Vol.2013-DPS-154 No.39
Vol.2013-CSEC-60 No.39
2013/3/15

# 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

大西健夫[†1]　城島貴弘[†1]　中島一彰[†1]

スマートフォンの利用拡大に伴い，端末上のアプリケーションに対してイベント通知などのメッセージをクラウド上のサービスから任意の時点で配信するプッシュ配信が注目されている．しかし，3G や LTE などのモバイル網では，端末の消費電力削減のため，無線のリソース確保状態(RRC 状態)を遷移させており，RRC 状態によってはプッシュメッセージの配信完了までに数秒単位の大きな遅延が発生する。また，プッシュ配信に利用する TCP 接続を維持するための Keep-Alive 信号がモバイル網に輻輳を引き起こし，問題となっている．本稿では，Keep-Alive 信号の不要な配信時接続確立方式のプッシュサーバにおいて，端末の RRC 状態に応じた配信処理スケジューリングを実施することで，配信遅延の抑制を実現する手法を提案し，評価実験により，配信遅延を約 15%抑制できることを示す．

# Latency reduction method for message delivery of server push in which RRC state of mobile terminal is considered

TAKEO ONISHI[†1]　TAKAHIRO SHIROSHIMA[†1]
KAZUAKI NAKJIMA[†1]

Push Notification, in which a message of event notice is delivered anytime from services on the cloud to applications on mobile terminals, attract rising attention according with expansion of smart phone. A transition of the radio resource control state (RRC state) aim at power consumption reduction in a mobile terminal with a mobile network such as 3G and LTE. But the transition of RRC state cause the delay with the several seconds on delivering a push message. Moreover, Keep-Alive signals which keep TCP connections used in Push Notification cause congestion in mobile networks. This paper proposes a scheduling method of delivery processing according to the RRC state in a messaging server which does not need Keep-Alive signals. It is shown that our method can reduce the delivery delay by 15 %.

## 1. はじめに

近年，移動体通信技術の進歩により，移動時にも高速通信が可能なスマートフォンの普及が拡大している．スマートフォンの特徴として，ユーザが任意のアプリケーションをインストールし，利用できることがあげられる．アプリケーションの中には、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，クラウド上のサーバからスマートフォンに対してメッセージをリアルタイムにプッシュ配信するプッシュサービスの利用が広まっている．モバイル端末向けのプッシュサービスとして SMS によるテキストメッセージが利用されてきたが，SMS はシグナリングチャネルを利用したサービスであるため一般のアプリケーション開発者が扱うには難しく，また、モバイル網に接続された端末でしか利用できないという問題があった．そこで，プッシュサービスを IP ネットワーク上で提供することにより，アプリケーション開発者が容易に利用でき，かつ，IP 通信が可能な幅広い端末で利用可能な IP プッシュが提供されている(図 1)[1][2]．IP プッシュでは，アプリケーションサーバ上で発生したイベント（コンテンツの更新など）をプッシュサーバ経由で TCP/IP やその上位レイヤの HTTP などを利用してモバイル端末上のプッシュクライアントに対して通知し，アプリケーションサーバと連携して動作するモバイル端末上のアプリケーションにイベントの発生をメッセージとして通知する．

モバイル網に接続されたモバイル端末の IP 通信は，W-CDMA，LTE といった高速パケット無線アクセスの規格に則り実現されている．連続稼働時間が重視されるモバイル端末では，電力消費量を抑えるために，通信の必要がない場合は確保した RAB(Radio Access Bearer)を解放し無線通信を休止させ，通信を行う際には RAB を確保して無線通信を再開する，といった RAB の確保状態を制御している．この制御は RRC（Radio Resource Control）[1]として規定されており，RAB の確保状態が RRC 状態として定義されている．RRC 状態間の遷移にはモバイル端末と無線局との間で RRC message とよばれるシグナリングが必要であり，このシグナリングに要する時間がサーバ・端末間での通信遅延という形で現れる．RRC の状態遷移は IP 通信の下位層で自動的に実施されるため，モバイル端末上のアプリケーションは必ずしも意識する必要はないが，IP プッシュにおいてはサーバからモバイル端末に対してメッセージを配


図 1　IP プッシュ

†1 日本電気株式会社
NEC Corporation.

