The numerical analysis net (NA-NET)

ETH

Eidgenössische
Technische Hochschule
Zürich

Institut für Informatik
Fachgruppe
Hochleistungsrechner

M. Kent

The Numerical Analysis Net (NA-NET)

January 1988

85

Author's address:

Department of Computer Science
Stanford University
Stanford, California 94305
USA

Abstract

The NA-NET has helped numerical analysts to communicate via electronic mail since early in this decade. Originally conceived and constantly maintained by Gene Golub, the NA-NET has grown in both members and capabilities. This report chronicles the development of the NA-NET from the early stages, to the present system. The future system, as we envision it, will also be discussed.

The numerical analysis community has strong contact together. This is due to the fact that leading numerical analysts like Germund Dahlquist, Gene Golub and the late Jim Wilkinson have always been open minded and willing to talk to other people.

The efforts of Gene Golub to maintain international contacts through electronic mail created an institution called NA-NET (free of charge!), that helps numerical analysts to feel even more like a big family.

I am very pleased that Mark Kent wrote in this report the history and the service of NA-NET which surely has grown to a facility that we would all miss if it were no longer available.

Walter Gander
Zürich, October, 1987.

1 The NA-NET

1.1 CAPABILITIES. The NA-NET exists to provide a means for people involved in numerical computation to easily communicate with each other. The NA-NET supports both individual and broadcast communication. The main feature of the NA-NET is the uniformity of addressing. All mail is addressed to the Arpa-Internet host `score.stanford.edu` at Stanford University. Here the mail is processed and forwarded to the intended recipients of the message. Hence, users need to know only how to get mail to `score.stanford.edu`, and need not be otherwise concerned with the complications of trans-network mail routings.

Admittedly, the domain naming system [4] will reduce the need for such a convenience as the NA-NET. The simplified addressing scheme will reduce the complexity of trans-network communication (from the users viewpoint). However, even years after the proposed implementation date the system is not fully operational. Additionally, the NA-NET provides other functions besides address simplification.

The basic functions of the NA-NET are:

Broadcast messages: Mail to `na@score.stanford.edu` is broadcast to all the members of the NA-NET. The messages also reach an unknown number of people who are not "*members*" of the NA-NET.[1] This is a moderated service, *i.e.*, the messages are queued up until the moderator actually broadcasts the message. Gene Golub is the primary moderator although Cleve Moler has assumed these duties for the 1987-88 academic year.

Individual messages: Each member of the NA-NET is assigned a key that is unique within the NA-NET. This key is usually the last name of the member, but because of duplications is sometimes modified. Mail to `na.(lastname)` is forwarded to the member with key `lastname`. For example, Gene Golub is

[1] A "member" will be a person or institution whose name appears in the main NA-NET database.

`na.golub@score.stanford.edu`. Mail so addressed will be automatically forwarded to the machine where Gene reads his mail, whether it is in Stanford or elsewhere. Hence, members of the NA-NET need not know where an individual is at any moment, much less know his correct address. As long as moving members alert the maintainer of the NA-NET of their new address the system works smoothly.

Non-unique last names are usually resolved by the prepending of the first initial to the last name. For example, if John Doe and Philip Doe were both members then `na.jdoe` would point to John Doe and `na.pdoe` would point to Philip Doe. Mail to `na.doe` would be returned to the sender with a message indicating that `na.doe` is ambiguous and that the possibilities were `na.jdoe = John Doe` and `na.pdoe = Philip Doe`.

Institutional mailing: There are a few `na.(lastname)` entries that address institutions rather than individuals. For example, mail addressed to `na.argonne@score.stanford.edu` would reach the numerical analyst community at Argonne National Laboratories. The locations that can be referenced in this way is currently limited to those that have a local mailing list implemented. Hence, `na.argonne` forwards to `ml_na@anl-mcs.arpa` which is a mailing list set up by the researchers at Argonne. Future versions of NA-NET will have a larger set of institutions and groups addressable by using the `na.(lastname)` facility.

Netlib: A separate facility, whose development was spurned on by the use of the NA-NET, is `netlib`. This was developed by Jack Dongarra of Argonne National Laboratories and Eric Grosse of Bell Telephone Laboratories. By simply mailing to `netlib@anl-mcs.arpa` or `netlib@research.uucp` users can obtain software from such well known packages as EISPACK, LINPACK as well as other contributed software. Over 32,000 requests were received (and subsequently serviced) by `netlib` in 1986 and over 43,000 requests have been processed in the first 9 months of 1987. More information can

be obtained by mailing to either address above with the message

`send index`

or by consulting [1].

