

## 推薦論文

## 移動機向けプッシュプロトコルの提案と評価

高橋 修<sup>†</sup> 上野 英俊<sup>†</sup> 石川 憲 洋<sup>†</sup>  
 角野 宏 光<sup>†</sup> 鈴木 偉 元<sup>†</sup> 水野 忠 則<sup>††</sup>

本論文では、IMT-2000 等の高速無線ネットワーク向けプッシュプロトコルの提案を行う。本提案は、WAP (Wireless Application Protocol) 1.x のプッシュアーキテクチャを基に、無線ネットワークにインターネットプロトコル (HTTP/TCP) を用いてプッシュを実現する。また、IMT-2000 ネットワークを想定したシミュレーション環境において、WAP1.1 のプッシュプロトコルと提案方式の比較を行う。性能評価より提案方式は、プッシュ完了までの通信時間が短くなり、特にデータサイズが大きい場合には提案方式が有効であることを示す。さらに本論文では、次世代の記述言語として有力な XHTML (Extensible HTML) に着目し、それとプッシュ機能を組み合わせたプッシュ方式を提案する。その際、XSLT (Extensible Stylesheet Language Transformations) を用いたデータの自動変換を実現する機能に着目し、古いデータと更新によって生じた新しいデータへの差分を自動に生成し、移動機にプッシュする方式を提案する。さらに提案方式に基づいたテストベッドシステムの試作を行い、その有効性についても検証を行う。

## Proposal and Evaluation on Push Protocol for Mobile Devices

OSAMU TAKAHASHI,<sup>†</sup> HIDETOSHI UENO,<sup>†</sup> NORIHIRO ISHIKAWA,<sup>†</sup>  
 HIROMITSU SUMINO,<sup>††</sup> HIDEHARU SUZUKI<sup>††</sup> and TADANORI MIZUNO<sup>††</sup>

This paper proposes and evaluates a push protocol for high-speed wireless networks such as IMT-2000. The push protocol is based on the WAP (Wireless Application Protocol) 1.x architecture with modifications to realize push functionality by using Internet protocols such as HTTP and TCP. Simulation result has shown that the communication time of the proposed push protocol is nearly equal to that of the WAP 1.1 push protocol except when pushing large contents, in which case the proposed protocol offers better performance than WAP 1.1. We also propose a new push method by combining the above push functionality with XHTML, which is expected to be the next generation standard markup language. The proposed method is that the differential file between an old XHTML content and a new XHTML content is pushed to a mobile device and then it creates a new content automatically from the old content by using XSLT (Extensible Stylesheet Language Transformations). We also have evaluated the availability of our proposal by using an experimental push system.

## 1. はじめに

WAP (Wireless Application Protocol)<sup>1)</sup> では、WAP version 1.1 (WAP1.1) からプッシュ機能が提供され、このプッシュ機能を用いた製品やサービスが市場に提供され始めている。しかし WAP version 1.x (WAP1.x) は、低速、高遅延といった特徴を持つ第 1 世代 (1G)、第 2 世代 (2G) の無線ネットワークを対象に検討されたものであり、IMT-2000 等の第 3 世

代 (3G) の高速な無線ネットワークには必ずしも適しているとはいえない。現在 3G の無線ネットワークに適したプロトコルの研究や開発が行われており、インターネットで用いられているプロトコル (HTTP, TCP 等) をそのまま用い、そのサブセットの定義や必要な機能の拡張を行うことにより無線への最適化が図られている<sup>2)~4)</sup>。これによって、インターネット上のアプリケーションと親和性の高いシームレスなサービスを提供することが可能となる。しかし、プッシュ

<sup>†</sup> 株式会社 NTT ドコモマルチメディア研究所  
 Multimedia Laboratories, NTT DoCoMo, Inc.

<sup>††</sup> 静岡大学情報学部

Faculty of Information, Shizuoka University

本論文の内容は 2001 年 6 月のマルチメディア、分散、協調とモバイル (DICOMO2001) シンポジウムにおいて報告され、MBL 研究会運営委員により情報処理学会論文誌への掲載が推薦された論文である。

プロトコルについては、インターネットにおいて標準化されたものが存在しないため、新たに提案する必要がある。

本論文では、IMT-2000 等の 3G 高速無線ネットワーク向けプッシュ機能の実現を目的とし、インターネットで用いられているプロトコルを用いてプッシュ機能を実現する方式を提案する。提案方式では、無線向け TCP プロファイル<sup>4)</sup>と HTTP<sup>5)</sup>を用い、WAP1.x と同様なオリジンサーバ起動型のプッシュ機能を実現する。さらに、HTTP の機能の中からプッシュ機能を実現するために必要な部分をサブセット化するとともに、必要な機能を追加してプッシュ機能を実現する。また、次世代のマークアップ言語として有力な XHTML ( Extensible HTML ) 1.0<sup>6)</sup>と XSLT ( Extensible Stylesheet Language Transformations <sup>7)</sup>に着目し、それと提案したプッシュ機能を組み合わせた新しい差分プッシュ方式を提案する。本提案は、オリジンサーバ側で古いデータと更新後の新しいデータとの差分を自動的に生成し、移動機にプッシュするものである。これによって移動機は、つねにアップデートされた最新のデータを表示可能となり、ネットワーク上でのチャット画面の自動更新や、株式情報チャートの自動更新等のサービスへ応用することが可能になる。なお、XHTML1.0<sup>6)</sup>は、HTML4.01<sup>8)</sup>を XML ( Extensible Markup Language <sup>9)</sup>に基づいて再定義したものであり、HTML4.01 の後継となるマークアップ言語として W3C ( World Wide Web Consortium ) で標準化されたものである。XSLT は、XML データの自動変換機能を提供するものであり、同じく W3C で標準化されたものである。

最後に、提案方式に基づいて実装したテストベッドシステムを紹介し、提案方式の有効性について検証した結果について述べる。

## 2. プッシュプロトコルの提案

### 2.1 WAP1.x プッシュ

WAP1.x では、インターネット上のホストから移動機へのプッシュ機能を提供するために、移動通信向けのプッシュアーキテクチャを規定している。

図 1 に WAP1.x のプッシュアーキテクチャを示す<sup>10)</sup>。WAP1.x のプッシュアーキテクチャでは、プッシュメッセージの送信ホストである PI ( Push Initiator )、メッセージプッシュのプロキシ機能および移動機のアドレス解決等を行う PPG ( Push Proxy Gateway <sup>11)</sup>、および移動機から構成される。

図 2 に WAP1.x で用いられるプッシュプロトコルの構成を示す。PI と PPG 間のインターネット部

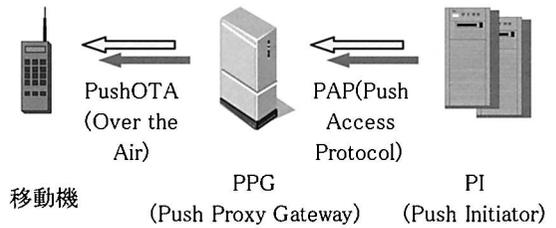


図 1 WAP1.x プッシュアーキテクチャ  
Fig. 1 WAP1.x push architecture.