信する際の遅延として現れるため，プッシュ配信に即時性を求められる場合に問題となる．

プッシュサーバからモバイル端末にプッシュメッセージを配信するには，一般に NAT などのミドルボックスが介在するため，モバイル端末からプッシュサーバに対して確立した TCP 接続を常時維持する（TCP 接続維持方式）．TCP 接続の維持には，定期的な Keep-Alive 信号の交換が必要となり，その際に発生する RRC 状態遷移によるシグナリングがモバイル網の輻輳の原因となっている[4][5]．また，モバイル端末の電力消費量を増大させることにもなる[6]．一方，通信事業者がプッシュサービスを導入する場合には，プッシュサーバをモバイル網内部に導入し，モバイル端末と同一のアドレス帯域内に所属させることで，プッシュサーバから直接モバイル端末に IP 接続を行ってメッセージを配信することができる（配信時接続確立方式）．配信時接続確立方式では，Keep-Alive 信号が不要となり，モバイル網の負荷低減やモバイル端末の電力消費量削減が可能となる．

本稿では，配信時接続確立方式において，端末の RRC 状態に基づくプッシュメッセージの配信制御を実施することで，メッセージの端末への配信遅延を抑制する方式を提案する．

## 2. モバイル端末における RRC 状態

図 2 は、W-CDMA における RRC 状態を示した図である．図中に示した状態の内，上部に行くほど通信遅延が小さいが，消費電力が大きな状態となる[6]．以下、それぞれの状態について説明する。

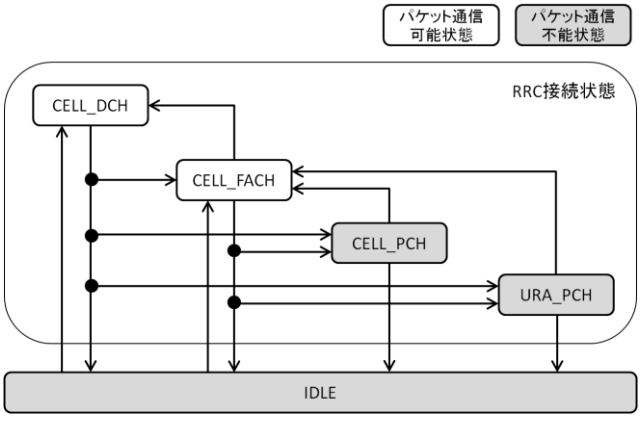


図 2　W-CDMA における RRC 状態

- CELL_DCH
  端末に個別物理チャネルが割り当てられ，5 つの状態の内で最も高いスループットが得られ，また，最小の遅延が得られる．
- CELL_FACH
  複数の端末が物理チャネルを共有する．CELL_FACH は通信量が少量の場合のみに用いられる．
- CELL_PCH
  IP 通信は不可能な状態で，着信確認のための間欠受信（DRX）のみ行う．RRC 状態の切り替えのための RRC コネクションは維持された状態である．
- URA_PCH
  CELL_PCH とほぼ同じであるが，モバイル端末の管理が基地局単位ではなく，基地局を幾つか束ねたエリア単位となる．モバイル端末を持つユーザが高速移動している際に用いられる状態である．
- IDLE
  IP 通信は不可能な状態で，着信確認のための DRX のみ行う．CELL_PCH と異なり，RRC コネクションが開放された状態であり，CELL_PCH に比べて，状態遷移時に必要な RRC message 数が多い．