1.2 COMMON USAGE. The NA-NET is used for different purposes by different members. Some are content to just read the broadcast messages and do not otherwise participate.

There seems to be a core group who use the `na.(lastname)` facility almost exclusively for communication within the numerical community. The `na.(lastname)` facility processes approximately 1000 requests a month and has well over 800 entries.

The most common type of messages mailed to the broadcast mechanism are conference announcements and conference schedules. Users even enroll in the conferences over electronic mail.

Members also request comments on books, software or hardware. Occasional reviews of products of interest to the numerical community are given. The NA-NET has also proved useful to those looking for unpublished notes or reports.

The NA-NET has also been used to pose problems and discuss possible solutions. One example of this has become known as *Trummer's problem.* On July 22, 1985, while giving a talk at Stanford, Manfred Trummer described a bottleneck in a certain computational procedure. Chris Anderson noted that the same problem occurs in the sort of calculations that he was engaged in and if the solution could be obtained quickly then it would help him a great deal too. Gene Golub proposed that since the problem was so easily stated that it should be broadcast over the NA-NET.

The problem generated a lot of discussion and partial results were quickly obtained and presented on the NA-NET. A paper written by Gastinel (in French) was unearthed at Stanford and the translation was mailed to interested parties. On October 30, 1985, a team from Rutgers presented a solution on the NA-NET and subsequently in [2].

Canada (60)	England (33)	W. Germany (25)
Sweden (19)	Australia (19)	Italy (18)
France (14)	Belgium (13)	Switzerland (13)
Norway (12)	Netherlands (11)	Israel (10)
Denmark (7)	Scotland (3)	Japan (3)
Ireland (3)	Finland (3)	Austria (3)
New Zealand (2)	Spain (1)	

Table 1.1: Distribution of NA-NET members outside the US.

1.3 DEMOGRAPHICS. The NA-NET distribution list has approximately 800 entries while the `na.⟨lastname⟩` facility has over 1000. Some of the entries in the distribution list reference "exploders" at remote sites and hence some of the people in the `na.⟨lastname⟩` facility do not appear in the distribution list.

Internally, members of the NA-NET are categorized by location. Typically, if there are at least two members from the same geographic region then they will be grouped together. Most of the members of the NA-NET are from the USA and are grouped according to State or institution. Those outside the USA are grouped as shown in in Table 1.1. The groups aid the maintainer of the NA-NET to ensure that an address is correctly specified.

Representation in NA-NET is not just from academic institutions as Table 1.2 shows. There are a substantial number of US research laboratories represented but only three foreign research laboratories: Defence Research Establishment (Canada), Inria (France) and the National Physics Laboratory (England).

There is also an interesting mix of computer companies from the older companies like IBM and DEC to new start-up companies like NeXT and Ardent.

At least three software suppliers are represented. A few people from IMSL and NAG are members and also MathWorks, the company that develops and markets `matlab`, is on the NA-NET. Additionally, administrative personel from the Society for Industrial and Applied Mathematics (SIAM) can be reached through NA-NET.

Air Force Office of Scientific Research	Ardent
Argonne National Laboratory	Ballistics Research Laboratory
Bell Telephone Laboratories	Boeing Computer Services
Bolt, Beranak & Newman	Convex
Defence Research Establishment	Digital Equipment Corporation
General Electric	General Motors
Hewlett-Packard	IBM
ICASE	INRIA
Intel Scientific Computers	Lawrence Berkeley Laboratory
Lawrence Livermore Laboratory	Los Alamos National Laboratory
NASA	National Bureau of Standards
National Center for Atmospheric Research	National Physics Laboratory
National Science Foundation	NeXT
Oak Ridge National Laboratory	Philips
RIACS	Sandia National Laboratory
Shell Oil	Sperry
Sun Microsystems	Symbolics
Tektronix	Thinking Machines

Table 1.2: Members of the NA-NET include people from the above organizations.

1.4 ELECTRONIC MAIL ADDRESSES. Electronic mail addresses can be quite complicated and difficult to type. Addresses *should* become simplified when the domain naming system becomes universal, but even some of these addresses are cumbersome. Users at many installations can set up aliases within their mail system. In this way a user could simply address a message as `gene` and have the mailer expand this to `golub@score.stanford.edu`. However, on some systems it is not possible to set up personal aliases and on others aliases are not possible at all.

