

AxMail-FAX

I – Introduction:

Since the early goings of packet radio, hams were exchanging electronic messages through the PBBS system. One could send a bulletin that could be read by a group defined by which group that bulletin was sent to using the SB command, or they could send a message directly to another ham using the SP command. During the development of such systems, a group of coders drafted specific standards at which this system was to operate under. These guidelines were then adopted by TAPR, the same entity that developed the type 2 TNC which most hams use to this day, or a replica of such.

As the internet came on the scene to the general public for usage, it's services were a “pay for” based service and had such information based tools such as Gopher, finger, and the web as 3 for example. File sharing was made via the FTP (file transfer) protocol and private messaging through electronic mail services using the SMTP (simple mail transfer) protocol. These and a bunch more services became attractive to the more computer savvy user and before you know it softwares were designed to be a lot easier to use as to attract more clients for the local ISP. Unfortunately the costs to use such a service back then was so expensive most hams didn't want to bother as their money was more invested in RF gear than wired services.

Fortunately for hams, an entity calling itself the “AMateur Packet Radio” organization prior to stricter rules and costs on blocks of IP addresses were able to secure a full class A (/8) IPv4 subnet of addresses beginning with 44. and thus the ampr.org or better known as the amprnet was born. Around this same time frame Phil Karn KA9Q was able to figure out that IP could be encapsulated underneath our existing ax25 protocol after answering a challenge from a co-worker on how to get simple DOS computers to speak IP with their VAX system, and thus NOS was born for the amateur community to use. SMTP based electronic mail was being passed via ham radio frequencies without the need for wired services however certain people were able to tie NOS into an ISP account which served as global gateways for the various NOS services.

With NOS sites sprouting like Dandelion flowers in the spring, hams found that by using the amprnet on a NOS station they could send and receive SMTP mail with other hams and friends around the globe at no fees to them. There couldn't be any fees assessed because hams were using their 1200 baud TNCs to connect into their desired NOS system. By the mid 1990s, if you were a ham on packet, most likely you had an @*.ampr.org email address which the globe recognized as a valid SMTP based email mailbox. Doing email on ham radio isn't anything new since NOS systems have been doing it for a few decades, yet there's a stigma formed from a site in Newington, Connecticut that makes it seem as if it's something new from a different software based system. Remember this lack of integrity when it comes renewal time.

Since NOS, there's been a few different SMTP and mail systems to hit the airwaves. The most powerful yet least recognized system is axMail-FAX. It's the most feature packed SMTP based mail system out there for amateur radio, never asks/begs for money/donations, and is only not recognized by a certain Newington, Connecticut based organization because it does not donate 6 figure or higher dollar amounts to said organization as was told to my face at a meeting in October, 2017 by certain key “political” figures.

II – How it came to be:

After coming out with my linux-based URONode front-end, one sysop who decided to try it out pressed me to compliment it with some sort of an SMTP based mail system as LinFBB (which I supply code for on occasion) was a perfect plugin for PBBS type mail, SMTP mail was lacking. Being hesitant, I tried to nicely push back thinking “somewhere somehow someone must have such

a project already completed, no need for me to re-invent the wheel”. Finally after days of searching I came across an incompleated project simply called axMail which had already went through 2 primary coders. The first of which was Heikki Hannikainen (OH7LZB) in 1996. He had a basic mailbox system that could allow existing users on a linux server the ability to read and send SMTP based mail. It lacked many needed features such as allowing new users via RF to create their own mail-only accounts, and other mailbox handling instructions. Next came Marius Petrescu (YO2LOJ) who's the current author of many other ham projects today such as ampr-ripd, and an amprnet injector for Microtik routers to name a couple. He tightened up some of the internal mailbox routines in 1998 but still couldn't figure out a few needed features. It had appeared to have been orphaned so I took a look at the code. By 2005 I was able to get a very basic NOS type SMTP system working which included user account creations. Users, existing and new, were able to send/receive SMTP based mail just like they did with NOS. I put out word about it's initial release and to my surprise I received a very complimentary email from YO2LOJ on the user accounting that I added. Satisfied with the approval from Marius, I considered axMail reborn.

Later on I was able to figure out how to take input from an end user and turn it into a fax-transmission system complete with automatic cover page creation with an SF (send fax) command that quite easily walks the sender through very layman's prompts on how to send a fax. This also includes receipt of fax job accepted into the outbound fax queue and afterwards a confirmation of the completion or failure of the fax, and thus axMail-FAX became the project's new name.

III – Feature packed

The difference with axMail-FAX and other SMTP mail systems out there for packet is that it's mega feature packed vs the rest. They really begin to shine the deeper into the various command tree you maneuver. For starters, your email address is displayed within the prompt so you can never forget your email address.. If you accidentally delete/kill a message you can easily recover it before you log out – no other system allows for this over RF. You can send carbon copies and also blind carbon copies (cc: and bcc: mail) in which no other email system for packet has. You can set a custom one-line signature for all your messages so you wouldn't have to type “-73 de C4LL” at the end of each message, and a LOT more. The basic Commands:

? - brings up the command listing.

