Vol.2013-IOT-20 No.1
2013/3/14

複数の配送手順を利用した ロバストなメッセージングシステムの実装と評価

甲賀 拓実[1]　石橋 由子[1]　桝田 秀夫[2]

概要：情報通信技術が発達し，場所や時間に制約されずにメッセージのやり取りができるようになった．異なる特性を持ったメッセージのやり取りをする手段が多数存在し，利用者はその状況に応じて適切な手段を選択しメッセージのやり取りをする．しかし，どの配送手段を用いれば相手に確実に届き，閲覧してもらえるかを判断することは難しい．様々な配送手段で同時にメッセージを配送すれば，ある一つの配送手段で届いていなかったとしても相手先に届く可能性が増えると考えられる．そこで本研究では，複数のメッセージの配送手順を利用し，メッセージのやり取りを実現するシステムを構築した．

キーワード：メッセージ，電子メール，ロバスト

Implementation and Evaluation of Robust Messaging System Using Various Messaging Procedure

Kouga Takumi[1]　Ishibashi Yoshiko[1]　Masuda Hideo[2]

Abstract: Information technology has developed, we can exchange messages not restricted to time or place. User can exchange messages using various messaging procedures. However, it is difficult to find the usable messaging procedures from sender's point of view because usable and appropriate procedure is changed by the environment or time. If message is delivered using various procedures in the same time, the possibility of success delivering will increase though one procedure cannot deliver. In this paper, we constructed the communication system using various messaging procedure.

Keywords: Messaging, e-mail,robust

1. はじめに

情報通信技術が発達し，場所や時間に制約されずにメッセージのやり取りができるようになった．現在では様々なメッセージのやり取りをする手段がある．利用者はその状況において適切な手段を選択しメッセージのやり取りをする．また，電話や電子メールは業務にも利用され，現代社会のインフラとしてなくてはならないものとなっている．企業や学術機関など，多くの人々が属しているような団体では，団体の管理にあたる者は構成員がどのような状態でいるか確認したいと思うことがある．以降，本稿ではこのことを安否確認と呼ぶ．送信先へメッセージを送るためには様々なサーバやネットワーク機器を経由して届けられることが多く，途中の経路に障害が発生するとメッセージが到達しない可能性がある．そのため，どの手段を用いれば相手に確実に届き，閲覧してもらえるかをメッセージ送信者が判断することは難しい．このように，利用者は様々な配送手段でメッセージのやり取りをすることができるが，どの手段が最適かということは環境，時間によって変化する．

そこで本研究では，複数のメッセージの配送手順を利用

[1] 京都工芸繊維大学大学院 工芸科学研究科
Graduate School of Science and Technology,
Kyoto Institute of Technology
[2] 京都工芸繊維大学 情報科学センター
Center for Information Science, Kyoto Institute of Technology

しロバストなメッセージのやり取りを実現するシステムを開発する．様々な手段で同時にメッセージを配送すれば，いずれかの手段によりメッセージが到達すれば相手先に届き，メッセージを読んでもらえる可能性が増えると考えられる．以降，2 章ではシステムに求められる要求，3 章では関連する技術・研究，4 章では仕様，5 章 では実装，6 章では評価，7 章では考察，最後に本論文のまとめとする．

2. システムに求められる要求

システムに求められる要求を以下に記す．

[**要求 1**] 既存の配送手段を利用すること
日常的に使われている既存の配送手段は，新たにポートを開放する必要がないと考えられる．また，電子メールや HTTP はそれぞれ違うプロトコルを使用しており，それぞれの配送手段には異なる特性があるため，同時に使用不可になりにくいと考えられる．

[**要求 2**] 利用者が配送手段を意識する必要がないこと
既存の配送手段を利用していても，利用者がどの配送手段が不通になっているかをわざわざ確認するような手間をかけさせるべきではない．

