

## 大規模メール配信とパフォーマンス

千 田 康 弘<sup>†</sup> 渡 辺 裕 明<sup>†</sup>

インターネットの普及に伴い、電子メールの普及が進み、携帯電話のメール機能がメール利用者数の増大に拍車をかけた。現在ではメール自体の利用価値も変化し、マーケティング・ツールとして利用された。メールをマーケティングに活用する場合、マーケティング手法と共に、配信速度のようにメール配信処理のパフォーマンスも重要となる。特に大規模メール配信を行う場合は、配信を開始してから配信し終えるまでの処理時間がサービスに大きく影響する。本文では、現在のメールを取り巻く環境と筆者らがメール配信業務を運用してきた経験を合わせ、大規模メール配信を行う際のパフォーマンス及び運用上の留意点について考察する。

### Large Scale E-Mail Delivery and Performance

YASUHIRO SENDA<sup>†</sup> and HIROAKI WATANABE<sup>†</sup>

The spread of e-mail is progressing with the spread of Internet and the e-mail function support of mobile phone is accelerating the number of e-mail's user. Now, the utility of the value of e-mail itself also changes, and it is beginning to be used as a marketing tool. When e-mail using as marketing tool, not only marketing technique but its performance (e.g. delivery speed) becomes important. Especially, the delivery process time from start to end is very important in case of large scale email delivery service. This paper discuss about the performance and operation issue in large scale e-mail delivery based on today's e-mail environment and author's experience of e-mail delivery service.

#### 1. はじめに

インターネットの普及に伴い<sup>1)</sup>、企業、個人を問わず電子メール（以降、メール）の普及が急速に進み、携帯電話によるメール機能のサポートがメール利用者数の増大にさらに拍車をかけた。現在では、「メール・マーケティング」という言葉が定着し、マーケティング・ツールの1つとしてメール自体の利用価値も変化している。

メールをマーケティング・ツールとして活用しようと考えた場合、オプトインメール、OneToOneメール、HTMLメール、Webとの連携、クリックカウント等のマーケティング手法そのものも重要であるが、配信速度のようにメール配信処理のパフォーマンスも重要となる。特に大規模メール配信を行う場合は、配信を開始してからメールを配信し終えるまでの処理時間がサービスに大きく影響する。そこで、本文では、現在のメールを取り巻く環境と筆者らがメール配信業務を運用してきた経験を合わせて、大規模メール配信を行う際のパフォーマンスおよび運用上の留意点について考察する。

#### 2. SMTPの概要とメール配信の種別

メールを配信する動作は、いくつかの作業に分割できる。MTA (Mail Transfer Agent) がメールの経路を決定し、MDA (Mail Delivery Agent) がMTAに代わってそのメールを配送し、MUA (Mail User Agent) がユーザ用のインターフェースを提供する。実際には、sendmail<sup>2)</sup>のようにMTAがMDAの機能を持っている場合が多い。MUAは、メーラーやメールソフトとも呼ばれることがある。

インターネットのメールを転送するために使われた最初のプロトコルはFTP (File Transfer Protocol)<sup>3),4)</sup>だが、現在ではSMTP (Simple Mail Transfer Protocol) が広く使用されている。これまで広くSMTPの標準として利用されてきた、RFC821<sup>5)</sup> (Simple Mail Transfer Protocol) がRFC1123<sup>6)</sup> (Requirements for Internet Hosts) やSMTP拡張機構を合わせてSMTPの基本仕様として扱われてきた。

RFC821は仕様としていくつかの問題があったものの長期間改定されなかったが、長い議論の末2001年4月によりやく改定され、RFC2821<sup>7)</sup>となった。今後はこちらがSMTPの仕様として用いられることになる。

<sup>†</sup> トランス・コスモス(株)  
Trance Cosmos, Inc.

