耐障害性の高い非常時一斉メール送信と冗長化した蓄積型配送を実現するメッセージングシステムに関する研究

石 橋 由 子

内容梗概

近年，パソコンや携帯電話などの情報端末の低価格化や通信手段の多様化により，人から人に非常に手軽にメッセージを送ることができるようになった．またメッセージを送るツールも数多く存在している．例えば，電話やファクシミリ，電子メールや Twitter，LINE などのコミュニケーションツールがあげられる．文字だけでなく音声や画像，動画なども手軽に送ることが可能となっている．メッセージを送る場面も伝言や待ち合わせの確認といった日常的な場面で利用されるだけでなく，重要なことがらや緊急の連絡手段としても私たちの仕事や生活に不可欠なものとなっている．

災害が発生したときの安否確認の手段として，インターネットを利用したいくつかのサービスがリリースされ，実際に 2011 年に発生した東日本大震災では多くの企業で利用された．安否確認システムは，電子メールや Web インタフェースを利用したものが多い．例えば，事前に登録された電子メールアドレス宛に一斉にメールで通知し受信者から返信を求めるものや，利用者が Web インタフェースから回答を入力するという仕組みになっている．企業ではこれらを利用して，従業員の安否を把握するだけでなく，災害後の業務再開に向けてとても重要な情報となっている．緊急時に連絡を取るためには，受信者に確実にメッセージを届けなければならない．

そこでまず，電子メールについて着目すると，企業では与えられたメールアドレスを社員は使用しているが，一方学生は，大学が発行したメールアドレスを大

学のメールサーバで閲覧しているケースもあるが，普段利用しているメールアドレスに自動転送して転送先で閲覧しているケースも多い．さらに，メールアドレスを頻繁に変更する傾向にあり，その際に自動転送先に指定したアドレスの変更を忘れがちである．存在しないメールアドレス宛にメールを転送しようとするため，メールを転送できなかったことを示すエラーメールが，メールの送信者やメールサーバ管理者に送られてくるが，転送設定者には何も通知されない．そのため転送エラーが発生していることに気がつきにくくなり，無効なメールアドレス宛への転送が止まらないまま放置され，エラーメールが送り続けられている．本論文では，誤った転送設定により発生するエラーメールを抑制し，転送設定者に通知するシステムを提案する．具体的には，メールサーバがメールを転送する際に，転送したメールがエラーとなった場合に返送されてくるアドレスを，メールの転送を行っているサーバが受信できる形に書き換え，自動転送時にエラーが発生した際にエラーメールを受信してエラーが発生していることを把握することが可能となった．

次に，メールボックスの冗長化について着目した．災害時に自組織のメールサービスを利用できなくなった場合でも，利用者が可能な限りメールの利用が可能となるシステムを提案する．一般的にはメールサーバのメールを受信する部分とメールを保存するスプールは一体化しているケースが多いが，これらを分離し，スプールは物理的に離れた複数箇所に配置しスプール間でデータを同期する．スプール部分の同期を行うためにネットニュースプロトコルを利用する．メールサーバが受信したメール１通を１つのネットニュースプロトコルの記事に変換して宛先メールアドレスに対応したニュースグループに投稿する．投稿された記事はネットニュースの記事として同期を行う他のサーバに配送される．利用者はアクセス可能なネットニュースサーバにアクセスしてメールを読む．これにより，自組織のメールサー

ビスが利用できなくても，利用者はメールを受信することが可能となった．

2011 年に発生した東日本大震災では，電話やメールはほとんど利用できなかったが，Twitter は断続的に利用が可能であった．災害時に有効となる通信手段は，時代や災害の状況によって大きく変化するので，今後，災害が発生した場合にどのような通信手段が有効であるのかを現時点で想定することは非常に困難である．そこで本論文では，Twitter やメール等，既存の通信手段を複数利用してできるだけ頑強にメッセージのやり取りを行えるシステムを提案する．日常的に利用されていないシステムを災害時に利用することは困難である．緊急時の利用を想定したシステムでは性能だけでなく使いやすさにも注目する必要がある．そこで，提案システムでは学習コストが少ない電子メールのユーザインタフェースを利用する．

これらの方法を利用することにより，災害が発生しているような緊急時であっても，より確実に受信者にメッセージを届けることが可能となる．

目 次

図 目 次

表 目 次

第1章　序論

近年，情報端末の低価格化や通信手段の多様化により，人から人に手軽にメッセージを送ることができるようになった．またメッセージを送るツールも数多く存在している．例えば，電子メールや Twitter，LINE などがあげられる．文字だけでなく音声や画像，動画なども手軽に送ることが可能となっている．メッセージを送る場面も伝言や待ち合わせの確認といった日常的な場面で利用されるだけでなく，重要なことがらや緊急の連絡手段としても私たちの仕事や生活になくてはならないものとなっている．

組織において，構成員に対して日常的にあるいは緊急時に同じメッセージを送るという場面は往々にして発生する．日常的には会議や健康診断などの各種のお知らせがこれに該当する．メッセージを送る場合は，多くは電子メールの宛先や Cc や Bcc に受信者のメールアドレスを指定したり，メーリングリストの活用や，グループウェアの通知機能が利用されている．また，緊急時にメッセージを送る例として，災害が発生したときの安否確認の手段として，インターネットを利用したいくつかのサービスが提供されており，実際に 2011 年に発生した東日本大震災では多くの企業で利用された．安否確認システムは，電子メールや Web インタフェースを利用したものが多い．例えば，事前に登録された電子メールアドレス宛に一斉にメールで通知し受信者から返信を求めるものや，利用者が Web インタフェースから回答を入力するという仕組みになっている．企業では社員の安否状況の確認だけでなく、収集した情報をもとに災害後の業務再開計画に大いに役立

てている．

しかし，送信者が受信者の正しいアドレスを把握できてなかったり，受信者が普段利用している電子メールをはじめとしたメッセージを送受信するツールが災害のために利用できないため，受信者にメッセージが届けられない可能性がある．そこで本論文では，受信者に確実にメッセージを届けるための方法について論じる．メッセージを送るためには受信者のアドレスを把握しておく必要がある．電子メールを利用する場合，事前に組織の構成員が持っている電子メールアドレスを収集していても，メールアドレスの変更に伴い収集したアドレスが無効となってしまうケースがあるため，平常時より有効なメールアドレスを把握することが重要であると考えられる．

また，受信者のメールボックスが保存されているメールサーバや，メールサーバにメッセージが届くまでの経路上で何らかの被害が発生し，メールサーバにメッセージが届かない可能性もある．さらに災害時に電子メールや Twitter など，どのコミュニケーションツールが利用できるかは，現時点では不明である．

そこで，このような状態であっても，メッセージを複数箇所に保存したり，複数のコミュニケーションツールを利用してメッセージを届ける等の方法によって，より確実に受信者にメッセージを届ける仕組みについて論じる．

以下，本論文では，第 2 章において非常時のメッセージ伝達方式について述べる．第 3 章では転送メールゲートウェイ方式について [1]，第 4 章ではネットニュースシステムを利用したメッセージング方式について述べ [2] ，第 5 章では複数のプロトコルを用いたメッセージの配送方式 [3] について述べ，第 6 章を結論とする．

第2章　非常時におけるメッセージの伝達方式

2.1　緒言

本章では，非常時におけるメッセージの伝達方式についての現状について述べる．

2.2　既存のサービス

2.2.1　電話を利用したサービス

NTT 西日本および NTT 東日本が，災害時に電話を利用して相手にメッセージを伝達するサービスを提供している．もっとも歴史のあるサービスである．災害時に被災地にいる人への安否をうかがう通話が急増するために，通話が輻輳し非常につながりにくくなる．このような場合に，伝言サービスが提供されている．

使用方法は，次の通りである．使用者は，指定された電話番号に電話をかけ、音声ガイダンスに従い音声を録音する．再生する際には，録音された電話番号を指定して，音声を聞く．1 回に録音できる時間や，1 つの電話番号あたりの伝言数の上限が決められている．

録音された音声とテキストを相互変換し，電話で登録した情報を Web インタフェースから確認したり，逆に Web インタフェースで登録した情報を電話で再生することも可能となっている．

このように，操作が非常に簡便な上，電話だけでなくインターネットからも登録および確認を行うことができるため，以前より災害時に多くの人に利用されてきた．しかし，音声を再生する人（あるいは，メッセージを閲覧する人）を限定できない上，メッセージを伝えたい相手に録音（登録）したメッセージがあることを伝えることができず，また，再生（閲覧）されたかどうかを確認することもできない．

2.2.2　メールと Web インタフェースを利用したサービス

非常時にメッセージを伝達するシステムは，メールと Web インタフェースのいずれかあるいは両方を利用したものが多い．受信者に向けて一斉にメールが送信され，受け取った際にメール本文の記述に従いメールで返信する，あるいは，メール本文内の URL をクリックし Web インタフェースを使って入力するものが一般的である．

各社から多くのサービスが提供されており，一斉メールの送信が管理者の操作または指示により実施されるものや，指定された地域で緊急地震速報が発報された場合に自動配信されるものがある．一斉にメールを送信した際に，受信者から何らかの応答があるまで何度も繰り返し送信したり，送信したメールがエラーで返送されてきた場合，誰宛のメールが届かなかったのかを把握する機能がついているものもある．

2.2.3　無線 LAN を利用したサービス

2011 年に発生した東日本大震災で，公衆無線 LAN サービス業者による独自の取り組みとして，公衆無線 LAN サービスを無料開放し，自社のユーザ以外にも提

供した．非常に有効な取り組みであったと評価されている．無線 LAN ビジネス推進連絡会は，2013 年 6 月に総務省が発表した「無線 LAN ビジネスガイドライン」[6] の提言を受け，携帯インフラが広範囲に被害を受け携帯電話やスマートフォンが利用できない状態が長時間継続する恐れがある場合に，無線 LAN アクセスポイントから災害用統一 SSID として「00000JAPAN」を無料開放することと決定した [7]．これにより，インターネットの利用が困難な場合であっても，00000JAPAN に接続することができれば，それ以降は平常時と同じ操作方法でインターネットを利用することができる．利用方法は，端末の Wi-Fi を ON にし，受信可能な SSID の一覧から 00000JAPAN を選択するだけでよく，非常時の利用であるためパスワード等は不要となっている．

簡便な操作で便利ではあるが，無線 LAN アクセスポイントから 00000JAPAN の SSID が送出されていても，アクセスポイントの上位回線が切断している場合は通信できない．また，SSID が同じで事業者が異なる状況で，アクセスポイント間を移動した場合，端末は移動前に割り当てられた IP アドレスを移動後も一定時間使い続けるため，新しい IP アドレスが割り当てられるまで通信ができなくなるという不具合もある．これらの事象は，ユーザにとって状況や原因が非常にわかりにくいため，利用できないと認識してしまう可能性も考えられる．

2.3 非常時に組織の構成員に向けてメッセージを伝達する方法

災害時などの非常時に，メッセージを送信する方法は上述したようにいくつか存在し，また研究も進められている．非常時に組織の構成員に向けてメッセージを伝達するために，安否確認システムが利用されている．安否確認システムの多

くは，電子メールで送られてきた Web ページにアクセスして，自身の居場所や健康状態を入力するものである．非常時に安否確認システムをスムーズに利用できるために，安否確認システムを利用する訓練を実施している組織もある．

組織において，非常時にも構成員にメッセージを確実に送るためには，メッセージを複数箇所に分散して保存したり，電子メールだけに頼らずに他の手段も利用できることが重要である．大学の場合，構成員である学生は，大学から発行されるメールアドレスを授業の課題提出時や履修登録など学生自身が必要な場合のみ使用し，日常的に利用していないケースも多いため，日常的に連絡を取れる状態にしておく必要がある．

大学で非常時にも構成員にメッセージを確実に送るためには，

- 構成員が平常時から電子メールを受信できる状態であること
- メッセージを広域な拠点で保存するといった冗長化対策ができること
- 電子メールだけでなく，Twitter などの他の配送手段も利用すること

が必要であると考えられる．

2.4　結語

本章では，これまでに提案されてきたサービスと研究の動向について概観した．次章以降では，組織の構成員に向けてメッセージを送信するために，本論文で提案する方式について述べる．

第3章　誤った転送設定によるエラーメールを削減する転送メールゲートウェイ方式

.

3.1　はじめに

インターネット上において，電子メールは手軽で最も利用されているコミュニケーションツールである．近年，企業や個人で利用できるプロバイダのメールサービスに加えて，無料で使用可能なフリーメールサービスや，携帯電話でも電子メールの送受信が可能となり，所有している複数のメールアドレス間でメールの自動転送 (以下 “転送” という) も頻繁に行われている．

しかし，フリーメールや携帯電話のメールアドレスは所有者が安易に変更する傾向にあるため，メールアドレスを変更した際に，転送先に指定したアドレスの変更を忘れがちである．この場合，メールが正しく転送されなかったという通知は，メールの送信者にのみエラーメールとして知らされるが，転送を設定した本人には何も通知がないために，転送エラーが発生していることに気がつきにくいという問題がある．

さらに，エラーメールには，何というアドレスにメールが届かなかったかは書か

れているが，メールの受信者が他のメールアドレスに転送設定をしていれば，転送先のアドレスがエラーとなっていても、それが宛先メールアドレスに指定したどのアドレスの転送先なのかは書かれていないという問題がある．このため宛先メールアドレスが複数ある場合やメーリングリストの場合は，エラーメールの受信者が，転送設定を変更すべき本人に指摘することが困難になるという構造もある．

本章では誤った転送設定によりエラーメールを抑制するシステムを提案する．メールサーバがメールを転送する際に，転送したメールがエラーとなった場合に返送されてくるエラーメールの受信者アドレスを，メールの転送を行っているサーバが受信できる形に書き換えることができれば，転送エラーが発生した際にエラーメールを受信してエラー状態を把握することができる．これにより，転送エラーが発生した際にはそれを検知してそれ以降の転送を抑制するなどの柔軟な制御が可能となる．

3.2　電子メール転送の問題点

3.2.1　電子メール送信時の手順

電子メールを送信するときの手順は SMTP (Simple Mail Transfer Protocol，簡易メール転送プロトコル) で定められており，RFC5381[9] として定義されている．メールを送信する場合の送信側がクライアント，送られてきたメールを受ける側がサーバとなる．

図 3.1 に，1 通のメールを送信する際にサーバとクライアントの間でやり取りされる SMTP コマンドと応答コードの例をあげる．図 3.1 中の MAIL FROM (以下 “エンベロープ From” という) に続く “ ” 内に送信元のメールアドレスが設定され，RCPT TO (以下 “エンベロープ To” という) に続く “ ” 内に宛先メールアド

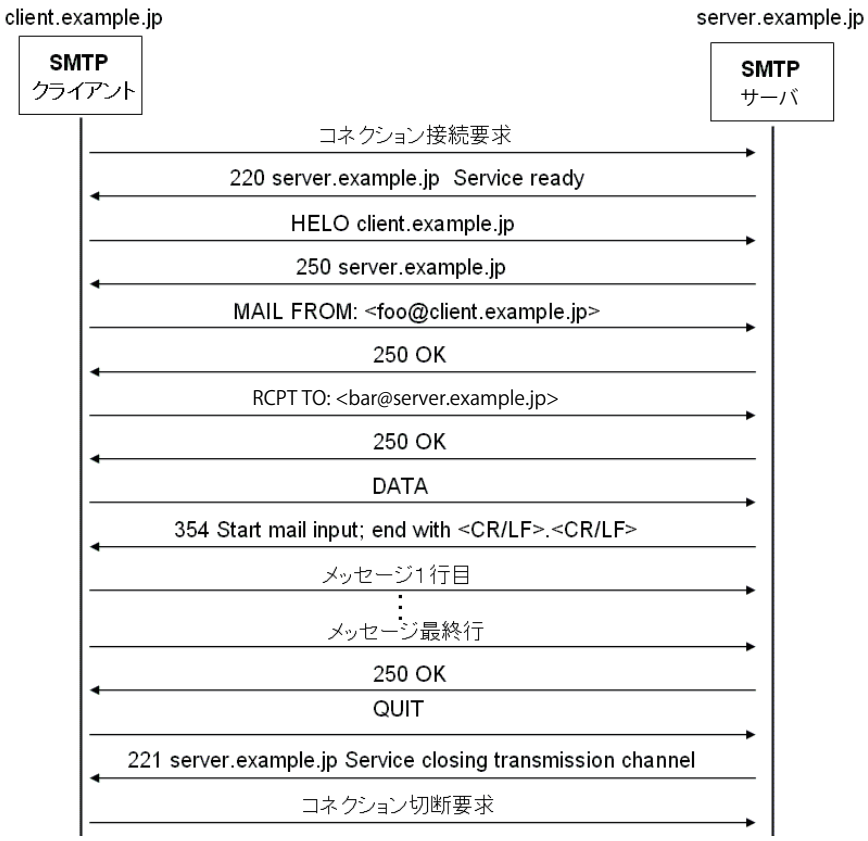

図 3.1: SMTP コマンドと応答コードによるメール送信手順

レスが設定される．SMTP サーバに存在しないメールアドレスがエンベロープ To に指定されたメールが届いた場合，エンベロープ From に指定されたアドレス宛にメールが届かなかったことを示すエラーメールが配送される．

送受信するメールはテキストファイルとして扱われ，RFC5382[10] に規定されている．メールアドレスの形式は `xxx@yyy.example.jp` となっており，RFC5382 では `@` の左側 (`xxx`) をローカルパート，`@` の右側 (`yyy.example.jp`) をドメインとよんでいる．ドメインは受信側のメールサーバを特定する名前が指定され，ローカルパートはメールボックスの名前が指定される．