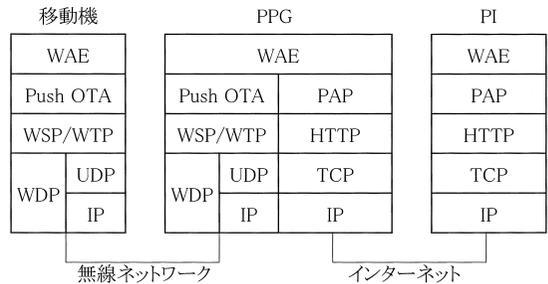


図 2 WAP1.x プッシュプロトコル  
Fig. 2 WAP1.x push protocol.

は、PAP ( Push Access Protocol <sup>12)</sup> が用いられ、その運搬用に HTTP の POST メソッドが用いられる。PAP は、XML<sup>9)</sup>により記述されるプロトコルであり、メッセージのプッシュ、結果通知、状態問合せ等の機能を持つ。PPG と移動機間の無線ネットワーク部では、トランスポートとして WDP ( Wireless Datagram Protocol <sup>3)</sup>、プッシュプロトコルとして、WSP ( Wireless Session Protocol <sup>14)</sup>を用い、その上で動作するプッシュ用のアプリケーションレベルのプロトコルを Push OTA ( Over the Air <sup>15)</sup>と呼ぶ。Push OTA は、PPG から移動機へのデータのプッシュ機能の実現や、移動機アプリケーションのアドレッシング、および各種制御情報の交換等の機能を提供する。なお、Push OTA では、コネクション型プッシュとコネクションレス型プッシュの 2 種類のプッシュ機能が提供され、プッシュするデータに求められる信頼度等の性質を基に、両者を使い分けることができる。

### 2.2 関連研究

WAP1.x プッシュは、プッシュメッセージの送信ホストである PI が任意のタイミングで移動機に対してプッシュの実行が可能であるオリジンサーバ起動型のプッシュ方式である。それに対し、移動機等のクライアントが、オリジンサーバに対してある間隔でデータの更新を問い合わせ、データが更新されている場合にのみ更新データを受信するプッシュの形態があり<sup>16)</sup>、スマートプルと呼ばれる。スマートプルは、HTTP 等

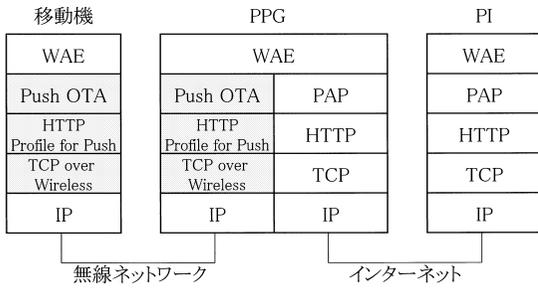


図 3 提案プッシュプロトコル  
 Fig. 3 Proposed push protocol.

のプル型のプロトコルを使用して擬似的にプッシュ機能を実現する。しかし、スマートプルの欠点として、1)データが更新されていない場合にも、不必要なデータ更新問合せが発生する、2)データが更新されたとしても、クライアントからのデータ更新問合せがあるまでは、クライアントにデータ提供が不可能であるため、更新されたデータを即時的に提供できない場合がある、等の問題点がある。携帯電話等による E-mail の例にあげられるように、移動機向けプッシュ機能では、スマートプル方式ではなく、WAP1.x プッシュのようなオリジンサーバ起動型のプッシュ方式を利用し、移動機に対して即時的なデータ提供ができるようになる必要性が高い。これまでの研究では、3G 高速無線ネットワークを対象としたサーバ起動型のプッシュプロトコルに関する研究や、3G 高速無線ネットワークにおける WAP1.x プッシュプロトコルの性能を明らかにした研究は存在しない。

プッシュ機能を提供する場合、クライアントが増えた場合のスケーラビリティが問題となる。特に大量のクライアントに対して同時にプッシュするときのプッシュセッションを同時に確立するかシーケンシャルに確立するか等の方式が考えられる。また、複数クライアントへ同時配信を実現する等の方法としてマルチキャスト<sup>17),18)</sup>を採用し、配信サーバのスケーラビリティを高める方法が存在する。

### 2.3 プッシュアーキテクチャ

提案するアーキテクチャは、WAP1.x のプッシュアーキテクチャ(図 1)に基づき、移動機、PPG、および PI からなり、以下のプロトコル構成とする(図 3)。移動機と PPG 間には、3G ネットワークの高速性の特徴を有効利用する新しいプロトコルを採用する。すなわち、無線ネットワーク部分におけるトランスポートプロトコルとして現在 IETF で議論されている無線向け TCP プロファイル<sup>4)</sup>と、その上に HTTP1.1<sup>5)</sup> の POST メソッドを用いる。プッシュ型配信では、イ

ンターネット上の PI から PPG を経由して移動機へのオリジンサーバ起動型のプッシュ機能を実現するため、PPG が HTTP のクライアントとして動作し、移動機が HTTP のサーバの機能を提供する。移動機は、CPU 能力や搭載可能なメモリの制限があるため、プッシュ機能の実現に必要な HTTP 機能のみを抽出し、サブセット化するとともに、必要な機能を拡張ヘッダとして新たに定義する。このプッシュ機能を持つ HTTP のことをプッシュ向け HTTP プロファイルと呼び、2.4 節で詳細を説明する。なお、インターネット側では、既存方式をそのまま用いる。すなわち、PI 上のアプリケーション、および PAP<sup>12)</sup> の変更を行う必要がなく、すでに WAP1.x 移動機向けにサービスを開始している PI については、そのまま提案方式を実装した移動機に対してもサービス提供を可能とする。

### 2.4 プッシュ向け HTTP プロファイル

プッシュ向け HTTP プロファイルでは、HTTP<sup>5)</sup> の必須項目のうちプッシュ機能を提供するうえで必要としないものについてはサポートをオプションとし、HTTP のオプション項目のうち必要な機能についてはサポートを必須と定義する。以下では、プッシュ向け HTTP プロファイルについて説明する。

