

Qmail in the FreeBSD Universe

Renato Botelho, Erwin Hoffmann

EuroBSDCon Karlsruhe/Germany
October 2010

1 Introduction

Qmail is a Mail Transfer Agent **MTA** submitted by Daniel J. Bernstein as open source in 1998 in the still (final) version 1.03. Since then, only very minor bugs have been identified (but never corrected for by the author), while the (E)SMTP protocol has significantly changed and enhanced.

However, the superior design of Qmail and in particular the new unrestricted distribution license allows a smart adoption to the current needs, which reflects its authority to be still a major MTA.

1.1 Scope

A lot of attempts have been made, to add additional functionality to Qmail 1.03. Russel Nelson's Web site www.qmail.org represents a good sample of enhancements. However, the diversities of the third-party developments, with different scopes, 'philosophies', and maturity level makes it difficult and frustrating for a potential Qmail user to adopt the required 'patches' and merge them into a functional Qmail setup.

Erwin Hoffmann's Spamcontrol is an attempt to accommodate Qmail to the requirements of the 21st century. Unlike heavy sets of patches (e.g. Bill Shupps 'Qmail Toaster'), it provides some lean improvements which have been incorporated by many other developers.

Renato Botelho has enhanced

Qmail+Spamcontrol with SPF to provide an even more complete solution for FreeBSD available as a port.

This paper tries to outline some of basic ingredients of the design concepts and usage under FreeBSD.

1.2 Authors

Renato Botelho is using FreeBSD since 1999. In 2004 he started to contribute to Ports Collection and in addition setting up Qmail as MTA in conjunction with Spamcontrol, as being described in some Brazilian tutorials. In march 2005, he added the Spamcontrol slave port, became shortly after that the maintainer of all Qmail ports and finally became FreeBSD ports committer in July the very same year. Right now he is working at BluePex Security Solutions in Limeira/Brazil dealing with Linux and FreeBSD based Security Appliances.

Erwin Hoffmann uses Qmail since version 0.99 and FreeBSD since 3.2 back in 1998. He is originator of the Spamcontrol patch for Qmail and other Qmail-related tools. Currently, he is Associate Professor at the University of Applied Sciences in Frankfurt/Main Germany.

2 Design and Components

2.1 Components

Since Dan Berstein has published Qmail in the mid of the 1990's he has successfully raised

a lot of companion products which improve Qmail as a MTS substantially:

- During those days Dan Bernstein wrote Qmail, some other (SMTP) MTAs were typically used. Certainly, **sendmail** was the major favorite, though it was recognized to have a substantial history of bugs. **smail** and **zmailer** were other candidates. **Postfix** and **Exim** were at their dawn in those early day.

Qmail included some revolutionary concepts:

- QMTP as *Quick Mail Protocol*.
 - VERP the *Variable Address Response Path*, now cannibalized as BATV¹.
 - Maildir, store one email as file and not stacked into a **mbox**.
 - Isolating MTA² from the MDA³ and the MS⁴ processed with minimal permissions.
 - Virtual Domains: Domains glued to a Unix user instead of the MDA default user.
- **Daemontools** is a set programs to care a about *daemon programs* utilizing
 - **supervise**, which not only provides *high availability HA* for those daemons but in addition with
 - **multilog** the logging is straightened and a unique **TAI64** timestamp is written to the log output.

¹*Bounce Address Tag Validation* draft by *John Levine*

²MTA: Mail Transfer Agent, responsible to transfer email from hop to hop via (E)SMTP

³MDA: Mail Delivery Agent, receives the email and stores in into a private container)

⁴MS: Message Store: Email is a store-and-forward system, the MS is effectively the 'Queue'

- **ucspi-tcp** is a set of services to be used in conjunction with **qmail-smtpd** and **qmail-qmtpd** since both rely on network socket-services, which are otherwise commonly known as **inetd** and **xinetd**:

- **tcpserver** is the main working horse, which typically is used as *tcpwrapper*.
- **rblsmtpd** provides a flexible lookup into *Relay Black Lists (RBL)*.

