An Introduction to the Berkeley Network

Eric Schmidt

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

May 1979
(revised March 1980)

ABSTRACT

This document describes the use of a network between a number of UNIX® machines on the Berkeley campus. This network can execute commands on other machines, including file transfers, sending and receiving mail, remote printing, and shell-scripts.

The network operates in a batch-request mode. Network requests are queued up at the source and sent in shortest-first order to the destination machine. To do this, the requests are forwarded through a network of interconnected machines until they arrive at their destination where they are executed. The time this requires depends on system load, inter-machine transfer speed, and quantity of data being sent.

The network enforces normal UNIX security and demands a remote account with a password for most commands. Information can be returned to the user in files, for later processing, or on the terminal for immediate viewing.

Introduction

A network between a number of UNIX machines on the Berkeley campus has been implemented. This document is a brief introduction to the use of this network. Information which is specific to the local network has been gathered into Appendix A. The new user should read both this introduction and Appendix A in order to learn to use the network effectively.

This document is subdivided into the following sections:

This manual is written in terms of three mythical machines, named X, Y, and Z. Specific names at Berkeley are in Appendix A, along with more local information.

Use of the Network

The network provides facilities for issuing a command on one machine (the *local* machine) which is to be executed on another (the *remote* machine). Network commands are available to transfer files from one machine to another, to send mail to a user on a remote machine, to retrieve one's mail from a remote account, or to print a file on a remote lineprinter. These commands are described below, as is the more general *net* command which allows users to specify the name of some command or shell script to be executed on a remote machine. Network requests are queued on the local machine and sent to the remote machine, forwarded through intermediate machines if necessary.

Most of the network commands require that you have an account on the remote machine. If a remote account is not needed for a particular command, it will be noted in the following discussion. The first example introduces procedures and responses which are applicable to all network commands.

1. Copying Files over the Network

Suppose that you have accounts on both the X and Y machines and that you are presently logged into the X machine. If you want to copy a file named 'file1' from your current directory on machine X to machine Y (the *remote* machine), use the command:

```
% netcp file1 Y:file1
```

The net will make a copy of 'file1' in your login directory on the Y machine. (The 'Y:' will not be part of the filename on the Y machine.) In order to verify your permission to write into the Y account, the *netcp* command will prompt you with:

```
Name (Y:your-name):
```

You should respond with your login name on the Y machine, followed by a carriage-return. If you have the same login name on both machines, just type a carriage-return. Next a password will be requested:

```
Password (Y:remote-name):
```

Now type in your password followed by a carriage-return (you must type it even if your passwords are the same on both machines). The *netcp* command will make a copy of your 'file1' in a queue destined for the Y machine, and will then return you to the shell.

Likewise if you wanted to transfer a file named 'scan.p' from Y to X,

```
% netcp Y:scan.p scan.p
```

would place that file in your current directory on X.

The network will "write" you when it has executed your request (if you are still logged in), or will "mail" you a message (if you are not). You may use the −**n** option (described later) to disallow the interruption and thus force mail to be sent. A typical message might look like this:

```
Message from Y:your-name at time ...
(command: netcp file1 Y:file1, R: 0, sent April 1 18:03, took 10 min 3 sec)
-------
```

The message includes the current time, the time you sent the command on machine X, and the exit code of the command (zero normally means success).

The network response will tell you if it was unable to execute the remote command successfully by returning an error message some time later. If, for example, you type the wrong password, you will get the response

```
Message from Y:your-name at time ...
(command: netcp file1 Y:file1, sent April 1 18:03, took 10 min 3 sec)
Error: bad login/password your-name
------
```

The *netcp* command is actually a generalization of the UNIX *cp* command, similar to *uucp*†. Its syntax is:

netcp [−**l** *login*] [−**p** *password*] [−**n**] [−**q**] [−**f**] *fromfile tofile*

where *fromfile* and *tofile* can be local or remote files. A filename which is not a full pathname is either from the current directory on the local machine or your login directory on the remote machine. The −**l** and −**p** options may be used to specify your remote login name and password on the command line. If the password contains shell meta-characters, it must be in quotes. (These options are useful in shell scripts, but be sure to make the shell script readable only by yourself if you've got passwords in it!) The −**n** option forces any responses from the remote machine to be mailed rather than written to you. The −**f** option forces prompting for a remote user name and password, even if they are set by other options or are in the ".netrc" file (see "Setting Defaults" below). Finally, the −**q** option prevents any confirmation messages from being sent back to you, if there were no errors, the exit code of the command is zero, and the command had no output.

