Vol.2010-GN-75 No.3
2010/3/18

双方向Ｗｅｂコミュニケーションにおける プロキシキャッシュを利用した負荷分散方式

城島貴弘　大西健夫　中島一彰
NEC サービスプラットフォーム研究所

Web サーバとブラウザで構成される Web 会議サービスや情報共有サービスなどの双方向コミュニケーションシステムでは，複数の参加者がいる場合には，コミュニケーションの最中にアドホックにブラウザを起動して途中参加する．途中参加者がブラウザで Web サーバにログインするとき，キャッチアップ処理として，サーバ上のメッセージを受信し，ブラウザの状態を最新状態に同期させる．このキャッチアップ処理は，突発的な負荷をサーバに発生させ，ほかのブラウザ間のメッセージ送受信に遅延が発生させる要因になっている．本論文では，キャッチアップ処理に対し，ブラウザと Web サーバとの間に存在するプロキシを利用し，ブラウザがキャッチアップ時にプロキシにキャッシュされたメッセージを読み込むことで，サーバの突発的な負荷発生を回避する方式について提案する．

A load-reducing method of Bi-Directional Web communication using the cache on web proxy

Takahiro Shiroshima　Takeo Oonishi and Kazuaki Nakajima
Service Platforms Research Laboratories, NEC Corporation

In Bi-directional web communication system such as Web-based conferences and data sharing services, users can launch a web browser at any time while a session is continuing. When a web browser starts by a user, it causes the catch-up processing which sends many messages from the server. So the catch-up processing generates unexpected load, it has the possibility of disturbing the other processing on the server. In this paper, we propose the method which uses the proxy cache between browsers and the server. The method enables to evade unexpected load on the server that a browser reads messages on the proxy cache at the catch-up processing.

1. はじめに

近年，インターネットに代表される IT 技術と，電話に代表されるネットワーク技術が融合した IT/ネットワーク融合による新しいコミュニケーションサービスの創出が試みられている．このような新しいコミュニケーションサービスとして，高品位テレビ電話[1]や IPTV[2][3][4]など広帯域なネットワークを活かしたビジュアルコミュニケーションが議論されており，そのひとつとして携帯電話上でのリッチコミュニケーションサービスを実現するための標準仕様として GSMA(GSM Association)では RCS(Rich Communication Suite)ワーキンググループ[5]にて，通話中の写真や動画の共有サービスが検討されている．

このようなリッチコミュニケーションを実現する方式としては，既存端末との相互接続性を確保するために Web を利用することが多くなっている．サービス提供形態としてブラウザを用いた Web サービス化が進んでおり，RIA(Rich Internet Application)と呼ばれるブラウザ上で動作する Flash や Java アプレット，Ajax(Asynchronous Javascript+XML)を用いたアプリケーションが普及している．この RIA を利用してコミュニケーションサービスを実装することにより，クライアントプログラムが不要となり，ユーザへのサービス展開が容易となる．しかし，RIA によるプログラムは，HTTP プロトコルを用いて Web サーバと通信することを前提としており，VoIP や IM のようにユーザ同士がコミュニケーションを取るために必要なクライアントからクライアントへの通信経路を持っていない．

RIA によるクライアントからクライアントへの通信を可能とし，ブラウザを利用したコミュニケーションサービス(双方向 Web コミュニケーション)を実現するためのシステムとして，Web コミュニケーションシステムが提案されている[6]．Web コミュニケーションシステムでは，Web サーバからブラウザへの能動的な通信を可能とするために双方向 HTTP 通信を用いる．双方向 HTTP 通信を実現する方式は各種存在するが，HTML 上の JavaScript 記述から XMLHttpRequest オブジェクトを用いて非同期で Web サーバと通信し，ポーリングにより Web サーバ上のメッセージを取得する手法が一般的に広く用いられている．Web サーバは双方向 HTTP 通信を用いて，ブラウザからメッセージ送信を受信し，他のブラウザに対してメッセージを中継して転送することで，Web サーバを経由した擬似的なブラウザからブラウザへの通信を可能とする．