- **ucspi-ssl** is the alternative to **ucspi-tcp** and written by *William E. Baxter* from Superscript⁵. In particular **sslserver** complements **tcpserver**'s capability marrying it with the **OpenSSL** library for transport layer encryption **TLS**.

- **djbdns** has revolutionized the recognition of the DNS protocol. Before the area of **djbdns**, BIND was considered simply as a synonym for DNS services.

- **dnscache** is a DNS full-resolver and cache server, though without DNSSEC capabilities.
- **tinydns** is a DNS content server, which avoids the complexity of a BIND zone file in a ingenious concept.

Furthermore, Dan Bernstein introduced initially the concept of DNS *port randomization* which became famous by Dan Kamininsky's description of the DNS *birthday attack*⁶. However, this was already recognized (and solved) by **djbdns** years before. Disentangling the tasks of a DNS content server from those of a DNS resolver and introducing a so-called *split horizon* view while responding to DNS queries can additionally be attributed to

⁵www.superscript.com

⁶<https://www.kb.cert.org/vuls/id/800113>

Dan Bernstein. Nobody even thinks today about running an open email relay. The very same approach is valid now regarding DNS caching services.

2.2 Qmail

Qmail 1.03 unlike – **sendmail** – is a highly composite piece of software. With it’s basic design, it includes

- a (E)SMTP client (**qmail-remote**) and server (**qmail-smtpd**),
- a QMTP client (**qmail-qmqpc**) and server (**qmail-qmqtd**, **qmail-qmqpd**),
- a POP3 server (**qmail-pop3d**),
- a local Mail Delivery Agent (**qmail-local**), and – of course –
- a Message Store (**qmail-queue**).

Compared to other MTAs, Qmail does not provide enough contemporary services, whether ESMTP RFC-compliant, nor best-practice (like *Greylisting*). However, it’s design and it’s robustness makes it an ideal platform for further enhancements. For example, IndiMail⁷ is a current fork of Qmail.

3 Spamcontrol

Spamcontrol is a patch for vanilla Qmail which was started in the year 2000, mainly to include some already existing patches for Qmail and to adjust Qmail in particular to the needs of public german providers while incorporating some anti-spam features in particular for **qmail-smtpd**.

Since version 2 however, it aims to incorporate the current ESMTP requirements following Dan Bernstein’s C coding conventions

⁷<http://www.indimail.org/>

Product	#Files	#WoC
Exim 4.66	272	1.125.411
Postfix 2.7.1	792	1.105.774
Qmail 1.03	279	180.553
+ Spamcontrol 2.6.23	323	231.003
+ ucspi-tcp	509	298.938
+ ucspi-ssl	711	389.273

Table 1: Comparison of the code base (Words of Code) for Exim, Postfix, and Qmail

and his approach for simplicity. Most of those developments are also separately available as add-ons for vanilla Qmail⁸.

As already mentioned, one of the fundamentals of Qmail and Spamcontrol is it’s minimalistic approach and avoiding initial complex solutions and of course coding.

This becomes obvious, if we compare Qmail+Spamcontrol with the current available competitors *Philip Hazel’s Exim*⁹ and *Wietse Venema’s Postfix*¹⁰ [Table 1].

Though this count disfavors Qmail substantially, we can see that the executable code base for Qmail is roughly 1/3 of the other MTAs. Conceptually, Qmail stays light-weight in the system and should perform as such.

