Ifmail: FidoNet(tm) technology implementation on UNIX platform

Eugene G. Crosser
Sovam Teleport
Moscow, Russia
crosser@average.org, http://www.average.org/~crosser/

Abstract

Fidonet(tm) is an amateur wide area computer network, mostly based on computers running MS/DOS and compatible systems. Ifmail program was designed, first, for running a Fidonet node on a UNIX machine, and second, for gatewaying Fidonet traffic to and from the Internet.

1 Architecture of the Fidonet

Fidonet is an network carrying email–like and netnews–like traffic over, mainly, asynchronous telephone links. The technology is very similar to UUCP.

The minimal chunk of information in Fidonet is a "message". It can be either Netmail (analog of RFC–822 message) or Echomail (Analog of RFC–1036 news article). One or more messages can be bundled into "packets", and one or more packets can be compressed into "arcmail files", that are just archive files produced by a compression/archiving utility, such as pkzip.

Fidonet technology also allows generic files transfer.

Packets, arcmail files and other files can be transferred from one node to another during "sessions" using a variety of file transfer protocols, variations of Xmodem, Zmodem and a few full–duplex protocols commonly used on Bulletin Board Systems.

The entities in Fidonet that exchange messages are "nodes" and "points"; difference between them is more political than technological. The node address has three 16 bit numbers: Zone, Net and Node. Point uses four 16 bit numbers: Zone, Net, Node and Point. Address can also have a "domain" - up to 8 ASCII characters.

Netmail message consists of a binary format header, including the addresses of the originating and destination systems, the names of the sender and of the recipient, subject and some auxiliary information, and the text of the message, that can also contain so called "kludges" - logical extensions of the header.

Echomail message contains, in addition to the above, the name of the discussion group it belongs to ("areatag"), its path (ordered list of the nodes it passed thru), and a "seen-by" list (sorted list of the nodes it was exported to by all of the nodes that it passed thru).

Fidonet standards are defined in "FTSC" (Fidonet Technical Standards Committee) documents.

More information about Fidonet and its technology is available at

`http://www.fidonet.org/`

2 Implementation Rationale

A majority of the Fidonet systems are running two pieces of software: "mailer" which

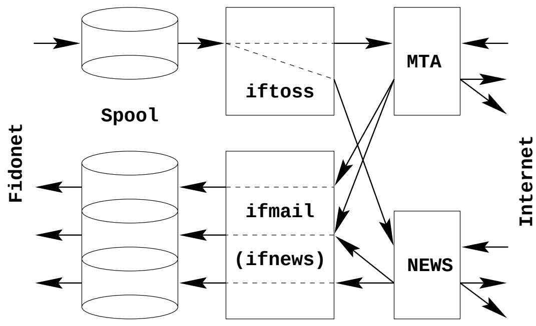

Figure 1: Gateway Structure

is the transport agent capable of establishing sessions to other nodes and sending/receiving files, and "tosser/packer" responsible for unpacking and parsing received files. It takes care of routing Netmail messages and exporting Echomail messages to other nodes and prepares files for sending.

On the contrast, ifmail does contain a transport agent, but does not contain the part responsible for routing and distribution of data. Instead, it has a gateway that converts Netmail messages into RFC-822 e-mail messages, Echomail messages into RFC-1036 news articles and vice versa. Then, existing UNIX programs, such as Sendmail and INN, may be used to manipulate the information and distribute it to other systems, in both Internet and Fidonet, as shown on figure 1.

The obvious advantage of this approach is that it avoids duplication of effort and allows using existing well tested programs to do part of the job. It follows the fundamental concept of the UNIX system of separating the task into smaller pieces, each proceeded by independent program.

But this approach also has some significant drawbacks, mostly caused by substantial differences between the concepts laying in the background of mail and news systems in the Internet and Fidonet. Some of these will be provided below.

3 Transport agent: ifcico

The transport part of the ifmail package is a single program; its functionality, as well as its name is modeled after uucico. When invoked in the "master role", it scans the spool for outbound data and calls appropriate fidonet systems to transfer the data. When called up in the "slave role", it accepts incoming Fidonet sessions. In order to accept incoming sessions, the program should either be started by init, to wait for incoming connections on a serial port with a modem connected to it, or (more often) invoked by uugetty when the latter detects Fidonet type incoming call. For this scheme to work, uugetty must be capable of recognizing Fidonet calls. This is necessary because Fidonet sessions do not start presenting username and password, but rather with a special sequence of bytes. Some modern uugetty programs, like "mgetty", "getty_ps" and the uugetty that comes with FreeBSD have this capability built in.

Ifcico is also able to originate connections over TCP/IP, and can be executed in "slave mode" by inetd.

As already mentioned above, ifcico transmits data to other Fidonet systems from the outbound spool directory. Data received from other systems is stored in the inbound spool directory. The structure of these directories was designed after a popular Fidonet "mailer" program - BinkleyTerm. This way it is possible to use ifcico transport program with the third party Fidonet compatible software, both UNIX and MS/DOS based, that also supports BinkleyTerm style spool.

