qmail

an internet mail transport agent

or

qmail – fast, secure, reliable. Pick any three!

Peter Samuel

Peter.Samuel@gormand.com.au

Presented to

2 September 2003, 25 September 2003

Contents

Contents

Definitions

MUA

Mail user agent. Allows an end user to read incoming mail, respond to this mail and compose new messages.

MTA

Mail transport agent. Responsible for transporting electronic mail from one location to another.

MDA

Mail delivery agent. Responsible for delivering an electronic mail message to its final destination. May be part of the same software suite as the MTA.

Message components

ENVELOPE

Name and address of both the message originator and message recipient(s). Used during the transaction between MTAs.

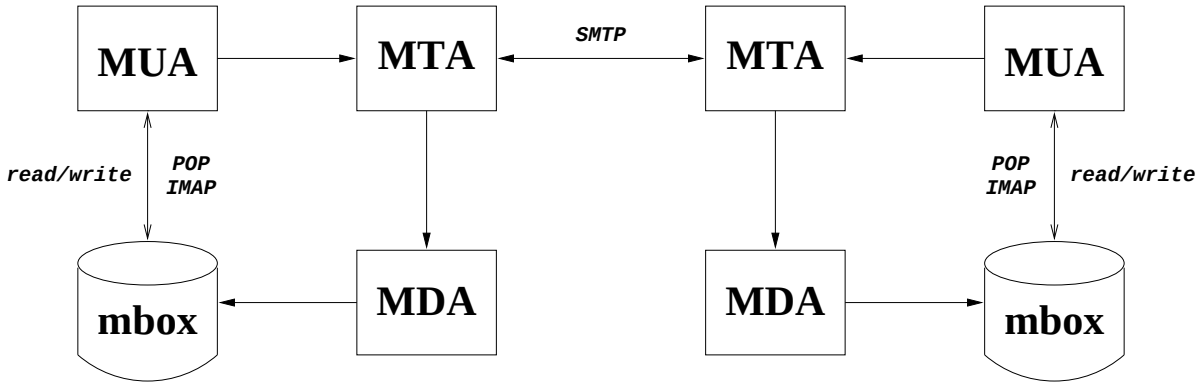
HEADERS

Name and address of both the message originator and message recipient(s) plus extra details such as the date, subject and delivery details.

BODY

Message data.

Simplified mail transaction



The user composes a message using an **MUA.**

The **MUA** gives it to the **MTA** to arrange delivery.

If the message is local, the **MTA** gives it to the local **MDA** which delivers it to the local mailbox where it is available for reading by the **MUA.**

If the message is for a remote recipient, the **MTA** gives it to a remote **MTA** which in turn gives it to that system's **MDA** which delivers it to the remote mailbox.

qmail history[1]

Dan Bernstein began developing qmail in 1995 in an attempt to avoid the security problems and other deficiencies he saw in sendmail.

The first beta version was 0.70 released in January 1996.

The first production version – 1.00 – was released in February 1997.

Version 1.03 was released in June 1998.

It should compile and run on almost any well behaved Unix system.

It is penetrating the sendmail market. Actual details are hard to determine but judging by the mailing list traffic, many people are either using qmail or want to know more.

[1]This section has been summarised from the qmail distribution files

Disadvantages of using qmail

qmail is an internet mailer. It will only talk **SMTP** to other MTAs. (It can be configured to use a more efficient protocol when talking to other qmail systems).

It will **ONLY** handle addresses which are in the form of *user@domain*. UUCP style addresses such as *domain!user* are definitely **NOT** supported. However transport through UUCP is easy to set up for both incoming and outgoing messages.

It makes every effort to conform to the existing standards and RFCs. Non conforming systems may have difficulty in communicating with qmail–smtpd. Some patches are available to alleviate this problem. See **http://www.qmail.org/.**

Header re-writing is minimal. However inserting your own rewrite program is easy.

Disadvantages of using qmail

The author is resistant to change unless it can be shown that the change is absolutely necessary. He is known as being quite arrogant and dogmatic. Unfortunately (or fortunately depending on your point of view) he's usually right. 

qmail requires a completely different mind set than sendmail. It does take some time to begin to think in a qmail fashion – especially if you are used to thinking in a sendmail fashion.

Distributing modified versions of qmail – including ports – require the author's explicit permission. Dan wants to make certain that distributions do not unintentionally compromise performance, operation and security.

Pre-compiled binary versions of qmail may be distributed provided the package conforms to the guidelines set down by the author. See **http://cr.yp.to/qmail/dist.html** for more details.

qmail security

Segmented architecture. Each stage of the message processing is handled by a different program. Each program runs under a different user ID. The qmail users don't even trust each other!

The only setuid program is qmail–queue and it runs as the user *qmailq.*

The only program run by *root* is the mail delivery agent qmail–lspawn.

qmail–lspawn **ALWAYS** changes its effective user id to the recipient **BEFORE** running any user specified program.

qmail **NEVER** delivers to a user whose UID is zero!

qmail will defer delivery if either the user's home directory or .qmail file is group and/or world writable.

qmail security

Parsing is cut down to a bare minimum. This avoids (or at the very least minimises) the risk of introducing a security hole with a configuration error.

The standard C library is avoided – particularly *stdio.*

"Keep It Simple". qmail has only one simple forwarding mechanism that handles forwarding, aliasing and mailing lists, instead of a separate mechanism for each. By default, qmail offers a single delivery mode instead of a selection.

Maildir format

Maildirs are essentially incorruptible storage areas for mail messages. They avoid the potential corruption problems of *mbox* and *mh* formats by removing the need for any file locking.

Maildir format stores each message in a separate file in a given directory:

./tmp/ – the delivery agent creates a file called ./tmp/*time*.pid.hostname where *time* is the number of seconds since midnight January 1 1970 GMT. The incoming mail message is stored in this file. If the file already exists, the delivery agent waits 2 seconds and tries again.

./new/ – successfully delivered messages are moved to ./new/*time*.pid.hostname. The modification time of the file is the delivery date.

./cur/ – messages read by a mail reader are moved to ./cur/*time*.pid.hostname.

Aliasing and .qmail files

Users control all addresses of the form *user–anything.*

Addresses which do not map to user names are handled by the special *alias* user.

Arbitrary addresses can be assigned to specific users using the qmail–users mechanism.

Delivery instructions are handled by .qmail–*ext* files where *ext* is an arbitrary extension.

If my user name is *peters,* I can create a ~/.qmail–jokes file so that mail sent to *peters–jokes* will be handled by that file.

Wildcard extensions are handled by the .qmail–default file. If this file does not exist, the message will bounce.

Aliasing and .qmail files

.qmail files may contain any number of delivery instructions. Each line is a separate delivery instruction. An instruction can be any one of the following:

- save the current message to a local mail store. The filename begins with either a slash '/' or a dot '.' character. If the last character is a slash, then the mail store is a Maildir.
- forward the current message to another address. Prepend the address with an ampersand '&' if the address does not begin with a number or a letter.
- run a Unix command. The command begins with a pipe '|' character.
- comment – a line beginning with a hash '#' character
- a blank line. Note that the first line of the .qmail file must not be blank.

Example .qmail **file**[1]

[1] This page is intentionally left blank to illustrate the fact that a .qmail file is not mandatory.

Example .qmail–*ext* **file**

A zero byte .qmail file instructs qmail–local to deliver the message to the user's default mailbox.

–rw–r—r— 1 peters 0 Oct 29 18:26 .qmail–*ext*

Example .qmail file

```
# This file is not empty but does nothing.
# Therefore the message will be successfully
# delivered to nowhere. This is the equivalent
# of delivering the message to /dev/null.
```

Example .qmail–jokes file

```
# Save msgs from the jokes list in a separate mbox.
# Incoming mail will be addressed to peters–jokes.
/home/peters/mail/JOKES
```

Example .qmail file

```
# My vacation .qmail file

# Completely ignore the message if the sender is
# ddts@example.com. 99 says that the message was delivered
# successfully but do not attempt further delivery
# instructions in the .qmail file. 0 says delivery
# was a success and continue processing the file.
| [ "$SENDER" = "ddts@example.com" ] && exit 99 || exit 0

# Run other msgs thru vacation. Use qmail-vacation to
# avoid problems with preline & other vacation progs.
| /usr/local/bin/vacation -j peters

# Send a copy to my ISP account. The & is only
# necessary if the address does not begin with
# an alphanumeric character such as | or /.
&peters@isp.example.com.au

# Save a local copy - most important, otherwise
# there will be no local delivery. Also save a copy
# to a maildir so that I can quickly scan individual
# messages when I return. The trailing slash
# indicates a maildir instead of an mbox.
/home/peters/Mailbox
/home/peters/Maildir/
```


qmail command exit codes

Exit codes from commands run from .qmail files are interpreted as follows:

`0`	successful delivery
`99`	successful delivery, but ignore all further delivery instructions
`100`	permanent delivery failure – hard error
`111`	temporary delivery failure, should be tried again in a little while – soft error

qmail command environment variables

Commands run from .qmail files are supplied with a number of environment variables:[1]

$SENDER	envelope sender address
$RECIPIENT	envelope recipient address
$NEWSENDER	forwarding envelope sender address, as described in dot–qmail(5)
$USER	name of the local user
$HOME	home directory of the local user
$LOCAL	local part of the recipient address
$HOST	domain part of the recipient address
$HOST2	portion of HOST preceding the last dot
$HOST3	portion of HOST preceding the second–to–last dot
$HOST4	portion of HOST preceding the third–to–last dot

[1] **WARNING:** These environment variables are not quoted. They may contain special characters. They are under the control of a possibly malicious remote user.

qmail command environment variables

$EXT	address extension
$EXT2	portion of EXT following the first dash
$EXT3	portion of EXT following the second dash
$EXT4	portion of EXT following the third dash
$DEFAULT	portion corresponding to the default part of the .qmail–… file name. **$DEFAULT** is not set if the file name does not end with default
$DTLINE	complete Delivered-To header, including newline
$RPLINE	complete Return-Path header, including newline
$UFLINE	complete UUCP style From_ line that qmail–local adds to mbox–format files

Address assignment

qmail's alias mechanism does not apply if the user exists. User names can be overridden using the qmail–users mechanism.

The file /var/qmail/users/assign contains details on who should receive mail for each given user name:

```
=local:user:uid:gid:homedir:dash:ext:
+loc:user:uid:gid:homedir:dash:pre:
```

Consider the following example:

```
=ftp:alias:29990:29998:/var/qmail/alias:-:ftp:
+ftp-:alias:29990:29998:/var/qmail/alias:-ftp-::
=rufus.t.firefly:groucho:1017:27:/home/groucho:::
```

- Local mail for the *ftp* user will be handled by the *alias* user according to the instructions in ~alias/.qmail–ftp.
- Local mail for the *ftp–anything* user will be handled by the *alias* user according to the instructions in ~alias/.qmail–ftp–anything.
- Local mail for the *Rufus.T.Firefly*[1] user will be handled by user *groucho* according to the instructions in ~groucho/.qmail.

[1]qmail forces all local user names to lower case before delivery.

Address assignment

qmail–pw2u can be used to build an assign file based on a V7 format password file.

```
/var/qmail/bin/qmail-pw2u < /etc/passwd \
    > /var/qmail/users/assign
```

/var/qmail/users/assign must not contain NULL characters, comments or blank lines. It must also end with a single dot on a line by itself

```
...
=ftp:alias:29990:29998:/var/qmail/alias:-:ftp:
+ftp-:alias:29990:29998:/var/qmail/alias:-ftp-::
=rufus.t.firefly:groucho:1017:27:/home/groucho:::
.
```

qmail–newu **must** be run to create /var/qmail/users/cdb which is used by qmail–lspawn.

Virtual domains

/var/qmail/control/virtualdomains

```
cardassia.empire:garok
ferengi.alliance:quark
klingon.empire:worf
starfleet.fed:kirk
```

Mail for *user@klingon.empire* will be rewritten as *worf–user@klingon.empire* and delivered locally to the user *worf* according to the instructions in ~worf/.qmail–*user* or ~worf/.qmail–default.

Virtual domains

Virtual domains can also be specified for user addresses as well.

```
postmaster@clown.org:alias-clown_org
postmaster@pond.net:alias-swamp
webmaster@clown.org:alias-clown_org
clown.org:bozo
pond.net:frogman
```

This example shows that mail specifically address to the postmaster or webmaster of the virtual domains will be handled by the alias user, whereas mail for any other users in these virtual domains will be handled by the users bozo and frogman respectively.

