

# サーバプッシュにおける配信タイミング制御のためのモバイル端末の RRC 状態推定方式

大西健夫<sup>†1</sup> 城島貴弘<sup>†1</sup>

スマートフォンの普及に伴い、端末上のアプリケーションに対して、クラウド上のサービスからメッセージを配信する IP プッシュの利用が拡大している。IP プッシュは、リアルタイム性が要求される用途にも利用されるため、メッセージ配信遅延を抑制することが重要である。LTE などのモバイル網における通信遅延は、端末の消費電力削減のために制御が行われている無線リソースの確保状態(RRC 状態)に影響される。RRC 状態が休止状態である端末にメッセージ送信する場合は、一旦状態を遷移させる必要があり、遷移に要する時間の分だけメッセージ配信遅延が増大する。そのため、プッシュサーバにおいて端末の RRC 状態を推定して、休止状態の端末の遷移が早期に開始されるように配信処理をスケジューリングすることで、配信遅延を抑制することができる。スケジューリングには端末の RRC 状態を推定する必要があるが、RRC 状態の遷移は端末の通信を契機に発生するため、端末の通信開始時刻や通信頻度などの情報をサーバに通知することで、プッシュサーバにおいて端末の RRC 状態を推定可能となる。本稿では、プッシュサーバにおいて低負荷・高精度で RRC 状態の推定する方式を提案する。

## A Estimation method of Radio Resource Control state for scheduling message delivery executions in Server Push

TAKEO ONISHI<sup>†1</sup> TAKAHIRO SHIROSIMA<sup>†1</sup>

Push notification in which a message of event notification is delivered anytime from services on the cloud to applications on mobile terminals attracts rising attention according with expansion of smart phone. The message delivery delay is a key factor, because push notification is used for the services in which push messages need to be delivered in real time. A control of the radio resource control state (RRC state) aim at power consumption reduction in a mobile terminal with a mobile network such as LTE and the RRC state influences the delay of delivering a push message. When a message is delivered to the terminal in the idle state, the RRC state must be transitioned from the idle state and this transition lengthen the delay. If the push server can estimate the RRC state, the delay can be reduced by scheduling the message delivery executions so that the transition from the idle state is triggered at an early point. We have proposed the estimation method of the RRC state based on the data transfer rate notified by the terminal which influences the RRC state. In this paper we propose a method Estimation method of Radio Resource Control states with high efficiency.

### 1. はじめに

近年、スマートフォンの普及に伴い、スマートフォンのアプリケーションに対してクラウド上のサーバから任意のタイミングでメッセージを配信(プッシュ配信)する IP プッシュの利用が広まっている[1][2]。IP プッシュにより、新規に発生した情報をリアルタイムにアプリケーションやユーザーに届けることで、様々なサービスを実現することが可能となっている。たとえば、IP プッシュは、SNS などの更新通知やマルチメディア通信の着信、災害情報の通知などに活用されている。

IP プッシュにおいて、メッセージ配信の遅延時間はサービス品質を規定する重要な要素の一つである。なぜならば、IP プッシュの用途には、マルチメディア通信の着信や災害情報の通知など、即時性が求められる用途があり、低遅延でメッセージを配信する必要があるためである。

モバイル網を介したメッセージ配信時の遅延は、3GPP において規定されているモバイル端末の RRC(Radio Resource Control)状態[3]の影響を受ける。LTE などのモバ

イル網を利用する端末では、電力消費量を抑えるために、通信の必要がない場合は無線モジュールを休止させ、通信の必要が生じたときのみ無線モジュールを活性化させるという制御を行っている。以下、RRC 状態に関して簡単に説明する。

図 1 は、LTE における RRC 状態を示した図である。LTE においては、RRC\_CONNECTED と RRC\_IDLE の二つの状態が規定されている。