[**要求 3**] 利用者にとって操作が容易であること
利用者に新たな操作方法を習得させることを考えると，学習にかける時間も必要であり，非常時に正しく使用してもらえない可能性がある．

[**要求 4**] 配送手段の追加が容易であること
Twitter，facebook のような利用者の多いサービスが出現した時に，容易に追加できる環境であることが重要である．

[**要求 5**] 相手が受け取った，読んだことを送信者が把握できること
相手がメッセージを受信していても内容を読んでいないということ，メッセージに返信していないということが把握できれば，メッセージを送るよう催促をすることが可能となる．

3. 関連研究

安否確認の重要性は企業だけでなく，学術機関にも高まっている．名古屋大学は学生，職員を合わせると約 2 万人にもなり，大規模な企業と同レベルの人数を抱えている．そこで名古屋大学では，名古屋大学ポータルというポータル web サイトにおいて安否確認システムを運営している [1]．学生や職員は現在の情報を入力すると，管理者はその状態を確認することができる．確認がとれていない学生，職員に対しては催促のメッセージも送ることが可能である．

田丸ら [2] はオーバレイネットワーク上に安否確認システムを構築し評価している．利用者が自由にインターネット環境を使えない状態にある場合に IC カードを読み取り機にかざすことにより，IC カードに登録されている情報をサーバに送信する．この安否確認システムは携帯電話などが使用できない環境に有効であると考えられる．
また，市販のシステムとして企業が提供しているシステムも存在する．富士通が提供している緊急連絡／安否確認サービス [3] は，通常時の連絡として使えることが特色としてあげられる．他にもニシハタシステムの e-安否 [4] などがある．ニシハタシステムの e-安否は Twitter のダイレクトメール機能と連動できるのが大きな特徴の一つである．これらのツールはどの配送手段を利用するかということを選ばなくてはいけないことが多い．本システムにおいては，それぞれの配送手段をあくまで配送経路の一部と考えている．このことが既存のシステムと異なる部分である．

4. 仕様

システムに求められる要求から，本システムに用いるプロトコルを考える．要求 1 を満たすためには，既存の配送手順をそのまま利用したうえで送信するメッセージの内容を工夫し，メッセージの内容によって処理を変えることが必要だと考えられる．要求 2 を満たすためには，送信に使用する配送手順をメッセージの送信者が指定しなくて良い仕組みである必要がある．要求 3 を満たすためには，GUI などで必要事項のみ入力すればよく，メッセージや日時等が一覧で見ることができる機能が必要である．要求 4 を満たすためには，メッセージ入力部分と送信部分が分離できる構成となっている必要がある．また，要求 5 を満たすためには，相手先に届いたかどうか，内容を見たかどうかを確認できる機能が必要である．その状態を管理するにあたっても，様々な配送手段を用いることによりメッセージの到達する可能性を高める．

以上のことから，システムの状態遷移図を図 1 に示す．

5. 実装

本システムは，PHP で記述した．利用者がメッセージの入力や閲覧に用いるアプリケーションを本稿ではクライアントアプリケーションと呼ぶ．また，クライアントアプリケーション以外に外部にデータを保存するサーバを用意した．本稿ではこのサーバを中間サーバと呼ぶ．本システムの動作環境を表 1 に示す．この構成はクライアントアプリケーション，中間サーバで同一である．また，システムの構成図を図 2 に，システムのディレクトリ構成を図 3 に示す．