Transferred files may or may not have the correct file protection mode; use the *chmod* (I) command to reset it. When files are to be brought from a remote machine, they are created zero-length at the time the command is issued; when they arrive, they assume their true length. Unlike *cp, netcp* does not allow the *tofile* to be simplified to a directory, if the files have the same name.

Examples:

% netcp file1 Y:file1	copy 'file1' from the current directory to Y
% netcp Y:file1 file1	copy 'file1' from Y to the current directory
% netcp Z:file1 Z:file2	*cp* command on remote machine
% netcp X:lex.c Y:lex.c	copy from X to Y
% netcp Y:subdir/file1 file1	copy from a sub-directory
% netcp file1 file2	an error— use the *cp* command

2. Listing Requests in the Network Queue

To see where your command is in the queue, type

```
% netq
```

A typical output of which looks like:

```
 From        To            Len   Code    Time           Command
X:your-name  Y:remote-name  100  b99999  Mar 23 18:05   netcp file1 Y:file1
```

The format is similar to that of the *lpq* command. The files are sent one at a time, in the order listed. If *netq* tells you the queue is empty, your request has been sent already. The queues for different destinations are totally separate.

```
% netq Y
```

will list just the queue destined for the Y machine. *Netq* summarizes requests from other users. The command

† See the UNIX Programmers Manual (Version 7 only).

```
% netq −a
```

will print the requests from all users.

3. Removing Requests from the Network Queue

If you want to cancel your net request, and “b99999” (see the *netq* example above) is your “Code,” use the command

```
% netrm b99999
```

which will remove the request (if it hasn’t already been sent). Furthermore,

```
% netrm −
```

will remove all your net requests in the queues on the local machine (you must have made the request in order to remove it).

4. Sending Mail over the Network

To send mail to remote machines, use the *mail* command with the remote account prefixed by the destination machine’s name and a ‘:’. “Y:schmidt”, for example, refers to an account “schmidt” on the Y machine. The full sequence is illustrated below:

```
% mail Y:schmidt
    {your message to user “schmidt” }
{control-d}
```

This will send to user “schmidt” on the Y machine the text you type in. As with intra-machine mail, the message is terminated by a control-d.

You do not need an account on a remote machine to send mail to a user there.

5. Reading Mail over the Network

It is also possible to read your mail on remote machines. From the X machine, the command

```
% netmail Y
```

sends a command to the Y machine to take any mail you may have and mail it back to you. As a precaution, the mail on the remote machine (Y in this example) is appended to the file “mbox”. Netmail has −**l**, −**p**, −**n** and −**f** options just like *netcp.* If a machine is not specified, the default machine† is used. If the −**q** option is specified (like *netcp)* no message is sent back if there is no mail.

Netmail also takes a −**c** option:

```
% netmail −c Y:username
```

which turns the command into a “mail check” command. A message is sent back telling the user whether the specified username has mail. No password is required. As above, the −**q** option suppresses the message if there is no mail. This command was designed to be put in C shell “.login” files.

† (see “Setting Defaults” below)

6. Using the Lineprinter over the Network

Remote lineprinters can be used with the *netlpr* command:

netlpr [−**m** *machine*] [−**c** *command*] *file1 file2 ... filen*

which sends the files its arguments represent to the lineprinter on *machine.* It will prompt you for an account and password. The −**l**, −**p**, −**n** and −**f** options may be supplied, as in the *netcp* command. If the −**c** option is specified, a different printing *command* (default is "lpr") can be specified; see Appendix A for the list of printers allowed. Copies of the files are not made in the remote account.

7. Net Prototype Command

The above commands all use internally one more general command—the *net* command which has the following form:

net [−**m** *machine*] [−**l** *login*] [−**p** *password*] [−**r** *file*] [−] [−**n**] [−**q**] [−**f**] *command*

Net sends the given command to a remote machine. The machine may be specified either with the −**m** option or in the ".netrc" file (for the specific names, see Appendix A). If not specified, a default is used. −**l**, −**p**, −**n**, −**q** and −**f** are as explained above for the *netcp* command. The −**r** option indicates the local *file* which will receive the output (the standard output and standard error files) of *command* when it is executed on the remote machine. By default this output is written or mailed to you. Thus, for example, to find out who is on the Y machine when you are logged in on the X machine, execute the following command:

```
% net −m Y "who"
```

which will run the *who* command on the Y machine; the response will be written or mailed to you. Similarly,

```
% net −m Y −r resp "who"
```

will take the output (result) and return it to you in file 'resp' on the local machine. If instead you want the result of the *who* command to remain on the Y machine the command

```
% net −m Y "who >resp"
```

will create a file 'resp' in your login directory on the Y machine. It is a good idea to put the command in quotes, and it *must* be in quotes if I/O redirection (<, >, or other syntax special to the shell) is used.

