

Technical Guides

Designing Reliable Mail Systems

Andrew Cormack

UKERNA

GD/JANET/TECH/005 (03/03)

JANET Technical Guides

JANET Technical Guides are one of a series of user guides available to JANET customers. Technical Guides contain detailed technical information and are intended for technical support staff at JANET sites, network specialists or those with a particular interest in the specialist area.

If you have any queries or comments about the Guide or would like to obtain copies, please contact:

JANET Customer Service

UKERNA

Atlas Centre, Chilton, Didcot

Oxfordshire, OX11 0QS

Tel: 01235 822212

Fax: 01235 822397

E-mail: service@janet.ac.uk

Further details of the documents in this series are available at:

<http://www.ja.net/documents/>

Contents

1. Introduction

2. Likely Problems

2.1 Hardware Failure

2.2 Mail Floods

3. Mail System Design

3.1 Separate Local and Network Functions

3.2 Redundancy

3.2.1 *Multi-site Redundancy*

3.2.1.1 *Relaying by Secondary Mailhubs*

3.2.2 *Hardware Redundancy*

3.3 Spreading Processing Load

3.4 Hardware Choice

3.4.1 *Minimal System*

3.4.2 *Basic System*

3.4.3 *High-Performance System*

4. Mail Server Tuning

4.1 What to Tune

4.2 Tuning Exim

4.3 Tuning Postfix

4.4 Tuning Sendmail

4.5 Tuning Microsoft® Exchange

4.6 Tuning Novell® GroupWise®

4.7 Tuning Filters

5. Emergency Measures

5.1 Restricting Clients

5.2 Rate Limiting

5.3 Making Things Worse

Appendix A: A Brief Guide to SMTP

A.1 Network Layer

A.2 Routing E-mail

A.3 Failures, Retries and Disposal

A.4 Dealing with Mail Floods

Appendix B: Glossary

Introduction

1

1 Introduction

E-mail is now one of the most vital services to educational organisations, both for communication within the organisation and with the outside world. Although e-mail does not have the high profile of other services such as the world wide web, local users will complain much sooner and more urgently if they cannot send and receive mail than if they cannot see a web site. Since e-mail is the normal method for arranging most of what goes on in a university or college, including meetings, lectures and research, this should not be a surprise.

With e-mail playing such a key role in the organisation, it is worth taking some care to make the systems that provide it as robust as possible, to ensure that some simple failure or event does not result in a catastrophic failure of the service. A robust mail service requires proper design to ensure that mail messages have the best chance of being delivered and then appropriate tuning of the individual mail systems to ensure they can handle heavy loads. If problems do occur there are also some temporary measures that can be taken to reduce their impact.

This document originated in a discussion on the uk-mail-managers mailing list run by Newcastle University. Thanks are due to the members of that list for their helpful and detailed contributions of which some are quoted in the text.

Likely Problems

2

Hardware Failure

Mail Floods

2 Likely Problems

One major advantage of e-mail is that the Simple Mail Transfer Protocol (SMTP), which mail servers use to get mail messages from place to place, is one of the most resilient of all Internet services. Many mail servers will continue to try to deliver a mail message for a week or more before finally admitting defeat and returning it as undeliverable to the sender. Mail messages are almost never lost, they will either be delivered to the recipient, returned to the sender or, if neither of those is possible, passed as a last resort to a human postmaster to deal with the problem.

Although SMTP is very reliable there are two common situations that can cause problems for an e-mail service. These are hardware failures, either of a server or a network link, and mail floods, where some event causes the volume of mail to exceed temporarily the capacity of the system to deal with it. Both types of event are relatively common so it is prudent to design any e-mail service in such a way that their effects are reduced as far as reasonably possible.

2.1 Hardware Failure

The term hardware failure is used here to include any event that causes part of an e-mail system to become unavailable. This includes failure of computer hardware or network links, but also other events affecting availability such as power outages or holiday shutdowns. Most of these events are likely to cause some disruption while they last, though even this can be reduced, as will be described later. However the very persistence of the e-mail protocol can cause problems when systems are restored after a failure. At this point there may be up to a week's backlog of e-mail waiting on thousands of servers elsewhere on the Internet. When the service is restored, all those servers are likely to discover this within a few minutes and all will then attempt to deliver all of their waiting messages immediately. The impact of this surge of delivery attempts can easily overload a system that has not been designed to cope with it; this can easily result in a software or even hardware failure that makes the system unavailable again without significantly reducing the backlog. This vicious circle of outage, overload, failure can be hard to break.

2.2 Mail Floods

Restoring a mail system after a failure may result in a mail flood. Such floods may also be encountered in normal operation as a result of the activities of local or remote users. Local users have been known to create mail floods simply by joining too many mailing lists when they first discover the facility. Web pages that allow users to subscribe to 'all lists' with a single mouse click are a particular hazard. Some recipients of this kind of flood may be the victims of 'jokes' by their peers. Mail floods are also often associated with Unsolicited Bulk E-mail (UBE) as unscrupulous advertisers send thousands of copies of their message to different, and possibly not even valid, addresses at the same organisation. A most unwelcome development is where advertisers do not wish to reveal their own e-mail address and instead put forged sender addresses in their mails. The victims of such forgeries may be harmed by their apparent association with the product advertised. Their mail systems are also likely to experience a mail flood as all the incorrectly addressed mails are returned to their 'sender'. In some cases this kind of flood, referred to as collateral spam, has involved hundreds of thousands of messages, sufficient to prevent servers handling legitimate mail for a number of hours.

Mail System Design

3

Separate Local and Network Functions

Redundancy

Spreading Processing Load

Hardware Choice

3 Mail System Design

The first stage in making a mail service robust is to ensure it is well designed. Although it is possible to implement a complete mail system on a single computer, for any but the smallest sites this is not a good idea. Dividing the various components between different computers can improve performance when the service is running normally as well as reducing the impact when there are problems. Selecting appropriate hardware can also provide a better service. Although mail systems do not usually require particularly powerful systems they do have some unusual requirements.

This section presents a number of recommendations for mail service design and concludes with three examples of well-designed mail services in use at JANET sites.

3.1 Separate Local and Network Functions

One simple choice that can greatly improve the users perception of the mail service is to use different computers to handle local and Internet mail. This immediately insulates local users from any external problems. If some event causes a mail flood from outside the organisation then it is possible to slow or suspend Internet mail while the problem is dealt with without interrupting the exchange of e-mail between local users. If mailboxes for local users are held on more than one system then these will need some way to exchange messages, either by allowing internal messages to pass freely through the internet mail system or by providing a mailhub dedicated to internal traffic. Likewise if there is a problem with the local system that prevents it from accepting mail, then external mail will still be delivered to the Internet-facing system(s). Here delivery parameters, such as the time before mail is returned to its sender, are under the control of the local site. During a long outage, this time could be increased (if sufficient disk space is available) to prevent mail being returned. The Internet system can, indeed must, also be tuned so it does not swamp the internal system with a flood of messages when it returns to service.

Separating the system or systems that hold mailboxes for local users from those that handle mail from the Internet also has security and performance advantages. The mailbox systems with all their user accounts can be shielded from external attacks and the different systems can be provided with different hardware and software configurations appropriate to their different requirements.

These two components of the mail system are known as mailstores (the systems with user mailboxes) and mailhubs (the systems that distribute mail around the network). The design of a mailstore is very much dependent on local circumstances, such as the number of users, type of user authentication, mailbox protocols and client software used. Mailhubs have many more common features, so this document will concentrate on them.

3.2 Redundancy

A single mailhub, receiving all an organisation's mail from the Internet, represents a single point of failure. If the mailhub is overloaded, suffers a failure or simply requires maintenance, then e-mail will not be able to reach the organisation from the outside world. In the short term, perhaps for an hour or so, this should not be a problem as internal mail should still be circulating within the mailstore. However a longer interruption will lead to mail being delayed or, possibly, returned to the sender, which is unlikely to be acceptable to either local or remote users. These problems can be avoided by providing a backup mailhub to share or take over mail processing, either on a temporary or permanent basis. Such configurations are very easy to set up as options to support them were designed into the SMTP protocol from the start.

As described in Appendix A, SMTP allows a site to advertise multiple mailhubs through its Mail Exchanger (MX) records in the Domain Name Service (DNS). Each mailhub is given a numeric preference. External systems should try first to deliver mail to the mailhub with the smallest preference value and then try those with larger preference values if the first choice cannot be contacted. If there are mailhubs with equal preference values then external systems should choose one of them at random. Thus a load-sharing system can be established simply

by advertising two mailhubs with the same preference, or a fallback system by giving the principal mailhub a lower preference value than its stand-in. Provided there is always at least one advertised mailhub available, it does not matter if one or more of the advertised systems are not running, or not providing an e-mail service, at any given time. The SMTP protocol will find the system that is in service and deliver messages to it. This mailhub can then deliver the messages to the appropriate mailstore, either directly or via other mailhubs. This gives a great deal of flexibility and control both for normal operations and for dealing with unexpected events. Note, however, that changes to DNS entries can take some time to propagate to all hosts on the Internet, so it is a good idea to plan any emergency measures in advance and create DNS entries for backup hosts before they are needed in an emergency. Setting short Time To Live (TTL) values, for example 30 minutes, on the DNS records for mailhubs should allow changes to these records to propagate more quickly.