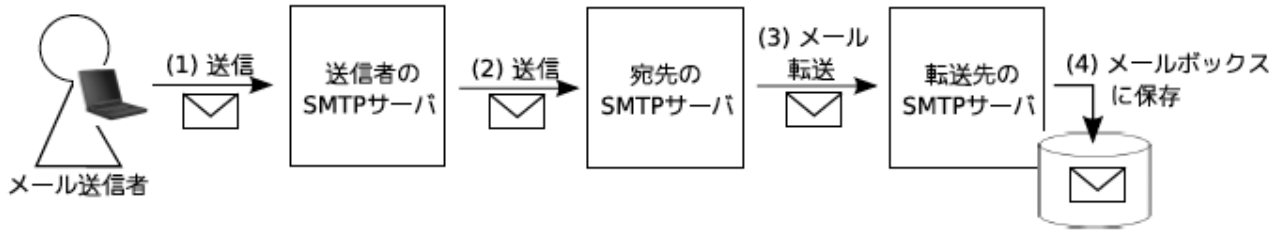

図 3.2: メール転送先アドレスが有効な場合

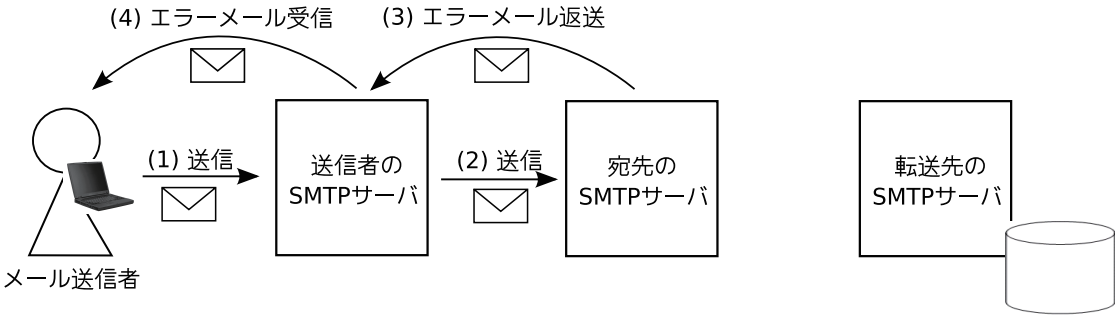

図 3.3: メール転送先アドレスが無効な場合 (転送先メールアドレスのドメインの誤り)

3.2.2　メール転送時の問題点

図 3.2 は転送先メールアドレスが有効 (メールを受信できる状態) である場合のメール転送の流れを示したものであり，図 3.3 と図 3.4 はメール転送先アドレスが無効 (メールを受信できない状態) である場合のメール転送の流れを示したものである．

図3.2では，宛先の SMTP サーバにメールアドレスを持つ人が，転送先の SMTP サーバにもメールアドレスを持っており，宛先の SMTP サーバ宛に届いたメール

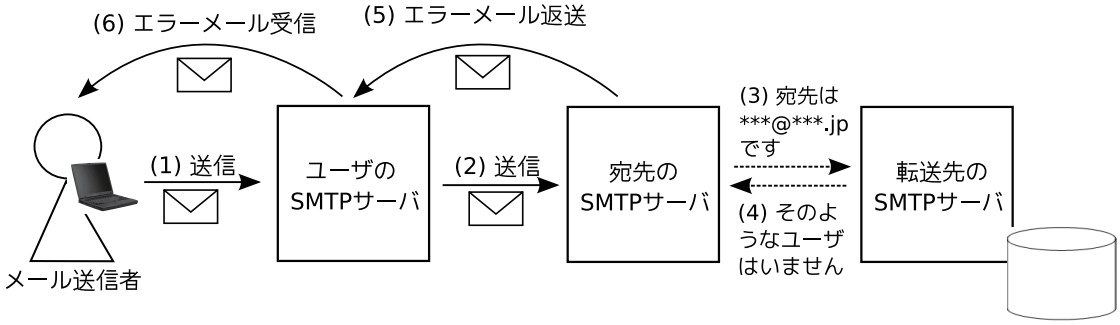

図 3.4: メール転送先アドレスが無効な場合 (転送先メールアドレスのローカルパートの誤り)

を転送先の SMTP サーバに自動的に転送されるように，宛先の SMTP サーバ内に自動転送の設定をしている．メール送信者は送信者の SMTP サーバにメールを送り (図 3.2 の (1))，送信者の SMTP サーバは宛先の SMTP サーバにメールを送信する (図 3.2 の (2))．宛先の SMTP メールサーバ内では受信者のメールを自動転送する設定となっているためこれに従い転送先の SMTP サーバにメールを転送する (図 3.2 の (3)).

図 3.3 では，メール送信者が送信したメールが宛先の SMTP サーバに到達するまでは図 3.2 と同じであるが，転送先メールアドレスのドメインが間違っているために宛先の SMTP サーバから転送先の SMTP サーバに配送することができなくなる．そこで配送されなかったことを示すエラーメールを元のメール送信者に送るために送信者の SMTP サーバにエラーメールを返送し (図 3.3 の (3))，元のメール送信者はエラーメールを受信する (図 3.3 の (4)).

図 3.4 では，メール送信者が送信したメールが宛先の SMTP サーバに到達するまでは図 3.2，図 3.3 と同じである．図 3.3 では転送先メールアドレスのドメインが間違っているために転送先の SMTP サーバに配送することができなかったが，図

3.4では，転送先メールアドレスのドメインは正しいので，宛先のSMTPサーバから転送先のSMTPサーバにメールの配送を試みた(図3.4の(3))．転送先のSMTPサーバでは，転送先メールアドレスのローカルパートに対応するユーザがいないため受信できないというメッセージを宛先SMTPサーバに返した(図3.4の(4))．転送先のSMTPサーバにメールを配送できなかったことを示すエラーメールを元のメール送信者に送るために送信者のSMTPサーバにエラーメールを返送し(図3.4の(5))，元のメール送信者はエラーメールを受信する(図3.4の(6))．転送先のSMTPサーバの設定によっては，転送先メールアドレスのローカルパートに対応するユーザが存在しなくても，一旦メールを受信した後，転送先SMTPサーバからユーザのSMTPサーバに向けてエラーメールを返送するケースもある．

図3.3の(3)および図3.4の(5)で，宛先のSMTPサーバから送信者のSMTPサーバにエラーメールを送信している．転送先メールアドレスのローカルパートが誤っている場合のエラーメールの内容は，“User Unknown：**転送先のアドレス**”となる．一方，転送先メールアドレスのドメインが誤っている場合のエラーメールの内容は，“Host Unknown：**転送先のアドレス**”となる．いずれの場合も，“**転送先アドレス**”はメール送信者が宛先に指定したメールアドレスではないため，見覚えのない宛先にエラーと思われる現象が発生しているというメールを受信することとなる．さらに，宛先に複数のメールアドレスを指定した場合のエラーメールは，誰宛のメールがエラーであるかの判断が極めて困難である．また，転送設定をした本人には，転送先メールアドレスがメールを受信できない状態にあることについて何ら通知されないので，エラーメールが送信者に返送されていることに気がつきにくくなっている．

以上の問題点をまとめると次のようになる．

- [P1] **元のメールの送信者**は，エラーメールを受信するのでメールが届かな

かったことはわかるが，エラーメールに書かれているメールアドレスは，宛先に指定したものとは異なるので，誰あてのもとに届かなかったのかの判断がつきにくい

- [**P2**] **転送設定者**は，転送先メールアドレスが無効となっているという通知が来ないので，転送先メールアドレスでメールが受信できていないことに気がつきにくく，放置してしまう

- [**P3**] **転送サーバの管理者**は，メール送信者が宛先メールアドレスの入力を間違ったのか，転送先メールアドレスが無効となったのかの判断がつきにくい

3.3 要求要件

3.2.2 節であげた問題点を解決するためのシステムの要求要件は次の 3 つである．

- [**R1**] 誰のどのアドレスに転送できなかったのかをエラーメールを受信したサーバで把握できること：

  どの転送設定者のどの転送先に転送できなかったのかを把握することで，連続して同じ転送先アドレスに転送できなかった場合には直接本人に連絡したり，また，継続して転送できていないアドレス宛には転送しないようにする等の柔軟な対応ができるようにする．

- [**R2**] 既存メールシステムを大幅に改変せず実現でき，本システム導入および取り外しの手順が容易であること：

  メールシステムの転送部分のみ置き換えを行う．また，導入の手順も容易であることが望ましい．

- [**R3**] ユーザ単位の導入が可能であること：

  本システムを利用するかどうかを，ユーザ全員を１つの単位とするのではなく，メールの転送設定を行っているユーザごとに決定できれば，サーバの負荷やメールの流量に配慮しながら導入することができる．また，本システムへの柔軟な切り替えが実現できる．

3.4　提案システムの設計

3.4.1　提案システムのキーとなるアイデア

エラーメールを受信したメールサーバが,「誰のどの転送先メールアドレスが受信できない状態となっているのか」を，返送されてきたエラーメールの内容から把握する必要がある．そこで，次の２つのアイデアに基づいて本システムを設計した．

1. メールを転送するたびに，各ユーザのメールの転送先ごとに異なるメールアドレスを生成して，メール転送時のエンベロープ From アドレスとする
2. 転送したメールに対するエラーメールをすべてエラーメール受信用に準備したユーザのメールボックスに保存されるようにし，エラーメールを受信するサーバで，誰のどの転送先がエラーとなっているのかというエラー状態を管理する

具体的な方法については，3.5.1 節で述べる．

3.4.2 システムの概要

本システムは従来のメールシステムを置き換えるものではなく，本システムを利用するユーザの転送先メールアドレス設定ファイルを後述の転送メール送信部の処理を行うプログラムに置き換える方法をとっている．これにより要求要件 [R2] および [R3] を満たすことができる．メール中継プログラム (postfix や sendmail 等) とは独立して動作するため，メール中継プログラムとして何が動作していても導入することができる．

システムの概要を図 3.5 に示す．本システムは，転送メール送信部，エラーメール受信部，転送アドレス管理部，メールキュー削除部，エラー通知部の 5 つから構成されている．

本システムの大まかな流れは次のとおりである．

(a) 転送設定を行っているユーザ宛にメールが届く

(b) メールを転送する際に，エンベロープ From アドレスを，メールを転送しているサーバで受信できるアドレスに変更して送出する (転送メール送信部)

(c) 転送したメールがエラーメールとして返送されてきたくれば，そのメールをエラーメール管理用のアドレスで受信し，エラーメールの宛先から誰のどの転送先がエラーとなっているのかを調べる．元の送信者にエラーが発生したことを通知する (エラーメール受信部)

(d) メール転送後一定時間経過し，エラーメールが返送されてこなかった場合は，正常に転送先メールアドレス宛に配送されたとみなす (メールキュー削除部)

(e) メールを転送してもエラーメールが返送されてくる状況が継続して発生している場合は，正常に転送できている他のアドレスがあればそちらに転送する，

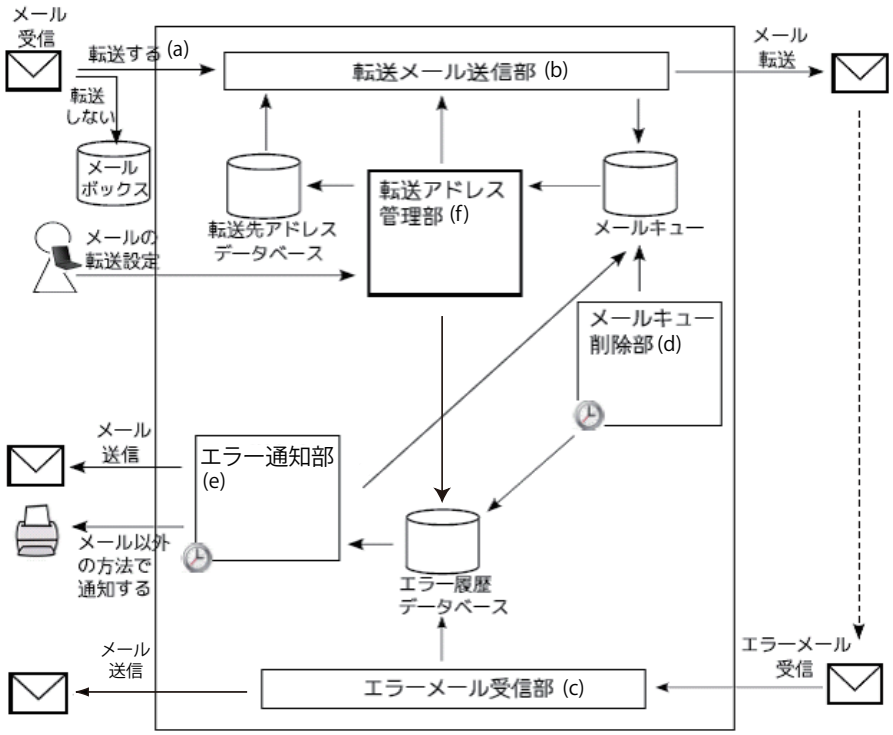

図 3.5: システム概要図

転送を一時的に中止する等の対策を施せるようにしておく (エラー通知部)

(f) エラーとなって受信できないメールアドレスを転送設定者が削除し新たなアドレスを追加した場合，エラーが発生していたことを新しい転送先アドレスに通知する (転送アドレス管理部)

以下，3.4.3 節から 3.4.7 節において，各部の動作の詳細を示す.

3.4.3　転送メール送信部

転送すべきメールが届く (このアクションがトリガーとなる)．転送先メールアドレスが設定されたファイルに，転送メール送信部のプログラムが起動するよう

に記述しておく．

転送メール送信部の処理内容は以下のとおりである．

1. エンベロープ From のアドレスを転送を行っているマシンのメールアドレスに変更し，あらかじめ設定されている転送先に転送する．
2. メールキューデータベースに，転送されてきたメールのエンベロープ From アドレス，メールヘッダと本文，転送設定者のメールアドレス，転送先メールアドレス等，メール転送に関する情報を書き込む．

3.4.4 エラーメール受信部

エラーメール受信部の処理内容は以下のとおりである．

1. 転送できなかったという通知メールが，本システムエラーメール受信用に準備したアドレス宛に届く
2. エラーメールの宛先より，誰のどの転送先アドレスがエラーとなったのかを調べる
3. エラー履歴データベースに，転送設定者のメールアドレス，エラーとなった転送先メールアドレス等の情報を書き込む
4. メール送信者にエラーの通知を行う

転送のエラーメールを，転送しているメールサーバの専用のアドレスで受信しエラー履歴データベースに記述することで，誰のどの転送先がエラーとなっているのかという状況を管理することができる．これにより要求要件 [R1] を満たすことができる．

3.4.5　メールキュー削除部

メール転送後，一定時間 (ここでは 48 時間) 以上経過したメールは転送先に届いたとみなし，メールキューデータベースの削除時刻に現在の時刻をセットする．

3.4.6　エラー通知部

継続して転送エラーが発生し続けているアドレスがあれば，転送設定者や管理者に通知する．

3.4.7　転送アドレス管理部

転送設定者がエラーが発生している転送先メールアドレスを削除した場合は，メールキューデータベースおよびエラー履歴データベースの削除時刻に現在の時刻をセットする．

3.5　提案システムの実装

転送メール送信部，エラーメール受信部，メールキュー削除部，エラー通知部，転送アドレス管理部はすべて Python で記述した．データベースとして PostgreSQL を，Python からデータベースへのアクセスのために PygreSQL ライブラリを使用した．転送メール送信部の Python スクリプトを，ユーザの転送設定ファイルから起動するために procmail[11] を利用した．

3.5.1 転送メール送信部

メールを転送する際に転送メール送信部を起動するため，procmail を利用した．procmail は，受信したメールの振り分けやフィルタリングを行うソフトウェアである．

本システムを利用する場合，利用するユーザごとに，転送先メールアドレスを記述したファイル $HOME/.forward ($HOME はユーザのホームディレクトリを表す) を procmail が起動するように "| exec /usr/bin/procmail" と記述する．$HOME/.procmailrc には procmail に関するルールを次のように記述する．

```
SHELL=/bin/sh
LOGFILE=$HOME/procmail.log
:0 c
| /usr/local/mailfwd/mail_transfer.py
```

これによって受信したメールが一通ずつスクリプト/usr/local/mailfwd/mail_transfer.py の標準入力に渡される．mail_transfer.py では，次の各処理を行う．

1. 標準入力をローカルファイルに保存する．
2. エンベロープ From アドレスを抜き出すため，メールヘッダにある Return-Path: ではじまる行に続くメールアドレスを抽出する．
3. 受信したメールを，受信メールデータベース (表 3.1) に insert する．
4. エンベロープ From アドレスを転送先アドレスごとに異なるアドレスに書き換えた後，メールを転送する．

表 3.1: 受信メールデータベースのスキーマ

カラム名	説　　明
pkey	主キー．型は整数．insert する度に自動的に 1 ずつ増える
受信したメールのエンベロープ From	受信したメールのエンベロープ From アドレス
メールキューへのパス	転送するメールを保存したファイルのフルパス名
作成時刻	データベースに insert した時刻

5. 転送した情報を転送データベース (表 3.2) に書き込む

ここで，転送する際のエンベロープ From アドレスは次の形式とする．

```
forward-a-b-c@fwd.dsm.cis.kit.jp
```

エンベロープ From のローカルパート forward-a-b-c のうち，a はメールキューデータベース (表 3.1) の pkey の値をセットする．b は転送先アドレスごとに 1 から順に異なる整数値を割り当てていく．c は転送設定者が設定した転送先アドレスの合計数をセットする．例えば pkey が 10 で，転送先アドレスを 3 個設定しているユーザの場合 3 通のメールを転送するが，2 番目に転送するメールのローカルパートは forward-10-2-3 となる．

エンベロープ From のドメイン fwd.dsm.cis.kit.jp は，本システムのために新しく作成したドメインである．宛先メールアドレスのドメインが fwd.dsm.cis.kit.jp となっているメールは，すべて本システムのエラーメール受信用に作成したユーザに届くよう設定する．

