Network News
From the trenches

Vikas Aggarwal
vikas@jvnc.net

March 22, 1991
Last Update: September 13, 1991

Contents

1 Overview **3**

2 "Theory" **3**

2.1 News Flow 3

2.1.1 Receiving/Creating news articles 4

2.1.2 Installing articles 4

2.1.3 Transmit article(s) 4

2.2 Expiry & History 5

2.3 Subscriptions vs. Distributions 5

3 Installation Details **5**

3.1 Directory & Disk Layout 5

3.2 C-News 6

3.2.1 lib files 6

3.2.2 sendbatches 7

3.2.3 inews 8

3.2.4 Receiving news via email 8
3.2.5 Miscellaneous 9
3.3 NNTP 9
3.4 RN 9

4 Administration **10**
4.1 Control Messages 10
4.2 Moderated Newsgroups 11
4.3 Monitoring & Status 11
4.4 expire 11
4.5 history 12
4.6 sys 12
4.7 rn 13
4.8 Miscellaneous Tips 13

A Programs in News **15**

B Mailpaths **15**

C News Flow analysis **18**

D sendbatches diffs **20**

E New site questionarre **21**

Abstract

Network News (*C News & NNTP)* has been installed at JvNCnet. This document describes the installation, maintenance, features and problems associated with it.

1 Overview

C News has been installed on newsserver.jvnc.net [1]. The intent is to localize and structure the news flow within the regional network by providing a central news-hub for sites within JvNCnet. Even though there are a number of major news sites within our regional network, providing a news-feed is a resource consuming process and it is intended that the regional network try and provide a centralized "shared" resource for its members.

There were two alternatives to the news software - *B News* and *C News*. B-News evolved from the original A-News version (1979), whereas C-News is a major rewrite of the software at the University of Toronto. C-news is *supposed* to be much faster (by an order of magnitude for some low-level programs) than B-news.

This document describes some 'useful' details in setting up *C news*, news-readers and NNTP on a Unix system. It provides tips on news maintenance and describes a complete setup for new network News setups.

2 "Theory"

News articles are similar to mail messages (SMTP format)[RFC1036], with some additional fields identifying the group, expiry period, etc. One of the important fields is the Message-ID field, which is unique (within the entire news network) for every news article. This message ID field is stored in a `history` file which serves to uniquely identify the article to prevent duplication. All articles are stored under a simple directory hierarchy in `/var/spool/news` (typically). Thus a message for *jvncnet.tiggers.announce* will be stored in the directory `./jvncnet/tiggers/announce` as a numbered file (under the spool directory `/var/spool/news`).

An article is *"injected"* into the news flow by simply sending it to the next news site, which in turn sends it to the next site, and so on. A loop is prevented by searching for the *Message-ID* field in the `history` file and discarding the article if it exists in this file.

2.1 News Flow

This section gives an example of an article flow thorough the news engine. The process is divided into three parts for clarity:

1. Creation and input of article(s) into news (receiving news).
2. Install article locally and mark to forward (or relay) to other news sites.
3. Transmit the news article to other sites.

[1] Use nslookup to find the CNAME entry to use in the nntp_access file.

Beside these processes, an expire program periodically deletes old articles.

2.1.1 Receiving/Creating news articles

An article is created using a user-freudly program like Pnews or postnews. Once the article is complete, it is injected into the news flow by feeding it to inews.

In case a batch of articles arrives (via any means like email, nntp, uucp), the batch is fed through rnews which simply copies the entire batch into an `in.coming` spool directory.

At this point, the article(s) have arrived and are stored locally on disk but have not yet been installed on the system (not been copied over to the articles directory hierarchy, etc.).

2.1.2 Installing articles

The articles are installed on the system and added to the list to articles to be relayed to another site by the relaynews program. This program is invoked by the newsrun shell script which simply uncompresses the batches (if necessary) and feeds them to relaynews. This program looks at the `sys` file which lists what newsgroups are acceptable locally and which newsgroups are to be relayed to other news sites.

The last field in the `sys` file indicates the method of transmission– if the previous field indicates that it is a file, then the directory and number of the article is stored in `out.going/filename`. Otherwise, the command in the last field is directly invoked with the news article as stdin.