RRC\_IDLE は休止状態となっており、IP 通信が不可能な状態となる。RRC\_IDLE の端末が IP 通信を行う際は、IP 通信可能な RRC\_CONNECTED に遷移する必要がある。RRC 状態の遷移には、モバイル網との間で RRC メッセージとよばれるメッセージの交換が必要であり、その交換に要する時間が通信遅延として現れる。RRC\_IDLE 時は、無線モジュールを休止させているが、定期的に無線モジュールを起こし、自端末宛での IP パケットが基地局に到着しているかを確認する間歇受信を行っている。間歇受信の際に短時間だけ無線モジュールを駆動させるのみであるので、RRC\_IDLE 時の電力消費量は低く抑えられており、RRC\_CONNECTED に比べて数%程度となる[4]。

<sup>†1</sup> 日本電気株式会社  
NEC Corporation

一方、RRC\_CONNECTED は、IP 通信が可能な状態となる。RRC\_IDLE 時に IP 通信を行う場合に比べ、状態遷移を行う必要がないため、通信遅延時間が短くなる。IP パケットの送受信のために無線モジュールを活性化させているので、電力消費量は RRC\_IDLE に比べて大きくなる[4]。

RRC 状態の遷移は、IP 通信の開始、もしくは、無通信状態が一定時間継続することによって発生する。前述したように、RRC\_IDLE 時に IP 通信を開始しようとすることで RRC\_CONNECTED への遷移が発生する。一方、RRC\_CONNECTED にある端末が一定時間(Inactivity Timer, 図中T1と表記)通信を行わないと、RRC 状態が RRC\_IDLE に遷移する。例えば、T1の値を実測すると約 120 秒であった(なお、この値は通信事業者の設定に依存する)。

以上のように、モバイル網を利用する端末への通信遅延は RRC 状態の影響を受ける。そのため、メッセージ配信時の遅延抑制には、メッセージ配信を行うプッシュサーバ側で端末の RRC 状態を考慮した制御が有効であると考えられる。そこで、我々は、端末の RRC 状態を考慮したメッセージ配信遅延抑制方式を提案してきた[5][6]。RRC\_IDLE の端末にメッセージを配信する場合、遷移が完了するまではメッセージが到着しないため、遅延が大きくなる。前述の方式では、遷移を開始させるパケットを優先的に送信するよう配信処理をスケジュールすることで、遷移開始時間を早め、遅延時間を抑制する。

上記を実現するためには、プッシュサーバが端末の RRC 状態を知る必要がある。前述したように、RRC 状態の遷移は、端末の通信に紐づいて発生する。そのため、端末から通信が開始したことや通信の頻度などをプッシュサーバに通知することで、プッシュサーバにおいて端末の RRC 状態を推定することが可能となる [7]。一方で、IP プッシュは多数の端末を対象としたサービスであるため、多数の端末から推定のための通知を受信した場合、通知を受信する負荷(通知負荷)は無視できないものとなる。したがって、通知負荷を抑制しつつ、より高い推定精度を得ることが重要である。本稿では、従来の推定方式を改良し、同一量の通知負荷が発生する場合において、より高い推定精度を得る方式を提案する。

## 2. 従来方式

我々はこれまで、RRC 状態の推定法として逐次通知方式とガードタイム通知方式[7]を提案してきた。両者の基本的な考え方は、端末が行った通信についての情報を、端末からサーバに通知し、サーバは通知された情報に基づき端末の RRC 状態を推定する。以下、それぞれについて述べる。

### 2.1 逐次通知方式

逐次通知方式における RRC 状態の推定法を図 2 に示す。逐次通知方式では、端末が通信を開始したタイミング(図

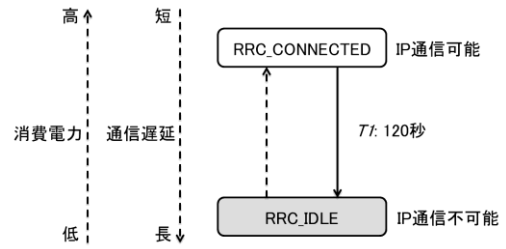


図 1 LTEにおける RRC 状態

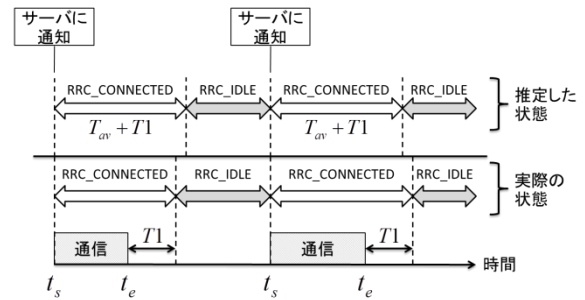


図 2 逐次通知方式における推定法

中の  $t_s$ ) に通信を開始したことをサーバに通知する。サーバは、端末から最後の通知を受信した時からの経過時間  $L$  に基づき、下記により RRC 状態を推定する。

