

携帯電話向け汎用アプリケーションゲートウェイフレームワークの開発と評価

窪田 聡史 安井 浩之 松山 実

武蔵工業大学

1 まえがき

携帯電話には通話機能に加え、メール、カメラ、WWW への接続、アプリケーションの実行などの機能があり、国内で使用されている多くの携帯電話では、Java で開発されたアプリケーションが実行可能である。

一般的な携帯電話用 Java アプリケーションは、以下の Fig.1 に示すように、ダウンロード元であるアプリケーション配布サーバとしか通信できず、通信方法も HTTP(S)に限られる^[1]。そのため、一般的なクライアント/サーバ型アプリケーションの実装には向いていない。そこで、この制約の下で外部サーバとさまざまなプロトコルで通信を行うためのフレームワークを提案してきた^[2]。本報告ではそのフレームワークの開発及び評価について述べる。

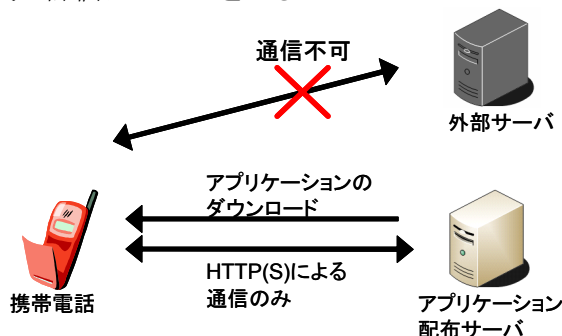


Fig.1 携帯電話用アプリケーションの通信

2 フレームワークの概要

本フレームワークの対象は現在契約数のシェアが最も高く、他の通信キャリアよりも比較的制約が厳しいNTT DoCoMoのiアプリのうち、NTT DoCoMoに認定され、制約が緩和されたトラステッドiアプリを除いた、通常のiアプリを対象とする。

このフレームワークはFig.2に示すように携帯電話用Javaアプリケーション開発のためのライブラリ、アプリケーション配布サーバ上で各種プロトコルに通信を変換するゲートウェイ及び中継プロセスで構成される。

アプリケーションの開発者は、このフレームワークのライブラリを用いて携帯電話用アプリケーションを開発することで、外部サーバへのアクセスを行うことが可能なクライアント/サーバ型アプリケーションを作成することができる。単純な携帯電話用クライアント/サーバ型アプリケーションであれば、アプリケーション配布サーバを中継して外部サーバと間接的に通信するという意識せずに開発でき、複雑なものもアプリケーション配布サーバ上の中継プロセスのプログラムを変更することで実現できる。

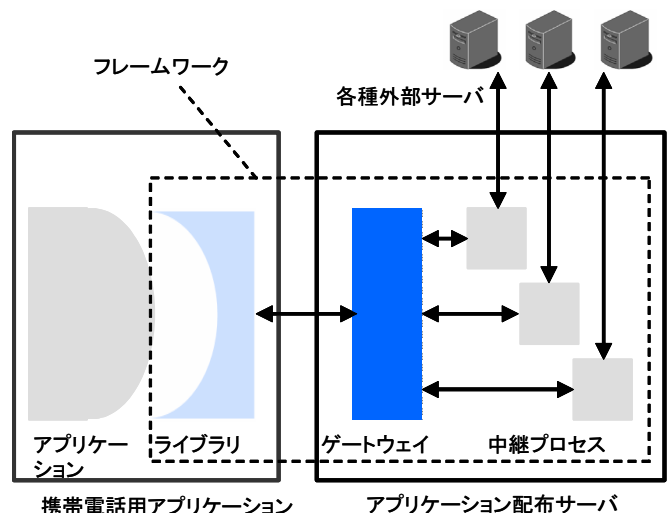


Fig.2 フレームワークのイメージ

3 フレームワークの開発

このフレームワークをFig.3に示すような流れで実装した。

①携帯電話アプリケーションがゲートウェイと通信し認証に成功するとセッションIDが発行され、外部サーバのアドレスとデータをゲートウェイに送信する。②初回の通信の場合、ゲートウェイはセッションIDと外部サーバのアドレスを基に、中継プロセスを生成する。③中継プロセスはゲートウェイからデータを受け取り、外部サーバにデータを送信し、外部サーバから返ってきた応答を受信してゲートウェイに返す。④ゲートウェイは携帯電話アプリケーションに