The `history & active` files are also updated by the relaynews program. In case an article belongs to a newsgroup that is not in the `active` file but is accepted by the `sys` entry is filed in the *junk* newsgroup and forwarded to the downstream sites. However, articles belonging to newsgroups that are not accepted by the `sys` file are *not* filed and discarded.

2.1.3 Transmit article(s)

The articles that have arrived on this system (whether from another host or generated on this host using Pnews) have to be relayed to the other neighboring netnews hosts.Articles can be transmitted using any of the following methods:

- via mail (mailnews– unbatched or viaemail– batched)
- via UUCP (uux) which then invokes rnews at the remote site
- via nntp (nntpxmit) to another NNTP server

The entire cycle of receiving, installing and transmitting is repeated at the other news sites.

2.2 Expiry & History

A database of all articles that are installed on the system or have been recently received (and possibly deleted) is maintained by news. This database is called the `history` file and serves to prevent a late arriving duplicate (of an existing or expired article) from being re-injected into the system and thus cause a loop.

Entries for incoming news articles are added to the `history` file by the relaynews program (which installs articles in the respective directories). Old news articles are deleted/purged from the news system by the expire program. expire looks at the `explist` file to decide how long articles of a particular newsgroup should be kept on the system. It also deletes very old references to articles from the `history` file, thus trimming its size.

expire also updates an `active` file for the sake of news reader programs. This file lists the highest and lowest number of the articles for each group that *exist* on the system. When a newsreader (*e.g* rn) starts up, it typically consults the `active` file to see which articles exist on the system.

2.3 Subscriptions vs. Distributions

A *subscription* is a newsgroup to which a particular article belongs. An article can also contain a DISTRIBUTION: header line which is used to alter the distribution scope of an article. If this line exists in an article, then the article is sent to only those sites which subscribe to the specified newsgroup *and* one of the specified distribution.

Thus, an article sent to the *misc.forsale* newsgroup, with a *distribution* header of *ny,pa* will only be sent to sites that get the *misc.forsale* **and** either of the *ny/pa* distributions. Newsgroups and distributions are both specified in the `sys` file.

3 Installation Details

This section explains minor/confusing details in setting up C-news and other news related programs. It also describes the author's experience regarding directory layout, maintenance and administration.

3.1 Directory & Disk Layout

There are three broad categories of files involving news:

1. The binaries/configuration files for news. These files do not change in size, and the *cnews* binaries take about 1MB (excluding the `lib` directory– binaries only), *nntp* takes 0.5MB, *rn* about 0.6MB.

2. The news articles installed by realynews on the local system (typically under `/usr/spool/news`). Keep this filesystem separate, and while creating the file system, try and increase the density of inodes in the file system by reducing the bytes per inode [2].

   The *size* of the filesystem depends on the number of newsgroups that you receive and the expiry time for the various newsgroups. Just for rough estimates, if you are planning on getting a full feed then look towards allocating 250MB or greater.

3. Log files, databases and in/out-bound batches. The `lib` directory in *cnews* contains `history` and log files which vary in size. Also, incoming batches need to be stored temporarily on disk before they can be unbatched, and the `outgoing` directory for the neighboring sites has files which list the articles to be transmitted (the `togo` file) and varies in size.

   A typical hourly batch is about 0.5–1.5MB, the size of each `togo` file is about 150KB and the *cnews* `lib` directory takes up about 35MB of disk space. Putting all of these on a separate file system with about 75MB (or greater) served well for the author.

   Note that putting these files on a separate filesystem than the articles' directory (`/usr/spool/news`) prevents a possible *catch-22* situation where the articles cannot be unbatched since the `history` file cannot be rebuilt which in turn causes incoming batches to stall and so on.

3.2 C-News

Available from ftp.cs.toronto.edu via *anonymous* ftp. Most of the configuration is easy and done using the `conf/build` script.

Once installed, if you are following the directory and disk layout as described above, move the `lib` directory to the separate filesystem (if any) and create a link from the original location to the new location.

