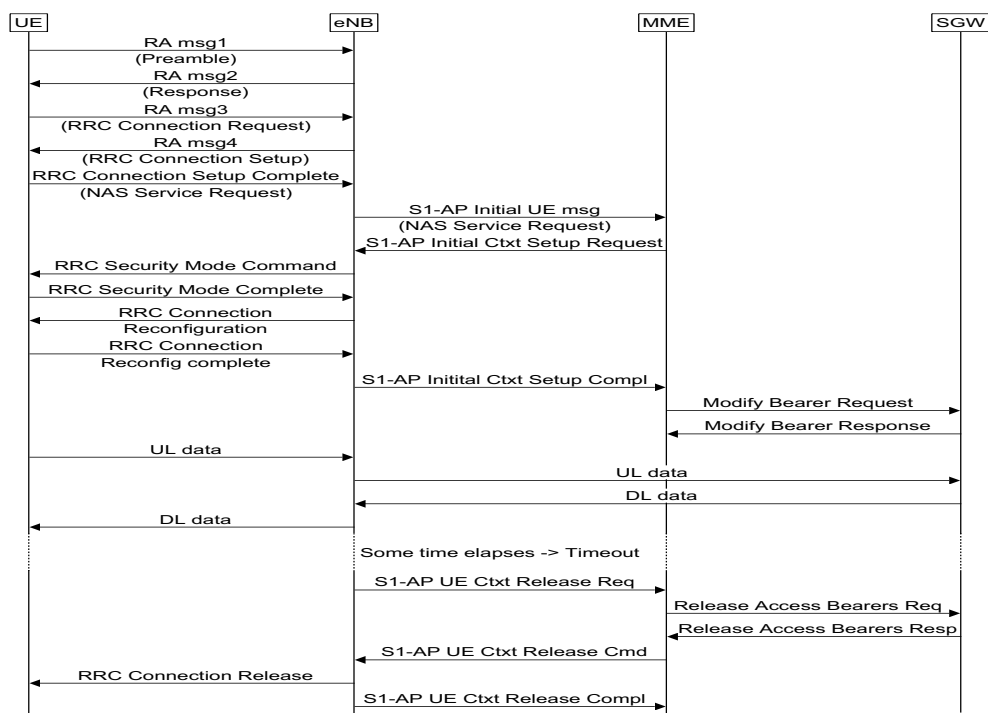


图 1: Legacy connection setup

MME が各シグナリングを受信、送信する際に保持する情報を以下の表 1 に示す。

S1-AP Initial UE msg を受信した際は、約 2KB の分の構造体を生成しており、この構造体に UE のコンテキストやベアラ情報を格納している。S1-AP Initial Ctxt Setup Compl の受信と Modify Bearer Request の送信処理では、メモリに情報を追加する処理は行っていない。Modify Bearer Response を受信した際は、MME の持つ UE のステートに関する情報の更新を行っている。しかし、メモリに保持する情報量に変化はない。

S1-AP UE Ctxt Release Req を受信した際は、MME が保持している UE のステートを Connected から Idle へ変更する処理や次のシグナリングである Release Access Bearers Req の準備等を行っているが、メモリに保持する情報の追加や削除は行われていない。Release Access Bearers Req の送信の処理ではメモリ操作はない。Release Access Bearers Resp を受信してから S1-AP UE Ctxt Release Cmd を送信するまでの処理で、MME の管理するベアラ情報の更新を行う。S1-AP UE Ctxt Release Compl を受信した際に UE のコンテキストを削除している。

表 1: シグナリングメッセージを処理する際に MME が保持する情報

シグナリング	情報名	情報量 (bit)
S1-AP Initial UE msg	ue_description_s	408
	ue_context_s	17470
S1-AP Initial Ctxt Setup Compl	-	0
Modify Bearer Request	-	0
Modify Bearer Response	-	0
S1-AP UE Ctxt Release Req	-	0
Release Access Bearers Req	-	0
Release Access Bearers Resp	-	0
S1-AP UE Ctxt Release Compl	ue_context_s	-17470

S1-AP Initial UE msg を処理する際に MME は、ue\_description\_s および ue\_context\_s という名前の構造体を保持することが表 1 より分かる。この構造体の中身を以下の表 2、表 3 に示す。

表 2: ue\_description\_s のメンバ

メンバ	情報量 (bit)
enb_description_s	32
s1_ue_state_s	160
enb_ue_s1ap_id_t	24
mme_ue_s1ap_id_t	32
sctp_stream_id_t (s ctp_stream_recv)	16
sctp_stream_id_t (s ctp_stream_send)	16
s11_sgw_teid	32
outcome_response_timer_id	32
s1ap_ue_context_rel_timer	64
合計	408

表 3: ue\_context\_s のメンバ

メンバ	情報量 (bit)
imsi	64
imsi_auth	1
enb_s1ap_id_key_t	64
enb_ue_s1ap_id_t	24
mme_ue_s1ap_id_t	32
sctp_assoc_id_t	32
ue_context_rel_cause	224
subscription_known	1
msisdn[MSISDN_LENGTH+1]	128
msisdn_length	8
mm_state	64
ecm_state	64
is_guti_set	8
guti	80
me_identity	240
e_utran_cgi	56
cell_age	64
access_mode	128
apn_profile	356
access_restriction_data	32
sub_status	96
subscribed_ambr	128
used_ambr	128
rau_tau_timer	32
*ue_radio_capabilities	8
ue_radio_cap_length	32
mme_s11_teid	32
sgw_s11_teid	32
paa	328
pending_pdn_connectivity_req_imsi[16]	128
pending_pdn_connectivity_req_imsi_length	8
pending_pdn_connectivity_req_apn	72
pending_pdn_connectivity_req_pdn_addr	72
pending_pdn_connectivity_req_pti	32
pending_pdn_connectivity_req_ue_id	32
pending_pdn_connectivity_req_qos	160
pending_pdn_connectivity_req_pco	784
pending_pdn_connectivity_req_request_type	32
default_bearer_id	8
eps_bearers[BEARERS_PER_UE]	13464
mobile_reachability_timer	64
implicit_detach_timer	64
initial_context_setup_rsp_timer	64
合計	17470

今回の調査の結果と文献 [2] で示されている、RRC Connected Inactive 状態から Connected 状態へ遷移する際のシグナリング図 (図 2) を参考にすることにより、各状態にある UE が MME に与えるメモリ負荷を推定することができた。結果を表 4 に示す。

図 2 を見ると、RRC Connected Inactive 状態から Connected 状態へ遷移する際には UE-RAN 間のシグナリングが 5 回発生している一方、コアネットワーク側にはシグナリングは発生していないことがわかる。よって RRC Connected Inactive 状態と Connected 状態では MME の状態には変化がなく、メモリ負荷も同じであると言える。