$$\hat{S} = \begin{cases} \text{RRC\_CONNECTED} & (L \leq T_{av} + T1) \\ \text{RRC\_IDLE} & (L > T_{av} + T1) \end{cases} \quad (1)$$

ここで、 $\hat{S}$  は RRC 状態の推定値、 $T_{av}$  は通信の継続時間の平均値を示す。図に示すように、1 回 1 回の通信の継続時間と、 $T_{av}$  には差が存在するため、その差が推定誤差となる。

端末からサーバへの通信開始の通知は、端末が RRC\_CONNECTED に遷移した直後に実施される。したがって、通知による RRC 状態への影響(通知によって RRC 状態が RRC\_IDLE からの遷移が発生する、もしくは、RRC\_CONNECTED 状態の時間が長くなる)はほとんど無視できる。すなわち、端末から見た場合の通知そのものによる負荷、たとえば電力消費量の増大などはほとんど無視できる。

一方で、逐次通知方式は端末が通信を行うたびにサーバに通知を行うため、サーバの負荷が課題となる。個々の端末からの通知は、1 時間に数回から数十回程度であるが、サーバは多数の端末を収容することを想定している。そのため、サーバが受信する通知数は膨大なものとなり、IP プッシュのメッセージを配信する負荷に比べて、推定のための通知を受信する負荷(通知負荷)が無視できないものとなる。次に述べるガードタイム通知方式は、推定精度をできるだけ維持しつつ、通知によるサーバ負荷を抑制する方式である。

### 2.2 ガードタイム通知方式

ガードタイム通知方式は、端末が RRC 状態推定のために

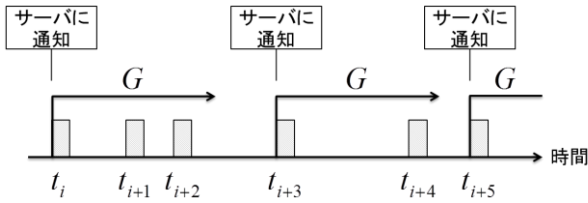


図 3 ガードタイムによる通知抑制

送信する通知の回数を削減することで、サーバにおける通知受信負荷を低減する。

ガードタイム通知方式では、ガードタイム $G$ を導入して、通知負荷を抑制する。すなわち、プッシュサーバに通知を行った直後は、ガードタイムが経過するまでは、IP 通信を開始しても通知を実施しない。図 3 は、ガードタイム通知方式における端末からサーバへの通知方法を示した図である。図では、端末が通信を行った時間（図中の四角）、ガードタイムと通知有無について記載している。なお、端末が通信を開始した時刻を $t_i$ としている（ $i$ は正数）。図に示すように、ガードタイム通知方式では、一度通知を行った後、ガードタイムが経過する前に新たな通信が発生してもサーバに通知を行わない（図中の $t_{i+1}$ 、 $t_{i+2}$ 、 $t_{i+4}$ における通知は抑制される）。

以上のように、ガードタイム通知方式では、サーバへの通知を制限するため、RRC 状態の推定に式(1)は利用できなくなる。代わりに、ガードタイム通知方式では、直近の端末の通信頻度を基に RRC 状態を確率的に推定する。時刻 $t_i$ における通信頻度 $q_i$ は、端末が行った通信間隔 $d_i$ の指数平滑移動平均 $D_i$ をもちいて、算出する。