5.1 クライアントアプリケーション

クライアントアプリケーションは，メッセージ送信部，メッセージ受信部，メッセージ表示部，データベース，設定ファイルから構成されている．

5.1.1 メッセージ送信部

メッセージを送信する場合利用者は post.php に Web ブ

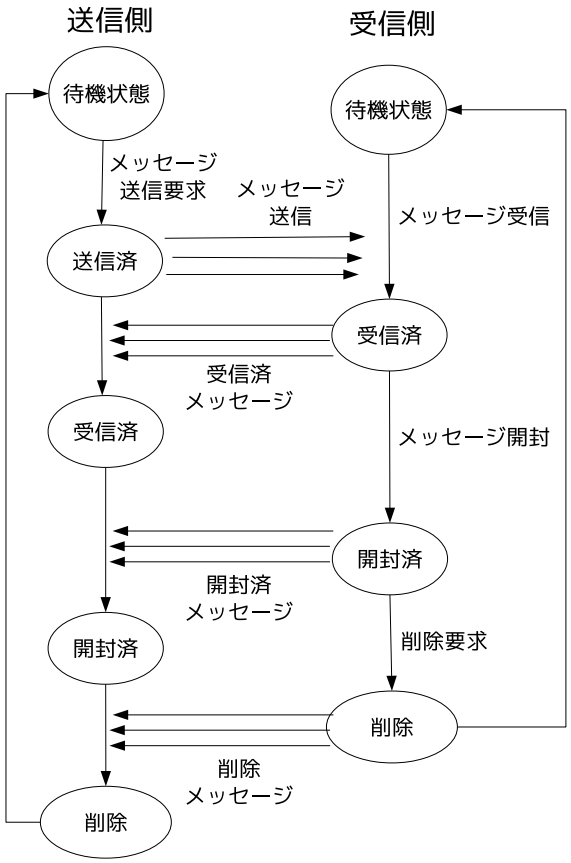

図 1 システムの状態遷移図
Fig. 1 State transition of the system

表 1 システムの動作環境
Table 1 Execution environment of the system

OS	CentOS 6.3
MySQL	5.1.66
Apache	2.2.15
PHP	5.3.3
メインメモリ	1GB
CPU	AMD Opteron 8222 3GHz

ラウザでアクセスする．Web ブラウザで表示した送信画面を図 4 に示す．

- post.php

利用者は HTML のページのフォーム上に情報を入力する．入力する内容は送信したい相手の ID と本文である．フォーム上にすべて入力をした上で送信ボタンを押すと，内部に登録されたアドレス帳の情報により，メールアドレス，送信先の IP アドレスを展開し，senddata に名前，メールアドレス，自分の IP アドレス，送信先の IP アドレス，本文のデータを渡す．なお，図 4 は，アドレス帳に情報がない場合を想定した入力画面である．

- senddata.php

post.php から渡されたデータを関数 senddata が受け取

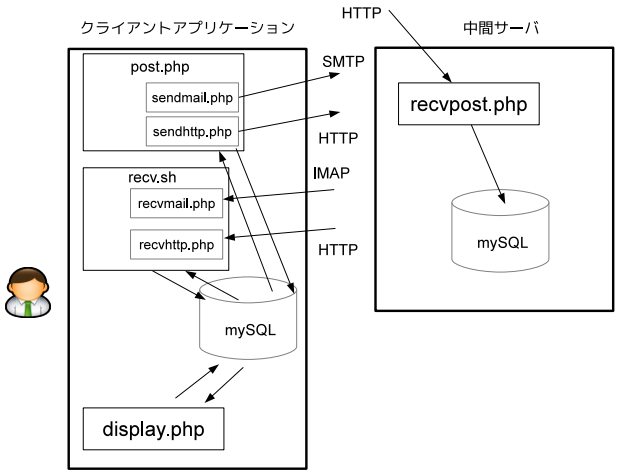

図 2 システムの構成図
Fig. 2 Structure of the system

```
/(ルートディレクトリ)
└ recvmodule/
      └ recvhttp.php
      └ recvmail.php
└ sendmodule/
      └ sendhttp.php
      └ sendmail.php
└ post.php
└ database.php
└ senddata.php
└ display.php
└ display_text.php
└ config.php
└ config.xml
└ recv.sh
```

