USENET Version B Installation

Matt Glickman
Computer Science Division
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, California 94720

Revised by Mark Horton for version 2.10
Revised by Rick Adams for version 2.10.3

1. Introduction

This document is intended to help a USENET site install and maintain the network news software. Please ask questions of Rick Adams†; such questions will help to point out areas that need to be addressed here.

The overall order of things to do is:

(a) Find somebody to link up with. You need a network connection of some kind, for example, ARPANET or UUCP. If you must use UUCP and have no connections, you must have at least a dialup and preferably a dialer, and find someone willing to call your machine. The USENET directory may be helpful in finding some other site geographically near yours to hook up to.

(b) Create a *localize.sh* script to make local changes to the makefile and *defs.h* files. (Section 2 gives more details about creating *localize.sh*.) Once you're finished editing *localize.sh*, create a *defs.h* and *Makefile* tailored for your site with the command

```
sh localize.sh
```

Inspect *defs.h* and *Makefile* to ensure that all your local customizations got into your final versions. If you saw a ‘‘?’’ when you ran *localize.sh*, one or both of the files is certainly wrong. It's a good idea to anchor the patterns in *localize.sh*'s *ed*(1) scripts, especially in its *Makefile*-editing lines. For instance, use **/^UUXFLAGS/** instead of **/UUXFLAGS/**.

(c) Compile the software using the *make*(1) command.

(d) *Su*(1) and type ‘‘make install’’. This will copy the files out to the right place and make directories containing most of the important files. It will configure you in with a connection to *oopsvax* via UUCP links. This is undoubtedly wrong, so you will have to configure links as needed. If you are upgrading from a version older than 2.10.3, do ‘‘make update’’. This will cause various checks to be performed on important files in **LIBDIR**. The results will be reported to you. If you are not sure if you should do ‘‘make update’’, do it. It will not hurt anything if you have already done it.

(e) After editing the configuration table, get your contact at the other end of the link to add you to their netnews *sys* file.

(f) Post a message to the **to.*sysname*** newsgroup which should be set up to go only to the site you are linked to, as a test. Have the other person send a message to your system using the same mechanism. If this doesn't work, find the problem and fix it. (Please don't use **net.test** unless there is no alternative. It is almost always possible to use **test**, or **to.*sysname*** or some ***local*.test** group, instead of **net.test**.)

---

† ARPANET: rick@seismo.CSS.GOV, UUCP: seismo!rick

(g) Fill out a USENET directory form (the file *dirform* in the *misc* directory). Post a copy to the USENET newsgroup **net.news.newsite** and mail a copy to *cbosgd!uucpmap*.

(h) Format the document *‘‘How to Read the Network News’’* (the file *howto.mn* in the *doc* directory), the document *‘‘How to Use USENET Effectively’’* (the file *manner.mn* in the *doc* directory) and the document *‘‘Copyright Law’’* (the file *copyright.mn* in the *doc* directory) and post them to your **general** newsgroup with a long expiration date. You can use *inews*(1) or *postnews*(1) to do this.

(i) It will probably be necessary to fix your uucp commands to allow *rnews* and to support the **–z** and **–n** options (if you are lucky enought to have the source).

2. Installation

2.1. Configuration

Local configuration of the USENET version B software requires you to edit a few files. Most importantly, the *defs.h* and *Makefile* files must be created from their templates *defs.dist* and *Makefile.dst*. You should create a shell script called *localize.sh* which copies the files and makes local changes to the copies. Even for a completely vanilla site, some changes will be necessary. For example, your script should start with *localize.v7* or *localize.usg*. You should include the name of the local organization (**MYORG**) and the uid of the local news super user (**ROOTID**). You should also choose how your hostname will be determined. If you are a USG site, define **UNAME** in *defs.h*. If you are running 4.[23] BSD, define **GHNAME** in *defs.h*. If you have your UUCP name in */etc/uucpname*, define **UUNAME** in *defs.h*. Otherwise, news will look in the file */usr/include/whoami.h* for a line of the form

```
#define sysname your-sysname
```

If you are running System 3 or System 5, you are a USG site. Otherwise, unless you are in AT&T, you are probably a V7 site. The previously mentioned defines are the only modifications that are *necessary* to install news at your site. However, you will probably want to change some of the ones listed below. If your compiler does not accept ‘‘(void)’’, the simplest thing to do is add ‘‘–Dvoid=int’’ to the **CFLAGS** line in the *Makefile*.