$$d_i = t_i - t_{i-1} \quad (2)$$

$$D_i = rd_i + (1 - r)D_{i-1} \quad (3)$$

$$q_i = \frac{1}{D_i} \quad (4)$$

なお、 $r$ は移動平均の重みを表す。式(4)はある事象が発生する平均時間間隔と平均発生頻度は逆数の関係にあることを用いている。 $q_i$ は、端末がサーバに行く通知の際に、一緒に送信することで、サーバ側で随時知ることができる。端末が $k$ 回目（ $k$ は正数）にサーバに通知を行った時刻を $T_k$ とし、 $T_k$ において通知された通信の発生頻度を $\lambda_k$ とする。すなわち、

$$\lambda_k = q_i \quad (5)$$

とする。サーバは、通知された $\lambda_k$ を基に図 4 のように端末が RRC\_CONNECTED である確率値を算出する。時刻 $t$ において端末の RRC 状態が RRC\_CONNECTED である確率を $P(t)$ とすると、 $P(t)$ は、 $t - T1$ から $t$ までの間に端末が通信を行った確率を算出すればよい。 $P(t)$ は通信頻度 $\lambda_k$ を用いて、下記のように算出できる[7]。

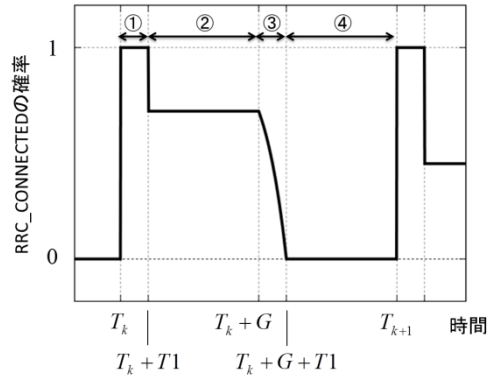


図 4 ガードタイム通知方式における RRC 状態の推定法

$$P(t) = \begin{cases} 1 & (T_k \leq t \leq T_k + T1) \\ 1 - \exp(-\lambda_k T1) & (T_k + T1 \leq t \leq T_k + G) \\ 1 - \exp[-\lambda_k \{(T_k + G) - (t - T1)\}] & (T_k + G \leq t \leq T_k + G + T1) \\ 0 & (T_k + G + T1 \leq t \leq T_{k+1}) \end{cases} \quad (6)$$

上記、式の場合分けの 1 行目から 4 行目は、図 4 の①から④の領域の推定値にそれぞれ対応している。①の部分は、通知を受けた直後はサーバ側で端末が過去 $T1$ の間に通信を行ったと判定できることから確率値が 1 となる。②の部分は、通信頻度が $\lambda_k$ のもとで過去 $T1$ の間に端末が通信を行う確率となっている。また、③④の部分に関しては、ガードタイムを過ぎた後の確率値となっている。ガードタイムが過ぎた後に端末から通知がない場合は、ガードタイムを過ぎた後に端末が通信を行っていないと判定できることを利用して確率値を算出している。

### 3. 提案方式

本章では、ガードタイム通知方式を効率化する方式を提案する。提案方式では、ガードタイムを端末の通信頻度に応じて設定することで、RRC 状態の推定精度を維持しつつ、サーバ負荷の低減をはかる。以下、従来のガードタイム通知方式の課題と、提案方式について詳細に説明する。

ガードタイム通知方式は、ガードタイムを設けて、端末からサーバへの通知を制限することで、RRC 状態の推定に必要な通知を受信するためのサーバ負荷を抑制する。ガードタイムを長くすればするほど、抑制される通知の数は増えるため、サーバ負荷は低減する。一方で、RRC 状態を確定的に推定できない時間（ $P(t)$ が 0 もしくは 1 とならない時間、すなわち、図 4 の②③の部分）が増加するため、RRC 状態の推定精度は悪化する。すなわち、サーバ負荷と RRC 状態の推定精度はトレードオフの関係にある。そのため、ガードタイムの設定は、ガードタイム通知方式において、重要な要素となる。