表 3.2: メールキューデータベースのスキーマ

カラム名	説　　明
pkey	主キー．型は整数．insert する度に自動的に 1 ずつ増える
受信メールデータベースの pkey	対応する受信メールデータベースの pkey
新エンベロープ From	新しく生成したエンベロープ From アドレス
転送者ユーザ名	転送設定者のユーザ名
作成時刻	データベースに insert した時刻
更新時刻	データベースに update した時刻
削除時刻	転送が成功したと判断された時刻

3.5.2 エラーメール受信部

エラーメール受信部は，一定時間ごとに起動するよう設定しておく．

すべての転送エラーメールを１箇所に集めて転送エラーの発生状況を管理するために，転送エラーメールは本システムのために準備した管理用アドレス宛に届くように設定した．また，受信したメールは１通ずつ取り出して処理しやすいように，１通ごとに個別のファイルに保存されるよう設定した．

エラーメール受信は，新着エラーメールを１通ずつ取り出し，次の処理を行う．

1. エンベロープ To アドレスを抽出し，メールキューデータベースから，受信したエラーメールが誰のどの転送先からエラーメールとして返送されてきたのかを調べる．ここで，エンベロープ To アドレスは，メールヘッダの Received: 行の for に続くメールアドレスを抽出することによって得られる．

2. エラー履歴データベース (表 3.3) にデータを追加する．

表 3.3: エラー履歴データベースのスキーマ

カラム名	説　　明
pkey	主キー．型は整数．insert する度に自動的に 1 ずつ増える
エンベロープ From	受信したエラーメールのエンベロープ From アドレス
エラーメールのファイルパス	エラーメールが保存されているファイルのフルパス名
作成時刻	データベースに insert した時刻
更新時刻	データベースに update した時刻
削除時刻	転送が成功したと判断された時刻

3.5.3　メールキュー削除部

メールキュー削除部は，一定時間ごとに起動される．ここでは，48 時間以内にエラーメールが返送されてこなかったメールは，転送先で正常に受信されたとみなし，次の処理を行う．

1. メールキューデータベースから，転送後 48 時間以上が経過し，かつ，削除時刻に何も設定されていないメールを抜きだす
2. 抜き出したメールがエラー履歴データベースに存在しなければ，メールキューデータベースの削除時刻に現在の時刻をセットする．

3.5.4　エラー通知部

エラー通知部は，一定時間ごとに起動される．継続して転送エラーが発生しているアドレスを発見するために次の処理を行う．

1. メールキューデータベースやエラー履歴データベースより，継続してエラーメールが発生しているアドレスを抜き出す

2. 抜き出したアドレスから，

   - 転送設定者のローカルのメールボックス，および，転送に成功しているアドレスにエラーが発生していることをメールで通知する
   - 管理者にメールで通知するとともに，ログファイルに記録する

3.5.5 転送アドレス管理部

転送アドレス管理部は，一定時間ごとに起動されるスクリプトである．次の処理を行う．

1. ユーザの転送先アドレスデータベースの更新日付を調べる

2. 更新された転送先アドレスデータベース内に，継続して転送エラーが発生しているアドレスが存在しなければ，転送エラーを発生させているアドレスが転送先アドレスから削除されたと判断し，メールキューデータベースおよびエラー履歴データベースの削除時刻に現在の時刻を設定して更新する．

3.6 評価と考察

エンベロープ From アドレスを変更してメールを送信 (転送) する手法は以前より存在した．既存技術との比較について述べる．また，本システムを導入することで発生するメール転送時の遅延について実験を行った．さらに，導入の容易さ，転送エラーの通知について考察した．

3.6.1　既存技術との比較

受信したメールのエンベロープ From を変更して再度送信する技術は既に存在し，Postfix の VERP(Variable Envelope Return Path)，sendmail の userdb がそれに相当する．

VERP は，エンベロープ From アドレスのローカルパートに受信者のメールアドレスを一部変更した文字列を自動的に挿入してメールを送信している．これにより，エラーメールの宛先から，誰宛のメールがエラーメールとなり返送されてきたかを把握することができる．例えば，送信者メールアドレスが sender@example.jp で受信者メールアドレスが receiver@example.com の場合，エンベロープ From アドレスは，sender+receiver=example.com@example.jp となる (ローカルパートの + と - は設定により変更可能)．

sendmail の userdb を利用しても，エンベロープ From アドレスを変更することができる．例えば userdb に

```
user:mailname   user-fwd@example.jp
user-fwd:maildrop receiver
```

このように設定しておくと，userdb を定義したメールサーバからユーザ名 user で送信するメールのエンベロープ From アドレスはすべて user-fwd@example.jp に変更される．

Postfix の VERP，sendmail の userdb とも，ユーザごとにエンベロープ From アドレスを変更することができるので，転送したメールがエラーメールとして戻ってきた場合，エラーメールの宛先メールアドレスから「誰」の転送先にメールががエラーになっているのかを把握できる．しかしここでは，ユーザの転送先ごとに

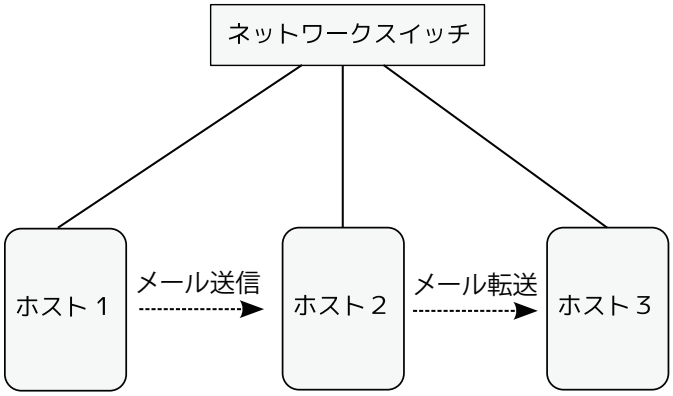

図 3.6: 転送時間を測定した環境

エンベロープ From アドレスを変更し,「誰」に加えて「どの転送先」がエラーとなっているかを把握したいので，既存のこれらの方法は利用することができない.

3.6.2 転送遅延時間の測定

本システムを利用した場合のメールの転送時間を測定した．比較のために.forward を使った従来の転送方法についても測定を行った．テスト環境は，図 3.6 に示すとおり，ネットワークスイッチの配下に３台の PC (以下ホスト１，ホスト２，ホスト３という) を接続し，ホスト１からホスト２宛にメールを送信し，ホスト２からホスト３に自動転送した．テストに使用したマシンの動作環境を表 3.4 に示す.

テスト環境のもとで，テスト 1，テスト２の２つのテストを行った．それぞれ，.forward を使用して転送した場合と，本システムを使用して転送した場合の２つのケースに分けて転送遅延時間を測定した.

表 3.4: テストマシンの動作環境

	ホスト１	ホスト２	ホスト３
機種	Panasonic Let's note	HP ProLiant ML115	NEC Mate MA13T
CPU	Pentium M 1.30GHz	AMD Athlon64 3.5GHz	PentiumIII 1.3GHz
メモリ	1024MB	512MB	512MB
OS	Ubuntu 9.04	CentOS 5.3	Vine 4.2
NIC	100Mbps	100Mbps	100Mbps

表 3.5: テスト１：1000 通のメールを受信して .forward で転送

	ホスト１からホスト２へ送信したメール数	ホスト２からホスト３へ転送したメール数	ホスト２上のメールキューのファイル数
1 分後	37	14	0
2 分後	484	431	64
3 分後	479	99	912
4-8 分後			912 (*1)
9 分後		434	124
10 分後		22	0
合計	1000	1000	

(*1) ホスト２の CPU ロードアベレージが 12-28 となりメール受信を拒否

テスト１：１つの転送先アドレスを持つメールアドレス宛に 1000 通のメールを送信

ホスト 2 からホスト 3 の１つのアドレス宛に自動転送を行った上で，ホスト１からホスト２に本文なしの 1000 通のメールを連続して送信した．このとき，ホスト２で受信したメール，ホスト３宛に転送したメール，ホスト２でのメールキュー (/var/spool/mqueue 配下) のファイル数について測定した．.forward を使用した場合の結果が表 3.5，本システムを使用した場合の結果が表 3.6 である．

表 3.6: テスト１：1000 通のメールを受信して本システムを使用して転送

	ホスト１からホスト２へ送信したメール数	ホスト２からホスト３へ転送したメール数	ホスト２上のメールキューのファイル数
1 分後	16	8	0
2 分後	358	340	0
3 分後	626	154	996
4-10 分後			996 (*2)
11 分後		385	0
12 分後		113	0
合計	1000	1000	

(*2) ホスト２の CPU ロードアベレージが 14-27 となりメールの受信を拒否

表 3.5 と表 3.6 を比較すると，転送を開始してからのホスト１からのホスト２へのメール受信数はほぼ同じである．ホスト３への転送メールは，いずれもホスト３への転送をスタートして 3 分ほど経過し 500 通ほどのメールを転送したところで，ホスト 2 が過負荷のため一時的に受信拒否された．その後 .forward を使用した方は 4 分経過して，本システムを使用した方は 6 分経過して，送信を再開した．送信再開に要した時間が 4 分と 6 分で２分の差があるが，その点をのぞけば両者はほぼ同じ動きをしていることがわかる．

テスト２：1000 の転送先アドレスを持つメールアドレス宛に 1 通のメールを送信

ホスト 2 からホスト 3 の 1000 のアドレス宛に自動転送を行った上で，ホスト１からホスト２に本文なしの 1 通のメールを送信した．このとき，ホスト 2 で受信したメール，ホスト３宛に転送したメール，ホスト２でのメールキュー (/var/spool/mqueue 配下) のファイル数ついて測定したものが表 3.7 である．

表 3.7 より，ホスト 2 からホスト 3 に 1000 通のメールが転送された際の転送に

表 3.7: テスト２：1000 通のメールを受信して本システムを使用して転送

	.forward を使用して転送			本システムを使用して転送		
	ホスト１から ホスト２へ 送信した メール数	ホスト２から ホスト３へ 転送した メール数	ホスト２上の メールキュー のファイル数	ホスト１から ホスト２へ 送信した メール数	ホスト２から ホスト３へ 転送した メール数	ホスト２上の メールキュー のファイル数
1 分後	1	1000	0	1	51	2
2 分後					229	2
3 分後					233	2
4 分後					229	2
5 分後					228	2
6 分後					30	0
合計	1	1000		1	1000	

かかる時間は，.forward を使用した場合は１分以内に送信を終えているのに対して，本システムを使用した場合は，１分間に 200 余りのメールを転送し続け，約 6 分後に転送を終えている．

このテストでは，1000 の転送先アドレスは異なるがすべて同じホスト３上のアドレスであったため，.forward を使ってメールを転送する場合は 1000 回転送を行うのではなく，同一ホスト宛のメールはエンベロープ From に複数のメールアドレスを指定してまとめて転送している．一方，本システムを使用した場合は，転送先アドレスが同じホスト上のアドレスであってもエンベロープ From を転送先アドレスごとに異なるものを生成しているため，まとめて転送せず転送先アドレスごとに１通ずつ転送している．このことが両者の転送にかかる時間差の１つの原因となっている．

表 3.7 より，本システムを利用してホスト 2 からホスト 3 に 1000 通のメールを転送した場合，１分間に 228-233 通のメールを転送している．つまり，1 通のメールの転送に要する時間は 0.26 秒となる．メールサイズが大きくなると転送にかかる

時間も長くなると予想されるが，実運用に耐えられるのではないかと考えられる．

3.6.3 従来のメール転送時の問題点

3.2.2 節において，[P1][P2][P3] の 3 つの問題点をあげた．継続して転送エラーが発生しているアドレスがあれば，転送に成功している他のアドレスに通知する等の方法をとっているので，[P2] は解決できている．転送エラーの状況 (発生回数や頻度等) は，エラー履歴データベースで管理することができるので，[P3] は解決できている．

[P1] は，転送エラーが発生しエラーメールを受信した元の送信者は，宛先に指定したアドレスとは異なるアドレスがエラーだったという通知を受理してしまうという問題点をあげた．これについては現状通り，つまり転送先がエラーであるという内容をエラーメール内に記述して返送することとし，[P1] の問題点については今後の課題とした．転送エラーが発生した場合，本システムが元の送信者にエラーメールを送っているので，エラーメールの内容を転送先のアドレスが無効ではなく，転送設定者のアドレスを明記し転送先アドレスが無効であるという意味のものに変更することは可能であるが，これにより転送設定者のメールアドレスを元の送信者が知ることとなり，これによる不都合があるのではないかと考えたからである．

3.6.4 本システム導入の容易さ

本システム導入の手順は，

1. 利用するユーザの転送先アドレス設定ファイルの内容を指定されたファイル名で保存

2. 転送先アドレス設定ファイルには，procmail が起動するよう記述

3. procmail のルールに転送メール送信部のスクリプトが起動するよう記述

の３つである．簡単なコマンドや Web インタフェースを準備すればユーザ自身の操作により，本システムへの移行や取りやめを行うことができる．当然ながら，管理者自身の操作によっても可能である．導入および取り外しは簡単に行えると考える．

3.6.5　転送エラーの通知

転送エラーメールが発生しないようにするために，転送設定者が誤った転送先アドレスを削除しなければならない．削除を促すために，転送設定者に転送エラーが発生していることを通知する．削除を促す通知を複数の手段で実行すると効果的である．具体的には，転送に成功しているアドレス宛に通知する，ローカルのメールボックスに保存する，頻繁にログインする Web サイトに表示する等が考えられる．

3.7　まとめ

本章では，誤った転送設定によりエラーメールが発生していることを転送設定者に通知することを目的として，メール転送時にメールのエンベロープ From の書き換えを行い，エラーメールの返送先を制御することが可能となった．これにより，エラー状態を転送を行ったメールサーバで把握し，継続してエラーが発生している場合には転送設定者に通知できるシステムを実装し，評価を行った．本システムを導入することで，元メール送信者にはメールの宛先や Cc に転送設定者

のメールアドレスが指定されていれば誰宛のメールがエラーとなったのか判別可能になり，転送設定者には他に転送に成功しているアドレスやローカルのメールボックス宛に通知が可能となり，サーバ管理者にはエラー履歴データベースにエラー状態が記録されることで現状を把握することができるようになった．転送先メールアドレスが無効となった場合，転送先アドレスの設定者が即座に変更後のメールアドレスを設定すれば，転送先メールアドレスの設定内容を最新の状態に保つことが可能となる．また，本システムはユーザ単位での導入が可能なため，システムの負荷を考慮しつつ導入するユーザを拡大していくことが可能である．今後の課題としては，実際のメール環境に導入し，転送エラー発生時にどのような方法で転送設定者に通知すれば誤った転送設定の修正が迅速に行えるのか等検討していきたい．

第4章　ネットニュースシステムを利用した耐障害性の高いメッセージング方式

4.1　はじめに

メールは重要なコミュニケーションツールの１つとして私たちの仕事や生活に必要不可欠なものとなっている．メールは手軽に利用できるため，日常的な連絡や情報交換だけでなく，緊急時の連絡手段としても大いに利用されている．しかし，阪神淡路大震災や東日本大震災のような甚大な被害をもたらす災害や，ハードウェアやソフトウェアの重大な障害が発生した場合（以下ではこれらをまとめて「障害時」という），自組織のメールサーバを１箇所で運用している環境では，長時間にわたりメールを利用できなくなる可能性がある．

これを避けるために，あらかじめメールを複数箇所に自動転送しておくという方法がある．障害時に自動転送先のうちいずれか１箇所にアクセスできれば，メールの受信が可能となる．しかしこの方法では，障害発生後に送られてきたメールを自動転送することができない上，常時メールを複数箇所に自動転送しておく必要があるため，大切な情報が複数箇所に蓄積されていくこととなり，セキュリティ上好ましいとは言えない．また，利用者側で複数の転送先メールアドレスを管理しなければならない．

そこで本稿では，障害時にメールを利用できなくなるという問題を解決するために，障害時であってもメールを利用できるシステムを提案する．さらに，障害時に本システムの稼働を開始するのではなく，常に稼働させておき平常時および障害時に利用できるシステムとする．本提案では，受信したメールをネットニュースプロトコル (NetNews Transfer Protocol) を使って遠隔地にあるサーバに配送し，サーバ間でメールの同期を行う．利用者はいずれかのネットニュースサーバにアクセスできればメールを閲覧することが可能となる．

4.2　要求要件

4.2.1　想定環境

通常時は，組織の管理者がメールシステムの運用・管理を行っているが，大規模な災害が発生した場合，広範囲にわたりライフラインが途絶え，ネットワークが寸断されてしまうと，管理者の手元にある機器やまわりのネットワークが全く利用できない可能性がある．本章では，このような場合を想定し，管理者が特別な操作をすることなく利用できるメールシステムの構築を目的としている．

4.2.2　メールシステムのモデル化

メールサーバは，メールを受信する部分（以下「受信部」という）と受信したメールをスプールに保存する部分（以下「保存部」という）から構成されていると考えることができる．

メールサーバを１台で運用している場合，図 4.1(a) に示すように，受信部と保存部は一体化された状態で管理・運用されている．メールサーバの設置場所に障害

が発生するとメールサーバはメールを受信することができなくなる．メールサーバが複数台存在するがクラスタリング構成等により物理的に近接した場所に設置されている場合，メールサーバが１台の場合と同様に，設置場所に障害が発生すると以降のメールを受信することができなくなる．

これを回避するために，図 4.1(b) に示すようにメールサーバを物理的に離れた複数箇所に配置すると，複数台あるメールサーバのいずれかがメールを受信することができれば，障害時であっても利用者はメールを受信することができる．しかし，メールはいずれか１台のメールサーバの保存部に格納されるため，メールサーバによって保存部のメールが異なることになる．これでは，利用者は全てのメールサーバに接続してメールを読む必要があり，さらに，1 台のメールサーバに障害が発生するとそこに保存されていたメールを読むことができなくなる．

