

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システムの試作とその評価

石橋 由子^{†1} 榊田 秀夫^{†1}

電子メールはインターネット上において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、ひとりで複数のメールアドレスを持ち、その間でメールを転送しているケースも多い。ところが携帯電話のアドレスのように何度も変更する傾向のあるアドレスを転送先に指定している場合、アドレスを変更した際に転送先のメールアドレスの設定変更を忘れるケースが多い。この場合、メールを転送できなくなり、元の送信者のもとにエラーメールが届けられるが、転送設定を行った本人の元には通知されないため、エラーとなっていることに気がつきにくい。そこで、本稿では転送先アドレスが無効であった場合のエラーメールを、転送を行っているメールサーバで受信し、以降のエラーメールを抑制するシステムを提案する。転送エラー時のメールを制御するために、転送先のメールアドレスに対応したメールアドレスにメールのエンベロープ `From` のアドレスの書き換えを行い、エラーメールを管理するデータベースを用意して、転送先のメールアドレスごとのエラー状態を把握できる構成とした。

Trial and Evaluation of a Mail Gateway System Reducing Bounce Mail Caused by Erroneous Forward Settings

YOSHIKO ISHIBASHI^{†1} and HIDEO MASUDA^{†1}

Email is one of the most popular and easy-to-use communication tool on the Internet. These days many people have more than one email address. While automatic forwarding mechanism is useful under such circumstances, one big problem has arisen. When one changes his or her email address that is a forwarding destination and leaves forward settings untouched, forwarded emails lose their way and cause a pile of error notification mails. This is often the case with some kind of email addresses, like for mobile phones. Since error notification only goes to the original sender and administrators, the person directly concerned would not be aware of the error. This paper describes a system to solve this problem. By changing the "envelope from address" of each forwarding mail, problematic destinations can be detected and the user originated the error is warned of what happens.

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転送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、メールの送信者へのみ

エラーメールとして知らされ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ものの、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。

本稿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アドレスを、メールの転送を行っているマシン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

^{†1} 京都工芸繊維大学

Kyoto Institute of Technology

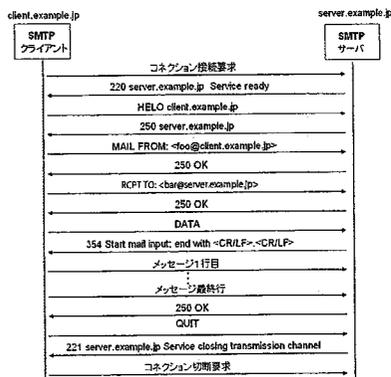


図1 SMTP コマンドと応答コードによるメール送信手順

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出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。

2. 電子メール転送の問題点

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、RFC5381¹⁾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レスが設定される。SMTPサーバに存在しない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、RFC5382²⁾に規程されている。メールアドレスの形式はxxx@yyy.example.jpとなっていて、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。

2.2 メール転送時の問題点

図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と図4はメール転送先アドレスが無効

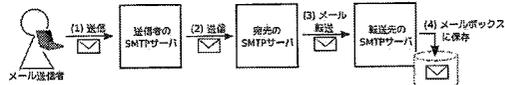


図2 メール転送先アドレスが有効な場合

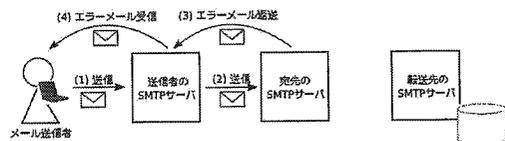


図3 メール転送先アドレスが無効な場合(転送先メールアドレスのドメインの誤り)

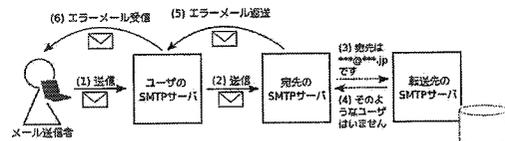


図4 メール転送先アドレスが無効な場合(転送先メールアドレスのローカルパートの誤り)

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2の(2))。宛先のSMTP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2の(3))。

図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の(3))、元のメール送信者はエラーメールを受信する(図3の(4))。

図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2、図3と同じである。図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ないので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4の(5))、元のメール送信者はエラーメールを受信する(図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の(3)および図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ため、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異なるので、誰あてのもの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えたのか、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. 要求要件