データを返すことで一旦終了する。⑤この時点でゲートウェイのプロセスは一度消滅するが、中継プロセスは外部サーバとの通信を維持する。⑥二度目以降の通信は、セッション ID に対応する中継プロセスに対してデータを送信することで行われる。セッション ID の有効期間が過ぎると、中継プロセスも破棄される。

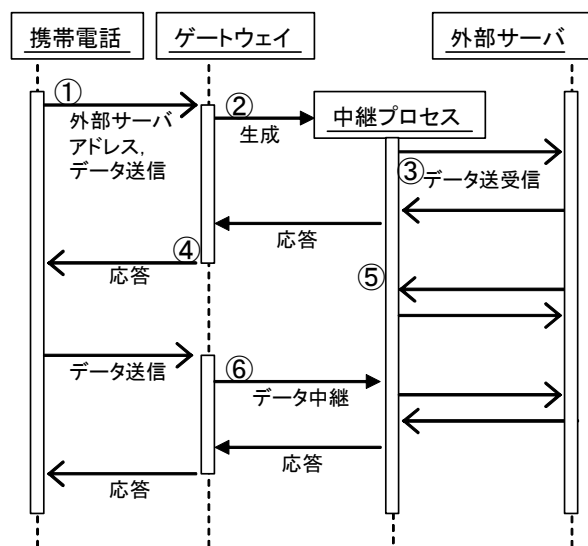


Fig.3 フレームワークのシーケンス図

4 評価と考察

フレームワークの有用性を検証するため、アプリケーション配布サーバを中継する際のクライアント／サーバ間の通信速度の低下を測定する実験を行った。実際にフレームワークを利用して作成した携帯電話用 IRC クライアントを PC クライアント上でエミュレータを用いて動作させ、アプリケーション配布サーバを中継して IRC サーバに接続した場合と、PC クライアントから IRC クライアントソフト Cotton を用いて直接 IRC サーバに接続する場合の 2 通りについて、その接続に要する時間をそれぞれ 10 回ずつ測定した。実験環境を Table1, Fig.4 に示す。

Table1 実験環境

マシン	CPU	メモリ	OS
PC クライアント	2.6 GHz	736 MB	Windows XP
アプリケーション配布サーバ	600 MHz	64 MB	Debian 3.1

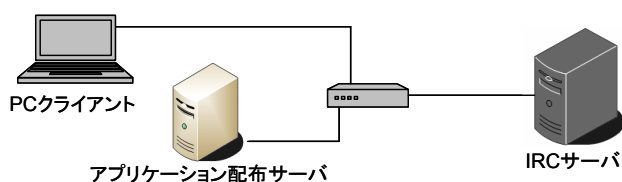


Fig.4 実験環境

実験の結果、フレームワーク不使用時は平均 11.78 秒であったのに対し、使用時は 12.81 秒となった。フレームワークを中継することによるロス約 1 秒であり、接続速度の低下率は約 9% となった。が、極端な速度低下とはならなかった。

次に、携帯電話用に POP サーバからメールを受信する POP クライアントを開発することを想定し、本フレームワークの利便性を定性的に検証する。

携帯電話用 POP クライアントは直接 POP サーバにアクセスできないため、フレームワークを使用しない場合、携帯電話用プログラムの他に、アプリケーション配布サーバ上にも POP に対応するプログラムを開発する必要がある。それに対し、フレームワークを使用して開発する場合、ライブラリを用いて携帯電話用プログラムを開発するだけで、アプリケーション配布サーバを経由して POP サーバからメールを取得することができる。

同様に、POP 以外のクライアント／サーバ型アプリケーションを開発する場合においても、フレームワークを使用しない場合は最初からアプリケーション配布サーバ上のプログラムを開発しなければならない。

以上のことから、本フレームワークを使用することにより開発の工程が短縮できるという利点があるといえる。

5 まとめ

本報告では、携帯電話アプリケーションの制限を回避するためのフレームワークの構成、実装方法を述べ、フレームワークを中継する際のパフォーマンスの測定を通してフレームワークの有用性、利便性について述べた。

今後の課題として、携帯電話アプリケーション開発用のライブラリ部分の使い易さなどの評価が必要である。

参考文献

[1] 作ろう i モードコンテンツ：i アプリコンテンツの概要 | サービス・機能 | NTT ドコモ
<http://www.nttdocomo.co.jp/service/imode/make/content/iappli/caution/index.html>

[2] 窪田, 安井, 松山: “携帯電話向け汎用アプリケーションゲートウェイフレームワークの提案と開発”, FIT2006 講演論文集, pp.25-26