図 3 システムのディレクトリ構成
Fig. 3 Directory structure of the system

ると，sendmodule 内ファイルを確認し，それらのファイルを require し，そのファイルに記述されている関数を実行する．この仕様により，sendmodule フォルダに入れたファイルは自動的に配送手段として利用される．そのため新たに配送手段が増えても他のソースコードに手を加える必要がない．新たに送信手段を追加するとき，ファイル名を testsend.php とするならば，ファイル内でのメッセージを送信する関数名は「.php」の部分より前を使用する．

- sendmodule/sendhttp.php

関数 senddata から呼び出される．引数で与えられたデータを HTTP の POST メソッドで送信できるように整え，中間サーバに POST する．

- sendmodule/sendmail.php

関数 senddata から呼び出される．引数で与えられたデータをメールのフォーマットにし，PHP の mb_send_mail によってメールでの送信を行う．

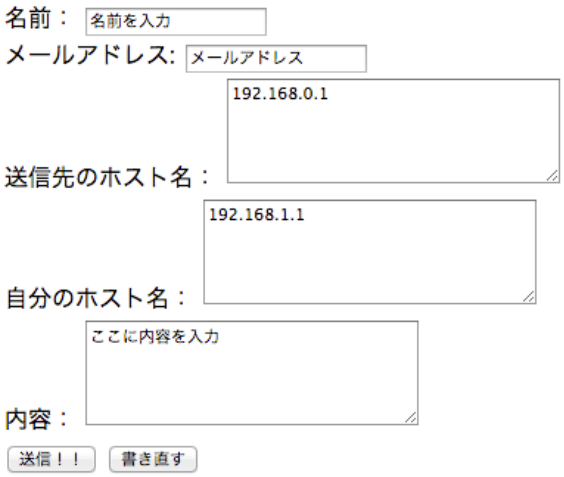

図 4 メッセージ送信画面
Fig. 4 Screenshot of sending message

図 5 メッセージ表示画面
Fig. 5 Screenshot of viewing messages

5.1.2 メッセージ受信部

メッセージを受信する場合，利用者は利用する端末上で recv.sh を CLI で呼び出す．

- recv.sh

recvmodule 内にある受信モジュールを指定された間隔で実行するシェルスクリプトである．モジュールを実行する間隔は recv.sh を実行する際のオプションの引数で指定する．

- recvmodule/recvmail.php

IMAP プロトコルによりメールサーバにアクセスし，未読のメールがあるかを確認する．未読のメールが有った場合，メッセージを受信し処理する．

- recvmodule/recvhttp.php

中間サーバに保存されたデータベースとローカルにあるデータベースを比較し，ローカルに保存されていないものがあれば取得する．

5.1.3 データ表示部

保存されたメッセージを確認するには，display.php に Web ブラウザを通じてアクセスする．

- database.php

データベースに関する処理を記した関数のライブラリ．

- display.php

データベースに保存されたメッセージを一覧で全て表示する．送信したメッセージ一覧では，メッセージが相手先に届いたかどうかなどの状態を確認することができる．

- display_text.php

メッセージ 1 件を単独で表示する．display.php で表示されたメッセージの一覧から詳細を閲覧したいメッセージのキーをクリックすると表示される．

5.1.4 データベース

テーブルを以下に示す．

- senddata

送信したメッセージについて保存するテーブル．要素は送信者の名前，メールアドレス，メッセージが送信された日付，本文，送信元の IP アドレス，送信先の IP アドレス，それぞれのメッセージを一意に識別するメッセージ ID となる．

- recvdata

受信したメッセージについて保存するテーブル．要素は senddata と同一である．

- sendstate

メッセージの現在の状態を管理するテーブル．要素はそれぞれのメッセージを一意に識別するメッセージ ID，送信元の IP アドレス，送信先の IP アドレス，メッセージの状態が更新された日付，メッセージの現在の状態となる．

5.1.5 設定ファイル

- config.php

config.xml の設定を変更するための php ファイル．利用者は config.php に Web ブラウザからアクセスして内容を変更する．

- config.xml

設定ファイルそのものとなる．

5.2 中間サーバ

- recvpost.php

クライアントアプリケーションから送信された POST データを処理する．送信されたデータが，メッセージの状態について管理する内容ならば stateDB に，メッセージそのものならば data に内容が保存される．