ここで、ガードタイムとサーバの通知負荷の関係について考察する。サーバの通知負荷は、通知回数に比例する。

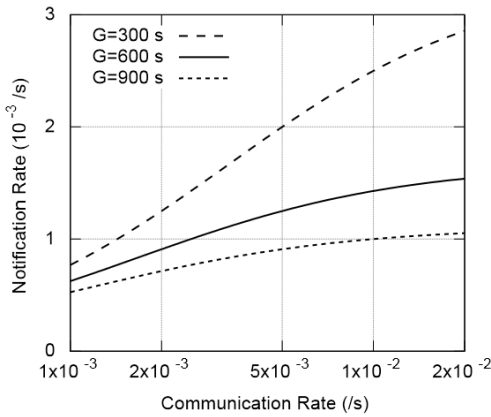


図 5 通信頻度と通知回数との関係

1 端末が単位時間あたりに行う通知回数  $z$  は、ガードタイムと端末の単位時間あたりの通信頻度を用いて、

$$z = \frac{1}{G + \frac{1}{\lambda}} \quad (7)$$

となる。式(7)の分母は、1 端末が行う通知の平均時間間隔となっており、その逆数で  $z$  を求めることができる。図 5 はガードタイムを 300, 600, 900 秒と設定した際に、通信頻度  $\lambda$  と通知回数  $z$  の関係を表した図となる。ガードタイムを長く設定することで、通知回数を抑制することができる。一方で、ガードタイムが一定の場合、通信頻度が高くなるにしたがい、通知回数が増加することがわかる。つまり、従来方式では、ガードタイムは定数としていたので、夕方から夜間にかけてなどの、端末が頻繁に通信する繁忙時間帯には、サーバの通知負荷が増大することを意味している。したがって、従来方式では、サーバの通知負荷がピークとなる繁忙時間帯に合わせて、通知負荷が許容範囲に収まるようにガードタイムを設定することとなる。このように設定されたガードタイムのもとでは、端末があまり通信を行わない閑散時間帯において、サーバの処理能力に余力があるにも関わらず、端末からの通知が無駄に抑制されてしまうという課題が発生する。その結果、通知が抑制された分だけ推定精度が悪化する。

そこで、提案方式では、端末の通信頻度  $\lambda$  に合わせてガードタイムを動的に設定する。時刻  $T_k$  において通知を行った時に適用するガードタイムを  $G_k$  とすると、 $G_k$  を下記の式により算出する。

$$G_k + \frac{1}{\lambda_k} = \Lambda \quad (8)$$

ここで、 $\Lambda$  は定数とする。式(8)を式(7)に代入すると、

$$z = \frac{1}{\Lambda} \quad (9)$$

となり、サーバへの通知回数が、通信頻度に依存しなくなることがわかる。 $1/\Lambda$  は期待されるサーバへの単位時間あたりの通知回数なので、 $\Lambda$  はサーバの通知負荷を調整する

パラメータ (負荷調整パラメータ) となる。このように、提案方式では、負荷調整パラメータ  $\Lambda$  を設定し、それに基づきガードタイムを動的に設定する。通信頻度が高い場合はガードタイムが長く、通信頻度が低い場合はガードタイムが短く設定されるため、従来方式のように、閑散時間帯において無駄に通知が抑制されることはない。負荷調整パラメータ  $\Lambda$  は、あらかじめサーバに設定しておく値とし、なんらかの方法で端末に通知されるものとする。なお、 $\lambda_k$  が小さい場合に  $G_k$  が負となるが、その場合は、 $G_k = G_{min}$  ( $G_{min}$  は正数) とする。

端末は、サーバに対して通知を行う時に、 $\lambda_k$  と  $\Lambda$  を用いてガードタイム  $G_k$  を計算して、次回通知時まで  $G_k$  を適用して通知を抑制する。サーバは端末から通知された通信頻度  $\lambda_k$  から、 $G_k$  を計算して、RRC 状態の確率値の算出する。確率値の計算は、従来と同様の式が利用可能である。すなわち、式(6)において  $G$  を  $G_k$  に置き換えた下記の式により算出する。

$$P(t) = \begin{cases} 1 & (T_k \leq t \leq T_k + T1) \\ 1 - \exp(-\lambda_k T1) & (T_k + T1 \leq t \leq T_k + G_k) \\ 1 - \exp[-\lambda_k \{(T_k + G_k) - (t - T1)\}] & (T_k + G_k \leq t \leq T_k + G_k + T1) \\ 0 & (T_k + G_k + T1 \leq t \leq T_{k+1}) \end{cases} \quad (10)$$