In this section we take a not too serious look at some current addresses.[2] When looking at these addresses try to imagine what it would be like to have to explain to a novice computer user how to mail to the specified address. Remember that on some systems certain characters must be "escaped" and so even a simple address

[2] All addresses shown are the addresses necessary to reach the party from `score.stanford.edu` in October, 1987. Apologies are extended to the owners of these addresses.

like `a!b!c` would have to be entered as `a\!b\!c`.

• Some of the problems with mail addressing occurs because users account names do not always appear to be logical to outsiders. For example,

annu76%buclln11.bitnet@forsythe.stanford.edu

Perhaps if we knew what the hostname (`buclln11`) stood for we would also gain some insight as to what the string 'annu' refers to. Whatever it is we know that there are at least 76 of them. In this example the hostname is also a possible source of error because of the similarity between the letter 'l' and the number '1'.

• The next example has an eight digit user name and we wonder whether they actually have over 75 million users.

75003678%vax2.nihed.hea.irl@forsythe.stanford.edu

Admittedly, in the domain naming system the address may become

75003678@vax2.nihed.hea.irl

but this is still quite a cryptic address (and even now there is some controversy about the country code '`irl`').

• Imagine trying to type this address without error:

mnetor!calgary!vuwcomp!auccvaxb!csc_burrage@uunet.uu.net

Note that if the host `auccvaxb` were mistyped it could take a few days before you were alerted of the fact.

• Some seem over redundant:

mcvax!inria!inria.inria.fr!lemarech@uunet.uu.net

stratis@uicsrda2.csrd.uiuc.edu

munthe_kaas%vax.runit.unit.uninett@nta-vax.arpa

The first address might become `lemarche@inria.inria.fr`, but why not simply `lemarche@inria.fr`? Working from the right in the second address we deduce that the address references an educational institution, that is in fact the Univeristy of Illinois at Urbana-Champagne. The next field indicates that it is the

Computer Systems Research Department. But then, the final field repeats some of this information with '`uicsrd`'. This is probably a result of naming the machine before the domain system came into effect. The name was probably something like `uicsrda2.arpa`.

What do you suppose the `uni` stands for in the latter address? Note that the under-bar (_) in the latter address is replacing a hyphen in the users real name.

• The length restriction on users names produces some unfortunate instances. Suppose you were successful in typing the following address to Professor M. Shimasaki but put the 'h' in the user name:

```
simasaki%kudpc.kudpc.kyoto-u.junet%utokyo-relay.csnet@relay.cs.net
```

• Some user names seem amazing:

```
"perkins a. louise%b.mfenet"@nmfecc.arpa
```

```
"na %uk.ac.bradford.central.cyber1"@cs.ucl.ac.uk
```

Those blanks are actually there! The last example points out another source of confusion. In the `uk` domain the most general domain is on the *left* of the address while elsewhere in the world the most general domain is on the *right*. This is, of course, similar to the style of driving in the UK.

• Think about attempting to mail a message to the following address from a host that is linked via uucp to a csnet host:

```
bfdf2544%vax1.centre.queens-belfast.ac.uk@nss.cs.ucl.ac.uk
```

• Many users express confusion over the use of the characters `!`, `%` and `@`. They would find little relief in the following address:

```
aukuni.ac.nz!john-b%waikato.ac.nz@relay.cs.net
```

• Poor Chet Koblinsky has to live with just a numeric Internet specification but compared to Bo Kagstrom it is hard to say who is worse off:

```
koblinsky@[128.8.250.3]
```

```
bo_kagstrom_university_of_umea%qzcom.bitnet@forsythe.stanford.edu
```

At least it is quite clear where Dr. Kagstrom works.

• Some addresses (as we have already seen) have a mixture of domain-style addresses and other styles:

`jg105%cam.phx%uk.ac.cam.eng-icf@cs.ucl.ac.uk`

• These next three addresses refer to people who work at the *same institution*!

`enea!ttds!capella!duvan!gustafs@uunet.uu.net`

`Germund_Dahlquist%qzcom@mit-multics.arpa`

`enea!draken.nada.kth.se!jespero@uunet.uu.net`

• Of course some addresses are particularly good:

`karp@ibm.com`

but after seeing

`gp1@ibm.com`

we are left wondering "*is there a* `gp2` *or* `gp0` *?*"

2 The first generation

The NA-NET started early in this decade and is now in the "*second generation*" of development. This chapter gives some of the history of the net and describes some of the problems encountered that led to the second generation.