(a) メソッド : HTTP メソッドには、HTTP POST を用い、他のメソッドのサポートはオプションとする。HTTP では、GET と HEAD メソッドが必須項目であるが、プッシュ機能の実現に必須でないためサポートをオプションとする。

(b) HTTP ヘッダ : HTTP ヘッダとして、Host、Content-Length、Content-Type に加え、Connection ヘッダのサポートを必須とする。Connection ヘッダは、TCP コネクションの持続的接続(パーシステントコネクション)を制御するために用いる。パーシステントコネクションは、連続するプッシュで同じ TCP コネクションを再利用できるようにしたものであり、TCP コネクションの開設フェーズ、切断フェーズを省略することができるため、通信コストが高い無線ネットワークにおいては有効である。

(c) 拡張ヘッダ : WAP1.x の Push OTA で提供する機能を実現するため、以下に示す新たな Push OTA プロトコル用ヘッダを定義する。ここで定義する拡張ヘッダは、HTTP<sup>5)</sup> のシンタックスに従って記述され、HTTP ヘッダ部に追加される。

- (ア) X-Wap-Application-Id : 移動機に複数実装されたアプリケーションを識別し、プッシュプロトコルがデータを渡すアプリケーションを選択する。
- (イ) X-Wap-Client-Id : 移動機を識別する ID を

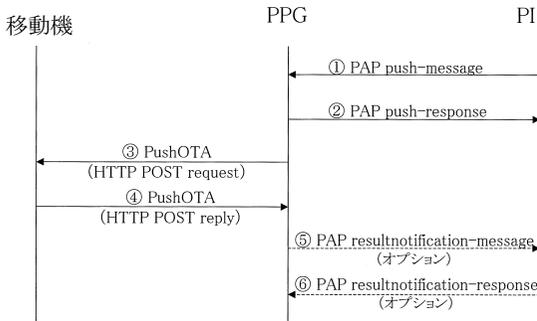


図 4 提案方式の通信例

Fig. 4 An example of the proposed method.

PPG と移動機間で共有することにより、セッション管理等を実現する。

(ウ) X-Wap-Push-Flag : WAP1.x Push OTA 仕様<sup>15)</sup>と同様に、データや PI の各種情報を定義する。

(エ) X-Wap-Push-Id : 複数のプッシュを識別するための ID を設定する。

2.5 提案方式のプッシュ通信例

提案プッシュ方式用いたプッシュ通信例を図 4 に示す。まず PI は、PPG に対して PAP の push-message メッセージを用いてデータをプッシュし(図 4 の①), それを受け取った PPG は、PI に対して PAP の push-response メッセージを返す(図 4 の②)。PPG は、push-message メッセージを受け取ると、移動機に対して無線向け TCP のコネクションを開設し、プッシュ向け HTTP プロファイルによって、移動機へとプッシュを実行する(図 4 の③)。その後、データを受け取った移動機は、PPG に対してレスポンスを返す(図 4 の④)。なお PI は、プッシュが正しく実行されたか確認することができ、もし PI が、図 4 の①で確認応答を要求した場合には、PPG は、resultnotifitation-message を PI に返す(図 4 の⑤)。

3. プッシュプロトコルの性能評価

3.1 性能評価

IMT-2000 ネットワークを想定したシミュレーション環境において、提案方式と WAP1.x との比較によって、提案方式の特性把握を行う。本研究では、WAP1.x のうち最初にプッシュ機能が提供された WAP1.1 を選択し、WAP1.1 と提案方式のテストベッドを試作し、PPG から移動機へのプッシュに関して、WAP1.1 プッシュプロトコルと提案方式のリクエスト、レスポンスに要する時間(以下、応答時間とする)の測定を行った。なお、ペアラに W-CDMA( Wideband CDMA )

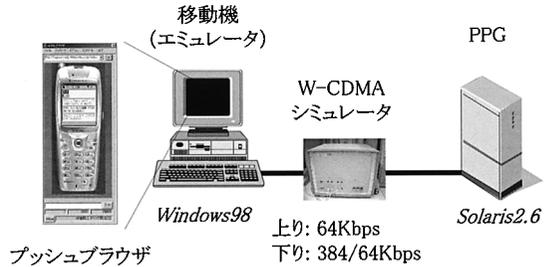


図 5 性能測定環境

Fig. 5 Performance measurement environment.

パラメタ	値
ペアラ速度(上り)	64Kbps
ペアラ速度(下り)	384Kbps, 64Kbps
エラー率	5% (FECフレーム毎)
レイヤ2	RLCプロトコル [19]
ペアラMTU	1500バイト

表 1 W-CDMA シミュレータパラメータ

Table 1 W-CDMA simulator parameters.

を用いることを想定するため、無線環境を擬似する W-CDMA シミュレータを移動機と PPG の間に介在させた。図 5, および表 1 に性能測定環境を示す。表 1 に示すように、本測定では、レイヤ 2 のプロトコルとして RLC( Radio Link Control )<sup>9)</sup> を使用し、ペアラ MTU( Maximum Transfer Unit ) サイズを 1,500 バイトとした。これらの各パラメータの値は、FOMA サービスで使用されているパラメータ値に基づいて設定してある。表 1 のエラー率は、W-CDMA の 1 FEC ( Forward Error Correction ) フレームごとのエラー率を示しており、無線通信で発生したビットエラーを、FEC フレームごとにエラー訂正を行い、その結果エラー訂正が行えなかった FEC フレーム数の平均比率を示している。なお、FEC により誤り訂正が行えない場合には、フレームのエラーとして上位層に通知され RLC プロトコルで再送制御が行われる。本論文では、このエラー率を、FOMA サービスで用いたネットワークの測定を基に導き出した標準的な値を採用している。上記の環境で、移動機( エミュレータ ) から、PPG までのラウンドトリップ時間( RTT: Round Trip Time ) は 350 ~ 400 ミリ秒程度であった。

本測定では、3 種類( 500, 10 K, 100 K バイト ) のサイズのデータを用意した。500 バイトのデータは、XHTML<sup>6)</sup> 等のテキストデータのプッシュを想定しており、TCP における最大セグメント長( MSS: Maximum Segment Size ) や WAP1.1 の 1 パケットに収まる最大のデータサイズに収まる。また、100 K バイトのデータは、i-mode サービス等における、MPEG4 デー

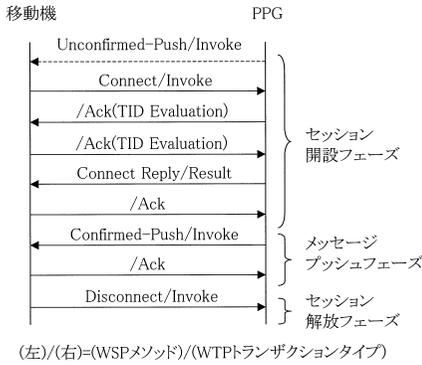


図 6 WAP1.1 の通信シーケンス  
Fig. 6 Communication sequence of the WAP1.1.