双方向 Web コミュニケーションでは，ユーザがブラウザ上の RIA を操作すると，その操作が操作メッセージとして Web サーバに送信される．Web サーバはユーザのコミュニケーション相手が操作するほかのブラウザに対して操作メッセージを転送し，操作メッセージを受信したブラウザがそのメッセージに記述された操作内容を RIA 上で実行することでブラウザの表示を同期させることが可能となる

Vol.2010-GN-75 No.3
2010/3/18

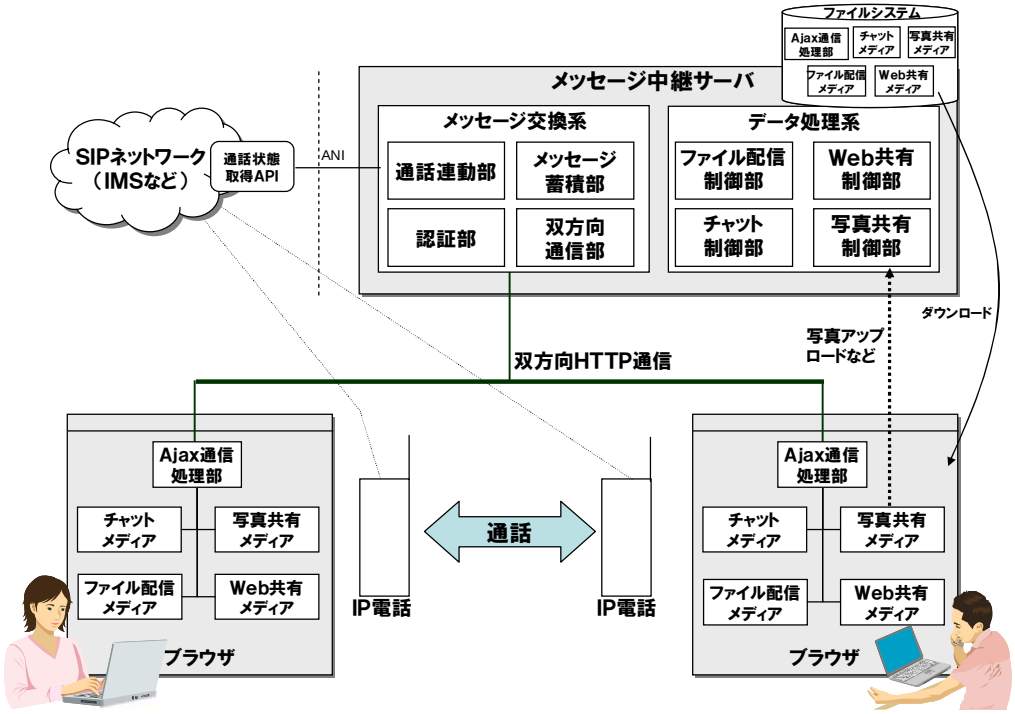

図 1 Web ベースの双方向コミュニケーションシステム

2. Web ベースの双方向コミュニケーションシステム

我々は IT/ネットワーク融合によるコミュニケーションサービスをより簡単にユーザに提供するためのプラットフォームとして Web ベースの双方向コミュニケーションシステムを提案している．本システムは，IMS が持つ ANI(Application Network Interface)[7][8]を利用して SIP ネットワークの通話セッションを取得し，その通話セッションとブラウザ間のメッセージ中継を連動させることで，通話相手と音声コミュニケーションをとりながら，ブラウザを利用した双方向 Web コミュニケーションを可能とする．

図 1 に Web ベースの双方向コミュニケーションシステムの概要を示す. **図 1** 中のメッセージ中継サーバは Web サーバとして動作し，双方向 HTTP 通信によりブラウザからのメッセージを受信して，同じく双方向 HTTP 通信によりメッセージ中継サーバに