2.1 GENESIS. In the beginning (1969) there was the Arpanet; and the Arpanet begat electronic mail; and Gene Golub saw that it was good and started using it (1978). Life was particularly simple then since Stanford was right on the Arpanet and had good connections and the only people who had mail addresses were all on the Arpanet. True, the UUCP network started in 1978 too, but it was probably only the systems-type of people who were using UUCP to communicate at that time.

During one of his visits to Stanford, Jim Wilkinson asked Gene if there was a convenient way to contact numerical analysts by electronic mail. Gene thought that he should put all the addresses he knew (about 20) in a file and make this "distribution list" available to other people. This was the start of the file `na.dis` on what was known then as `su-score`.

Later, the system manager for `su-score`, Mark Crispin, set up the first `na.`⟨`lastname`⟩ facility. This took the form of explicit aliases in the system file `<mail>mailing-lists.`{`txt,bin`} of the form

```
na.golub= golub@su-score
```

To change the system mail aliases it is necessary to edit the text file (extension `.txt`) and then rebuild the binary (extension `.bin`) file using a special program. The mail system looks up the addresses of incoming mail in the binary alias file to determine the forwarding address.

2.2 OPERATION. At conferences and meetings Gene would try to get other numerical analysts interested in electronic mail and get their addresses for inclusion in `na.dis`. Adding someone to

the list was fairly simple. The file `na.dis` is simply a text file with entries like

```
!lastname, first!        user@address,-
```

The exclamation mark delimits comments and each line ends with a comma followed by a dash. Gene kept this list sorted and merely inserted the appropriate record in the file using a text editor.

Changes to the `na.⟨lastname⟩` facility required the intervention of system personnel. The alias file was considered a fairly important system file and hence only select people could change it. Gene would mail a message to the `score` system manager with the required updates, which could be deletions, changes or additions.

Finally, Gene would mail three files to the new members: one was an introduction to the NA-NET, another was a few of the "back-issues" of broadcast messages, and the last was a copy of `na.dis`.

Everything worked fairly well. The `su-score` mailer, `MM`, was an excellent mailer[1] and, as mentioned previously, `score` had excellent connections to the Arpanet.

2.3 PROBLEMS. Over time problems developed. These problems were mainly caused by the increased size of the list as well as the development of other networks thus adding to the complexity of the addressing schemes. Adding members was no longer trivial. A message sometimes passes through a few mailers on route to the NA-NET maintainer, and a few years ago not all mailers were careful with what they did to the message header. Hence, the return address was often difficult to determine.

To add a person it was necessary to maintain four views: one in `na.dis`, one in the list of changes to be made to the `na.⟨lastname⟩` facility, one in a new (temporary) mailing list of new members, and one in the incoming mail messages. To avoid typing addresses, the editor was used to move the information around in the four views. This is quite challenging on a 24 line by 80 character terminal.

[1] In fact, it is known as the Cadillac of mailers in some areas (the Mercedes of mailers in Europe).

The use of multiple windows on a workstation helped matters, but it was still cumbersome.

An additional complication is the generation of the key to be used by the `na.⟨lastname⟩` facility. With a large list of people it is hard to remember whether there is already a `na.newman` and it is necessary to search the list. In the unfortunate circumstance that there is a clash then two changes must be made: one to change the preceding entry and one to add the new one.

Other problems come from the system side. The system alias file is a binary file that is created from a human-readable file. The large number of `na.⟨lastname⟩` entries were causing the rebuilding of the binary file to take over 15 minutes.

Inevitably, due to errors and omissions, the data in the system alias file that drives the `na.⟨lastname⟩` facility and the data in the file `na.dis` diverged. Since the maintainers of the NA-NET rarely looked in the system alias file this was a problem that went unnoticed for some time.

One problem that affected everyone on the net was inadvertent replies. Mail to `na@score` went directly to the members of the list with no human intervention. The message would be broadcast with the `From:` field indicating the original sender and the `To:` field indicating `na@score`. Occasionally a user would reply to a broadcast message and, because the wrong reply command was used (for example, capital `R` instead of small `r`) or because of a deficient local mailer, the reply was also sent to `na@score`. If a message was sent out announcing a conference, then almost inevitably, this message would be quickly followed by a reply message stating for everyone's benefit

```
Yes, I am interested.
Please send me further information.
```

This is relatively harmless but a few embarrassing messages have slipped out inadvertently.