5.2.1 データベース

テーブルを以下に示す．

- data

クライアントアプリケーションから受信したメッセージについて保存するテーブル．要素は送信者の名前，メールアドレス，メッセージが送信された日付，本文，送信元の IP アドレス，送信先の IP アドレス，それぞれのメッセージ

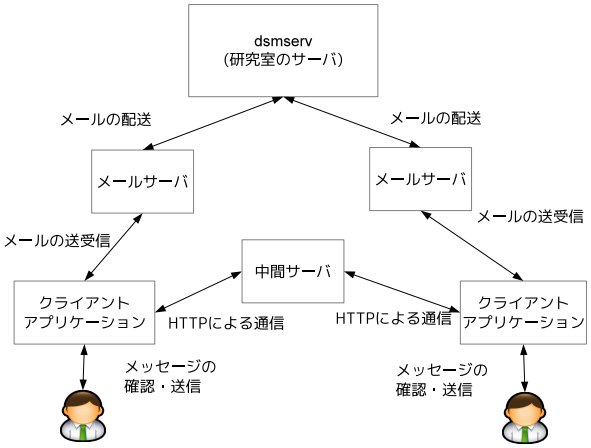

図 6 実験の構成
Fig. 6 Configuration of experiment

を一意に識別する ID となる．

- stateDB

メッセージの現在の状態を管理するテーブル．要素はそれぞれのメッセージを一意に識別する ID，送信元の IP アドレス，送信先の IP アドレス，メッセージが更新された日付，メッセージの現在の状態となる．

6. 評価

6.1 実験 1

クライアントアプリケーションを使用した場合のネットワーク通信量の変化や，メッセージのやり取りにかかる時間やその挙動を観測するために実験を行った．

6.1.1 評価環境

本実験におけるシステムの関係図を図 6 に示す．実験用に用意した 3 台のサーバは同じサブネット内に属しているため，サーバ同士で直接やり取りを行いすぐにメールの送信が終了してしまう．そのため，本実験においては MTA（Mail Transfer Agent）である Postfix の設定で relayhost を研究室に設置しているサーバに設定した．この設定を行うことにより，一旦研究室のサーバにメールが渡され，その後目的のサーバにメールが渡されるので実状に近い状態になると考えられる．また，受信モジュールの実行をする recv.sh において受信モジュールの確認間隔を 5 秒に設定し，新しいメッセージの確認を 5 秒ごとに行うようにした．

6.1.2 評価方法

メッセージを 1 通のみ送った場合ではシステムの変化が起こっているか観測しにくいため，メッセージを 1000 通送った．post.php は POST メソッドを利用しているため，curl とシェルスクリプトの記述の組み合わせにより 1000 通のメッセージを送信した．また，メッセージにそれぞれ一意な数字であるメッセージ ID を割り当てており，1 通送るごとにその数字を増やす構成としている．メッセージ