図 2 双方向コミュニケーションの実行画面

接続しているほかのブラウザに対してメッセージを転送する．

また，ブラウザ上ではメッセージ中継サーバから RIA で実装されたコミュニケーションメディア(以下メディア)がダウンロードされ，双方向 Web コミュニケーションにおけるユーザインターフェイスを提供する．**図 2** にブラウザで動作するメディアについて示す．図ではテキストを送受信できるチャットメディア，任意のファイルをほかのユーザに配信可能なファイル配信メディア，同じ写真を表示し，その写真に対して任意図形の書き込みが可能な写真共有メディアが動作している．また，同じ Web ページを閲覧し，スクロールやフォームへの入力を同期可能な Web 共有メディアも動作している．ブラウザ上のそれぞれのメディアは，JavaScript により構成された Ajax 通信処理部を通してメッセージ中継サーバと双方向 HTTP 通信にて通信する．また，メッセージ中継サーバは，データ処理系と，メッセージ交換系とに分けて構成される．

データ処理系は，ブラウザ上で動作するメディアと対応した制御部が動作し，それぞれのメディアが要求するデータ処理を行う．たとえば，写真共有制御部は，ブラウザの写真共有メディアからほかのブラウザと共有して表示したい JPEG などの写真フ

ファイルをアップロードにより受け取り，写真共有メディアで共有操作可能[a]な形式に変換して蓄積し，各ブラウザの写真共有メディアからの写真ダウンロード要求に応じてそのファイルを配信する．

メッセージ交換系は，ANI を通して SIP ネットワークから通話セッション情報を取得して現在の通話状態を管理する通話連動部，ブラウザからのアクセス時にそのブラウザを利用しているユーザが利用している電話の電話番号とそのアクセスとを関連付ける認証部，ブラウザ上の Ajax 通信処理部と双方向 HTTP 通信によりメッセージの送受信を行う双方向通信部，双方向通信部で受信したメッセージを一時的にキューに蓄積しておくメッセージ蓄積部とで構成される．

ブラウザ上のメディアから送信されたメッセージは全てメッセージ蓄積部のキューに蓄積される．このキューは，通話のセッションと１：１で対応しており，ユーザ間で通話が開始されると，その通話セッションに対応したキューが作成され，通話が終了するとそのキューが消去される．キューによるメッセージ管理により，双方向 Web コミュニケーションシステムにおいて以下の機能が実現されている．

- **キャッチアップ**
  通話中のユーザは，任意の時点でブラウザを起動することができ，ブラウザ上のメディアはキューに蓄積されたメッセージを最初から取得し，実行することにより，その時点までにユーザ間で行われた操作をいつでも再現できる．
- **マイグレーション**
  ある端末でブラウザを終了し，別の端末で再度ブラウザを起動してキャッチアップ処理を実行することにより，ユーザは双方向 Web コミュニケーションで利用する端末を任意の時点で変更することができる．
- **マルチデバイス**
  キューに蓄積されたメッセージは，何度でも各ブラウザに送信可能であるため，一人のユーザが複数のブラウザを同時に利用することが可能となり，ユーザは複数の端末上のブラウザを同時に用いて双方向 Web コミュニケーションを実行することができる．

3. 双方向 Web コミュニケーションにおける課題

双方向 Web コミュニケーションでは通話中のユーザがブラウザを起動したタイミングでキャッチアップを実行し，ブラウザ上の各メディアの状態を最新の状態に更新させる処理が実行される．メッセージ中継サーバは，ブラウザからのキャッチアップ要求に対し，キューに蓄積されたメッセージを規定のサイズ(バンドルサイズ)毎に集約し，複数回に分けてクライアントに送信する．この処理はメッセージ中継サーバに対して突発的な負荷を発生させるため，ほかのブラウザ間のメッセージ送受信に遅延が発生するといった，コミュニケーションの品質を劣化させる可能性がある．

