1

Background and History

While many people use electronic mail (hereafter "e-mail") in today's internetworked world, comparatively few people are concerned with the details of its delivery. This is as it should be. Just as we don't have to know the details of "physical mail" (or postal mail, hereafter, "p-mail") to be able to use it, we would normally prefer to go about our business of using e-mail without knowing its inner workings. In e-mail, as in p-mail, only the Postmasters have to know how it works. Sendmail is a tool of the e-mail Postmaster, and it has handled more e-mail than any comparable tool. Sendmail is to e-mail as delivery vans, airplanes, sorting machines, letter carriers, and all the rest of the postal system are to p-mail. Even if you had received only one e-mail message in your life, chances are better than even that Sendmail had a hand in its delivery.

1.1. A Brief Introduction to Sendmail

On the Internet, the majority of servers run the UNIX operating system or some derivative thereof (such as BSD, Linux, SunOS, ULTRIX, HPUX, or OSF/1). Sendmail is the program most of these systems use to handle e-mail. Some systems include alternate mail transport agents such as MMDF (which ships with SCO UNIX) or SMAIL (which ships with Interactive 386/ix), but even on these systems you will very likely find Sendmail lurking in a corner somewhere, deprecated by the vendor but available if you want it.

Sendmail was written by Eric Allman to handle the problems of address mapping between heterogeneous mail and network environments and to "help bridge the gap between the totally ad hoc world of networks that know nothing of each other and the

clean, tightly coupled world of unique network numbers.''[1] Sendmail is able to route or deliver mail between just about any mail user agent (hereafter ''UA''—any one of many programs you might use to read and send e-mail) and message transfer agent (hereafter ''MTA''—a program that moves mail between hosts using a particular network protocol or language). One of Sendmail's design goals was to easily accommodate the addition of new UAs and MTAs with only minor configuration changes (see Figure 1-1).[2]

Sendmail supports distribution lists in the form of ''aliases'' for people or sets of people. It supports the use of individual user `.forward` files to allow forwarding of incoming e-mail to programs or to other mailboxes. Sendmail also facilitates the rewriting of e-mail addresses to allow for gateways—which deliver mail between different kinds of mail networks—and to provide a mechanism for bridging various gaps between different systems. We will see examples of this later.

Further, Sendmail provides for message queuing when a retryable error is encountered (e.g., when a host or a network is temporarily unreachable), automatic routing, and returning the e-mail to the sender when an unrecoverable error is encountered (such as mail addressed to nonexistent users or hosts).

1.2. History

In the beginning, there was no such thing as IP/TCP or Ethernet or the Berkeley UNIX operating system (''BSD'') or the Internet as we know it today (2001). Words had 16 bits, and we felt lucky to have that many. Networking meant serial (RS232) links in which ''high speed'' was 9600 baud, or it meant simplex coaxial point to point links at a whopping 2 megabits. The wide area networks of the time mostly used X.25 or SNA, and most traffic on them was between timesharing hosts and remote terminal users.

''Peer to peer'' communication, as it was called then, meant that programs on one computer would speak directly to programs on some other computer; the protocols could be designed for programmers and programs rather than for humans and teletypes. Peer to peer was a distant ideal to most programmers, since the vendors were competing to provide proprietary visions of the future; the only way you could get one vendor's computer to talk to some other vendor's computer was with serial links and programs like Kermit.

Even in the unlikely event that all the ''peers'' you wanted to speak to were on other computers made by the same vendor as yours, you still had to learn your

[1] Allman, Eric, ''SENDMAIL—An Internetwork Mail Router (1983).''

[2] Sendmail is also an MTA as it contains an implementation of the SMTP protocol.

vendor's unique programming interface to their networking layer. Woe would frequently visit unto thee if your computer did not have a direct link to whatever other computer you wanted to communicate with, since this would imply that some computers in the network had to be gateways and nobody, either vendors or system administrators, had enough expertise about running gateways. (This is arguably still true.)

One of the interesting lost technologies of this era was Berknet. Berknet was a web of then called "high speed" serial asynchronous links. The old, pre-POSIX BSD `tty` driver had a line discipline for Berknet that allowed a user level process to send and receive full speed serial data without flow control or special character interpretation. Berkeley Sendmail included Berknet support until R5 (1993).