では、メール配信での簡単な SMTP コマンドのやり取りを紹介する。まず、SMTP サーバ (mail.example.com) の SMTP ポート (通常、TCP 25 番) ヘクライアントがコネクションを張りに行く。SMTP サーバからコネクション確立の応答 (220) が返ってくると、SMTP セッションを開始する準備が整ったことになる。SMTP セッションは、HELO コマンドから始まる。各コマンドは以下の順で実行する。説明を簡略化するために、細かい規約は無視する。

**HELO** 接続元のマシンのホスト名 SMTP サーバに接続しているクライアントのホスト名を引数に指定して HELO コマンドを実行する。SMTP セッションの開始である。

**MAIL FROM:** (差出人のメールアドレス) 差出人のメールアドレスを入力する。メール送信のトランザクションの開始である。

**RCPT TO:** (宛先のメールアドレス) 宛先のメールアドレスを入力する。

**DATA** DATA コマンドでは、送信するメール本文の内容を入力する。DATA コマンド入力後、本文を入力し、その最後に、"."だけの行を送ると、入力が終了すると同時にトランザクションも終了となる。

**QUIT** SMTP セッション終了とコネクション切断のために QUIT コマンドを実行する。

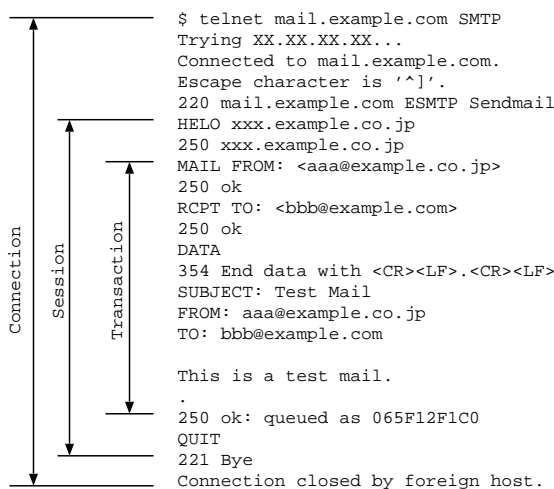


図 1 簡単な SMTP 接続の例 (1transaction, 1recipient)

図 1 は 1session, 1transaction, 1recipient の例であるが、このように HELO, MAIL FROM, RCPT TO, DATA,

ここで本文とは、メールのヘッダも含めたメッセージ全体である。メール・ヘッダとメール・ボディの間は一つの空行で区切られる。

QUIT の順に実行すると、一通のメールが一人の宛先に配信される。

## 2.1 メール配信における負荷要因

次にメール配信における負荷について考えてみよう。メール配信での負荷のうち主なものは、「ネットワーク負荷」と「サーバ処理での負荷」に分類することができる。

ネットワーク負荷を単純に考えると、その負荷はメール本文のサイズ (文字数) に比例することは容易に想像できる。パーソナルコンピュータ (PC) 向けメールと携帯向けメールでは当然、そのサイズは異なるが、ここ数年で PC 向けメールマガジンのサイズは特に大きくなってきている。その原因の一つとして HTML メールが挙げられる。HTML メールには画像データをエンコードしてメール本文に埋め込むタイプがあり、この埋め込まれた画像データがメールのサイズを極端に大きくしている。100KB を超えるメールマガジンと言うのは現実的には稀であるが、配信数が多ければそれだけネットワーク負荷が跳ね上がることは事実である。

仮に一通 100KB のメールを 1 万アドレスに配信する場合を想定しよう。前節で簡単な SMTP コマンドのやり取りを示したように、一回一回コネクションを張り、1 万アドレスに配信しようとしたら、メール本文だけでも合計 1GB のデータをネットワーク上に流すことになる。

このような実装の MTA もあるが、パフォーマンスがネットワークの負荷により頭打ちになることが容易に想像できる。また、同じドメインでの受信者が多いサイトに対しては、受信側のメールサーバにも高負荷をかけることになる。

したがって、大量のメールを配信する場合、ネットワークへの負荷および送受信サーバの負荷を下げるための対策が必要である。次節以降で、「一斉同報配信」「OneToOne 配信」の 2 点についてその対策を紹介する。

## 2.2 一斉同報配信

通常、メール本文が同じならば、同じドメイン宛のメールはまとめて配信することができる。一般的に、この「1transaction, 複数 recipient」で送る形式を「一斉同報配信」と呼んでいる。例で示す通り、一回の transaction で複数の RCPT TO コマンドを乗せるのである。ただし、SPAM メール対策ということで、1transaction に乗せられる recipient 数に制限をかけているサーバが殆どである。