A sample localize shell script can be found in *localize.sample*. The most important parameters are:

2.1.1. ROOTID

The numerical uid of the person who is the news super user. This should not be set to 0. Normally it is set to the uid of the news contact person for the site. If it is not defined, the uid of **NOTIFY** will be looked up in */etc/passwd* and used instead.

2.1.2. N_UMASK

Mask for *umask*(2) system call. Set it to something like 022 for a secure system. Unsecure systems might want 002 or 000. This mask controls the mode of news files created by the software. Insecure modes would allow people to edit the files directly.

2.1.3. DFLTEXP

The default number of seconds after which an article will expire. Two weeks (1,209,600 seconds) is the default choice. If you wish to expire articles faster than two weeks, it is recommended that you use the **–e** flag to expire instead of decreasing **DFLTEXP**.

2.1.4. HISTEXP

Articles which were posted more than **HISTEXP** ago are considered too old and are moved into the junk directory. This is because they are too old to be in the history file, so it is impossible to tell if they really should be accepted or are endlessly looping around the network. (This was theoretically possible before this feature was added.) The articles are removed after **DFLTEXP** seconds, but a copy of their ‘‘Message-ID’’ is kept in the history file for **HISTEXP** seconds (the default is 4 weeks).

2.1.5. DFLTSUB

The default subscription list. If a user does not specify any list of newsgroups, this will be used. Popular choices are **all** and **general**,**all.general**.

2.1.6. TMAIL

This is the version of the Berkeley *Mail*(1) program that has the −**T** option. If left undefined, the −**M** option to *readnews*(1) will be disabled.

2.1.7. ADMSUB

This newsgroup (or newsgroup list) will always be selected unless the user specifies a newsgroup list that doesn't include **ADMSUB** on the command line. That is, as long as the user doesn't use the −**n** flag to *readnews* on the command line, **ADMSUB** will always be selected. This is usually set to **general**. (The intent of this parameter is to have certain newsgroups which users are required to subscribe to. A typical site might require **general**.)

2.1.8. PAGE

The default program to which articles should be piped for paging. This can be disabled or changed by the environment variable **PAGER**. If you have it, the Berkeley *more*(1) command should be used, since the **+** option allows the headers to be skipped.

2.1.9. NOTIFY

If defined, this character string will be used as a user name to send mail to in the event of certain control messages of interest. (Currently these are **newgroup**, **rmgroup**, **sendsys**, **checkgroups**, and **senduuname**.) As distributed, mail will be sent to user *usenet*. It is recommended you create such a mailbox (have it forwarded to yourself) if possible, since this makes it easier for another site to contact the site administrator for your site. If you are unable to do this (*e.g*., you are not the super user) you should change this name to yourself. Also, messages about missing or extra newsgroups are mailed to this user by the **checkgroups** control message.

2.1.10. DFTXMIT

This is the default command to use to transmit news if no explicit command is given in the fourth field of the *sys* file. It normally includes *uux*(1) with the −**z** option. You should install this modification to UUCP at once; otherwise your users will start being bombarded with annoying *uux* completion messages. However, you can turn this off to get news installed.

2.1.11. UXMIT

This is the default command used if the **U** flag is present in the flags portion of a *sys* file line. In this case, the second ‘‘%s’’ refers to the name of a file in the news spool area, not a temporary file. It can usually only be used when local modifications are made to the uucp system, such as the −**c** option to *uux*.

2.1.12. DFTEDITOR

This is the full path name of the default editor to use during followups and replies. It should be set to the most popular text editor on your system. As distributed, *vi*(1) is used.

2.1.13. UUPROG

If this is defined, it will be used as a command to run when the **senduuname** control message is sent around. Otherwise the command *uuname*(1) will be run. Normally, this program should be placed in **LIBDIR**.

2.1.14. MANUALLY

If this is defined, incoming **rmgroup** messages will not automatically remove the group. News will instead mail a message to **NOTIFY** advising that the group should be removed. If you define **MANUALLY**, you should have **NOTIFY** defined. **MANUALLY** is defined by default to protect you against accidental or malicious removal of an important newsgroup.