4 Gateway: ifgate

4.1 Basic Concepts

There is one basic idea that makes design of ifmail's gateway different from the other existing Fidonet-to-Internet gateway programs: it is transparent. Meaning that an e-mail message or a news article when passed from Internet to Fidonet and back, keeps all the essential information. The same applies to messages originated in Fidonet, passed to the Internet and back to Fidonet. Obviously it sounds good, but many other existing gateway programs does not comply with this principle.

Two most important parts of a message to preserve (aside from the content) are Message I.D and tracking information (Path and/or Received headers and their analog in Fidonet). But in ifgate, more general approach was chosen. All "system" information of Fidonet message that does not naturally map to RFC–822/RFC–1036 headers is encapsulated into custom headers with standard prefix, "X–FTN–". When an Internet message is passed to Fidonet, and this message contains X–FTN–... headers, these headers are converted into corresponding elements of the Fidonet message.

The same way, RFC headers of an Internet message that cannot be mapped to the elements of an Fidonet message, are converted into so called "kludges" that are special lines in a Fidonet message designed to carry information invisible (by default) to the end user.

There are two technical solutions in the gateway that worth special discussion: Name–Address conversion and Message–ID conversion.

4.2 Name–Address Conversion

In the Internet sender's and recipient's addresses are "user@domain" format, usually accompanied by a comment containing free form name, such as:

```
Paul Jones <pjones@acme.com>
```

In the Fidonet addresses are numeric, and, when written in plain text, look like this:

```
Paul Jones of 2:345/67
```

There is a commonly accepted way to denote Fidonet addresses in the Internet domain notation: for example, the above address will look like this in the Internet:

```
<Paul.Jones@f67.n345.z2.fidonet.org>
```

Fortunately, there is no problem to map Fidonet address into Internet domain format. But there is a problem to specify Internet address in the Fidonet. It is common practice to specify the Fidonet address of the gateway in the Fidonet message, and either put the Internet domain address into the name field, or hide it in a special "kludge".

Both ways have drawbacks: the name field in the Fidonet address is limited by standard to 36 characters, which may be not enough for a domain address, and if the address was embedded into a kludge, not all user-end software will be able to produce a proper "reply" to a message.

Ifgate uses another way. It keeps a DBM database with free form names of the Internet correspondents as keys, and their domain addresses as values. When a message is passed from the Internet to the Fidonet, its sender's name and address is stored in the database and when message arrives from Fidonet to the gateway, name field is matched against this database and the message goes to the corresponding domain address. Of course, this scheme fails miserably when another person appears with the same free form name but different address.

4.3 Message–ID Conversion

Another tough problem arizing when Internet messages pass to Fidonet and vice versa is Message-ID to MSGID conversion. MSGID is a standard "kludge" in a Fidonet message designed for the same purpose as Message-ID in the Internet. But while the Internet Message-ID has the format:

```
<local-part@domain>
```

where "local-part" is almost any ASCII string, Fidonet MSGID has the format:

```
address hexadecimal-number
```

where "address" is usually a Fidonet style address "Z:NET/NODE.POINT" and "hexadecimal-number" is an arbitrary 32 bit number in hexadecimal notation.

It is obvious that Fidonet MSGID can be easily mapped to Internet Message-ID, but not any Message–ID can be mapped to MSGID without information being lost.

Ifgate when producing the MSGID uses domain part of Message-ID for the address part of MSGID and 32 bit hash of the local part of Message-ID for the number part of MSGID. In addition, like other headers the original Message-ID is stored into a separate kludge.

4.4 Caveats

There are a few problems coming out of the ifgate design.

First, it is processing of cross-posted articles. In Fidonet a message can only belong to one "echo area" and a message cross-posted to several echo areas are actually multiple messages, with different MSGIDs. That's why, Usenet articles posted to multiple newsgroups have to be converted to multiple Fidonet messages on the gateway.

Now imagine a Fidonet node subscribed to a single newsgroup at the gateway. Article arrives to be posted into multiple newsgroups, including the one node is subscribed to. The news system will pass a single copy of the message to the gateway to export into the fidonet system. As the gateway program has no information to which newsgroups the destination system is subscribed, it will produce multiple copies of the message, one for each newsgroup, and pass them all to the destination system. The destination will thereby receive several unwanted copies. This effect could be avoided if the distribution process was not separated from the gatewaying.

Another Message-ID related problem arises when a big Usenet article is being split to several Fidonet messages (because the maximum size of the message in Fidonet is limited). The parts must have different Message-ID's when they are gatewayed back to Internet and it is possible that the original article and the same article split to chunks are presented at the same time, which may be confusing.

There are several other similar problems that seem impossible to avoid. Although, these problems are non fatal and the technology works well in most cases.

5 Acknowledgments

People who provided me with reports, helped to find the bugs and contributed the code to the program are too numerous to mention them all, but I am greatly obliged to all of them. It is their continuous support that allowed to make the program useful for many people.

6 Availability

Ifmail is "Open Source" software; it is free for use, distribution and modification. The source is available via anonymous FTP from

`ftp.average.org/pub/ifmail/`

Information is also available on the WWW at

`http://www.average.org/ifmail/`

There is a mailing list dedicated to ifmail, see the WWW page above for the subscription address.