3.2.1 lib files

Take a look at the `lib/server, lib/whoami & lib/mailname` files to make sure they are what you want.

active Get an `active` file from your friendly news neighbor. Edit it to remove "obviously" undesirable newsgroups and run maint/adddirs to create all the directories. Update the `active` file by running "expire/upact" & "expire/recovact". Run "maint/act.to.times" to create an `active.times` file.

Edit the `explist` file to match what you want (to control expiry of old articles and trim the `history` file). Pay particular attention to the last field in the *bounds* line which controls unconditional expiry.

[2] On a SunOS 4.1 workstation, the command is: *newfs -i 512*

sys Edit the sys file to control what you get. Create an entry for your upstream site(s) and add the same newsgroups that you are getting from them since you have to transmit locally generated news articles back to them. If using batched transfer mode, then write out the list of articles to a `togo` file.

mailpaths It lists the email addresses of moderators of moderated groups. Get an updated `mailpaths` file from your news neighbor (periodically posted in the *news.admin* newsgroup). Also see the appendix.

newsgroups Gives details of various newsgroups. Can get it from the same location as the `mailpaths` file (see appendix) in file `List_of_Active_Newsgroups`. It is used in user programs to give a brief description about a specified newsgroup.

3.2.2 sendbatches

Typically, news that is sent out in batches at a hub site is spooled in an outgoing directory (`out.going/site`). At periodic intervals from the crontab, sendbatches is invoked which looks at the `batchparams` file and sends the articles via mail or uucp (as prescribed by the last field in the `batchparams` file).

The standard viaemail script for sending out batched news via mail sends the news out to enews on the remote host (one cannot specify a specific user). An "undocumented" (or perhaps overlooked) feature of sendbatches is that one can specify any program for sending a batch off, *and store that program in the site's out.going directory*– sendbatches sets its path to include the site's *outgoing* directory before trying to run the sender program. Thus, in each `out.going/site` directory, a file batchemail can be created which mails out a batch to the specified user:

```
(echo;sed s/\^\~/ \~/) |/usr/ucb/mail -s "News Batch (`date +%m%d%y,%H:%M`)" user@host
```

Then this program name has to be specified as the *sender* in the `batchparms` file for sendbatches to invoke. Note that tilde's at the beginning of a line have been replaced with a space since the Berkeley mailer treats that as a command, but potentially one could encode the batch using bencode and then mail it out.

This method has been used by the author to mail news batches to remote sites that cannot run NNTP (including UUCP sites with good mail setups). The receiving side simply feeds the batches into recenews or rnews.

3.2.3 inews

There were 4 versions of inews between the C news and the NNTP distribution; and with the documentation describing each's purpose, no wonder they need gurus. The following were the versions available:

`nntp/inews/inews`– A *mini-inews* in the NNTP distribution
`cnews/relay/sh/inews`– Main inews program with the *C-news* distribution

Also, the following exist in the `cnews/contrib/nntpmail/nntp_support/` directory:

`frontend_inews`– Post articles by invoking the NNTP's mini-inews
`server_inews`– For running on the NNTP news server.

Of these, the `server_inews` program is derived from the main C-news `relay/sh/inews` program and can be ignored.

To ease the confusion, the NNTP inews was installed as mini-inews, and its reference in the frontend_inews program was correspondingly changed to reflect this change in name. The `cnews/relay/sh/inews` program was installed as the main inews program that the NNTP server invokes.

1. Install `cnews/relay/sh/inews` as the distribution installs it. `nntpd` should use this program (the INEWS define in the conf.h file)
2. Edit the `frontend_inews` file and change:

   `NEWSBIN/nntp/inews` to `NEWSBIN/nntp/mini-inews` (towards the end of the shell script)

   Then install `frontend_inews` as inews in `/usr/local/bin` where Pnews and other user related programs can find it easily.
3. Finally, install the `nntp/inews/inews` program (the one that came with the NNTP distribution) as mini-inews under NEWSBIN/nntp so that frontend_inews can find it.

3.2.4 Receiving news via email

Articles can be received via electronic mail and then injected into the news flow also. You should set up the mail `aliases` file so that mail to *rnews* be fed into rnews and mail to *enews* be piped into relay/recenews.