上記のように，RRC 状態には IP 通信が可能な状態と不可能な状態が存在する．IP 通信不可能な状態にある端末が IP 通信を行うには，RRC 状態を IP 通信が可能な状態に切り替える必要があり，切り替えに必要な時間が通信遅延時間として現れる．切り替えに必要な時間は端末と無線局との間で交換する必要のある RRC message の数が多いほど長くなる．表 1 にモバイル網で実測した RRC 状態遷移に伴う遅延時間を示す．CELL_PCH から CELL_DCH に切り替えには 2 秒程度の時間が必要であった．より多くの RRC message の交換が必要となる IDLE から CELL_DCH へ切り替えには，3 秒程度の時間を要していた．なお，IDLE から CELL_FACH，URA_PCH から CELL_DCH などの遷移は観測されなかったため，表 1 には示していない．

図 3 は，LTE における RRC 状態を示した図である．以下、それぞれの状態について説明する．

- RRC_CONNECTED
  IP 通信が可能な状態で，RRC コネクションが確立された状態である．
- RRC_IDLE
  IP 通信は不可能な状態で，着信確認のための DRX のみ行う．RRC コネクションが開放された状態である．

LTE は 2 状態のみとなっており，状態数を簡略化することで RRC message 数を削減し，RRC 状態遷移の高速化を図っている[7]．RRC_IDLE から RRC_CONNECTED への遷移は 500 ミリ秒から 1 秒程度と W-CDMA よりは高速化されているものの，大きな遅延が発生することになる．

表 1　RRC 状態の遷移による遅延

| 遷移前の状態 | 遷移後の状態 | 遅延時間 |
|---|---|---|
| CELL_PCH | CELL_DCH | 2 秒 |
| IDLE | CELL_DCH | 3 秒 |
| RRC_IDLE | RRC_CONNECTED | 0.5 - 1 秒 |

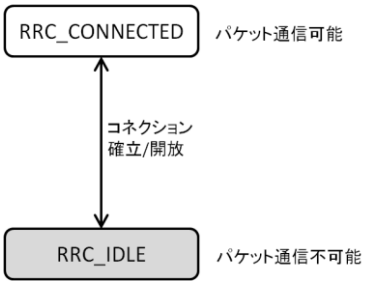


図 3　LTE における RRC 状態

## 3. メッセージプッシュ配信

本章では，IP プッシュの方式として，TCP 接続維持方式と配信時接続確立方式の二つの方式について説明する．通信事業者が IP プッシュを導入する際には，配信時接続確立方式がモバイル網にかける負荷を抑えられる点で優れている．

### 3.1 TCP 接続維持方式

TCP 接続維持方式では，モバイル端末からプッシュサーバに対して TCP 接続を確立し，その後 TCP 接続を維持しておく．プッシュサーバからモバイル端末にメッセージを送信する時は，プッシュサーバとモバイル端末間で確立済みの TCP 接続を用いてメッセージを送信する．GCM（Google Cloud Messaging）[1]などがこの方式を用いている．一般的には，TCP 上で動作する HTTP ロングポーリング[8]や WebSocket[8][9]などを用いて実装される．TCP 接続維持方式の利点は，プッシュサーバとモバイル端末の間に，NAT やファイアーウォールなどのミドルボックスが介在する場合にも，メッセージを配信可能な点である．

一方で，TCP 接続維持方式はモバイル網の輻輳を引き起こしたり，モバイル端末の消費電力が増大したりするという問題がある．これらの問題は，TCP 接続の維持に必要となる Keep-Alive 信号に起因している．また，プッシュサーバが多数の TCP 接続を維持した状態となるため，プッシュサーバの負荷が高くなるという問題もある。

### 3.2 配信時接続確立方式

配信時接続確立方式は，TCP 接続維持方式と異なり，モバイル端末がプッシュサーバからの接続用ポートを開いておき，メッセージ配信時にプッシュサーバがモバイル端末に接続してメッセージを配信する方式である．