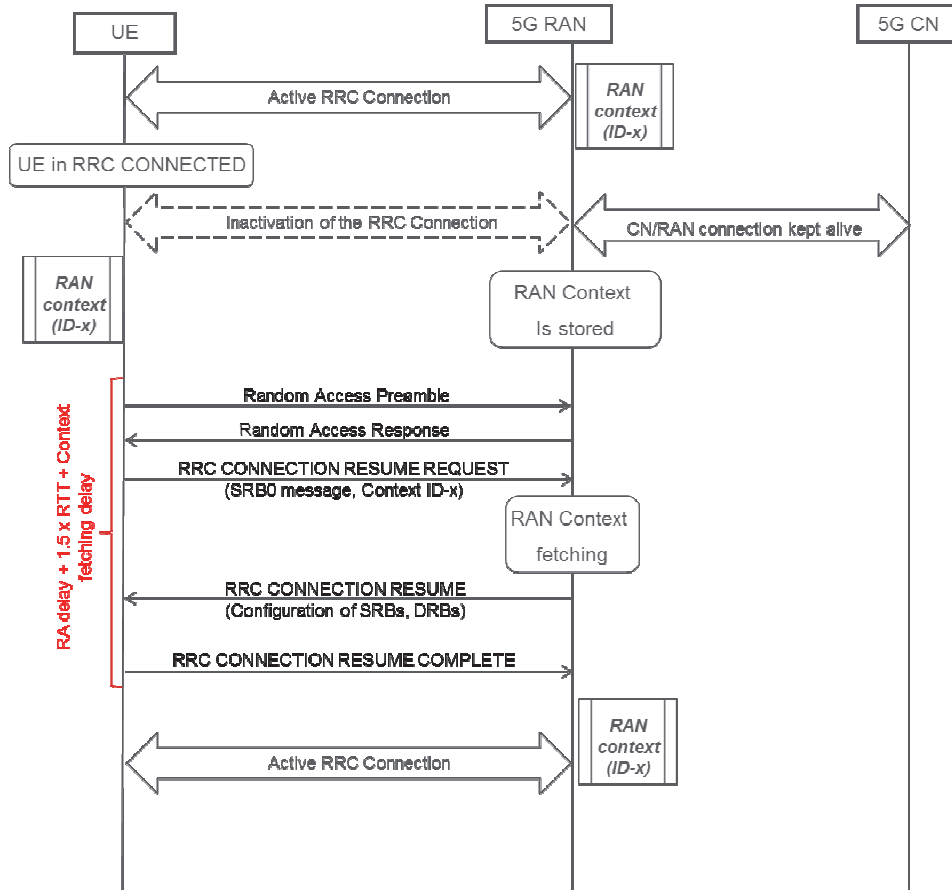


図 2: Signaling for the RRC CONNECTED INACTIVE to RRC CONNECTED transition for the novel state model

表 4: MME が保持する情報

UE のステート	情報名	情報量 (bit)
Connected	ue_description_s	408
	ue_context_s	17470
Connected Inactive	ue_description_s	408
	ue_context_s	17470
Idle	ue_description_s	408

## 1.2 eNodeB が追加された時のメモリ負荷

新規の eNodeB が接続した際には、`enb_description_s` という構造体が生成され、この情報を MME が保持する。この構造体の中身を表 5 に示す。この構造体のサイズは 225 bytes であることが分かった。

表 5: `enb_description_s` のメンバ

メンバ	情報量 (bit)
<code>s1_state</code>	128
<code>enb_name[150]</code>	1200
<code>enb_id</code>	32
<code>default_paging_drx</code>	8
<code>nb_ue_associated</code>	32
<code>ue_coll</code>	320
<code>sctp_assoc_id</code>	32
<code>next_sctp_stream</code>	16
<code>instreams</code>	16
<code>outstreams</code>	16
合計	1800

## 2 シグナリング数の調査

文献 [1] および [2] を調査することにより、状態遷移に伴うシグナリングの発生数が明らかになった。図 3 に示す状態遷移図と共に、状態遷移に伴って発生するシグナリングに関する情報を、表 6 に示す。

表 6: Signaling Load

遷移 ID	シグナリング処理数				遷移条件
	UE	RAN	MME	SGW	
1	9	12	5	2	Packets transmission
2	5	5	0	0	2 or more packets transmission
3	1	1	0	0	Connected timer expiration
4	4	4	0	0	One packet transmission
5	0	3	5	2	Connected Inactive timer expiration
6	0	0	0	0	Packets transmission
7	1	4	5	2	Connected timer expiration

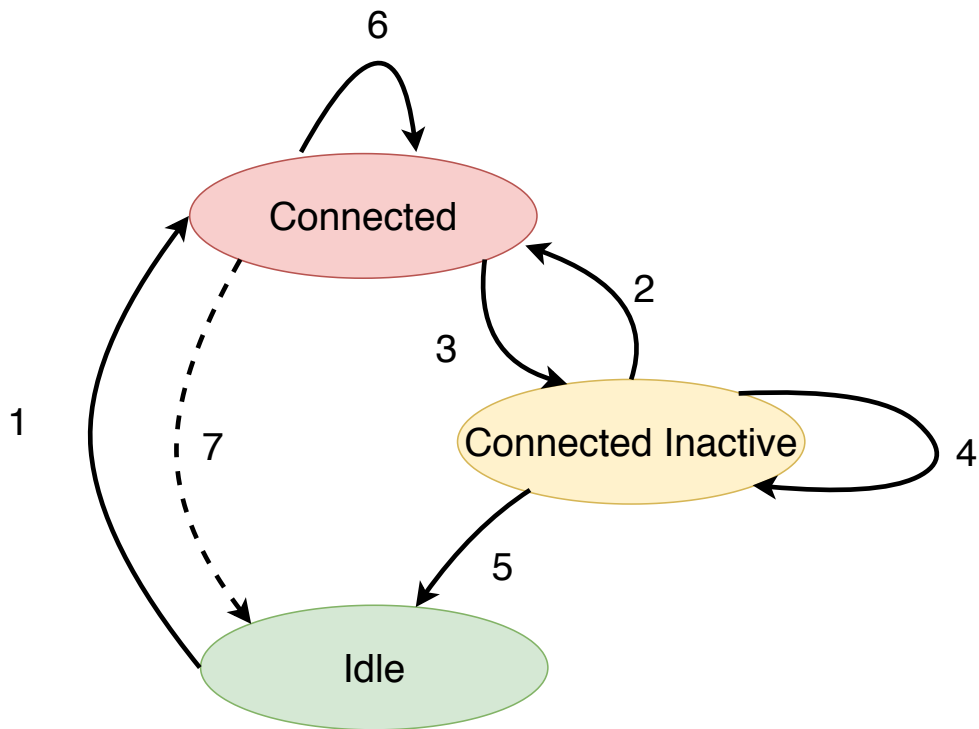


図 3: state transition

### 3 考察

今回の調査結果から、UE が Idle 状態と Connected 状態および Connected Inactive 状態にあるときに MME のメモリに与える負荷が明らかになった。今回の結果では、UE 1 台が Idle 状態である時は、約 408 bit のメモリ負荷が発生していると考えられる。また Connected 状態および Connected Inactive 状態である時は、約 17,878 bit のメモリ負荷が発生していると考えられる。

