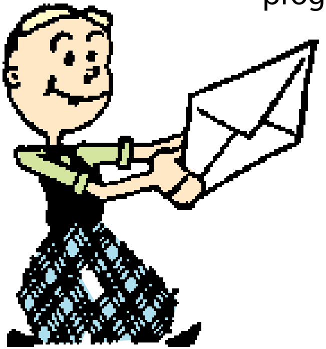
You’ve got mail!

How E-mail gets from here to there

According to virtually all studies of the Internet, tens of millions of people use E-mail every day. Some of these people have local mail systems, running programs like Microsoft Mail, CE Software’s QuickMail, or Lotus cc:Mail. Others are out on the Net, using online mail readers such as pine or elm, two UNIX-based terminal mail readers, or local mail clients like Netscape’s mail module, the Microsoft Exchange mail client, or Eudora. Millions more subscribe to online services that provide proprietary mail software, like America Online and CompuServe.



With all of these millions using mail, the issue of how it gets from one place to another is rarely addressed (as it were)—except when there’s a failure, and a major online service accidentally loses several hours or several thousand pieces of E-mail.

In the real world, it seems pretty clear how a letter gets from one house to another. The delivery address corresponds to a fixed geographical location. Postal codes, regardless of country, get the letter close to the final location, and the postal services of the world have divided and subdivided the planet into numbered pieces.

The virtual world doesn't have such a nice arrangement. Even though there is a topology to the Internet—routers and modems and computers do exist in discrete physical locations—that structure isn't mapped in a physical way. Instead, a separate structure sits inside the topological one, providing its own map for delivery.

Mail call

E-mail delivery sits on the back of the Domain Naming System (DNS), discussed in a couple of previous columns. DNS is a distributed system that contains information about machines at specific organizations; the data are managed by those organizations, and a few machines at the very top level point to the hundreds of thousands of machines that maintain this individualized information.

Part of DNS allows the numbering of specific machines; this is what connects your browser to a Web site. Another part, structured in a similar manner and at the same level, specifies which machines receive mail for which domains and subdomains.

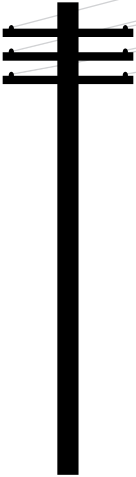
Mail records (called MX for mail exchange) are a kind of domain information; they tell remote systems where to deliver mail for machines within a given domain. A sample MX record looks like this:

```
lardo.com.          IN MX 10          pickle.lardo.com.
```

What this record says is, "Any mail addressed to '@lardo.com' should be sent directly to the machine called 'pickle.lardo.com.' " The "10" indicates relative priority. If a site receives a lot of mail, as, for example, a big commercial

Internet service provider would, that mail server may be busy. So several mail servers can be set up in a chain. For instance,

```
lardo.com.     IN MX 10          pickle.lardo.com.
               IN MX 20          chip.lardo.com.
               IN MX 30          dill.lardo.com.
```



This set of records says, "Try to deliver mail for '@lardo.com' first to 'pickle.lardo.com'; if that machine says it's too busy, try 'chip.lardo.com,' and then 'dill.lardo.com.' " For big sites, this kind of cascading structure is necessary to keep the site from losing mail and causing clogs elsewhere.

The process

Electronic mail uses domain names as a means of specifying the "postal code" of the final address; the user ID or name is a bit like a combination of the name of the person to whom you're sending the message and their street address. Domain names are not quite as unique as street addresses; where you'd be hard pressed in most places to have two Bill Joneses in the same house, you could have 10,000 Bill Joneses on America Online, a service catering to 6 million people (and the world's largest boardinghouse, maybe). CompuServe and MCIMail, to name two, usually assign unique numbers to avoid this problem, but who wants to be known as 740744.1342@compuserve.com? (This is a fake number, but you get the point.)

What happens next is less like letter carriers walking a route, and more like two very expert operators reading a letter to each other over the phone. The process starts with whatever "client" or local software you use for mail—the simplest example would be a program like Eudora, which is a POP (Post Office Protocol) client. Let's take the example of two people who are both using Eudora at different locations.

1. Using the local mail function, the sender composes a pieces of mail, queues

it, and then sends it.

2. Eudora connects to a "mail server," a program that handles mail coming and going. Eudora uses SMTP (Simple Mail Transfer Protocol) to send the E-mail. This process is a little like call and response. Eudora says, literally, "Hello!" (spelled HELO in the four-letter codes that SMTP uses), and in a series of predefined steps, transmits the sender's address, any recipient addresses, and the body of the message.
3. The mail server takes the message it's received and examines it. It figures out which part is the domain (pretty easy, generally), and then looks to see where mail for this domain should be delivered using DNS.
4. The sender's mail server then tries to open a direct connection to the recipient's main mail server. If it's busy, it'll try the next mail server, if any. If it can't connect to any of them, it queues the mail and tries later.
5. Once the sender's mail server opens the connection to the recipient's mail server, the process in step 2 is repeated, virtually identically. The information

is passed, with usually a little more included in the "headers," describing the various steps in the operation with time and date stamps.

6. The recipient's mail server adds the incoming message to the queue awaiting delivery for that user. Generally, the queue for delivery is just a text file; new messages get appended to the end of that text file until the user retrieves the queue. The local mail program he or she uses chops the big text file into individual mail messages again.
7. The recipient checks his or her mail. Eudora connects to a POP server, which has the sole function of relaying mail from the user's queue to the local machine the user is on.
8. The recipient reads the message!

This process changes for people using systems like cc:Mail or Prodigy's E-mail system only in that more complex activity takes places between steps 1 and 2. Proprietary mail systems, as opposed to the open POP and SMTP standards, usually have their own mail servers that distribute and receive mail, and pass on messages intended for addresses outside their own local systems to a gateway that actually speaks the SMTP language.

Drop it in the slot

Although E-mail is one of the simpler functions of the Internet to explain, it has a myriad of technical underpinnings. For instance, the configuration of "sendmail" servers is a thing of wonder and mystery, in which the symbol "@" can mean five different things on the same line of the configuration file depending on the context.

Fortunately for all of us, the function of E-mail so far outweighs the method that it's one of the primary benefits of the growing size of the Internet. I recall getting my first E-mail from Finland back in 1991, and being pretty amazed—partly because it only took a few minutes to send a message and get a reply back.

Now it's old hat to get mail from France, South Africa, my parents, and India all in the same day. What's that about the global village, again?