筆者らも実際に一通 100KB を超えるメールマガジンを受け取ったことがある。

不特定多数に送信される広告や、詐欺まがいの情報、チェーンメールなど、迷惑とされる電子メール一般を指し、UBE (Unsolicited Bulk Email) や UCE (Unsolicited Commercial Email) などとも呼ばれている。

るため、recipient 数には注意が必要である。もし、1transaction に制限以上の recipient 数を乗せるとエラーで返される。また、recipient 数の制限はサーバにより異なり公開もしていない。一斉同報配信の場合、recipient 数の制限値を見つけることもパフォーマンスを上げることの一つである。ただし、いつ制限値を変更されるかわからないので、受信側サーバの設定変更に敏感である必要がある。

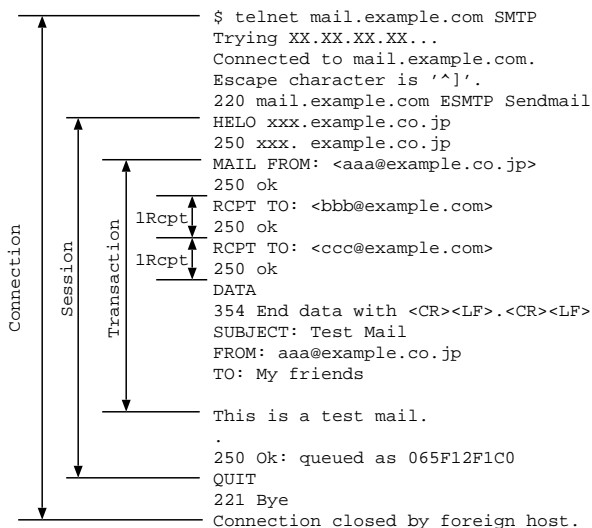


図 2 1transaction, 複数 recipient の例

### 2.3 OneToOne 配信

一斉同報メール配信は、メール配信の負荷を下げるために効率的な手段であるが、1transaction, 1recipient でしか配信できないケースもある。

例えば、メール本文中に名前やユーザ ID 等の個人を特定するための情報をメール本文に挿入するメールマガジンでは、配信先アドレス毎にメール本文の内容が異なるため、1recipient ずつ配信することになる。このような配信形式を「OneToOne 配信」と呼ぶ。

では、1transaction, 1recipient で配信する限り、配信のパフォーマンスを上げる方法はないのであろうか。OneToOne 配信の場合、メール本文のサイズにより、一斉同報配信に比べて配信速度は極端に落ちてしまうが、「RSET」コマンドを使用することで工夫はできる。RSET コマンドは、SMTP セッションをリセットするもので、RSET コマンドを発行すると HELO コマンドを発行した後の状態になる。

このため、RSET コマンドを利用すると、MAIL FROM コマンドから DATA コマンドの transaction を繰り返すことにより、1session でメール本文の内容が異なるメールを

複数アドレスに配信することが可能となる。この方法でのポイントは、メールサーバとのコネクション処理を削減できることにある。TCP コネクションを確立するサーバ負荷は、決して無視できない。携帯メール向け配信が遅延する理由の一番大きな問題は、メールサーバの負荷が高すぎて、メールサーバとコネクションが確立できない為である。一度掴んだコネクションは、最後まで放さない事が重要なのである。