TCP 接続維持方式と異なり，Keep-Alive 信号が不要なため，モバイル網に輻輳を引き起こさず，電力消費量も抑えられる．一方で，プッシュサーバとモバイル端末の間にミドルボックスが存在せず，同一のアドレス体系に属している必要があるという問題があるが，通信事業者がプッシュサーバを導入する場合は，プッシュサーバをモバイル網内に導入することで条件を満たすことができる．そのため，配信時接続確立方式は，通信事業者がプッシュサービスを導入する場合に適した方式であると言える．

本稿では、配信時接続確立方式における遅延削減方式を提案する．

## 4. 提案方式

本章では，モバイル端末の RRC 状態に応じて通信遅延が異なることに着目し，配信時接続確立方式のプッシュサーバにおけるメッセージ配信処理をスケジュールすることで，プッシュメッセージ配信遅延を低減する方式を提案する．

### 4.1 配信時接続確立方式の処理

図 4 は，配信時接続確立方式におけるメッセージ配信時の処理の流れを示している．プッシュサーバはメッセージ配信要求を受け取ると，端末情報データベースにアクセスし，宛先となる端末の情報を取得する（①）．この時，取得する情報はトリガを宛先となる端末に送信するための IP アドレスなどの情報である．

次に，プッシュサーバは配信するメッセージをメッセージデータベースに保存する（②）．その後，宛先となっている端末に対して，配信するメッセージが存在することを示すトリガパケットを送信する（③）．トリガパケットを受信した端末はプッシュサーバにメッセージ取得要求を送信する（④）．プッシュサーバは負荷分散のために複数台で構成されていてもよく，その場合，メッセージ取得要求は，いずれかのプッシュサーバに割り振られる．メッセージ取得要求を受信したプッシュサーバは，メッセージデータベースを参照し，配信するメッセージを取得する（⑤）．最後に，プッシュサーバはメッセージ取得要求に対する応答として，メッセージを端末に送信する（⑥）．

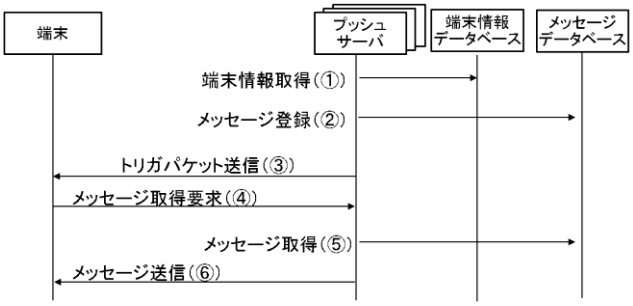


図 4　メッセージ配信処理

上記で述べたメッセージ配信における遅延の要因としては，プッシュサーバにおけるメッセージ配信処理遅延とプッシュサーバ・端末間の通信遅延があげられる．メッセージ配信処理遅延はデータベースからの端末情報取得処理や，データベースへのメッセージ登録処理などに起因する遅延であり，個々の処理遅延は小さいが，メッセージ配信要求が短時間に集中した場合に数秒程度の通信遅延に比べて無視できない大きさの遅延となる．

プッシュサーバ・モバイル端末間の通信遅延は，プッシ

ュサーバから端末に送信されるトリガパケットの通信時間，モバイル端末からプッシュサーバに送信されるメッセージ取得要求の通信時間，および，プッシュサーバからモバイル端末へ送信されるメッセージの通信時間から構成される．このうち，トリガパケットの送信時に発生する遅延が最も大きな値になりうる．なぜなら，RRC 状態が通信不可能な状態のモバイル端末に対してトリガパケットを送信しようとすると，RRC 状態の遷移が発生するため，RRC 状態の遷移に要する時間が通信遅延に加算されるためである．例えば，IDEL 状態のモバイル端末に対してトリガパケットを送信しようとすると，CELL_DCH への遷移が発生し，トリガパケットがモバイル端末に到達するまでに数秒の時間を要することになる．その後のメッセージ取得要求送信時やメッセージ送信時には，モバイル端末の RRC 状態が既に通信可能状態に遷移しているため，RRC 状態の遷移は発生せず，おおよそ数百ミリ秒程度の遅延となる．