3.1 <Forward-Path> Validation

One criticism, in particular carried out by Matthias Andree¹¹, is due to the fact, that Qmail accepts *any* email in the first place which targets to an acceptable domain (**control/rcpthosts**). Since today spam mails simply use forged recipient addresses, the generated Bounce email (NDR) is a heavy load on the system. In particular, because the advertised <Return-Path> address is almost

⁸<http://www.fehcom.de/qmail.html>

⁹<http://www.exim.org/>

¹⁰<http://www.postfix.org/>

¹¹<http://www.dt.e-technik.uni-dortmund.de/~ma/qmail-bugs.htm>

always fake, this results in unwanted double-bounces.

Here, Spamcontrol provides with the *Recipient mechanism* a state-of-the art solution. **qmail-smtpd** may ask a particular back-end for address-validation. The back-end might be a local **cdb** (*Constant Data Base*) or (currently supported only) a LDAP directory. Since the Recipient extension facilitates Dan Bernstein's **checkpassword** API by means of a PAM, practically any remote source can be queried.

3.2 ESMTP Authentication

While SMTP was originally designed as Host-to-Host protocol, with ESMTP Authentication it is actually acting as a User-to-Host communication. Once *authorized* (typically by means of *Userid* and *Shared Secret*), the MTA enables for the authenticated user the privilege for *unrestricted relaying*. If this information is transmitted over un-encrypted channels, the *Userid* is subject of a substantial *data leakage*, while at least by means of the Challenge/Response method **CRAM-MD5**, the *Shared Secret* can be protected.

Apart from the different authentication schemes (in Qmail+Spamcontrol are supported **LOGIN**, **PLAIN**, and **CRAM-MD5**), Qmail uses solely *Pluggable Authentication Modules PAMs*, as crafted by Dan Bernstein's *checkpassword* API¹². The SASL¹³ framework (even *Postfix* users neglect it's simplicity and security¹⁴) was never an alternative for Qmail.

3.3 TLS

Neglecting spams and mailing list newsletters, the whole email traffic should be considered as *private*. Thus it deserves *confidentially*. Due

¹²<http://cr.yip.to/checkpwd/interface.html>

¹³RFC 4422

¹⁴Peer Heinlein in 'Linux Magazin' 9/2010 introducing *Dovecot* version 2

to the lack of an accepted end-to-end encryption, *Transport Layer Security TLS* is a reasonable mean to provide *confidentially* at least to the next (E)SMTP relay, and perhaps the final destination.

Based upon the **ucspi-ssl** environment, Qmail+Spamcontrol provides the means for **TLS** both while receiving (**qmail-smtpd/qmail-pop3d**) and sending (**qmail-remote**). However, the complexity and (often recognised) incapability of today's **PKI Public Key Infrastructure** is not yet exploited fully in Spamcontrol regarding **CRLs Certificate Revocation Lists** and **OCSP Online Certificate Status Protocol**.

Here, Spamcontrol supports fully 'virtual hosting' for TLS:

- Separately hosted domains can be bound to specific **qmail-smtpd** instances on different IP addresses presenting a domain-specific X.509 certificate instead the standard host certificate.
- Reversely, **qmail-remote** can be advised to bind to individual IP addresses based on the <Return-Path> of authoritative addresses and furthermore supports domain-specific certificate, in case the ESMTP servers requires this.

In addition, the chosen cipher suite (aka the security context) can be defined per hosting domain.

On the other side, some current 'gadgets' like TLS session caching are not implemented (yet). Within the TLS approach, the server has to carry the burden of calculating cryptographic informations in the first place. However, considering the fast development of hardware and CPUs in particular, it seems to be inadequate to compensate for the design problems of TLS with even more sophisticated SW solutions.

3.4 Virus prevention

Spamcontrol provides three solutions for the (fortunately currently not so virulent) potential infections of the target systems:

1. The 'Warload' algorithm allows to reject particular attachments based on their MIME type and Loader recognition in wire-speed¹⁵.
2. A seamless integration of the AV scanners, like in particular ClamAV, can be facilitated by means of the 'Qmail High Performance Scanner Interface' **QHPSI**. Here, the AV scanner directly access the email in the Qmail queue. Since most AV scanners have a built-in understanding of BASE64 attachments and also support scanning of archives like ZIP, there is simply no particular reason why sophisticated 'staging areas' are required.
3. Spamcontrol also supports the `QUEUE.EXTRA` scheme, allowing the replacement of `qmail-queue` by a wrapper in the first place, to do virus and spam scanning.