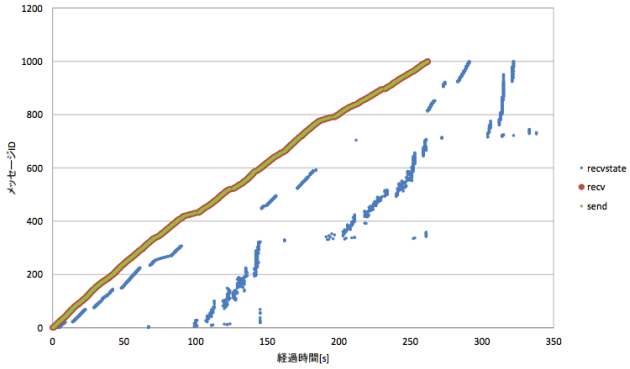

図 7 メールと HTTP を両方利用した場合のグラフ
Fig. 7 Graph of using e-mail and HTTP procedure

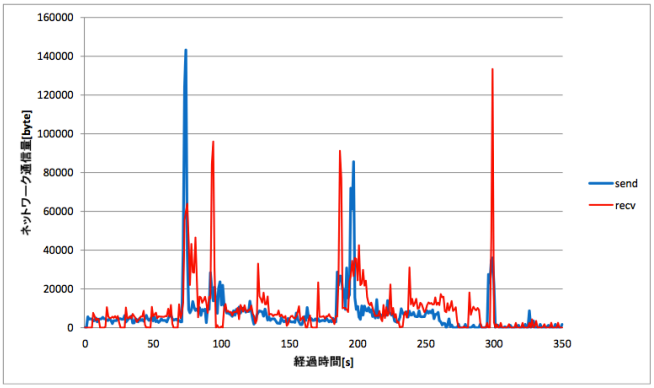

図 8 ネットワーク通信量
Fig. 8 Network traffic

送信側と受信側のクライアントで，メッセージ送信後の時刻・メッセージの現在の状態が変更された時刻を記録し，1 秒ごとに/proc/net/dev からステータスを読み出すことによりネットワークインタフェースの通信量を計測した．

6.1.3 評価結果

実験 1 の結果のグラフを図 7，図 8 に示す．図 7 は図 1 の状態遷移図の送信側において，1 通目のメッセージが送信済になった時間を 0 とし，1000 通目のメッセージが受信済になった時間まで記録している．縦軸はメッセージ ID の値である．グラフの send はメッセージを送信側が送信済になった時刻とメッセージ ID，recv は受信側で受信済になった時刻とメッセージ ID，recvstate は送信側が受信済になった時刻とメッセージ ID である．図 8 はネットワーク通信量を示したグラフであり，send はメッセージ送信側で測定したネットワーク通信量，recv は受信側で測定したネットワーク通信量である．300 秒あたりで通信量の変化は一段落している．これは図 7 からもわかるように，メッセージそのもののやり取りが終了したからだと考えられる．

6.2 実験 2

ある配送手順が正しく動作していないとき，他の配送手

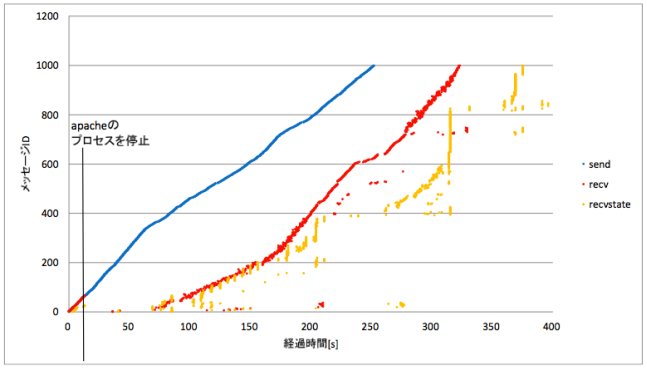

図 9 HTTP による配送を途中で停止させたときのグラフ
Fig. 9 Graph when stopped in the middle of messaging by HTTP

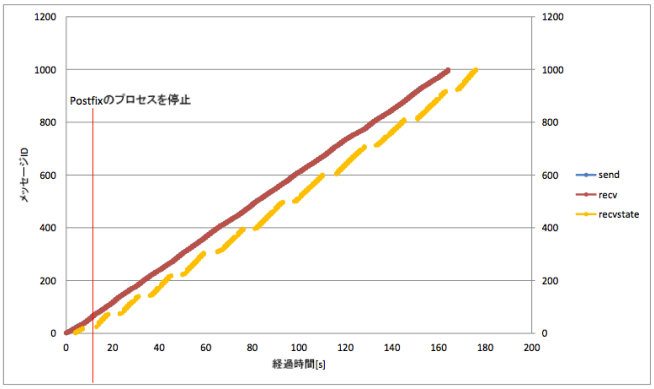

図 10 電子メールによる配送を途中で停止させたときのグラフ
Fig. 10 Graph when stopped in the middle of delivering e-mail