3.2.1 Multi-site Redundancy

Using duplicate mailhubs provides protection against a number of different types of failure, however if all the hubs are in a single physical location then there are failures that can affect all of them. For example a power cut at the site will stop all hubs from operating, or a failure of the network access link into the site may render them all unreachable from the outside world.

To protect against this type of failure, it is considered good practice if possible to have at least one mail exchanger advertised in the DNS that is at another location, with separate power and network connections. This may be another location belonging to the same organisation (provided independent power and network connections are available), or it may be another organisation.

As the offsite mailhub is only intended to be used in emergencies, it is common to give it a larger preference value than any local mailhub. In this way all mail will normally be delivered straight to the destination site, but if an emergency causes all systems at that location to be unreachable then the offsite backup will provide temporary storage for mail until the main systems are returned to operation. Once this occurs, the backup machine will send any stored mail to the local mailhubs for normal processing and delivery. An offsite mailsystem with a larger preference value is normally referred to as a secondary mailhub.

The requirement on a secondary hub is simple: it should accept mail addressed to the domain for which it acts, but should do as little processing as possible. If a mailhub with a smaller preference value is available then the secondary should simply pass messages immediately to that machine. If no such mailhub is currently reachable then messages should be stored until delivery is possible or a retention period has passed.

Although this is a simple configuration, it does require additions to the normal mailhub configuration to accept messages addressed to a domain other than the local one. Also, although the secondary mailhub will normally not receive any messages, it must be prepared (and have sufficient network, disk and processing resources) to handle the entire mail traffic to the site for which it is acting if there are any problems at that site. In some cases it will be appropriate to dedicate a computer to the secondary mailhub function. In others an existing mailhub for a domain at another site will have sufficient spare capacity to act as a secondary for another domain. The choice of option must be agreed with the site hosting the secondary before it is advertised in the DNS, as the presence of a secondary mailhub is likely to cause unpredictable surges in mail flow, disk and memory space requirements, and network traffic.

If a dedicated secondary mailhub is used then it may be possible for system managers at the home domain to administer it. Remember, that the secondary is there to protect against lost power or network connections so ensure it can be managed even in these circumstances. More commonly, the secondary will be administered by a member of the organisation where it is located. A trusted person in this role can be a great help, for example tuning the secondary in emergencies to ensure optimum mail retention and delivery. However the administrator of the secondary mailhub also has the ability to disrupt the mail service or breach the privacy of

users. It may therefore be appropriate to consider some more formal agreement that defines the service provided. Indeed in some cases the provider of a secondary mailhub could be considered a processor of personal data making the arrangement subject to the Data Protection Act.

3.2.1.1 Relaying by Secondary Mailhubs

The purpose of a mailhub is to accept messages from anywhere on the Internet so long as they are addressed to a domain it serves. For on-site mailhubs messages arrive from off-site and are then distributed within the site. However an offsite secondary mailhub must accept messages from off-site systems and then send them to another off-site system, usually an on-site primary mailhub. This means that the secondary mailhub will be relaying mail between two systems, both of them outside to its own site. Systems that act too freely as relays are often abused to distribute unsolicited bulk e-mail, so it is essential that the mailhub be set up to relay only messages that are addressed to its primary domain.

It is also important to ensure that if the primary mailhub does any checks before accepting Transmission Control Protocol (TCP) connections, for example to reject connections from systems that are on blacklists or that do not have properly configured DNS entries – then these checks must also be done by the secondary. If this is not done then a blacklisted host could connect to the secondary, that will accept the connection and pass the mail to the primary. The primary will accept the connection in good faith from its trusted secondary, unaware that the mail came originally from an untrusted source.

On the other hand as a secondary mailhub will pass all messages to the primary for delivery, it should not be necessary to duplicate any content-based filtering, for example for viruses, on the secondary. Where multiple systems can deliver messages in parallel, it is important that they should all do the same filtering so that messages that happen to follow a particular delivery route cannot evade these checks.

3.2.2 Hardware Redundancy

It is possible to protect a computer against the failure of a single hardware component by using hardware redundancy. Experience suggests that the most likely parts of a mailhub to fail are the hard disks. There are various ways to provide duplicate disks so that the computer will continue to run even if one disk fails. Methods that use multiple disks to improve some aspect of performance are known as RAID systems (Redundant Array of Independent Disks) and can be implemented in either hardware or software. However there are different classes of RAID systems that address different operational requirements and not all of these are suited to use on a mailhub.

On a normal mailhub, each message is simply received and passed on to the appropriate destination. This means that the spool area, where messages are queued, will be required to do a roughly equal number of writes and reads, one for each message. Systems that prioritise reading over writing, such as RAID 5, are therefore not particularly suitable for the spool disk, though they may be appropriate for the operating system, software and configuration areas. In fact the simplest RAID method, having two disks that duplicate (or mirror) each other's content so if either fails the system can continue to run with just the other, gives good protection against disk failure and performance that will normally be sufficient. This is known as RAID 1. For the ultimate in performance from a disk system, mirroring can be combined with disk striping in a RAID 10 arrangement, but for a mail service it is probably more effective to share load between multiple mailhubs. Any mirroring system requires data to be written to two separate disks, thus doubling the amount of information that needs to pass over the connection between the disk and the processor. To reduce the chance of a bottleneck occurring it is often a good idea to connect the parts of the mirror to different disk controllers so the writes can be done in parallel. This also makes the system resistant to the failure of a single disk controller.

Most operating systems include software to support mirrored disks, either as part of the basic operating system or as an option. These are usually reasonably straightforward to set up, though setting up mirroring of the system boot disk can be a more complex process. However

if the decision has been taken to use mirrored disks then it is worth providing this protection for all of the filesystems: it always seems to be the one unprotected disk that fails!

Although it is possible to duplicate other hardware components, for a mailhub in a typical educational environment this is seldom worthwhile. Unless special arrangements have been made, network and power failures are usually more likely to occur outside the computer than inside.

3.3 Spreading Processing Load

Moving e-mail from place to place requires each computer on the route to make decisions, usually based on the address to which the message is directed. Typically each step decides first whether to accept or reject the message then, if it is accepted, whether the mail can be delivered to a local mailbox or needs to be passed on. Mail addresses consist of two parts: the local part before the @ character and the domain part after it. Many mailhubs work only with the domain part, accepting messages for domains they serve or that they are allowed to forward, and leaving it to the final mailstore to either place the message in the correct mailbox for the local part or else to decide that the local part is invalid and reject the message.

However if an organisation has multiple mailstores, with different local parts corresponding to different mailstores, then the mailhub must examine the local part of the address to determine which mailstore should receive the message for filing. This means that the mailhub may well have a list of all valid local parts and can immediately reject messages addressed to unknown users without passing these to the mailstore machine. This can greatly reduce the load on the mailstore at little cost to the mailhub. The address look-up is a simple process and can be implemented efficiently with an indexed file. The benefits are particularly clear when the site suffers a mail flood with out of date or forged addresses. These can be rejected as soon as they reach the site, without loading the internal mail systems.

In fact the benefits are such that even sites with a single mailstore may choose to do the initial check for valid local parts on the mailhub. The check still needs to be done on the mailstore as well, since this may be the only place where local mail can be checked, but the reduction in load on that machine can be significant. This requires some automated process to maintain a list of all valid local parts on the mailhub. A few programs can build their own dynamic list by remembering the results of lookups, others make it relatively easy and quick to check incoming mail against a static list. It is not essential that the list be absolutely up to date. Many sites update it each night. This means that when users leave, there will be a period of up to 24 hours when the mailhub passes on messages that are then discarded by the mailstore using its more up to date list. When new users are created these addresses will not be able to receive mail from the Internet until the following day. Neither of these should be a major problem.

3.4 Hardware Choice

Mailhubs do not usually need to be high-performance computers: for most applications ordinary, or even second-hand, hardware will be sufficient. The system that is likely to have most impact on performance is the spool disk, though as most mail files are small it is the ability to create and delete files, rather than the speed at which bytes can be transferred to and from the disk surface, that limits performance. If the mail software supports it then it may be beneficial to spread the spool area across a number of small disks, rather than one large one, as this will normally increase the rate at which new files can be created. The total size of the spool area will limit how much mail can be queued if there is a problem elsewhere in the delivery system: the size required can be found by multiplying the peak hourly message volume by the length of outage that needs to be covered. In normal operation the spool area should be nearly empty as messages should spend very little time in the queues.