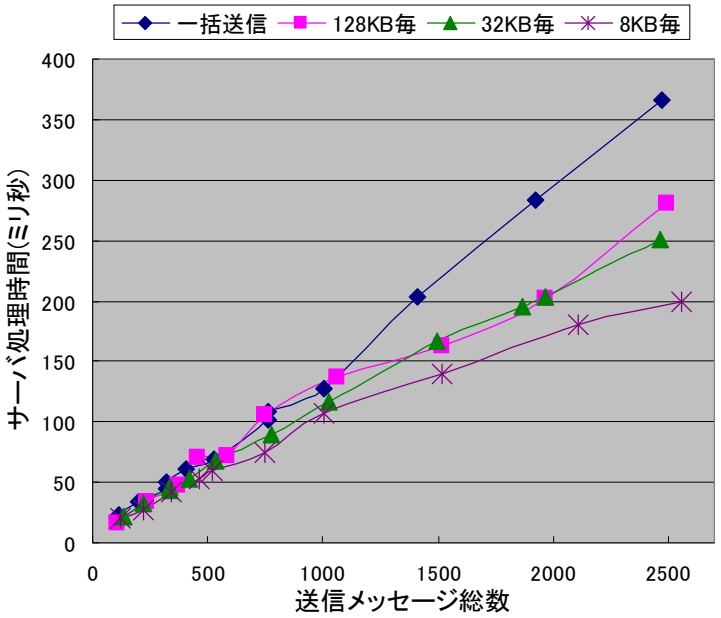

図 3 キャッチアップ時のサーバ負荷

突発的な負荷を避けるためには，バンドルサイズを小さくし細かくブラウザに送信することで負荷を均す方法が考えられるが，その場合ブラウザでのキャッチアップ終了までの時間が延びてしまう．以下でキャッチアップ処理のサーバ負荷およびキャッチアップが終了するまでの時間について，試作環境にて評価を実施した．

3.1 キャッチアップ時のサーバ負荷

図 3 は，**表 1** に示した評価環境において，送信するメッセージ数に応じてキャッチアップ時にかかるメッセージ中継サーバの負荷を，バンドルサイズ毎に計測したものである．評価は Web 共有メディアで Web ページをスクロールすることでスクロール操作メッセージを指定回数発生させ，再度ブラウザを起動してキャッチアップを実行させたときのメッセージ中継サーバでの処理時間を計測した．なお，一つのメッセージサイズはおおよそ 150 バイト程度である．

a 現在写真共有メディアは Flash を利用して作成されているため、SWF ファイルに変換される。

Vol.2010-GN-75 No.3
2010/3/18

表 1 評価環境

	メッセージ中継サーバ	端末
CPU	Intel Xeon X5460(3.16GHz)	Intel Core 2 6700 (2.66GHz)
RAM	40GB	4GB
OS	Windows 2003 Server SP2	Windows XP SP2
その他	Oracle Weblogic Server 11g	Internet Explorer 8 Flash 10

評価実験の結果，サーバ側の負荷としては 8K バイト毎にバンドルしてクライアントに送信した場合が最もサーバ側の負荷が低く，バンドルサイズが上がる毎にサーバの処理時間が延びることが分かった．これはバンドルサイズが上がると，キューからのメッセージ取得時の文字列操作や実際の送信処理におけるメモリコピーの際に処理負荷がかかるためと考えられる．