3.2.5 Miscellaneous

Create a `.forward` file so that all mail to *news@your.host* gets forwarded to *news-admin*. Also, create a mailing alias for *news-admin* which should point to a real human account.

Edit your crontab to run the lines in the sample `conf/cron`. Run expire after midnight (see Administration section below).

3.3 NNTP

Available via *anonymous* ftp from lib.tmc.edu. Most of the configuration is set in the `conf.h` file. Make sure that the definitions of SPOOLDIR, INEWS, RNEWS match that in the C-news distribution. Take special care to define DOMAIN and HIDDENNET appropriately.

Note that the `lib` directory in both NNTP and C-news can be the same (link them to each other if needed).

Install the `inews/inews` file as mini-inews as discussed in the C-news section above.

Edit your `/etc/syslog.conf` to log syslog info in the desired file. The author found it convenient to specify the news `lib` directory itself for creating the syslog files. Edit the root crontab to cleanup these syslog files.

Edit the `lib/nntp_access` file which limits access to your NNTP server.

nntpsend Articles generated at your site also need to be transmitted to the other NNTP hosts. For transmitting news to another site using NNTP, nntpsend is used (which in turn uses nntpxmit).

Note that this program is "undocumented" and "hidden" in the `nntp/xmit` *directory.* The shell script was modified by the author to use the same directory structure as the C news sendbatches program. The changes include looking for a `togo.nntp` file in the `out.going/`*site* directory hierarchy.

The diffs listing the changes made are in the appendix.

Edit the crontab to invoke nntpsend every hour with the list of NNTP neighbors as the arguments to the script.

3.4 RN

A popular news reader program available from bcm.tmc.edu via *anonymous* ftp. Presently maintained by *sobbcm.tmc.edu*

Installation is easy using the Configure script. Note that rn needs its private library directory

(RNLIB). The inews used by rn is the `frontend_inews` installed by C-news.

rn can access the news articles using NNTP (in which case it is known as rrn) or directly from disk (possibly using NFS). Note that to post articles, they can either be posted using NNTP (frontend_inews) or by directly invoking the main inews. It is preferable to use the NNTP method since the other method might require inews to do an rsh to the remote host and it could fail.

Note that rn/Pnews simply look for an inews to *post* articles– since the version of inews that it finds in its path uses NNTP, the posting is done using NNTP. However, while *reading* news, if compiled as rrn it uses NNTP to communicate with the remote host but if compiled as rn, it reads articles assuming that the articles are on local disk)or NFS mounted).

Cancel an article using the 'C' option. This sends out a control message to cancel a previous article.

Typically, users use Pnews or rn to post or reply to news articles. These programs then call inews which just creates and checks headers of the article and decides whether to feed it to relaynews (if it exists) or to mail it off to a news hub or to mail it to a moderator of a news group.

You might want to edit the newsetup shell script to subscribe to only a few newsgroups and mark all others unsubscribed (rn calls this program to create a `.newsrc` file for new users). Else, all the users will have to go through the rigmarole of answering whether they want to join a newsgroup for about 1200 newsgroups.

4 Administration

This section contains topics related to news administration. It also discusses some disaster recovery methods which have been used and have worked for the author.

4.1 Control Messages

Control messages are ordinary news articles with the special header line CONTROL: in them. The SUBJECT: line in the article contains the control command and its arguments. These commands are: ihave, sendme, newgroup, rmgroup, cancel, sendsys, senduuname & version.

These articles are not checked in the `active` file, so can belong to any newsgroup/distribution. In fact, the combination of the newsgroup and distribution is used to limit the scope of the control message.

Control messages must be sent very carefully with a careful selection of newsgroups and distribution. An unjudicious use of these messages could generate a lot of network related traffic, and you could end up with more harm than good. A relay site which "leaks" out to newsgroups could potentially relay the control message intended for a single host to an entire network of Usenet hosts.

4.2 Moderated Newsgroups

Some news groups are moderated, and for an article to be posted on these groups, a person (moderator) has to "approve" [3] the article. Some news hub sites maintain an up-to-date list of all the moderators and their email addresses. Other sites simply mail the article to a designated email address at these hub sites which then forwards the article to the real moderator's mailbox.