Mandatory qmail control file

qmail requires a single control file for its operation:

/var/qmail/control/me

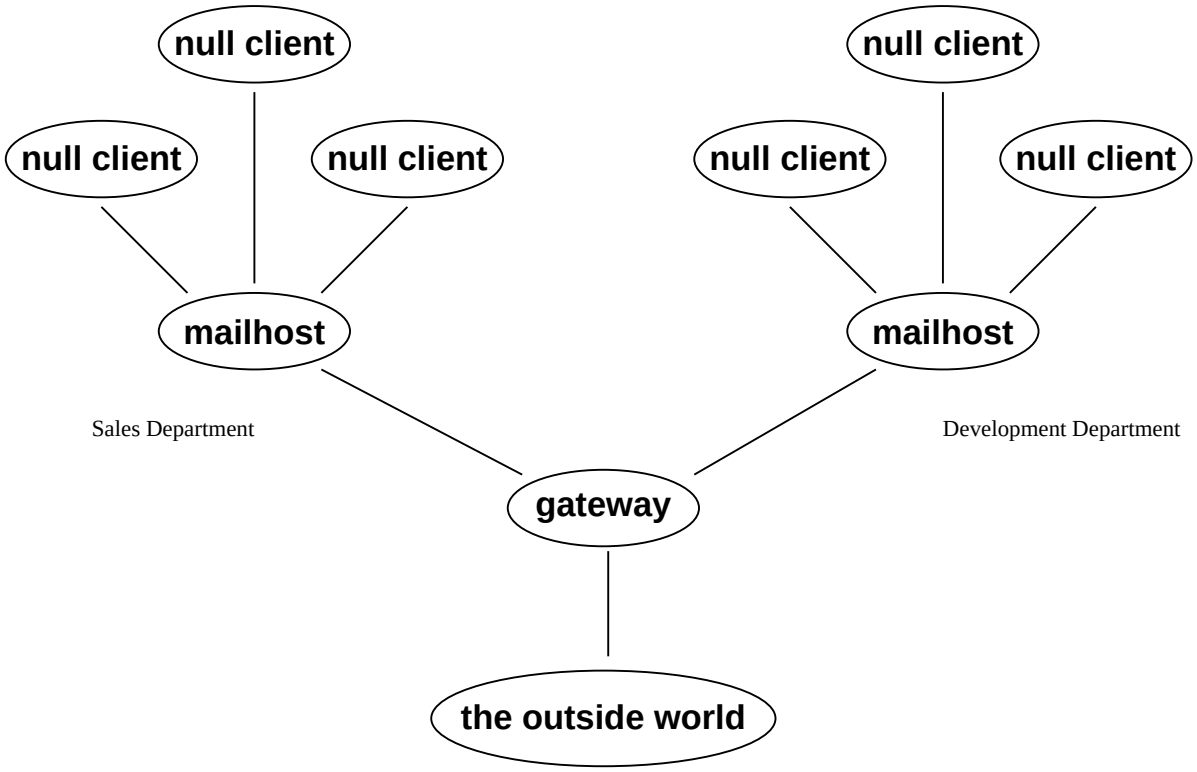
This file contains the fully qualified host name of the current host. If this file is missing, the major qmail programs, qmail–smtpd, qmail–send and qmail–remote, will refuse to run.

Optional qmail control files[1]

control file	default value	used by
badmailfrom	(none)	qmail–smtpd
bouncefrom	MAILER-DAEMON	qmail–send
bouncehost	*me*	qmail–send
concurrencylocal	10	qmail–send
concurrencyremote	20	qmail–send
defaultdomain	*me*	qmail–inject
defaulthost	*me*	qmail–inject
databytes	0	qmail–smtpd
doublebouncehost	*me*	qmail–send
doublebounceto	postmaster	qmail–send
envnoathost	*me*	qmail–send
helohost	*me*	qmail–remote
idhost	*me*	qmail–inject
localiphost	*me*	qmail–smtpd
locals	*me*	qmail–send
morercpthosts	(none)	qmail–smtpd
percenthack	(none)	qmail–send
plusdomain	*me*	qmail–inject
qmqpservers	(none)	qmail–qmqpc
queuelifetime	604800	qmail–send
rcpthosts	(none)	qmail–smtpd
smtpgreeting	*me*	qmail–smtpd
smtproutes	(none)	qmail–remote
timeoutconnect	60	qmail–remote
timeoutremote	1200	qmail–remote
timeoutsmtpd	1200	qmail–smtpd
virtualdomains	(none)	qmail–send

[1] Taken from the qmail–control manual page

Example mail network



Configuring a null client

A null client is a machine which does **NO** local mail delivery.

ALL outgoing mail is sent to a single mailhost.

/var/qmail/control/locals – an empty file

/var/qmail/control/virtualdomains – does not exist

/var/qmail/control/smtproutes

```
:mailhost.devel.example.com
```

If you can't resolve host names you can use the IP address instead.

/var/qmail/control/smtproutes

```
:[123.45.67.89]
```

Configuring a null client

If you are using qmail–1.03 or later you can eliminate all local mail queueing by using **QMQP** (Quick Mail Queueing Protocol) instead of **SMTP.**

On the client

/var/qmail/bin/qmail–queue –> /var/qmail/bin/qmail–qmqpc

/var/qmail/control/locals – an empty file

/var/qmail/control/virtualdomains – does not exist

/var/qmail/control/qmqpservers `123.45.67.89`

On the server

/etc/init.d/qmqpd – or an appropriate startup file

```
tcpserver -u 29991 -g 29998 -c 100 -v -R \
    -x /var/qmail/etc/qmqpd/rules.cdb    \
    0 qmqp /var/qmail/bin/qmail-qmqpd
```

/etc/services

```
qmqp   628/tcp         mail
```


Configuring a mailhost

A mailhost is a machine that will perform all the local deliveries for a site.

It accepts mail from null clients and from external sources.

It sends **ALL** outgoing mail to a single mail gateway.

/var/qmail/control/rcpthosts and /var/qmail/control/locals

```
mailhost.devel.example.com
devel.example.com
```

/var/qmail/control/smtproutes

```
:gateway.example.com
```

/service/qmail/run – or an appropriate startup file

```
exec tcpserver -u 29991 -g 29998 -c 100 -v -R   \
    -x /var/qmail/etc/smtpd/rules.cdb           \
    0 smtp /var/qmail/bin/qmail-smtpd 2>&1
```

Configuring a gateway

A gateway is a machine that sends incoming mail to the relevant internal mailhost.

It accepts mail from mail hosts and from external sources.

It sends **ALL** outgoing mail to the relevant MX host.

Only accept mail for the corporate domain – don't accept departmental domains. Mail for users in each department is redirected by appropriate .qmail files.

/var/qmail/control/rcpthosts and /var/qmail/control/locals

```
gateway.example.com
example.com
```

/var/qmail/control/smtproutes

```
devel.example.com:mailhost.devel.example.com
sales.example.com:mailhost.sales.example.com
```

Configuring a gateway

/service/qmail/run – or an appropriate startup file

```
exec tcpserver -u 29991 -g 29998 -c 100 -v -R   \
    -x /var/qmail/etc/smtpd/rules.cdb           \
    0 smtp /var/qmail/bin/qmail-smtpd 2>&1
```

Host masquerading

"How do I set up host masquerading?"

/var/qmail/control/defaultdomain

```
devel.example.com
```

Unqualified `From:` addresses will now have `devel.example.com` appended.

Users can also modify their `From:` headers by using environment variables.

```
export QMAILHOST=admin.example.com
export QMAILNAME="Peter Samuel"
export QMAILUSER=Peter.Samuel
export QMAILINJECT=f
```

This becomes:

```
From: Peter Samuel <Peter.Samuel@admin.example.com>
```

Handling SPAM[1]

SPAM is another name for Unsolicited Commercial Email or UCE. qmail has a number of coarse mechanisms for limiting the amount of SPAM received.

The default configuration does not support the *"percent hack"* address convention. An address of the form

user%domain1@domain2

will **NOT** be forwarded to *domain1*. The *"percent hack"* can be enabled for given domains using a control file.

qmail can be configured to prevent mail relaying by specifying valid incoming domains in /var/qmail/control/rcpthosts. This can be overridden by conditionally setting the **$RELAYCLIENT** environment variable for each invocation of qmail–smtpd using tcpserver or tcpd.

Mail from individual users or complete domains can be banned by specifying the address or domain in /var/qmail/control/badmailfrom.

[1]SPAM is a social problem. There is no completely technical solution to prevent SPAM.

Handling SPAM

Specific IP address connection control can be achieved using tcpserver, part of Dan's ucspi–tcp package (**http://cr.yp.to/ucspi–tcp.html).**

As part of the ucspi–tcp package, Dan has released rblsmtpd which blocks spam according to information provided by a number of real time black list sites. See **http://cr.yp.to/ucspi–tcp/rblsmtpd.html** for details on rblsmtpd.

There are a number of anti-spam patches for qmail–smtpd available from **http://www.qmail.org/**

If SPAM control is a serious issue, you may be better off using a different **SMTP** daemon such as mailfront from Bruce Guenter **http://untroubled.org/mailfront/** or qpsmtpd **http://develooper.com/code/qpsmtpd/.**

Virus scanning

Virus scanning can be performed by either wrapping qmail–queue, or by using an MDA which supports third party extensions such as procmail.

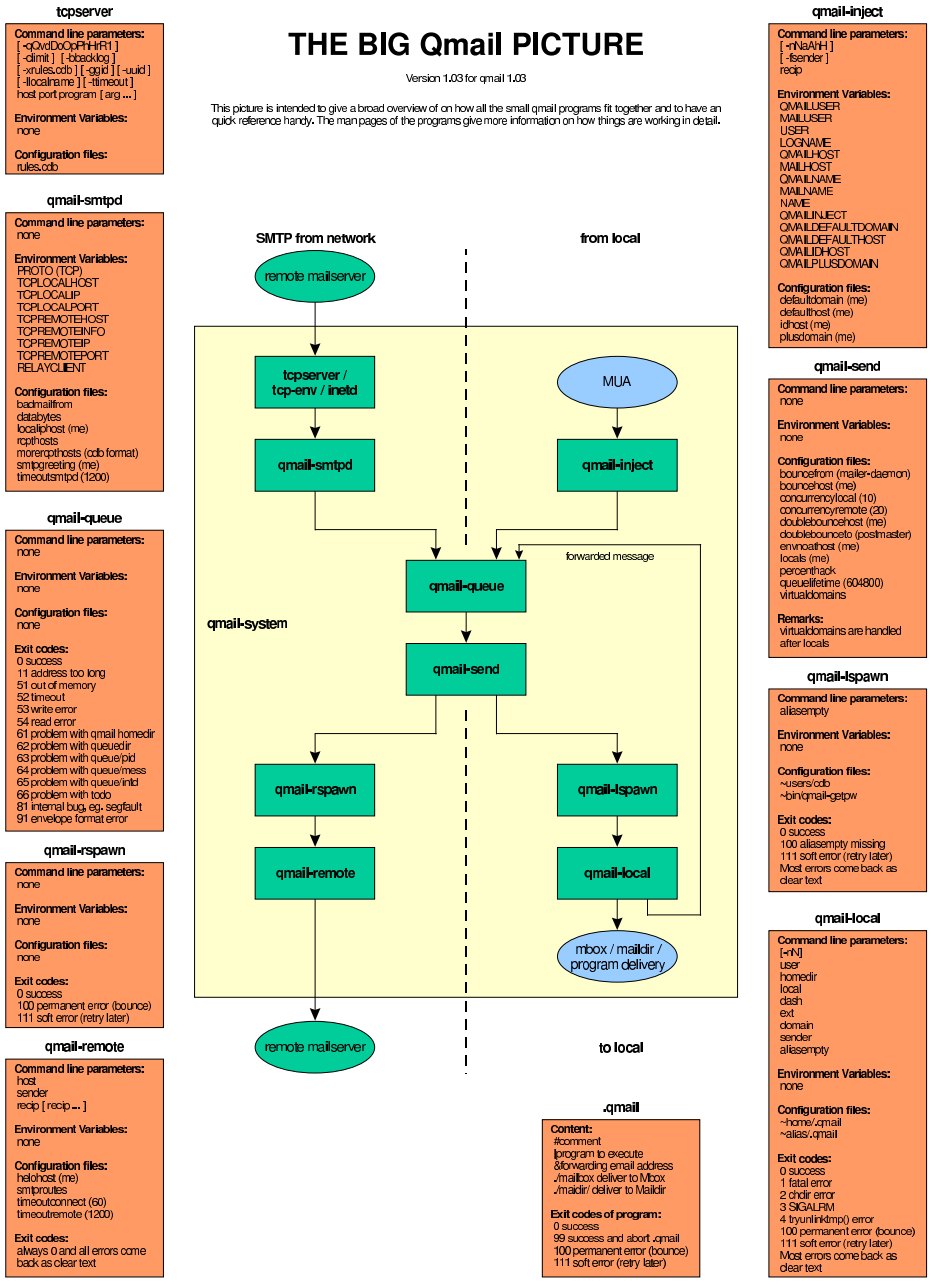
Jason Haar's qmail–scanner package wraps qmail–queue and then calls a third party virus scanner[1] such as McAfee's or AVP's products. Suspect messages are quarantined and the quarantine officer is notified. See **http://qmail–scanner.sourceforge.net/.**

The current version of AMaViS has support for qmail. See **http://www.amavis.org/.**