以上のように，トリガパケットの送信時に大きな遅延が発生するため，IDLE 状態などのパケット通信が不可能な RRC 状態になっているモバイル端末に対するメッセージ配信遅延は大きなものとなる．それに加え，プッシュサーバで多数のプッシュメッセージ配信処理が発生すると，プッシュサーバでの処理遅延が増大し，RRC 状態遷移による遅延と合わせると，より大きなメッセージ配信遅延が発生する．

### 4.2 トリガ先行方式

前章で述べた通り，RRC 状態の遷移に伴う遅延に配信処理遅延が加わると大きな遅延が発生する．そこで，RRC 状態の遷移とメッセージ配信の為の処理を並列に実施させることで，遅延の抑制を図る方式が考えられる．

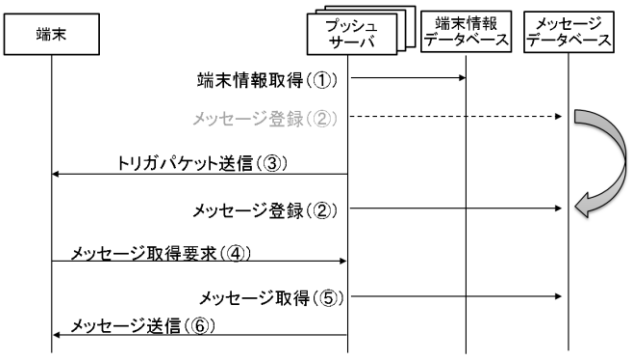


図 5　トリガ先行方式のシーケンス

図 5 のようにトリガパケット送信処理を先に実施する（トリガ先行方式）．トリガパケットを送信後，モバイル端末からのメッセージ取得要求が到着するまでには，数百ミリ秒から数秒程度の時間的余裕が存在するため，その間にメッセージ登録処理を行う．トリガパケットの送信を先行して行うことで，モバイル端末の RRC 状態遷移が早期に誘発されるようになるため，メッセージの配信遅延が低減することが期待できる．

処理順序を後にしたメッセージ登録処理は，モバイル端末からのメッセージ取得要求が来るまでには完了しなければならない．トリガ送信後メッセージ登録処理が完了していない場合，メッセージデータベースを参照しても，配信すべきメッセージが登録されていない．そのため，メッセージが登録されるまでメッセージの配信を待たなければならず，メッセージの配信遅延が増大することとなる．

トリガ送信後，モバイル端末からのメッセージ取得要求が到達するまでの時間は最短で 100 ミリ秒程度（RRC_CONNECTED における通信時間）なので，100 ミリ秒でメッセージ登録処理が終了するようにすればよい．

### 4.3 RRC 状態スケジューリング方式

トリガ先行方式では，メッセージ登録処理をトリガ送信後，固定時間（100 ミリ秒）後までとした．しかし，モバイル端末の RRC 状態は IDLE の場合もあり，その場合は，メッセージ登録処理をさらに遅らせることができ，その分，他のトリガ送信処理を優先して実施することが可能になる．

そこで，提案方式では，RRC 状態を考慮して，トリガ送信処理を優先的に実施しつつ，モバイル端末からのメッセージ取得要求が来るまでにメッセージ登録処理が完了するように，処理をスケジューリングする（RRC 状態スケジューリング方式）．メッセージ登録処理をデッドラインスケジューリングによりスケジューリングする．メッセージ登録処理のデッドラインとなる時刻（T）は下記の式により決定する．