そこで，メールサーバの冗長性を保ちつつ，どのメールサーバに接続しても同じメールを読めるようにするため，図 4.1(c) に示すようにメールサーバの受信部と保存部を分離し，受信部を複数箇所に配置し保存部を１箇所に配置すると，どのメールサーバに接続しても保存部には同じメールが格納されており，いずれかの受信部に障害が発生しても他の受信部がメールを受信し保存部に格納することができる．しかしこの方法では保存部が単一障害点となるため，保存部に障害が発生するとメールを受信することも閲覧することもできなくなる．

この問題を解決し，4.2.1 章で示した想定環境で動作するために，図 4.1(d) に示すように受信部と保存部を分離した状態で，受信部も保存部も複数箇所に配置し，保存部間で同期を取ることとした．これにより，単一障害点をなくすことができ，利用者はどのメールサーバに接続しても同じメールを読むことができる．また，受信部および保存部に障害が発生しても，他の受信部および保存部が稼働していればメールの受信および閲覧が可能となる．本章では図 4.1(d) のモデルが望ましい

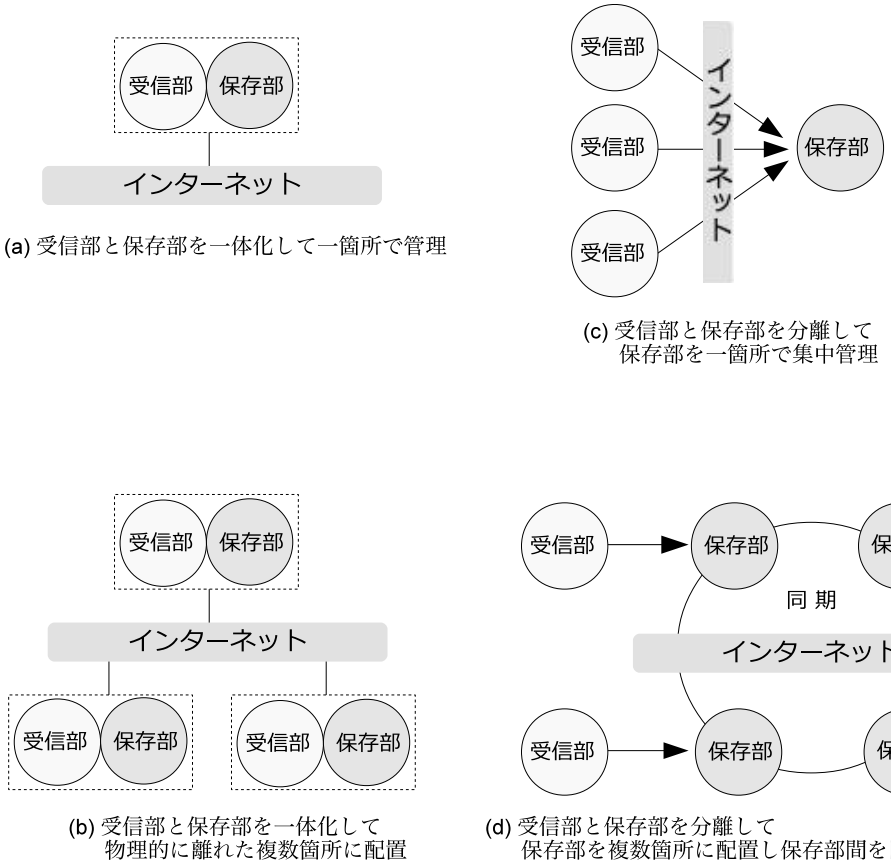

図 4.1: 従来のメールシステムおよび提案するメールシステムのメール受信部分とメールボックス (保存部) の関連図

と考えている.

4.2.3　要求要件

4.2.2 章 の図 4.1(d) で示した提案モデルをシステム化するために求められる要求事項は次のとおりである.

[要求 1] 受信したメールを保存部に格納し，複数の保存部間で同期を取ること：

受信したメールは保存部に格納する．保存部は物理的に離れた箇所に設置されているので，利用者が複数箇所にあるいずれのサーバにアクセスしても同じメールを閲覧できるようにするために，保存部間で受信したメールの同期を行う．

[要求 2] 障害により利用者が普段利用しているメールサーバにアクセスできない場合でも，自分宛のメールを閲覧できること：

障害により，利用者がメールを受信する際に利用しているメールサーバにアクセスできない場合は，代替となる別のメールサーバにアクセスして利用者宛に届いたメールを閲覧できるようにする．

[要求 3] 障害により通常受信用に指定されているメールサーバがメールを受信できない場合でも代替のメールサーバが受信できること：

障害時にメールサーバが自組織宛のメールを受信できない場合であっても，別のメールサーバが代替となってメールを受信し保存部に格納する．

[要求 4] 複数箇所に配置している保存部のメールを，利用者本人以外が内容を読めないこと：

複数箇所に配置した保存部のメールは利用者本人だけが内容を確認できるものとする．他の利用者にメールの内容を知られてはいけない．

[要求 5] 障害時だけでなく平常時も利用できること：

障害時および平常時に利用できる機能としては，指定した受信者にメッセージを送ることができること，指定された受信者宛のメッセージを当該受信者が取り出すことができることの 2 つとする．

性能としては，平常時の配送遅延時間は 4 時間以内とする．一般的には，送信したメールを受信すべきメールサーバが受信しなかった場合，4 時間経過すると警

告メールを送信者に返送する設定にしていることが多いので，配送遅延時間を 4 時間以内とする．一方，障害時の配送遅延時間とメールクライアントからメールボックスへのアクセス時間の合計値を 12 時間以内とする．人命救助をする際には災害発生から 72 時間を超えると生存が望めないと言われているが，国土交通省発行「阪神・淡路大震災の経験に学ぶ」[13] によると，災害発生から 24 時間経過すると生存率が大幅に下がることがわかる．そこでメッセージの往復にかかる時間を 24 時間以内とし，送信者が送信してから受信者が閲覧するまでにかかる時間を 24 時間の半分の 12 時間とした．

4.3　提案システムの概要

本章では，提案するシステムの概要について述べる．

4.3.1　ネットニュースの概要

4.2.2 節の図 4.1(d) のモデルを実現するために，本章では複数箇所に設置された保存部間のメールを同期するための技術としてネットニュース [14] を利用する．

図 4.2 にニュースシステムの概要図を示す．ネットニュースはインターネット初期の 1980 年後半から，メーリングリストや Web ベースの電子掲示 板や SNS (Social Networking Service) が普及する 2000 年代半ばごろまで長年にわたり有用な情報交換の場として広く長く利用されてきた．現在でも国内外で一部の情報交換用としてネットニュースが利用されている．ネットニュースは，複数のサーバにまたがった掲示板システムである．記事（メッセージ）を蓄積しながら近隣のサーバにバケツリレー式に記事のコピーを配送することで，結果的にどのサーバにも同じ記事が保存されるという仕組みで動作している．

話題ごとにニュースグループが作成され，それぞれのニュースグループは階層構造になっている．ニュースグループは、カテゴリ別に階層構造になっている．図 4.3 がニュースグループの一例である．ニュースグループ名は，comp や fj といった最上位のカテゴリから下位のカテゴリをドット "." で区切って表現する．例えば，図 3 の場合 comp.lang.c++ や fj.editor.emacs となる．comp.lang.c++は，計算機の中の言語の中の C++に関する話題に関するニュースグループになっている．記事がニュースサーバ内に保存される際には，comp.lang.c++の場合には $DIR/comp/lang/c++ の配下に 1 つの記事が 1 つのファイルとして保存される ($DIR は記事が保存されるトップディレクトリを示す).

ニュースサーバ間で記事を送受信するプロトコルは NNTP (Network News Transfer Protocol) が主流となっている．隣接するニュースサーバ間で記事の送受信を行うが，送受信するニュースサーバのホスト名，送受信するニュースグループについてあらかじめ定義ファイル内に明記しておく．また記事には記事ごとにユニークな Message-ID が付与されており，各ニュースサーバは過去に受信した記事の Message-ID を記録している．記事を受信する際には，隣接するニュースサーバから NNTP プロトコルで接続され、定義されたホスト名であるか，定義されたニュースグループであるか，過去に受信した Message-ID と同じかどうかの確認を行う．これにより，許可されていないホストやニュースグループの記事や過去に受信した記事を受信しない仕組みとなっている．記事を送信する際には，定義ファイル内に記載されたホストに定義されたニュースグループの記事を NNTP プロトコルで隣接するニュースサーバに接続して配送を行う.

送信されてきた記事が条件に合致していればニュースサーバ内に保存していくため，到着の順番も保証していない上，全ての記事が届くことも保証していないが，記事が配送されてくる上流のサーバを複数配置する等により，極力記事の欠

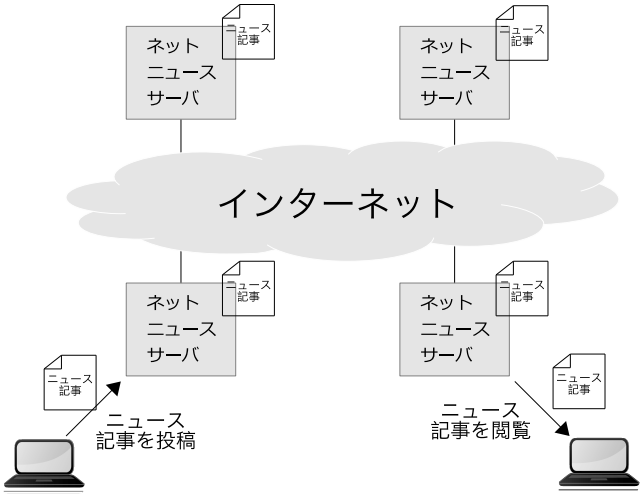

図 4.2: ニュースシステムの概要図

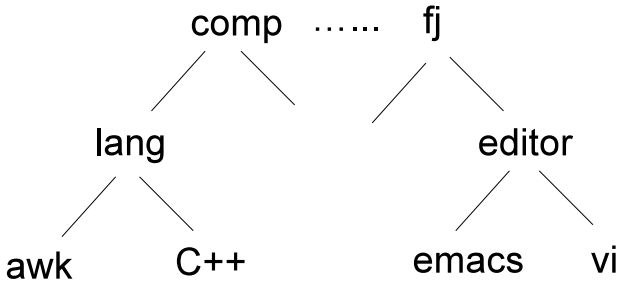

図 4.3: ニュースグループの構造

落を回避している．

4.3.2　提案システムの構成図

ネットニュースの仕組みを利用して保存部を同期するために，受信したメールをネットニュースの記事に変換し，ネットニュースシステムを使って記事（メール）を複数サーバに配送し，利用者が自分宛に届いたメールを閲覧するシステム

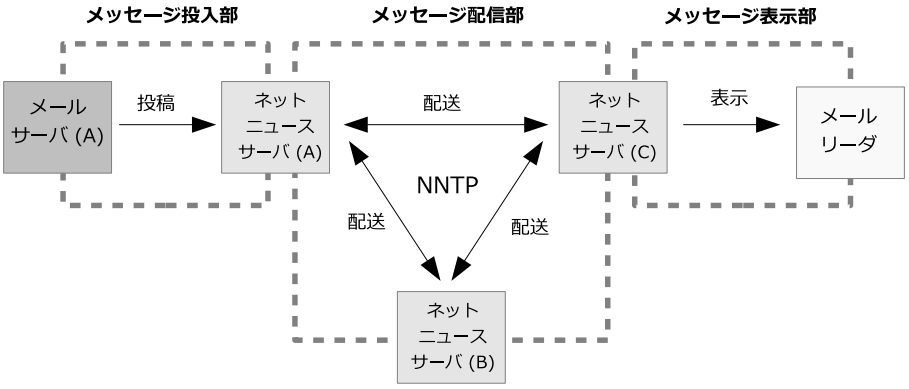

図 4.4: 提案システムの概略図（メールサーバにアクセスできる場合）

を提案する．

図 4.4 に，メールサーバが稼働している場合のシステム概要図を示す．メッセージ投入部，メッセージ配信部，メッセージ表示部から構成されている．図中のメールサーバ (A) は，利用者がメール受信用に利用しているサーバである．メッセージ投入部では，メールサーバ (A) が受信したメールをニュース記事に変換してネットニュースサーバ (A) に投稿する．メッセージ配信部では，ネットニュースサーバ (A)(B)(C) の間でニュース記事を配送する．メッセージ表示部では，利用者宛のメールを表示する．

図 4.4 のメールサーバ (A) が障害によりメールを受信できなくなった場合のシステム構成図を図 4.5 に示す．メールサーバ (A) に障害が発生した場合は，メールサーバ (A) の代替となるメールサーバ (B) がメールを受信しニュース記事に変換してネットニュースサーバ (B) に投稿する．投稿された記事は他のネットニュースサーバ (A)(C) に配送される．メールサーバ (A) 復旧後は，障害発生中にメールサーバ (B) が受信しネットニュースサーバ (A) に配送したメールをネットニュースサーバ (A) がメールサーバ (A) の利用者用メーブボックスに保存する．

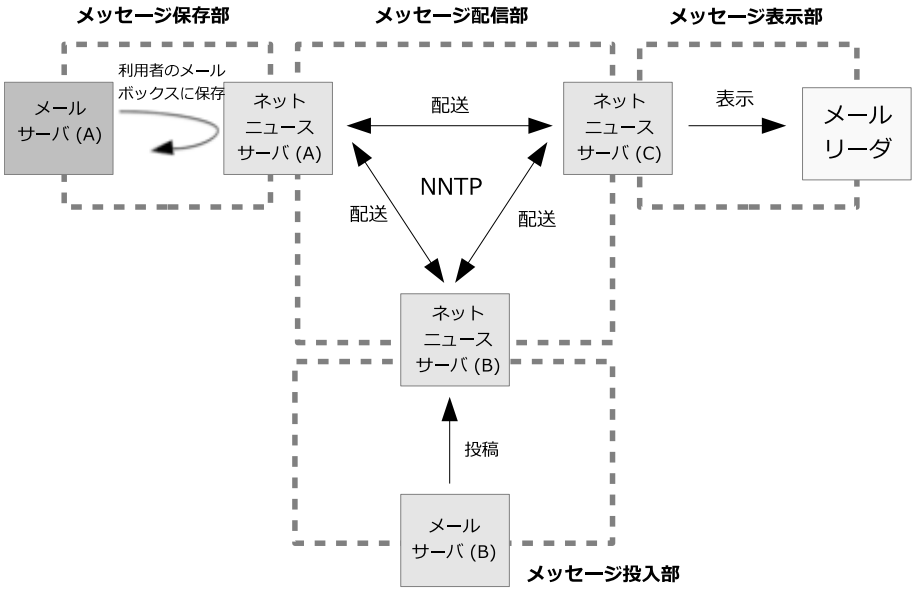

図 4.5: 提案システムの概略図（メールサーバにアクセスできない場合）

4.3.3　記事の暗号化と復号

ネットニュースの記事はアクセスした利用者が誰でも閲覧することができ，制限されていなければ誰でも投稿もできる仕組みになっている．一方，メールは私信であるため受信した本人以外は読めないようにしなければならない．そこでネットニュースサーバにアクセスした人が自分宛ではないメールを読めないようにするために，つまりネットニュースサーバの利用者から，他人が受信したメールの閲覧をできないようにするために，メールサーバに届いたメールをニュース記事として投稿する際に，受信者本人のみが解読できる方法で暗号化を行い，閲覧時に復号を行うこととする．

暗号化を行うために暗号化用鍵ファイルとペアとなる復号用鍵ファイルを作成する．暗号化用鍵ファイルはニュースグループに投稿しておく．また，復号用鍵ファイルのパスフレーズは，利用者がメールを受信する際に利用するパスワード

と同じものにしておく．

4.3.4 メールアドレスとニュースグループのマッピング

受信したメールをニュース記事に変換して投稿するために，1 つのメールアドレスに 2 つのニュースグループを割り当てる．ニュースグループの 1 つは受信したメールを保存するため，もう 1 つは暗号化用鍵ファイルを保存するためである．

メールアドレスの @ の右側（ドメイン）はピリオドで区切った階層構造になっており，右端が上位階層で主に国を表し左端が一番下位の階層となっている．メールアドレスのドメイン部分の階層構造はニュースグループと逆の記述方法になっている．そこで，メールアドレスに対応したニュースグループ名は，メールアドレスの @ をピリオドに変え，左から右に向かって階層構造が上位から下位になるように並び替えた後，右端にメール保存用のニュースグループには .mail を付加し，暗号化鍵ファイル保存用のニュースグループには .key を付加する．例えば，メールアドレスが user@subdomain.example.jp の場合のニュースグループ名は jp.example.subdomain.user.mail と jp.example.subdomain.user.key とする．ニュースグループ名は特に制限をかけていなければニュースサーバにアクセスした人の目に触れるため，ニュースグループ名の一覧を取得することでどのようなメールアドレスが存在するのか推測が可能となる．これを避けるために，メールアドレスの @ の左側の部分（ユーザパート）をハッシュ値に置き換えを行う．メールアドレスが user@subdomain.example.jp の場合のニュースグループ名は表 4.1 となる．

表 4.1: メールアドレスが user@subdomain.example.jp の場合のニュースグループ名

メール保存用ニュースグループ名
jp.example.subdomain.ee11cbb19052e40b07aac0ca060c23ee.mail
暗号化用鍵ファイル保存用ニュースグループ名
jp.example.subdomain.ee11cbb19052e40b07aac0ca060c23ee.key

