**LINUX ADMINISTRATION**

**Seminar Report On**

**MAIL ROUTING**

**Submitted by**

**Anagha P**

**Roll no: 10**

**S1, MCA**

**ITEC Palayad**

**How Does Mail Routing Work?**

The process of directing a message to the recipient's host is called routing. Apart from finding a path from the sending site to the destination, it involves error checking and may involve speed and cost optimization. There is a big difference between the way a UUCP site handles routing and the way an Internet site does. On the Internet, the main job of directing data to the recipient host (once it is known by its IP address) is done by the IP networking layer, while in the UUCP zone, the route has to be supplied by the user or generated by the mail transfer agent.

**Mail Routing on the Internet**

On the Internet, the destination host's configuration determines whether any specific mail routing is performed. The default is to deliver the message to the destination by first determining what host the message should be sent to and then delivering it directly to that host. Most Internet sites want to direct all inbound mail to a highly available mail server that is capable of handling all this traffic and have it distribute the mail locally. To announce this service, the site publishes a so-called MX record for its local domain in its DNS database. MX stands for Mail Exchanger and basically states that the server host is willing to act as a mail forwarder for all mail addresses in the domain. MX records can also be used to handle traffic for hosts that are not connected to the Internet themselves, like UUCP networks or FidoNet hosts that must have their mail passed through a gateway.

MX records are always assigned a preference. This is a positive integer. If several mail exchangers exist for one host, the mail transport agent will try to transfer the message to the exchanger with the lowest preference value, and only if this fails will it try a host with a higher value. If the local host is itself a mail exchanger for the destination address, it is allowed to forward messages only to MX hosts with a lower preference than its own; this is a safe way of avoiding mail loops. If there is no MX record for a domain, or no MX records left that are suitable, the mail transport agent is permitted to see if the domain has an IP address associated with it and attempt delivery directly to that host. Suppose that an organization, say Foobar, Inc., wants all its mail handled by its machine mailhub. It will then have MX records like this in the DNS database:

**green.foobar.com. IN MX 5 mailhub.foobar.com**

This announces mailhub.foobar.com as a mail exchanger for green.foobar.com with a preference of 5. A host that wishes to deliver a message to joe@green.foobar.com checks DNS and finds the MX record pointing at mailhub. If there's no MX with a preference smaller than 5, the message is delivered to mailhub, which then dispatches it to green.

**Mail Routing in the UUCP World**

Mail routing on UUCP networks is much more complicated than on the Internet because the transport software does not perform any routing itself. In earlier times, all mail had to be addressed using bang paths. Bang paths specified a list of hosts through which to forward the message, separated by exclamation marks and followed by the user's name. To address a letter to a user called Janet on a machine named moria, you would use the path eek!swim!moria!janet. This would send the mail from your host to eek, from there on to swim, and finally to moria.

The obvious drawback of this technique is that it requires you to remember much more about network topology, fast links, etc. than Internet routing requires. Much worse than that, changes in the network topology -- like links being deleted or hosts being removed -- may cause messages to fail simply because you aren't aware of the change. And finally, in case you move to a different place, you will most likely have to update all these routes.

The first step in disambiguating hostnames was the founding of the UUCP Mapping Project. It is located at Rutgers University and registers all official UUCP hostnames, along with information on their UUCP neighbors and their geographic location, making sure no hostname is used twice. The information gathered by the Mapping Project is published as the Usenet Maps, which are distributed regularly through Usenet. A typical system entry in a map (after removing the comments) looks like this:

Moria

bert(DAILY/2),

swim(WEEKLY)

This entry says moria has a link to bert, which it calls twice a day, and swim, which it calls weekly. We will return to the map file format in more detail later.

Using the connectivity information provided in the maps, you can automatically generate the full paths from your host to any destination site. This information is usually stored in the paths file, also called the pathalias database. Assume the maps state that you can reach bert through ernie; a pathalias entry for moria generated from the previous map snippet may then look like this:

moria ernie!bert!moria!%s

If you now give a destination address of janet@moria.uucp, your MTA will pick the route shown above and send the message to ernie with an envelope address of bert!moria!janet.