また、eNodeB が新しく追加された時は、約 1,800bit のメモリ負荷が発生することも分かった。

ここで以前調査した、上野さんの実験データに基づく MME のメモリ負荷を図 4 に示す。この図では、UE 及び eNodeB を増加させつつ UE のアタッチ処理を完了した際にどれほどのメモリ負荷が MME に発生するかを示している。図の結果では、UE 1 台のアタッチ処理のために約 750 KB のメモリ負荷が発生している。

図 4 の結果と今回の調査結果は直接比較することは、2 桁ほどずれている点は留意すべきである。上野さんに確認したところ、図 4 の結果は MME を起動しているシステム全体のメモリ消費を見ているということであるため、OAI 以外のレイヤーの通信プロトコルの処理負荷も含まれていることが分かった。図 4 の結果と今回の調査結果の間で大きなずれがある理由は、他の通信プロトコルの負荷の影響である可能性が高い。



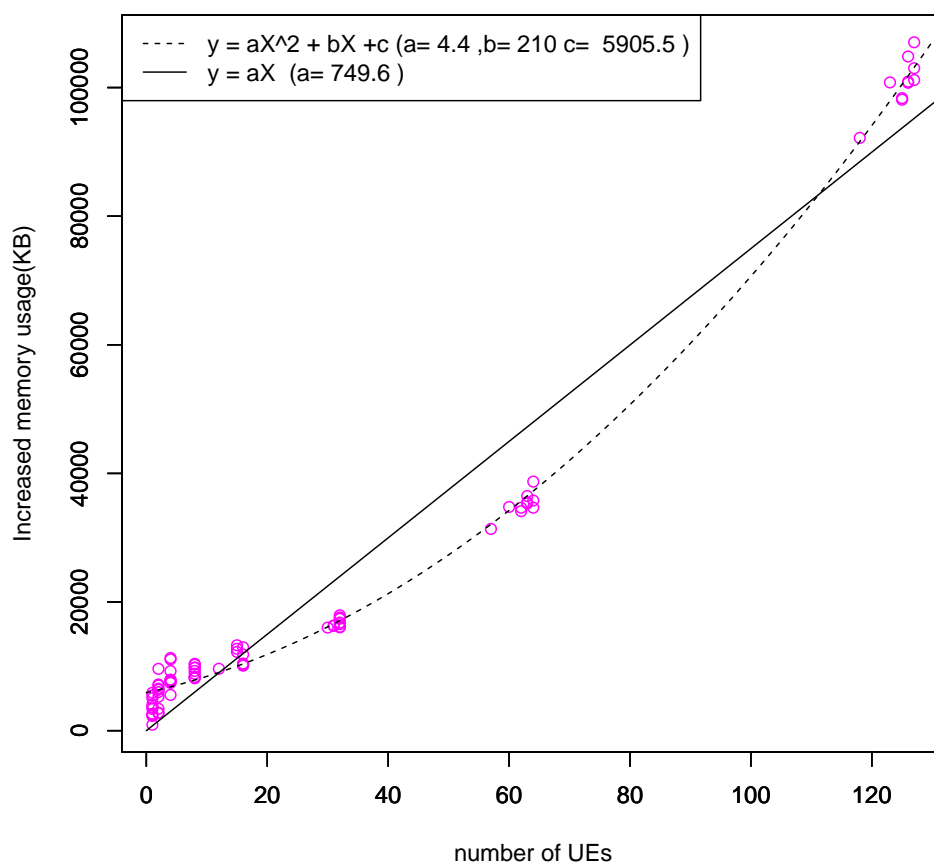


図 4: UE、eNodeB の増加とアタッチ処理の完了によって増加したメモリ負荷 (MME)

## 4 MME の負荷の試算方法

MME の負荷に合わせて、UE の状態を制御することにより、どの程度メモリおよび CPU に与える負荷のバランスを変化させることが可能であるかを試算する。UE の状態の制御は、UE が最後にデータを送信したあと、Idle 状態に移るまでの時間 ( $T^i$ ) を設定することで実現する。例えば、CPU が過負荷である場合は、 $T^i$  を長く設定することにより、どの程度メモリの負荷が増加し、CPU の負荷が低下するかを試算する。またその逆に、メモリが過負荷である場合は、 $T^i$  を短く設定することにより、どの程度 CPU の負荷が増加し、メモリの負荷が低下するかを調べる。

試算を行う上で決定すべきパラメータを以下に示す。

- UE の通信周期:  $D$
- 最後のデータ送信から Connected Inactive 状態へ遷移するまでの時間:  $T^{ci}$

### 4.1 CPU 負荷の算出

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。Connected Inactive 状態では、小さいデータ量であれば、コントロールプレーンを使ってデータ送信を完了させることが可能である。この場合、Connected 状態への状態遷移は発生しない。状態遷移が発生しない送信が可能であるデータサイズの上限を  $\alpha$  とした時、UE  $u_h$  がデータ送信を行う際に、そのデータサイズが  $\alpha$  を超える割合を  $d_h$  とする。最後のデータ送信から Connected Inactive 状態へ遷移するまでの時間を  $T^{ci}$ 、最後のデータ送信から Idle 状態へ遷移するまでの時間を  $T^i$ 、 $u_h$  の通信周期を  $T_h$  とする。また、状態遷移に伴うシグナリングの発生回数をそれぞれ表 7 のように定義すると、 $s_h$  は以下の式 (1) で表せる。ここで留意すべきは、 $T_h$  が  $T^{ci}$  以下であるような UE は、一度 Connected 状態に移ると、常時 Connected 状態を維持し、状態遷移によるシグナリングが発生しないため、 $s_h$  は 0 と定義したことである。また、私の研究では  $T^i$  を制御するのであるが、この値が  $T^{ci}$  以下になるような設定はしないことを前提にしている。そのため以下の式でも  $T^i \geq T^{ci}$  が常に成り立つものとしている。