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、導入の手順も極力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。

4. 提案システムの設計

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でメールを受信できないのか」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、5.1節で述べる。

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ければ、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、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、4.3節から4.7節において、各部の動作の詳細を示す。

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)
- (2) 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。
- (3) 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。

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。

4.5 メールキュー削除部

メール転送後、一定時間(設定項目)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。

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。

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。

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 (バージョン 2.5) 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 を利用した。

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、procmail³⁾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
```

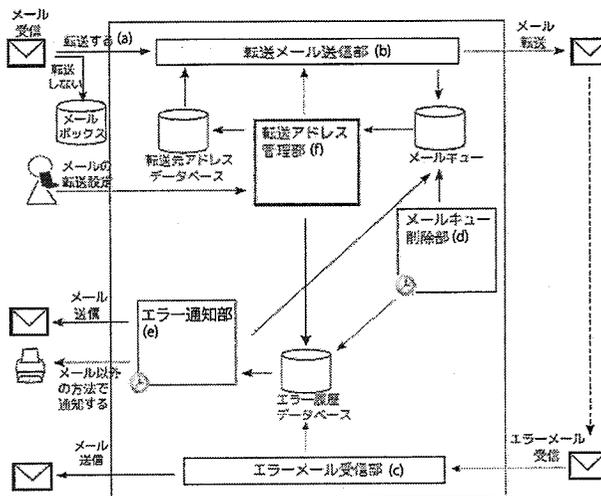


図 5 システム概要図

表 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

| /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 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。
- (5) 転送した情報を転送データベース (表 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-

表 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

b-c のうち、a はメールキューデータベース (表 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 ので、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。

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

表 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

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) にデータを追加する。

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ものとみなし、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出す
- (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。

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

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ことと判断し、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。

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が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。

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)⁴⁾ とし、sendmail の userdb がそれに相当する。

VERP は、エンベロープ From アドレスに受信者のメールアドレスを挿入している。これにより、返送されてきた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、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 とともに、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

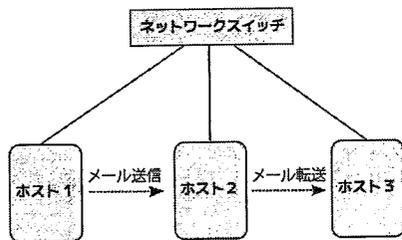


図6 転送時間を測定した環境

表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'sHP ProLiant note	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

から「誰」の転送先にメールがエラーになっているのかを把握できる。しかし本稿は、ユーザの転送先ごとにエンベロープ 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。

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。

6.2.1 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を表5、本システムを使用した場合の結果を表6である。

表5表6を比較すると、転送を開始してからのホスト1からホスト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

表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 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

表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 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

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。

6.2.2 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) のファイル数について測定したものが表7である。

表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

表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	

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。1000通のメールの送信開始から終了まで、転送に要する時間は常にほぼ一定の数値(0.26秒)を示した。運用時の負荷を想定する際に、転送に要する時間や、本システムを利用するユーザ数、利用するユーザ宛に届くメールの到着間隔が参考になると考えられる。

6.3 従来のメール転送時の問題点

2.2節“メール転送時の問題点”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ため、[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、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信するという問題点をあげた。転送エラー時は、本システムが元の送信者にエラーメールを返送する。エラーメールの内容は、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、転送設定者のアドレスを明記して転送先のアドレスが無効であるとする。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なければ、転送先のアドレスを明記してこのアドレスが無効であるとする。これは、転送設定者のメールアドレスを明記すると元の送信者が知ることであり、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。

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られる。

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。

7. まとめ

本稿では、誤った転送設定によるエラーメールを削減することを目的として、エンベロープ From を書き換えることでエラーメールの返送先を制御し、エラー状態を把握できるシステムの試作と評価を行った。今後の課題としては、実際のメール環境に導入し、改善点を洗い出していきたい。

参考文献

- 1) J. Klensin (ed.): “Simple Mail Transfer Protocol”, RFC5381, IETF (2008).
- 2) P. Resnick (ed.): “Internet Message Format”, RFC5382, IETF (2008).
- 3) Geoff Mulligan: “SPAMの撃退”, ピアソンエデュケーション, (1999).
- 4) D. J. Bernstein: “Variable Envelope Return Paths”, <http://cr.yip.to/proto/verp.txt>, (参照2009-11-09).