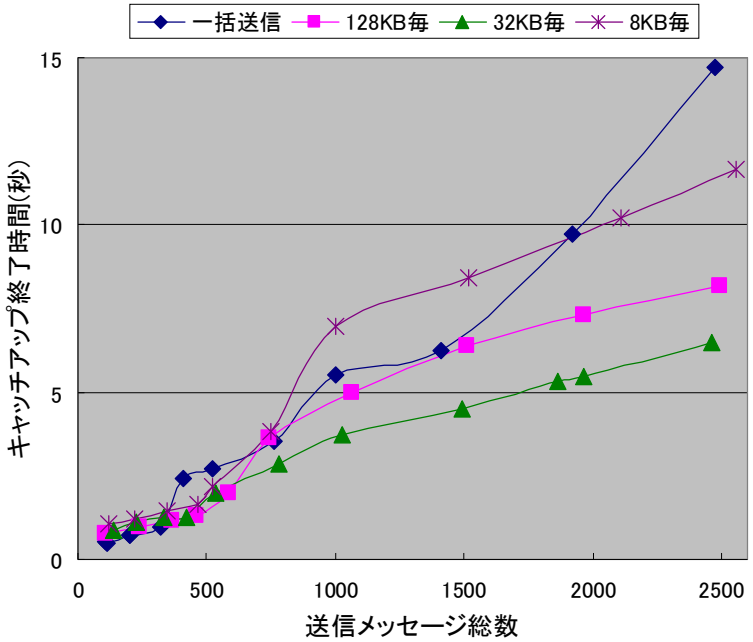

図 4 キャッチ終了までの時間(通信遅延 数ミリ秒)

3.2 ブラウザでキャッチアップが終了までの時間

次に**図 4** は，3.1 節の評価において，ブラウザ側でキャッチアップ処理が実際に終了するまでの時間を LAN 環境(通信遅延が数ミリ秒程度)にて計測した図である．ブラウザ側では，メッセージ送受信に加えて，JavaScript でのメッセージ解析処理および各メディア(ここでは Web 共有メディア)でのメッセージ実行処理が加わるため，キャッチアップが終了するまでに相応の時間が必要となる．

また，**図 5** は，300 ミリ秒の遅延がある場合(携帯電話でのパケット通信の遅延に相当)のブラウザでのキャッチアップ終了までの時間を，シミュレーションにより計測した結果である．

ブラウザでのキャッチアップ処理は，バンドルサイズが小さくなるとその分通信回数が多くなり，通信遅延によりキャッチアップ終了までに多くの時間がかかるようになる．サーバでの処理時間が一番短い 8KB でバンドルした場合は，32KB や 128KB でバンドルした場合に比べ，キャッチアップ終了までにかなりの時間を要することがわ

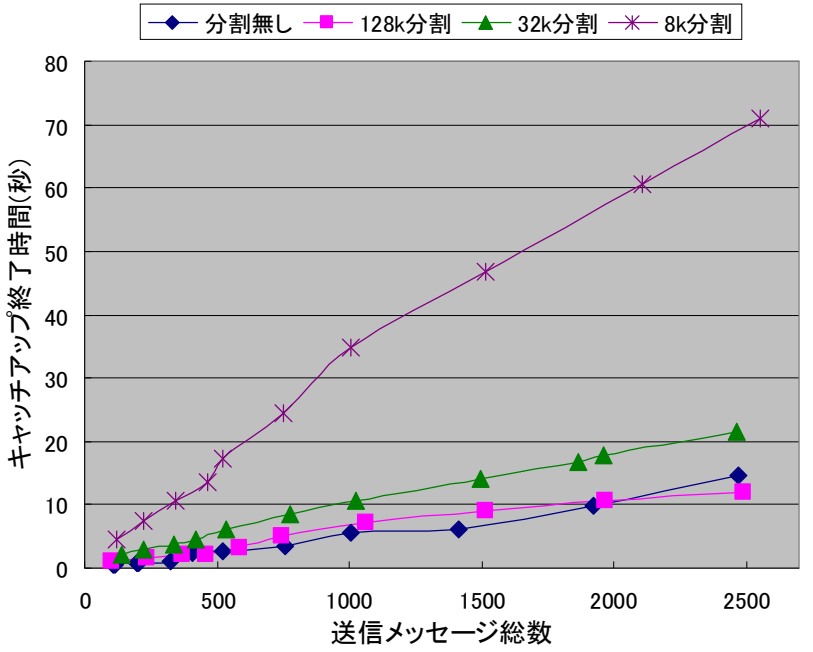

図 5 キャッチアップ終了までの時間(通信遅延 300 ミリ秒)