**Mixing UUCP and RFC-822**

The best cure for the problems of mail routing in UUCP networks so far is the adoption of the domain name system in UUCP networks. Of course, you can't query a name server over UUCP. Nevertheless, many UUCP sites have formed small domains that coordinate their routing internally. In the maps, these domains announce one or two hosts as their mail gateways so that there doesn't have to be a map entry for each host in the domain. The gateways handle all mail that flows into and out of the domain. The routing scheme inside the domain is completely invisible to the outside world.

This works very well with the smart-host routing scheme. Global routing information is maintained by the gateways only; minor hosts within a domain get along with only a small, handwritten paths file that lists the routes inside their domain and the route to the mail hub. Even the mail gateways do not need routing information for every single UUCP host in the world anymore. Besides the complete routing information for the domain they serve, they only need to have routes to entire domains in their databases now. For instance, this pathalias entry will route all mail for sites in the sub.org domain to smurf:

.sub.org swim!smurf!%s

Mail addressed to claire@jones.sub.org will be sent to swim with an envelope address of smurf!jones!claire.

The hierarchical organization of the domain namespace allows mail servers to mix more specific routes with less specific ones. For instance, a system in France may have specific routes for subdomains of fr, but route any mail for hosts in the us domain toward some system in the U.S. In this way, domain-based routing (as this technique is called) greatly reduces the size of routing databases, as well as the administrative overhead needed.

The main benefit of using domain names in a UUCP environment, however, is that compliance with RFC-822 permits easy gatewaying between UUCP networks and the Internet. Many UUCP domains nowadays have a link with an Internet gateway that acts as their smart host. Sending messages across the Internet is faster, and routing information is much more reliable because Internet hosts can use DNS instead of the Usenet Maps.

The only remaining problem is that the UUCP transport programs can't deal with fully qualified domain names. Most UUCP suites were designed to cope with site names of up to eight characters, some even less, and using nonalphanumeric characters such as dots is completely out of the question for most.

Therefore, we need mapping between RFC-822 names and UUCP hostnames. This mapping is completely implementation-dependent. One common way of mapping FQDNs to UUCP names is to use the pathalias file:

moria.orcnet.org ernie!bert!moria!%s

This will produce a pure UUCP-style bang path from an address that specifies a fully qualified domain name. Some mailers provide a special file for this; sendmail, for instance, uses the uucpxtable.

The reverse transformation (colloquially called domainizing) is sometimes required when sending mail from a UUCP network to the Internet. As long as the mail sender uses the fully qualified domain name in the destination address, this problem can be avoided by not removing the domain name from the envelope address when forwarding the message to the smart host. However, there are still some UUCP sites that are not part of any domain. They are usually domainized by appending the pseudo-domain uucp.

The pathalias database provides the main routing information in UUCP-based networks. A typical entry looks like this (site name and path are separated by tabs):

moria.orcnet.org ernie!bert!moria!%s

moria ernie!bert!moria!%s

This makes any message to moria be delivered via ernie and bert. Both moria's fully qualified name and its UUCP name have to be given if the mailer does not have a separate way to map between these namespaces.

Writing a pathalias file is acceptable only when you are running a site that does not have to do much routing. If you have to do routing for a large number of hosts, a better way is to use the pathalias command to create the file from map files. Maps can be maintained much more easily, because you may simply add or remove a system by editing the system's map entry and recreating the map file. Although the maps published by the Usenet Mapping Project aren't used for routing very much anymore, smaller UUCP networks may provide routing information in their own set of maps.

A map file mainly consists of a list of sites that each system polls or is polled by. The system name begins in the first column and is followed by a comma-separated list of links. The list may be continued across newlines if the next line begins with a tab. Each link consists of the name of the site followed by a cost given in brackets. The cost is an arithmetic expression made up of numbers and symbolic expressions like DAILY or WEEKLY. Lines beginning with a hash sign are ignored.