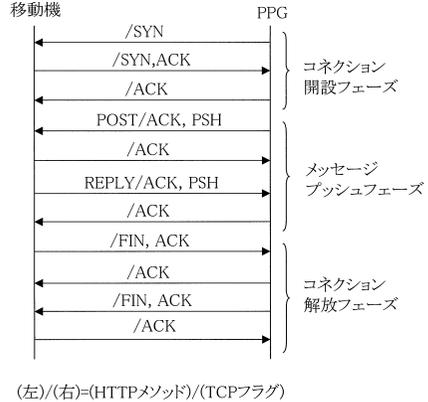


図 7 提案方式の通信シーケンス  
Fig. 7 Communication sequence of the proposal.

たや Java データのプッシュを想定しており、WAP1.1 においては WTP (Wireless Transaction Protocol) の分割再組み立て (以下 SAR: Segmentation and Reassembly とする) 機能<sup>20)</sup> を用いることにより、WAP1.1 の 1 パケットに収まる最大のデータサイズを越えるデータの転送が可能となる。この SAR 機能は、WAP1.1 において本来オプションであるが本テストベッドシステムでは実装を行っている。

なお、今回行った評価では、図 5 に示す OS (PPG=Solaris2.6, 移動機=Windows98SE) に実装されている TCP とその初期設定値をそのまま用いており、無線向け TCP プロファイル<sup>4)</sup> に準拠していない。

3.2 通信シーケンスの比較

PPG と移動機間における接続型プッシュの WAP1.1 の通信手順を図 6 に、提案方式の通信手順を図 7 に示す。いずれの方式も、1) WSP<sup>14)</sup> セッション/TCP 接続開設フェーズ (以下、開設フェーズとする)、2) プッシュ実行フェーズ、3) WSP セッション/TCP 接続切断フェーズ、から構成される。なおプッシュを連続して行う場合には、WSP セッションの共用、または HTTP1.1 のパーシステント接続を用いることにより、新たな開設フェーズは必ずしも必要ない。

3.3 開設フェーズ

図 8 に開設フェーズに要する時間の測定結果を示す。提案方式は、WAP1.1 よりも開設フェーズに要する時間が短い。この理由は WSP セッション開設に要するシーケンス数 (= 本測定結果では 5 シーケンス) が TCP 接続開設に要するシーケンス数 (= 本測定結果では 3 シーケンス) よりも多いことが考えられる。また、ベアラの下り速度が 64 kbps から 384 kbps と高速になっても、それぞれに要する時間を比べると

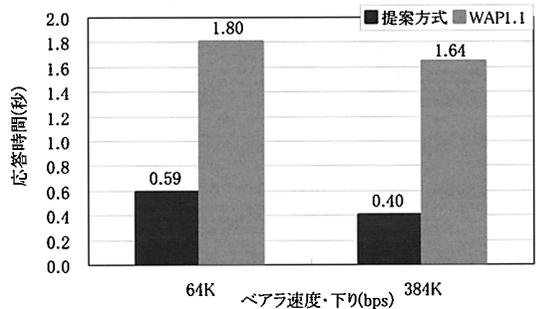


図 8 開設フェーズの比較  
Fig. 8 Comparison of the establishment phase.

それほど向上しないことが分かる。これは、開設フェーズで用いられるパケットはヘッダのみで構成されており、そのサイズは大きくないことが原因と考えられる。すなわち、ベアラが高速になってもパケットサイズが大きいため、それが開設フェーズの時間短縮にはつながらず、むしろネットワーク遅延等が開設フェーズに要する時間に影響していることを示している。なお、WSP セッションは移動機のみ開設することが可能であるため、図 6 の破線で示す unconfirmed-Push (非確認型プッシュ) を行うことにより、移動機のセッション起動を要求することが可能である。図 5 に示す本測定環境では、移動機が受信可能状態に入る際 (移動機の電源を ON にする等) に、WSP セッションを開設し、PPG と移動機間でそれを保持するという状況を想定しているため、図 8 に示す開設フェーズの通信時間の測定では、この unconfirmed-push のシーケンスが測定時間には含まれていない。

3.4 プッシュ実行フェーズ

図 9 は、プッシュ実行フェーズの応答時間を示す。データサイズが小さい場合 (= 500 バイト) には、

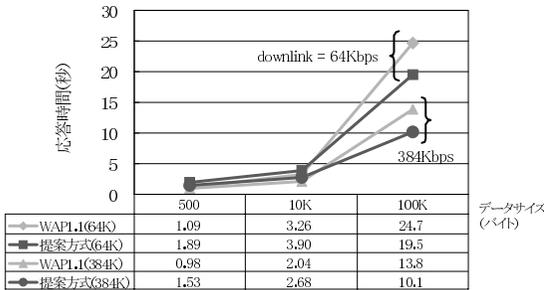


図9 プッシュ実行フェーズの比較

Fig. 9 Comparison of the push submission phase.

WAP1.1の応答時間が優れるという結果が得られた。また、データサイズが大きくなるに従いその応答時間の差は小さくなり、データサイズが大きな場合(=100Kバイト)には、逆に提案方式の応答時間が優れるという結果が得られた。なお、応答時間は、ベアラの下り速度が高速になればWAP1.1および提案方式ともに短くなるものの、ベアラ速度が変化することによって各プロトコルで得られる応答時間に逆転が生じる等の違いは得られなかった。すなわち、これらのプロトコルは、両方ともベアラ速度にほぼ比例して応答時間が変化することが分かる。

プッシュ前に開設フェーズを必要とする単独プッシュでは、プッシュ実行フェーズ(図8の結果)に開設フェーズ(図9の結果)が加わるため、提案方式がWAP1.1よりもプッシュ完了までの時間が短い。以上から、サイズが大きいデータや単独のプッシュ時には、提案方式はWAP1.1よりも有利であることが明確となった。一般的な利用方法では、多くのプッシュは単独で行われるため、提案方式の応答時間はWAP1.1の応答時間よりも短いことを示した。