表 7: state signaling

Source	Destination	The number of signaling occurrences
Connected	Connected	$n_{MME}^{c \rightarrow c}$
Connected Inactive	Connected Inactive	$n_{MME}^{ci \rightarrow ci}$
Connected	Connected Inactive	$n_{MME}^{c \rightarrow ci}$
Connected Inactive	Connected	$n_{MME}^{ci \rightarrow c}$
Connected Inactive	Idle	$n_{MME}^{ci \rightarrow i}$
Idle	Connected	$n_{MME}^{i \rightarrow c}$

$$s_h = \begin{cases} \frac{1}{T_h} \cdot n_{MME}^{c \rightarrow c} & \text{if } T_h \leq T^{ci} \\ \frac{1}{T_h} \cdot (n_{MME}^{ci \rightarrow c} + n_{MME}^{c \rightarrow ci}) \cdot d_h + \frac{1}{T_h} \cdot n_{MME}^{ci \rightarrow ci} \cdot (1 - d_h) & \text{if } T^{ci} < T_h \leq T^i \\ \frac{1}{T_h} \cdot (n_{MME}^{i \rightarrow c} + n_{MME}^{c \rightarrow ci} + n_{MME}^{ci \rightarrow i}) & \text{otherwise} \end{cases} \quad (1)$$

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。

$$S = \sum_{h=1}^{N_{\text{UE}}} s_h \quad (2)$$

この  $S$  より、CPU にかかる負荷を算出する。

## 4.2 メモリ負荷の算出

まず、UE  $u_h$  が Connected 状態である時間割合を  $p_h^c$ 、Connected Inactive 状態である時間割合を  $p_h^{\text{ci}}$ 、Idle 状態である時間割合を  $p_h^i$  と定義し、これらを求める。これらの値は、 $T_h$  および  $T^i$ 、 $T^{\text{ci}}$  を用いて以下の式 (6)、(7)、(8) で表せる。

$$p_h^c = \begin{cases} 1 & \text{if } T_h \leq T^{\text{ci}} \\ \frac{T^{\text{ci}}}{T_h} & \text{otherwise} \end{cases} \quad (3)$$

$$p_h^{\text{ci}} = \begin{cases} 0 & \text{if } T_h \leq T^{\text{ci}} \\ \frac{T_h - T^{\text{ci}}}{T_h} & \text{if } T^{\text{ci}} < T_h \leq T^i \\ \frac{T^i - T^{\text{ci}}}{T_h} & \text{otherwise} \end{cases} \quad (4)$$

$$p_h^i = \begin{cases} 0 & \text{if } T_h \leq T^{\text{ci}} \\ 0 & \text{if } T^{\text{ci}} < T_h \leq T^i \\ \frac{T_h - T^i}{T_h} & \text{otherwise} \end{cases} \quad (5)$$

$$p_h^c = \begin{cases} 1 & \text{if } T_h \leq T^{\text{ci}} \\ \frac{T^{\text{ci}}}{T_h} \cdot d_h + \frac{0}{T_h} \cdot (1 - d_h) & \text{if } T^{\text{ci}} < T_h \leq T^i \\ \frac{T^{\text{ci}}}{T_h} & \text{otherwise} \end{cases} \quad (6)$$

$$p_h^{\text{ci}} = \begin{cases} 0 & \text{if } T_h \leq T^{\text{ci}} \\ \frac{T_h - T^{\text{ci}}}{T_h} \cdot d_h + \frac{T_h}{T_h} \cdot (1 - d_h) & \text{if } T^{\text{ci}} < T_h \leq T^i \\ \frac{T^i - T^{\text{ci}}}{T_h} & \text{otherwise} \end{cases} \quad (7)$$

$$p_h^i = \begin{cases} 0 & \text{if } T_h \leq T^{\text{ci}} \\ 0 & \text{if } T^{\text{ci}} < T_h \leq T^i \\ \frac{T_h - T^i}{T_h} & \text{otherwise} \end{cases} \quad (8)$$

また、各状態におけるメモリ負荷をそれぞれ表 8 のように定義すると、UE  $u_h$  が MME に与えるメモリ負荷の平均  $b_h$  は以下の式 (9) で表せる。

$$b_h = m_{\text{MME}}^c \cdot p_h^c + m_{\text{MME}}^{\text{ci}} \cdot p_h^{\text{ci}} + m_{\text{MME}}^i \cdot p_h^i \quad (9)$$

表 8: 各状態における UE1 台当たりの MME のメモリ負荷

state	load of MME memory
Connected	$m_{\text{MME}}^c$
Connected Inactive	$m_{\text{MME}}^{\text{ci}}$
Idle	$m_{\text{MME}}^i$

MME に対してネットワーク全体で発生する平均的なメモリ負荷の合計を  $B$  と定義する。 $B$  は  $b_h$  を用いて以下の式 (10) で表せる。

$$B = \sum_{h=1}^{N_{\text{UE}}} b_h \quad (10)$$

この  $B$  よりメモリ負荷を算出する。

## 5 MME 負荷の試算

第 1 章で調査したメモリ負荷および、第 2 章で調査したシグナリング負荷を参考にし、第 4 章で述べた数式を用いることで、MME に発生する負荷を試算した。今回、試算するにあたり、以下のように条件やパラメータを仮定した。

- UE ごとに通信周期は固定であり、途中で変化することはない。
- 最後の送信が終了したあと、Connected 状態の UE が Connected Inactive 状態へ遷移するまでの時間 ( $T^{\text{ci}}$ ) は全 UE で共通かつ不変の値として  $T^{\text{ci}} = 10\text{s}$  とした。
- UE の送信するデータサイズは十分大きいものとする。つまり、データ送信を行うタイミングで必ず Connected 状態に遷移するものとする ( $d_h = 1$ )。
- ネットワークに存在する UE の台数は 600 台とし、通信周期に対する UE 台数の分布は以下の図 5 のように仮定する。つまり、通信周期に対して UE の台数は一様分布である。
- データ送信にかかる時間は十分短いものとし、送信が失敗することはないと仮定する。

$T^i$ (最後のデータ送信から Idle 状態へ遷移するまでの時間) を 10s から 600s の範囲で 1s の精度で変化させた時に、MME に対して発生する 1s あたりのシグナリング数と MME のメモリ負荷の変化を図 6 に示す。