The cure to this problem was provided by the `MM` mailer. The mail to `na@score` would be forwarded to Gene. He would then remail the message to a secret address in such a way that the `To:` field of the outgoing message would be simply

```
To: :;
```

No where on the outgoing message was the address **na@score** listed. Hence, it was not possible for mailers to automatically put this address on the list of recipients of a reply message.

The empty `To:` field violated some of the rules that are supposed to govern mail transfer but it solved our problems. The csnet gateway would automatically complain with every broadcast message. Some primitive mailers on new networks would just look at the message header rather than the SMTP "envelope" information [4] and would come up empty handed.

Some members of NA-NET were also confused because they would receive a message in their mailbox with no indication of who the message was to. "*Was this a message meant just for me?*"

There was, of course, the delay caused by having broadcast messages wait for human intervention before jetting to their destination. When Gene was at Stanford, or travelling to somewhere on the Arpa-Internet, the delay was usually less than a day. But when Gene travelled to areas with limited network connections the delay could be much more. When users mail a message to be broadcast they like to see it in their own mailbox before the end of the day. If not, they suspect that something is wrong and, typically, send a message asking what the hold up is.

Also, even though the mailer **MM** supports the concept of *remailing*, many other mailers do not. Although the `From:` field and/or the `Reply-To:` field clearly indicated the original sender, Gene would continue to get replies that were intended to go to the original sender.

Another minor problem was the fact that if the system administrator was away then the **na.**⟨`lastname`⟩ facility could not be updated.

In spite of all these problems the NA-NET continued in this fashion until October of 1986 when the development of the second generation was started.

2.4 GROUPS IN **na.dis**. Near the end of the first generation a few sites implemented local mailing lists for people interested in

numerical computation. The addresses of the mailing lists were included in `na.dis` and the individuals on the remote mailing lists removed from `na.dis`. This shortened the length of time required by the `score` mailer to process the list.

Although the `score` mailer would only send one copy of a broadcast message to each Arpa-Internet site (regardless of the number of recipients at that site), remote mailing lists did reduce traffic on various other networks. The Australians were probably the first to set up a remote mailing list expressly to reduce traffic. This happened after a flurry of messages on NA-NET caused the only connection to Australia to be tied up for days.

It was quickly recognized that the `na.⟨lastname⟩` facility could now have a `na.australia` entry. Other places were asked to set up remote mailing lists and the number of new "institutional" entries in the `na.⟨lastname⟩` facility grew. Some convenient entries were set up by using local mailing lists on `score`. Unfortunately, the second generation saw the demise of the `score` mailing lists.

The list of new groups available to the members of the NA-NET was broadcast over the net. Apparently some feelings were hurt. Some groups reported that they must have been overlooked and so these people were hastily added. Admittedly, the initial groups were set up according to the bias of the maintainer at that time. Hence there was a `na.canada`, `na.canadians`, `na.halifax` and `na.toronto`. Table 2.1 shows the institutional mailing list at its fullest. Many of these are not supported in the second generation of NA-NET but it is hoped that they will be in the third generation.

The file `na.dis` was restructured to reflect the new groups. Unfortunately, this added a little bit of extra complexity to the routine for adding a new person.

There are also problems with the remote mailing lists. The maintainers lost a lot of control over the operation of the net. It was no longer known who received the broadcast messages. Some of the remote mailing lists rebroadcast messages to dozens of other installations. One remote mailing list in `na.dis` was without a name for sometime. All that was known was that it was somewhere in Canada. So, a message was sent to the list asking

ARGONNE	AUSTRALIA	BC	BELL
BOULDER	BRL	CANADA	CANADIANS
CLEMSON	CORNELL	DREA	DUKE
HALIFAX	IBM	ICASE	ILLINOIS
LANL	LBL	LLL	MARYLAND
MICH	MIT	NORWAY	NYU
ORNL	PENN-STATE	PITTSBURGH	PURDUE
RIACS	RICEU	RUTGERS	SANDIA
STANFORD	TEXAS	TORONTO	UCLA
UK	WAYNE	WATERLOO	WISC
YALE			

Table 2.1: The institutional entries in the first generation `na.(lastname)` facility.

for the people to identify themselves. No reply was received. A week later, another message was sent and one reply was received. It stated that the address was "the gateway to the CDN-net" and gave no more information.