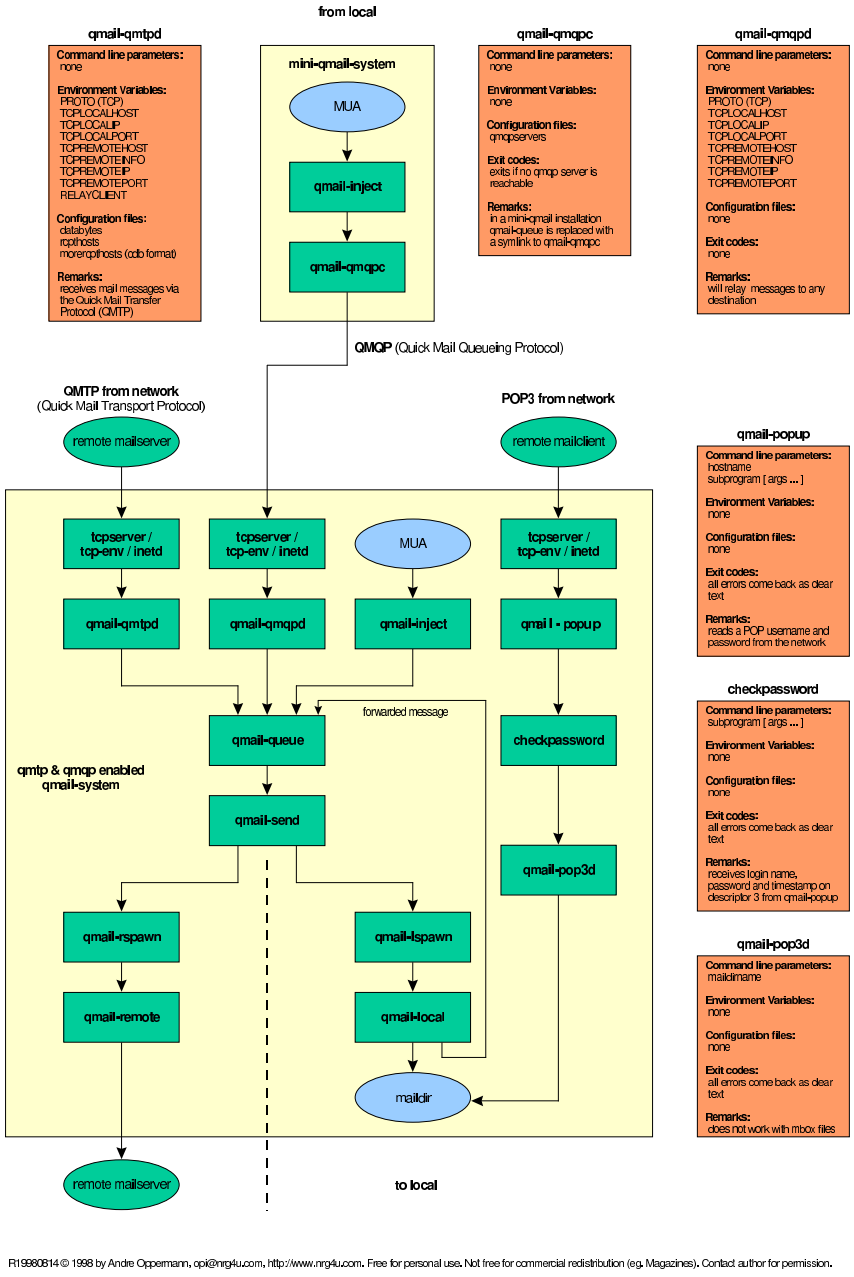
Kaspersky Labs – **http://www.avp.ru/** – has provided qmail support for AVP's virus scanner – **http://www.avp.com/.**

[1] Ensure that the vendor has a version that will run on your particular hardware platform.

The big qmail picture



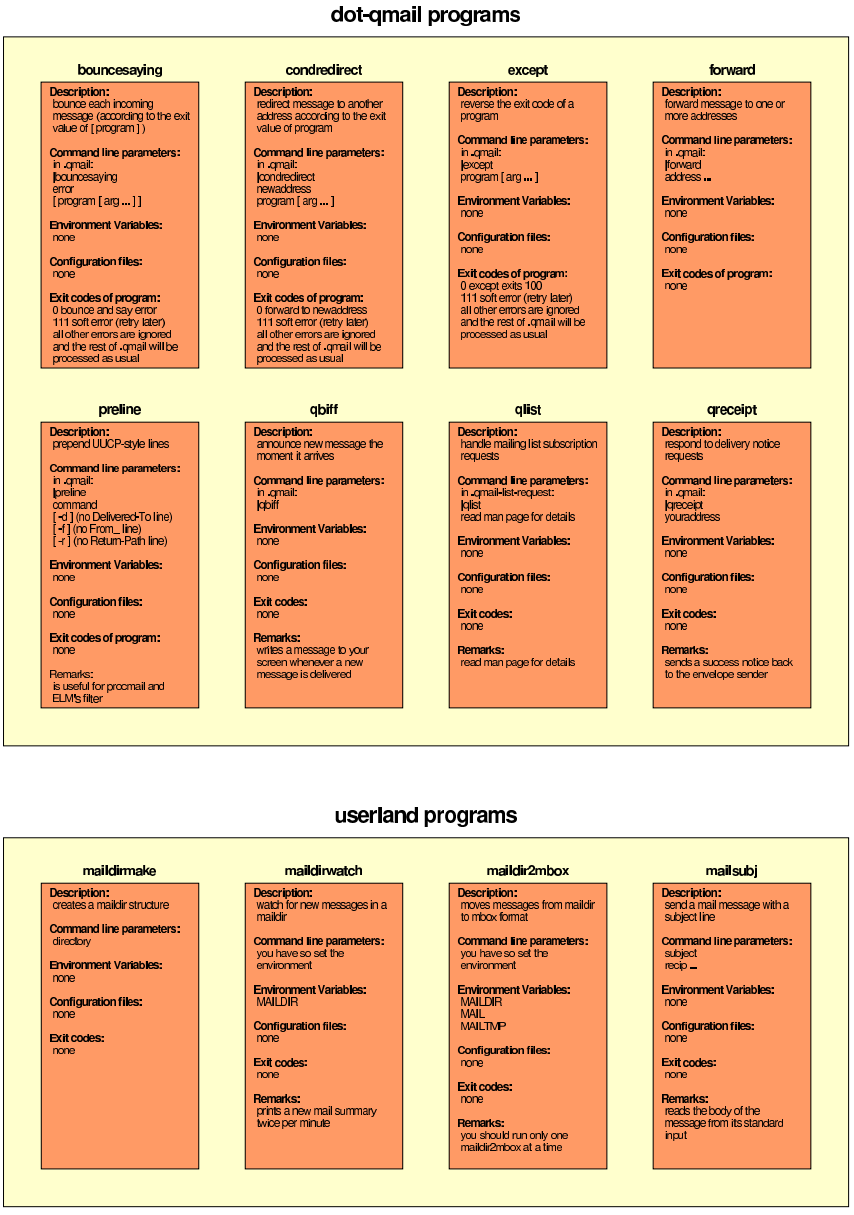
The big qmail picture



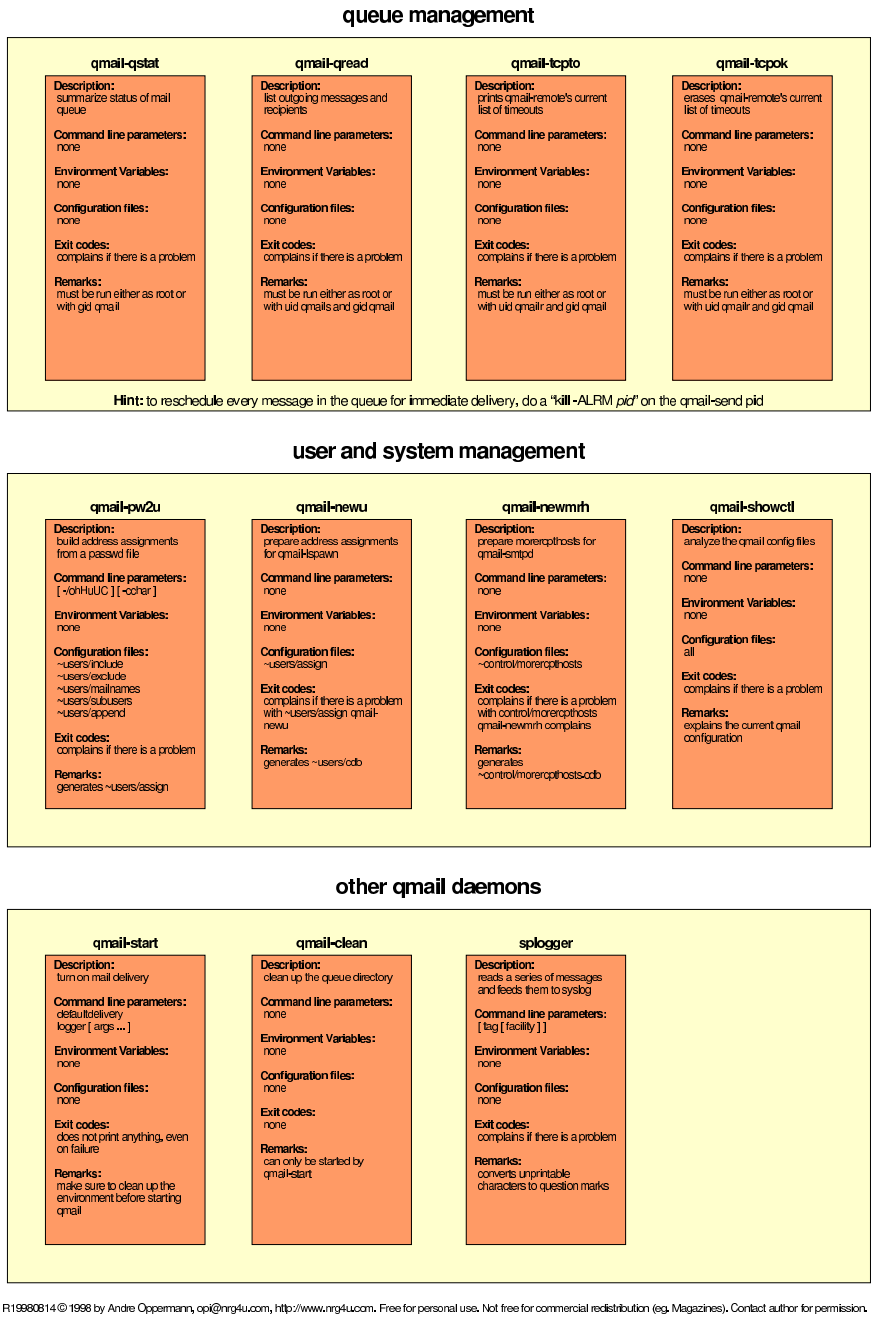
R19980814 © 1998 by Andre Oppermann, opi@nrg4u.com, http://www.nrg4u.com. Free for personal use. Not free for commercial redistribution (eg. Magazines). Contact author for permission.

The author's version of this diagram is missing the qmail–popup box. It has been fixed in this version.

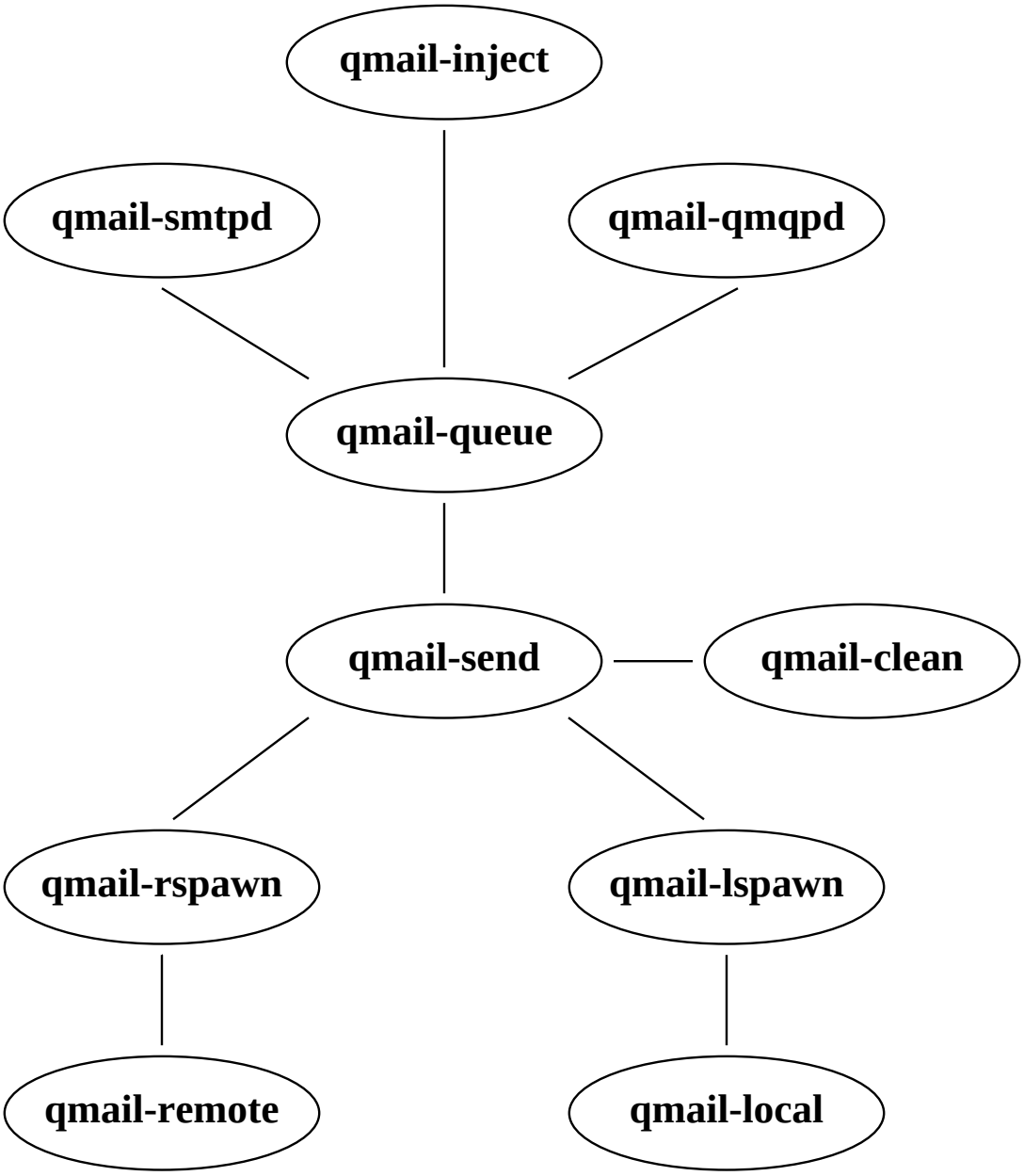
The big qmail picture



The big qmail picture



How qmail works



How qmail works[1]