If you do not specify the remote machine explicitly (or in the ".netrc" file, explained below), the default machine will be used (see Appendix A).

The − option indicates that standard input from the local machine is to be supplied to the command executing remotely as standard input, thus if defaults for the login name and password are set up correctly as described below,

```
% net −m Y − "mail ripper"
    { message to ripper }
{control-d}
```

is equivalent to

```
% mail Y:ripper
    { message to ripper }
{control-d}
```

The net command also has other options not documented here. See the UNIX Programmer's Manual sections for more details.

Setting Defaults

Instead of repeatedly typing frequently-needed options for every invocation of the various network commands, the user may supply in his login directory a file ".netrc", which contains the repeated information. The ".netrc" file is typically used to specify login names on remote machines, as well as other options. An example of such a file is given below:

```
default Y
machine Y, login dracula
machine Z login dracula, quiet yes
```

This example sets the default machine to Y so that for net commands where a remote machine is not explicitly specified, the command will then be executed on the Y machine. The second and third lines indicate for the Y and Z machines a login name of "dracula" should be used to network commands, and to assume the "quiet" option on all commands destined for the Z machine. The complete list of options that may follow the machine indication is:

.netrc options for each machine

Option	Parameter	Default	Comment
login	name	localname	login name for remote machine
password	password	(none)	password for remote login name
command	command	(none)	default command to be executed
write	yes/no	yes	if possible, write to user
force	yes/no	no	always prompt for name and password
quiet	yes/no	no	like the −**q** option

In setting up the ".netrc" file, if the "default" option is present, it must be the first line of the file. The information for each machine starts with the word "machine" and the machine name and continues one or more lines up to another machine indication (or the end of the file). Input is free-format. Multiple spaces, tabs, newlines, and commas serve as separators between words. Double quotes (") must surround passwords with blanks or special characters in them.

How to Specify Remote Passwords

For the commands which require the password for the account on the remote machine, there are a number of ways to specify the password:

1) letting the command ask you, as in the *netcp* example in Section 1,

2) specifying it with an alias (if using the C shell),

3) putting it into the current environment if the local machine is running UNIX Version 7,

4) specifying it on the command line with the −**p** option,

5) storing it in the ".netrc" file, described in the previous section.

These can be ranked in order of security, from 1 = greatest security to 5 = lowest security, from the point of view of security of passwords from unauthorized use by other users and possibly an illicit super-user. Each is described in turn:

1) If you make no effort to specify the remote password elsewhere, the network commands will prompt you with:

```
Password(mach:username):
```

Type your password, followed by a carriage return. This is the most secure mode of specifying passwords. If the net command is executed in the background (i.e. with "&") then the command can't read the password from your terminal and one of options 2-5 below must be used.

2) The alias feature of the C shell can be used to specify the remote password. The command

```
% alias netcp netcp −l godzilla −p $pass
```

in the ".cshrc" file, followed by

```
% set path=your-passwd
```

right before using the network will set for subsequent *netcp* commands the login name "godzilla" and password "passwd". This alias command must be given everytime you login (see the UNIX Programmers Manual section for the C shell (csh (1)) for more information about *alias.* Do *not* put this alias command in your ".login" file.

3) If running on a Version 7 UNIX system, the password can be put in the current environment. The command (to the C shell)

```
% setenv MACHmch `netlogin −m mch`
```

or (to the default Version 7 "Bourne" shell)

```
% MACHmch=`netlogin −m mch`
% export MACHmch
```

will prompt you for a login name and password for the remote machine *mch* and put an encrypted copy of the password in your environment. (Note the back-quotes to the shell.) Subsequent network commands will find it in your environment and not prompt you for it. These encrypted passwords are invalidated after the user logs out. Type "man netlogin" for more information on the *netlogin* command.

4) Each net command takes a −**p** option on the command line to specify the password. These are usually put in shell command scripts. These shell script files should have file mode 0600 − use the chmod(I) command to set the mode.

5) The remote password can be specified in the user's ".netrc" file. If passwords are present, the ".netrc" file must have mode 0600 (as in #4 above).

The system managers recommend options 1-3 and warn against 4 and 5. Should someone break into your account on one machine, and you use option 4 or 5, you will have to change your passwords on all net machines for which your passwords have been stored in shell script files or in the ".netrc" file.