To create a locally moderated group, all one has to do is add the **m** flag in the `active` file, and then put the name of the moderator in the `mailpaths` file. An entry for the non-local groups should exist as:

```
backbone %s@rutgers.edu
```

The `rutgers.edu` can be replaced by anyone of the twenty "hub" sites that maintain an up-to-date list of all moderators (see Appendix).

To approve an article, simply edit the article and add the APPROVED: line in the headers. You might want to use the inews -h option to repost the approved article.

4.3 Monitoring & Status

To see batches arriving, use `syslog`. Information about all articles that have been received recently is in the history file. To list all articles that have arrived around a certain date, use histfrom, newsfrom. To see the age of an article, run ctime (converts number of seconds to date and time) on the number of seconds from the `history` file.

4.4 expire

Articles can contain a EXPIRES: header field, which might be larger than the default expiry period in your `explist` file. To unconditionally clean out such articles, set the third field in your *bounds* line of the `explist` file to a low value [4]. This technique can be used to clean up and recover some disk space in case of rearrangement of disks, etc.

Sometimes doexpire complains about not having sufficient space to rebuild the `history` files. This is tricky, since the `history` file is not trimmed and its size goes on increasing (doexpire will never get enough space to rebuild the file). One drastic way around this is to bring the system to singleuser mode and then delete the history database files and rebuild using mkhistory. Another way is to shorten the number of history days in the `explist` file and then call expire directly (do this in single user mode else the locking screws up and expire just freezes if another NNTP batch comes in

[3] inews does not do a stringent test on the `approved` line so potentially it can be fooled

[4] Just make sure that this value is not smaller than the highest of the 'expire' dates for the various groups, else you will get a *preposterous date* error.

or the like). Or edit the doexpire script and remove the "spacefor" check– this has worked fine for the author since the extra disk space that Unix keeps aside was all that was needed and spacefor didn't account for that.

Expire should be run after midnight to conserve disk space. Since it only looks at the day the article was received (and not the hour), if invoked *before* midnight, it does not expire the articles which would have been expired if it had been called after midnight (since that would have been the beginning of the *next* day). Effectively, the articles are staying around for one day more than you want them to.

Also, try and time expire and an incoming news batch sufficiently far apart so that expire has adequate disk space to rebuild the `history` files before the next large news batch comes in. In case the news articles are stored on a separate disk partition, running newshourly so that the incoming batch is split into articles before expire starts up will help save disk space.

4.5 history

The history files can be rather large, and news admins could run out of temporary disk space while trying to rebuild the history files (see section 4.4). Try reducing the *expired* field to a small value in your `explist` file so that it doesn't keep information about old news articles for a long period of time.

Use addmissing to add missing news articles to history file.

4.6 sys

The `sys` file contains the list of a sites' subscriptions and distributions. It is important to control the articles coming into a host, so the use of `all` is discouraged. The list of top-level newsgroups and distributions is not very large and can easily be specified in the `sys` file.

A judicious entry of the system name in the `sys` file can save a lot of re-transmission of articles. relaynews tries to match the system name in the `sys` file with the list of hosts in the *Path:* header of an article. It does not send an article to a system that exists in the *Path:* header. A sample line from the `sys` file is:

```
princeton:all,!jvncnet,to.idunno/all:F:idunno.princeton.edu/togo.nntp
```

and a sample *Path:* line from an article is:

```
jvnc.net!princeton!att!linac!pacific.mps.ohio-state.edu!
```

The use of princeton instead of the real hostname (idunno.princeton.edu) is deliberate in the `sys` file– they use the name princeton for news purposes (their `whoami` file) and that name reflects in the

Path: header. By using their *news*-name as the system entry in the `sys` file, relaynews does not send the the article to them again. Observe that even if we had used another name, idunno.princeton.edu would have rejected the article by tracking the message id in its `history` file. However, judicious use of the sitename in the `sys` file can prevent extensive requeueing of articles and searches in the `history` database. One can also specify a list of sites to search for in the PATH: header after the site name in the `sys` file.