この図を見ると、 $T^i$  の増加に対して、MME のシグナリング負荷は単調減少であることが分かる。これは、 $T^i$  が増加することにより、Idle 状態へ遷移する UE の台数が少なくなり、その分のシグナリングが削減されるからである。また、 $T^i$  の値が小さい時は MME のシグナリング負荷の減少量が大きく、 $T^i$  の値が大きい時は MME のシグナリング負荷の減少量が小さいことも分かる。これは、 $T^i$  の値が小さい場合、 $T^i$  の変化が影響を与えるのは通信周期が短い UE(シグナリング負荷への寄与が大きい UE) であり、 $T^i$  の値が大きい場合、 $T^i$  の変化が影響を与えるのは通信周期が長い UE(シグナリング負荷への寄与が小さい UE) であるからである。

また、 $T^i$  の増加に対して、MME のメモリ負荷は単調増加であることが分かる。これは、 $T^i$  が増加することにより、Idle 状態へ遷移するまでの時間 (Connected Inactive 状態を維持する時間) が

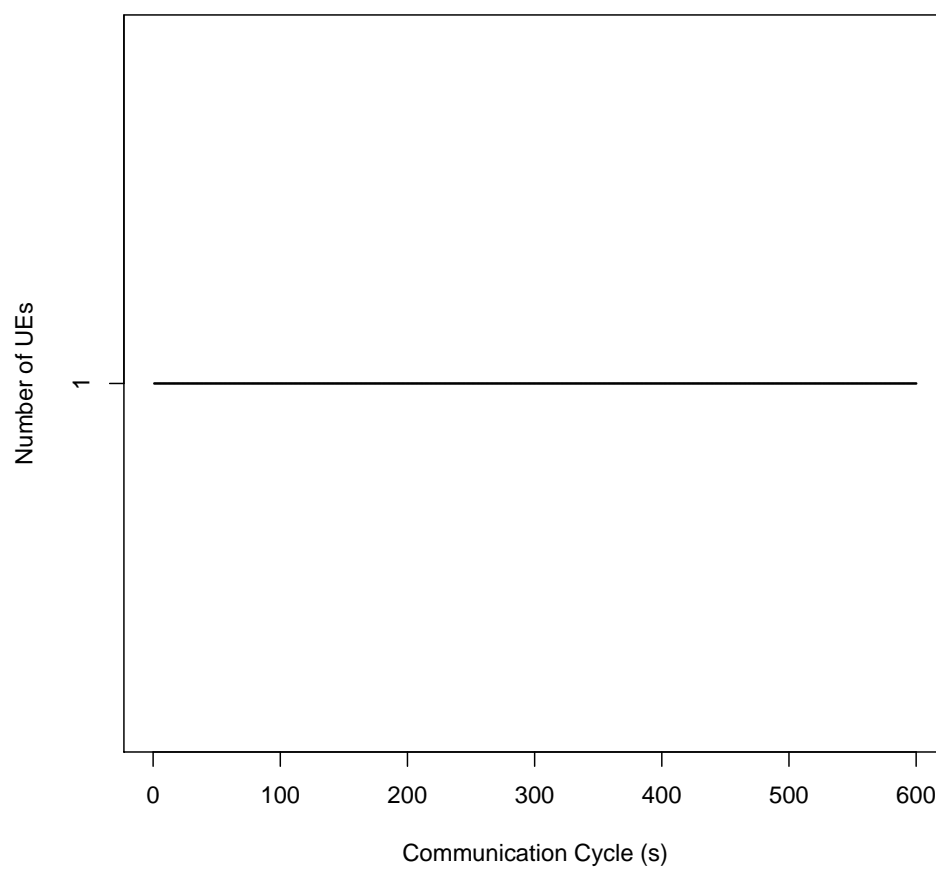


図 5: 通信周期に対する UE 台数の分布

長くなるため、および、Idle 状態へ遷移する UE の台数が少なくなるためである。また、 $T^i$  の値が小さい時は MME のメモリ負荷の増加量が大きく、 $T^i$  の値が大きい時は MME のメモリ負荷の増加量が小さいことも分かる。これは、 $T^i$  の変化は  $T^i$  よりも通信周期の長い UE の状態遷移に対してのみ変化を与えるためである。つまり、 $T^i$  の値が大きい場合は小さい場合と比べて、 $T^i$  の変化が影響を与える UE 台数が減るため、メモリ負荷の変化量が小さくなっている。