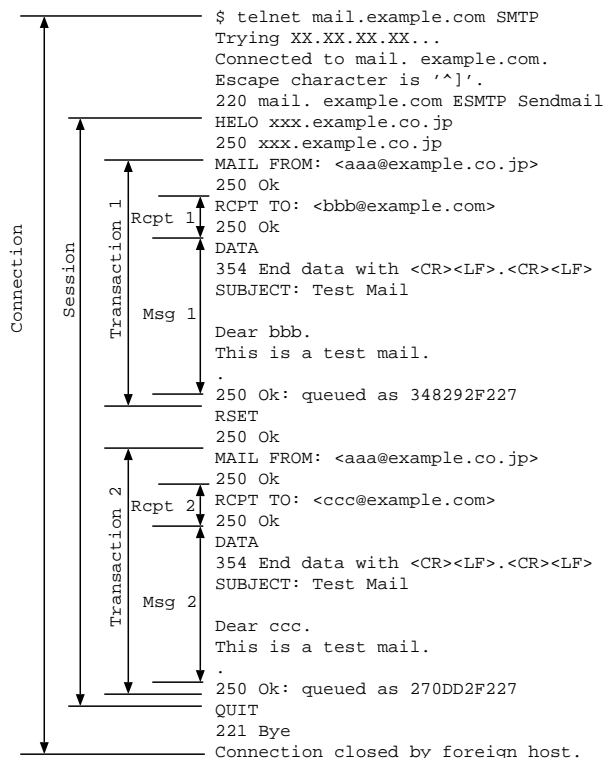


図 3 1session, 複数 transaction の例

## 3. メールを取り巻く環境と配信速度

単位時間当たりの配信数を増やす最も単純な方法は、配信プログラム（プロセス）を並列実行することである。殆どの大規模メール配信アプリケーションサービスプロバイダ (ASP) では、複数台の配信サーバを有し、配信プログラムも並列処理させている。しかし、携帯メールや無料で取得できるメールアドレス（以降、無料メール）が普及してからは、配信サーバの増強だけでは、期待したパフォーマンスが得られなくなってきているばかりか様々な問題が

ただし、現在、多くの MTA では、このような実装をしていないので、RSET を利用した OneToOne 配信を実現するためには、配信処理を独自で実装する必要がある。

表面化している。

本節では、これらの問題を紹介すると共に、その対策を紹介する。

### 3.1 配信先ドメインの偏り

携帯メールや無料メールが普及する以前では、配信先のドメインが適度に分散されていた。10万通のメールマガジンの配信を行うと、同一ドメインに対する配信は、多くて2,000通から3,000通であった。ところが、携帯向けメールマガジンでは、10万通の内5万通が同一ドメインという場合がある。当然、同一ドメインへ配信が増えれば、配信速度は配信先である受信サーバの処理能力に依存することになる。

配信先ドメインの偏りによる配信処理への影響を整理するために、同一ドメイン（メールサーバ）に対し1sessionで1transaction, 1recipientの配信を1万アドレスに対して行うことを想定してみる。1transaction, 1recipientでしか配信できないMTAでも、1万アドレスに配信するのであれば、1万プロセスにて並列に配信すれば短時間で配信できると思うかも知れない。

しかし、この方法は現実的にはうまくいかない。配信サーバ側で1万のプロセスを同時並行というのは無理がある。また、メールを受ける側も、同じ配信元から同時に1万コネクションも張られたのでは、サーバ自体が過負荷になり、別の配信元からのサービスに影響が出る。しかも、高負荷によりサーバがダウンする恐れがある。

このような状況に対処するため、プロバイダでは、同じ配信元から一定以上のコネクション要求が来た場合、アクセスを拒否するように設定している。同じ配信元の特長は、配信元のドメインかIPアドレスで行う。コネクションの確立を拒否されたメール配信は再送処理にまわされるが、並列度に頼った配信方法では、再送時にもコネクションの確立ができず、最終的にエラーメールとなる確率が高い。特に携帯キャリアでは、同じ配信元からのコネクション数を厳しく制限する傾向が強いため、このような事情を踏まえると、一度掴んだコネクションは最後まで放さない事の重要度がご理解頂けると思う。

### 3.2 エラー処理

メール配信エラーは、大きく分けて2種類である。3桁の数字で表されるSMTP応答コードの最初の数字が4から始まる「一時的エラー」と5から始まる「致命的エラー」である。

一時的エラーとは、受信側メールサーバから応答がなかった(421)、受信側メールサーバのディスクに空き容量が不足している(452)など通信処理やメールサーバ側に問題があった場合であり、一時的エラーとなったメールは再送処理にまわされる。一方、致命的エラーとは、配信先