## 4. 提案方式の評価

本章では、提案方式の評価について述べる。まず、評価方法について述べ、その後、評価結果について述べる。なお、逐次通知方式に関しては、通知負荷が高くなり、実運用が非現実的となるため、評価からは除外する。以降、従来方式を固定ガードタイム通知方式、提案方式を可変ガードタイム通知方式と記載する。

### 4.1 評価方法

表 1 は評価に用いた各種条件をまとめたものである。評価に用いた端末は Android 4.0 端末で、約 9 日間の IP 通信のデータを収集した。モバイル網は LTE とし、RRC\_CONNECTED から RRC\_IDLE へ遷移する際の Inactivity Timer 値は 120 秒としている。式(4)における通信間隔の指数平滑移動平均の重み  $r$  は 0.5 とした。なお、最適

表 1 評価における諸条件

| 項目                          | 条件値          |
|-----------------------------|--------------|
| 端末                          | Android 4.0  |
| データ収集期間                     | 約 9 日        |
| モバイル網                       | LTE          |
| Inactivity Timer 値 ( $T1$ ) | 120 s        |
| 移動平均の重み ( $r$ )             | 0.5          |
| ガードタイム最小値 ( $G_{min}$ )     | 10 s         |
| 通知負荷許容量 ( $L_{max}$ )       | 0.001-0.1 /s |