$$T = t + D_T(RRC_T) + D_R(RRC_R) \quad (1)$$

ここで，t はトリガを送信した時刻，$D_T(RRC_T)$はトリガパケットがプッシュサーバからモバイル端末に到達するまでの時間，$D_R(RRC_R)$はメッセージ取得要求がモバイル端末からプッシュサーバに到達するまでの時間である．$D_T$ はトリガパケット受信前のモバイル端末の RRC 状態（$RRC_T$ と表記）に，$D_R$ はメッセージ取得要求送信前のモバイル端末の RRC 状態（$RRC_R$ と表記）に依存する値である．トリガパケットを受信することにより，RRC 状態は遷移するため，$RRC_T$ と $RRC_R$ は一致するとは限らない．表 2 にモバイル網において実測した RRC 状態の遷移を示す．デッドラインスケジューリングにより，上記(1)式で与えられたデッドラインに近いメッセージ登録処理を優先的に実行することで，メッセージ取得要求がプッシュサーバに届いた時には，メッセージ登録処理が完了した状態となるように処理をスケジューリングする．なお，端末情報取得・トリガパケット送信処理の優先度は中程度とし，デッドラインに近いメッセージ登録処理が存在しない場合は，優先的にトリガパケットの送信が行われるようにする．

Vol.2013-DPS-154 No.39
Vol.2013-CSEC-60 No.39
2013/3/15

表 2　RRC 状態の遷移

| トリガパケット受信前 ($RRC_T$) | トリガパケット受信後 ($RRC_R$) |
|---|---|
| CELL_DCH | CELL_DCH |
| CELL_FACH | CELL_FACH |
| CELL_PCH | CELL_DCH |
| IDLE | CELL_DCH |
| RRC_CONNECTED | RRC_CONNECTED |
| RRC_IDLE | RRC_CONNECTED |

以上のようにすることで，プッシュサーバで多数のメッセージ配信処理が発生していても, 早期に RRC 状態の遷移を誘発することができる．また，モバイル端末からのメッセージ取得要求が届いた時には，メッセージ登録処理が完了しており，即座にメッセージを送信することが出来るため，メッセージ配信遅延を低減することができると期待される．

## 5. 提案方式の評価

提案方式について，シミュレーションによる評価を実施した．本章では，評価方法と結果について述べる．

### 5.1 評価方法

多数のモバイル端末を利用した評価が困難であったため, モバイル網での通信遅延を再現する基地局シミュレータと端末シミュレータを用いてシミュレーションによる評価を実施した．

図 6 に評価システムの構成を示す．配信サーバは，メッセージのプッシュ配信を行うプッシュサーバである．端末情報（メッセージ配信対象端末の IP アドレスやポートなど），および，メッセージを保存するデータベースには，KVS(Couchbase Server 2.0[10])を利用した．配信要求プログラムは，端末シミュレータへのプッシュメッセージ配信を要求するメッセージ配信要求を配信サーバに送信する．配信サーバ，KVS，および，配信要求プログラムは同一 PC 上に配置した．配信要求プログラムは，端末シミュレータへのプッシュメッセージ配信要求を配信サーバに送信する．基地局シミュレータは，配信サーバと端末シミュレータの間に存在し，モバイル網の通信遅延を再現する．通信遅延の発生には，netem[11]を利用した．端末シミュレータは，配信サーバからメッセージを受信し，メッセージを受信した時間を記録する．

基地局シミュレータで発生させた通信遅延の詳細について説明する． 表 3 に，基地局シミュレータで発生させた RRC 状態毎の片道通信遅延を示す. 遅延値には実測値を用いた. 端末シミュレータには, 表 3 に示した確率で RRC 状態を割り振った．RRC 状態の確率は，典型値として，モバイル端末を使用した際に実測した確率とした．基地局シミュレータで発生させる遅延の内，IDLE，CELL_PCH，RRC_IDLE の遅延に関しては、配信サーバから端末シミュレータに送信するトリガパケット(トリガには, TCP を利用しているため，より正確には，TCP の SYN パケット)のみに適用し，その後の通信については，遷移後の RRC 状態に基づく遅延を適用した．式(1)のデッドライン値は，表 2，表 3，に基づき導出した．