If the mailhub also performs virus checking or other types of content filtering, this can require substantial processing power and a fast Central Processing Unit (CPU) with sufficient Random Access Memory (RAM) will be required. There are a number of different ways to implement

filtering and these will make different demands on CPU and memory. Vendors of these types of products should be able to give an idea of the requirements of their own products.

The following sections describe a range of mail system configurations used by JANET sites. Although the hardware specifications are relatively modest, all the administrators regard them as more than sufficient for the current load and anticipated future growth. In each case the DNS MX configuration is shown below the diagram showing possible routes for inbound mail messages.

3.4.1 Minimal System

The smallest resilient mail system consists of two computers, a mailhub and a mailstore, with an offsite secondary mailhub to protect against local failures. The size of the mailstore depends on the number and size of user mailboxes, but the hub can easily be implemented on the type of PC hardware that has been retired from desktop use. This type of system can easily handle a load of 20,000 messages per week.

The offsite secondary function is provided by agreement by a mailhub at another site.

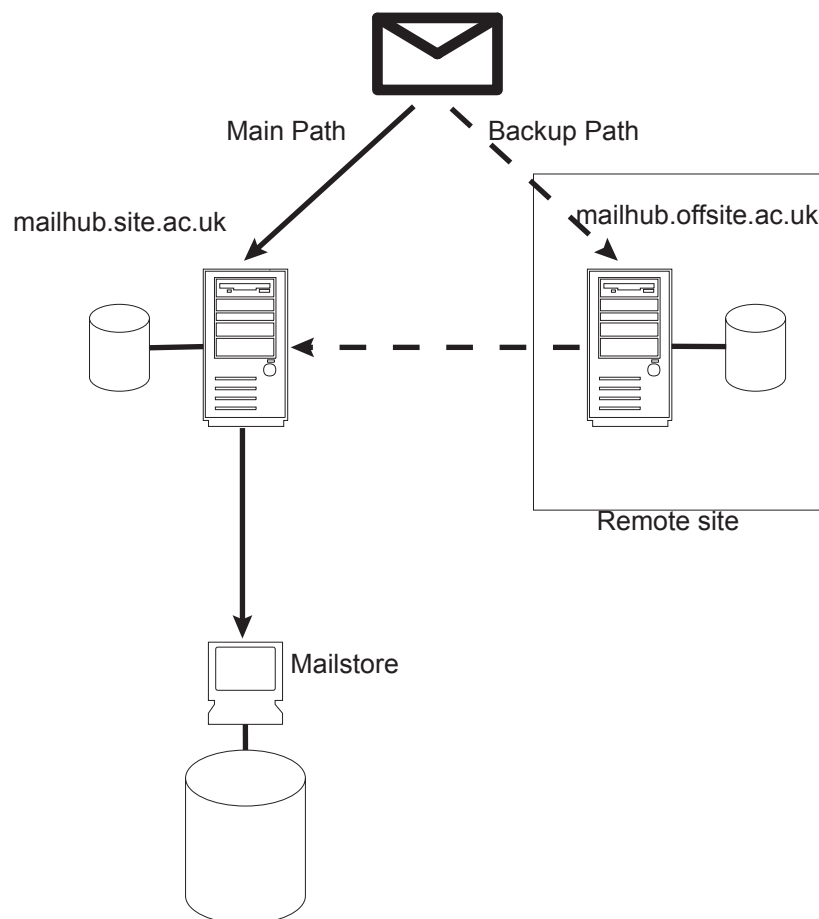


Fig 1: A Minimal System

DNS Configuration for minimal system:

site.ac.uk MX preference = 10, mail exchanger = mailhub.offsite.ac.uk

site.ac.uk MX preference = 5, mail exchanger = mailhub.site.ac.uk

3.4.2 Basic System

The next stage is to provide resilience locally by running a pair of mailhubs. For further resilience an offsite secondary could be added as fallback. Again, the hub systems do not need

to be particularly powerful. In normal operation the load of processing mail will be shared between them. If one hub suffers a failure then the whole load will fall on the other, but this should be tuned so that an overload results in a gradual reduction of the rate of delivery rather than a catastrophic failure (see section 4).

Using 'retired' workstation PCs, is more than sufficient for a small college. One example uses 500MHz Intel® Pentium III® computers with 128Megabytes (MB) of RAM and a 4Gigabyte (GB) hard disk running Linux and sendmail™ to provide a service to 400 local users, receiving about 5000 messages per week. At this load, one of the hubs can very easily handle all the traffic including virus checking each message.

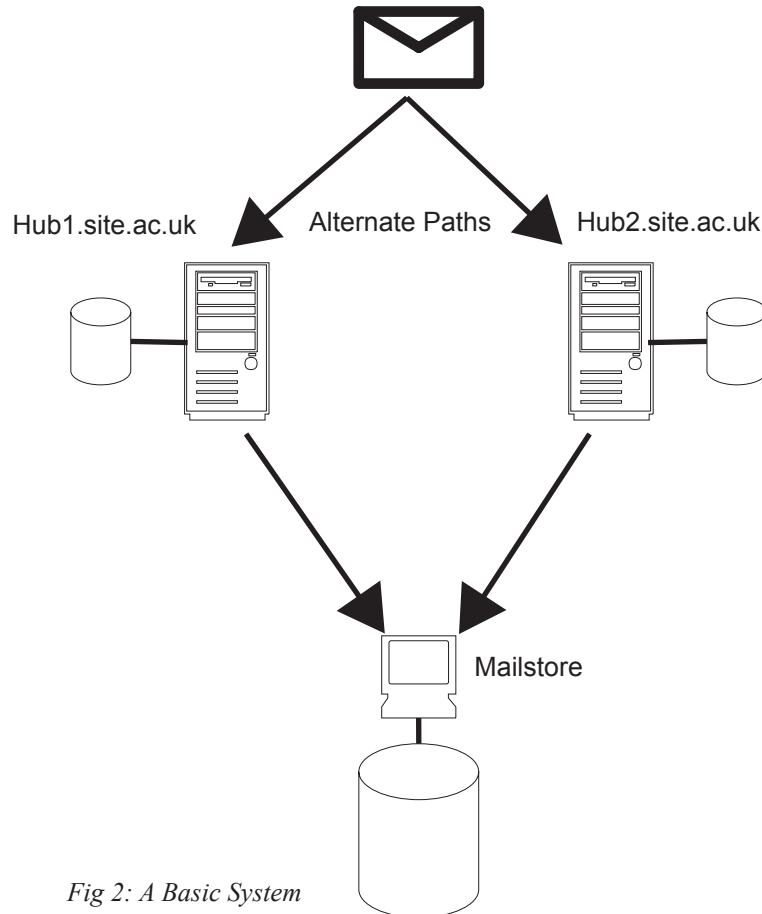


Fig 2: A Basic System

DNS Configuration for basic system

site.ac.uk MX preference = 10, mail exchanger = hub1.site.ac.uk

site.ac.uk MX preference = 10, mail exchanger = hub2.site.ac.uk

3.4.3 High-Performance System

For a high-reliability system serving a large university, PC hardware is still sufficient, but this should probably come from the server range, rather than the desktop. Software RAID is used to protect against failures of any one disk, with the mirrored disks connected to separate controllers so that the controller is not a single point of failure. With many thousands of end-user mailboxes it is likely that messages will need to be directed to many different mailstores. For resilience each of the mailhubs must be able to deliver direct to every mailstore.

For a large number of end-users using a multitude of mail clients the mailhubs are a particularly effective place to scan incoming messages for viruses and signs of junk mail. In the particular case considered here the site runs two independent virus scanners and one junk mail scanner on each incoming message. This requires much more CPU power than a simple mailhub but the dual-processor systems used are still not unusually powerful machines.

'In our case we take resilience through redundancy one step further and mirror all of the disks on our MX hosts. This approach need not be expensive. We recently upgraded our Mail Hubs to Dell™ 2550 boxes with 1GB of memory and dual 900MHz processors. They each have 4 x 36GB SCSI disks and dual SCSI backplanes. They each cost about £3,300. We had previously been running Sun™ boxes with Solaris™. We now run Linux (Red Hat 7.2) + Sendmail™ + MailScanner + SpamAssassin + Sophos + McAfee® on each Mail Hub which are clones of each other. We can easily cope with 18,000+ users and 400,000 incoming messages/week. We could probably deal with more than twice the message load with just two MX hosts.'