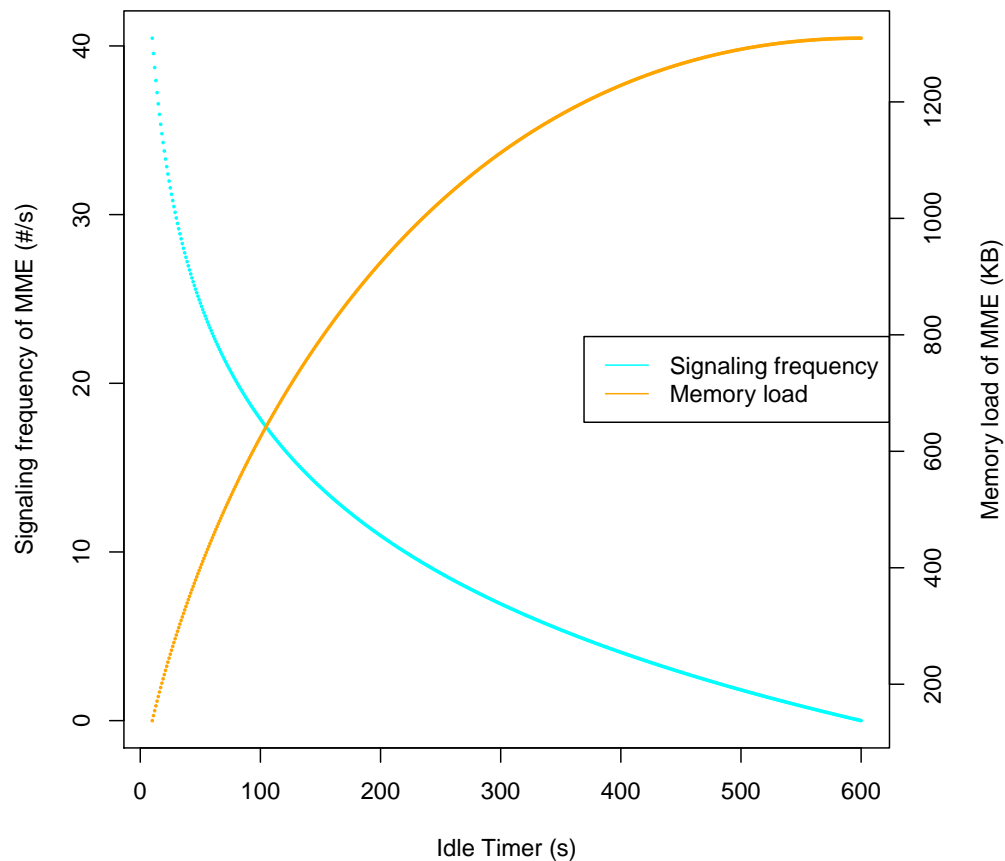


図 6:  $T^i$  に対する、MME に対して発生する 1s あたりのシグナリング数とメモリ負荷

MME のシグナリング負荷と MME のメモリ負荷の関係を示した図を図 7 に示す。

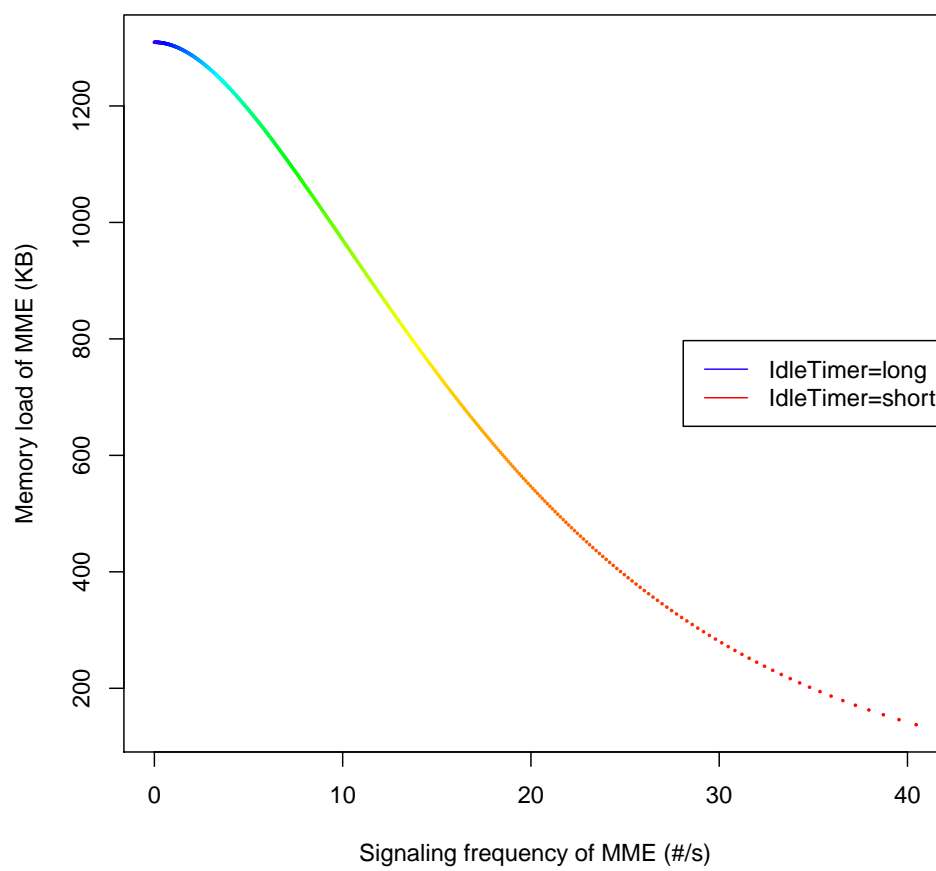


図 7: MME に対して発生する 1s あたりのシグナリング数に対する MME のメモリ負荷

## 6 MME が単位時間あたりに処理可能なシグナリングの最大数の推定

図 6 および図 7 に IdleTimer とメモリ負荷およびシグナリング負荷の関係を示した。メモリ負荷に関しては MME に搭載されているメモリのサイズや一般的なサーバに搭載されているメモリサイズなどと比較することにより、どの程度までの負荷が許容されるのかを議論することは容易である。

一方、シグナリング負荷に関しては、シグナリング負荷と CPU 負荷との関係が自明でないため、負荷の許容範囲を明らかにすることは、一般的に容易ではない。そこで、今回は、上野さんの実験結果を参考にしつつ、どの程度までのシグナリング負荷が許容されるのかを推定した。ここで留意すべきは、上野さんの実験から得られる情報は、ある特定の時刻において突発的にシグナリングが発生した場合の CPU 負荷であるが、私が知りたい情報は、継続的にシグナリングが発生し続けるような場合の CPU 負荷であり、両者は異なることである。そこで、以下に示すように、上野さんのデータを加工することにより、継続的にシグナリングが発生し続けるような場合の CPU 負荷を推定した。

以下に上野さんの実験データとそれを加工したものを示す。これらのデータを用いて、何台の UE がアタッチ処理を実行すると MME の CPU が過負荷になるのかを調査した。

図 8 に、128 台の UE がアタッチ処理を実行した時の実験データを示す。グラフの横軸が時間 ( $t(s)$ ) を示し、縦軸が MME の CPU の使用率を示す。 $t = 0$  ので UE のアタッチ処理が行われている。 $t = 0$  以降、MME の CPU 負荷が大幅に上昇していることがわかる。CPU 使用率は  $t = 0, 1, 2, 3$  の時点でそれぞれ、83%、100%、100%、59%となり、 $t = 4$  以降では、通常値 (0%~4%) に戻っている。図 9 は、図 8 を参考にし、 $t \geq 0$  のとき、毎秒 128 台の UE がアタッチ処理を実行したと仮定して、MME の CPU 負荷を推定したものである。この図では、CPU 使用率が 100%に達していることから、CPU が過負荷状態であると言える。よって、毎秒 128 台の UE がアタッチ処理を実行した際に発生するシグナリング負荷は CPU を過負荷状態にすることが分かった。

図 10 に、64 台の UE がアタッチ処理を実行した時の実験データを示す。グラフの横軸が時間 ( $t(s)$ ) を示し、縦軸が MME の CPU の使用率を示す。 $t = 0$  ので UE のアタッチ処理が行われている。 $t = 0, 1$  で MME の CPU 負荷が 45%、51%となり、 $t = 2$  以降では、通常値 (0%~4%) に戻っている。図 11 は、図 10 を参考にし、 $t \geq 0$  のとき、毎秒 64 台の UE がアタッチ処理を実行したと仮定して、MME の CPU 負荷を推定したものである。この実験では、CPU 使用率が 95% 近くに達しているが、100%には達していない。この結果より、毎秒 64 台の UE がアタッチ処理を実行した際に発生するシグナリング負荷をかけても、CPU の使用率は 100%を超えないため、過負荷にはならないことが分かった。

図 13 は、図 12 には、UE の台数が 32 台の場合の評価結果を示す。この結果では、CPU 使用率が十分小さく、100%を超えることはないため、過負荷ではないと言える。