4.7 rn

rn maintains three files in a users home directory:

1. .newsrc List of all subscribed-to newsgroups.
2. .rnlast Stores three items for next time rn starts up- last new group, last time these values were recorded, last `active` size. It uses these values to check for new newsgroups, etc. The size is used to see the newly added newsgroups, assuming that the `active` file was not edited by hand and new newsgroups were added at the end of the `active` file. It uses `.rnsoft` to see if the active file was "munged" and if it should trust the values in `.rnlast`.
3. .rnsoft Contains file offset (line offset) pointers into the `active` file for faster speed or to see whether edited manually (CHECK).

Edit newsetup shell script to subscribe to only a few newsgroups for new users (rn calls this program for new users).

4.8 Miscellaneous Tips

This section lists useful tips not covered elsewhere in the document.

1. *Transmitting all available articles to a new downstream site*

   Run `cd /usr/spool/news ; find . -print > togo.nntp` to create the `togo` file for a site. This simply writes out every article on disk into a 'togo' file. Copy this file over to a site's `out.going` directory and invoke nntpsend.

2. *Locking news manually and then unlocking it.*

   locknews locks all news processes from running. It actually puts you into a sub-shell, so use `exit` to exit from the sub-shell and unlock news.

3. *Updating* `active` *file from the "junk" newsgroup.*

   Periodically, look at the articles under the *junk* newsgroup and add desirable newsgroups to the `active` file by running addgroup.

References

[RFC1036] Horton M., Adams R. *Standard for the Interchange of USENET messages (RFC 1036)*

[RFC977] Kantor B., Lapsley P., *Network News Transfer Protocol (RFC 977)*

[CNews] Spencer H., Collyer G. *C News Installation & misc articles*

[BNews] Glickman M., *USENET Version B Installation*

[Man] *Manual pages from C News, B News, NNTP, rn, nn*

Appendix

A Programs in News

inews It takes a *single* article, ensures that *Message-ID, From, Path* are proper and then injects the article into the newsflow (either directly calls relaynews or else mails it to the moderator if the newsgroup is moderated).

rnews just copies the incoming news batch to the `in.coming` spool directory after stripping off the garbage mail headers, etc. It simply names the incoming batch based on the time and type of batch (compressed, c7encoded or plain text). This is a simple shell script which actually calls on newsspool to do all the work.

newsrun Simple shell script that uncompresses and un-decodes a batch and feeds to relaynews.

relaynews the heart of C-news. Looks at the `sys` file and installs articles locally (updating `history & active` files in the process) and also adds the articles to the list of articles to be forwarded to other sites (in file `out.going/togo.`).

B Mailpaths

This lists all the moderators for the various newsgroups. One can get the `List_of_Moderators` file from the location listed below and edit it to include the *backbone* entry (and trim the header stuff).

The message below is a posting on the `news.admin` newsgroup and also available online via *anonymous* ftp from pit-manager.mit.edu in:

`pub/usenet/news.lists/How_to_Construct_the_Mailpaths_File`

```
Xref: jvnc.net news.lists:431 news.admin:7794
Path: jvnc.net!...!zaphod.mps.ohio-state.edu!purdue!spaf
From: spaf@cs.purdue.EDU (Gene Spafford)
Newsgroups: news.lists,news.admin
Subject: How to Construct the Mailpaths File
Message-ID: <15989@ector.cs.purdue.edu>
Date: 9 Sep 91 00:15:07 GMT
Expires: 8 Dec 91 00:15:07 GMT
Followup-To: news.lists
Organization: Dept. of Computer Sciences, Purdue Univ.
Lines: 118
Approved: spaf@cs.purdue.EDU
Supersedes: <15404@ector.cs.purdue.edu>
```

```
Original-from: Gene Spafford (spaf@cs.purdue.edu)
[Most recent change: 7  Sep 1991 by spaf@cs.purdue.edu (Gene Spafford)]
```

News 2.11 has been designed to make it simpler to submit articles to moderated newsgroups and to reply via mail to posted articles. For these functions to work, the file "mailpaths" must exist in the news library and contain current information. This file describes the syntax of the contents of the file and how to construct it for your site.