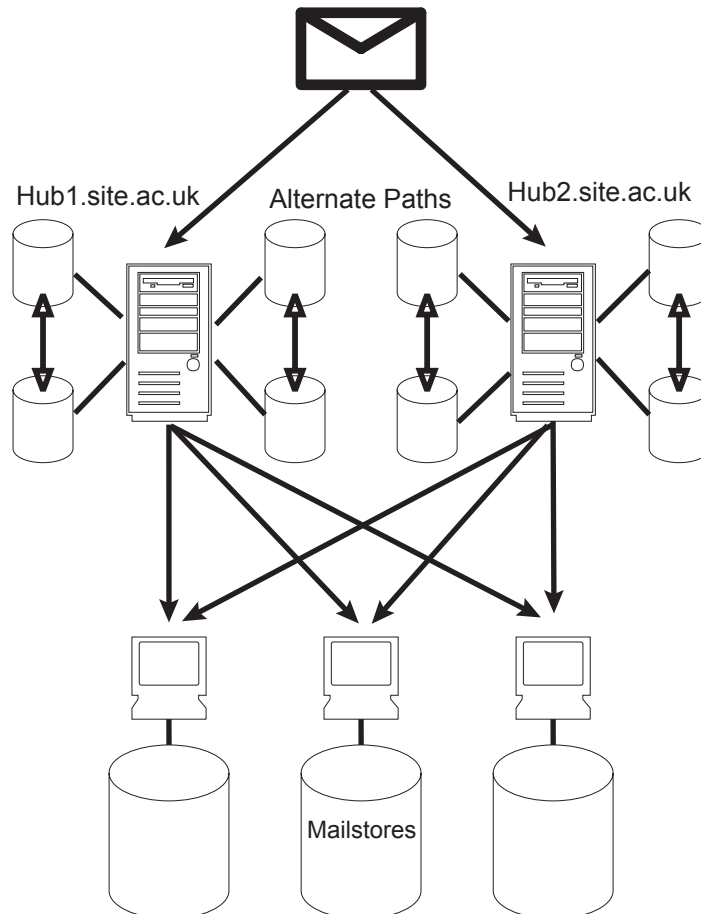


Fig 3: A High-Performance System

DNS Configuration for high-performance system

- site.ac.uk MX preference = 10, mail exchanger = hub2.site.ac.uk
- site.ac.uk MX preference = 10, mail exchanger = hub1.site.ac.uk

Mail Server Tuning

4

What to Tune

Tuning Exim

Tuning Postfix

Tuning Sendmail

Tuning Microsoft® Exchange

Tuning Novell® Groupwise®

Tuning Filters

4 Mail Server Tuning

One definition of a perfectly tuned system is one where everything fails at the same time. This is not the aim of tuning a mail system. Here the intention is to ensure that the load the system will accept is limited by documented software controls that cause performance to degrade gracefully rather than undocumented hardware or software capacity limits that result in a catastrophic failure.

When tuning mail servers it is important to remember that the mail protocol is inherently reliable, and that electronic mail is not a time-critical service where everything must be completed in seconds. If there is the slightest doubt as to whether the system can handle one more connection or one more message, the correct course is to return a temporary failure to the sending system, asking that it try again later. Limits should therefore be set conservatively, mail systems should always have some spare capacity and in normal operation will usually run just a little above idle level. Squeezing the last ounce of performance from a mail system may be exciting but it will not result in a reliable service.

Unfortunately many mail systems are delivered with either optimistic limits or no limits at all. A system running in this state will almost certainly eventually suffer an overload, with catastrophic results. Overloaded systems will usually either run out of memory and crash, or else create so many processes to accept and deliver mail that the processes starve one another of resources and the CPU spends all its time deciding which process to run next, a situation referred to graphically as ‘thrashing’. A well-tuned mail server will have limits set to keep it well away from either of these situations.

4.1 What to Tune

There are three limits that all mail systems should support, and which should be used to keep the system well below its overload limit:

- maximum number of simultaneous inbound connections;
- maximum number of simultaneous outbound connections (note that one incoming message, for example with a list of recipients, can result in multiple outgoing ones);
- maximum number of simultaneous local delivery attempts.

The third of these will normally apply only to mailstores, but mailhubs should also have a limit set to ensure that large numbers of error messages, for example, cannot cause the system to fail. All three of these limits control both the number of processes running on the system and the amount of memory used: each TCP connection uses up a significant amount of memory from what may, on some systems, be a pool of limited size. The correct value of each limit is something that will need to be determined individually for each mail system, however it is unlikely that anything over single figures will be efficient on any normal system.

In each case the resilience of the SMTP protocol allows the mail system to make the conservative choice. If a connection attempt would exceed a limit then it should be refused with a temporary failure; the message will still be held on the connecting system. If a message requires more processes or outbound connections than are currently available, then it should be held in a queue directory until resources become available. Disk space is unlikely to be a scarce resource on mail systems, memory may well be.

These three limits should be sufficient to deal with unusual numbers of mail messages. It may also be useful to have limits to deal with unusually large messages. For the extreme case every mail system should have a maximum size of message it is prepared to accept. This should be well above the likely size of any normal message (unless there are policy reasons to restrict message size) but should be less than the size of the spool area. Few systems can cope gracefully with a message that will not fit on the disk, and most will suffer from reduced performance while they try. If the system is also performing CPU-intensive tasks such as filtering, it may be useful to take account of the current CPU load to dynamically reduce the process limits. This protects the system against the situation where a number of hard to process

messages can starve others of resources. However the inverse operation, of relaxing limits when the system appears to have resources to spare, is not advisable. Mail traffic tends to come in bursts that can surprise systems that try to adjust gradually to circumstances.

4.2 Tuning Exim

Configurations for the Exim mail server are edited into the text configuration file. The location of this file varies, but can be found by typing the command:

```
$ exim -bP configure_file
```

This can limit the number of simultaneous processes that will be used to accept incoming mail, as well as the maximum number allowed to accept connections from a single host. The number of simultaneous processes permitted to deliver mail can also be controlled, but the documentation should be read with care to ensure that the combination of parameters is having the desired effect.

Parameter	Meaning
smtp_accept_max	Total number of simultaneous accepting processes
smtp_accept_max_per_host	Number of simultaneous local processes accepting from a single host
remote_max_parallel	Number of simultaneous remote delivery processes per queue-runner process (see Exim documentation)

Exim also has a variety of options to reduce functionality when the system is overloaded. Connections can either be rejected outright or accepted but with messages read into a pending queue which is not processed until the load drops. It is also possible to specify important hosts from which connections will be accepted even when these controls are imposed on others. Version 4 of the program introduces the ability to limit the rate at which connections will be accepted from a single host.

Exim also has a limit on the maximum size of a message.

Parameter	Meaning
message_size_limit	Maximum message size

Full documentation is available at:

<http://www.exim.org/docs.html>

4.3 Tuning Postfix

Configurations for the Postfix mail program are edited into the text configuration file main.cf. This has an overall limit on the number of simultaneous processes it will run, as well as individual limits on simultaneous delivery attempts to the same destination:

Parameter	Meaning
default_process_limit	Total number of processes
local_destination_concurrency_limit	Number of simultaneous local delivery processes
default_destination_concurrency_limit	Number of simultaneous remote delivery processes

The Postfix configuration files can also set the maximum number of each individual type of process so it is possible to explicitly restrict the number of simultaneous inbound connections. There is also an option to delay, or close down entirely, the acceptance of connections by the mail system when the rate of new connections or errors becomes excessive.

Postfix also has a limit on the maximum size of a message, as well as on various components of the message such as line length, number of recipients etc.

Parameter	Meaning
message_size_limit	Maximum message size

There are also limits on the sizes of various internal queues and numbers of objects in memory.

Full documentation is available at:

<http://www.porcupine.org/postfix-mirror/docs.html>

'The SPARCstation™ 2 (!) which is the mailhost for site.ac.uk copes fine. When the mail server was set up, we added severe limits to Postfix to limit concurrent deliveries and the maximum number of smtpd processes it would spawn. Looking at the current configuration, we only accept 3 concurrent smtp connections and have a maximum of 4 concurrent outgoing smtp connections & 2 local deliveries. For a beefier server, these can be tweaked up, but for the SPARCstation™ 2 they seem to be a good balance. OK, so it takes a while to clear the backlog after an outage, but the server does survive.'

4.4 Tuning Sendmail

The sendmail™ program uses a macro processor to generate its configuration file, so variables need to be defined in the input file to the macro processor (usually extension .mc) rather than the derived configuration file (.cf). If this is not done, their values may be lost when the mail software is upgraded or rebuilt.

Sendmail™ has a configuration parameter for the maximum number of processes that may be run. Each of these processes handles both acceptance and delivery of a message, so there is no independent control of these operations. However there is a further parameter that limits the number of connections a single process will accept per second.

Parameter	Meaning
MAX_DAEMON_CHILDREN	Total number of processes
CONNECTION_RATE_THROTTLE	Number of connections accepted per second per process

Sendmail™ can also be configured to respond to high loads (measured by the operating system load average parameter), by either refusing new connections completely or refusing to accept a new connection until one second after the previous one has completed.

Sendmail™ also has a limit on the maximum size of a message.