3.5 Anti-Spam tools

Though Spamcontrol provides some advanced means to control the SMTP envelope, spam sender have (mostly) accommodated to that and we are left with some basic means:

- Employing `tcpserver` or `sslserver`, the existence of a qualified DNS record of the sender can be checked. This information can be used to allow a `badmail-fromunknown` and `badmailfromwell-known` filtering.
- `rblsmtpd` allows the lookup into several *Relay Black Lists* **RBL**.

¹⁵published in: *c't* 11/2004 'Exe und Hopp'

- A patched version of `rblsmtpd` permits to *Greetdelay*¹⁶ the ESMTP connection. Though the idea points back to *tarpitting*, in spite of the million Bot net PCs commanded to send spam mails, this mechanism is surprisingly efficient and easy to use.
- Currently, the alternative *Greylisting* is not supported by Spamcontrol, though some solutions exist for Qmail. Actually, I am not a friend of *Greylisting*, it's effect on uncontrollable delivery latency and in particular it's substantial coding requirements, and potential failures.
- To recognize spam mails on behalf of it's *reputation* and/or *content* is the traditional regime of SpamAssassin¹⁷. The complex operations SpamAssassin performs, requires (a) CPU power, (b) fast disk access (c) network connectivity. Item (b) is currently a limiting factor. Spamcontrol comes with wrapper-script `qmail-queue.scan` which allows to perform all virus and spam scanning activities commonly on a RAM disk, rather than on the real disk, speeding up those tasks significantly in addition with a substantial reduction of I/O load.

3.6 QMQ

One important lesson of today's email processing is the use of *multi-tier* systems. Often, the last member in the chain is a Microsoft *Exchange* server, hosting the mailboxes of the users, while the front-end servers (close the Internet) are recruited from systems operating under the Unix OS. However, those *Exchange* systems often need to be serviced individually and protected in particular from spam and virus storms.

¹⁶<http://www.fehcom.de/qmail/Greetdelay.pdf>

¹⁷<http://spamassassin.apache.org/>

The 'Qmail Multiple Queue' option (QM_Q) of Spamcontrol provides an appropriate solution for this task. Here, *one* Qmail system (using Spamcontrol) is responsible for the Internet connectivity, while vanilla Qmail instances are used to transmit emails from and to the target email systems. All Qmail systems may communicate with the QMTP protocol instead of SMTP. Since the 2nd line systems can be adjusted to the required *Service Level Agreements* SLAs, this results practically in a *predictable* delivery delay to the upstream MTA.

By the same token, a dedicated *Bounce Host* can be setup, which not only releases the standard Qmail queue from delivering NDR's (most to none-existing user account in case of spam), but also helps to avoid accidentally blacklisting the standard Qmail instance.

4 FreeBSD Qmail Ports

Since it's Dan Bernstein's concept¹⁸ instead of having an 'all-inclusive' solution rather to use dedicated and mutually 'untrustworthy' services, a complete installation of Qmail depends on very many components which are now available as ports for FreeBSD (mostly done by Renato):

- 'mail/qmail-spamcontrol', the most recent port of Spamcontrol (2.6.23_1) is available thru **portsnap**.
- 'sysutils/ucspi-tcp' (0.88_2), including man-pages and optional IPv6 support (among) other.
- 'sysutils/ucsp-ssl' (0.70_1) contributed by *David Thiel* and including my patch for *STARTTLS* support.