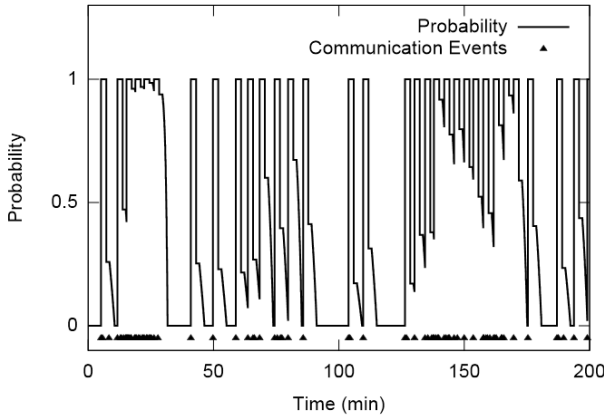


図 6 固定ガードタイム通知方式における状態推定

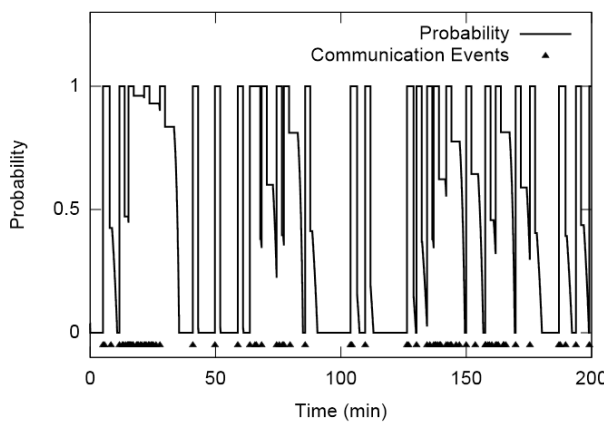


図 7 可変ガードタイム通知方式における状態推定

な $r$ の値については、検討を要するが、 $r \approx 0.5$ において良好な推定精度が得られたため、ここでは、0.5としている。

評価では、サーバが許容する通知負荷（通知負荷許容値 $L_{max}$ ）を設定し、その際の推定誤差を評価した。通知負荷許容値は、単位時間に受信する1端末あたりの通知数とする（サーバが収容する端末数を乗ずることで、サーバが受信を許容する秒間通知数となる）。通知負荷許容値を超えない通知負荷となるように、固定ガードタイム通知方式においてはガードタイム $G$ を、可変ガードタイム通知方式においては負荷調整パラメータ $\Lambda$ をそれぞれ設定した。すなわち、1端末あたり・単位時間あたりの通知回数 $z$ の最大値 $z_{max}$ とすると、 $z_{max} = L_{max}$ となるように、 $G$ と $\Lambda$ を設定する。固定ガードタイム通知方式では、通知回数 $z$ は、式(7)より、端末の通信頻度 $\lambda$ に対して単調増大となる。 $\lambda$ が十分大きい時は、 $z \approx 1/G$ となるため、 $z_{max} = 1/G$ となる。可変ガードタイム通知方式では、式(9)が成り立つため、 $z_{max} = 1/\Lambda$ となる。すなわち、通知負荷許容値 $L_{max}$ に対して、 $G$ と $\Lambda$ を下記のように設定する。

$$G = \Lambda = \frac{1}{L_{max}} \quad (11)$$

なお、評価においては、取得した端末の通信ログを基に、

提案方式のアルゴリズムで通知を実施したと仮定して、サーバにおいて得られる推定値を仮想的に計算している。

評価では、取得した通信ログと Inactivity Timer の値を用いて、時刻 $t$ における端末の RRC 状態 $R(t)$ を算出した。

$$R(t) = \begin{cases} 1 & (S = \text{RRC\_CONNECTED}) \\ 0 & (S = \text{RRC\_IDLE}) \end{cases} \quad (12)$$

$R(t)$ は推定値ではなく、実際の RRC 状態値と考えることができる。次に、推定方式のアルゴリズムを用いて RRC\_CONNECTED の確率 $P(t)$ を計算し、 $R(t)$ と $P(t)$ を比較することにより、推定誤差を評価する。推定誤差 $E$ は下記の式により算出した。

$$E = \frac{1}{T} \int_0^T |R(t) - P(t)| dt \quad (13)$$

なお、時刻 $t$ はデータ収集開始時刻を 0、終了時刻を $T$ と定義する。推定誤差 $E$ は、実際の RRC 状態と推定した状態の確率値の差分の時間平均となっている。

#### 4.2 評価結果

固定ガードタイム通知方式、可変ガードタイム通知方式における RRC 状態の推定結果を図 6、図 7 にそれぞれ示す。図において、実線は推定アルゴリズムを用いて推定した RRC\_CONNECTED の確率値を示しており、三角の点は端末が通信を行った時間を示している。ここで、ガードタイムの長さを変更することによって、長さが変わる区間、すなわち、図 4 の②の区間に着目する。②の区間は、状態の推定値が 0 もしくは 1 にならない区間となっている。まず、両者に共通して言えることは、端末の通信頻度が高い時間帯においては、推定した確率値が高く推定されている点である。直近の通信頻度を用いて状態の確率値を評価することで、端末の通信頻度の変動に追従して、RRC 状態を推定できていることがわかる。固定ガードタイム通知方式では、②の区間の時間幅は常に一定となっている。一方で、固定ガードタイム通知方式では、②の区間は通信頻度が高い時間帯では長く、通信頻度が低い時間帯では短くなっており、端末の通信頻度にあわせて、ガードタイムを設定できていることがわかる。

評価によって得られた通知負荷許容値と推定誤差の関係を図 8 に示した。可変ガードタイム通知方式の方が推定誤差を抑えられていることがわかる。例えば、通知負荷許容値が 0.003 ( $G = \Lambda = 300$ ) の場合、可変ガードタイム通知方式の推定誤差は、固定ガードタイム通知方式の 1/3 となっている。可変ガードタイム通知方式の推定誤差が小さくなるのは、可変ガードタイム通知方式が、通知負荷許容量の範囲内でガードタイムを下げるようにしているためである。図 9 は、通知負荷許容量とガードタイムの関係を示した図である。固定ガードタイム通知方式は、式(11)によって計算される値、可変ガードタイム通知方式は、式(8)で計算されたガードタイムの平均値を図に示している。図によれば、可変ガードタイム通知方式の方が、ガードタイ