また、UE 台数がさらに小さいような場合でも同様に、CPU が過負荷になることはないことが確認できた。

以上の調査から、64 台の UE がアタッチ処理を実行する際に発生するシグナリング負荷ならば、MME は処理可能であると言える。図 14 に示すように、MME は 1 つのアタッチ処理完了するために、15 回のシグナリング処理を行っている。このことから、毎秒約 960 回のシグナリング負荷であれば、CPU は過負荷にならないと推定できる。



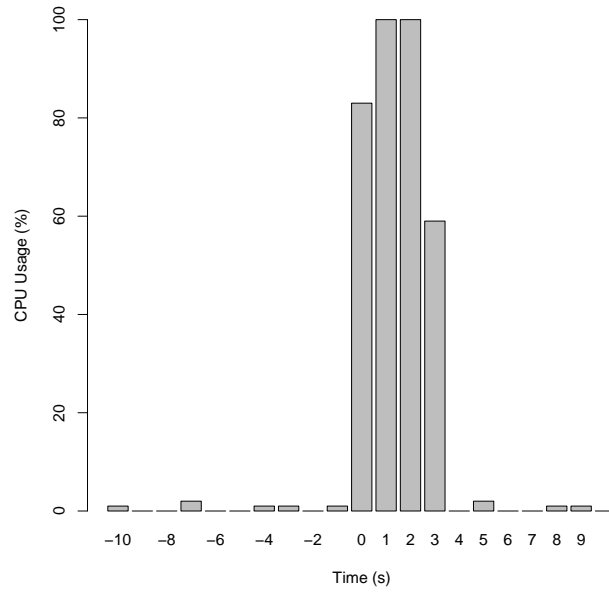


図 8:  $t = 0$  で 128 台の UE がアタッチ処理を実行した時の CPU 負荷 (実験データそのまま)

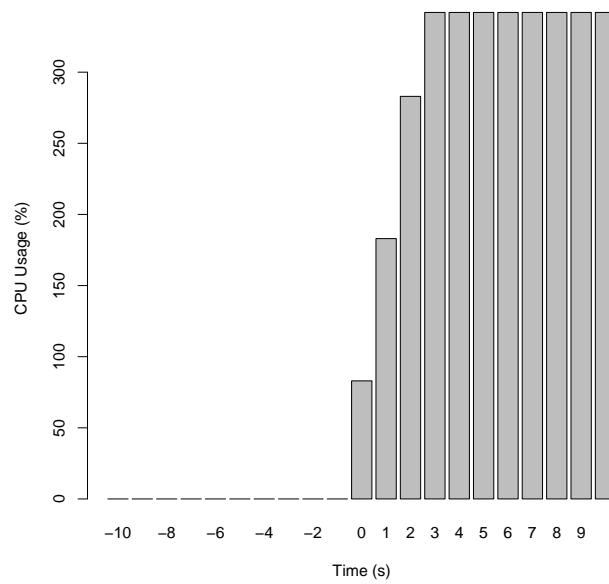


図 9:  $t \geq 0$  で毎秒 128 台の UE がアタッチ処理を実行した時の CPU 負荷 (推定)

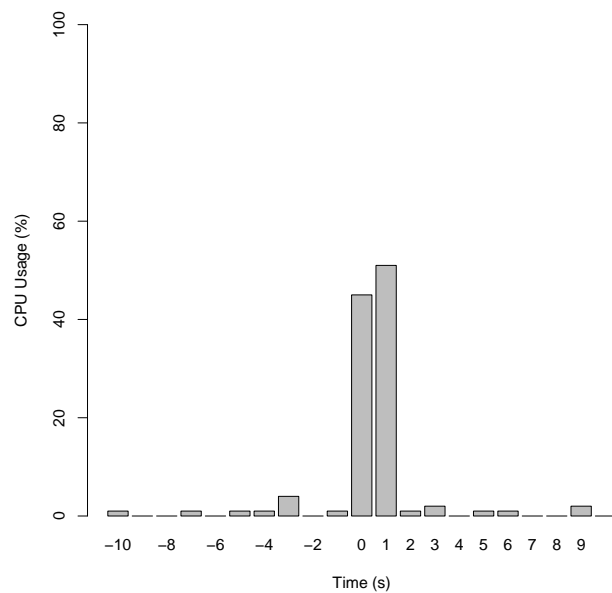


図 10:  $t = 0$  で 64 台の UE がアタッチ処理を実行した時の CPU 負荷 (実験データそのまま)

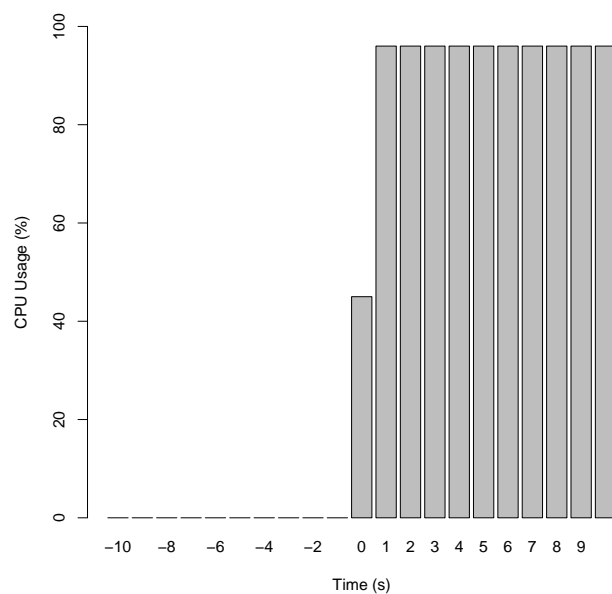


図 11:  $t \geq 0$  で毎秒 64 台の UE がアタッチ処理を実行した時の CPU 負荷 (推定)

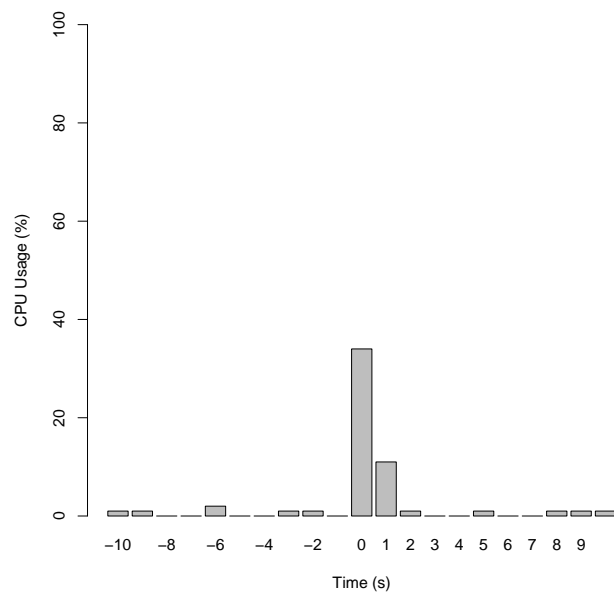


図 12:  $t = 0$  で 32 台の UE がアタッチ処理を実行した時の CPU 負荷 (実験データそのまま)