Incoming mail arrives via either an **SMTP** connection to qmail–smtpd, or a **QMQP** connection to qmail–qmqpd, or via indirect queue injection from qmail–inject.

qmail–queue is called to place the message in the queue.

qmail–queue splits the message into three files, the message itself and two copies of the envelope details. It then signals qmail–send to begin delivery.

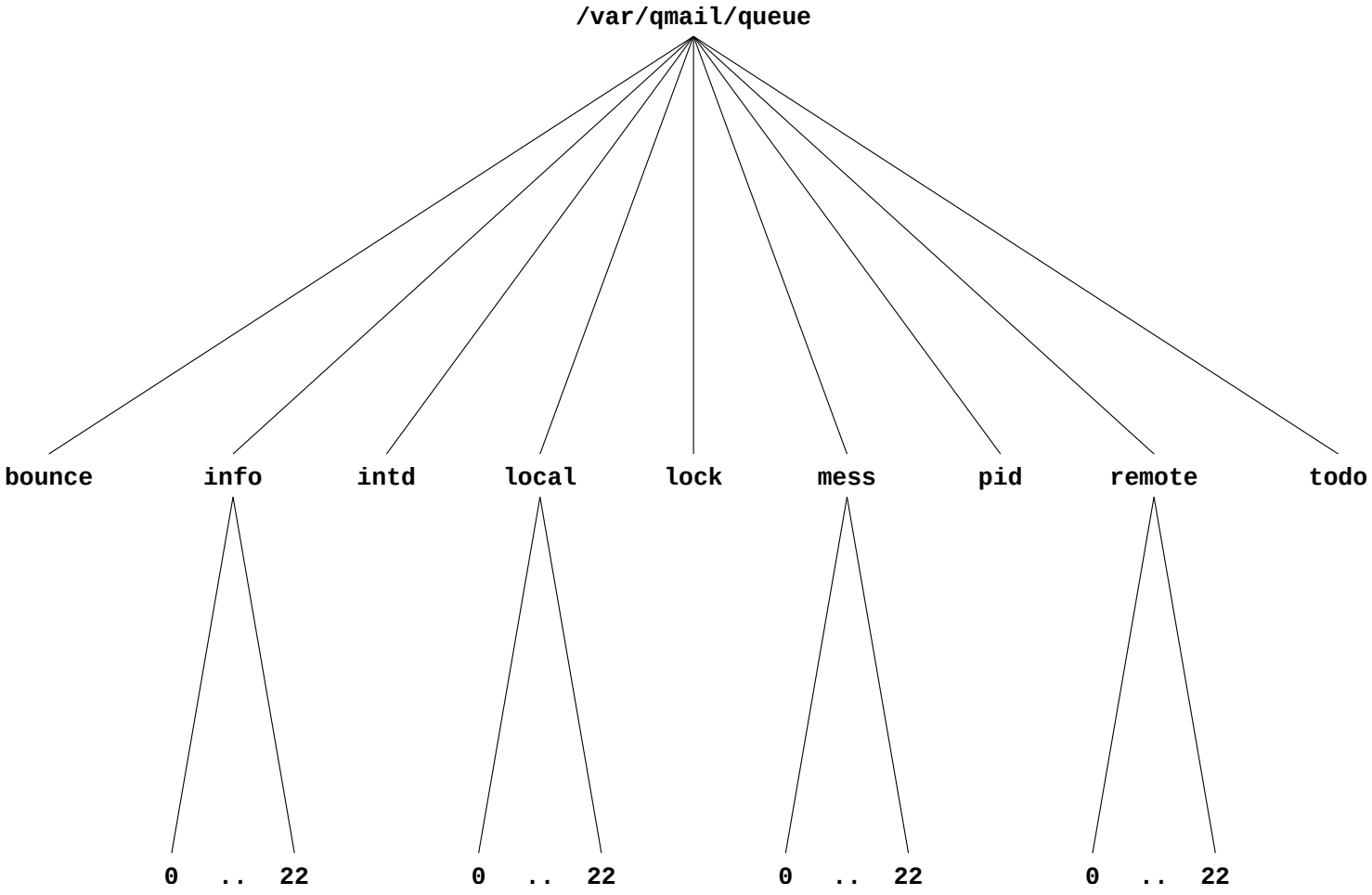
If the message is for a local recipient, qmail–send tells qmail–lspawn to deliver the message. qmail–lspawn calls qmail–local to deliver the message. qmail–local handles alias expansion and, if necessary, calls qmail–queue to place a new message in the queue. qmail–local finally delivers the message to the user's home directory.

If the message is for a remote recipient, qmail–send tells qmail–rspawn to deliver the message. qmail–rspawn calls qmail–remote to deliver the message to a remote MTA.

---

[1]For a detailed explanation of qmail internals, see the INTERNALS file distributed with the qmail source code.

Queue Structure



Message processing

qmail–queue processes incoming messages as follows:

/var/qmail/queue/pid/*tempfile*

Temporary file with unique name based on process id and system time.

/var/qmail/queue/mess/0..22/*nnnnn*

Hard link to /var/qmail/queue/pid/*tempfile*. Filename is the same as its inode. Contains message headers and body. /var/qmail/queue/pid/*tempfile* is removed.

/var/qmail/queue/intd/*nnnnn*

Message envelope in creation.

/var/qmail/queue/todo/*nnnnn*

Hard link to /var/qmail/queue/intd/*nnnnn*. Contains completed message envelope. /var/qmail/queue/intd/*nnnnn* is removed.

/var/qmail/queue/lock/trigger

Named pipe for communication with qmail–send.

The message has now entered the queue.

Message processing

qmail–send processes queued messages as follows:

/var/qmail/queue/info/0..22/*nnnnn*
: Envelope `From:` address. Created from /var/qmail/queue/todo/*nnnnn*.

/var/qmail/queue/local/0..22/*nnnnn*
: Envelope `To:` address if recipient is local. Created from /var/qmail/queue/todo/*nnnnn*. This file will not exist if there are no local recipients.

/var/qmail/queue/remote/0..22/*nnnnn*
: Envelope `To:` address if recipient is remote. Created from /var/qmail/queue/todo/*nnnnn*. This file will not exist if there are no remote recipients.

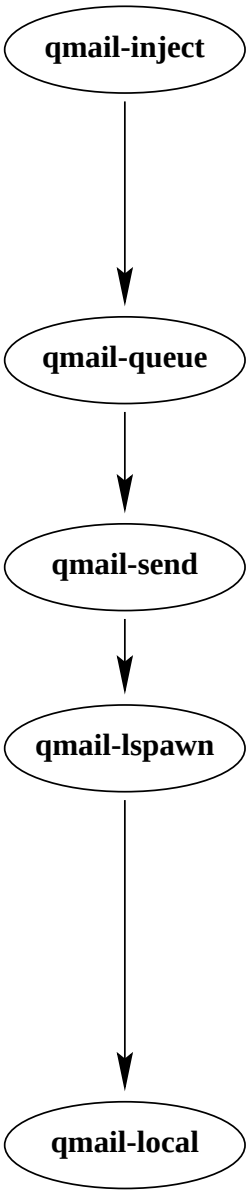
After creating the info file and the optional local and remote files, the message is considered to have been pre-processed and is now ready for delivery.

Message processing

qmail–send delivers pre-processed messages as follows:

- Each address listed in the /var/qmail/queue/local/0..22/*nnnnn* file causes qmail–send to notify qmail–lspawn to delivery the message locally. Each successful delivery causes the address in the /var/qmail/queue/local/0..22/*nnnnn* file to be marked as done.
- Each address listed in the /var/qmail/queue/remote/0..22/*nnnnn* file causes qmail–send to notify qmail–rspawn to delivery the message remotely. Each successful delivery causes the address in the /var/qmail/queue/remote/0..22/*nnnnn* file to be marked as done.
- If a delivery (local or remote) has a permanent failure, /var/qmail/queue/bounce/*nnnnn* is created and/or appended to. When all deliveries are finished a bounce message is created by combining the details from this file with /var/qmail/queue/mess/0..22/*nnnnn* and injected into the queue.

Local to local delivery[1]



Original message:

To: fred
Hi.

Fill in the complete envelope and header:

**(envelope) from joe@heaven.af.mil **
to fred@heaven.af.mil
From: joe@heaven.af.mil
To: fred@heaven.af.mil

Hi.

Store message safely on disk.
Trigger qmail-send.

Look at envelope recipient, fred@heaven.af.mil.
Is heaven.af.mil in locals? Yes.
Deliver locally to fred@heaven.af.mil.

qmail-lspawn ./Mailbox

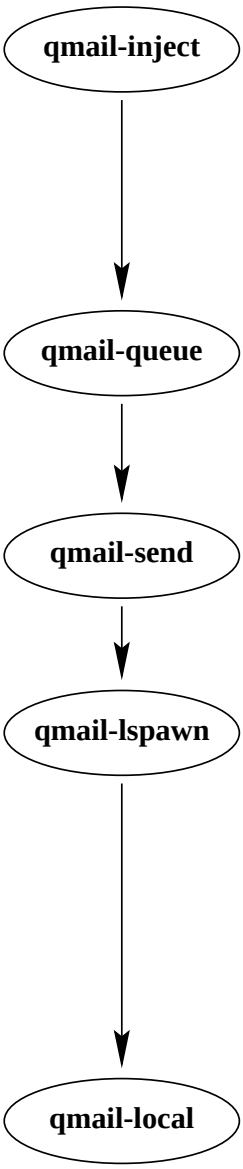
Look at mailbox name, fred.
Is fred listed in qmail-users? No.
Is there a fred account? Yes.
Is fred's uid nonzero? Yes.
Is ~fred visible to the qmailp user? Yes.
Is ~fred owned by fred? Yes.
Give control of the message to fred.
Run qmail-local.

**qmail-local fred ~fred fred '' '' heaven.af.mil **
joe@heaven.af.mil ./Mailbox

Does ~fred/.qmail exist? No.
Write message to ./Mailbox in mbox format.

[1]Taken from the `PIC.local2local` file distributed with the qmail source code.

Local to local extension delivery[1]



Original message:

To: fred-sos
Hi.

Fill in the complete envelope and header:

**(envelope) from joe@heaven.af.mil **
to fred-sos@heaven.af.mil
From: joe@heaven.af.mil
To: fred-sos@heaven.af.mil

Hi.

Store message safely on disk.
Trigger qmail-send.

Look at envelope recipient,
fred-sos@heaven.af.mil.
Is heaven.af.mil in locals? Yes.
Deliver locally to fred-sos@heaven.af.mil.

qmail-lspawn ./Mailbox

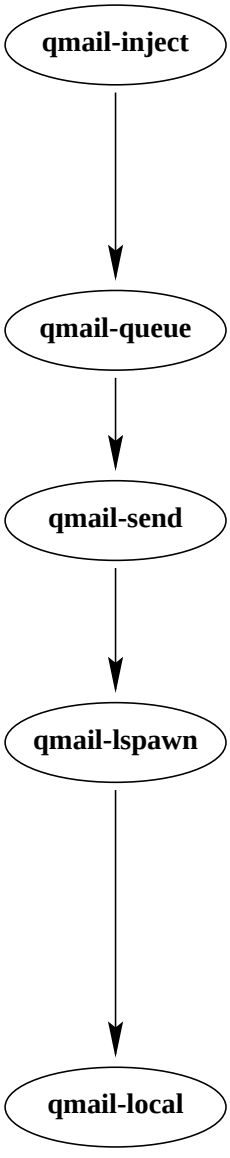
Look at mailbox name, fred-sos.
Is fred-sos listed in qmail-users? No.
Is there a fred-sos account? No.
Is there a fred account?Yes.
Is fred's uid nonzero? Yes.
Is ~fred visible to the qmailp user? Yes.
Is ~fred owned by fred? Yes.
Give control of the message to fred.
Run qmail-local.