4.3.5　ニュースグループのモード

ニュースグループ作成時に，ニュースグループ名を指定するとともに，ニュースグループのモードを (1) 投稿できるニュースグループ，(2) 投稿できないニュースグループ，(3) 投稿できるがモデレータ機能を持っているニュースグループの 3 種類から選択する．(3) のモデレータ機能を付加すると，投稿されたニュース記事はすぐに投稿されず一旦指定されたメールアドレス宛てに送信される．

メール保存用ニュースグループは通常利用しているメールサーバや代替メールサーバで受信したメールをすぐに投稿するために (1) の投稿できるニュースグループとして作成する．暗号化鍵用ファイル保存用ニュースグループは，暗号化用鍵ファイルを更新する際に投稿を行う．いたずら等の目的で暗号化用鍵ファイルを更新されないために，(3) の投稿できるがモデレータ機能を持っているニュースグループとして作成する．

(3) のモデレータ付きニュースグループは，ヘッダに Approved: が存在すれば指定のメールアドレス宛てにメールを送信することなく、記事が投稿されてしまう．このことを知っていれば誰でもモデレータ付きニュースグループに記事を投稿することができる．ネットニュースにこれを回避する機能はない．そこで、この部分は実装できていないが，投稿されている暗号化用鍵ファイルが正しいかどう

かを検証する方法として，暗号化用の鍵をニュースグループに投稿する際に PGP で署名を行い，鍵を利用する 際に署名を検証する仕組みを導入すれば，投稿された暗号化用の鍵が正しいものであるかどうかの判断が可能となるのではと考えている．ニュースグループへの投稿そのものを抑制することは困難であるが，大量に投稿されることを避けるためには，非常に多くのニュースグループに投稿するマルチポストや短時間に多数の記事の投稿を抑制する仕組みが別途必要となると考えられる．

4.3.6 メッセージ投入部

メッセージ投入部は，受信したメールを暗号化して宛先メールアドレスに対応したニュースグループに投稿する．処理の流れは次の通りである．

1. 新着メールが届いていないか定期的にチェックを行い，届いていれば (2)(3) を行う
2. メールの受信者の暗号化用鍵ファイルが保存されているニュースグループにアクセスし，暗号化用鍵ファイルを取り出す
3. 受信したメールを (2) で読み出した暗号化用鍵ファイルを使って暗号化し，受信者のメールアドレスに対応したニュースグループに投稿する

4.3.7 メッセージ配信部

ネットニュース配送用アプリケーションを利用して，ニュースグループに投稿された記事を他のネットニュースサーバに配送する．

4.3.8　メッセージ表示部

利用者はネットニュースサーバに接続して，次の手順でメールを閲覧する．

1. ニュースリーダを使ってニュースサーバにアクセスする
2. メールアドレスに対応したニュースグループに投稿された記事を，復号用鍵ファイルとパスワードを使って復号し閲覧する

4.3.9　メッセージ保存部

図 4.5 中のメールサーバ (A) が障害によりメールを受信できない場合は，代替としてメールサーバ (B) がメールを受信しネットニュースサーバ (B) に投稿する．投稿されたニュース記事（メール）は，メッセージ配信部によりネットニュースサーバ (A) および (C) に配送される．メールサーバ (A) が障害復旧後は，ネットニュースサーバ (A) が受信したニュース記事を利用者の復号用鍵ファイルを使って復号し，利用者のメールボックスに保存する．

4.4　提案システムの試作

提案システムのメッセージ投入部およびメッセージ保存部は Python 2.7 で記述した．各サーバの OS は CentOS 6.3，メールサーバは Dovecot 2.0.9 および Postfix 2.6.6，ネットニュースサーバは inn 2.5.4 [15]，ニュースリーダに Thunderbird 17.0.7 [16] ，Thunderbird のアドオンに Enigmail 1.5.2 ，暗号化および復号に PGP を使用した．

4.4.1 準備

管理者が (a) ネームサーバへ MX レコードを追加，(b) ニュースグループの作成および (c) 暗号化用鍵ファイル・復号用鍵ファイルの作成を行っておく．

(a) ネームサーバへ MX レコードを追加

障害時に図 4.5 中の代替となるメールサーバ (B) がメールを受信できるようにするため，ネームサーバの MX レコードにメールサーバ (A) と代替メールサーバ (B) を登録する．メールサーバ (A) の優先度を高く代替となるメールサーバ (B) の優先度を低く設定する．これにより，通常時はメールサーバ (A) がメールを受信し，障害時は代替となるメールサーバ (B) がメールを受信することができる．

(b) ニュースグループの作成

ネットニュースサーバ上に，メール保存用ニュースグループと暗号化用鍵ファイル保存用ニュースグループを表 4.1 に示す命名規則に従って，図 4.4 のメールサーバ (A) に登録されている利用者のメールアドレス分のニュースグループを作成する．１台のネットニュースサーバ上にニュースグループが作成されると，ネットニュースシステムの機能により，自動的に他のネットニュースサーバにも同じニュースグループが作成される．

(c) 暗号化用鍵ファイル・復号用鍵ファイルの作成

PGP による暗号化用鍵ファイルおよび復号用鍵ファイルは，利用者のパスワードを生成/変更するタイミングで作成する．鍵の種類は RSA ，鍵長は 2048 bit とする．暗号化用鍵ファイルは (b) で作成した専用のニュースグループに投稿しておく．投稿した暗号化用鍵ファイルを図 4.6 に示す．復号用鍵

```
Newsgroups: dsm.test
Subject: key1
Date: Sun, 10 Feb 2013 14:24:13 +0000 (UTC)
Organization: A poorly-installed InterNetNews site
Lines: 30
Message-ID: <kf8ah6$62o$2@isbs1.dsm.cis.kit.jp>
NNTP-Posting-Host: isbs1.dsm.cis.kit.jp
X-Trace: isbs1.dsm.cis.kit.jp 1360506253 6232 133.16.241.49 (10 Feb 2013 14:24:13 GMT)
X-Complaints-To: usenet@isbs1.dsm.cis.kit.jp
NNTP-Posting-Date: Sun, 10 Feb 2013 14:24:13 +0000 (UTC)
Xref: isbs1.dsm.cis.kit.jp dsm.test:6

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2.0.14 (GNU/Linux)

mQENBFCvRusBCAD4MFPDG4+bLQ35rSbRhVELKEEsxaWXm6fi7PgylafC9IFcdFUr
Jh5rRPKdHkARyUSd3tVl+AKr7K7RhQgEcHhyRRgHP321IKigPNAayi7/F8kGdNia
G1ZJl08tW3o2DlCEA5vO4VmtILMU2oVU6VYfpcdtPzM4Jl567rSp0UeGeCKGnhjk
Qg0VdcDnx5YbuM3QN8QpQbG/UhkM5VPaiqsrsa053A6MfMthcKfE5HLgoDMEFDPm
e3p5Eg7NmIv1oIWkWZwwD81fuXHDzJgJTCmTCRO/K3/agLtNjX6vRqEzVo+vJOCK
5VBGaTeOTM4RIPD8IeffuZ2yBop0tEQoHGObABEBAAG0MUlzaGliYXNoaSBZb3No
aWtvIDx5LWlzYnMwOEBpc2JzMS5kc20uY2lzLmtpdC5qcD6JATgEEwECACIFAlCv
RusCGwMGCwkIBwMCBhUIAgkKCwQWAgMBAh4BAheAAAoJEGrxIxzzq6x7cUUH/2XI
kuTPtYez8CdTtfF0jjEGmdZi3vij6mamAktRzUcYkoUwkdIrP7ZHTItNwJxBBCgS
Py94vrXn2qpABHmFh2mM5u3PZq386pCHVwVXby6CpveJI3LeNmMR2k2bEnGEy2Lt
n2u8KhSlY9uaMT7culaRfYsWBlX1FLOdYEheoTD0HwwzpfM5yMOtNY6dxDjr0iVy
BcsW6UZ/KNscv0NRWC2lckQKsxK8XUY5tBi/yV646zGwGPXFF+XIV4/3d7eVVy9r
0NldM+C1i4X03NDvdoHNa+ETHTGRkxCE9PGyLYwWWq1bH9gHztm/7ctabNFvW9xJ
eLLHa46KKyDslIXbJ3G5AQ0EUK9G6wEIAKRVgGo8Ifw+cmJxdYR73jYvzKPh6vdf
DZ8lKiQO9q/ZJceoGxbDYXNV3CmhYH0UsGYSqS3VXlO8Pkp7WhEnCuvpmPtnp9NI
JVfSxPV8OKyfVR5qWiuv9ydHc+mI6feE8bYFJujth5tkpk6BgR8E4JFD462viPTV
l0rODBGzbW+WrwYhdy0NRc4LuKD068yd/R3J56sdjkMUnm/8T4umhb7HEH2GI0Jb
TBut7exNwkuydsWC/sCqpwL/utjwJANmfIcf5tfeh3jRFi+Q+Ocujs1l/2Ip2Fcj
m/EJG0FLSC5Zvordws3darz5Kg32IoCnQeLbKxl5S821h+fekcn32mcAEQEAAYkB
HwQYAQIACQUCUK9G6wIbDAAKCRBq8SMc86usew+BB/4r1TkTmhutGSOQ4bVLzme4
kB2Rf3iF/EdlRZu6goQ6I5Dk2Lm6GzSi1yO2Cs0tZaNs6Adf36/0CWikX10Kkf0m
YezVT2+yIVxw0Gx8MmQMRZ13p+GoTPQpvs7B4Szb4ucts5dk3+VZpOTwg5VJFdgI
9cg82LYdEG1pwGTvKpskmyLveyizChYE46VPZOhDywhyvKiPQ6JuVrnpd+cI7SYC
+p2/a0NpbY2g6dyTVFyEGxd4h7HTaVcWb2HzT9EadMzKbZIcZPRnuc6ryYe40koL
6SEzcr+u02d5N7XokXBBayQKMAYzfx4fmY5PKyHi4buLe+lWK7UqTQdnwbFPK6Kq
=cqw6
-----END PGP PUBLIC KEY BLOCK-----
```

図 4.6: 投稿された暗号化用鍵ファイル

ファイルは，利用者が閲覧時に使用するだけでなく，障害時に代替メールサーバが受信したメールを自組織のメールサーバのスプールに保存する際にも使用する．自組織のメールサーバ上の復号用鍵ファイルは，利用者のパスワードを使ってパスフレーズ入力済みとして扱うものとする．

4.4.2　メッセージ投入部

メッセージ投入部では，NNRP (Network News Reader Protocol) [17] を使ってメッセージを投稿する．メールサーバに届いた１通のメールごとに次の処理を行う．

1. メールのヘッダから本文の最後まで，１通のメール全体を取出す．

```
Newsgroups: jp.ac.kit.cis.dsm.user
Subject: test mail
Date: Sun, 10 Feb 2013 15:11:46 +0000 (UTC)
Organization: A poorly-installed InterNetNews site
Lines: 21
Message-ID: <kf8dai$707$2@pc1.dsm.cis.kit.jp>
NNTP-Posting-Host: pc1.dsm.cis.kit.jp
X-Trace: pc1.dsm.cis.kit.jp 1360509106 7175 133.16.241.49 (10 Feb 2013 15:11:46 GMT)
X-Complaints-To: usenet@pc1.dsm.cis.kit.jp
NNTP-Posting-Date: Sun, 10 Feb 2013 15:11:46 +0000 (UTC)
Xref: pc1.dsm.cis.kit.jp dsm.test:7

-----BEGIN PGP MESSAGE-----
Version: GnuPG v2.0.14 (GNU/Linux)

hQEMA6Dii3S2i62KAQf/TC1dtyeWlydnsJ0PXi007IbWZga72iXICo5LweIOB6b7
pMWlgbWU0YhtBX+gRTxs23ETLPP4CjxAA0x+7Y0jUqVZ5N9ZaWurIBNr0QZVcHd3
5h8dafa+uGR+W6tMeI1tC3AmuAo1JiETpuyf+NHAbG+9jZ8e2uZZOfaftCY3+n26
HxOvSEkj0BWrlbhXGr9+Oc8Epq53Iir7iL/0I011/LjS7AIJWM/t5983ErqBf+KH
nppYn/nMt0vBPSlA7qjsSoGywwWAdTPuSL1LFsdMLDpPB6zFcUNHHTeYD9OEgLjT
hR300Krfw+RZiMnI+jNo51AtxLGLTUwL4PO4JTXK/tLA7QHoDPeKY0cJDydT6uuU
5SJv0I5sprwu3fnuIGwaF2oFLEEhZLKiM22Kz6F1uf8cuEovwXHmAZvou2+DD1fl
ct8TdRuROV+Ba2hSWIFovF/g4o4FtTuLH94LQ225R5SdxUEkKUcylWUoZMYuBXgE
UVrH7LGQXVEpQyEvrRUMSu2OAmJXkpRWe/h1Y1T7h8n/59zQvvauFLFoO5owtEdi
8J9LJww0SqFWeYPRp0y5enqa4D0V4HrjqzXPkYjMMc0dx0uivsHx7uoml3QSmvdh
0RBJBXXIKI6YNsiKNI0Zw7o84lwuXx9wJFD6y7a4yv6dVph8g0yxEc7Hj8axhftY
t/1ZarRRaSM5EUxXgQmH5GN+x1TKO/8zeb92c3of6M+yBWKqGdKl3p+w4vqADRiN
hPZL2FxcwR0YkW1BIFUnwPmWE8UIxii7K7JYpFG4wDm+FwTMD2CC9OSdotuFYDn6
65B2TE8XHtnNVbOjRAjWNBZvUkriHqxGUJUChwJlwUgt0y1uD5yx6YC8fihwBgx8
DZc/7YS11bl8MDcDvTsMA7Jr7sSVU+8F7PplpdzJlw==
=2UYh
-----END PGP MESSAGE-----
```

図 4.7: 投稿された暗号化後のメール

2. 表 4.1 に示すような受信者のメールアドレスに対応するメール保存用ニュースグループ名，暗号化用鍵ファイル保存用ニュースグループ名を決定する．ハッシュ値は md5 で算出する．

3. (2) で決定した暗号化用鍵ファイル保存用ニュースグループにアクセスして最新の記事を入手し，暗号化用鍵を取出す．

4. (3) で入手した暗号化用鍵を使って (1) のメールを暗号化する．

5. (2) で決定したメール保存用ニュースグループに (4) で作成した暗号化したメールを投稿する．

投稿した暗号化後のメールを図 4.7 に示す．

4.4.3　メッセージ配信部

inn を使ってニュース記事を配送する．他のネットニュースサーバから配送されてきたどのような記事を受信するかという設定を incoming.conf ファイルにて行う．実際には incoming.conf ファイルに接続を許可するニュースサーバのホスト名と受信するニュースグループのパターンを記載する．また newsfeeds ファイルにて，他のニュースサーバにどの記事を送信するのかという設定を行う．具体的には newsfeeds ファイルに送信を許可するニュースサーバのホスト名や送信するニュースグループのパターン、送信時に使用するアプリケーション名を指定する．incomming.conf および newsfeeds とも，ニュースグループ名にはワイルドカードでの指定が可能となっているので，表 4.1 の場合は jp.example.subdomain.* と指定することができる．ニュースグループごとに記事の有効期間を日数で設定することができる．有効期間を過ぎた記事は自動的に削除される．記事の有効期間は，ストレージの空き容量やメールシステムが復旧するおおよその日数を推測して設定する．今回は，30 日とした．この設定により，30 日を経過した記事は自動的に削除される．

4.4.4　メッセージ表示部

ニュースを閲覧するたのニュースリーダとして Thunderbird を利用する．また PGP により暗号化されたデータを復号するために，Thunderbird のプラグインである Enigmail を利用する．これらを使って，指定されたニュースサーバに接続し，自分宛のメールが保存されたニュースグループにアクセスする．ニュース記事（メール）閲覧時に復号用鍵ファイルを指定しパスフレーズを入力すると，復号されたメールが表示される．

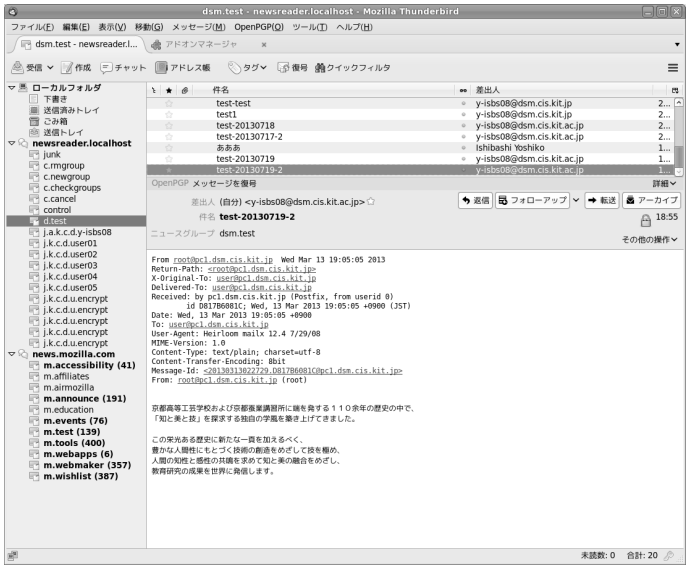

図 4.8: 復号したメール

4.4.5 メッセージ保存部

図 4.5 のネットニュースサーバ (A) にニュースログファイル投稿用のニュースグループを作成しておき，定期的にニュースログファイル `$NEWS/log/news` (`$NEWS` は inn がインストールされたディレクトリを表す) を投稿する．ニュースログファイルには，記事が投稿された日付，記事のメッセージ ID，記事の送信元 FQDN (Fully Qualified Domain Name) 名が保存されている．メッセージ ID は全ての記事でユニークになっている．