段で相手先にメッセージを到達させることができるかを確認した．

6.2.1 HTTP による通信が不通状態の場合

送信開始から 14 秒後に中間サーバの Apache のプロセスを停止させ，HTTP による送受信ができない状態にした．実験の結果のグラフを図 9 に示す．グラフの send，recv，sendstate は実験 1 の場合と同様の構成である．

結果，HTTP のサーバに問題が発生してもメールで引き続きメッセージの送受信を行い，全てのメッセージがやりとりすることができた．1000 通を送ったことに対して，約 250 秒で送信は完了し，約 360 秒で受信が完了し，約 400 秒で受信できたことを送信者が把握できる．また，recvstate のグラフが HTTP とメールによる通信を両方利用している場合と同様に，不規則な変化をしている．このことから，このようなグラフとなる要因が電子メールの配送においての環境に依存すると考えられる．

6.2.2 電子メールが不通状態の場合

1 通目の送信開始の約 10 秒後に，クライアントアプリケーションの送信側と接続している Postfix のプロセスを停止させ，メールが送受信できない状態にした．実験の結果のグラフを図 10 に示す．HTTP の配信を止めた時に比べ，メッセージのやり取りが全て完了するまでにかかる時間が短い．これは，電子メールを扱う場合一度のセッションには限度があるためだと考えられる．したがって，通常時は HTTP による通信を利用すればレスポンスのよい結果が得られると考えられる．

7. 考察

基本的には電子メールによる送受信よりも HTTP による送受信の方が早い結果となった．そのため，平常時の場合にはメールによる送受信が無駄になってしまうことになる．メールサーバ側で変更を行うことにより状況が変わることも考えられるが，要求 1 に反するので解決策とはならない．よって，本システム側で通信状態によって最適な配送手段を制御する機構が必要であると考えられる．一方で，HTTP の場合は送信側と受信側の双方が定期的にアクセスするため，中間サーバの負荷がネックとなりうる．本評価においては考慮に入れなかったが，多数のクライアントアプリケーションが接続すれば影響が起こりうると考えられる．よって，中間サーバの性能の測定も必要となる．

8. 結言

本研究では，複数のメッセージ配送手順を用いてロバストなメッセージングシステムの構築を行った．複数の配送手段のうち一つでも稼働していればメッセージが到達できるかを実証するために，電子メール，HTTP の中間サーバがそれぞれ稼働していない状態にし，メッセージが相手先に届くか実験を行った．実験の結果，他の手段が不通であっても正しく配送することができた．単独の配送手段を利用する場合よりもネットワークの使用量は増加するが，メッセージを相手に確実に届けたいというときに本システムを導入すれば相手に到達する可能性が高まることが期待できる．今後の課題として，実際の環境においてシステムの導入が容易に可能かどうかの検証，ユーザが問題なく使用できるか，また実際に運用した場合のネットワークへの影響などを測定することがあげられる．

参考文献

[1] 太田 芳博 , 梶田 将司 , 林 能成 , 若松 進: 名古屋大学安否確認システムの構築と運用，電子情報通信学会技術研究報告, 108(409), pp.77-82, 2009
[2] 田丸 純 , 阿部 紘一 , 島 和之 , 前田 香織: オーバレイネットワーク上に構築した安否確認システムの有効性に関する実験的評価, インターネットと運用技術シンポジウム 2012 論文集, pp. 79-85,2012
[3] 富士通，緊急連絡／安否確認サービス，入手先 ⟨http://jp.fujitsu.com/solutions/safety/bc/incident/emergency-email/⟩ (2013.2.7).
[4] ニシハタシステム，e-安否，入手先 ⟨http://www.e-anpi.jp/⟩, (2013.2.7).