Parameter	Meaning
MAX_MESSAGE_SIZE	Maximum message size

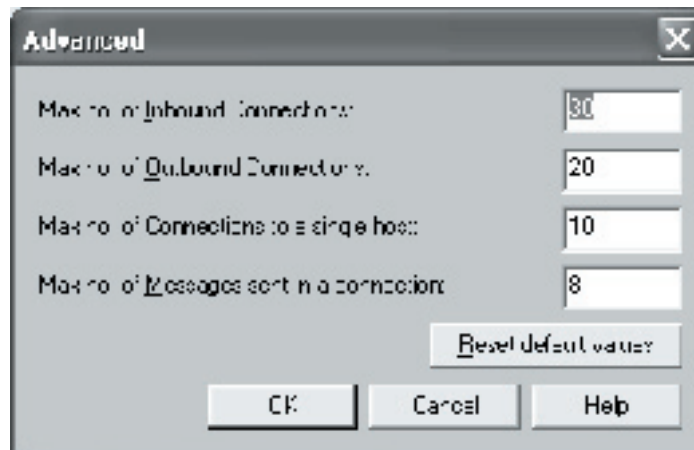
Full documentation is available at:

<http://www.sendmail.org/>

4.5 Tuning Microsoft® Exchange

Microsoft® Exchange has a graphical control program, which provides all the standard limits. Under Internet Mail Service/Connections/Advanced separate limits can be set for the maximum number of simultaneous inbound connections, the maximum number of simultaneous outbound connections and the maximum number of connections to a single host.

The maximum number of messages that can be sent down a single connection can also be set here (see Appendix A).



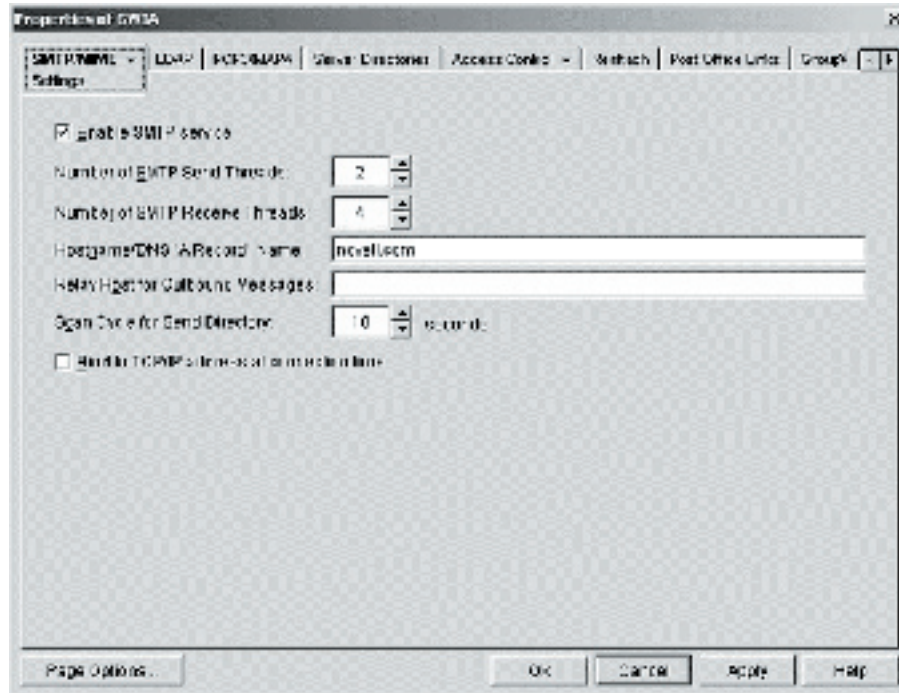
The maximum size of message that will be accepted is set on the Internet Mail Service *General* tab:



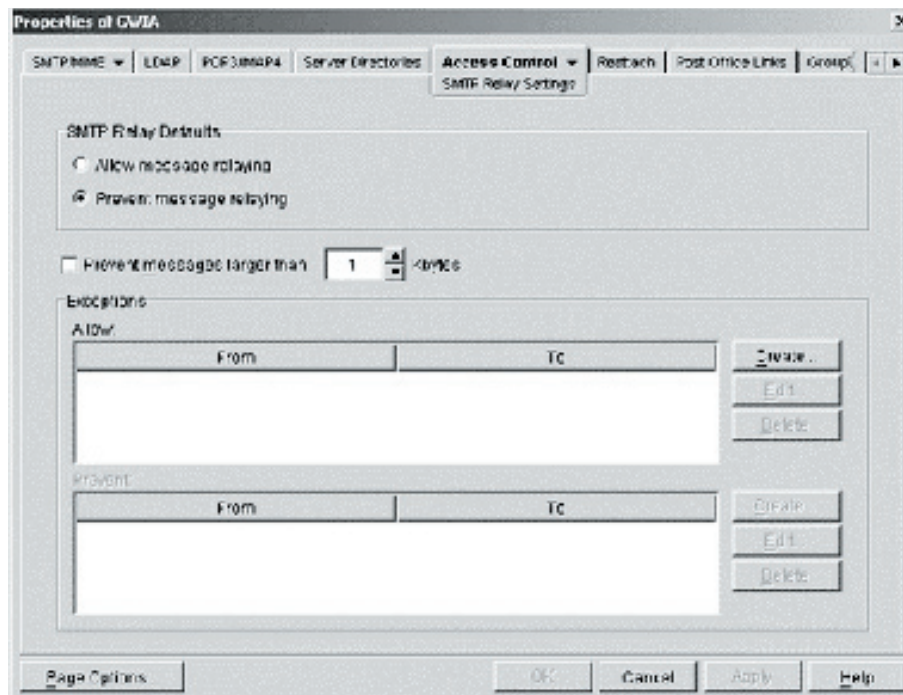
Screen shots of Microsoft® Exchange used by permission from Microsoft Corporation

4.6 Tuning Novell® GroupWise®

The Groupwise® mailserver is configured using the ConsoleOne® program. Right-clicking on the Internet Agent object and selecting *Properties* brings up a series of menus. To set the number of simultaneous threads available to read inbound SMTP messages ‘SMTP receive threads’ from the Internet or the number of simultaneous threads that will write outbound SMTP messages ‘SMTP send threads’, go to the SMTP/MIME tab and click on *Settings*:



Selecting the *Access control* tab and clicking on *SMTP relay settings* displays an optional limit on the maximum size of message that will be accepted by the software. Note that the default limit, fortunately not enabled by default, seems very small.



Screen captures included by permission of Novell, Inc.

Groupwise® can also set a maximum for the rate at which connections will be accepted from hosts. Groupwise calls this Mailbomb Protection and is controlled from the SMTP/MIME Settings tab and the Security Settings menu. Note that, unlike some other systems, this control does not limit the rate at which connections will be processed, but will simply reject all messages that arrive at too fast a rate. The control should therefore be used with care.

Full documentation is available from:

<http://www.novell.com/documentation/>

4.7 Tuning Filters

If the mailhub is running any additional software, for example to check for inappropriate content such as viruses or junk mail, then it is important to ensure that this cannot overload the mailhub. Content filtering can be a resource-intensive process, especially when checking large messages or attachments, and this should not be allowed to starve the main mail processing tasks of resources. An upper limit should be placed on the resources (CPU, memory and disk) that can be consumed by a single filtering process. This prevents the system being choked by a pathological message, but leaves the question of what to do when a message cannot be handled within the limits. Simply handing the problem to the user is probably not the correct solution so manual intervention is likely to be needed. Filtering controls may be set by the mail software, the filtering package or the operating system; different systems will have different options and recommended settings.

Emergency Measures

5

Restricting Clients
Rate Limiting
Making Things Worse

5 Emergency Measures

A well-designed, tuned mailhub should survive any mail flood, even if the network connection is swamped, however there is still likely to be some impact on users. This section considers additional steps that can be taken when emergencies occur to improve the situation.

Networks and mail systems usually treat all callers alike, however this is unlikely to reflect the actual situation when a mail flood is in progress. In many cases it will be possible to identify a small number of external hosts that require special treatment, either to discard them as being the cause of the problem, or else to treat them as higher priority because they are sending messages that must get through.

At the network level it may also be possible to identify higher or lower priority services: a mail flood may well fill a site's access link, stopping all other traffic, when in fact it would be preferable to limit mail traffic to allow other services to continue to work.

Controls of this kind involve setting priorities making some systems or services suffer to improve others. These decisions should not be made by technical staff in the heat of the moment, but must be planned in advance. Decide which are the important services that must be kept running, which can be sacrificed and which sites you need to remain in e-mail contact with. Ideally, this decision should be agreed with senior management, so that changed priorities are not imposed suddenly in the middle of an incident. Changes will often require particular features of the network configuration and the cooperation of other sites. If these are not available then the best option may be just to wait until the storm subsides.