のメールボックスがない(550)、配信先のメールサーバのディスクが満杯だった(552)などで、致命的エラーでは再送されることはない。

大規模メール配信における致命的エラーの殆どは、配信先のメールボックスがない(550)である。無料のメールアドレスや携帯メールアドレスは、アドレスの有効期間が短いため、大規模メール配信において大量の550エラーが発生する可能性がある。

SMTP応答コードが”550”になったメールアドレスに対し、配信を続けたらどうなるであろうか。メールボックスの存在しないアドレス宛てに大量のメール配信を行った場合、企業によってはSPAMメールと判断し、その配信元からのメールを受け付けなくする処理をしている。エラーアドレスを大量に含んだ配信を意図せず行ったにせよ、相手サーバから拒否された場合、正常なアドレスまで配信されないことになる。その結果、配信のパフォーマンスが極端に低下する。

このような事態に対処するため、大抵の大規模メール配信ASPでは、致命的エラーとなったメールアドレスを記録し、次の配信時には配信対象からメールアドレスを削除するクリーニング機能を持っている。しかし、この機能を有効にするかどうかの判断は、ASPを利用するユーザ次第であるため、アドレスクリーニングをしていない配信があると、他の配信に影響を与えかねない。また、メールを受信する側のプロバイダが、あるメールをSPAMメールと判断した場合、正常なメールアドレスでも550を返す場合がある(と予想している)。

この状況になると、アドレスクリーニング機能は正常なアドレスまで削除してしまう。つまり、配信パフォーマンスを上げるのは、ユーザの使用方法にも関係しているのである。

### 3.3 再送処理

大規模メール配信における再送処理の考え方を一変したのは、携帯向けメール配信である。大規模メール配信ASPでは、再送処理の為に専用サーバを用意するケースが多い。その理由は、通常の配信と再送処理の負荷を分散するため、再送専用サーバが1、2台あれば問題はなかった。

しかし、現状の携帯向けメール配信では、配信アドレス数の90%以上が再送処理にまわされており、再送処理の負荷を無視できなくなってきている。配信サーバと再送サーバのバランスが悪いシステム構成の場合、多重配信が起こる可能性がある。

仮に、10台の配信サーバと1台の再送サーバという構成で、10万通の携帯向けメールマガジンを配信したとする。10台の配信サーバで10万通を均等に按分したとして、1台の配信サーバで1万通の配信を受け持つことになる。

現状を踏まえると、各配信サーバにて一回目で配信できるのは1,000通程度であるため、約9,000通は再送サーバに送られる。再送サーバは1台なので、配信サーバ10台の分の再送要求メール約9万通を受け取ることになる。仮に9万通のメールが1台の再送サーバへ送られた場合、サーバダウンはしないまでも、各配信サーバからの再送要求メールを処理できないと予想される。再送サーバで受け付けられなかった再送要求は、再送サーバに対して再び再送処理を行うという悪循環が発生する(図4)。

この悪循環は、多重配信を引き起こす可能性がある。多重配信の起こる過程を見てみよう。配信サーバは、再送サーバに対してメール本文の送信(DATAコマンド終了)まで終了しているとする。しかし、再送サーバは高負荷であったため、DATAコマンドの受信通知を返信する過程で、応答を待っている配信サーバ側のセッションでタイムアウトが発生したとする。再送サーバでは、メール本文まで受け取っているため、受け取ったメールの再送処理を始める。

一方、配信サーバではタイムアウトが発生したので、再び再送要求をすることになる。この状況は、再送サーバの負荷が下がるまで発生することになるため、最悪の場合一人に何十通もの同じメールが配信されることになる。また、上記で示したような受信サーバの高負荷が引き起こす多重配信の仕組みは、再送処理の過程だけではなく、通常の配信処理でも起こりうることに注意するべきである。このような再送処理の問題に対処するためには、全体のシステム構成を考えた上で、再送サーバを用意することが重要である。さらに、DATAコマンドのタイムアウトが引き起こす多重配信問題については、各SMTPコマンドのタイムアウト時間を調節することも検討するべきである。