2.1.15. NONEWGROUPS

If this is defined, incoming **newgroup** messages will not automatically create the group. News will instead mail a message to **NOTIFY** advising that the group should be created. If you define **NONEWGROUPS**, you should have **NOTIFY** defined. **NONEWGROUPS** is undefined by default to make it easier to automatically maintain the news system.

2.1.16. BATCH

If set, this is the name of a program that will be used to unpack batched articles (those beginning with the character ‘‘#’’.) Batched articles normally are files reading

```
#! rnews 1234
article containing 1234 characters
#! rnews 4321
article containing 4321 characters
. . .
```

Batching is *strongly* recommended for increased efficiency on both sides.

2.1.17. LOCALNAME

Most systems have a full name database on line somewhere, showing for each user what their full name is. Most often this is in the gecos field of */etc/passwd*. If your system has such a database, **LOCALNAME** should be left undefined. If not, define **LOCALNAME**, and articles posted will only receive full names from local user information specified in *NAME* or ***$HOME**/.name* by the user. If you have a nonstandard gcos format (not *finger*(1) or RJE) it will be necessary to make local changes to *fullname.c* as appropriate on your system.

2.1.18. INTERNET

If your system has a mailer that understands ARPA Internet syntax addresses (‘‘user@site.domain’’) turn this on, and replies will use the ‘‘From’’ or ‘‘Reply-To’’ headers. Otherwise, leave it disabled and replies will use the ‘‘Path’’ header.

2.1.19. MYDOMAIN

When generating internet addresses, this domain will be appended to the local site name to form mailing address domains. For example, on system *ucbvax* with user *root*, if **MYDOMAIN** is set to ‘‘.UUCP’’, addresses generated will read ‘‘root@ucbvax.UUCP’’. If **MYDOMAIN** is ‘‘.Berkeley.EDU’’, the address would be ‘‘root@ucbvax.Berkeley.EDU’’. If your site is in more than one domain, use your primary domain. The domain always begins with a period, unless the local site name contains the domain; in this case **MYDOMAIN** should be the null string.

2.1.20. CHEAP

Do not *chown*(1) spool files to *news*. This will cause the owner of the file to be the person that started the *inews* process. This is used for obscure accounting reasons on some systems.

2.1.21. OLD

Define this if any of your USENET neighbors run 2.9 or earlier versions of B news. It will cause all headers written to contain two extra lines, ‘‘Article-I.D.’’ and ‘‘Posted’’, for downward compatibility. Once all your neighbors have converted, you can save disk space and transmission costs by turning this off. It is strongly encouraged that they convert. 2.10.3 is *much* faster than 2.9. The performance difference is dramatic.

2.1.22. UNAME

Define this if the *uname*(2) system call is available locally, even though you are not a USG system. USG systems always have *uname*(2) available and ignore this setting.

2.1.23. GHNAME

Define this if the 4.[23] BSD *gethostname*(2) system call is available. If neither **UNAME** or **GHNAME** is defined, *inews* will determine the name of the local system by reading */usr/include/whoami.h*.

2.1.24. UUNAME

Define this if you keep your UUCP name in */etc/uucpname*.

2.1.25. V7MAIL

Define this if your system uses V7 mail conventions. The V7 mail convention is that a mailbox contains several messages concatenated, each message beginning with a line reading ‘‘From *user date*’’ and ending in a blank line. If this is defined, articles saved will have these lines added so that mail can be used to look at saved news.

2.1.26. SORTACTIVE

Define this if you want the news groups presented in the order of each person’s *.newsrc*(5) instead of the active file.

2.1.27. ZAPNOTES

Define this if you want old style notesfile id’s in the body of the article to be converted into ‘‘Nf-Id’’ fields in the header.

2.1.28. DIGPAGE

If this is defined, *vnews*(1) will attempt to process the subarticles of a digest instead of treating the article as one big file.

2.1.29. DOXREFS

Define this if you are using *rn*(1). *Rn* uses this option to keep from showing the same article twice.

2.1.30. MULTICAST

If your transport mechanism supports multi-casting of messages, define this. Currently ACSNET is the only network that can handle this.