評価では，配信要求プログラムが 1500 の端末シミュレータ宛てのメッセージ配信要求を配信サーバに送信し，各端末がメッセージを受け取るまでの時間を測定した．評価は，処理スケジューリングを実施しなかった場合（以下、FiFo 方式と表記），トリガ先行方式，RRC 状態スケジューリング方式の 3 種類について行った．なお，トリガ先行方式では，メッセージ登録処理がメッセージ取得要求の受信前になるように，デッドラインを，

$$RRC_T = RRC_R = \text{RRC\_CONNECTED} \quad (2)$$

として導出した．

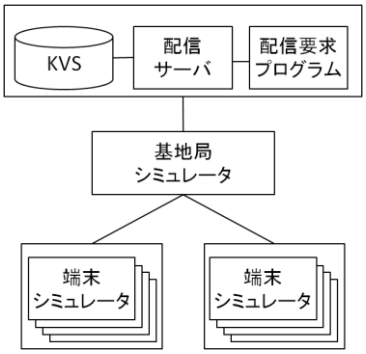


図 6　評価システムの構成

表 3　RRC 状態毎の遅延と確率分布

| RRC 状態 | 遅延(ms) | 確率(%) |
|---|---|---|
| CELL_DCH | 50 | 3 |
| CELL_FACH | 150 | 1 |
| CELL_PCH | 2000 | 9 |
| IDLE | 3000 | 56 |
| RRC_CONNECTED | 25 | 29 |
| RRC_IDLE | 500 | 2 |

### 5.2 評価結果

図 7 は，FiFo 方式，トリガ先行方式，および，RRC 状態スケジューリング方式において，メッセージの受信が完了した端末の累積数と時間の関係を表したグラフである．なお，時間原点は，配信要求プログラムが配信要求を送信した時間となっている．グラフを見ると，FiFo 方式，トリガ先行方式，RRC 状態スケジューリング方式の順にメッセージ配信遅延が小さくなっていることがわかる．

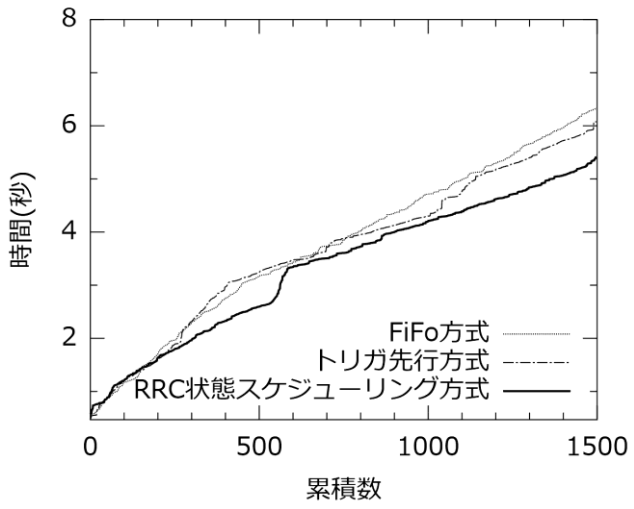


図 7　メッセージ受信時間

RRC 状態スケジューリング方式の結果を見ると，約 2.6 秒を境に傾きが小さくなっていることがわかる．これは，約 2.6 秒の時点で，RRC 状態が RRC_CONNECTED，CELL_DCH，CELL_FACH の端末への配信が完了するためである．約 2.6 秒から約 3.2 秒の間にメッセージを受信している端末は CELL_PCH の端末で，端末数が少ないため，グラフの傾きが小さくなる．その後、端末数の多い IDLE の端末がメッセージを受信するため，3.2 秒以降で再びグラフの傾きが大きくなっている．一方で，トリガ先行方式，FiFo 方式の結果を見てみると，RRC 状態スケジューリング方式のように RRC 状態毎の構造が明確には現れてはいない．