Syntax

All lines in the file consist of a keyword followed by whitespace, followed by a printf–style format string which is used to encode a mail address. The format string should be constructed so that there is a single "%s" in the field, and other information necessary to construct a valid address to the appropriate site (see below).

The keyword field consists of either the word "internet", the word "backbone", or some newsgroup pattern. These are examined by the software to determine which format-string to use. Note that the first appropriate line matched is the one used – thus, the "backbone" keyword line should be placed last in the line if regional or local moderated groups are supported.

...

Submissions to moderated groups

When you attempt to post to a moderated newsgroup (indicated by the letter "m" in the 4th field of the "active" file for that group), the action of "inews" is to mail the submission to the moderator. This is done by searching through "mailpaths" file for a keyword matching the newgroup being posted to, or, by default, the keyword "backbone". Matching occurs as in the "sys" file – thus, "world" and "all" will also match everything, while "news" or "news.all" will only match articles posted in the "news" category. This feature can be used to support local moderated groups.

Once a line has been matched, the name of the moderated group is transformed into an address. This is done by first turning all the imbedded periods within the newsgroup name into hyphens (e.g., "news.lists" becomes "news-lists") because many mailers mishandle addresses with periods in the username (periods are supposed to be delimiters only in the host/domain part of RFC822 addresses, but some mailers "overreact"). Next, the transformed group name is encoded into an address using the format-string present on the line matched from "mailpaths" and the article is mailed. Thus, a posting to "news.lists" from a site with the line

```
backbone        emory!gatech!%s
```

in the mailpaths file would have the article mailed to "emory!gatech!news-lists". Likewise, a line like

```
backbone %s@gatech.edu
```

would result in the article being mailed to "news-listsgatech.edu". To make the mechanism for moderated postings simpler, a subset of the best-connected sites (plus some others) have committed to keeping a complete up-to-date set of mail aliases for the moderated groups. Therefore, to build this line in your "mailpaths" file, you need only construct an address to get the mail to one of these sites. In turn, once the submission reaches that site, it will be forwarded on to the appropriate moderator's mailbox.

The sites currently maintaining these lists are given in the following list. Pick the one that is "closest" to your site and use it in your "backbone" format string:

```
ames.arc.nasa.gov       beaver.cs.washington.edu  cis.ohio-state.edu
decuac.dec.com          decwrl.dec.com            eddie.mit.edu
kddlab.kddlabs.co.jp    linus.b.mitre.org         math.waterloo.edu
mcnc.org                ncar.ucar.edu             rutgers.edu
tektronix.tek.com       ucbvax.berkeley.edu       ucsd.edu
uflorida.cis.ufl.edu    uunet.uu.net              wrdis01.af.mil
```

Internet Mail

If you define the "INTERNET" flag when you build 2.11 news, the software will use the internet-style "From:" header when addressing replies mailed to postings. If your mailer does not have this capability, you can still (possibly) achieve the same thing by defining the "internet" line in your "mailpaths" file to forward such mail to a host that does have a mailer which understands internet-style addresses.

The format of this line is very similar to the "backbone" line. As an example, consider:

```
internet emory!gatech!mcnc!%s
```

Thus, if "INTERNET" was defined when this version of news was built, any reply to a news article would not travel along the "Path:", but would instead be sent to "mcnc" for interpretation and remailing. For example, to reply to this article, the mail would be sent to:

```
emory!gatech!mcnc!spaf@cs.purdue.edu  -> emory!gatech!mcnc!cs.purdue.edu!spaf
```

...

The following sites will accept internet-format mail for forwarding:

```
beaver.cs.washington.edu    decuac.dec.com  kddlab.kddlabs.co.jp
math.waterloo.edu           mcnc.org        ucsd.edu
uunet.uu.net
```

Gene Spafford
NSF/Purdue/U of Florida Software Engineering Research Center,
Dept. of Computer Sciences, Purdue University, W. Lafayette IN 47907-1398
Internet: spaf@cs.purdue.edu phone: (317) 494-7825

C News Flow analysis

(Taken from Brian Reid's (reiddecwrl.dec.com) article on *news.lists*)

The flow data collected at decwrl shows the path that each article took to reach its destination. By combining this data and merging by host, we can compute the "influence" of each site. The *influence* of a site is a measure of how much news it carries for other places. Influence is measured in percent. An influence rating of, say, 21%, means that 21% of all stored news articles, worldwide, have passed through that site on the way to their final destination.