¹⁸D.J. Bernstein: 'Some thoughts on security after ten years of qmail 1.0' <http://cr.yip.to/qmail/qmailsec-20071101.pdf>

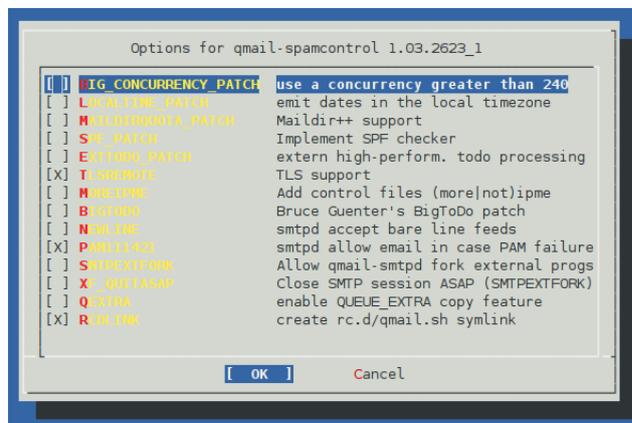


Figure 1: Setup of qmail-spamcontrol options

- 'sysutils/daemontools' (0.76_15) provided by *Peter Pentchev* coming with the *Gerrit Pape*'s man-pages.
- 'dns/djbdns' (1.05_13) again from *Peter Pentchev* supporting in addition IPv6.
- 'security/checkpassword' (0.90) as a potential PAM to access `/etc/passwd`.
- 'mail/qmailanalog' (0.70_2) package to statistically analyse the **qmail-send** log.
- 'mail/qmailmrtg7' (4.2.4) port from *Alex Dupre* using *Inter7*'s¹⁹ presentation of *Tobias Oetiker*'s MRTG²⁰.

Apart from 'qmail-spamcontrol' there exist other major enhancements for Qmail: 'qmail-tls' is based upon *Frederik Vermeulen*'s TLS implementation²¹ and 'qmail-ldap' providing *Andre Oppermann*'s LDAP²² solution. However, due to their complexity those ports conflict with each other.

End of 2007, Dan Bernstein has released almost all of his software into the *public domain*

¹⁹<http://www.inter7.com/>

²⁰<http://tobi.oetiker.ch/hp/>

²¹<http://inoa.net/qmail-tls/>

²²<http://www.nrg4u.com/>

which not only allows (private) *modifications* but also permits the *distribution* of the modified sources without conflicting with any license restriction.

Let's focus on two important add-ons for `qmail-spamcontrol`, Renato has included.

4.1 SPF enhancement

The **Sender Policy Framework**²³ adds *reputation* to the sending MTA by means of either the specific *SPF* Resource Record or a standard *TXT* Record:

```
example.com. 3600 IN SPF "v=spf1 mx -all"
```

Both, the mail administrator and the DNS administrator have to carefully define and setup those records. On the other hand, it is task of SMTP server to

1. query this information,
2. perhaps to reject mails in case of inadequate SPF settings, or
3. to forward this information to external spam checkers, like SpamAssassin, and
4. to report those findings in an additional `Received-SPF:` header.

Within the `qmail-spamcontrol` port, **qmail-smtpd** is responsible for those operations. Renato took the native (DJB-like) code from *Christophe Saout*²⁴. The resulting *spfbehavior* can be adjusted by means of control files – and even more fine-grained – by environment variables defined for certain IP addresses or domain names.

4.2 Concurencies and Big-Ext- Todo

Since Qmail was written primarily as RFC 821 MTA, it incorporates the idea of 'single message transaction'. This means, for every email

²³RFC 4408

²⁴<http://www.saout.de/misc/spf/>

one (E)SMTP session is used, even though the email targets many recipients on the same system; which has been discussed with many pros&cons in the past. However, this behavior guarantees that each email is stored individually in the *Message Store*. This – in turn – requires that the process **qmail-send** cares about each and every email and updates it's status in the queue concurrently: in it's spare time. However, on a busy email system this conflicts with other tasks of **qmail-send** and may result in a race condition named 'silly qmail syndrome'.