### 3.4 夜間配信問題

夜間配信問題が話題に上がってきたのは、ここ1、2年前のことである。携帯メール利用者から、深夜にメールが届き、着信音のため睡眠を妨害されるというクレームが相次いだ。既存のメールユーザからすれば時間を問わず送れることがメールの常識であるため意外なことかもしれないが、このように常に携帯電話を手元に置いている携帯メールユーザは、これまでの常識で考えることはできないのである。

大規模メール配信ASPでは、メールの配信開始時刻を指定する機能がある。では、ASP利用者が、わざわざ夜間に配信を開始するように時刻を設定しているのだろうか。翌朝一番に読んで欲しいメールマガジンでは、ASP利用者が明け方に配信を開始するように時刻を設定する場合

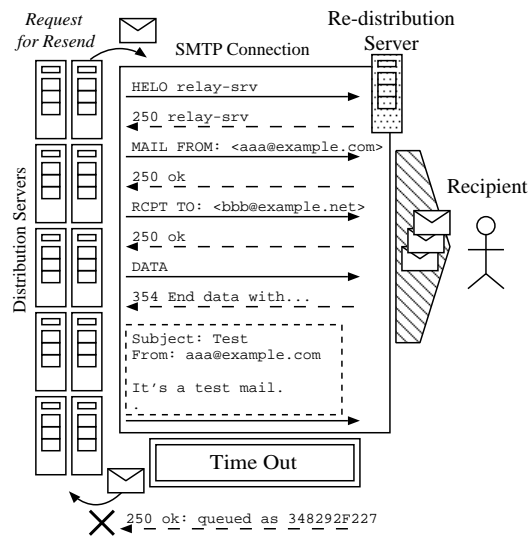


図4 再送処理での多重配信

はあるが、それもPC向け配信の話である。

では何故、携帯向け配信で夜間配信が起こるのだろうか。原因は、受信サーバの高負荷にある。メール配信の際、受信サーバが高負荷であると、コネクションを確立できない場合があり、再送処理を行うことになる。しかし、日中は再送処理を繰り返してもコネクションが確立できず、比較的負荷の低い深夜になってから、配信されてしまうのである。

夜間配信問題を難しくしているのは、大規模メール配信業者も受信サーバ側も正常な配信処理をしているということである。しかし、受信者や受信側のプロバイダに対し、状況を説明しても理解してもらえないとは限らない。やはり、何らかの対策が必要である。夜間配信問題の対策の1つに、配信時間帯を制限することである。

例えば、09:00から21:00を配信対処時間帯とし、21:00以降は配信しないように設定する。配信には、もちろん再送処理も含む。むしろ、再送処理こそ、この問題の主役である。再送処理の節では、全体のシステム構成を考えた上で、再送サーバを用意することが重要であると述べたが、配信時間帯制限機能を実装した場合、再送処理の比重はより重くなる。配信対象時間でないメールは、溜まる一方であるからだ。

我々がサービスを行ってきた大規模配信ASPにて、配信時間帯制限機能を実装した際、再送サーバの負荷を軽く見積もりすぎて、再送サーバがダウンしたことがある。メールサーバの復旧は、未配信、多重配信が無いように、何処まで配信が終了しているかを調査するため、復旧作業に時間が掛かり、再送サーバをもう一台導入するよりも人件費が高かった。この経験から、配信時刻制限機能を実