2.1.31. BSD4_2

Define this if you are running 4.2 or 4.3 BSD UNIX†.

2.1.32. BSD4_1C

Define this if you are running 4.1C BSD UNIX.

2.1.33. SENDMAIL

Use this program instead of *recmail*(8) for sending mail.

2.1.34. MMDF

Use MMDF instead of *recmail* for sending mail.

---

†UNIX is a trademark of AT&T Bell Laboratories.

2.1.35. MYORG

This should be set to the name of your organization. Please keep the name short, because it will be printed, along with the electronic address and full name of the author of each message. Forty characters is probably a good upper bound on the length. If the city and state or country of your organization are not obvious, please try to include them. If the organization name begins with a ‘‘/’’, it will be taken as the name of a file. The first line in that file will be used as the organization. This permits the same binary to be used on many different machines. A good file name would be */usr/lib/news/organization.* For example, an organization might read ‘‘AT&T Bell Labs, Murray Hill’’, ‘‘U.C. Berkeley’’, ‘‘MIT’’, or ‘‘Computer Corp. of America, Cambridge, Mass’’.

2.1.36. HIDDENNET

If you want all your news to look like it came from a single machine instead of from every machine on your local network, define **HIDDENNET** to be the name of the machine you wish to pretend to be. Make sure that you have you own machine defined as *ME* in the sysfile or you may get some unnecessary article retransmission.

2.1.37. NICENESS

If **NICENESS** is defined, *rnews* does a *nice*(2) to priority **NICENESS** before processing news.

2.1.38. FASCIST

If this is defined, *inews* checks to see if the posting user is allowed to post to the given newsgroup. If the username is not in the file **LIBDIR**/*authorized* then the default newsgroup pattern in the symbol **FASCIST** is used.

The format of the file *authorized* is:
user:allowed groups

For example:

```
root:net.all,mod.all
naughty_person:junk,net.politics
operator:!net.all,general,test,mod.unix
```

An open environment could have **FASCIST** set to **all** and then individual entries could be made in the authorized file to prevent certain individuals from posting to such a wide area.

Note that a distribution of **all** does *not* mean to allow postings only to local groups − **all** includes **all.all**. Use **all**,!**all.all** to get that behavior

2.1.39. SMALL_ADDRESS_SPACE

Define this if your machine has 16 bit (or smaller) pointers. If you are on a PDP-11†, this is automatically defined.

2.2. Makefile

There are also a few parameters in the *Makefile* as well. These are:

2.2.1. OSTYPE

This is the type of UNIX system you are using. It should be either **v7** or **USG**. Any BSD system is v7. Any System 3 or System 5 system is USG. This is normally set by *localize.sh*.

2.2.2. NEWSUSR

This is the owner (user name) of *inews*. If you are a superuser, you should probably create a new user id (traditionally *news*) and use this id. If you are not a superuser, you can use your own user id. If you are able to, you should create a mail alias *usenet* and have mail to this alias forwarded to you. This will make it easier for other sites to find the right person in the presence of changing jobs and out of date or nonexistent directory pages. **NEWSUSR** and **ROOTID** do not need to represent the same user.

---

†PDP-11 is a trademark of Digital Equipment Corporation.

2.2.3. NEWSGRP

This is the group (name) to which *inews* belongs. The same considerations as **NEWSUSR** apply.

2.2.4. SPOOLDIR

This directory contains subdirectories in which news articles will be stored. It is normally */usr/spool/news.*

Briefly, for each newsgroup (say **net.general**) there will be a subdirectory */usr/spool/news/net/general* containing articles, whose file names are sequential numbers, *e.g*., */usr/spool/news/net/general/1*, etc.

Each article file is in a mail-compatible format. It begins with a number of header lines, followed by a blank line, followed by the body of the article. The format has deliberately been chosen to be compatible with the ARPANET standard for mail documented in RFC 822.

You should place news in an area of the disk with enough free space to hold the news you intend to keep on line. The total volume of news in **net.all** currently runs about 1 Mbyte per day. If you expire news after the default 2 weeks, you will need about 14 Mbytes of disk space (plus some extra as a safety margin and to allow for increased traffic in the future.) If you only receive some of the newsgroups, or expire news after a different interval, these figures can be adjusted accordingly.