**qmail-local fred ~fred fred-sos - sos heaven.af.mil **
joe@heaven.af.mil ./Mailbox

Does ~fred/.qmail-sos exist? Yes. "./Extramail".
Write message to ./Extramail in mbox format.

[1]Taken from the `PIC.local2ext` file distributed with the qmail source code.

Local to virtual delivery[1]



Original message:

To: dude@tommy.gov
Hi.

Fill in the complete envelope and header:

**(envelope) from joe@heaven.af.mil **
to dude@tommy.gov
From: joe@heaven.af.mil
To: dude@tommy.gov

Hi.

Store message safely on disk.
Trigger qmail-send.

Look at envelope recipient, dude@tommy.gov.
Is tommy.gov in locals? No.
Is dude@tommy.gov in virtualdomains? No.
Is tommy.gov in virtualdomains? Yes:
"tommy.gov:fred"
Deliver locally to fred-dude@tommy.gov.

qmail-lspawn ./Mailbox

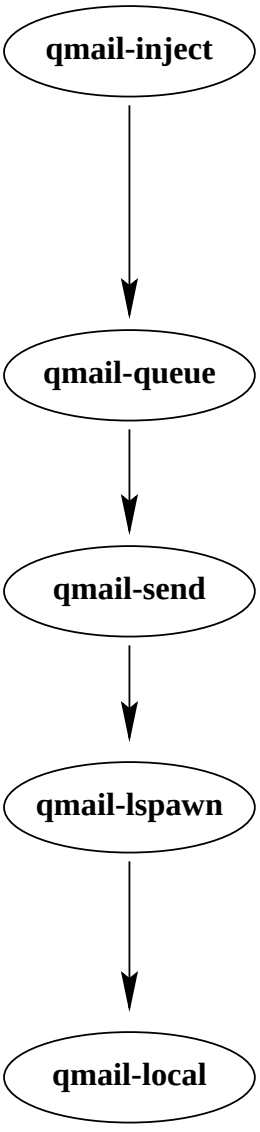
Look at mailbox name, fred-dude.
Is fred-dude listed in qmail-users? No.
Is there a fred-dude account? No.
Is there a fred account?Yes.
Is fred's uid nonzero? Yes.
Is ~fred visible to the qmailp user? Yes.
Is ~fred owned by fred? Yes.
Give control of the message to fred.
Run qmail-local.

**qmail-local fred ~fred fred-dude - dude tommy.gov **
joe@heaven.af.mil ./Mailbox

Does ~fred/.qmail-dude exist? No.
Does ~fred/.qmail-default exist? Yes: "./Mail.tommy".
Write message to ./Mail.tommy in mbox format.

[1]Taken from the `PIC.local2virt` file distributed with the qmail source code.

Local to local alias delivery[1]



Original message:

To: help
Hi.

Fill in the complete envelope and header:

**(envelope) from joe@heaven.af.mil **
to help@heaven.af.mil
From: joe@heaven.af.mil
To: help@heaven.af.mil

Hi.

Store message safely on disk.
Trigger qmail-send.

Look at envelope recipient, help@heaven.af.mil.
Is heaven.af.mil in locals? Yes.
Deliver locally to help@heaven.af.mil.

qmail-lspawn ./Mailbox

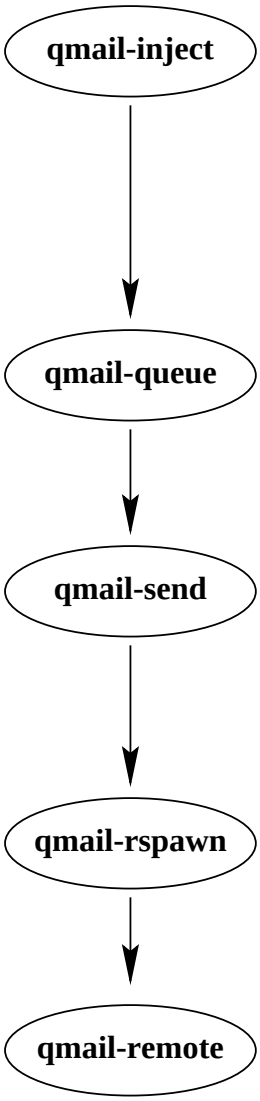
Look at mailbox name, help.
Is help listed in qmail-users? No.
Is there a help account? No.
Give control of the message to alias.
Run qmail-local.

**qmail-local alias ~alias help - help heaven.af.mil **
joe@heaven.af.mil ./Mailbox

Does ~alias/.qmail-help exist? Yes: "john".
Forward message to john.
(delivery is now a local to local delivery)

[1]Taken from the `PIC.local2alias` file distributed with the qmail source code.

Local to remote delivery[1]



Original message:

To: bill@irs.gov
Hi.

Fill in the complete envelope and header:

**(envelope) from joe@heaven.af.mil **
to bill@irs.gov
From: joe@heaven.af.mil
To: bill@irs.gov

Hi.

Store message safely on disk.
Trigger qmail-send.

Look at envelope recipient, bill@irs.gov.
Is irs.gov in locals? No.
Is bill@irs.gov in virtualdomains? No.
Is irs.gov in virtualdomains? No.
Is .gov in virtualdomains? No.
Deliver remotely to bill@irs.gov.

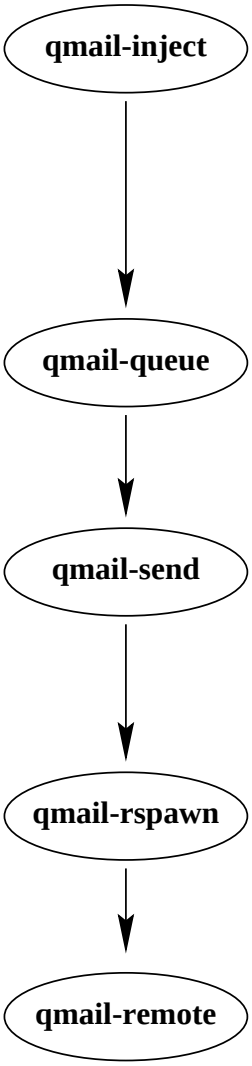
Run qmail-remote.

Look at host name, irs.gov.
Is irs.gov listed in smtproutes? No.
Look up DNS MX/A for irs.gov and connect to it by SMTP.

MAIL FROM: <joe@heaven.af.mil>
RCPT TO: <bill@irs.gov>

[1]Taken from the `PIC.local2rem` file distributed with the qmail source code.

Null client delivery[1]



Original message:

To: bill@irs.gov
Hi.

Fill in the complete envelope and header:

**(envelope) from joe@heaven.af.mil **
to bill@irs.gov
From: joe@heaven.af.mil
To: bill@irs.gov

Hi.

Store message safely on disk.
Trigger qmail-send.

Look at envelope recipient, bill@irs.gov.
Is irs.gov in locals? No.
Is bill@irs.gov in virtualdomains? No.
Is irs.gov in virtualdomains? No.
Is .gov in virtualdomains? No.
Deliver remotely to bill@irs.gov.

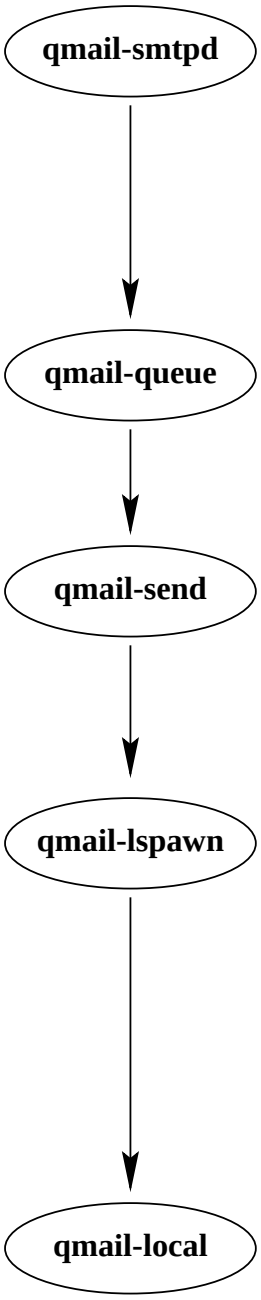
Run qmail-remote.

Look at host name, irs.gov.
Is irs.gov listed in smtproutes? Yes:
"bigbang.af.mil".
Look up DNS A for bigbang.af.mil and
connect to it by SMTP.

MAIL FROM: <joe@heaven.af.mil>
RCPT TO: <bill@irs.gov>

[1]Taken from the `PIC.nullclient` file distributed with the qmail source code.

Remote to local delivery[1]



Receive message by SMTP from another host:

MAIL FROM: <bill@irs.gov>
RCPT TO: <joe@heaven.af.mil>

Is $RELAYCLIENT set? No.
Is heaven.af.mil in rcpthosts? Yes.
Accept RCPT.

Store message safely on disk.
Trigger qmail-send.

Look at envelope recipient, joe@heaven.af.mil.
Is heaven.af.mil in locals? Yes.
Deliver locally to joe@heaven.af.mil.

qmail-lspawn ./Mailbox

Look at mailbox name, joe.
Is joe listed in qmail-users? No.
Is there a joe account? Yes.
Is joe's uid nonzero? Yes.
Is ~joe visible to the qmailp user? Yes.
Is ~joe owned by joe? Yes.
Give control of the message to joe.
Run qmail-local.

**qmail-local joe ~joe joe '' '' heaven.af.mil **
bill@irs.gov ./Mailbox

Does ~joe/.qmail exist? No.
Write message to ./Mailbox in mbox format.

[1]Taken from the `PIC.rem2local` file distributed with the qmail source code.

Unauthorised relaying[1]

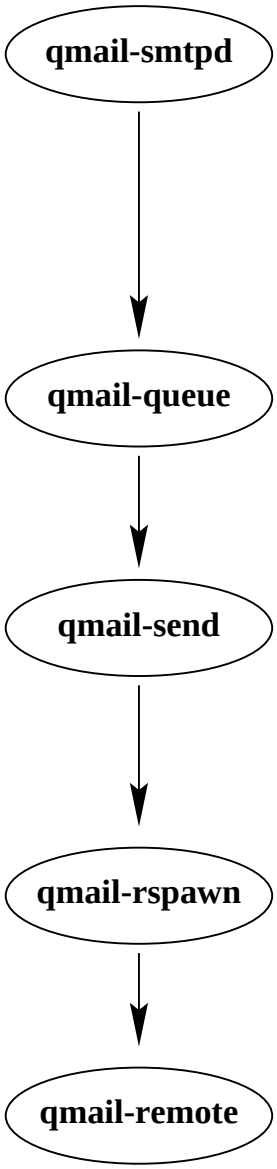
Receive message by SMTP from another host:

MAIL FROM: <spammer@aol.com>
RCPT TO: <bill@irs.gov>

Is $RELAYCLIENT set? No.
Is irs.gov in rcpthosts? No.
Reject RCPT.

[1]Taken from the `PIC.relaybad` file distributed with the qmail source code.

Authorised relaying[1]



Receive message by SMTP from another host:

MAIL FROM: <joe@heaven.af.mil>
RCPT TO: <bill@irs.gov>

Is $RELAYCLIENT set? Yes.
Accept RCPT.

Store message safely on disk.
Trigger qmail-send.

Look at envelope recipient, bill@irs.gov.
Is irs.gov in locals? No.
Is bill@irs.gov in virtualdomains? No.
Is irs.gov in virtualdomains? No.
Is .gov in virtualdomains? No.
Deliver remotely to bill@irs.gov.

Run qmail-remote.

Look at host name, irs.gov.
Is irs.gov listed in smtproutes? No.
Look up DNS MX/A for irs.gov and connect to it by SMTP.

MAIL FROM: <joe@heaven.af.mil>
RCPT TO: <bill@irs.gov>

[1]Taken from the `PIC.relaygood` file distributed with the qmail source code.

Independent message retry schedule

Instead of repeatedly retrying every deferred message, each message is given its own independent retry schedule. qmail uses a quadratic retry schedule for each unsuccessfully delivered message. Older messages are retried less often. Local messages are retried more often than remote messages.