かった．

ただし，バンドルサイズが小さい方がブラウザでの JavaScript を用いたメッセージ解析時の文字列処理において，メモリコピーなどの処理負荷が小さくなり，たとえば**図 4** のように通信遅延が少ない環境では，一括送信(バンドルなし)や 128KB でバンドルするよりも 32KB でバンドルした方がキャッチアップ終了までの時間が短くなる．

4. キャッチアップ処理の負荷分散方式

メッセージ中継サーバでキャッチアップ処理時に複数のメッセージをバンドルして送信する際，メッセージ中継サーバの処理負荷を下げるためには 8K バイト程度の比較的小さいサイズでバンドルする必要がある．しかし，ブラウザのキャッチアップを早く終了させるためには 32KB～128KB 以上のバンドルサイズとする必要がある．特に遅延が大きい環境では，より大きなバンドルサイズを用いる必要があり，これはキャッチアップ時にメッセージ中継サーバに対して大きな負荷をかける．

ここでキャッチアップ時に送信するメッセージは，キューに蓄積されたメッセージの再送である点に着目する．ほかのブラウザがキャッチアップを実行したことがあるならば，そのキャッチアップ時に作成したメッセージをキャッシュすることで，そのキャッシュを再送信することにより，メッセージ中継サーバの負荷を考慮せず，ブラウザのキャッチアップ終了時間が最速となるバンドルサイズで送信することが可能となる．

その一つの方法としては，メッセージ中継サーバにおいてキャッチアップ用にバンドルしたメッセージを，キューとは別にキャッシュしておき，ブラウザからのキャッチアップ要求時にキャッシュしたバンドルメッセージを送信する方式が考えられる．しかし，この方式では，メッセージのバンドル処理の負荷は軽減されるが，バンドルされたメッセージの送信処理は必要であり，メッセージ中継サーバへの一時的な負荷を避けることができない．また，メッセージの蓄積に必要な記憶領域を倍以上消費してしまう．

4.1 プロキシキャッシュによるキャッチアップ負荷の分散

Web サーバの負荷分散のために用いられるリバースプロキシ[9]をメッセージ中継サーバの前段に用意し，過去ブラウザに対して送信したキャッチアップ用にバンドルされたメッセージを，リバースプロキシのキャッシュに保存させ，ブラウザからのキャッチアップ要求があった場合，そのキャッシュを読み込むようにブラウザに指示することで，メッセージ中継サーバの負荷を回避する．