2.2.5. BATCHDIR

This directory will contain the list of articles to send to each system. It is normally */usr/spool/batch.*

2.2.6. LIBDIR

This directory will contain various system files. It is normally */usr/lib/news.*

2.2.7. BINDIR

This is the directory in which *readnews*, *postnews*, *vnews*, and *checknews*(1) are to be installed. This is normally */usr/bin*. If you decide to set **BINDIR** to a local binary directory, you should consider that the *rnews* and *cunbatch* commands must be in a directory that can be found by *uuxqt*, which normally only searches */bin* and */usr/bin*.

2.2.8. UUXFLAGS

These are the flags *uux* will be called with.

2.2.9. LNRNEWS

This is the program used to link *rnews* and *inews*. If you have symbolic links, you can replace the ‘‘ln’’ with ‘‘ln –s’’.

2.2.10. SCCSID

If this is defined, sccs ids will be included in each file. If you are short on address space, don’t define this.

3. FILES

This section lists the files in **LIBDIR** and comments briefly what they do.

3.1. active

A list of active newsgroups. It is automatically updated as new newsgroups come in. The order here is the order news is initially presented by *readnews*, so you can edit this file to put important newsgroups first. If you have **SORTACTIVE** defined, after the first time the user invokes *readnews*, it will be presented in the order of his *.newsrc*. Each line of the active file contains four fields, separated by a space: the newsgroup name, the highest local article number (for the most recently received article), the lowest local article number that has not yet expired, and a single character used to determine if the user can post to that newsgroup. If the character is ‘‘y’’ the user is permitted to post articles to that group. If the character is ‘‘n’’ the user is not permitted to post articles to that groups. (This field takes the place of the *ngfile* in earlier versions of news. Local article numbers begin at 1 and count sequentially within the newsgroup as articles are received. They do not usually correspond to local article numbers on other sites. The article numbers are always stored as a five digit number (with leading zeros) to allow updating of the file in place.

The active file should contain **all** active net-wide active newsgroups (**net.all**and **mod.all**). It is important that they all be present, as they are used as a check for valid newsgroup names and invalid newsgroup names are removed from any articles processed by *inews*. You should use the *sys* file to keep out unwanted newsgroups.

3.2. aliases

This file is used to map bad newsgroup names to the correct ones. (For example, **net.unix.wizards** is mapped into **net.unix-wizards**). Each line consists of two fields separated by a space. If the first field is found in the newsgroup list of the incoming article, it is changed to the second field. This change takes place in the article before it is passed on to other systems, not just locally.

3.3. batch

This program reads a list of filenames of articles and outputs the articles themselves. It is typically used by the shell script *sendbatch*.

3.4. c7unbatch

This is used to decompress news that has been **encoded** for transmission over a network that only supports 7-bit transfers (e.g X.25.)

3.5. caesar

This is a program to do Caesar decoding of rotated text, on a line by line basis. The standard input is copied to the standard output, rotating each line according to a static single letter frequency table. If an integer argument is given (*e.g*., 13), every line is rotated by that argument, without regard to letter frequencies. This program is invoked by the **D** *readnews* command. It is also used by *postnews* with the ‘‘13’’ argument to encode selected material for posting.

3.6. checkgroups

Checkgroups is a shell file to aid in automatically checking the accuracy of your active file. It is executed by the **checkgroups** control message and mails a list of out of date newsgroups to the person defined by **NOTIFY** It also updates the *newsgroups* file that is used by *postnews* as a helpfile for newsgroup selection.

3.7. compress

This program does a modified Lempel-Ziv data compression. It is used by the compressed batching scheme. It averages 50% compression on a typical batch of news.

3.8. distributions

This is a list of distributions that are valid for your site. Each line has two fields separated by the first space on the line. The first field is the name of the distribution (*e.g*., **usa**, **na**, etc.). The second field is text describing the distribution. As distributed, this file is only correct for sites in the USA. You should examine this file and add or delete the appropriate distributions.

3.9. encode

This program transforms an 8-bit binary file into a file suitable for sending over a link that only allows 7-bit characters. It is used by **sendbatch -c7.**

3.10. errlog

This file contains the ‘‘important’’ error messages found in the log file. These errors usually indicate that something was wrong with an article. This file should be watched closely. The *log* file contains much more verbose information and it is often difficult to detect errors in it.