Log File

The file "/usr/spool/berknet/logfile" has a record of the most recent requests and responses, each line of which is dated. Lines indicating "sent" show the file name sent; lines indicating "rcv" show commands executed on the local machine (C:), their return code (R:), and their originator. For example, on the Y machine, the logfile:

```
Feb 28 10:29: rcv X: neil (neil) R: 0 C: netcp design Y:design
Feb 28 10:43: sent tuck to Z (z00466, 136 bytes, wait 2 min 3 sec)
Feb 28 11:05: rcv X: bill (bill) R: 0 C: netcp structures Y:structures
```

shows three entries. In this example, there are two *netcp* commands sending files from the X machine to Y, each from a different user. The second command sent was originated here by "tuck" and is 136 bytes long; the command that was sent is not shown. The command

```
% netlog
```

will print the last few lines of this file. Its prototype is

netlog −*num*

where *num* is an integer will print the last *num* lines from the file.

Acknowledgements

Special thanks go to Bob Fabry, Bill Joy, Vance Vaughan, Ed Gould, Robyn Allsman, Bob Kridle, Jeff Schriebman, Kirk Thege and Ricki Blau of Berkeley, and Dave Boggs of Xerox Parc for their help in making this network possible.

Appendix A

The Network at Berkeley

1. The Configuration (March 1, 1980)

	The Current State of the Berkeley UNIX Network			
Machine	Internal	Run	Default	Other
Name	Name	By	Machine	Name(s)
A	A	Computer Center	C	
B	B	Computer Center	D	
C	C	Computer Center	A	
D	D	Computer Center	C	
E	E	Computer Center	C	
Ing70	I	INGRES Group	IngVAX	Ingres
IngVAX	J	INGRES Group	Ing70	
Image	M	Sakrison	ESVAX	
ESVAX	O	EE-CE Research	CSVAX	
SRC	S	Survey Res. Cent.	D	
CSVAX	V	CS Research	Cory	
Cory	Y	EECS Dept.	CSVAX	
EECS40	Z	EECS Dept.	ESVAX	

If a path exists from the local machine to the requested remote machine, the network will forward the request to the correct machine. Thus Cory users may communicate with all the other machines on the network as well as C and CSVAX (with a degradation in speed because of the intermediate machine(s)). The links between Ing70−IngVAX, Ing70−CSVAX, A−C, C−D, C−E, and B−D run at 9600 Baud, the other links run at 1200 Baud.

2. Documentation

The network commands *(net, netq, netrm, netlog, netcp, netmail, netlpr, netlogin)* are all documented in the UNIX Programmers Manual. For example,

% man netq

will print the *netq* manual section.

There are two more documents available:

Network System Manual
Berkeley Network Retrospective

The Manual is intended for the systems staff who will maintain the network. The Retrospective is my Master's report and details the history of the project, discusses the design, and lists future plans.

There is an up-to-date news file:

% news net

or

% help net

or

% cat /usr/net/news {if those fail}

which prints news about the network, dated and with the most recent news first.

The UNIX Programmer's Manual, section I, has information on the *chmod, cp, mail, who,* and *write* commands mentioned in the text. Also, the *help* command has information about file protections:

% news access {on the Cory machine}

or

% help permissions {on the CC machines}

3. Features at Berkeley

a) There is a built-in character limit of 100,000 characters per single transmission, which cannot be overridden. The limit is 500,000 characters between the INGRES machines. Longer files must be split into smaller ones in order to be sent.

b) The 1200 Baud links between machines seldom transmit any faster than 50 characters per second (for 9600 Baud links, 350 characters a second), and can slow to a fraction of that in peak system loading periods. This is due to an expansion of the data packets to accomodate a seven-bit data path, wakeup time on the machines, and the packet sent in acknowledgement. Heavy file transfer is faster by magnetic tape.

c) On the CSVAX, IngVAX, and ESVAX the net commands are all in '/usr/ucb'. Your search path on these VAX's should be set to include the directory '/usr/ucb'; otherwise you will have to prefix all net commands by '/usr/ucb', as in '/usr/ucb/netcp'.

d) Limited Free Commands

Users who do not have accounts on remote machines may still execute certain commands by giving a remote login name of "network", and no remote password. The commands currently allowed are:

bpq
epq
finger
lpq
netlog
netq
ps
pstat
rcs
rcslog
rcsq
trq

The *lpr*
command is allowed on the INGRES machine.
Also,

mail
to remote machines and
netlpr
between Computer Center machines do not require a remote account.
The EECS40 machine allows no free commands (but allows the sending of mail).