図 4.5 のメールサーバ (A) が復旧後，ニュースサーバ (A) のニュースログファイル投稿用のニュースグループから，最新のニュースログファイルを取り出す．ニュースログファイルより，記事の送信元 FQDN 名がニュースサーバ (A) ではない記事のメッセージ ID を取り出す．取り出されたメッセージ ID が，メールサーバ (A) が停止中に代替メールサーバが受信しネットニュースの記事として投稿されたものとなる．利用者のメール保存用ニュースグループ内に保存されている記事にア

クセスし，先ほど取り出したメッセージ ID と合致すれば，記事の本文を利用者の秘密鍵で復号して利用者のメールボックスに保存する．この処理を全ての利用者について行う．

4.5　関連研究

4.5.1　コンテンツ分散配置技術

受信したメールを複数のサーバで同期させるための技術的な方法が存在する．

メールの自動転送

受信したメールを自動転送し，複数のメールサーバ間でお互いにメールを持ち合う形態について考えてみる．例えば 3 台あるメールサーバをそれぞれ A, B, C とする．メールサーバ A が受信したメールを B，C のメールサーバに転送する．メールサーバ B および C も同様に，受信したメールを他の 2 台のメールサーバに転送する．3 台のメールサーバが互いにメールを持ち合う構成となる．メールサーバに加えて，認証サーバのレプリカも複数台設置を行う．持ち合うサーバの台数が増えた場合，転送設定のルールを記述することが非常に困難なものになる．このような方法と本システムを比較すると，本システムは暗号化されたメールを利用者が持つ復号用鍵ファイルで復号する仕組みをとっているので認証サーバは不要であること，他組織のネットニュースサーバを利用することで自組織で管理するサーバを増やすことなくメールの保存先を増やすことができることが優位点としてあげられる．

rsync

rsync [18] は遠隔地のサーバにあるファイルやディレクトリを同期することができる．同期するデータを転送する際に，データの圧縮や暗号化を行うオプションを備えている．mailsync [19] や offlineIMAP [20] を利用すると，imap サーバ上にあるメールを同期することが可能となる．具体的には，imap サーバにログインして指定したフォルダのメールを別の imap サーバや手元の PC にコピーを保存することができる．rsync，mailsync，offlineIMAP を利用することで，メールサーバに保存されているデータを遠隔地のサーバや手元の PC に定期的に保存することができるが，これらの共通点として，保存先のサーバや認証サーバの管理が必要となる．サーバ台数が増えた場合，同期を取る組み合わせが増大し，同期のための通信路が非常に高くなる．

P2P オーバーレイネットワーク

P2P オーバーレイネットワーク上で動作するファイル共有アプリケーションを使用することでデータを同期することができる [21] [22] [23]．P2P オーバーレイネットワークは，ネットワーク上に仮想的なネットワークを構築し，各ノードが直接通信を行う．サーバ・クライアント型のネットワークに比べ，一部のノードに負荷が集中しにくいという特徴を持つ．

P2P は，トラフィックの統制が効きにくくどのようなデータが配送されているかを把握することが困難な状態にある．一方，ネットニュースは 1980 年代後半から利用されてきた成熟したプロトコルであり，ニュースサーバ同士がニュース記事を配送したり利用者がニュースグループにアクセスして記事の閲覧や投稿を行っている等，利用状況を把握しやすいプロトコルとなっているため，明示的に禁止している組織やプロバイダの存在を確認することはできなかった．

4.5.2　ネットニュースシステムを利用したデータ伝送

井澤らの研究では，本研究と同様にデータ配送部分にネットニュースシステムを利用して，配送したいデータを共通鍵で暗号化しテキストデータにエンコード後ニュース記事として投稿し，ネットニュースシステムを使って遠隔のサーバに配送し，受信側ではデコード後共通鍵で復号してデータを取出している [24]．取出したデータはデータベースに保存している．これにより，データを投入するサーバと取出すサーバで同じデータを保持することが可能となる．

本提案システムは，組織やグループで同じデータを持ち合うことが目的ではなく，利用者ごとにネットニュースサーバに接続して記事（メール）を閲覧することを目的としている．そのために，受信者のメールアドレスごとにニュースグループを分け，受信者本人のみが解読できる方法で暗号化を行っている．さらに，通常受信用に利用しているメールサーバに障害が発生した場合は，代替となる別のサーバがメールを受信してニュース記事として投稿するために，利用者ごとに公開鍵をあらかじめニュースグループ内に投稿することで，いつでも暗号化鍵を取出せるようにしている．

4.5.3　メールサーバの冗長化

中筋らの研究では，メールサーバがメールを受け取り送信するまでの間にメールシステムに障害が発生した場合にメールが消失する可能性が高い点に注目し，メールがキューに取り込まれるタイミングで同期を取ることでメールの消失を回避するシステムを提案している [25] ．金らの研究は，インターネットに複数の接続経路を持つマルチホーム環境において，ネットワークの経路制御の観点から，DNS (Domain Name System) サーバへの問い合わせに対する応答を複数の経路からそ

れぞれ異なる内容のものを返すことで最適な経路を選択する手法を提案している [26] ．これらの研究はメールサーバ内の障害やメールサーバまでの経路の障害に対応することが主眼であって，いずれもメールサーバが稼働していることが前提として設計されているが，本研究では障害時にメールサーバにアクセスができない状況を想定しているためこの点が異なっている．

大隅らの研究では，物理的に離れた複数箇所に同じ仮想 IP アドレスを割り当てたサーバを設置し，経路情報を広告する際にメインのサーバはメトリックを小さくしバックアップのサーバはメトリックを大きくしておくことで通常時はメインのサーバに接続され障害時には自動的にバックアップサーバに接続されるよう設計されている [27]．この方法により，サーバ単体やネットワークの障害だけでなく，サーバが設置されている地域一帯が災害によりサービスできなくなった場合であっても，バックアップサーバが遠隔に設置されていれば，遠隔地のバックアップサーバを使って継続してサービスを提供することができる．しかし，コンテンツの同期については触れられていないため，復旧時の対応が難しくなる．

4.6 評価

4.6.1 保存部に格納したメールの同期

本システムを利用すると，メールサーバに到着したメールはネットニュースサーバに投稿された後，ネットニュースの仕組みを使って他のネットニュースサーバに配送される．このため，各ネットニュースサーバには同じメールが保存される．よって [要求 1] を満たすことができると言える．

遠隔地にあるファイルを同期する他の手法として，rsync やメールの自動転送がある．いずれも同期するサーバの台数が多くなると，自組織で管理するサーバ数

が増え管理コストが高くなる．一方，ネットニュースシステムを利用すると，データを分散配置するネットニュースサーバは，他組織が管理しているサーバを利用することも可能なため，管理コストはほとんどかからないというメリットがある．

4.6.2　自分宛のメールの閲覧

本システムを利用すると，メールシステムに障害が発生しアクセスできない状態であっても，ネットニュースサーバにアクセスできれば，利用者が最後にメールを受信した後に配送されたメールに加えて，障害が発生した後に送られてきたメールも読むことができるようになった．一方，本システムを利用しない場合でもメールの自動転送を行っていれば，メールシステムに障害が発生する前に送られてきたメールは転送先に自動的に転送されるので自動転送先で閲覧することができるが，障害発生後に送信されたメールは自動転送されないため閲覧することができない．さらに，本システムも自動転送も利用しなければ，利用者が最後にメールを受信した後に送られてきたメールを閲覧することができない．これらをまとめたものが表 4.2 である．

本システムを利用し，ネットニュースサーバに接続することができ，かつ，ネットニュース記事の保存期間内であれば，常に受信したメールの閲覧が可能となり，[要求 2]，[要求 3] を満たすことができると言える．

普段アクセスしているメールボックスが利用できない場合に自分宛のメールを閲覧するための他の手法として，メールを複数サーバに自動転送しておく方法がある．この方法により，障害時も自分宛のメールを閲覧できるが，障害発生以降のメールは読むことができない．一方，提案システムでは，障害発生以降のメールも読むことができるという点が優位点としてあげられる．

このシステムはメールの配送の仕組みに冗長性を与えたものである．配送の仕

表 4.2: 利用するシステムと閲覧できるメール

	メールを読んだ後に配送されてきたメール	メールシステムに障害が発生した後に送られてきたメール
本システムおよびメールの自動転送とも未使用	閲覧不可	閲覧不可
メールの自動転送利用時	自動転送先で閲覧可	自動転送されないため閲覧不可
本システム利用時	ネットニュースの保存期限内であれば閲覧可	ネットニュースの保存期限内であれば閲覧可

組みに追加の機能を持っており，本来のメールシステムでは送信できなかった状況であっても，ニュースシステムに専用の手順でメッセージを投入することで，相手にメールとして届けようとしてくれる機能が追加されている．これはメールシステムが障害で利用できない状況であっても配送を試みるところが新しく，自分宛のメールを読む手段が増えている．

4.6.3 他人宛てのメールの閲覧

１つのメールアドレス宛てに届いたメールを１つのニュースグループの記事として保存している．アクセスしたニュースサーバの全てのニュースグループを閲覧することができるので，他人宛てのメールが保存されているニュースグループにもアクセスできるが，保存されているメールは暗号化されている．復号用鍵ファイルは利用者本人あるいはメールサーバ内のみ保存されているため，他人宛のメー

ルを復号して読み出す事ができない．よって [要求 4] を満たすことができると言える．

4.6.4　ネットニュースサーバの運用

関西大学発行の「学の実化」[28] によると，メール受信数が一番多い月が 2 月で 1,032,189 通 (学部生)，177,675 通 (院生) で学生数が 28,325 人となっているので，一日平均 43,209 通のメールを受信していることになる．総務省情報通信政策研究所発行の我が国の情報流通量の指標体系と計量手法に関する報告書 [29] によると，メール 1 通のサイズを 18.5Kbyte と仮定しているのでこの値を利用する．受信したメールをネットニュースに投稿する際には PGP による暗号化を行っている．暗号化処理により約 30%増量していると仮定すると，ネットニュースに投稿する 1 通の記事は 24.1Kbyte となる．1 日平均 43,209 通のメールを受信しているので，ネットニュースの記事にすると 1 日平均 1.04Gbyte となる．提案システムでは記事の有効期限を 30 日に設定しているので，31.2Gbyte のデータを保有するディスクスペースがあれば学生数 28,000 名ほどの必要なメールをニュース記事として保存することができる．ネットニュースシステムの運用コストは，自組織でネットニュースシステムを管理する場合は，管理者が定期的にログを確認し，記事の送受信が行えているか，ディスクの空き容量やソフトウェアの更新作業が必要となる．

4.6.5　指定した受信者へのメッセージの送信，自分宛メッセージの取り出し

障害時および平常時に本システムに求められる要件のうち，機能に関しては指定した受信者にメッセージを送信できることと，自分宛のメッセージを取り出せ

ることの 2 つである．ここでの障害時とは，通常利用しているメールボックスが保存されているサーバにアクセスできない場合と定義すると，代替となるメールサーバが稼働していれば，指定した受信者にメッセージを送信することは可能であり，また，いずれかのニュースサーバが稼働していれば自分宛のメッセージを取り出すことができる．平常時は，通常利用しているメールサーバを利用して指定した受信者にメッセージを送信することができ，また，自分宛のメッセージを取り出すこともできる．

性能について求められている要件は，平常時の配送遅延時間は 4 時間以内、障害時の配送遅延時間と閲覧時間の合計値は 12 時間以内である．4.6.4 節で 1 通のニュース記事のサイズを 24.1Kbyte とした．現在稼働しているネットニュースサーバにアクセスし，複数グループに保存されている記事を無作為に 100 通取り出し，記事が配送されてきたサーバ名が保存されている Path: 行を抜き出したところ，平均で 10 のネットニュースサーバを経由していることがわかった．

18.5Kbyte のメールが 24.1Kbyte の記事としてネットニュースに投稿される際にかかる時間は 1 秒以内であった．10 のネットニュースサーバがバケツリレー式に 24.1Kbyte の記事を配送した場合，途中の 9 箇所の回線速度を 100Kbps，途中のネットニュースサーバがリアルタイムに記事を配送せず 10 分おきに配送していると仮定すると，始点のニュースサーバから終点のニュースサーバに到達するためには，サーバ間の伝送に 2 秒かかり，次のサーバに配送されるまで最大 10 分待つことになる．おおよそ 90 分あれば始点のニュースサーバから終点のニュースサーバに記事が配送されることになる．実際には，リアルタイムで記事を配送しているサーバも多いため，90 分よりもっと短い時間での配送が可能と考えられる．平常時の配送遅延時間の 4 時間以内をみたしており，さらに障害時の配送遅延時間とアクセス時間の合計値の 12 時間もみたしていると言える．よって [要求 5] を満

たしていると言える．

4.7　考察

4.7.1　通常時および障害時の運用

図 4.5 のメールサーバ (A)(B) は自組織で運用を行う．ネットニュースサーバ (A)(B)(C) も自組織で運用を行ってもよいが，ネットニュースサーバ (B)(C) は他組織で運用されているサーバを利用することも可能となっている．東日本大震災のような大規模災害時での利用やメールサーバの故障時に利用できることを想定しているので，ネットニュースサーバ (B)(C) はネットワーク的にも物理的にも遠隔地に設置されることが望ましい．

利用者はいずれかのネットニュースサーバにアクセスしてメールを閲覧するので，全てのネットニュースサーバが停止している場合はメールを閲覧することができないが，ネットニュースサーバが 1 台でも稼働していればメールの閲覧が可能となる．さらに，メールサーバが稼働していて，メールサーバが新着メールを投稿するネットニュースサーバが稼働していれば，利用者は障害発生前にメールサーバが受信したメールに加えて，障害発生後に受信したメールも閲覧することができる．図 4.5 を例にして，どのサーバが稼働していればどこまでのサービスを提供できるのかについてまとめたものが表 4.3 である．

メールサーバ (A) が稼働している通常時は，利用者はメールサーバ (A) またはいずれかのネットニュースサーバにアクセスを行う．メールサーバ (A) が停止した場合は，利用者はいずれかのネットニュースサーバにアクセスを行う．メールサーバ (A) の管理者は，サーバ自身の障害であれば復旧に努めるが，大規模な災害により広範囲にネットワークが寸断された状態の場合は復旧を待つことになる．

表 4.3: 稼働しているサーバと提供可能なサービス

	稼働しているサーバ	提供できるサービス
(1)	メールサーバ (A) とネットニュースサーバ (A)	利用者は過去に受信したメールに加えて新着メールも閲覧できる
(2)	メールサーバ (B) とネットニュースサーバ (B)	同上
(3)	ネットニュースサーバ (A)(B)(C) のいずれか稼働，ただしメールサーバ (A)(B) とも停止	利用者はメールサーバ (A) または (B) が稼働している間に受信したメールを閲覧できるが，メールサーバ (A)(B) ともがダウンしてから届いたメールは閲覧できない
(4)	ネットニュースサーバ (A)(B)(C) が全て停止	利用者はメールを閲覧できない

一方で，セカンダリメールサーバを 1 台あるいは複数台用意し，メールサーバの送信待ちキューに保存されている利用者宛のメールを読む仕組みを提供する方法が考えられる．しかし，利用者に見せるためには認証サーバのレプリカを持つ必要があることや，どのセカンダリメールサーバの送信待ちキューに残っているかは利用者には簡単にはわからないので，見つけ出すことが困難になる．

4.7.2 暗号化用鍵ファイル、復号用鍵ファイルの再作成

暗号化用鍵ファイルおよび復号用鍵ファイルを再作成した場合の処理について述べる．旧暗号化用鍵ファイルは，ネットニュースサーバの暗号化鍵ファイル保存用ニュースグループに保管されている．旧復号用鍵ファイルは利用者および図 4.5 のメールサーバ (A) で保管されている．メール保存用ニュースグループに保存

されているメールは旧暗号化用鍵ファイルで暗号化されているので，ニュース記事（メール）を 1 通ずつ取出し，旧復号用鍵ファイルで復号した後，新暗号化用鍵ファイルで暗号化を行い投稿する．旧暗号化用鍵ファイルで暗号化されたニュース記事は削除を行う．暗号化用鍵ファイルおよび復号用鍵ファイルの再作成や利用者が復号用鍵ファイルを入手する部分はまだ実装できていないので，今後実現していきたい．

4.7.3　記事の既読・未読管理

障害時に図 4.5 のネットニュースサーバのいずれかで閲覧したメールは，障害復旧後にメールサーバ (A) で新着メールとして再度閲覧することとなるが，どのようにすれば既読メールと見なせるかについて述べる．

ネットニュースのクライアントソフトは，ニュース記事に割り当てられた記事番号を利用して既読・未読の管理を行っている．記事番号はニュースサーバごとに発番されているため，同じニュース記事であってもネットニュースサーバが異なれば，記事番号が異なる．ニュース記事を投稿する際に全てのニュース記事でユニークな Message-ID が自動的に付加され，以降書き換えられることはない．そこで，ネットニュースのクライアントソフトが持つ既読・未読を管理するデータベースに Message-ID の情報を追加し，何らかの方法で（例. ニュースグループに投稿する等）障害復旧時に図 4.5 のメールサーバ (A) が参照できるようにしておく．メールサーバ (A) がスプールに書き込む際に，あるいは，利用者がメールサーバ (A) のメールを受信する際に，既読であるという情報を付加できないかと考えているが，このあたりは今後の課題としたい．

4.7.4 メールサーバ障害時の切り替え

提案システムでは，DNS の MX レコードに複数のメールサーバを登録しておき，利用者のメールボックスを保有するメールサーバの MX レコードは優先度を高く，障害時に代替となるメールサーバの優先度を低く設定しておく．メールサーバの障害時には，自動的に代替となるメールサーバにメールが届き，ネットニュースシステムにより各ネットニュースサーバにメールが分散配置されるので，利用者はいずれかのニュースサーバに接続して自分宛のメッセージを閲覧する．