3.11. expire

This program expires old articles and archives them if archiving is selected. It is typically run once a day from *cron*(8).

3.12. help

This contains a list of commands printed when an illegal command is typed to *readnews*.

3.13. history

A list of every article that has come in to your system. It is used to reject articles that come in for the second time (presumably via a different path). This file will grow but is cleaned out by the *expire*(8) command.

3.14. history.d

On USG systems, this directory contains 10 files (history.[0-9]) which are used as part of a simple hashing algorithm to speed up history searches. Since V7 systems have DBM, this is not used on V7 systems.

3.15. history.dir,history.pag

These two files are used on V7 systems as a hashed version of *history*, containing the message id's of all articles in history. They are only used if **–DDBM** and **–ldbm** appear in *Makefile*.

3.16. inews

This is the program that actually sends and receives news. All other programs interface eventually with it. It is not intended to be used directly by a human, so it is no longer in */usr/bin*.

3.17. log

If present, a log of articles processed and error conditions is kept here. This file grows without limit unless cleaned out periodically. The *trimlib* script in *misc* can be invoked from *cron* daily or weekly to keep the log short.

3.18. moderators

This file contains a list of the moderators and their mailing addresses for each moderated newsgroup. Each line consists of two fields. the first is the name of the moderated group. The second is the mailing address of the group's moderator. As distributed, they are almost certainly wrong. You will need to modify the paths so they work from your site.

3.19. newsgroups

This file is displayed by *postnews* when a user hits **?** in response to its request for newsgroups. It is also used by *vnews* when it displays the newsgroup name. It is updated automatically by the **checkgroups** control message.

3.20. notify

If this file is present, its contents will be taken as the name of the user to notify in case of a problem. If the file is empty, nobody will be notified. (This overrides the **NOTIFY** option in *defs.h*). Having a null file is useful if one person administers several systems and does not want multiple copies of control message notifications.

3.21. oactive, ohistory, ohistory.dir, ohistory.pag

These are copies of the corresponding *active*, *history*, *history.dir*, and *history.pag* files before *expire* ran. They are kept in case something happens to the originals.

3.22. recmail

This program can serve as a link between news and your local mailer. If you have *sendmail*(8), don't use *recmail*. *Sendmail* is much more useful.

3.23. recnews

A program which allows you to send mail to get news posted. You usually need to run *sendmail* or *delivermail*(8) to be able to use this.

3.24. recording

A list of newsgroup classes and filenames to display recordings for. The recording feature is analogous to the recordings played in some areas when you dial directory assistance, trying to be annoying and make you think twice. Recordings on certain newsgroups are intended to remind the user of the rules for the newsgroup, or, in the case of a company worried about letting proprietary information out, reminding authors that anything they say is seen outside the company and so proprietary information should not be included.

The file contains one line per recording. The line contains two fields, separated by a space. The first field is the newsgroup class (*e.g.*, **net.all**), the second field is the name of the file containing the recorded message. If the file name does not begin with a slash, it will be searched for in **LIBDIR**. Sample recording files can be found in the *misc* directory.

3.25. rmgroup

This shell file should be used to remove any groups that are no longer used.

3.26. sendbatch

This shell file is used to send batched articles to other systems. It is typically run from *cron*. See the manual page for more details.

3.27. sendnews

A program to send news internally from one computer to another. It is useful if you must use mail links to transmit articles.

3.28. seq

This file contains the current sequence number for your system. It is used to generate unique article id's.

3.29. sys

This file contains a list of all your neighbors, which newsgroups they get, and how to send news to them. The format is documented below.

3.30. unbatch

This program is used to unbatch the incoming batched news and feed each article to *inews*. It's horrible and will go away in the future.

3.31. users

A list of users that have read news on your system.

3.32. uurec

A program to receive news sent by *sendnews*(8).

3.33. vnews.help

This is the helpfile used by *vnews*.

4. Setting Up Links

There are two basic types of links for exchanging news: those that use mail and those that don't. The ones that use mail are more indirect, yet more versatile, while the ones that don't are simpler. The default method does not use mail, so that is discussed first.