5.1 Restricting Clients

When a mail flood occurs it is sometimes possible to identify a group of external IP addresses that are sending mostly flood messages, or at least a group that are sending mostly legitimate, and possibly important messages. If such a distinction can be made, then it may be useful to dedicate different mailhubs to handling connections from the two groups. Thanks to the SMTP protocol, this can be done very easily by blocking and/or permitting SMTP connections from the two groups at the site router. Provided all the hubs in use are advertised with DNS MX records, then mail connections will find their way to the 'right' hub. If a flooding host tries to contact the legitimate hub then its connection attempt will be blocked at the router and it will be forced by the protocol to use the other, emergency, hub. Note however that this will increase the number of connection attempts, placing a small increased load on the router and the external network. It is also likely that some legitimate mail will end up with the flood on the emergency mailhub, and these messages are likely to be delayed. Increasing the MX preference value for the emergency hub will reduce this risk, but the change may take several hours to propagate through the DNS system.

Before designating a mailhub as the recipient of the flood, it must be tuned appropriately for this role. Its ability to make simultaneous outgoing connections must be reduced, or possibly turned off entirely. The purpose of this machine to accept and store messages as fast as it can, to reduce the number of messages that have to be retried, and then to pass these on at a much reduced rate to the other mail systems. The number of messages that can be accepted must, of course, be limited below the system's disk and memory capacity, or else it may crash and make the situation even worse. If the messages that form the flood can be identified reliably, then it may be possible to program this hub to discard them silently, however this must be done in such a way that the sending host does not try the same message again later. Having the mail system reject messages other than with a 'permanent failure' response will just make the situation worse.

If it is not possible to distinguish between flood and legitimate messages based on the Internet Protocol (IP) address delivering them, then there may still be benefits in sending all mail to a specially configured emergency hub. For example if a site with a low-speed access link has an offsite secondary with better connectivity, then the load on the access link can be reduced if the secondary receives all mail and then forwards it at a reduced rate to the primary at the site. This transfers the impact of the mail flood, so must only be done with the informed

consent of the site where the secondary is located. Implementing this change is very simple: the primary site router should be set to allow SMTP connections only from the secondary hub. The secondary is likely to need either sufficient disk space to store all of the flood (which may be many hundred thousand messages), or else software and processing power to identify the flood messages and delete them.

5.2 Rate Limiting

When a mail flood occurs, it is quite possible for the volume of packets to fill a network link, thus disrupting all kinds of traffic and not just electronic mail. If this occurs on a link that is shared with user applications, for example the site access link, then the disruption to web browsing and other activities is unlikely to be acceptable. On a site access link, packet congestion will, of course, also affect outside users, making the organisation's website very slow or even disrupting the DNS on which all others depend. In the worst case a large mail flood could effectively disconnect the site from the Internet.

Some routers can limit the bandwidth available to particular sources or protocols. It may be possible to limit the impact of a mail flood using these controls but to be effective the control must be applied before the network link that is suffering from congestion. In the case of an access link, this router may not be controlled by the site. Applying rate limiting also places a load on the router, and if it is providing other services then these may be affected. The managers of the router may therefore be unable or unwilling to apply the limit for a single site. Applying a rate limit is also likely to increase the traffic on the networks upstream from the limit as it may cause connection attempts to fail and be retried.

One rate-limiting option that may be available to a few sites is to direct mail to a secondary mailhub with a low bandwidth connection. This reduces the load to the capacity of the primary link, but at the cost of sacrificing the secondary hub and its connection. The intention here is to create network congestion to act as a rate limit, in other words to damage the e-mail service in order to protect others. This should be a last resort when all other measures have failed.

5.3 Making Things Worse

When considering emergency measures, great care must be taken that they will not in fact make matters worse. The SMTP protocol applies the same diligence to delivering flood mails as to any other. Preventing mail connections from reaching one mailhub will simply transfer them to another, with the added network load of the failed connection attempts. If the secondary mailhubs are no better placed to handle the load than the primary then they will suffer as badly from the flood of requests.

One reaction that will not help deal with a mail flood is to simply refuse to accept messages. Many mail relays will continue to try to deliver mail for a week or more. Turning off a mail system just converts a mail flood lasting a few hours into a connection flood lasting several days.

Appendix A: A Brief Guide to SMTP

Network Layer

Routing E-mail

Failures, Retries and Disposal

Dealing with Mail Floods

Appendix A: A brief guide to SMTP

The Simple Mail Transfer Protocol (SMTP) is described in Request For Comment (RFC) 821 and a host of later RFCs. The use of the DNS for mail routing is described in RFC974. This appendix cannot hope to describe the whole protocol, but highlights some key aspects that are needed to tune a mail service, to understand, and even predict, what the effect of changes will be.

A.1 Network Layer

SMTP is implemented as a TCP protocol. This means that for a mail message to be transferred between any pair of hosts, a TCP connection must be established. Such a connection is established between two IP addresses, not two mailboxes or even two mail addresses; thus it is not possible for network devices to control the flow of mail traffic based on the sending and receiving e-mail addresses but only on the IP addresses involved in a particular connection.

Furthermore, it is unlikely that the workstation sending a mail message will be able to communicate directly with the mailstore where the message will eventually be delivered. This means that any message is likely to travel along at least two connections, first from the client to a mailhub and then from the mailhub to the mailstore. In practice there will usually be more connections, known as hops, than in this simplest case. Unlike the case of a web browser requesting a page via a cache, these hops are completely independent, and may take place some time apart. At each stage the receiving host accepts the message and stores it in a queue; at this stage the sending host may end the connection as it has no further role to play. At some future time the receiving host will process its queue of messages and may forward them on through further TCP connections. For this reason SMTP is known as a 'store and forward' protocol.

Setting up a TCP connection is a large overhead for a single message, so the protocol allows the sending host to transmit successively as many separate messages as it chooses along the same connection. These messages need not have the same sender or recipient e-mail address, and usually will not do so. The only condition is that they all have the same 'next hop' on the way to their destination. What a router sees as a single connection may well, therefore, contain many mail messages from different senders to different recipients.

The fact that messages progress by a series of steps means that there need be no obvious relation between the sender and recipient e-mail addresses and the computers involved in any particular SMTP connection. An e-mail from one business to another is very likely to involve a connection between two mail servers at their respective ISPs, neither of which server has any obvious relation to the addresses in the e-mail. It is thus very hard (in practice impossible) for a receiving host to determine whether a particular sending host should be handling mail to or from the particular pair of e-mail addresses. This gives a great opportunity for forging sender addresses, since only the originating system has any chance of determining whether a given user has the right to use a particular sender address. This situation is made somewhat worse by the fact that most implementations of the SMTP protocol in fact contain two sets of sender and recipient addresses: one pair, known as the envelope addresses, are used by the SMTP client and server processes to direct the routing of messages, whereas the other pair, contained within the text of the message, are the ones that are visible to the end users. When the protocol is used correctly, the two pairs should be the same, but this is not often checked. There are many reasons, a few good but most bad, for making the envelope addresses different from the content. The most obvious is to conceal, or even mis-represent, the identity of the sender of a message.

In fact, the only thing a mail server receiving a message can be sure of is the IP address of the immediately preceding host in the delivery chain, the one that requested the TCP connection down which the mail message came. Most mail systems add a *Received:* header to each message they process, this contains the IP address of the mail system itself and the IP address of the preceding host. Only the numeric address is reliable: the hostname may be temporary or forged. Of course a malicious mail server can choose not to add a *Received:* header, or to add headers containing false information. A malicious person can create a message that starts with a

series of false *Received:* headers to lay a false trail.

The lack of any authentication and the opportunities for forgery, make it very difficult to trace the source of a mail flood. Anyone attempting to lay blame for a flood must remember that the apparent source may itself be an innocent victim.

A.2 Routing E-mail

Any system involved in forwarding an e-mail message must therefore decide what is the most appropriate 'next hop' for the message, which host should it contact to try to move the message towards its destination. In most cases the only useful information available will be the domain part of the destination address. Some systems use the simplest possible rule. They forward all messages to the same, fixed, system. This is a commonly used rule for client machines that are the origin of the message. It may also be appropriate for passing mail through firewalls or relays. Since it is hoped that the next hop machine will have a more complex rule, it is normally referred to as the smarthost. The smarthost is normally specified as a host name.

A smarthost rule can be a fixed configuration entry in each client system, but more complex rules are likely to need dynamic configuration information. By convention DNS is used as the way to publish such information. The simplest case is where a DNS domain has no information specifically for e-mail. Mail messages must then have the name of a particular host at the destination site as the domain part of the destination address. Any mail system processing a message for this kind of address will simply look up the address (A) record for the host and make an SMTP connection to the IP address returned by the DNS lookup.