### 3.5 性能測定の考察

本節では、WAP1.1とHTTP/TCPのデータ転送の仕組みを比較し、図8, 9の結果について考察する。

小さいデータでWAP1.1が優れた理由としてTCPの初期輻射ウィンドウサイズの決定方法がその理由としてあげられる。TCPでは、通信初期の輻射ウィンドウサイズを最大セグメントサイズ(MSS)に設定し、その後、スロースタートアルゴリズムに従ってウィンドウサイズを徐々に伸張する。したがって、小さいデータのプッシュの場合には、ネットワークの帯域幅を十分に使いきるためのウィンドウサイズの伸張がないままデータ転送が終了する。反面WAP1.1のフロー制御を提供するWTP<sup>20)</sup>では、固定のウィンドウサイズを用いたウィンドウ制御方式を採用している。したがって、小さいサイズのデータの場合には、通信

初期の段階からウィンドウサイズの範囲内でデータ転送が行えるため、WAP1.1が有利であるという結果が得られた。

小さいデータでWAP1.1が優れたもう1つの理由として、それぞれのプロトコルに必要なパケット数の違いがあげられる。WSPのconfirmed-pushメソッドの確認応答は、WTPのAckの2パケットを要している(図6)。それに対しHTTP/TCPでは、HTTP POSTメソッドとそれに対するTCPのACKと、HTTP POSTリプライとそれに対するACKということで4パケットを要している(図7)。つまり、HTTPでは、“200 OK”といったHTTPレイヤでの確認応答に対するTCP ACKが必要となり、HTTP/TCPでは、必要なパケット数が多くなっている。以上のことから、小さいデータ送信時には、必要とするパケット数の差についても応答時間に影響したものと考えられる。

無線パケットネットワークの多くは、固有のネットワーク規制メカニズムを備えるため、ネットワークの輻射対策は、ベアラネットワークで対処を行うことが可能である。したがって、どこで輻射が発生するか推定しにくいインターネットの場合には、スロースタートを用いた輻射回避対策が有効であるが、提案方式のように無線ネットワークに閉じた環境でTCPを適用する場合には、トランスポート層でスロースタートアルゴリズムを提供する必要があるかどうか疑問である。むしろ小さいデータ向けに初期ウィンドウサイズ伸張等の対処を行うことで、サイズが小さいデータに対して有効なデータ転送が行えることが期待できる。

大きいデータでTCPが優れる理由については、TCPとWTPのウィンドウ制御方式の違いが考えられる。TCPでは、ピギーバック方式を採用し、データパケットの送受信ともにACKパケットが送受信される。対してWTPでは、ACK用の独立したパケットが存在し、さらに、ウィンドウサイズ内のデータパケットをすべて受信した後にACKパケットを送出する。すなわち、TCPでは、データの送受信中にあわせてACKの送受信を行っているのに対し、WTPでは、データ送受信と、ACKの送受信が明確に区別されている。したがってWAPでは、データサイズが大きくなるに従い、このACK応答のための待ち時間(図10)が全体的な応答時間に影響したものと考えられる。なお、図10は、WTPとTCPの通信シーケンスを模式的に示したものである。WTPでは、ACK応答のための待ち時間が存在し、TCPでは、通信初期のスロースタートフェーズにおいて、帯域幅を使いきっていない様子が確認できる。

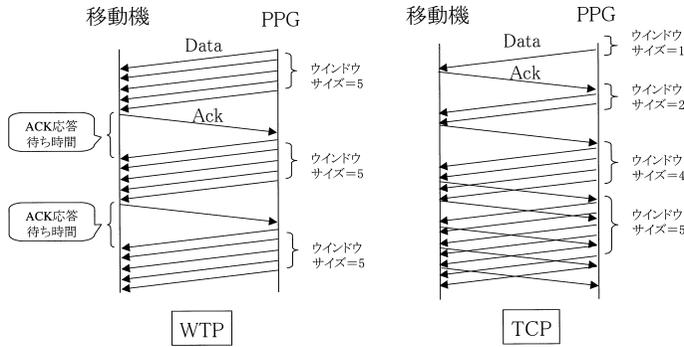


図 10 WTP と TCP のウィンドウ制御方式の比較  
 Fig. 10 Window control comparison between WTP and TCP.

TCP では、一定時間パケットの到着がない場合には、その監視タイマの満了により ACK を返す制御を行うため、移動機から PPG 方向(上りリンク)に送信される TCP ACK の数は、WAP1.1 の WTP の Ack 数と比較して多い。また先に述べた HTTP とそれに対する TCP ACK の存在のために、必要なパケット数が WTP よりも多くなる傾向にある。しかし、W-CDMA 等の無線ネットワークでは、下りと上りリンクで別々の帯域を確保しており、さらに、TCP ACK サイズとその発生頻度に対して上りリンクの帯域は十分に余裕がある容量であるため、大きいサイズのデータのプッシュ時においては、上りの TCP ACK パケット数が全体的な性能には影響しないことが分かった。

トランスポートの性能のほかに WSP のヘッダコンパクトエンコーディング<sup>14)</sup> も小さいデータで WAP が有利であった原因の 1 つに考えられる。WSP においては、ヘッダのバイナリ化を行うことによって約 35%のデータ量を減少させるため、小さいデータにおいて WAP1.x の応答時間が短縮された。しかし、WSP ヘッダのコンパクトエンコーディングは、データサイズが小さい場合には有効であるが、データサイズが大きい場合には、データパケット全体から見て、ヘッダが占めるデータ量が相対的に小さいため、コンパクトエンコーディングの効果が出にくい。

以上の結果から、サイズの大きいデータが提供されることが予想される、W-CDMA 等の高速ベアラにおいては、提案方式が採用する HTTP/TCP により実現したプッシュプロトコルの有効性が示された。なお、今回の評価では、現在インターネットで用いられている TCP をそのまま用いており、無線向け TCP プロファイル<sup>4)</sup> に準拠していない。本研究では、W-CDMA のような高速な無線ネットワークを想定しているため、W-CDMA の回線性能をさらに引き出すことが可能な

無線向け TCP プロファイルを用いることにより、提案方式の応答時間がさらに短くなることが予想できる。無線向け TCP プロファイル<sup>4)</sup> の概要については、付録を参照されたい。

我々は、本研究結果に基づいた提案プッシュプロトコルを WAP フォーラムに提案し、提案方式は、WAP2.0 プッシュプロトコル仕様のベースとして採用された。WAP2.0 プッシュプロトコルは、提案方式に加えて、移動機の機種等の能力情報を PPG に伝達する手段や、移動機と PPG の相互認証機能等が追加されている<sup>21)</sup>。したがって本研究結果は、IMT-2000 等の高速ネットワークにおける WAP1.x と WAP2.0 プッシュプロトコルの性能比較の上で参考になるものと考えられる。ただし、WAP2.0 と提案方式の詳細な比較については、今後の研究対象としたい。