Local messages:

$$time\ in\ queue = 100 * delivery\ attempt_n^{\ 2}$$

$$delay\ till\ next\ retry = 200 * delivery\ attempt_n + 100$$

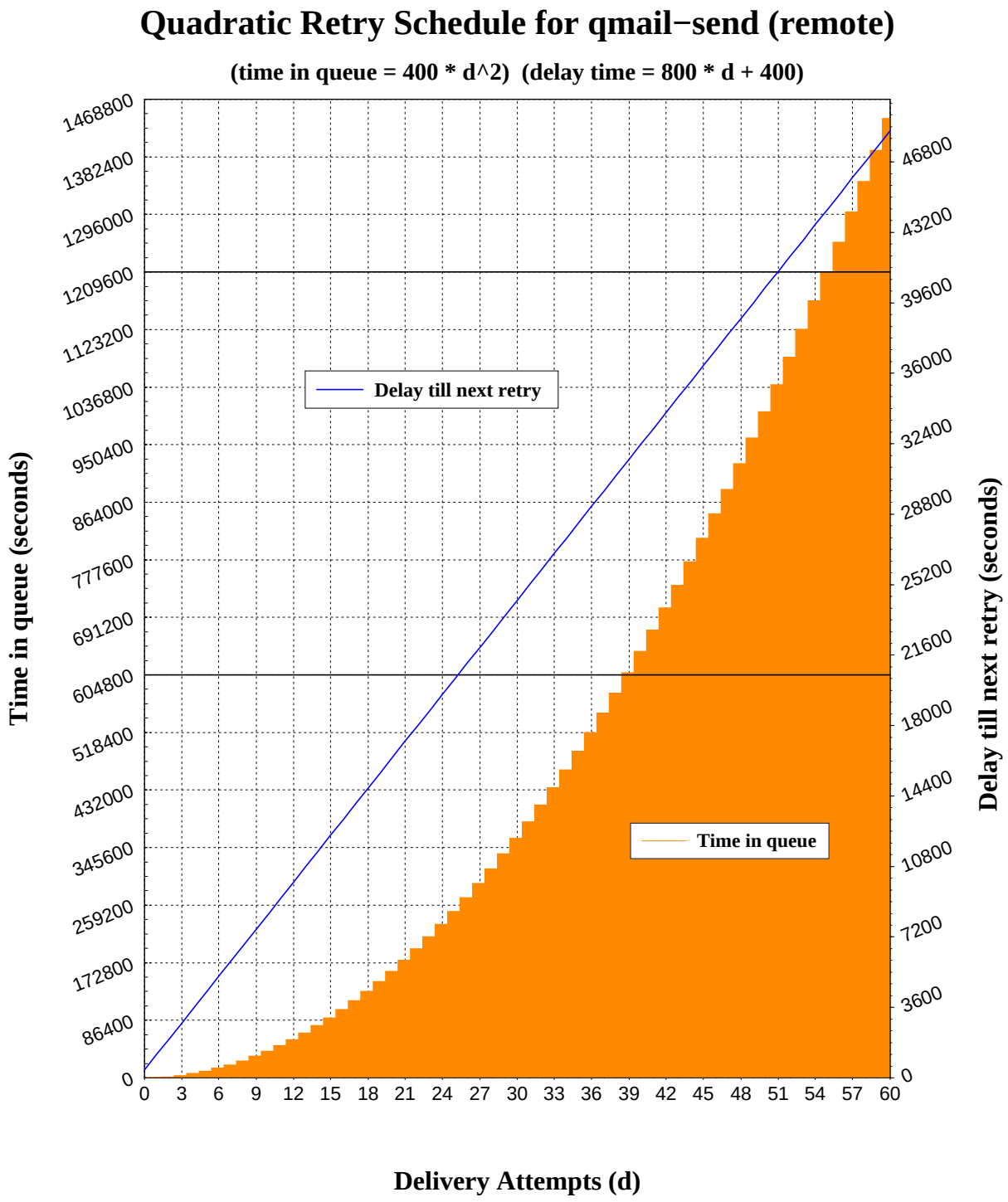
Remote messages:

$$time\ in\ queue = 400 * delivery\ attempt_n^{\ 2}$$

$$delay\ till\ next\ retry = 800 * delivery\ attempt_n + 400$$

Delivery d_0 is the first delivery attempt.

Independent message retry schedule



Remote host retry strategy

If a remote host does not respond to two connection attempts (separated by at least two minutes with no intervening successful connections), qmail automatically leaves the host alone for an hour. At the end of the hour it *slow-starts*, allowing one connection through to see whether the host is up.

The IP addresses of these unresponsive hosts are stored in /var/qmail/queue/lock/tcpto. The contents of this binary file can be examined using /var/qmail/bin/qmail–tcpto. The file can be reset using /var/qmail/bin/qmail–tcpok.

Compiling qmail

- download the qmail source distribution from **http://cr.yp.to/qmail.html** and unpack the distribution
- Modify the configuration files as required:
  - ./conf–break — user extension delimiter
  - ./conf–cc — compiler options
  - ./conf–groups — group names
  - ./conf–ld — linker options
  - ./conf–patrn — .qmail permission mask
  - ./conf–qmail — installation directory
  - ./conf–spawn — delivery concurrency limit
  - ./conf–split — number of queue subdirectories
  - ./conf–users — user names

Compiling qmail

- create the qmail installation directory

```
mkdir /var/qmail
```

- Create the qmail users and groups

/etc/passwd

```
alias:*:29990:29998::/var/qmail/alias:/bin/true
qmaild:*:29991:29998::/var/qmail:/bin/true
qmaill:*:29992:29998::/var/qmail:/bin/true
qmailp:*:29993:29998::/var/qmail:/bin/true
qmailq:*:29994:29997::/var/qmail:/bin/true
qmailr:*:29995:29997::/var/qmail:/bin/true
qmails:*:29996:29997::/var/qmail:/bin/true
```

/etc/group

```
qmail:*:29997:
nofiles:*:29998:
```

- Build and install the binaries

```
make setup check
```

Recommended patches

Don't patch qmail unless you really need functionality outside the default behaviour.

qmail will run quite happily on the bulk of system without any patches at all. However there are a few cases where patching qmail is useful. Recommended patches can be found at **http://www.qmail.org/** as well as searching the archives of the qmail mailing list.

- If you wish to use and/or compile qmail on a system using glibc-2.3.*x* you'll need to patch qmail so that it includes errno.h rather than defining `errno` as an external integer.[1]

- qmail does not recognize 0.0.0.0 as a local IP address, so remote deliveries to this address will loop.

- For sites that handle very large numbers of messages you may wish to apply the *big–todo* along with the *big–concurrency* patches. The first patch allows you to handle more incoming messages by providing a sub-directory structure to the todo directory. The second patch allows you to sustain a greater number of concurrent deliveries.[2]

---

[1]Other packages from Dan Bernstein, such as daemontools, also require this patch.

[2]It will **not** magically increase your network bandwidth. 

Other patches

There are many other third party patches for qmail. Almost none of them are required for normal[1] operation. Search the archives of the qmail mailing list for details.

If you need a simple mechanism to block Microsoft executables at the **SMTP** level, have a look at Russell Nelson's patches at

http://www.qmail.org/top.html#microsoft

These patches to qmail–smtpd reject messages if they contain executable Microsoft attachments. They are not meant as a comprehensive anti-virus scanner, but they do block much of the infected traffic coming into your system via **SMTP.**

If you apply these patches and your users complain that they can no longer send executable and/or zipped attachments via email, remind them that there are better mechanisms for transferring files. 

[1]It all depends on how *you* define *normal.*

Starting qmail

The qmail distribution provides a number of startup examples in /var/qmail/boot/*. The basic mechanism is

```
env - PATH="/var/qmail/bin:$PATH" \
    qmail-start [ aliasempty [ logger arg ... ] ]
```

The *aliasempty* argument is given to qmail–lspawn. This argument represents the default delivery instructions for those deliveries where there is no .qmail file available. It can be a simple mbox filename such as ./Mailbox or a maildir directory name such as ./Maildir/.[1]

The *aliasempty* argument may also be a different delivery program such as /bin/mail or procmail for example.

qmail's activity records are sent through the program specified by the *logger* argument – typically daemontools' multilog.

The start up environment is eventually passed to qmail–local so you may need to sanitise the environment before starting qmail.

You may also wish to place limits on qmail's resource usage using appropriate `ulimit(1)` settings.

[1]The trailing slash is **vital.** It signifies that the delivery destination is a maildir **directory** rather than an mbox **file** called Maildir.

Recommended qmail startup scripts

Dave Sill's excellent "Life with qmail" page – **http://www.lifewithqmail.org/lwq.html** – details the recommended[1] way in which to start the various qmail processes. This resource is commonly referred to as *LWQ.*

If you are not comfortable with rolling your own startup scripts, this resource is **highly** recommended.

If you ever ask the qmail mailing list for help, be prepared to justify why you are not following the LWQ methods.

[1]The scripts detailed in LWQ deviate from Dan Bernstein's original documentation. The LWQ methods are those recommended by the general qmail community and even Dan has deprecated some of his original start up methods. Note that LWQ takes advantage of software released by Dan since the last release of qmail.

Migrating from sendmail

By default, qmail will deliver to each user's home directory. If you wish to preserve your /var/spool/mail/user mail boxes, instruct qmail to deliver mail using a different MDA:

```
qmail-start \
    '| preline -f /bin/mail -r \
    "${SENDER:-MAILER-DAEMON}" -d $USER' 2>&1
```

Examples of different MDA configurations are available in /var/qmail/boot/*.

/etc/aliases **compatibility**

If you don't want to move away from your existing /etc/aliases file you can use the fastforward package.

Create a hash table of alias definitions by running

/var/qmail/bin/newaliases[1]

If you have `:include` references, create hash tables for each of these include files by running

```
newinclude /path/to/include/file
```

Enable the alias database(s) by creating ~alias/.qmail–default which contains

```
| fastforward -d /etc/aliases.cdb
```

fastforward will automatically read the contents of any `:include` references made in /etc/aliases provided you have run the newinclude command.

fastforward can also be used to maintain separate alias databases for each supported virtual domain.

[1] This is the fastforward version of newaliases and is not to be confused with the traditional sendmail version.

Using fastforward

fastforward can also be used to handle any number of aliases instead of using individual .qmail files.

Create a file to hold the alias definitions:

```
all@example.com:
    staff@example.com,
    managers@example.com,
    contractors@example.com;

staff@example.com: peters, syin, charles;

managers@example.com:
    jonathan, david, warwick, cindy;

contractors@example.com: gordonr, suresh, charlie, raoul;
```

Create a constant database file

```
setforward file.cdb file.tmp < file
```

Create an appropriate entry in .qmail–default

```
| fastforward -p file.cdb
| bouncesaying "Sorry, no such address. (#5.1.1)"
```


.forward **compatibility**

If you don't want to move away from your existing .forward files you can use the dot–forward package.

Start qmail by specifying .forward compatibility

```
qmail-start '| dot-forward .forward[1]
./Mailbox' 2>&1
```

If a user has a .forward file, qmail will obey the instructions in that file and no further delivery will take place. If the .forward file does not exist, qmail will deliver to the user's default mail box – specified above as ./Mailbox or follow instructions in the user's .qmail file if it exists.

If you don't wish to provide .forward compatibility to all users, individual users can take advantage of dot–forward by specifying dot–forward delivery in their own .qmail files.

```
| dot-forward .forward
./Mailbox
```

[1] The quotes and the newline here are **vital.** The dot–forward delivery instruction is separate from the ./Mailbox delivery instruction.

Mailing list management

Majordomo version 1.9*x* needs some modification to work with qmail. Version 2*.x* should support qmail *"out of the box"*. Version 2*.x* doesn't have a projected release date and minor modifications to version 1.9*x* are still being released (1.94.5 was released on 18 January 2000).

Russ Allbery has an excellent FAQ on how to modify majordomo – **http://www.eyrie.org/~eagle/faqs/mjqmail.html.** It contains pointers to a number of patches and auxiliary programs that allow you to continue using majordomo with qmail.

The basic steps involve

- Apply patches to majordomo
- Eliminate majordomo's wrapper program
- Create ~majordomo/.qmail–*list* files
- Change majordomo config files to use *list*–owner instead of owner–*list*

Mailing list management

Mailman version 2.1 will work with qmail. You must remember to make use of the preline command to ensure a *UUCP* style From header is provided for each incoming message.

```
| preline /home/mailman/mail/mailman post listname
```

For more details, see the mailman web site

http://www.list.org/

ezmlm – a qmail mailing list manager

ezmlm is specifically tailored for qmail. Features include:

- secure and simple subscribe/unsubscribe mechanism
- automatic message archiving
- open or closed lists
- smart bounce handling using qmail's Variable Envelope Return Path (VERP) to reliably determine the recipient address and message number for every incoming bounce message
- automatic unsubscription after 10 days of bounces
- any user can create an ezmlm list
- customisations available through .qmail files

Creating an ezmlm list

To create a *peters–jokes* list

```
ezmlm-make ~/JOKES ~/.qmail-jokes \
    peters-jokes my.example.com
```

Subscribe/unsubscribe requests are sent to

peters–jokes–subscribe@my.example.com
peters–jokes–unsubscribe@my.example.com

Postings are sent to

peters–jokes@my.example.com

List help is sent to

peters–jokes–help@my.example.com

The list owner can be contacted at

peters–jokes–owner@my.example.com

Users send retrieval requests for message 12345 to

peters–jokes–get.12345@my.example.com

Customising an ezmlm list

Disable message archiving

```
mv ~/JOKES/archived ~/JOKES/notarchived
```

Prevent public subscription

```
mv ~/JOKES/public ~/JOKES/notpublic
```

Force a `Reply-To: list` header

```
echo "Reply-To: peters-jokes@my.example.com" \
    >> ~/JOKES/headeradd
```

Allow closed user group subscription

```
touch ~/JOKES/public
```

and then edit ~/.qmail–jokes–default and insert

```
| domaincheck listname domain1 domain2 ...
```

Domaincheck is an executable of your own design that examines **$SENDER** and exits 0 if the sender's domain is in one of the listed valid domains. If the sender is not from one of these domains, domaincheck prints an appropriate message to stdout which will appear in the bounce message. Domaincheck then exits 100 which informs qmail that no further processing is to take place.