Since the **qmail-smtpd** process is lightweight, a mailserver connected to the Internet can easily service 5000 concurrent SMTP connections, and of course needs to get rid of those emails not keeping those in the Message Store for too long²⁵.

There are a few solutions to prevent/overcome this situation:

1. The more sending channels Qmail (**qmail-remote**) has, the faster the emails can be released from the **MS**. Dan Bernstein has initially foreseen a very conservative limit for a remote concurrency of 255, while within the FreeBSD Qmail port this is raised to 509.
2. *Andre Oppermann* has enhanced Qmail to simply split **qmail-send**'s responsibility for handling those queue operations by means of the `EXTTODO_PATCH` available as option in the port.
3. Another possibility is, to setup Qmail as a multi-tier system: **QMQ**. A script 'multiple-queues.ksh' (customizable via the file 'conf-qmq') allows to configure up to 100 Qmail instances, which is the preferred way for Spamcontrol.

²⁵In this sense *Greylisting* impacts the book keeping of a mailserver significantly, while *Greetdelay* puts the burdon on the network stack

5 Qmail 'Up and Running' ?

After the required ports have been installed, there is unfortunately no-such-thing like a running Qmail. While checking the FreeBSD handbook, section 28.4 ('Changing Your Mail Transfer Agent') provides some ideas about it and in particular references *Postfix*, there is practically no information about Qmail.

While the script 'enable-qmail' provides some hints at the end of the installation process, and within the created directory '/var/qmail/boot/' some start-up scripts exist, they only allow a very basic set-up of Qmail. Actually, they require the completely outdated use of **inetd** and rather should completely removed from the port!

In addition, the recommended settings of `qmailsmtpdenable="YES"` becomes completely obsolete in spite of **supervise**.

On the other side, the qmail-spamcontrol FreeBSD port supports the recommended 'mailwrapper' functionality while adding the appropriate Qmail modules to the file `/etc/mail/mailler.conf`.

So what is missing ?

5.1 supervise

We consider to run Qmail (and it's components) *supervised*. Having already installed the *Daemontools* port, we need to enable it. While Dan Bernstein's recommends the setting of `csch -cf '/command/svscanboot &'` into **rc.local**, the FreeBSD port

- provides the script `svscan` within `/usr/local/etc` and allows a customization by means of FreeBSD's standard `/etc/rc.conf` variables:
- `svscan_enable="YES"`,
- `svscan_servicedir` defaulting to `/var/service/` – which needs to be

created before – , and

- some specific settings of `svscan_logdir`, `svscan_lognum`, `svscan_logmax` which may be left to the defaults in the beginning.

5.2 multilog

A busy mail server provides a lot of logging information. If you run an ISP-like service, it is important to keep those logs for a certain time period, typically a few days. Those logs are also required to provided *monitoring* based on **MRTG** and perhaps *alarms*.

In addition to the standard **qmail-send** log, Spamcontrol provides extensible logging for **qmail-smtpd**. This information is vital to identify in particular spam storms and other attacks.

While for low-volume mailservers **syslogd** may be an appropriate choice, a busy mailserver (we are talking about 1-10 mio/mails per day on a single server) **multilog** is a must. Here, the logs of the individual daemons are written to particular directories with automated log-file cycling and backups. Easily 1 GByte logging information per **qmail-smtpd** instance can be written a day, and are in particular handled. In case, Erwin's **Newanalyse**²⁶ script together with the `qmailanalog` port is used, automatic reports and backups are provided. In addition, every single email processed by Qmail can be easily transaction-safe tracked.

Typically, we keep the logs in directories named

- `/var/log/qmail-send/`
- `/var/log/qmail-smtpd/`
- `/var/log/qmail-pop3d/`

²⁶<http://www.fehcom.de/qmail/newanalyse.html>

and setting the ownership to 'qmail', Qmail's dedicated log user.