Of course the broadcast messages are meant for all those who are interested in reading them. However, problems occur when an address is in error on a remote mailing list. Often a maintainer for the remote mailing list can not be found. It is also difficult at times to identify what entry in `na.dis` caused the reference to the bad address. Hence, error messages are returned to the sender and remailer of the message. This can be quite annoying. The author received well over one megabyte of error messages during a recent one month period. A large fraction of these were due to problems in remote mailing lists. Incidently, none of the more than 1Mb worth of errors could be fixed in the NA-NET files. This is typical and also frustrating. The following all can, and often do, cause some sort of error message to be returned:

- a host is down;
- a mailer does not know its own host name and rejects mail;
- a user is over his disk allocation;
- a relay host has a broken name server;
- a relay host has strangely non-monotonic information about the hosts it can reach.

3 The second generation

The burden of maintaining the NA-NET became heavier over time and so Gene Golub enlisted the help of two of his students, Mark Kent and Ray Tuminaro, to help with the maintenance chores. A short while later Mark and Ray decided to make some changes that would make life easier for the NA-NET maintainers. They had two specific goals:

1. get control of the `na.⟨lastname⟩` facility. Updates should be possible without the aid of system personel.
2. make a central database from which the broadcast distribution list *and* the entries for `na.⟨lastname⟩` would be generated.

It was felt that the external characteristics of the NA-NET could not be changed. Hence, everything still had to revolve around `score` and the `MM` mailer on the TOPS-20 operating system.

3.1 AN IMPROVED `na.⟨lastname⟩` FACILITY. The first step in gaining more control of the `na.⟨lastname⟩` facility was to move it to a UNIX[1] system. The host `navajo.stanford.edu` was chosen. Here a system alias was created that pointed to a program in a directory controllable by us. A change was made to the `score` mail system so that all mail addressed to `na.`x would be sent to the alias on `navajo`. All of the `na.⟨lastname⟩` entries in the `score` system alias file were removed, much to the relief of the `score` system administrators.

The program on `navajo` is activated whenever mail is sent to the alias. The incoming mail message is read in by the program and the intended recipients are extracted from the various header fields. Additionally the address of the original sender is extracted. All of the `na.⟨lastname⟩` addresses are looked up in a file called `na.xx` and the members actual address is determined. The message is then routed through the mailer on `navajo`, with the complete recipient addresses specified. Additionally, the message is

[1] UNIX is a trademark of AT&T Bell Laboratories.

sent so that it appears as if it is from the original sender rather than the mailer daemon on `navajo`.

If a `na.⟨lastname⟩` key cannot be found then the mail message is returned to the sender with a suitable error message. However, any correct addresses will still reach their destination. The program could send as many as three mail messages in one invocation: one to the correctly specified recipients, one containing error messages to the original sender and one to the maintainer of the NA-NET if there are any system problems.

An extra convenience was included in this second generation NA-NET. Since not all of the family names are unique within the net some entries have been modified, usually by the inclusion of the initial of the first name. For example, there are two members with the last name Hall: Chuck Hall (`na.chall`) and George Hall (`na.ghall`). If mail is addressed to `na.hall` then it will be returned with a message indicating that the address is ambiguous and the list of members with last name Hall will be reported.

The file `na.xx` is a text file with a very simple format that can be changed at any time by the maintainers of the NA-NET. There are two types of entries: one for ambiguous last names and the other, regular, entry. The entry for an ambiguous last name looks like

```
hall      -2
$$hall, chuck = na.chall
$$hall, george = na.ghall
```

The `-2` indicates that it is an ambiguous entry with two possibilities on the following lines. The regular entry has the form

```
chall      1
$$"up-icma!cahall%pitt.csnet"@relay.cs.net
```

Some members have more that one address in the `na.xx` file, for example,

```
boley      2
$$boley@umn-cs.arpa
$$dlb@score.stanford.edu
```

3.2. The database.

One problem with the present arrangement is due to the fact that we didn't want to do any system programming. Hence we only look at the message header to determine the recipients rather than the SMTP list of recipients [3]. Thus, blind carbon copies (`bcc:`) are not supported.[2]

There is also an extra level in indirection in this second generation implementation since all mail goes first to `score` and then to `navajo`. Additionally, on `navajo` the mail actually leaves the control of the mailer, is handled by our program, and the given back to the mailer.

The second generation `na.(lastname)` facility was implemented in the C programming language.

3.2 THE DATABASE. The second major part of the second generation NA-NET is the database and associated database handler. The database was designed so that it contained all the information needed to build both `na.dis` and `na.xx`. The Prolog language was used to define and handle the database. There are interactive commands for queries and updates and batch updating commands. Updates are rarely made interactively since we want to avoid retyping addresses.