しかも、ユーザ数を考えれば日本国内では携帯メールユーザの方が主流なのである。

装する場合は、再送サーバの負荷をきちんと見積もると共に、再送処理の方法も再度検討することをお勧めする。

### 3.5 ログの取りこぼし

未配信、多重配信、ユーザからのクレーム等の配信トラブルに関して、トラブルシューティングの情報源として最終的に行き着くところは配信ログである。配信ログは、配信トラブルを解決するための生命線である。もし、配信ログに信用性が無くなったらどうなるか。配信サービスを保証するものは何もなくなるのである。大規模メール配信サーバのプラットフォームの殆どは UNIX であろう。UNIX では、独自にログ処理をしているケースを除き、配信ログは syslog に出力する。しかし、syslog に大量な書き込みを行うと、パツファが満杯であるため書き込みに失敗したという旨のエラーを返す。エラーを受け取ったプロセスは、ログ出力をリトライするが、設定したリトライ回数を超えた場合、ログ出力を諦め次の処理を始めるため、ログを出力し損ねることがある。この状況を筆者らは「ログの取りこぼし」と呼んでいる。ログの取りこぼしが起こる原因はサーバの高負荷にあるが、配信処理が集中すると高負荷になるのは必然である。このログの取りこぼし問題への対処方法は、配信サーバ台数を増やし負荷分散を図るか、syslog にログが出力できるまで処理を繰り返すかであるが、一番簡単な対処は、ログ出力のリトライ回数を十分な大きさに増やすことである。もし、ログの取りこぼしが発生しているのであれば、配信スピードを上げることも先に対処すべき問題である。

### 3.6 RFC 規約外実装メールサーバへの対処

メールサーバの実装メールに関する諸々の規約は、メール配信と SMTP の章でも述べたが、RFC で規定されている。しかし、RFC に準じない独自の拡張機能を実装しているメールサーバも少なくない。そればかりか、規約に違反した実装をしているケースもある。規約に違反した実装をしているサーバを発見した場合、サーバ管理者へ通知し、修正を依頼することになるのだが、そう簡単に修正される訳ではない。大規模なキャリアであれば尚更である。この場合に対応をするのは、むしろ大規模配信 ASP を提供する業者である。本意であるが、違反しているサーバの仕様に合わせた対応をすることになる。この規約違反に対応するケースは極端であるが、相手サーバの実装に合わせた対応は、地道だが配信パフォーマンスを上げる手段の1つ

である。

## 4. ま と め

本文では、メールを取り巻く環境と合わせて、大規模メール配信を行う際のパフォーマンス上の留意点について、メール配信の種別、配信先ドメインの偏り、携帯向け配信、エラー処理、再送処理といった観点から考察してきた。

今後は、メールの利用価値の変化と共に、提供するサービス内容や配信処理の比重も変化する。メールの新しい利用価値が発見されれば、違った視点でメール配信を捉える必要がある。大規模メール配信 ASP では、配信速度の表記がある。一時間当たり 100 万通と言った表記である。ただし、本文で考察してきた配信処理の内容までは記述されていないので、配信速度の数値だけは、メール配信全体のパフォーマンスを判定することは困難なのである。もし、大規模メール配信 ASP を利用する機会があるならば、まず配信速度の意味を確認することをお勧めする。

謝辞 本稿の作成にあたり、多大な協力をいただいた石山元浩氏に、謹んで感謝の意を表する。

## 参 考 文 献

- 1) 総務省: インターネット接続サービスの利用者数等の推移【平成15年2月末現在】(速報), [http://www.soumu.go.jp/s-news/2003/030331\\_3.html](http://www.soumu.go.jp/s-news/2003/030331_3.html) (2003).
- 2) Sendmail Inc.: `sendmail`, <http://www.sendmail.com>
- 3) J. Postel: RFC765 “File Transfer Protocol” (Obsoleted by RFC959), <http://www.ietf.org/rfc/rfc765.txt> (1980).
- 4) J. Postel, J. Reynolds: RFC959 “File Transfer Protocol (FTP)” <http://www.ietf.org/rfc/rfc959.txt> (1985).
- 5) J. Postel: RFC821 “Simple Mail Transfer Protocol” (Obsoleted by RFC2821), <http://www.ietf.org/rfc/rfc821.txt> (1982).
- 6) Robert Braden: RFC1123 “Requirements for Internet Hosts – Application and Support”, <http://www.ietf.org/rfc/rfc1123.txt> (1989).
- 7) John C. Klensin: RFC2821, “Simple Mail Transfer Protocol” <http://www.ietf.org/rfc/rfc2821.txt> (2001).

---

ちなみに、今のところ欧米では、夜間配信問題はそれ程注目されていない。しかし、日本の携帯メールのように、需要が増えれば必ず突き当たる問題である。夜間配信問題1つを取っても、携帯メールに関しては、日本が他国を圧倒していることは事実である。当然、ログを出力し損ねる過程は、MTAの実装に依存するが、大抵のケースで当てはまると考えてよい。