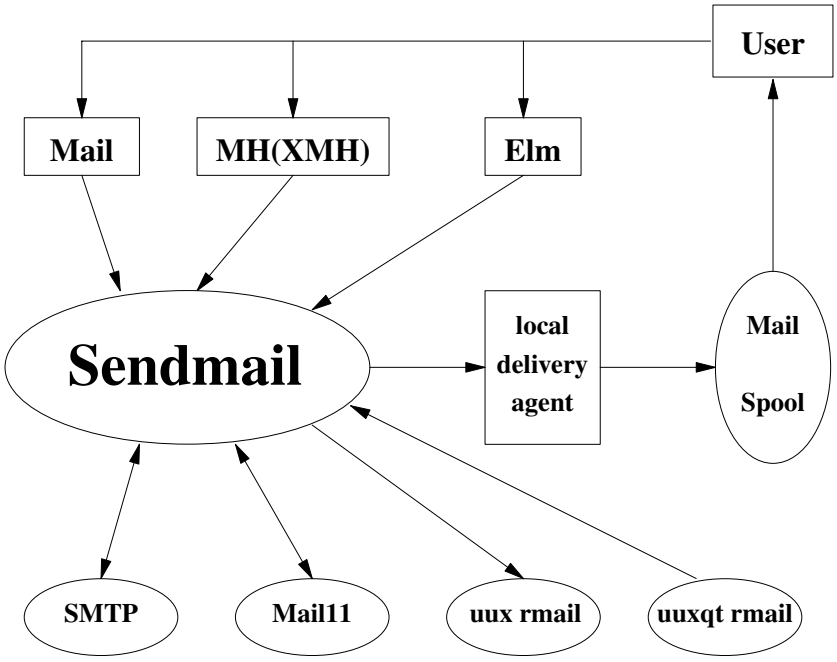

Figure 1-1. Sendmail's External Data Flow

This situation was quite grim. But inevitably the technology got cheaper and faster and bigger and better, and things that had once seemed to be pie in the sky began to look implementable. The biggest single influence on subsequent events was, obviously, the UNIX operating system and the C programming language. These two artifacts had the same effect on minicomputers (and, later, workstations and some mainframes) that CP/M had had on the 8 bit microcomputers of the 1970s: vendors were no longer encouraged to develop their own proprietary programming environments but, rather, to ‘‘port’’ the common UNIX and C environment to their computers so that customers could theoretically buy ‘‘iron’’ from any vendor and get the same basic look and feel. Programs that ran on any UNIX computer could supposedly run on any other UNIX computer. This was only a theoretical ideal because vendors being only human, there were always local ‘‘enhancements’’ that made your code unportable if you used them; if you did not use them, your code ran poorly or not at all.

In parallel to the growing popularity of UNIX and C, computers were getting faster, memories were getting larger, network pipes were getting fatter, and everything was getting cheaper. Programmers everywhere were getting their hands on all this new hardware, along with the UNIX source code, and new and improved versions of UNIX and its subsystems began sprouting up all over. One such sprout occurred at the University of California at Berkeley, which was more ambitious and better organized than most other places and so its version of UNIX actually got a name—BSD—and was widely distributed to other UNIX license holders.

Meanwhile, the United States Department of Defense Advanced Research Projects Agency (ARPA) had sponsored the creation of a wide area network connecting its various research contractors for the purposes of general research collaboration and the basic network specific research that later led to IP and TCP. This network was called ARPANET and is second only to UNIX and C in its influence on the modern computer industry. ARPA sponsored the programmers at U C Berkeley's Computer Science Research Group to port their BSD UNIX software from the PDP-11 to the VAX, and then to add networking capability so that BSD UNIX computers could connect directly to the ARPANET, which by this time was switching from NCP to TCP as its reliable stream protocol.[3]

A few years later, ARPA declared its ARPANET experiment a success and decided to stop funding it; the National Science Foundation wanted more research in wide area internetworking and, further, decided that many of its grantees would benefit from being connected to a wide area IP/TCP network. Thus was born the NSFNET,

[3] Historians will note several gross oversimplifications in this account, but it's close enough to explain where Sendmail came from.

which was originally proposed as a way to link the NSF Supercomputer Centers together but became the original backbone of what we now call the Internet. After that came, in no particular order: commercial Internet carriers, the web, NASDAQ, and spam.

1.2.1. Sendmail Arrives

Given that background, it now makes sense to declare that Sendmail came out of the BSD effort at the UCB Computer Science Research Group (CSRG). Sendmail was written by Eric Allman, who also wrote the `syslog` subsystem and contributed code and ideas to many other parts of the BSD UNIX system.[4] Sendmail was originally an expanded and generalized version of Eric's previous mail transport, `delivermail`.

The `syslog` facility was originally part of Sendmail and used only by Sendmail. This is evidenced by the fact that in 4.2 BSD the `syslog` file (which contains the actual logged data) was stored in Sendmail's queue directory. Because `syslog` was such a general tool, it migrated into the base BSD system and is now used by almost all UNIX systems for logging of background daemon activity. Several other BSD UNIX library calls also trace their origins to Sendmail.