Both the smarthost and A record rules are somewhat more flexible than they appear, because they normally return a hostname, rather than an IP address. The DNS can be configured to choose an IP address from a list for a single name, thus allowing the duties of the receiver to be shared between a number of systems. However this provides only load-sharing, not resilience, for if a particular IP address does not accept SMTP connections the client has no means to obtain or use a different address.

Resilience, load-sharing and shorter mail addresses are among the benefits of the third method for finding next hops. This uses a specific type of record in the DNS, the MX record. Each MX record declares a given host to be a mail exchanger for some part of the DNS name space; this may be a hostname, a sub-domain or a full domain. Thus, unlike the A record method, which requires addresses of the form **user@mailhost.maths.camford.ac.uk**, MX records can also be defined to allow for addresses such as **user@maths.camford.ac.uk** or **user@camford.ac.uk**. A mail server processing mail for one of these addresses looks up the DNS for an MX record for the domain part of the address and uses the host returned as the next hop. If both MX and A records exist for the same domain, then the MX record takes precedence for mail delivery.

MX records also provide load-sharing and resilience because each record also includes a numeric preference value. If a mail server finds that a given domain has multiple MX records, then it is required to choose the one with the smallest numeric preference value. If this host does not respond then the next smallest value should be used. If there are a number of MX records with the same value, then one of these should be chosen with equal probability. Only once all the hosts with the same preference value have been tried and failed may a larger preference record be used. In the absence of other restrictions this results in hosts with the same value taking roughly equal shares of the load, while hosts with larger values will only be used as backups if the smallest value hosts are unavailable or overloaded. Note that only the order of the numeric values is significant: the actual values do not matter. MX records with preference values of 10 and 20 will not share the load in the ratio of 2:1 as might at first appear.

To prevent loops occurring, there is one additional rule. A host which is itself a mail exchanger for a domain can only forward messages by SMTP to another mail exchanger for the same domain if that host has a strictly smaller preference value. This means that the exchanger(s) with the smallest preference must be able to deliver messages by some other means: either direct to a local mailbox or through one or more smarthost rules to mailstores.

This last example illustrates the fundamental difference between smarthost and DNS rules. DNS rules are set by the destination domain and propagated globally through the DNS. Smarthost rules are set locally on individual machines and are outside the control of the destination domain.

A.3 Failures, Retries and Disposal

Sometimes, a hop will fail. The simplest type of failure is one that occurs at the TCP level. A system with a queued message will decide on the next hop, as described above, but will find it cannot complete a TCP connection to that host. The sending host may fail to make the connection at all, or the connection may be broken part way through. In both cases the message must remain on the queue and attempts to deliver it must continue. If the next hop was chosen on the basis of an MX record, and there are other MX records available for the destination address, then those hosts should be tried, in the correct sequence, as next hops. If a TCP connection cannot be established and completed to any of them, then the message remains on the queue and the attempts will be re-tried at a later time.

Alternatively the delivery attempt may fail at the SMTP level. In this case the TCP connection is established and completed successfully, but the host to which the connection is made uses the SMTP protocol to indicate that it cannot accept the message. The receiving host may be too busy or have no free storage space; it may not be prepared to accept messages for the given e-mail address, or may know that the address does not exist. The message may fail a virus or UBE filtering check, and therefore be unacceptable, or the host may have other reasons for rejecting the message. The SMTP protocol includes a number of different failure codes, and these can be supplemented by human-readable descriptions of the problem. The codes are divided into two groups: transient failures and permanent failures. A transient failure means that the receiving host expects to be able to accept the message at some point in the future (for example it is currently busy). A permanent failure indicates that the message is never likely to be acceptable (for example the username to which the message is addressed does not exist). A sending host that receives a transient failure reply should queue the message and try again later. A sending host that receives a permanent failure should not continue to try to deliver the same message but should attempt to notify the original sender of the message that there was a permanent failure.

It should be noted that a hop could, in fact, fail at an even earlier stage, if the sending host is unable to determine the IP address of the next hop. This can occur if the sending host does not have a smarthost rule to apply to the message and cannot find an MX or A record for the delivery address, or if the hostname of the next hop cannot be resolved into a numeric IP address. Any of these situations is likely to be considered an immediate permanent failure, so a robust DNS delivering correct information is vital for a robust mail system.

The time that a sending host will wait before repeating an attempt to deliver a message is entirely its own choice. The RFC821 standard merely observes:

‘It is difficult to assign a meaning to “transient” when two different sites (receiver- and sender-SMTPs) must agree on the interpretation. Each reply in this category might have a different time value, but the sender-SMTP is encouraged to try again.’

Nor can the originator of the message specify the retry time. In practice most mailhub software will wait at least a few minutes before repeating the delivery attempt, and will increase the period between successive retries so long as transient failures are received and until an ultimate disposal time limit is reached. However different mailhub administrators may well set very different retry and disposal policies on their systems.

Some messages will fail to reach their destination, either because a permanent failure is received or because repeated attempts to deliver them always resulted in transient failures and a disposal time limit has been reached. In this case the sending system that discovers the failure must try to inform the originator of the failure. It will usually do this by sending an e-mail message back to the originator. Most mail systems will include at least part of the failed message in this reply along with the reason for the failure, insofar as it is known. Since this message is delivered by normal e-mail process, it may itself be subject to delays. It is common

for the originator to learn of the failure of a message several days after it was sent if a system is suffering from a long-term transient failure. In the worst case, the error reply message may itself fail. In this case the system discovering this failure should try to deliver the message to the address postmaster at the originating domain. Some systems may also try the postmaster address at the destination domain. Only if all these attempts fail will the message eventually be lost without trace.

A.4 Dealing with Mail Floods

It should be apparent from the above that causing transient failures either at the TCP or SMTP level is the worst possible way to deal with a mail flood, since the retries will amplify and prolong the effect. The best strategy for the recipient site is to accept the messages and then silently delete them. Returning permanent failures will dispose of the initial flood but may result in a later flood to postmaster if the failure messages cannot be returned to the originator. This will occur if, as is common with unsolicited bulk mail, the sender addresses have been forged.

Appendix B: Glossary

Glossary

This glossary provides brief descriptions of some of the technical terms used in the document. Terms are in alphabetical order.

Address Forgery	Constructing an e-mail message so that the From: and To: addresses seen by the recipient are not those of the actual sender or recipient. Often the From: address will be forged to conceal the origin of a message, sometimes the To: address will be forged to conceal the list of recipients.
Blacklist	A list of hosts from which connections should not be accepted, or which should be treated as lower priority.
Collateral Spam	Messages (usually errors or complaints) received by a site whose domain has been inserted into a sender address during the process of Address Forgery. If the address to which the original message is sent does not exist, then error replies will be directed to the site whose address was forged, making them an innocent victim of the forgery. For more information see : http://www.ja.net/CERT/JANET-CERT/mail/junk/collateral.html
Content Filtering	Any process of examining the content of an e-mail message. Content filtering may be used to inspect messages for viruses or other unwelcome content defined in local policies. Filtering may be used to decide whether a message is accepted or rejected, or simply to mark the message so that the recipient may deal with it differently.
Disposal	Dealing with a message after too many retries ('too many' is defined by the mail system choosing to do disposal). Normally a non-delivery reply (NDR) will be generated as an e-mail message and sent back to the apparent originator of the message.
DNS	The Domain Name Service provides mappings between Internet names and numeric addresses and may also be used to distribute information about e-mail routing. The Domain Name Service functions as a distributed directory, containing millions of records. The term DNS may be used in different contexts to refer to the whole of the directory, the computer or software that provides a site's component of the directory, or the information in the directory.
DNS (A record)	Address (A) records are part of the Domain Name Service information that translates an Internet name into a numeric address.
DNS (MX record)	Mail exchanger (MX) records are part of the Domain Name Service information that indicates that a particular host is prepared to accept e-mail messages addressed to a given domain. MX records also include a preference field that is used to decide the order in which MX hosts should be contacted.