#### 4. 差分プッシュ方式

##### 4.1 提案の背景

WWW( World Wide Web )のプルモデルやスマートプルでは、データが更新された後すぐにクライアント側に反映できないといった即時性の問題がある。また、一般にプッシュ型サービスでは、プッシュしたデータが移動機にとって必要な情報かどうか送信側が判断できないという問題がある。ネットワークリソースの浪費を防ぐためには、プッシュデータのサイズをできる限り小さくすることが課題となる。本論文では、データの更新を契機として、オリジンサーバから移動機に対してデータをプッシュすることによって即時性を確保し、さらに更新されたデータの差分のみをプッシュすることによってネットワーク利用効率にも優れた新しいプッシュ方式を提案する。

関連する研究として、スマートプル方式を応用したデータの差分情報転送方式がある。この方式では、

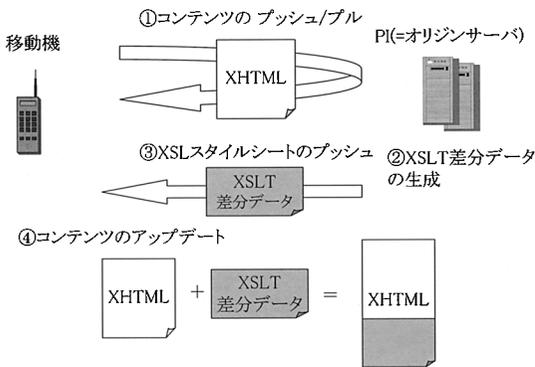


図 11 差分プッシュ

Fig. 11 Differential file push.

HTTP 等のプル型プロトコルを用いて、クライアントがオリジンサーバに対して定期的にデータの更新問合せを行い、オリジンサーバ側でデータが更新されている場合には、以前にクライアントに提供した更新前データと、更新後データの差分データを提供する方法である。HTTP において、この機能を実現するための方法として、HTTP のデルタエンコーディング<sup>22)</sup>が仕様化されている。HTTP デルタエンコーディングでは、UNIX の diff コマンド等により任意のテキストデータの差分を抽出することができるとしているが、XHTML<sup>6)</sup>等の XML のデータ構造に着目した差分データの検出法については述べられていない。また、オリジンサーバ起動型のプッシュ方式における差分プッシュ方式に着目した研究は行われていない。そこで本論文では、2.2 節で説明したスマートプル方式の問題点を解決するオリジンサーバ起動型プッシュを応用した差分プッシュ方式を提案する。

#### 4.2 差分プッシュ方式の提案

移動通信網は有線ネットワークと比較して、ネットワーク利用コストが高いためできるだけネットワーク利用効率を高める必要がある。このため、差分情報のみを転送することによって、ネットワークを流れるデータ量の削減を図ることが重要となる。本論文では、XHTML 等の XML の文書構造に着目した差分プッシュ方式を提案する。

提案プッシュ方式の処理内容を以下に示す(図 11)。

①初めに移動機は、プル型もしくはプッシュ型にて差分データの元となるデータを取得する。②PI (=オリジンサーバ)でそのデータが更新された場合、PI は XSLT<sup>7)</sup>で記述されたスタイルシートで表現した差分データを生成する。③PI は更新前のデータを以前に提供した移動機のリスト(ユーザリスト)を保持しており、ユーザリストに登録されている移動機に対して

差分データをプッシュする。④差分データを受け取った移動機は、XSLT を用いて更新前の旧データと差分データから新データを自動生成する。

XSLT は、XML 文書の構造変換機能を提供するための W3C 仕様であり、ある XML データに対して特定範囲だけを指定した体裁で表示・出力することや、構成要素の並べ替えを行うことが可能である。この技術を応用することにより、更新前データから更新後データへの差分情報を、XSLT を用いた変換ルールとして記述することが可能である。更新前データに XSLT 差分データを適用するソフトウェアを用いて、自動的に更新後データを生成できる。任意に与えられた 2 つの XML 文書の差分データの生成方法として、更新前後の XML 文書の DOM (Document Object Model)<sup>23)</sup> ツリーを生成し、その 2 つの DOM ツリーを比較することによって、XML の差分箇所を検出し、その XSLT 差分データを生成する方法等がある<sup>24)</sup>。掲示板の新たな発言内容の追加等の例で考えられるように、決まった更新箇所にデータが追加されるような単純な XML データの更新の場合には、更新がある箇所の追加情報を XML テンプレートで記述するといった単純な方法で実現することもできる。本提案方式では、この方法を用いて XSLT 差分データを生成することによって、差分プッシュを実現した。

新データの画面への表示は、XHTML<sup>6)</sup>に新たな onpush イベントタグを定義し、差分データ受信時の画面更新のための動作を規定する。該当する旧データが画面上に表示されておらず、移動機のローカルキャッシュ中にそのデータが存在するような場合においても、移動機はそのローカルキャッシュを書き換えてもよい。これによってユーザは、ローカルキャッシュ中に存在するものについては再読み込み(リロード)をしなくても最新情報を取得することが可能になる。

以下では、最初のデータをプル型で取得し、差分データをプッシュ型で移動機に提供する方式について述べる。WWW のブラウジングでは、リンクをたどりながら目的のデータを探す、もしくはたどり着くのが一般的であることから、オリジンサーバが提供するデータのうち、あるユーザにとって最も必要となる可能性の高いデータは、当該ユーザがオリジンサーバ上でアクセスした最後のデータである確率が高い。この考えに基づけば、オリジンサーバにおいてユーザリストを作成する際に、そのデータを最後にアクセスしたユーザのみをリストアップし、そのユーザにのみ差分データをプッシュすることによって、ユーザにとって重要なデータの更新情報のみを送信するようにするこ

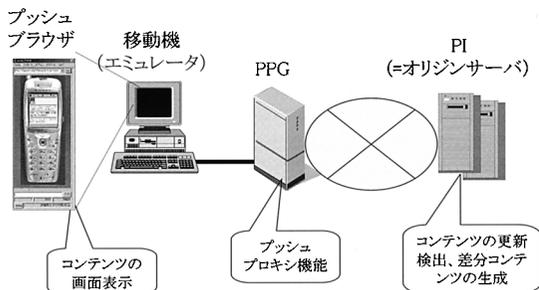


図 12 チャットアプリケーションデモシステム

Fig. 12 Demonstration system of chat application.