一方，組織やプロバイダのメールサーバは，DNS や負荷分散装置等を使って複数台で運用されていることが多い．メールサーバ障害時は，障害のあったメールサーバを DNS の切り替えや負荷分散装置からの切り離しにより，メールサービスの停止を回避していると考えられるが，負荷分散装置を含むメールシステム全体の障害や大規模災害発生時にメールサービスを継続することは難しいと予想される．

障害時のシステム切り替え方法とアナウンス方法について，提案システムと組織やプロバイダのメールシステムの比較を行う．

障害時のシステムの切り替え方法は，提案システムでは自動的に代替となるメールサーバがメールを受信するが，組織やプロバイダのメールシステムでは障害のあったメールサーバを負荷分散装置から切り離すことでサービスを継続できる場合もあれば，復旧するまでサービス停止を余儀なくされる場合もある．

障害時のアナウンス方法は，提案システムでは平常時に利用者に自分宛のメッセージが保存されているニュースサーバの情報を伝えておくことで障害が発生しても特にアナウンスを行う必要がないが，組織やプロバイダのメールシステムでは Web システム等を利用してアナウンスを行うこともあるが，障害の程度によりアナウンスが行えない場合も考えられる．

4.7.5　記事のウイルスチェック，SPAM チェック

新着メールがメールサーバに到着後、ウイルスチェックや SPAM チェックの判定が終了し、利用者のメールボックスに保存されるという流れになっていることが多い．利用者のメールボックスに保存されるメール全体（ヘッダとメール本文）を本システムで取り出し暗号化してニュース記事として投稿している．そのため，既存のウイルスチェックおよび SPAM チェックサーバの処理を利用しつつ本システムを稼働させることができる．

4.7.6　メッセージを閲覧するクライアントソフト

ニュースサーバに自分宛に届いたメッセージを閲覧する際には，ニュースサーバに接続後，PGP による復号が必要となる．本章内では本機能を有する Thunderbird を利用した．本機能を有しないクライアントソフトでは，別途本機能に相当する仕組みの提供が必要となるため，今後の課題としたい．

4.8　おわりに

本章では，メールサーバの受信部と保存部を分離して受信部および保存部を複数箇所に設置し，保存部をインターネットを介して同期するモデルを考案した．また，保存部の同期にはネットニュースの仕組みを利用した．受信したメールは受信者本人のみが解読できる方法で暗号化されニュースの記事として投稿され，ニュースサーバに配送される．受信者はメール/ニュースクライアントソフトである Thunderbird を使って自分宛のメールが保存されているニュースグループにアクセスし，暗号化されたメールを復号して閲覧するという流れになっている．メールシステムが障害により受信できなくなっている間に送られてきたメールも，代

替となるメールサーバが受信し暗号化を行いニュース記事としてニュースサーバに投稿することで，メールシステムに障害が発生している間でもニュースサーバにアクセスすることができればメールを閲覧できることを確認した．本システムは保存部を単純に分散させるシステムとは異なり，いずれかのニュースサーバに接続することができれば，自分宛に送られているメールを読むことができるという機能を提供できている．本システムより，メールシステムの障害が利用者に与える影響を軽減することが可能となった．今後は実運用を行い，それによって得られた知見を元にシステムの改善を行ってゆきたい．

第5章　災害時の利用を想定した複数の通信手段を併用するメッセージング方式

5.1　はじめに

阪神淡路大震災や東日本大震災をはじめとした甚大な被害をもらたす災害が発生した際に，組織の構成員がどのような状態におかれているのかを把握することは，組織の管理者にとって極めて重要である．阪神大震災の際には，電話はほとんど通じなかった．また 16 年後の東日本大震災でも電話もメールもほとんど利用できなかったが，Twitter は断続的に利用が可能であった．災害時に有効となる通信手段は，時代や環境によって大きく変化しているので，今後，災害が発生した場合にどのような通信手段が有効であるのかを現時点で想定することは非常に困難である．そこで本章では，Twitter やメール等，既存の通信手段を複数利用してできるだけ頑強にメッセージのやり取りを行えるシステムを提案する．

日常的に利用されていないシステムを災害時に利用することは困難である．緊急時の利用を想定したシステムでは性能だけでなく使いやすさにも注目する必要がある．そこで，提案システムでは学習コストが少ない電子メールのユーザインタフェースを利用する．

5.2　関連研究

安否確認の重要性は企業や教育機関でも高まっており，すでに様々なサービスが提供されている．NTT 西日本では災害用伝言ダイアル [30] を，NTT 東日本では災害用掲示板 [31] を提供している．いずれも災害時に電話がつながりにくい状況が発生すると，提供が開始されるサービスである．災害用伝言ダイアルは，1 回の伝言で 30 秒以内の録音が可能で 48 時間保存される．災害用掲示板は，1 回の伝言で 100 文字まで，最大 20 件の蓄積が可能で 8ヶ月間保存される．

富士通が提供している緊急連絡 / 安否確認サービス [32] は，緊急時はもちろんのこと通常時の連絡手段としても利用できることが特徴としてあげられる．災害時にはネットワークが輻輳することを考慮して，メールの返信はひらがな 1 文字にする，専用アプリで回答する等の機能を有している．

田丸ら [33] は，オーバーレイネットワーク上に安否確認システムを構築している．利用者が自由にインターネット環境を使えない場合でも，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仕組みとなっている．携帯電話などが使用できない環境で有力となると考えられる．

梶田ら [34] は，名古屋大学ポータルと呼ばれている安否確認システムを運営している．災害時に自分自身の安否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．自分自身の安否情報だけでなく，連絡先情報や友人の安否情報も入力することができる．名古屋大学では毎年安否確認システムを利用した災害訓練も行われている．名古屋大学が発行するメールアドレスとは別に緊急時連絡用メールアドレスを収集しているが，すべての構成員の情報が入手できていないこと，安否確認手段をメールだけでなく電話やハガキ等の手段を考慮する必要があることが今後の課題としてあげられている．

5.3 災害時に利用するシステムの問題点

災害時に利用するシステムについての課題を 3 つ示す.

- [**課題 1**] 受信者にメッセージが届いたかどうかの把握が困難であること

  災害時にはメッセージの配送の遅延や配送の失敗が発生する可能性が高くなる．電子メールを送信した際，受信者のメールボックスに配送できなかった場合はエラーメールが返送されてくるので，送信が完了しなかったことがわかる．電子メールは送信に成功しても受信者からの返信を受け取るまで，受信者がメッセージを閲覧したかどうかは把握できない．同様に SNS (Social Networking Service) を使ってメッセージを送信した場合も，返信を受け取るまで受信者がメッセージを閲覧したかどうかは不明である．災害時において，受信者がメッセージを閲覧したかどうかは，安否を確認する上で重要な情報であるが，これを確認するすることはとても困難である．

- [**課題 2**] 災害時に利用するシステムが使っている配送手段の障害に弱い

  利用する配送手段が災害時に障害により利用できない場合，相手にメッセージを送ることができない．2011 年に発生した東日本大震災では，ある場所では携帯電話の基地局が被災したために通話やキャリア通信によるメッセージの送受信ができなかったが，インターネット回線は稼働していたので，SNS や Twitter を利用してメッセージをやりとりすることができた．逆に SNS や Web サービスのサーバに障害が発生すれば，これらを利用したメッセージの送受信ができないが，携帯電話を利用できるという場面も考えられる．ひとつの配送手段に頼ると，その配送手段が利用できない場合はメッセージの送受信ができなくなるという問題がある．

- [**課題 3**] 災害時だけの利用を想定災害時のみに利用され，平常時の利用は想定されていない

  災害時に実際に利用する際には，使い慣れていないため，使い方がわからない，使い慣れていないシステムを使いたくない，といった理由により，利用されにくくなると考えられる．

  また，災害時だけ利用されるシステムは，定期的なメンテナンスが行われていない可能性があるため，実際に使用する際に正しく動作するか，受信者のアドレス帳は正しく設定されているかどうかは不明である．

5.4　要求要件

提案システムに求められる要求要件を示す．

- [**要求 1**] 既存の配送手段を利用すること

  災害用のために準備した配送手段ではなく，日頃から利用されている配送手段を用いることとする．通常利用する手段は，定期的にメンテナンスされていると考えられる．

- [**要求 2**] 配送手段を複数利用すること

  単一の手段に頼った場合，動作しなければメッセージを配送する手段を失うこととなる．これをぜひとも避けるために，配送手段を複数用意することで，より確実にメッセージを配送することが可能となる．

- [**要求 3**] 配送手段を複数利用すること

災害時に複数ある配送手段のどれを利用することが適しているかを送信者が確認することは困難であるため，送信者が配送手段を意識することなくメッセージを配送できることが望ましい．

- [**要求 4**] 配送手段の追加が容易なこと

  今後新しい配送手段が出現した際に，容易に追加できることが重要である．

- [**要求 5**] 受信者にメッセージが到着したこと，受信者が閲覧したことを送信者が確認できること

  メッセージの返信がない場合，受信者にメッセージが届いていないのか，受信者が返送したメッセージが届いていないのか，受信者が怪我で返送できないのか等の原因が考えられる．そこで，メッセージが正常に配送できたのか，受信者が閲覧したのか等の情報を送信者が確認できることは非常に重要である．

- [**要求 6**] 災害時にも平常時にも利用できること

  災害時の利用を前提としたシステムは日常的に利用されることを想定していない場合が多い．そのため，利用する際に操作方法がわからない，システムへのアクセスの方法がわからない，あるいはシステムの存在を知らないといった問題が発生する．このことから平常時に利用するシステムと災害時に利用するシステムを別のものと考えるのではなく，災害時に利用するシステムと全く同じものを平常時に利用する，あるいは，平常時に利用しているシステムの機能を有効的に利用して災害時にも利用することを考える．

- [**要求 7**] 利用者にとって学習コストが低いこと

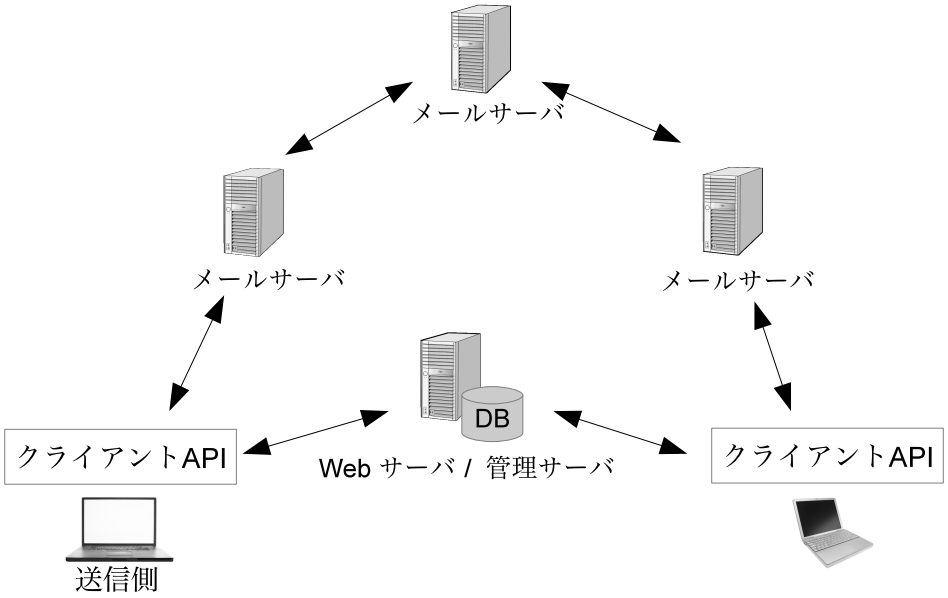

図 5.1: 先行研究：システム構成遷移図

災害時に新たな操作方法を学習することは困難である．また災害時には人的資源が限られるため，誰でも操作できるシステムが必要である．

5.5　先行研究

甲賀ら [35] は，複数の手段でメッセージを届けるシステムを提案している．図 5.1 にシステム概要図を示す．

メッセージを送信する手段として，メールサービスと Web サービスを利用している．メッセージが到着した，メッセージを閲覧した等のメッセージの状態をデータベースを使って管理している．

甲賀らのシステムは，5.4 章の要求 1 から要求 4 を満たしている．しかし，新しい通信手段を追加しようとすれば，システム全体を作り直す必要がある．これは

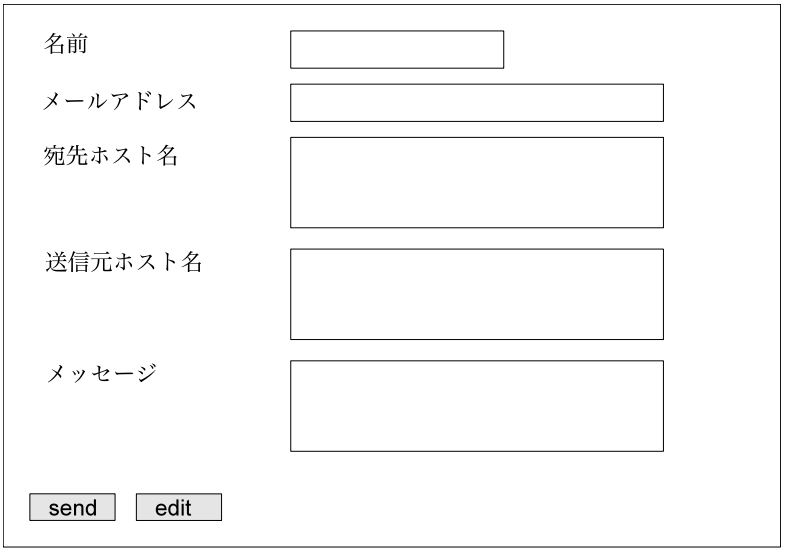

図 5.2: 先行研究：管理者用メッセージ送信画面

要件 5 を満たさない．また，災害時に利用することを想定して作られたシステムであるので，日常的に利用することは考えられていない．これは要件 6 を満たさない．

図 5.2 は，送信する際に利用するメッセージ送信フォームである．これを利用するためには，受信者のメールアドレスや Web サーバのホスト名を指定する必要がある．また，受信したメッセージを閲覧する際，利用者は新たにメッセージを閲覧するためのツールを使用しなければならない．これは要件 7 を満たさない．

5.6 提案システム

5.4 章の要求要件より，提案システムの概要について述べる．提案システムは 5.3 章に示すように，複数の配送手順でメッセージを配送するメッセージ配送部と，配送されたメッセージを表示するメッセージ表示部から構成されている．

5.6.1　メッセージ配送部

複数の配送手段で共通に使用する配送の仕組み

配送手段はメール，Twitter，facebook 等複数あり，それぞれ特徴や制限が存在する．今後は新たな配送手段が普及していくと考えられる．そこで，配送手段ごとに異なる仕組みを用いてメッセージを送信するのではなく，すべての配送手段で共通に利用できる仕組みを定義し，配送手段ごとに共通の仕組みで配送を行えるように実装を行う方針とする．

メッセージの種類

共通の仕組みで配送するための，メッセージの種類について述べる．

既存の複数の通信手段を使ってメッセージを配信するために，やり取りを行うメッセージを次の 3 種類とする．

- **本文送信**

  送信者が受信者へメッセージを送信する．

- **到達確認**

  メッセージが到着すると，到着を確認したことを意味するメッセージが送信者に返送される

- **閲覧確認**

  受信者がメッセージを閲覧すると，閲覧したことを意味するメッセージが送信者に返送される

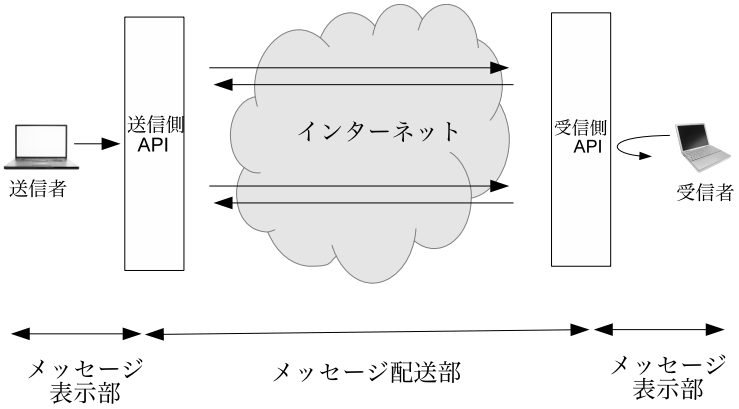

図 5.3: システム概要図

メッセージを送信後，受信者が閲覧するまで，これら 3 種類のメッセージとシステムの状態遷移を図 5.4 に示す.

メッセージ送信の 4 つのモデル

メッセージを送る方法として次の４つのモデルを定義し，複数の通信手段をいずれかのモデルにあてはめて実装を行う.