The operation is as follows. Requests for updates are put in a file on `score` called `na.change`. Incoming requests are edited to extract the pertinent information and then an entry in `na.change` is made. The use of the editor means that we never have to type in an address. Periodically, the file `na.change` is taken from `score` to `navajo` where it is processed by the Prolog program. At the end of the updates the new files `na.dis` and `na.xx` are written and put in their proper places (one on `score` one on `navajo`). A record of the changes made is kept for future reference (or debugging).

3.2.1 Key generation. The key generated for new members is usually the last name of the member. The Prolog program automatically handles non-unique last names. The following is the output generated from the command

[2]The intention of a 'bcc:' is to send a copy to an individual without the knowledge of the primary recipients.

```
new(wheeler,ed,usabitnet,'elee1zh%uhvax1.bitnet@forsythe.stanford.edu')
```

in the file na.change:

```
| ?- We already have mary wheeler in group rice.
Adding
na.ewheeler=ed wheeler in group usabitnet
          on mail list(s) [na]
          with address [elee1zh%uhvax1.bitnet@forsythe.stanford.edu]
          and status all
Done.
Here are the other keys: [wheeler,ewheeler]
Deleting na.wheeler=mary wheeler in group rice
          on mail list(s) [na]
          with address [rice!mfw123@rice.edu]
          and status all
Adding na.mwheeler=mary wheeler in group rice
          on mail list(s) [na]
          with address [rice!mfw123@rice.edu]
          and status all
Adding na.wheeler=x wheeler in group ambig
          on mail list(s) []
          with address [mwheeler,ewheeler]
          and status x
```

The original na.wheeler is deleted and then a new record is created for na.mwheeler. Finally, a record for na.wheeler is created in the group ambig (for ambiguous). This record lists all the keys for members with the last name Wheeler. When the file na.xx is written the following entry would be produced:

```
wheeler     -2
$$wheeler, mary = na.mwheeler
$$wheeler, ed = na.ewheeler
```

together with entries for na.ewheeler and na.mwheeler.

If the simple technique of appending the initial of the first name is not sufficient to create a unique key then the program reports this in the log. The entry is still made but the new person will not appear in the file na.xx. Manual intervention is then necessary to create a suitable key.

3.2.2 Database format. The database is composed of regular Prolog clauses that have the following form:

```
na(last, first, key, group, status, [mailing_lists], [addresses]).
```

Here `last` and `first` are the last and first names, `key` is the `na.`⟨`lastname`⟩ key. Members are grouped according to geo-political areas and `group` is set to the members group. The `status` field determines what addresses are written to the file `na.dis` and is one of

`all`: all of the addresses in the address list should appear.

`none`: no address should appear.

`remote`: this person is on a remote mailing list and so no address should appear.

`first`: only the first address in the address list should appear.

The list `[mailing_lists]` is a list of the mailing lists that this member is on. Currently only the mailing list '`na`' is used but other lists could be specified. If a mailing list called, say, `seminar` were defined, then a file `seminar.dis` would be written at the end of each update session.

The list `[addresses]` is a list of all the electronic mail addresses for the member.

For interactive use, a set of mail address *templates* are defined. In the database these take the form

```
template(group, mail_template)
```

The `mail_template` demonstrates the form of usual address for the group. This can be an aid when adding long, complicated addresses.

If a record has the special group `ambig` then the address list is a list of unique keys that are associated with the key of the record. For example, the record

```
na(wheeler,x,wheeler,ambig,x,'[]',[mwheeler,ewheeler]).
```

correponds to the example given above.

In summary, the second generation system produces/uses the following files:

`na.dis`: the main distribution file. Read by the `score` mailer, written by the Prolog program.

`na.xx`: the file used for the `na.`⟨`lastname`⟩ facility. Read by the `C` program on `navajo`, written by the Prolog program.

`na.dat`: the database file. Read and written by the Prolog program.

`na.pretty`: a pretty version of `na.dat`. Read by humans, written by the Prolog program.

`na.change`: the file of updates to be made to `na.dat`. Read by the Prolog program, written by the maintainers of the NA-NET.

3.3 THE DIGEST. During the second generation of the NA-NET a change occured that was independent of the developments mentioned elswhere in this chapter. The change was the introduction of the *digest* form for the broadcast messages. Messages sent to `na@score` would be appended to a file on `score` called `na.incoming`. The moderator reads the messages in this file, edits them, and then issues a command that sends all the accumulated messages in one package called a digest.