This scheme can be extended to other services as well. We keep in mind, that the size of the attached filesystem and it's speed is an important factor here.

5.3 run scripts

In the **supervise** context a *daemon* or *service* is identified as (directory) name, while the start script always is named 'run'²⁷. Thus, for any of the serviced directories we need a particular **run** script operating under the control of **supervise**. However, unlike FreeBSD's traditional start/stop scripts (in particular those coming with the qmail-spamcontrol port), **run** scripts simply catch *signals* from **supervise** and/or the command **svc** to operate and hence do not require arguments like 'start' or 'stop'. *Gerrit Pape* has released a nice collection²⁸ of **run** scripts for daemons, like **apache**, **postfix** and many others.

The logging to **multilog** is facilitated as 'co-process' simply piping the information available at FD1 and FD2 to this process placed in an adjacent `./log/` directory.

Here are two samples of my own (slightly trimmed) **run** scripts:

qmail-send:

```
#!/bin/sh
exec env - PATH="/var/qmail/bin:$PATH"
qmail-start ./Maildir/
```

qmail-smtpd:

```
#!/bin/sh
QMAILDUID='id -u qmaild'
QMAILDGID='id -g qmaild'
HOSTNAME='hostname'
```

²⁷Frankly, this is the opposite idea w.r.t. *System V*'s run-level scripts to be found in `etc` container (`/etc/init.d`) containing alpha-numerically ordered *symlinks* to the start/stop scripts

²⁸<http://smarden.org/runit/runscripts.html>

```
MAXCONCURRENCY="100"
export HELOCHECK="A"
export QHPSI="clamscan"
exec sslserver -sven -Rp\
-l $HOSTNAME\
-c $MAXCONCURRENCY\
-x /var/qmail/etc/tcp.smtpd.cdb\
-u $QMAILDUID -g $QMAILDGID\
192.168.192.1 smtp\
rblsmtpd -WCr sbl-xbl.spamhaus.org\
/var/qmail/bin/qmail-smtpd\
cmd5checkpw true 2>&1
```

Ouch. It's obviously getting difficult to run a MTA!

5.4 Configuration & Customization

In the Qmail universe, in particular **qmail-smtpd** needs

1. *site-specific configuration*, providing the basic setup of the daemon, resource limits, bindings and adjacent programs to call. These settings stay constant as long as the process is running. Environment variables can be mapped into a directory by means of Daemontools' **envdir** program, and
2. *service-specific customizations* allowing run-time modifications, for instance SMTP envelope filters and in particular the *cdb* which defines the rules for **tcpserver** or **sslserver**.

Though Qmail – in it's basic setup – can very well operate with a minimum of configuration & customization, in practice the more settings are provided, the more are used (and of course are often only very barely documented).

The FreeBSD port 'mail/qmailadmin' seems to solve that problem; but too bad, this package is only suited for Inter7's *Vpopmail*²⁹.

²⁹<http://www.inter7.com/?page=vpopmail>

5.5 Twisted Edges

In practice – due to the dependency of Qmail with for instance djbdns and the queried LDAP servers for email address validation the virus scanning, SpamAssassin and other tools – a suitable Qmail system becomes even more complex. Thanks to **supervise** and **multi-log**, we guarantee high-availability and non-repudiation of email transactions, providing a soft pillow for administrators, though not solving the case of

- configuration management and
- alarming (incident management).

6 Summary

Running the ported Qmail on FreeBSD provides – under the condition of careful configuration & customization – a high-available and high-performance MTA. On current hardware, 1-10 mio/emails per day can be handled with full virus and spam scanning on a single machine with low thrupt latency. Due to the concept of virtual domains, Qmail is also very well suited for large-scale end-user support.

However, the final bill includes substantial positions regarding the preparation of the Qmail system to fit seamlessly into the FreeBSD universe.