Domain(DNS):	Internet names are structured hierarchically, starting with a country or function (e.g. .uk or .com) and with each subsequent dot-word in the name indicating a deeper branch of the hierarchy. Each branch may have its own manager. The term domain is used for any part of this name-space so, for example, camford.ac.uk, ac.uk and uk are all Internet domains. Some domains may contain sub-domains within the same organization, for example science.camford.ac.uk might be a sub-domain containing, among others, the host mail.science.camford.ac.uk.
Domain Part	The portion of a mail address after the @ character. This normally identifies the mail system to which the message should be delivered and is the main piece of information used to route the message across the Internet (see also Local Part)
DoS	Denial of Service: a situation where a system is receiving more network traffic than it can handle, causing disruption or failure of its normal service. DoS attacks may be created deliberately, or may be a consequence of a mail flood or inappropriate tuning
Envelope Address	The e-mail addresses used in the SMTP protocol to indicate the sender and recipient of a mail address, and by each mail system to decide on the correct next hop for the message. Note that the envelope addresses may not be the same as the addresses in the text of the mail that are seen by the recipient.
Failure (Transient):	SMTP reply indicating that, although a transaction cannot be completed now, retrying the same transaction at a later time is expected to be successful. The numeric reply code for a transient failure begins with a 4
Failure (Permanent)	SMTP reply indicating that a transaction is unacceptable and that there is no point in repeating it. The numeric reply code for a permanent failure begins with a 5
Filtering	Any process that examines the headers or content of an e-mail message for purposes other than simple routing. Filtering may be used to reject or mark messages with particular characteristics and may be based on the system forwarding the mail, the headers or the content (see Content Filtering for this last option).
Forgery	See Address Forgery.
Hop	The transfer of a mail address from one mail system to another across a single SMTP connection, is referred to as a hop. Most messages will take several hops to get from the sender's e-mail client to the recipient's mailbox (see Store and Forward)
IP Address	The Internet Protocol uses numeric addresses to uniquely identify each computer directly connected to the Internet. Under the current version 4 of the Internet Protocol, each address consists of 32 bits, usually written as four decimal numbers, for example 192.168.0.1. Communications on the Internet are established between a pair of IP addresses, the source and the destination.
Load-sharing	Configuring more than one system to perform the same function so that the workload is shared between them (see also Resilience).
Local Part	The portion of a mail address before the @ character. This normally identifies a particular mailbox within a site mail system so is not usually of interest to other mail systems (see also Domain Part)

Macro Processor	A program that combines text files according to various programming instructions. The sendmail program uses a macro processor to expand a file of definitions supplied by the user into a configuration file used by the program.
Mail Address	The address of a mailbox on the Internet. All mail messages should have at least two mail addresses: one for the sender and one for the recipient. Addresses appear in the text of the message but are also part of the protocol used to transfer the message between mail systems (these are known as the envelope addresses). The envelope addresses should be the same as the text addresses, but this is rarely checked by mail systems, giving an opportunity for Address Forgery.
Mail Flood	An unusually large volume of e-mail messages that may overload the receiving site's mail systems
Mailhub	A computer whose primary purpose is to transfer e-mail messages from one place to another. Mailhubs may be advertised through DNS MX records: if this is done then the mailhub(s) with the smallest MX preference value are known as 'primary mailhubs', these will normally handle all incoming e-mail from the Internet. Other mailhubs with larger preference values are known as 'secondary mailhubs' and will not normally handle any incoming e-mail. Mailhubs are also known as Mail Transfer Agents (MTAs).
MailScanner	An example of a filtering program that can be used to check messages for viruses or against various UBE blacklists.
Mail Service	A combination of computers, networks and software used by a site to allow its users to send and receive e-mail messages within the site and to other locations on the Internet.
Mailstore	A computer whose primary purpose is to hold mailboxes for end-users.
Mail System	A computer forming part of the mail service. May be a Mailhub, Mailstore or perform some other function.
MTA	Mail Transfer Agent, see Mailhub. Also used for the software run by Mailhubs
MX	Mail Exchanger. The type of DNS Resource Record used to advertise computers that will accept mail addressed to a domain from the Internet. Sometimes also used to refer to one of those computers
NDR	Non-delivery report. A mail message generated by a mail system when it finds it cannot deliver a message. The NDR will initially be directed to the sender of the message, however when the sender address has been forged the NDR will be sent as Collateral Spam to the site whose addresses have been forged. If an NDR cannot be delivered, a report will be sent to the Postmaster at the apparent originating site.
Postmaster	The human who deals with problems with the mail service. The e-mail address postmaster is commonly used as a last resort for messages that have unrecoverable delivery problems.
Primary Mailhub(s)	See Mailhub

Queue	An area of storage within a mail system where messages are held temporarily, pending further processing. Queues are usually implemented as directories on disks; one of the aims of tuning is to ensure that mail systems handle gracefully the situation where a mail queue fills up as a result of a bottleneck. Also known as a Mail Spool.
RAID	Redundant Array of Inexpensive/Independent Disks. A range of architectures whereby multiple physical disks appear to the computer as a single disk with improved characteristics. Different types of RAID system can provide improved read or write speeds or increase the resilience of the disk system to the failure of a single physical disk. Different types of RAID system are distinguished by numbers, for example RAID 1 (mirroring) or RAID 5 (distributed striping with parity check). RAID can be implemented in software or hardware.
Rate Limiting	Setting an upper limit on the rate at which a particular process can take place. For example a mail system might have a rate limit that prevented it accepting more than 10 connections per second.
Recipient	Various fields in the e-mail headers and protocols are supposed to contain the e-mail address of the intended recipient of the message. However it is relatively easy for a malicious person to insert another e-mail address into some of these fields, or even to put information in the message that is different from that used by the protocol (see Address Forgery).
Resilience	Configuring more than one system to perform the same function so that workload can be transferred between them in cases of failure or other problems (see also Load-Sharing).
Retry	A repeat attempt to transfer an e-mail message after a transient failure has occurred.
RFC	Request for comments, one of the series of documents that define the standard Internet protocols. SMTP was originally defined in RFC821 and the use of DNS records to route mail in RFC974.
Secondary Mailhub(s)	See Mailhub
Sender	Various fields in the e-mail headers and protocols are supposed to contain the e-mail address of the originator of the message. However it is easy for a malicious person to insert another e-mail address into these fields, or even to put information in the message that is different from that used by the protocol (see Address Forgery). If a message cannot be delivered the mail system will normally return it to the sender as identified by these fields.
SMTP	The Simple Mail Transfer Protocol used to transfer electronic mail messages between mail systems across the Internet and many local area networks.
SpamAssassin	An example of a filtering program that rates messages according to their similarity to common characteristics of UBE.
Spool	See Queue.

Store and Forward	The SMTP protocol uses a store and forward design to move mail messages across the Internet. With this design, each mail system only concerns itself with delivering the message closer to its destination: when it has passed the message to another system its job is complete. On each system, the message is received, stored, and then forwarded to another system.
TCP	The basic Internet protocol that provides connection-oriented communications. The SMTP protocol is built on TCP.
TCP/IP	The set of basic communications protocols used on the Internet. Includes IP, TCP, UDP and ICMP.
Throttling	Reducing the rate at which traffic flows across a network. Note that throttling will often cause an increase in traffic elsewhere on the network as communications that fail because of the throttle will often be re-sent.
Tuning	The process of setting static or dynamic limits on a mail system to ensure that it handles overloads and failures gracefully.
UBE	Unsolicited Bulk E-mail is the most common term for large volumes of e-mail, often containing advertising material, sent to recipients who have not requested it.
Whitelist	A list of hosts from which connections should be accepted, or which should be treated as higher priority.

Tell us what you think

Technical Guides are produced and published by UKERNA for use within the JANET Community. We welcome your comments on all aspects of this document and on any other UKERNA publication. Please direct feedback to JANET Customer Service, at the address below, or e-mail us directly at:

documentation@ukerna.ac.uk

UKERNA manages the networking programme on behalf of the higher and further education and research community in the United Kingdom. JANET, the United Kingdom's education and research network, is funded by the Joint Information Systems Committee (JISC).

For further information please contact:

JANET Customer Service
UKERNA
Atlas Centre, Chilton, Didcot
Oxfordshire, OX11 0QS

Tel: +44 (0) 1235 822 212
Fax: +44 (0) 1235 822 397
E-mail: service@janet.ac.uk

Copyright:

This document is copyright The JNT Association trading as UKERNA. Parts of it, as appropriate, may be freely copied and incorporated unaltered into another document unless produced for commercial gain, subject to the source being appropriately acknowledged and the copyright preserved. The reproduction of logos without permission is expressly forbidden. Permission should be sought from JANET Customer Service.

Trademarks

ConsoleOne® is a registered trademarks of Novell, Inc. in the United States and other countries.

Dell™ is a trade mark of Dell Computer Corporation in the United States and other countries.

Intel® and Pentium III® are trademarks, or registered trademarks, of Intel Corporation or its subsidiaries in the United States and other countries.

JANET®, SuperJANET® and UKERNA® are registered trademarks of the Higher Education Funding Councils for England, Scotland and Wales. The JNT Association is the registered user of these trademarks.

McAfee® is a registered trademarks or trademarks of Network Associates, Inc. and/or its affiliates in the US and/or other countries

sendmail™ is a trademark of sendmail, Inc

Sun™ and Solaris™ SPARCstation™ 2 are trademarks of Sun Microsystems, Inc. in the United States and other countries

Disclaimer:

The information contained herein is believed to be correct at the time of issue, but no liability can be accepted for any inaccuracies.

The reader is reminded that changes may have taken place since issue, particularly in rapidly changing areas such as Internet addressing, and consequently URLs and e-mail addresses should be used with caution.



© The JNT Association 2003

**Joint Information
Systems Committee**