とも可能である。

移動機は、差分データを受信し、その処理を行った後に PI に対してリプライメッセージを送信し、データ更新が成功したか、あるいは失敗したか等の情報を含めることも可能である。この方法を用いれば、PI は移動機へのプッシュの成功、不成功を確認できるとともに、その理由を知ることができ、次回からのプッシュに反映させることが可能である。

本提案に基づく差分プッシュ方式は、2 章において説明した提案に基づくオリジンサーバ起動型のプッシュを用いて動作する。したがって、オリジンサーバにおいてデータが更新された場合に、即時にデータを移動機に提供することが可能であり、移動機においてつねに最新のデータを保持することが可能である。また、データが更新された場合に差分プッシュを実行することになるため、スマートプル方式を利用した従来の差分プッシュ方式で問題となっていた、定期的に問合せすることによる不必要な問合せが発生しないという点を特徴とする。

#### 4.3 テストベッドの試作

提案したプロトコルおよび差分プッシュ方式を基に試作したテストベッドについて紹介する。テストベッドでは、データの更新があった場合に、その更新部分を自動生成し、移動機へとプッシュを行うことが可能である。試作では、アプリケーションの例としてチャットを実現するアプリケーションの構築を行った。図 12 を用いてそのチャットアプリケーションの概要を説明する。チャットアプリケーションでは、移動機の画面表示のためのコンテンツ記述方法として XHTML を想定している。チャットアプリケーションでは、あるユーザが新たな発言を行った場合にその新たな発言内容を差分データとする XSLT 差分データを自動生成し、チャットに参加しているメンバに対してプッシュを行う。チャットにおける新たな発言は、決まった更新箇所にデータが追加されるような単純な XML デー



図 13 画面更新イメージ

Fig. 13 An image of the display renewal.

タの更新であり、更新がある箇所の追加情報を XML テンプレートで記述するといった単純な方法で XSLT 差分データを生成できる。その XSLT 差分データを受け取った移動機は、XSLT 変換を行うことで古いデータからの更新を行い、新しい発言内容が画面上に反映される仕組みになっている。図 13 は、あるユーザの発言により画面が更新された様子を示している。

テストベッド試作により、差分データの生成、差分データのプッシュ、および旧データと差分データからの新データの生成がすべて自動で行えることを確認した。図 13 に示す例における XSLT 差分データサイズは 793 バイトであり、これに対する更新後データのサイズは 2,090 バイトであった。以上のことから、差分データ XSLT で表現することにより、1297 バイトのデータが削減された(図 14)。差分プッシュ方式では、図 14 に示す XSLT 差分データを更新前 XHTML データに適用することによりデータが更新される。このデータサイズの差は、チャットアプリケーションを用いた場合の例であり、用いる XHTML データにより傾向は異なるが、一般的にデータの差分を用いることによりデータサイズの減少を図ることができる。また、チャットデータを差分データの生成時間と、旧データと差分データから新データの生成時間を測定したところ、差分データを生成しない場合と比較して、110 ms 余分に要することが分かった。この時間によるユーザへのサービス性の低下への影響は小さく十分実用性があると考えられる。

#### 5. ま と め

これまでの研究では、IMT-2000 等の高速無線ネットワークにおける WAP1.x プッシュプロトコルの性能評価は行われていなかった。本研究では、WAP1.1 プッシュプロトコルと HTTP/TCP をベースとした提案プッシュプロトコルと比較することにより、その

## XHTMLデータ

```

<?xml version="1.0" encoding="EUC-JP"?>
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"
"http://www.wapforum.org/DTD/xhtml-mobile1.0.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
</head>
<body style="background-color:#ff00">
<form>
(フォーム定義省略:703バイト)
</form>
<hr/>
<div style="color:#ff0000">
ひろこ:Thank you!(C_¥) (2001/05/15 10:46:16)</div>
<hr/>
<div style="color:#ff0000">
ひで:いつもの場所 (2001/05/15 10:44:03)</div>
<hr/>
<div style="color:#800080">
ひろこ:どこで? (2001/05/15 10:43:30)</div>
<hr/>
<div style="color:#ff0000">
けい:今度のバスケットは10時からだよー(o) (2001/05/15 10:43:10)</div>
<hr/>
<div style="color:#ff0000">
ひろこ:今度のバスケットっていつだったっけ?おしえて!! (2001/05/15 10:41:01)</div>
<hr/>
<div style="color:#ff0000">
ひで:こんどのばすけたのしみだね、久しぶりに体動かせるし (2001/05/15
10:39:31)</div>
<hr/>
</form>
(フォーム定義省略:346バイト)
</form>
</body>
</html>

```

2090バイト

新たな  
発言内容1297バイト  
削減

## XSLT差分データ

```

<?xml version="1.0" encoding="Shift_JIS"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xhtml="http://www.w3.org/1999/xhtml"
exclude-result-prefixes="xhtml">
<xsl:output (アウトプット定義省略) />
<xsl:template match="@*|node()">
<xsl:copy>
<xsl:apply-templates select="@*|node()" />
</xsl:copy>
</xsl:template>
<xsl:template match="/xhtml:html/xhtml:body/xhtml:hr[1]">
<xsl:copy>
<xsl:apply-templates select="@*|node()" />
</xsl:copy>
</xsl:template>
<div style="color:#ff0000">
ひろこ:Thank you!(C_¥) (2001/05/15 10:46:16)</div>
<hr/>
</xsl:template>
</xsl:stylesheet>

```

793バイト

新たな  
発言内容

図 14 チャットアプリケーションにおける XSLT 差分データの例  
Fig. 14 An XSLT delta file example on chat application.

性能を明らかにした。さらに本論文では、IMT-2000等の高速無線ネットワークに適した移動機向けプッシュプロトコルを新たに提案した。提案方式であるHTTP/TCPをベースとしたプッシュプロトコルは、IMT-2000ネットワークを想定したシミュレーション環境においてWAP1.1のプッシュプロトコルと比較し、プッシュ完了までの通信時間が短くなることを示し、特にデータサイズが大きいデータの場合には提案方式が有効であることを示した。我々は、本研究結果に基づいた提案プッシュプロトコルをWAPフォーラムに提案し、WAP2.0プッシュプロトコル仕様は、本提案をベースとして採用された。

さらに本論文では、XHTMLとプッシュ機能を組み合わせさせた差分プッシュ方式を提案し、テストベッド試作によりその有効性を検証した。

クライアントが増えた場合のスケラビリティに関しては、今後の検討課題としたい。特に複数クライアントへ同時配信を実現する際のマルチキャスト方式<sup>17),18)</sup>と本方式の共存方式等について検討し、それとあわせたスケラビリティに関する詳細な検討について今後の検討課題とする予定である。