Customising an ezmlm list

Further possible customisations include:

- modify ~/JOKES/headerremove to exclude specific headers
- modifying administrative text messages by editing ~/JOKES/text/*
- adding list specific footer messages by modifying ~/.qmail–jokes (all on one line):

```
| cat - /home/peters/JOKES/text/footer[1] |
    /usr/bin/ezmlm-send '/home/peters/JOKES'
```

- modifying `Subject:` headers and adding a footer message by modifying ~/.qmail–jokes (all on one line):

```
| subject-prefix[2] JOKES |
    cat - /home/peters/JOKES/text/footer |
    /usr/bin/ezmlm-send '/home/peters/JOKES'
```

- just about anything you can think of to do in a .qmail file

[1] footer is a text file of your own making

[2] subject–prefix is a program of your own design that modifies the `Subject:` header of the message

ezmlm–idx

Many of the above customisations to ezmlm (and many others) are available in the third party package ezmlm–idx. See **http://www.ezmlm.org/.**

ezmlm–idx is based on ezmlm 0.53. It supports:

- message digests
- remote list administration
- subscriber only posting
- list moderation
- subject prefixing
- specific non-subscriber posting
- web accessible message archiving
- in-place list reconfiguration
- and more
- just about anything you can think of to do in a .qmail file

daemontools

The daemontools package allows you to control startup, shutdown and automated restarting of any number of processes – without needing to know their process ID. See **http://cr.yp.to/daemontools.html.**

svscan watches up to 1000 subdirectories of a named directory – typically /service. Each of these subdirectories (which may be symbolic links to other locations) contains a run script that starts a process in the foreground. svscan starts up a supervise process for each of these run scripts. If the process dies, supervise automatically restarts it.

If the /service/*xyz* directory has the sticky bit set,[1] and it has a subdirectory called /service/*xyz*/log, svscan will create a pipe between the original supervise process and another supervise process which is controlling the log/run script. This allows for controlled log gathering using multilog (or any other logging system you desire).

multilog automatically rotates logs when they reach a given size or on receipt of an **ALRM** signal. The number and size of the logs are configurable. multilog can also filter input lines based on limited regular expressions.

[1]The sticky bit is not required for versions of daemontools greater than 0.70.

Using daemontools

Create the service directory /service and start svscan from one of:

- /etc/rc.local[1]

```
/command/svscanboot &
```

- /etc/inittab
  - add an entry to /etc/inittab[2]

```
sv:2345:respawn:/command/svscanboot
```

  - signal init to re-read /etc/inittab

```
telinit q
```

[1]Use whatever system startup script is appropriate for your operating system.

[2]Solaris users will need to modify this entry to read

```
sv:2345:respawn:/command/svscanboot </dev/null >/var/log/svscan 2>&1
```


Using daemontools

Create a qmail startup script,[1] /var/qmail/supervise/qmail–send/run:

```
#!/bin/sh
exec env - PATH="/var/qmail/bin:$PATH" \
    qmail-start ./Mailbox 2>&1
```

Create a qmail log script, /var/qmail/supervise/qmail–send/log/run:

```
#!/bin/sh
exec setuidgid qmaill multilog t /var/log/qmail
```

Toggle the sticky bit on the service directory:[2]

```
chmod +t /var/qmail/supervise/qmail-send
```

Make a symbolic link from the master service directory

```
ln -s /var/qmail/supervise/qmail-send /service/qmail-send
```

Within a minute, qmail will be started and all details will be logged in the automatically rotated file /var/log/qmail/current.

---

[1]Comprehensive start up scripts are available from the *Life with qmail site.*

[2]Not required for versions of daemontools greater than 0.70.

Using daemontools

Processes can be stopped using

```
svc -d /service/qmail
```

Processes can be started using

```
svc -u /service/qmail
```

Processes can be restarted using

```
svc -dx /service/qmail
```

To avoid losing any data if there is a supervised log process, svscan will keep the pipe open, even while the supervised process is down.

Using tcpserver

tcpserver is part of Dan Bernstein's ucspi–tcp package. It is a replacement for inetd and, optionally, tcp_wrappers. It is available from **http://cr.yp.to/ucspi–tcp.html.**

tcpserver is the preferred mechanism for starting qmail network daemons (and any other **TCP** daemons that might normally be started from inetd).

```
tcpserver [options] host port program [args]
```

Example:[1] Run tcpserver in verbose mode, as user *qmaild*,[2] as group *nofiles*,[3] allow up to 100 simultaneous connections, allow all interfaces to listen for incoming **SMTP** connections:

```
tcpserver -v -u 29991 -g 29998 -c 100      \
    0 smtp /var/qmail/bin/qmail-smtpd 2>&1
```

The daemontools package has a utility called envuidgid which sets the value of **$UID** and **$GID** to the user and group ids of the specified user. tcpserver can be told to use these variables with the **–U** option. (This requires ucspi–tcp version 0.88 and above).

---

[1]Comprehensive start up scripts are available from the *Life with qmail* site.

[2]You **must** use the UID number rather than a name.

[3]You **must** use the GID number rather than a name.

Using tcpserver

tcpserver can be configured to allow or deny access on an IP basis in a similar fashion to tcp_wrappers.

Create a rules file, one file for each service under the control of tcpserver. /etc/smtp.tcp for example.

```
149.135.128.:allow,RELAYCLIENT=""
127.:allow,RELAYCLIENT=""
149.135.13.:deny
peters@149.135.13.17:allow,RELAYCLIENT=""
:allow,DATABYTES="1048576"
```

Create the rules database

```
tcprules /etc/smtp.cdb /etc/smtp.tmp < /etc/smtp.tcp
```

Example: Run tcpserver in verbose mode, as user *qmaild*, as group *nofiles*, allow up to 100 simultaneous connections, allow all interfaces to listen for incoming **SMTP** connections based on the rules in /etc/smtp.cdb.

```
tcpserver -v -u 29991 -g 29998          \
   -x /etc/smtp.cdb -c 100 0 smtp       \
   /var/qmail/bin/qmail-smtpd 2>&1
```

Using inetd instead of tcpserver[1]

The use of inetd for qmail (and just about everything else) has been deprecated by the qmail community.

One of inetd's most annoying *"features"* is its rate-limited shutdown. If inetd determines that the rate of incoming connections for a particular service is too high, it will stop servicing that port for 10 minutes. Given qmail's parallel delivery mechanism, even a small system can bombard a remote inetd rapidly. I have actually flattened a largish Australian ISP's inetd from a 14k4 modem. They quickly changed to tcpserver. 

If you must use inetd, consider an alternative such as xinetd, **http://www.xinetd.org/,** or rlinetd, **http://www.rcpt.to/rlinetd/.** Neither of these programs suffer from the rate-limited shutdown problem. Both programs have a different configuration syntax to inetd and both provide inbuilt IP access control using the tcp_wrappers library libwrap, unlike inetd which relies on the tcp_wrappers binary program tcpd.

If you're still desperate to use inetd, examples of /etc/inetd.conf configuration entries can be found in the FAQ distributed with the qmail source.

---

[1]Both inetd and tcpserver may be used on the same system, however only one of them can service a particular port and IP address combination at a time.

Tips and troubleshooting

- Did you read **ALL** the man pages and the FAQ?
- Did you search **http://www.qmail.org/** ?
- Did you examine the qmail logs?
- If mail is not being delivered, but a burst of deliveries is seen every time you start qmail–send, check your trigger file. Its permissions are quite specific:

  prw--w--w- 1 qmails qmail 0 Apr 13 14:26
  /var/qmail/queue/lock/trigger

- If mail is not being delivered to some users, check the permissions on the user's home directories, mailboxes, Maildirs and .qmail files. The qmail log file should mention permission problems. Some programs may hold a lock on the mailbox file, this prevents local delivery.
- You can set up multiple qmail–smtpd programs and switch between them by setting **$PATH** in your tcprules. (Requires tcpserver).

Tips and troubleshooting

- If you need to move the qmail queue (or it has been trashed), use queue–fix to repair broken inodes, permissions etc. **http://www.netmeridian.com/e–huss/queue–fix.tar.gz**
- Use qmail–lint to check your qmail control files. **http://www.glasswings.com.au/qmail/qmail–lint–0.55**
- Disable **IDENTD** lookups by running tcpserver with the -R option.
- Run qmail–start, tcpserver etc from supervise to ensure they auto restart when necessary. supervise is part of the daemontools package.
- Use fixcrio[1] to ensure CRLF end of line chars are enforced. Start qmail–smtpd from tcpserver:

```
tcpserver -v -u 29991 -g 29998          \
    -x /etc/smtp.cdb -c 100             \
    0 smtp sh -c 'fixcrio               \
    | /var/qmail/bin/qmail-smtpd' 2>&1
```

[1] fixcrio is from the ucspi–tcp package

Tips and troubleshooting

- Hide your machine identity using tcprules.

```
149.135.128.11:allow,RELAYCLIENT="",
    TCPREMOTEHOST="mail.tansu.com.au",
    TCPREMOTEIP="192.168.0.1"
:allow
```

  This will affect the `Received:` header used by qmail–queue, qmail–qmqpd and qmail–qmtpd.

- Broken file permissions in /var/qmail can be restored by running `make setup check` from the source directory.

Performance tuning tips

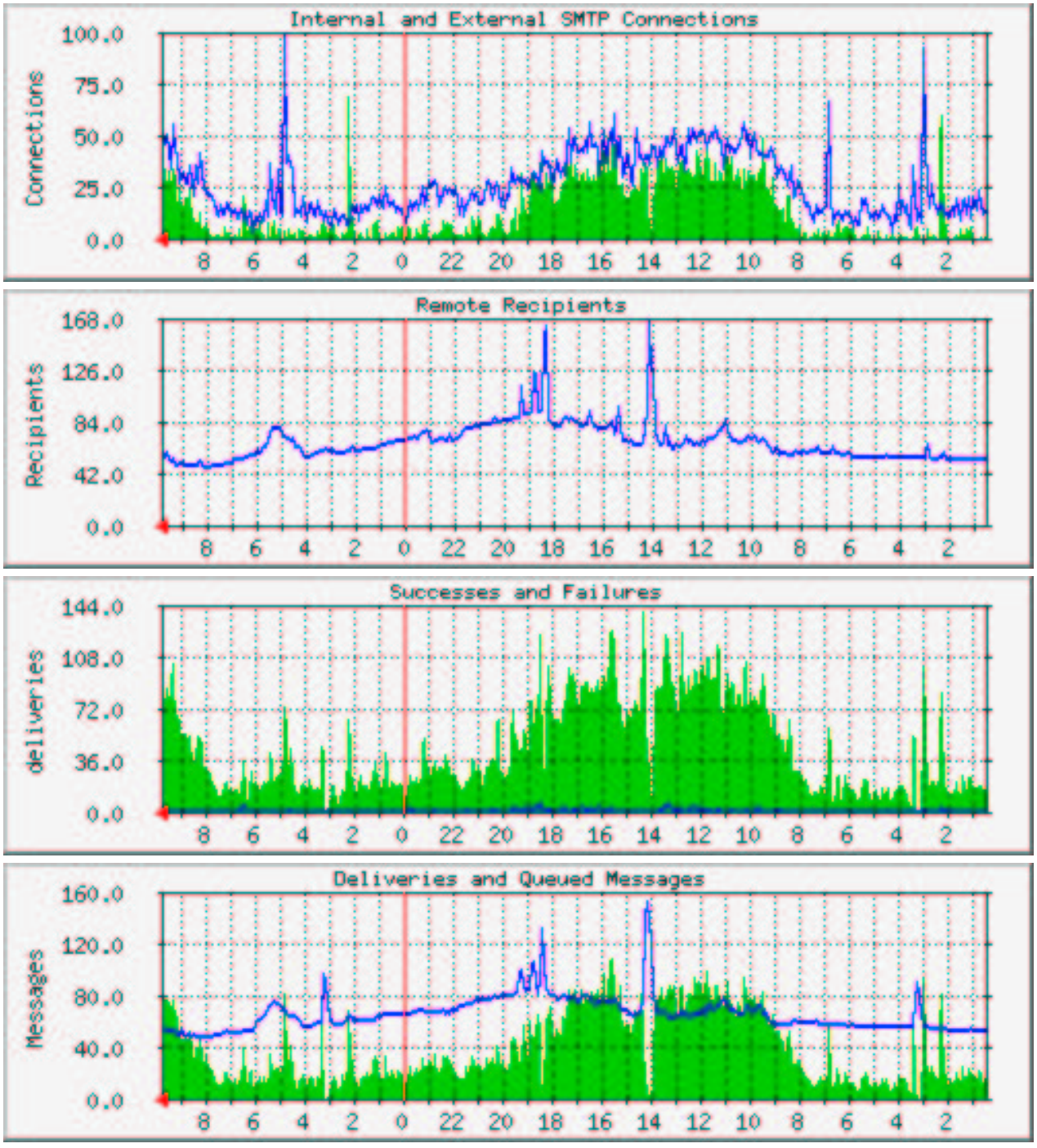
- Avoid syslogd. On some operating systems it can inhibit performance with its use of the `sync(2)` or `fsync(3)` calls. Use multilog from Dan's daemontools package instead. **http://cr.yp.to/daemontools.html**
- Place /var/qmail in its own file system. If possible, place it on a separate disk and/or controller and take full advantage of available RAID style tools to improve I/O performance.
- Postfix comes with some good performance loading tools, smtp–source and smtp–sink. You can use these to test **SMTP** daemons and MTA systems under various load conditions.
- Run qmail on multiple machines, each with the same MX value.
- Run multiple instances of qmail on the same machine. Each has its own /var/qmail directory: /var/qmail, /var/qmail2, etc. Modify conf–qmail and recompile for each instance you wish to support.
- Incoming **SMTP** messages can be directed to a random qmail–smtpd process using a simple wrapper program of your own design called from tcpserver.