以下，プロキシキャッシュを用いたキャッチアップ処理について説明する．

① ブラウザからキャッチアップ要求が送信され，リバースプロキシ経由でメッセージ中継サーバに受理される．

この時ブラウザから送信される URL が，キャッシュにアクセスするための URL となる．

② ブラウザを利用しているユーザの通話セッションを認証部で判別し，その通話セッションと関連づけられたキューを特定する．

③ ②で特定されたキューに対して過去キャッチアップが実行されているか?
→ キャッチアップが実行済みの場合は⑧以下を実行する．

※ 以下は初回のキャッチアップ時の処理．

④ キャッチアップ用のメッセージを規定のバンドルサイズで作成する．

⑤ ブラウザからのリクエスト URL(①もしくは⑦でブラウザから送信されてきた

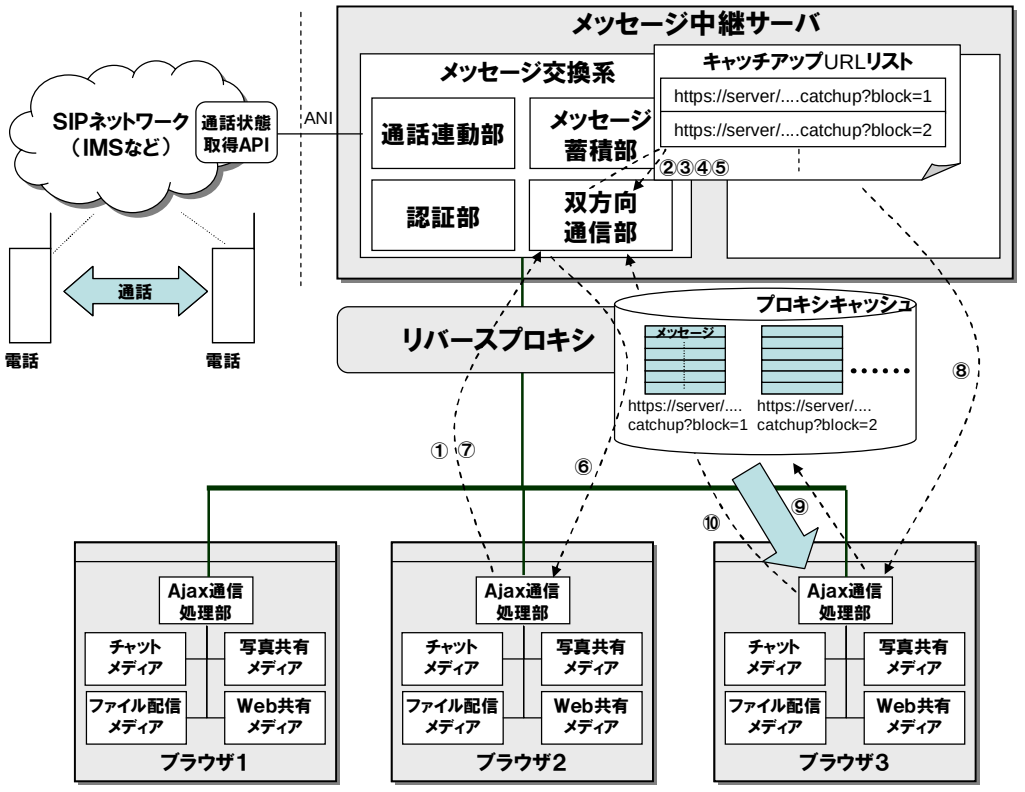

図 3　プロキシキャッシュを用いたキャッチアップ処理の負荷分散処理

URL)をキューにキャッチアップ URL リストとして保存する．
⑥ バンドルされたメッセージを返信する．
このメッセージがリバースプロキシにキャッシュされる．
⑦ ブラウザから再度キャッチアップの続きの要求を受信する
→ 全てのメッセージを送信するまで④～⑦を繰り返して終了する．
※ 以下は過去にキャッチアップが実行されている場合の処理
⑧ ⑤で保存されたキャッチアップ URL リストをブラウザに送信する．
⑨ ブラウザでリスト中の URL を順番に送信してキャッシュされたメッセージを受信する．
⑩ ⑦を実行して，キャッシュされているキャッチアップからの差分を通常のキャッチアップとして受信して終了する．

上記の方式の概略を**図 6** に示す．ブラウザ２が起動してキャッチアップ処理を実行することにより，リバースプロキシ上のプロキシキャッシュにキャッチアップのメッセージが蓄積され，ブラウザ３が起動（２章におけるマルチデバイスによる別端末でのブラウザ起動，もしくは，マイグレーションによりブラウザ２が別端末で起動した場合に相当)した際に，メッセージ中継サーバから送信されてくるキャッチアップ URL リストに従って，リバースプロキシ上のプロキシキャッシュからキャッチアップメッセージを取得している．

4.2 考察

前節で述べた方式では，以下の前提条件において効率よく動作する．
a) ブラウザ上で動作するメディアの種類が全て同じ
b) バンドルされたメッセージを蓄積可能なだけの十分なキャッシュがリバースプロキシに確保されている