RRC 状態スケジューリング方式においてのみ RRC 状態毎の構造が現れるのは，RRC 状態を考慮したデッドラインスケジューリングにより，メッセージを早期に受信可能な端末に対する処理は先に，RRC 状態が IDLE になっており，すぐにはメッセージを受信できない端末の処理は後に実施され，RRC 状態毎にメッセージ受信時間が分離するためである．FiFo 方式やトリガ先行方式では，メッセージを早期に受信できる端末の処理が，すぐにはメッセージを受信できない端末の処理のために待たされることとなる．その結果，0 から 3 秒の範囲において，他の方式に比べ RRC 状態スケジューリング方式の方が配信遅延を小さくすることができている．

3 秒以後の範囲においても，RRC 状態スケジューリング方式が配信遅延の点で最も優れている．3 秒以後にメッセージを受信している端末は，主に IDLE 状態の端末で構成される．RRC 状態スケジューリング方式では，トリガ送信を先に実施し，IDLE 状態の端末の RRC 状態遷移開始時間が早まるため，メッセージ配信の最大遅延が小さくなる．トリガ先行方式においても，トリガ送信処理を先に行う．しかし，本来優先的に行う必要のない IDLE 状態への端末のメッセージ登録処理の優先度を下げられないため，トリガ送信処理が待たされてしまい，結果，メッセージ配信の最大遅延が大きくなっている．

表 4 に FiFo 方式と提案方式におけるメッセージ配信遅延と遅延の最大値を示す．RRC 状態スケジューリング方式は，平均値で 12%，最大遅延で 15%改善している．

表 4　メッセージ配信遅延

| | 平均 | 最大遅延 |
|---|---|---|
| FiFo 方式 | 3.8 秒 | 6.4 秒 |
| トリガ先行方式 | 3.7 秒 | 6.1 秒 |
| RRC 状態スケジューリング方式 | 3.4 秒 | 5.4 秒 |

## 6. おわりに

本稿では，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た．モバイル端末では，RRC 状態遷移による遅延が大きいため，メッセージ配信時には早期に RRC 状態の遷移を発生させる必要がある．提案方式では，モバイル端末の RRC 状態を考慮し，配信処理スケジューリングを行うことで，RRC 状態の遷移を発生させるトリガパケットを早期に送信することで，メッセージの配信遅延を抑制する．シミュレーション評価を行った結果，メッセージ配信遅延を 15%抑制することができることが示された．今後の課題としては，メッセージ配信要求が時間的に分散して到着する場合などのより複雑な条件下での評価があげられる．

## 参考文献

1) Google Cloud Messaging for Android
http://developer.android.com/google/gcm/index.html
2) Apple Push Notification Service
http://developer.apple.com/library/mac/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/ApplePushService/ApplePush
3) 3rd Generation Partnersihp Project (3GPP), Tech. Spec 36.331 V8.17.0(2012).
4) SMARTPHONES AND A 3G NETWORK, SiGNALS Research Group(2010)
5) Fast Dormancy Best Practises Version 1.0, GSM Association Official Document TS.18(2011)
6) Haverine, H. et al.: Energy Consumption of Always-On Applications in WCDMA Networks, IEEE 65th Vehicular Technology Conference, pp 964-968(2007)
7) 大久保尚人，ウメシュアニール，岩村幹夫，新博行: 高速・大容量・低遅延を実現する LTE の無線方式概要, NTT DOCOMO テクニカルジャーナル, Vol.19, No.1, pp. 11-19 (2011).
8) Pimentel, V. and Nickerson, B.G.: Communicating and Displaying Real-Time Data with WebSocket, Internet Computing, IEEE, Vol.16, Issue.4, pp45-53(2012)
9) Fette, I. and Melnikov, A.: The WebSocket Protocol, IETF, RFC 6455(2011).
10) Couchbase Server
http://www.couchbase.com/
11) netem
http://www.linuxfoundation.org/collaborate/workgroups/networking/netem