- 受信者数

  - 1 対 1 のメッセージ:

  ひとりの受信者へ１回でメッセージを送るモデル

  - 1 対多のメッセージ:

  複数の受信者へメッセージを一括して送るモデル

- メッセージサイズ

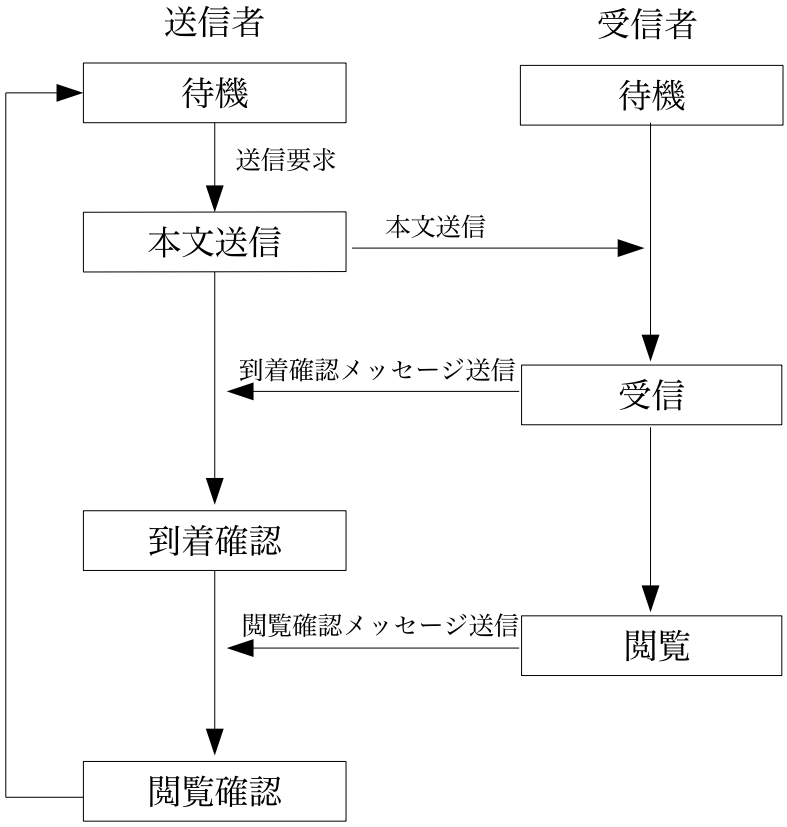

図 5.4: システム状態遷移図

- 分割するメッセージ:

1回で送信できるサイズを超えているため，1つのメッセージを複数に分割して送信するモデル

- 分割しないメッセージ:

1つのメッセージを分割せずにそのまま送信するモデル

メッセージフォーマット

共通の仕組みを使って配送するためのメッセージフォーマットについて述べる．本文送信，到着確認，閲覧確認を通知する際の共通のメッセージフォーマットを

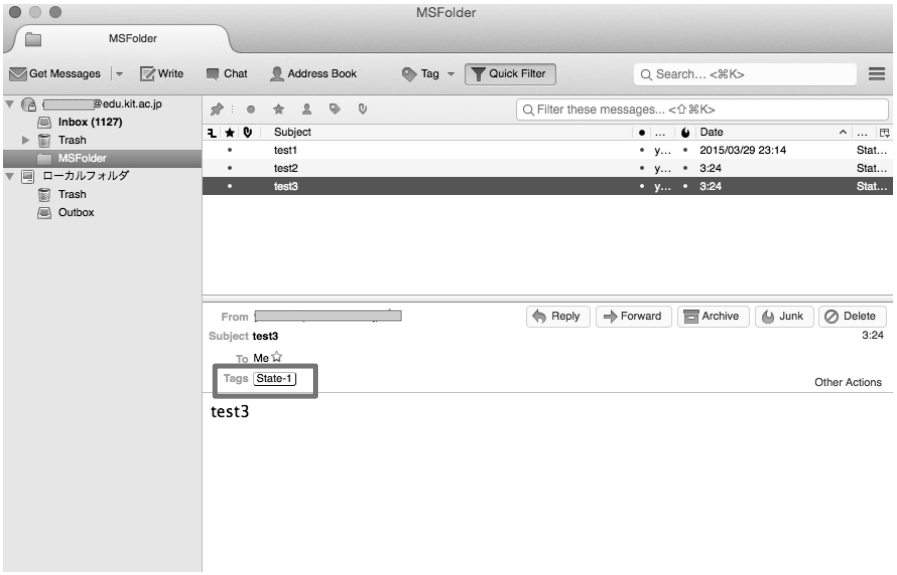

図 5.5: Thunderbird での状態 ”state-1” の例

表メッセージの状態を表 5.1 に示す.

5.6.2 メッセージ表示部

メッセージの閲覧には，電子メールクライアントを利用する．電子メールクライアントは多くの人が利用しており，学習コストが非常に低いと考えられる．そのため，災害時だけでなく，日常的な利用が可能であると考える．ここでは，IMAP クライアントとして広く利用されている Thunderbird [36] を利用する.

ステータスの遷移

メッセージ送信者が次の 3 つの状態を把握できることが求められている.

- **状態 ”state-1”**

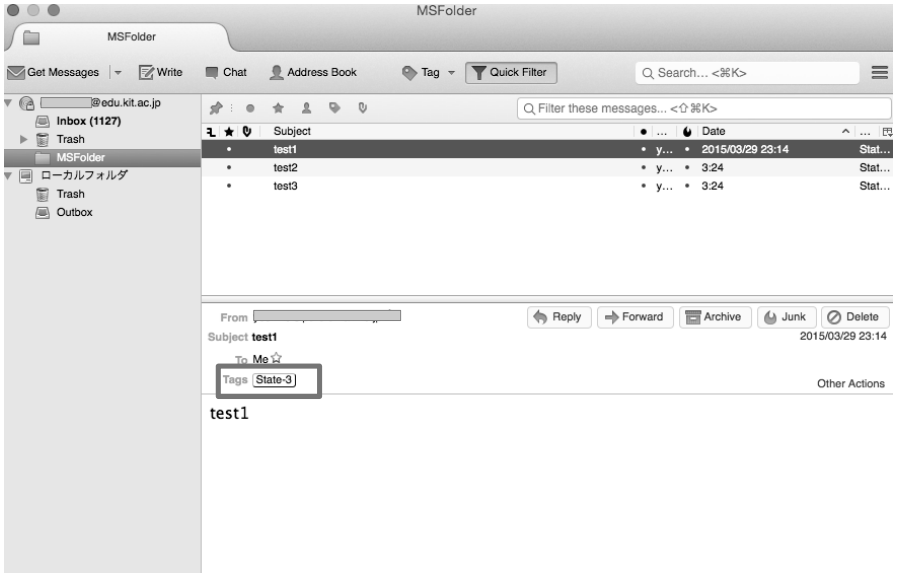

図 5.6: Thunderbird での状態 ”state-2” の例

送信者が受信者へメッセージを送信する．この時点ではまだ受信者には届いていない．メッセージは指定されたフォルダ (MS フォルダ) に保存される．

- **状態 ”state-2”**

  メッセージが受信者に届く．受信者にメッセージが届いたことを確認するために，確認メッセージが送信者に送られる．

- **状態 ”state-3”**

  受信者がメッセージを閲覧すると，閲覧したことを意味するメッセージが送信者に返送される

提案システムの状態遷移図を図 5.4 に示す．

表 5.1: メッセージフォーマット

項目	摘要
送信者アドレス	送信者のアドレス
受信者アドレス	受信者のアドレス
メッセージ ID	メッセージごとにユニークとなる ID で，送信時に発番
ペイロード	本文
送信日時	送信した日時
受信日時	受信した日時
使用経路	資料する配送手段
状態フラグ	メッセージの状態を表したもの（本文，到着確認，閲覧確認）

ステータスの表示

メッセージのステータスの表示は，IMAP のタグを利用している．

メッセージの取り扱い

受信者にメッセージが到着したとき，また，受信者が閲覧する操作を行ったとき，到着したことや閲覧したことを示すメッセージが自動的に送信者に送信される．送信者は，メッセージを送信する際に専用フォルダに送信したいメッセージを保存する．新しいメッセージが専用フォルダに保存されたことをシステムが検出すると，受信者宛にメッセージを送信する．受信者側から通知される到着確認や閲覧確認のメッセージにより，保存されたメッセージの状態を IMAP のタグに対応づけて，状態 ”state-1” から状態 ”state-2” や状態 ”state-3” に変更していく．

5.7　おわりに

本章では災害時に安否確認を行うシステムを提案した．提案システムでは，信頼性の高い送信を実現するために複数の配送手段を利用している．今後新たな配送手順が普及しても，メッセージ送信の３つのモデルやメッセージフォーマットを利用して実装すれば，本提案システムに組み込むことが可能となる．今後の課題として，実環境で実装を行い，利用者の利便性等の評価やアドレスの管理方法等について検討していきたい．

第6章　結論

本論文では，非常時にメッセージをより確実に伝達する方式について論じた．

メール受信時に自動転送を行った際、自動転送先のメールアドレスが無効となった場合のエラーメールを、自動転送を行っているメールサーバで受信するシステムを提案し有効性を確認した．このシステムは常時より稼動させておくことで，自動転送設定を行った利用者が，自動転送先のメールアドレスが無効になっていることに気が付き修正することを可能にしている．これにより，転送エラーを示すエラーメールを削減できるだけでなく，転送先メールアドレスを常に最新の状態で保持することができるため，非常時にメールを送信した際により確実に受信者に届けることができる．

また，受信したメッセージを複製した上で遠隔にある複数箇所で保存しておき、受信者はメッセージが保存された複数箇所のどこか 1 箇所にアクセスすることができれば，自分宛のメールを受信することができるシステムを提案し有効性を確認した．受信したメッセージを複数箇所に配送して保存する技術は，インターネットで古くから利用されているネットニュースを用いた．ネットニュースは何年にもわたり多量のメッセージを配送し続けた非常に成熟したプロトコルであり，現在でもメンテナンスされている．本来ネットニュースは多くの人が閲覧することを前提に設計された仕組みであるため，メールの受信者本人だけが解読できる方法で暗号化した上でネットニュースを使って配送するようにした．遠隔地にメッセージの保存箇所を増やす場合は，遠隔地のサーバにネットニュースを稼動させるだ

けでよく，非常に低コストで堅牢なシステムを構築することが可能である．災害時だけでなく，メールサーバやネットワーク機器の故障時にも遠隔地のサーバにアクセスできれば自分宛のメールを受信できるため，有効に利用できる方法であるといえる．

さらに，災害時にどのような通信手段が有効かを推定することは困難であるため，Twitter やメールなど既存の複数の通信手段を利用していずれかの方法で相手に到達できればメッセージのやり取りが行えるシステムのプロトタイプを作成し動作確認を行った．これにより，何か１つの通信手段が利用できれば相手にメッセージを届けることが可能となった．今後，新しい通信手段が広まった場合にも，柔軟に対応できるようなシステムとしている．

大規模な災害時にはインフラが寸断されてしまうが，そのような状況であっても通信手段を確保するために，災害地にいる人の携帯端末から携帯端末へデータを伝送する技術が考案されている．本論文で論じた伝達方式に，インフラが寸断されている状況で通信手段を確保する技術を組み合わせることで、より強固にメッセージを伝達することが期待できる．

非常時におけるメッセージの伝達は，今後もその重要性を増し，ますます種々のサービスが提供され欠くことのできないものとなることは疑いの余地がない．最近頻発する大地震や異常気象による大災害やパンデミックなど，緊急時に組織の構成員がどのような状態に置かれているのかを早期に把握することができれば，その後の復旧計画や事業継続に向けた行動に移ることができる．

本論文で述べた低コストで堅牢なメッセージング方式やその仕組みのが，今後さらに開発が進むであろう非常時のメッセージングシステムに基礎的な技術の一部として役に立つと考えられる．

謝辞

本論文は，京都工芸繊維大学大学院工芸科学研究科設計工学専攻において分散システム研究室での研究をまとめたものです．

京都工芸繊維大学情報科学センター桝田秀夫教授には，本論文の主査として，また，主任指導教員として，研究室に入ってから一貫して辛抱強くご指導いただきました．研究内容への指導はもとより，研究の進め方，関連研究の検索の方法，論文の書き方，プレゼン資料の作成の方法，発表の仕方に至るまで，研究に関する全ての面において繰り返し繰り返しご指導いただきました．親身にかつ根気強くご指導いただけたおかげで，この論文をまとめることができました．本当にありがとうございます．

また，ご多忙中にも関わらず貴重な時間を割いて本論文を査読していただいた京都工芸繊維大学大学院工芸科学研究科 渋谷 雄教授ならびに同研究科 稲葉 宏幸教授に深くお礼申し上げます．

社会人として勤務しながらの学生生活であり，両者にバランスよく時間を使えなかったため，博士後期課程に入学してから5年の時間が過ぎてしまいました．桝田秀夫先生には，いつもご心配ばかりおかけしてしまいました．

分散システム研究室のみなさまには，いつも多くの貴重なご意見をいただきました．また研究室の雰囲気がいつも非常によく，楽しく過ごせたことも，私にとってとても有難いことでした．本当にありがとうございます．

いつも応援してくださっていた職場のみなさまに感謝します．

最後になりましたが，迷惑をかけることが多かった家族に深く感謝し，謝辞の言葉とさせていただきます.

参考文献

[1] Yoshiko Ishibashi, Takumi Kohga, Hideo Masuda : ”Implementation and Evaluation of a Mail Gateway System Reducing Bounce Mail Caused by Erroneous Forward Settings”, SNPD2012, pp. 710-715 (2012).

[2] 石橋 由子，桝田 秀夫：ネットニュースシステムを利用した耐障害性の高い電子メール型メッセージングサービスの提案，情報処理学会論文誌，Vol.57, No.3，pp.976-988，(2016)

[3] Yoshiko Ishibashi, Ryuki Takeda, Hideo Masuda : ”Proposal of messaging system for use in the disaster with one or more communication metods”, ACIT2015, pp. 137-141 (2015).

[4] google.org プロジェクト，Google パーソナルファインダー，https://www.google.org/personfinder/japan/，(2016.6.15).

[5] エヌ・ティ・ティレゾナント株式会社，Janpi
UTFFF5E 安否情報まとめて検索
UTFFF5E，http://anpi.jp/，(2016.6.15).

[6] 総務省，無線 LAN ビジネスガイドライン．http://www.soumu.go.jp/main_content/000233881.pdf，(2016.6.15).

[7] 無線 LAN ビジネス推進連絡会，大規模災害発生時における公衆無線 LAN の無料開放に関するガイドライン，http://www.wlan-business.org/info/pdf/Wi-Fi_Free_Guideline_v1.01_20140527.pdf，(2016.6.15).

[8] 小山 由，水元旭洋，今津眞也，安本慶一：災害データベース・Twitter と連携する DTN ベース災害安否確認システムの提案，マルチメディア通信と分散処理ワークショップ 2011 論文集，pp.89-93，(2011)

[9] J. Klensin (ed.): “Simple Mail Transfer Protocol”, RFC5381, IETF (2008).

[10] P. Resnick (ed.): “Internet Message Format”, RFC5382, IETF (2008).

[11] Geoff Mulligan: “SPAM の撃退”, ピアソンエデュケーション, (1999).

[12] D. J. Bernstein: “Variable Envelope Return Paths”, http://cr.yp.to/proto/verp.txt, (参照 2009-11-09).

[13] 国土交通省 近畿地方整備局 震災復興対策連絡会議発行，阪神・淡路大震災の経験に学ぶ，http://www.kkr.mlit.go.jp/plan/daishinsai/index.html, (2015.09.23).

[14] C. Feather : ”Network News Transfer Protocol (NNTP)”, RFC3977, IETF (2006)

[15] INN (IFull-featured, flexible and configurable news server), http://www.isc.org/software/inn, (2015.06.01).

[16] Thunderbird, http://www.mozilla.jp/thunderbird/, (2015.09.26).

[17] Henry Spencer, David Lawrence : Usenet ネットニュース管理, オライリー・ジャパン, (1999).

[18] http://rsync.samba.org/ (2015.06.01).

[19] http://mailsync.sourceforge.net/ (2015.06.01).

[20] http://offlineimap.org/ (2015.06.01).

[21] 江崎 浩 (監修)：P2P 教科書, インプレス R&D, (2007).

[22] ネットワーク高度利用協議会：http://www.isc.org/software/inn, (2015.06.01).

[23] BitTorrent：http://www.bittorrent.com/, (2015.06.01).

[24] 井澤 志充：NetNews を使った信頼性のあるデータ通信の技法, 分散システム運用技術, 9-9, pp.49-54 (1998-05-15).

[25] 中筋 香里 , 泉 裕 , 齋藤 彰一 , 塚田 晃司 , 上原 哲太郎 , 國枝 義敏：メールシステムの信頼性に関する一考察, 情報処理学会研究会報告, 2004-DSM-034, pp.13-18 (2004-07).

[26] 金 勇 , 山井 成良 , 岡山 聖彦 , 清家 巧 , 中村 素典 ：マルチホーム環境における DNS 応答の多重化による自組織宛メール配送の動的経路選択手法, 情報処理学会論文誌, Vol. 51, No. 3, pp.998-1007 (2010-03-15).

[27] 大隅 淑弘 , 山井 成良 , 藤原 崇起 , 岡山 聖彦 , 河野 圭太 , 稗田 隆 ：IP alias と経路制御を用いた複製サーバ冗長化構成, 情報処理学会研究報告インターネットと運用技術（IOT）,2012-IOT-18(4),1-6.

[28] 学校法人関西大学自己点検・評価委員会（大学部門委員会），関西大学「学の実化」，http://www.kansai-u.ac.jp/Jikotenken/pdf/databook2013.pdf，(2015.09.23).

[29] 総務省情報通信政策研究所，我が国の情報流通量の指標体系と計量手法に関する報告書，http://www.soumu.go.jp/main_content/000030652.pdf, (2015.09.23).

[30] NTT 東日本，災害用伝言ダイヤル (171)、https://www.ntt-east.co.jp/saigai/voice171/, (2015.03.10).

[31] NTT 東日本，災害用掲示板 (web171)，https://www.web171/jp/, (2015.03.10).

[32] 富士通，緊急連絡 / 安否確認システム，http://jp.fujitsu.com/solutions/safety/bc/incident/emergency-email/, (2015.03.10)

[33] 田丸 純，阿部 紘一，島 和之，前田 香織：オーバー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，イ情報処理学会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

[34] 梶田 将司，他：名古屋大学安否確認システムの現状と東日本大震災からの教訓，電子情報通信学会技術研究報告，電子情報通信学会技術研究報告，111(247)，45-50，2011

[35] 甲賀 拓実，石橋 由子，桝田 秀夫：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，情報処理学会研究報告インターネットと運用技術，2013-IOT-20(1)，1-6

[36] Thunderbird (2015, Mar 25), [Online] http://www.mozilla.jp/thunderbird