1.2.2. Sendmail 8.*

In the mid 1980s, Allman's involvement with Sendmail seemed to wane. Several others picked up the mantal for a time. At the University of Linkbping in Sweden, Lennart Lövstrand released IDA Sendmail later further supported by Neil Rickert of Northern Illinois University and Paul Pomes of the University of Illinois. IDA Sendmail added the concept of external databases to Sendmail. Paul Vixie, then at Digital Equipment Corporation, created KJS (King James Sendmail).

As Eric Allman and Greg Shipley describe things, ''In late 1989, Allman returned to U.C. Berkeley, and not long thereafter was drawn back into Sendmail development. By July of 1991, serious work on what would become Sendmail 8 had begun. Many ideas were taken from IDA Sendmail and KJS ... Sendmail 8.1 was released with 4.4 BSD in mid-1993. Sendmail 8 quickly became a unifying influence, as ven-

[4] Including the TROFF ''me'' macro set used to typeset this book.

dors converted from their hacked versions to the newer version."[5] At the time of writing these words, we are up to release 8.11. Versions 8.8 onward of Sendmail, are, among other things, responses to the changes in Internet use and the resulting Internet environment. The Internet of today, especially related to Internet mail, is very different than just a few years ago. No longer are users assumed to be at fixed desktops within the walls of an enterprise, or connected via telephone to a central modem bank. They are more likely to be connected through an Interent connection from their homes or a cyber cafe in a foreign city.

The Internet of today is also an Internet where "spammers"—senders of junk e-mail—commit rampant theft of service by randomly dumping unrequested, usually unwanted, e-mail on uninterested recipients. Because of the volumes of junk e-mail often automatically sent in an individual e-mail transaction, some corporate e-mail gateways crash or are otherwise rendered unusable under the load.

This is also an Internet of potential eavesdroppers. E-mail communication between gateways was never safe from the eavesdropper. The growth and availability of Internet connections has raised the importance of protecting the exchange of e-mail between end points from disclosure. So, Sendmail 8.10 supports SMTP client authentication, SMTP transfer confidentiality (through TLS), an API for e-mail filters, other anti-spam features, and a host of new e-mail standards.

The changes has also spawned a new company. Sendmail, Inc. was cofounded by Allman to produce a commercially supported Sendmail product while still supporting the interests of the open source community, and the open source availability of Sendmail.

1.3. Summary

In summary, some hippies at Bell Labs spawned UNIX and C, which in conjunction with the technology curve spawned BSD UNIX, which in conjunction with the ARPA and ARPANET and IP/TCP spawned VAX BSD, which contained the first freely redistributable implementation of the IP/TCP protocol stack, including a set of working applications. Among these was Sendmail. That gets us to 1983 or so. In the mid-1990s, Sendmail was even ported to Microsoft Windows NT.

So, why are we still using Sendmail? Given Sendmail's many suboptimal design features (an obscure configuration file, monolithic "one program does it all" packaging, extreme ease of misconfiguration, catastrophic failure modes, and so on), why are we still using it? Why, so many years after Sendmail was first written, are we

[5] Greg Shapiro and Eric Allman, "Sendmail Evolution: 8.10 and Beyond," Proceedings of the FreeNIX Track: 1999 Usenix Annual Technical Conference, Monteray, CA, June 6—11, 1999.

writing a book about it? Why, after so many years, are you reading this book and using Sendmail?

Inertia, partly. Sendmail comes free with every modern UNIX system, which makes it a fairly attractive way to solve the average computer's mail transport problem. Also, the cost of switching has yet to be lower than the cost of living with things as they are. There are alternative mail transport agents such as Postfix, PP, and SMAIL, and these each have strong and loyal user communities with more or less active ongoing enhancement and development.

However, in the opinion of many people (including those who choose the mail product strategies for the major UNIX vendors), the alternatives to Sendmail all have worse problems. Some are too large and have an even larger configuration matrix than Sendmail's (MMDF and PP); some are too small and have no configuration variables at all other than what you choose at compile time (earlier versions of SMAIL). And then there's simple inertia. Sysadmins generally already know how to cope with Sendmail and we expect to see it inside every server we install.

Sendmail, once you get it working, works *really well*. Unless you run a large mail gateway, you don't have to spend much time watching or reconfiguring Sendmail. This, combined with the fact that Sendmail comes with almost every UNIX system sold, makes it a solid winner with no relevant competition. Here, perhaps, is an instance of ''good enough'' being the enemy of ''the best,'' but we can't argue with the results.

As this edition went to print, Sendmail 8.12 was released, continuing the commitment to enhanced security and better performance.