以下，それぞれについて考察する．

双方向 Web コミュニケーションシステムでは，ブラウザ上で全てのメディアが必ず動作する必要はなく，例えば携帯端末上のブラウザで動作するメディアはチャットだけといった利用方法も可能であり，a)の条件が成り立たない場合がある．

ブラウザ上で動作するメディアの種類が異なる場合，全てのメッセージをリバースプロキシから取得すると，無駄なメッセージの転送も含むことになる．よって，キューに保存するキャッチアップ URL リストを端末毎に用意する，また，端末毎にリバースプロキシのキャッシュを利用するかどうかを分ける，といった対処が必要である．

また，b)の条件が成り立たない場合として，一つのメッセージ中継サーバで処理するセッション数が多く，かつ，それぞれのセッションにおいて多数のメッセージが交換される場合や，リバースプロキシが双方向 HTTP 通信だけでなく，データ処理系でのデータ送受信をキャッシュしている場合などが考えられる．

リバースプロキシに十分な容量のキャッシュがない場合，キャッチアップ URL リストに基づきブラウザがキャッシュを取得しようとしても，キャッシュミスにより再度メッセージ中継サーバにリクエストが飛ぶことになる．この場合，リバースプロキシでのキャッシュ処理の分だけオーバーヘッドが発生し，逆にシステム全体の性能を押し下げてしまう．よって，事前に一つのメッセージ中継サーバで処理するセッション数と１セッションで交換されるメッセージの総量を見積り，必要十分なキャッシュサイズを用意する，また，メッセージ交換系とデータ処理系で利用するリバースプロキシを変更する，といった対処が必要となる．

5. おわりに

本論文では，IT/ネットワーク融合により，電話による音声コミュニケーションを双方向 Web コミュニケーションと連動させることで，ブラウザを利用したコミュニケーションサービスを簡単に提供する双方向 Web コミュニケーションシステムついて提案した．また，その双方向 Web コミュニケーションにおいて課題となるキャッチアップ処理時に突発的に発生する負荷について評価を実施し，最後にプロキシキャッシュを利用した負荷分散方法について示した．

今後は，本方式について評価実験を実施しその有効性を確認すると共に，4.2 節の考察で示した，動作するメディアの種別が異なる場合のキャッチアップ手法や，リバースプロキシで必要となるキャッシュサイズの見積りモデルについて検討を行っていく予定である．

参考文献

1) 高品位トリプルプレイサービス: NGN フィールドトライアル | NEC, http://www.nec.co.jp/solution/ngn/trial/te2.html
2) ETSI TS 182 027 v2.0.0: IPTV Architecture; IPTV functions supported by the IMS Subsystem （ETSI TS 182 027 v2.0.0：IPTV アーキテクチャ。IMS サブシステムがサポートする IPTV 機能）
3) ETSI TS 182 028 v2.0.0: IPTV Architecture; Dedicated Subsystem for IPTV functions. （ETSI TS 182 028 v2.0.0：IPTV アーキテクチャ。IPTV 機能専用サブシステム）
4) ETSI TS 185 009 v2.0.0: Architecture and interfaces of a customer network device for IMS based IPTV services （ETSI TS 185 009 v2.0.0：IMS 方式 IPTV サービスに採用するカスタマーネットワーク・デバイスのアーキテクチャとインタフェース）
5) Rich Communications Suite (RCS), http://www.gsmworld.com/our-work/mobile_lifestyle/rcs/index.htm
6) Breeze システムの概要, http://www.adobe.com/jp/products/breeze/productinfo/system_architecture/
7) ETSI OSA Parlay X, http://docbox.etsi.org//TISPAN/Open/OSA/ParlayX30.html
8) NGN ミドルウェアパートナープログラム, http://www.nec.co.jp/solution/ngn-pg/
9) Analysis of Design Alternatives for Reverse Proxy Cache Providers, CICIANI B, QUAGLIA F, ROMANO P(Univ. Rome), DIAS D(IBM) MASCOTS 2003, p316-323,2003