## 参 考 文 献

- 1) WAP Forum: WAP Architecture Specification, WAP-100-WAPArch-19980430-a (1998).
- 2) 鈴木偉元ほか: 無線シミュレータを用いたWAPプロトコルの性能評価, *IPSSJ Symposium Series (DICO 2000)*, Vol.2000, No.7, pp.283-288 (2000).
- 3) 上野英俊ほか: WAPプロトコルの実装とプッシュ型アプリケーションへの適用, *IPSSJ Symposium Series (DICO 2000)*, Vol.2000, No.7, pp.289-294 (2000).
- 4) 石川太朗ほか: W-CDMA向けTCPプロファイル, 情報処理学会研究報告, Vol.2000, No.112, pp.17-24 (2000).
- 5) Fielding, R., et al.: Hypertext Transfer Protocol — HTTP/1.1, RFC2616 (1999).
- 6) World Wide Web Consortium: XHTML 1.0: Extensible HyperText Markup Language (2000).
- 7) World Wide Web Consortium: XSL Transformations (XSLT) Version 1.0 (1999).
- 8) World Wide Web Consortium: HTML 4.01 Specification (1999).
- 9) World Wide Web Consortium: Extensible Markup Language (XML) 1.0 (1998).
- 10) WAP Forum: WAP Push Architectural Overview, WAP-165-PushArchOverview-

- 19991108-a (1999).
- 11) WAP Forum: Push Proxy Gateway Specification, WAP-151-PPGService-19990816-a (1999).
  - 12) WAP Forum: Push Access Protocol Specification, WAP-164-PAP-19991108-a (1999).
  - 13) WAP Forum: Wireless Datagram Protocol Specification, WAP-200-WDP-20000219-a (2000).
  - 14) WAP Forum: Wireless Session Protocol Specification, WAP-203-WSP-20000504-a (2000).
  - 15) WAP Forum: Push OTA Protocol Specification, WAP-189-PushOTA-20000217-a (2000).
  - 16) 神場知成ほか：プッシュ型とプル型を統合したパーソナライズ情報配信システムの提案と実装，情報処理学会論文誌，Vol.39, No.5, pp.1514-1522 (1998).
  - 17) Deering, S.: Host Extensions for IP Multicasting, RFC1112 (1989).
  - 18) Fenner, W.: Internet Group Management Protocol Version 2, RFC2236 (1997).
  - 19) 3rd Generation Partnership Project (3GPP): RLC protocol specification (2001).
  - 20) WAP Forum: Wireless Transaction Protocol Specification, WAP-201-WTP-20000219-a (2000).
  - 21) WAP Forum: Push Architectural Overview, WAP-250-PushArchOverview-20010703-a (2001).
  - 22) Mogul, J.C., et al.: Potential benefits of delta-encoding and data compression for HTTP, *Proc. SIGCOMM 97* (1997).
  - 23) World Wide Web Consortium: Document Object Model (DOM) Level 2 Core Specification Version 1.0 (2000).
  - 24) 上野英俊ほか：XML コンテンツの差分生成法とプッシュ型配信への応用，情報処理学会研究会モバイルコンピューティングとワイヤレス通信，Vol.2002, No.24, pp.107-114 (2002).

## 付 録

### 無線向け TCP プロファイルの概要

文献 4) では，従来のアクセス回線と比較して，広帯域でかつ比較的高遅延という特徴がある W-CDMA において，回線性能を十分に引き出すことが可能な TCP のプロファイルを提案している．文献 4) では，TCP プロファイルの各要素（バッファサイズの拡大，初期輻輳ウィンドウ拡大，MTU (Maximum Transfer Unit) サイズの拡大，セレクティブアクノレジメント方式の採用）に対する評価を述べ，W-CDMA の回線性能を十分に引き出すことが可能であることを示している．W-CDMA のような高速な無線ネットワーク

は，有線ネットワークと比較してエラー率が高いため，セレクティブアクノレジメント方式のようなパケット欠落に対する耐性の高い方式が有効である．同様に，無線ネットワークは，比較伝播遅延時間が大きいので，初期輻輳ウィンドウサイズ拡大や帯域遅延積に基づいた適切なバッファサイズの拡大が有効である．

(平成 13 年 11 月 9 日受付)

(平成 14 年 9 月 5 日採録)

## 推 薦 文

本論文は，IMT-2000 等の高速移動網向けの，HTTP/TCP を用いたプッシュプロトコルを提案し，シミュレーション環境において WAP1.1 との性能比較を行っている．また，即時性とネットワーク利用効率を高める差分プッシュ方式を提案している．プッシュプロトコルに関して，詳細な性能評価と考察がなされているとともに，差分プッシュ方式に関してもテストベッドの試作により実用性が示されており，論文として評価できる．( MBL 研究会運営委員 寺岡 文男 )



高橋 修 (正会員)

1975 年北海道大学大学院情報工学専攻修士課程修了．同年日本電信電話公社入社．現在，NTT ドコモマルチメディア研究所主幹研究員．主としてモバイルマルチメディア通信サービスとプロトコルの研究開発に従事．電子情報通信学会会員．



上野 英俊 (正会員)

1997 年筑波大学第三学群情報学類卒業．1999 年同大学院工学研究科電子・情報工学専攻修士課程単位取得退学．同年より NTT ドコモマルチメディア研究所勤務．モバイルインターネットアプリケーションとプロトコルの研究に従事．電子情報通信学会会員．



石川 憲洋(正会員)

1978年京都大学工学部情報工学科卒業。1980年同大学院工学研究科情報工学専攻修士課程修了。同年日本電信電話公社(現NTT)入社。現在,NTTドコモマルチメディア研究所に所属。ATM,インターネットプロトコル,マルチメディア通信,モバイルインターネット等の研究開発に従事。電子情報通信学会会員。



角野 宏光

1997年名古屋大学大学院工学研究科修士課程修了。同年よりNTTドコモマルチメディア研究所所属。以来,モバイル向け通信プロトコル,ゲートウェイの研究開発に従事。電子情報通信学会会員。



鈴木 偉元

1989年千葉大学工学部機械工学科卒業。1991年同大学院修士課程修了。同年NTT入社。以来,ビデオオンデマンドシステムの研究開発に従事。現在,NTTドコモマルチメディア研究所・主任研究員。モバイルインターネットの研究開発に従事。電子情報通信学会会員。



水野 忠則(正会員)

1945年生。1968年名古屋工業大学経営工学科卒業。同年三菱電機(株)入社。1993年静岡大学工学部情報知識工学科教授,現在,情報学部情報科学科教授。工学博士。情報ネットワーク,モバイルコンピューティング,放送コンピューティングに関する研究に従事。著書としては「プロトコル言語」(カットシステム)、「コンピュータネットワーク概論」(ピアソン・エデュケーション)等がある。電子情報通信学会,IEEE,ACM各会員。当会フェロー。