An administrator of `score` suggested to Gene that this format was a good idea because it would reduce the load on the system. It was taking over twenty minutes for the mailer to parse `na.dis` and get messages on their way. The magic command to initiate broadcasting of the digest was to be given only at night.

The digest format was adopted, but not just to accomodate the `score` system administrators. The digest cured some of the problems mentioned previously. The `To:` field is no longer empty and there is an `Errors-To:` field that actually seems to work! (*I.e.*, other mailers seem to respect the field). An address is given to receive error messages and so these are kept separate from other mail.

However, since the digest format was adopted the number of submissions to the net has reduced dramatically. Additionally, reports are heard that people don't read the digest anymore. When it is issued it is just too big for someone to interrupt their work

and read five independent messages all bunched in one mail message. Also, there is no chance to reply to a sender of a particular message; users must type in the return address themselves.

Of course the digest is misnamed. It is not a digest since (usually) the messages are relayed verbatim.

4 The third generation

The third generation of the NA-NET is in the design process as this report is being written. Plans are for the system to be as automatic as possible, requiring human intervention only to handle unforeseen errors. Details still have to be worked out, but the following summarizes the general plans.

The NA-NET will be moved to a machine that is totally under the control of the maintainers, and said machine will be almost totally devoted to the NA-NET. The machine `score.stanford.edu` has been associated with NA-NET from the beginning and it is expected that this will be the case for some time. People will still mail to some address at `score...` *unless* the new machine has a suitable name that is easily remembered and associated with the NA-NET. Hence, the machine will be named `na-net.stanford.edu`. This will render the prefix `na.` redundant and it may be dropped from the `na.`⟨`lastname`⟩ facility (and hence the facility itself may have to be renamed). The same keys will be retained.

The `na.`⟨`lastname`⟩ aliases will reside in the system mail alias file and so messages will not be removed from the control of a mailer. This will bring back the capability of using the `Bcc:` field. An exception to this will be the ambiguous last name entries. These will still point to a program that returns the mail with a suitable message attached.

Direct feed for broadcast messages may be revived. Provisions for catching inadvertent replies are being planned. The mailing of broadcast messages will be spread over eight processors (more as they become available) and this should speed up the handling considerably.

A `netlib`-like program will handle the updates to the system automatically. A syntax will be defined for requests to the system and if a users mail message does not conform to the syntax then it will be returned with a help message attached.

Only proven addresses will be inserted in the NA-NET. A message will be sent to each new address with instructions for the recipient to reply with a certain message. If the message is not

returned, exactly as specified, then the entry will not be accepted.

These are not new concepts. The successful `netlib` system has shown that a reasonable syntax for users to follow can be defined. Additionally, there exists on `navajo.stanford.edu` (and elsewhere) a `mailer` program that accepts specific requests from local users to modify aliases, mailing lists, *etc.* The new system will be patterned after these successes.

ACKNOWLEDGEMENTS. Gene Golub started the NA-NET and has continually supported its operation. Ray Tuminaro and Mark Kent have both acted as maintainers of the net. Ray also wrote the C program that handles the `na.(lastname)` facility and Mark wrote the Prolog material. Mark Crispin and Dan Kolkowitz have lent their systems expertise to keep the NA-NET functional. Cleve Moler acted as moderator of the NA-NET broadcast messages for the 1987–88 academic year.

This report was written while the author was enjoying the kind hospitality of Prof. Walter Gander of the Institut für Informatik, Eidgenössische Technische Hochschule, Zürich.

REFERENCES

[1] JACK J. DONGARRA AND ERIC GROSSE, *Distribution of Mathematical Software via Electronic Mail*, CACM, 30, 5, pp. 403–407, (987.

[2] A. GERASOULIS, M.D. GRIGORIADIS AND LIPING SUN, *A fast algorithm for Trummer's problem*, SISSC 8, 1, pp. s135–s138, 1987.

[3] JONATHEN B. POSTEL, *Simple Mail Transfer Protocol*, RFC 821, Information Sciences Institute, USC, 1982.

[4] JONATHEN B. POSTEL AND ZAW-SING SU, *The Domain Naming Convention for Internet User Applications*, RFC 819, 1982.

[5] JOHN S. QUARTERMAN AND JOSIAH C. HOSKINS, *Notable Computer Networks*, CACM, 29, 10, pp. 932–971, 1986.