Performance tuning tips

- If you need to support many local and/or virtual domains, store the extra domains in the morercpthosts control file, and create the constant database file using qmail–newmrh.
- For systems handling enormous amounts of mail, increase the number of queue directories by editing conf–split before compiling. The default value is 23. Choose a number such that each queue directory contains no more than 1000 files at most.[1]
- Apply the conf–split structure to the todo directory, instead of its default flat structure, by applying the *big–todo* patch. See **http://www.glasswings.com.au/qmail/big–todo.103.patch.**
- Allow the value of concurrencyremote to be increased above 240 by applying the *big–concurrency* patch. See **http://www.glasswings.com.au/qmail/big–concurrency.patch.**

[1]Some file systems such as reiserfs can handle large queue sizes with a conf–split of 1. Make sure you do some profiling before you make such a change.

Performance analysis[1]



[1]Real data from a corporate mail gateway. Gathered by MRTG after analysing qmail and tcpserver logs.

Performance analysis

- The system handles more **SMTP** connections on its external network interface than its internal network interface.
- Traffic profiles seem to follow core business hours on both interfaces.
- The number of items in the queue remains around 50. This indicates that remote sites are either busy or under powered.
- Just after 2am yesterday, a large number of **SMTP** connections arrived on the internal interface. Around 3am on the same day, an even larger number of **SMTP** connections arrived on the external interface. This lead to a corresponding increase in deliveries at these times.
- The large increase in queued messages, and the corresponding drop in deliveries at 2PM yesterday, indicates all the possible qmail–remote processes were busy with large messages. A similar situation arose at around 6:30PM yesterday and again at around 3am this morning.

Ancillary software

qmail is distributed with a number of ancillary programs.

qmail–qstat – show the number of queued and pre-processed messages in the queue.

qmail–qread – shows the sender and recipient details and delivery status of all pre-processed messages.

qmail–pop3d – a **POP3** daemon that supports maildir formats. Requires a checkpassword program.

sendmail – wrapper allowing local MUA programs to be used unchanged. They can still compose mail and deliver it via `/usr/lib/sendmail -t`. Calls qmail–inject.

Ancillary software

datemail – similar interface to the sendmail wrapper. Sets `Date:` header in local time instead of GMT for those MUAs that do not set the header. Calls qmail–inject.

preline – inserts a **UUCP** style `From` *`user`* header before calling an external program. Preline insists on sending the entire message to stdout. It can be upset by downstream programs, such as some vacation programs, that close stdin before preline has finished with stdout.

forward – forwards mail messages to the specified list of addresses. List can be constructed on the fly.

And many more.

Additional software

Dan Bernstein, and others, have written a number of other packages that can be used with qmail. Dan's software is available from **http://cr.yp.to/software.html.**

qmailanalog – analyses log files and generates mail statistics.

ezmlm – mailing list manager specifically for qmail. Supports automatic subscription, archiving and retrieval of messages.

checkpassword – required by qmail–pop3d. Originally written by Russell Nelson.

fastforward – sendmail like alias clone for sites that don't wish to move away from /etc/aliases files.

dot–forward – sendmail like .forward clone for sites that don't wish to move away from .forward files.

Additional software

serialmail – allows batch processing of mail transport over slow/intermittent serial links.

VmailMgr – A Virtual mail manager designed to manage multiple domains of mail addresses and mailboxes on a single host. See **http://www.vmailmgr.org/** for more details.

qmail–vacation – a vacation package designed to work with qmail. See **http://www.gormand.com.au/peters/tools/qmail/.**

eliminate–dups – eliminates duplicate messages arriving in your mailbox. See **http://www.gormand.com.au/peters/tools/qmail/.**

Additional software

There are a host of other third party packages available from the qmail web site – **http://www.qmail.org/.**

Extensions to ezmlm.

Patches to provide maildir support for MUAs.

Alternative checkpassword programs.

Smart startup/shutdown wrappers.

HTML versions of the manual pages.

qmail patches for better SPAM blocking etc.

qmail books

At present there are two books dedicated to qmail.

"The qmail Handbook" by Dave Sill, published by Apress in October 2001. ISBN 1-893115-40-2. This book covers everything mentioned in Dave's online "Life with qmail" pages but with greater detail. It also covers topics not included in "Life with qmail". This book has been very well received on the qmail mailing list.

"Running Qmail" by Richard Blum, published by SAMS in August 2000. ISBN 0-672-31945-4. This book covers much that isn't qmail specific and leans more towards users of BSD based systems. It has not been well received on the qmail mailing list.

Erwin Hoffmann has written a book in German. Unfortunately his publisher cancelled the project in April 2003. Erwinn has made the text available on-line at http://www.fehcom.de/qmail/qmailbook.html.

John Levine has been working on a qmail book for O'Reilly & Associates. It was supposed to be published in 1999, however that date has slipped somewhat.  John still claims to be working on the book so look out for it *real soon now.*

Commercial support

Several organisations are willing to provide commercial support for qmail. For more details see

http://www.qmail.org/top.html#paidsup

Community support

There are a number of web pages devoted to qmail. The best of these are

http://www.qmail.org/

http://www.lifewithqmail.org/lwq.html

These sites have the best reputation among the qmail community.

Dan Bernstein maintains a mailing list for qmail support at **qmail@list.cr.yp.to.**[1] Dan hardly ever makes an appearance on the list. The list is medium volume and discussion can be quite heated at times. If you plan on posting to the list make sure you have done your homework first, otherwise you'll be flamed to death. A good reference to ensure you ask sensible questions on the list can be found at

http://www.qcc.ca/~charlesc/writings/12–steps–to–qmail–list–bliss.html

[1]See the **URLography** on page 103 for subscription details.

The future

Zero administration for null clients. Replace **SMTP** with **QMQP** for communication between null clients and central mail host. A full implementation of **QMQP** ships with qmail version 1.03. Dan plans to release a mini qmail package with **QMQP** support enabled by default.

Faster installation. Provide pre-compiled packages for popular operating systems.

Split log analysis. Improvements to qmailanalog. The next version will be designed to run periodically, keeping crucial information in a table on disk. qmailanalog-0.70 has some of these features now.

[Speed] Wide-area QMTP support. Use **QMTP** between qmail hosts to cut down on **SMTP** latency. Requires encoding the **QMTP** availability in the hosts MX record. See RFCMXPS and RFCQMTP in the qmail distribution for more details.

The future

[Speed] Asynchronous compressed journaling. Reduce qmail's disk I/O by feeding new mail through a separate journaling process that saves messages in compressed form; qmail–send will rebuild the queue from the compressed journal when it starts.

[Speed] Local–host DNS lookups. A future version of qmail will support an alternate, much smaller **DNS** library that talks to a nameserver on the same host. Work has already begun on this project. Dan's dnscache package is a replacement for BIND. It is available from **http://cr.yp.to/dnscache.html.**

Dynamic subscription agents. Why should users have to deal with dozens of different mailing list subscription mechanisms? The new dynasub package will accept subscription requests from local users and negotiate subscriptions with remote mailing lists. It will automatically set up a local sublist for each remote list, to speed delivery and protect user privacy.

Don't ask when qmail–2.0 will be released! Take the *zen* approach and let it happen. 

URLography[1]

Source Code

http://cr.yp.to/qmail.html
ftp://moni.msci.memphis.edu/pub/qmail/
– qmail RPMS
http://www.qmail.org/
– more RPM references

Help

http://www.qmail.org/
http://cr.yp.to/qmail/faq.html[2]
http://www.lifewithqmail.org/lwq.html
http://www.flounder.net/qmail/qmail–howto.html
http://www.CULTe.org/projets/doc/qmail/v0.4/
– en français
http://www.palomine.net/qmail/relaying.html
http://www.eyrie.org/~eagle/faqs/mjqmail.html
– qmail with majordomo FAQ
http://www.ornl.gov/cts/archives/mailing–lists/qmail/
– searchable
http://www.egroups.com/list/djb–qmail?refstop=1
http://msgs.securepoint.com/qmail/
– searchable

[1]Some of these links and/or their contents may be out of date.

[2]Start up details have been deprecated in favour of using tcpserver and daemontools.

URLography

Other Mailing List Details

http://cr.yp.to/lists.html
qmail–subscribe@list.cr.yp.to
cdb–subscribe@list.cr.yp.to
ezmlm–subscribe@list.cr.yp.to
password–subscribe@list.cr.yp.to
qmailannounce–subscribe@list.cr.yp.to
serialmail–subscribe@list.cr.yp.to

Ancillary software

http://cr.yp.to/software.html
http://cr.yp.to/checkpwd.html
http://cr.yp.to/dot–forward.html
http://cr.yp.to/ezmlm.html
http://cr.yp.to/fastforward.html
http://cr.yp.to/mess822.html
http://cr.yp.to/qmailanalog.html
http://cr.yp.to/serialmail.html
http://cr.yp.to/ucspi–tcp.html

URLography

Third Party Ancillary Software

http://www.qmail.org/
– loads of references
http://www.ezmlm.org/
– ezmlm with loads of extras
http://www.gormand.com.au/peters/tools/qmail/eliminate–dups-2.1.tar.gz
http://www.gormand.com.au/peters/tools/qmail/qmail–vacation–1.4.tar.gz
http://www.netmeridian.com/e–huss/queue–fix.tar.gz
http://qmail.sericyb.com.au/qmail/qmail–lint–0.55
http://qmail.sericyb.com.au/qmail–qsanity–0.52
http://www/vmailmgr.org/
– multiple virtual domain manager
http://www.nrg4u.com/
– qmail with **LDAP** support, *Big Picture* home page
http://www.inter7.com/qmailadmin.html
http://untroubled.org/mailfront/
– alternate **SMTP** daemon
http://sourceforge.net/projects/qmhandle
http://www.qmail.org/qmLogsort
http://people.ee.ethz.ch/~oetiker/webtools/mrtg/
http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/

URLography

Anti SPAM

http://maps.vix.com/rbl/
http://www.orbs.org/
http://www.obtuse.com/smtpd.html
http://spam.abuse.net/spam/

Other MTAs

http://www.exim.org/
– version 4.20, (August 2003)
http://www.postfix.org/
– version 2.0.14, (August 2003)
http://www.sendmail.org/
– freeware version
– version 8.12.10, (September 2003)
http://www.sendmail.com/
– commercial version
ftp://ftp.uu.net/networking/mail/smail/
– version 3.2, (July 1996; beta 112, February 2001)
ftp://ftp.cs.toronto.edu/pub/zmailer/
– version 2.2e10, (March 1996)

Patches

http://www.qmail.org/
– loads of references
Search the qmail mailing list archives

URLography

Web Based Mail Interfaces[1]

http://netwinsite.com/webmail/
http://www.horde.org/imp/
http://www.emumail.com/
http://www.endymion.com/
http://www.inter7.com/sqwebmail.html
– only supports Maildir/
http://www.landfield.com/hypermail/
http://www.lugs.ch/LUGS_Members/norbert.kuemin/wmf.html
http://www.netwinsite.com/
http://www.sendmail.com/products/mailcenter_webmail.shtml

[1]These may or may not work with qmail.