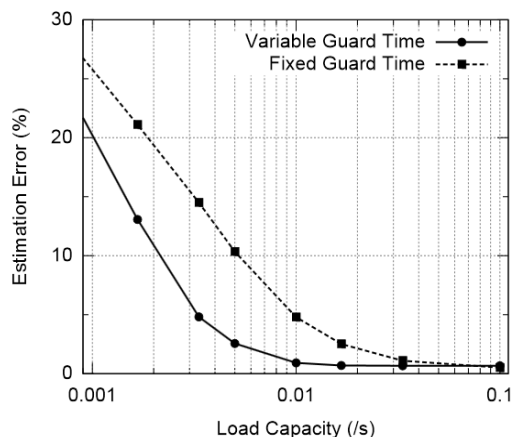


図 8 通知負荷許容値と推定誤差

ムが小さくなっていることがわかる。前述したように、ガードタイムが短い方が、推定誤差は小さくなるため、可変ガードタイム通知方式が固定ガードタイム通知方式に比べ良好な結果が得られる。

## 5. おわりに

本稿では、プッシュサーバにおいて、端末の RRC 状態を推定する際の推定精度を向上する方式について提案を行った。従来方式では、RRC 状態の推定に必要な端末からの通知を固定的なガードタイムを用いて抑制し、サーバの通知受信負荷を抑えていた。本稿では、新たに端末の通信頻度に応じてガードタイムを動的に設定する方式を提案した。提案方式によって、従来方式と同程度の負荷で、より高い推定精度を得ることができた。今後の課題としては、多数のユーザによるサンプルを用いた RRC 状態の推定精度の評価や、推定した状態に基づくメッセージ配信制御による遅延抑制効果について評価を行うことがあげられる。

## 参考文献

- 1) Google: Google Cloud Messaging for Android, Android Developers(online), available from <<http://developer.android.com/google/gcm/index.html>> (accessed 2013-9-13).
- 2) Apple: Local and Push Notification Programming Guide: Apple Push Notification Service, iOS Developer Library(online), available from <<https://developer.apple.com/library/ios/documentation/NetworkingInteNetw/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html>> (accessed 2013-9-13).
- 3) 3GPP TS 36.331, Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification, V8.17.0, 2012.
- 4) Huang, J., Qian, F., Gerber, A., et al.: A close examination of performance and power characteristics of 4G LTE networks, *Proc. 10th international conference on Mobile systems, applications, and services(MobiSys '12)*, pp 225-238, ACM, 2012.
- 5) 大西健夫, 城島貴弘, 中島一彰: サーバプッシュにおけるモバイル端末の RRC 状態を考慮したメッセージ配信遅延抑制方式, 情報処理学会研究報告, 2013-DPS-154(39), pp.1-6, 2013.

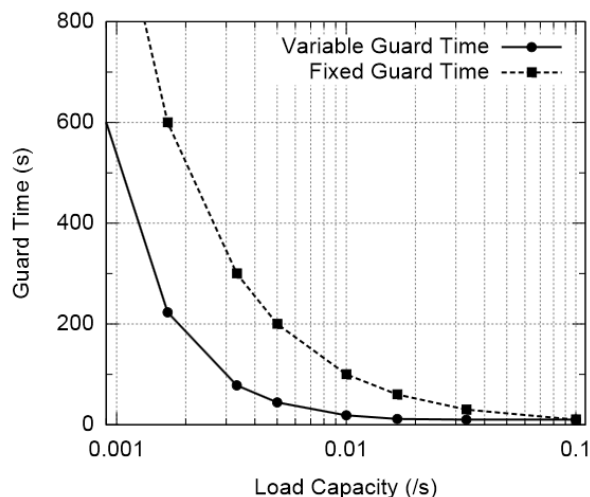


図 9 通知負荷許容量とガードタイム

<http://office.microsoft.com/ja-jp/word-help/CH010097020.aspx>

- 6) 城島貴弘, 大西健夫: モバイル端末における RRC 状態遷移に伴う通信遅延を考慮したメッセージング方式の提案, 信学技報, vol. 112, no. 463, NS2012-225, pp. 349-354, 2013.
- 7) 大西健夫, 城島貴弘: サーバプッシュにおけるメッセージ配信処理スケジューリングのための RRC 状態推定方式, 信学技報, vol. 113, no. 304, MoNA2013-47, pp. 35-40, 2013.