4.1. Non-mail Links

The basic theory behind a non-mail link is that the *rnews* program is invoked on the remote system with the article being transmitted as the standard input. This is possible on several networks, but the most common implementation is via the UUCP network. Using the *uux* command, the command which is forked to the shell looks like:

```
uux – –r –z remotesys!rnews < article
```

This is the default transmission method. In order to set up such a link, obviously a UUCP link with the remote system must be in effect. In addition, *rnews* must be available and executable by *uuxqt* on the remote machine. In most cases, this means that *rnews* must be in */usr/bin* so *uux* can find it. Also, the list of allowed UUCP commands (in */usr/src/usr.bin/uucp/uuxqt.c* or */usr/lib/uucp/L.cmds*, depending on the version of UUCP) should be checked to make sure that *rnews* is an allowed command.

Other networks that allow remote execution include the BERKNET, BLICN (*usend*(1)), many Ethernets, and the NSC hyperchannel (*nusend*(1)). It is important, however, that a spooling mechanism be available. Otherwise, if system *A* tries to send an article to system *B* via a remote execution command, and *B* is down, the article could be lost. Spooling arranges that the system will try again when *B* comes back up.

4.2. Mail Links

When using mail to transmit articles, two intermediary programs are necessary. These are *sendnews* and *uurec*(8). The idea is that when system *A* wants to send an article to system *B*, the *sys* file on system *A* has an entry for system *B* such as:

```
/usr/lib/news/sendnews –a rnews@B
```

which runs *sendnews* on the article. The **–a** option specifies that the mail should be formatted for the ARPANET. *Sendnews* packages the article and mails it to ‘‘rnews@B’’. Somehow, the B system is expected to make sure that all mail to user ‘‘rnews’’ is fed as input to the program *uurec*. This program unpackages it and invokes *rnews*.

The best way to get mail to ‘‘rnews’’ fed into *uurec* is to use *sendmail* or *delivermail*, if you are on a system running them. Create an alias in */usr/lib/aliases* as follows:

```
rnews: "|/usr/lib/news/uurec"
```

and *sendmail* will handle it. If you do not have a facility for forwarding mail to a program, you can gimmick your mailer to watch for it (using *popen*(3S), this is easy) or, if you don’t want to do any programming, you can have *cron* invoke *uurec* every hour with */usr/spool/mail/rnews* as standard input. This solution is messier because *uurec* must potentially deal with multiple messages, something that has never been tested.

5. Format of the *sys* file

To set up a link to another site, edit the *sys* file in **LIBDIR**. This file is similar to the *L.sys* file of UUCP. Each line contains four fields, separated by colons:

(1) The system name of a site to which you forward news. Normally all systems you have links to will be included. You should also have a line for your own system. If this field is *ME*, it will be used as if it were your local system name. If the system name is followed by a ‘‘/’’, the article will not be forwarded to this system if it has already passwd through any of the (comma separated) list of sites immediately following the ‘‘/’’. For example, if the sysline was:

```
yoursite/sitea,siteb,sitec:net,mod,na,usa,to.yoursite::
```

the incoming article would only be forwarded to *yoursite* if it had not already been to any of *sitea*, *siteb*, or *sitec*. This is normally used to reduce the number of duplicate articles received at a site that has multiple main newsfeeds.

(2) The newsgroups to be forwarded to them. This is a pattern of the same kind as a subscription list. Generally, you will list classes of newsgroups, that is, using **all** for everything. A typical forwarding list for a new site would be

net,mod,na,usa,to.*sysname*

where *sysname* is the name of the remote system. (Of course, if you are not in the USA or North America, you would remove those distributions and replace them with the ones appropriate for you). In particular, you don't want to forward **all** since local newsgroups (those without dots) should not be sent. For the line describing your own system, this field describes the newsgroups your site will accept from remote sites. Thus, if another site insists on sending you a newsgroup you don't want, for example **net.jokes**, include **!net.jokes** here.

(3) This field contains flags describing the connection. An **A** will indicate that the other site is running an A version of netnews. A **B** indicates a B version. Leaving it empty defaults to **B**. If you are reading this document, you have a B version. Some existing sites run A versions. If you aren't sure, ask your contact at the other site, with whom you should be talking to set this up anyway. The **F**