Here are the top X sites on USENET for AUG 91, ranked by influence, in percent:

Rank	weight	name
1	40.87	uunet
2	32.03	zaphod.mps.ohio-state.edu
3	17.08	mcsun
4	13.12	samsung
5	11.44	spool.mu.edu
6	11.10	cs.utexas.edu
7	11.01	usc
8	10.99	sdd.hp.com
9	9.57	elroy.jpl.nasa.gov
10	8.96	rpi
11	8.22	unido
12	8.17	mips
13	7.82	sol.ctr.columbia.edu
14	7.76	wupost
15	7.75	caen
16	7.56	think.com
17	7.32	cis.ohio-state.edu
18	6.99	att
19	6.54	swrinde
20	6.53	decwrl
21	6.08	ira.uka.de
22	5.92	linac
23	5.87	ukc
24	5.38	news-server.csri.toronto.edu

25	5.21	math.fu-berlin.de
28	4.65	munnari.oz.au
29	4.62	pacific.mps.ohio-state.edu
30	4.25	lll-winken
31	3.92	utgpu
32	3.91	uwm.edu
33	3.66	olivea
34	3.58	qt.cs.utexas.edu
35	3.52	pacbell.com
36	3.47	agate
37	3.31	fub
38	3.25	apple
39	3.24	wuarchive
40	3.21	yale.edu

D sendbatches diffs

```
r2d2% diff nntpsend.csh nntpsend.jvnc

5c5,7
< set batchdir=/usr/spool/news/batch libdir=/usr/spool/news/lib
---
> set batchdir=/usr/spool/news/out.going
> set batchfile=togo.nntp
> set libdir=/usr/spool/news/lib # for nntpxmit, shlock
8a11
> set spooldir=/usr/spool/news # cd if not absolute pathnames
25,26c28,34
< foreach host ( {cad,zen,jade,cartan}.berkeley.edu decvax.dec.com cgl.ucsf.edu ucdavis.edu 128
<  set lock=NNTP_LOCK.${host} tmp=${host}.tmp send=${host}.nntp
---
> foreach host ( $* ) # all hosts on the command line
>  if (! -d ${host} ) then
>  echo ${pname}: "[$$]" ${host} is not a directory, skipping
>  continue
>  endif
>  set lock=${host}/NNTP_LOCK
>  set tmp=${host}/togo.tmp send=${host}/togo.nntpxmit batch=${host}/${batchfile}
33c41
<  if ( -e ${host} ) then
---
>  if ( -e ${batch} ) then
38c46
<  mv ${host} ${tmp}
---
>  mv ${batch} ${tmp}
40c48
<  mv ${host} ${send}
---
>  mv ${batch} ${send}
45c53
<  time nntpxmit ${host}:${send}
---
>  (cd ${spooldir} ; time nntpxmit ${host}:${batchdir}/${send} )
```

E New site questionarre

The following lists the necessary information needed to start a new NNTP newsfeed.

1. System details
   (a) Operating system and version
   (b) Version of news (B/C)
   (c) Internet name of host to which our system will be connecting to.
   (d) Hostname that needs NNTP access– this host *must* have a reverse domain entry for the NNTP security check.
   (e) Name of your site for news purposes (`newshostname` or the contents of your `lib/whoami` file).
2. Administrative details
   (a) Contact name for News administration
   (b) Telephone number, email address
3. Newsfeed details
   (a) Newsgroups desired
   (b) Subscriptions desired

At the time of writing this document, the following newsgroups and subscriptions are carried by JvNCnet:

```
alt,bionet,comp,control,ddn,dimacs,general,gnu,ieee,mail,\
misc,news,rec,sci,soc,talk,trial,u3b,\  # standard newsgroups
inet,us,usa,na,world,\                  # top level distributions
bit,biz,clari,k12,pubnet,vmsnet,\           # uncommon ones
can,nj,ny,pa,phl,\                      # distribution states
cit,jvncnet,princeton,to                # local
```