lpq For example, to execute an
command on the A machine, the user would type:

e)
If no machine specification is in the front of a full path name,
the first four characters are checked and the machine
is inferred from that if possible.
In the command
the second file name is equivalent to "C:file1", if you are "schmidt"
on the C machine.

f) The network can only send files in one direction at a time.
Thus commands can slow down any file transfer.
If you regularly use a shell script to transfer a set of files, the
−q
option to
netcp
will improve transfer time.

g) The network throughput is adversely affected, and thus is expensive to
run. If general user charges are ever charges will be
implemented on the Computer Center machines.

h) When transferring files, quota overflows will result in a partial copy,
so you should check the space requirements of the file being sent.

i) The Computer Center "A" machine's phototypesetter is usable from other
network machines.
If on one of the B-E machines, you do not need an account on the A machine.
You simply type
instead of troff then sends it to the local machine and the phototypesetter
The troff command is executed on the A machine.
You will be sent mail both when the file is queued and when it is finally
typeset.
To see your place in the
troff
queue, type:
on any Computer Center machine.
There is a command
(where
code
is the code from the
trq
command)
to remove queue files before they have been typeset.
Trrm
must be executed on the same machine from which the job was submitted.

nettroff If you are on a non-Computer Center machine, you may use the
command:
which is similar to "troff −Q" command and the
You will need an account on the A machine.
trrm
command doesn't work from a non-Computer Center machine.

If using
nettroff,
no more than 15 pages may be sent to the typesetter.
If using
troff
more than 15 pages may be sent only if the
−s
option is specified (see troff(1) for more information).
The network will not transfer any file longer than

100,000 characters to the A machine.
(It is best to aim for files of 25,000 characters or less)†.

troff's
output, not the user's source files.
It is our general experience that
troff
outputs roughly twice as many characters as are in the source file
(before any
eqn
or
tbl
preprocessing.)

or
For more information, type

man nettroff {on the othe

The
nettroff
command is not supported by the Computer Center.

i) *netlpr* The
command allows "epr", "bpr", and "vpr" as alternate lineprinters
(using the −**c** option).

4. Bugs in systems at Berkeley (As of March 1, 1980)

a) If you are using the shell (/usr/pascal/sh, /usr/ucb/pascal/sh)
and have a ".profile" or ".shrc" file to change your
shell prompt, you must make sure that you don't turn on "prompting"
for non-interactive shells.
This will interfere with the net commands.
You should use this shell command to change your prompt:
where "P" is the prompt desired.
This will avoid the problem.

time If you set the variable
in the C shell, extraneous time stamps may appear in response messages.
The correct way to set the variable
time
in the C shell is

set time=*num*

endif
where
num
is the time interval in seconds.

b) *netcp* The file mode should be preserved by
and it should be possible to default the second file name to a directory
as in cp(I).

c) Various "response" messages are lost.
This includes "fetching" files where the file being retrieved never arrives.
I suspect this has something to do with unreliable delivery of error messages,
but this is not reliably reproducible.

d) The network makes no provision for "File System Overflow".
machines, such as "No more processes" or errors in transit on intermediate
While these occur only rarely, when they do, no message or
notification is sent to anyone.

e) The network commands are too slow on heavily-loaded instructional machines.
The
net
command has to read the password file, ".netrc" file and the
"/etc/utmp" file.

f) The queue files are normally sent shortest-job first.
Unfortunately, under heavy loading the queue search becomes
too expensive and the network will choose the next file
to send from the first 35 queue entries it finds in the queue directory,
so the user should not depend on the requests being sent shortest-first.

g) Comments and suggestions are encouraged and can be sent by
local or remote mail to “csvax:schmidt”.

Appendix B

Getting Started — An Example

The best way to start is to follow this example:
Suppose you're a Cory user, and you have accounts on the A and CSVAX machines.

1) Add a file “.netrc” (mode 600) to your login directory, as in
the following example:
machine A login *yourNameOnA*
machine CSVAX login *yourNameOnCSVax*
(If “default” occurs, it must be the first line of the file.)

2) Make sure that,
a) if you are on one of the VAX's, you have in your search path
the directory ‘/usr/ucb’.
Otherwise, on those machines you will have to prefix all commands by ‘/usr/ucb/’
(e.g. ‘/usr/ucb/netcp’).

b) on the Computer Center machines, if you choose to set your shell prompt,
you have done so correctly (details in Appendix A).

3) Then type

```
% netq
w
```

which will send a
command to the CSVAX; some undetermined time later you will
have written (or mailed) to you the output from the command
executed on the CSVAX machine.

The effect of the command may try:
net with the effect of being routed more slowly through an intermediate link in the