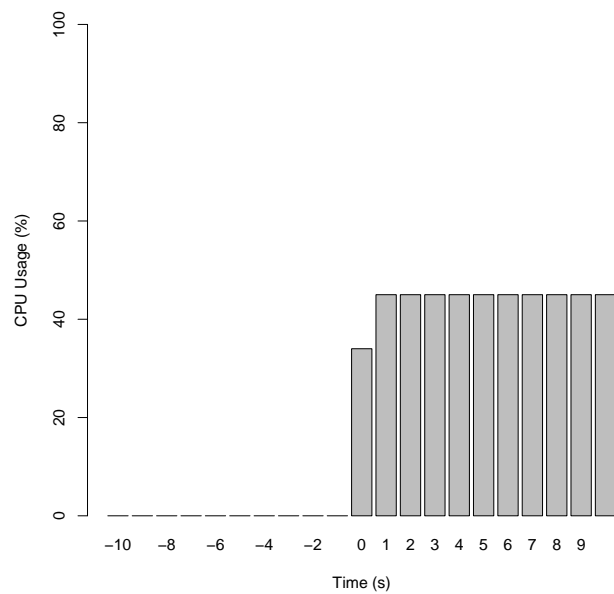


図 13:  $t \geq 0$  で毎秒 32 台の UE がアタッチ処理を実行した時の CPU 負荷 (推定)

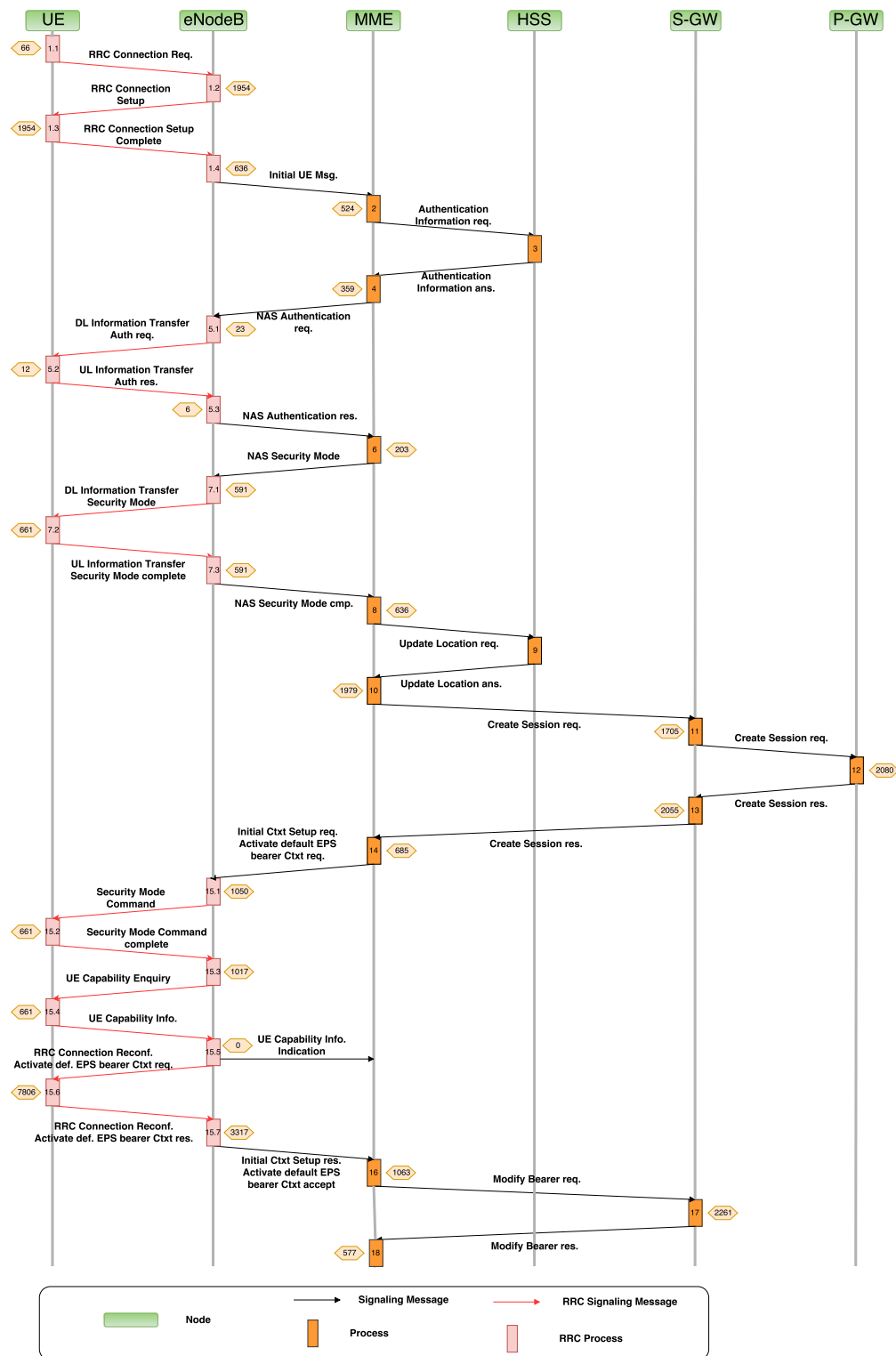


图 14

## 7 今後の予定

- UE の通信周期の分布やデータサイズなど、様々なパラメータを変化させた場合の試算を行う。
- MME 負荷の試算
- 必要に応じて SGW や eNodeB などのメモリ負荷も調査する。
- Connected Inactive 状態において “状態遷移を伴わないデータ送信” が可能なデータ量を調査する。

## 参考文献

- [1] 3GPP, “Study on architecture enhancements for Cellular Internet of Things (CIoT),” 3rd Generation Partnership Project (3GPP), Technical Report (TR) 23.720, Mar. 2016, version 13.0.0. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2894>
- [2] I. L. Da Silva, G. Mildh, M. S ä ily, and S. Hailu, “A Novel State Model for 5G Radio Access Networks,” in *Proceedings of 2016 IEEE International Conference on Communications Workshops (ICC)*, May 2016, pp. 632–637.