Autofwd – sets or unsets a mailbox you wish your incoming mail to get forwarded to and not saved on your axMail host.

Bye, Exit, Quit – Save your mailbox as you have it and exit the program.

Cancel – Quit the program but discard any mailbox changes.

Delete, Kill – Followed by a number or series of numbers to kill/delete mail(s).

Help – Followed with a full command name gives detailed help on a specific command.

Info – Information about axMail-FAX.

List – Verbosely lists your mail messages.

Name – Sets your name which is appended to your email address when sending mail.

Read – Followed with a number reads a specific mail message.

Send – Sends an email message. This walks the end user through the whole process asking them if they wish to send cc/bcc mail, flag it priority mail, request a read receipt, and then the subject line. Entering the body is next followed first by a confirmation prompt. If the user accidentally flags the body as complete but wishes to continue, C for continue message will allow them to continue adding to the email message. Another feature ALL OTHERS LACK. This confirmation prompt will continue to loop until the user selects Yes upon which they will be asked if they desire a server delivery receipt. This receipt will email you a confirmation that your message was in fact delivered to the recipient's mailbox. Remember, read receipts are generated at the option of the reader and can easily be ignored thus leaving the sender in a state

of unknown as to whether or not the mail was received and/or read. This receipt comes from the remote server itself so if the receiver fails to acknowledge reading your mail, they can **NOT** say it was never received. Here's a key example of how this may come into play:

User A uses ARES selected email system to mail FEMA in a disaster requesting 3 days worth of food and bedding for 50 affected citizens, and requests a read receipt.

User B uses axMail-FAX and emails FEMA requesting 5 days worth of food and bedding for 25 affected citizens and requests a server receipt along with a read receipt.

Outcome: FEMA fails to respond, communications personnel claims no mails were received from EITHER A or B tries to fault both for failing to relay emergency communications. User A is stuck! He has no proof communications were sent, while User B has the server receipt **from FEMA's mail server** as proof the FEMA personnel failed to do their job and User B is saved from any punishments that may have come his way that User A will receive. This one is HUGE.

SFax – Send a facsimile communication. Very much like with Send SF alone will walk the end user in sending a fax and will automagically create a cover page.

Sig – Set or delete your signature. If not set, you'll be reminded of such when you send a mail.

Spers – This command is more geared for HF users. It shortens the prompting of the standard “Send” command and auto flags the mail priority since it's coming via amateur radio.

Sreply – Send a reply to the current message number listed in your prompt.

SStatus – Brief status of your mailbox.

Unkill – Followed by a number, restores a previously killed message proper to your logging off.

Verbose – Like with the Read command, this command shows you the message in full, meaning all the message headers, etc that are contained within the mail message.

When you first login you'll be asked a series of basic questions. The version number will be displayed prior to asking you anything. Please insure that the version is 2.9 or above. If it isn't ask your sysop to upgrade as soon as possible. You'll be asked your name and instructed to contact your sysop with a desired password. I've taken every step possible so that not even a portion of your password is exchanged via plain text anywhere when you login!.. unlike other softwares that do.

Once on the system your prompt may look like this:

axMail-Fax v2.9

You have 7 messages (1 new).

n1uro@n1uro.ampr.org

Current message 0 of 7

=>

It has the program/version

Your mail count (and any that are new)

Your email address in full

Current message you read

and the command arrow =>

When I type L at the arrow I will get a list of my mail:

```
l
St Num From          Subject          Date          Size
U  1 postmaster@portal.ampr.or [AMPRNet] Encap file  Mon, 16 Apr 2018 49295
U  2 uro-jokes-owner@n1uro.amp Uro-jokes post from lu9d Sun, 15 Apr 2018 6705
U  3 root@asterisk.n1uro.ampr. Cron <root@asterisk> /va Mon, 16 Apr 2018 1144
U  4 root@asterisk.n1uro.ampr. Cron <root@asterisk> /va Mon, 16 Apr 2018 1144
U  5 root@asterisk.n1uro.ampr. Cron <root@asterisk> /va Mon, 16 Apr 2018 1144
U  6 root@n1uro.ampr.org (Cron Cron <root@n1uro> /usr/l Mon, 16 Apr 2018 7166
N  7 postmaster@portal.ampr.or [AMPRNet] Encap file  Mon, 16 Apr 2018 49280
n1uro@n1uro.ampr.org
Current message 0 of 7
=>
```

Status, message number, From, Subject, Date, and message size are listed.
U in status is for Unread, while N is a new message which has not been listed or read yet.

I chose to read message #3:

```
=> r 3
Message 3: (Unread)
From: root@asterisk.n1uro.ampr.org (Cron Daemon)
To: root@asterisk.n1uro.ampr.org
Subject: Cron <root@asterisk>
Date: Mon, 16 Apr 2018 00:00:02 -0400 (EDT)
--- end of message #3 ---
n1uro@n1uro.ampr.org
Current message 3 of 7
```

Now I wish to kill it and then list my mailbox to see it's killed:

```
=> k 3
Message 3 killed.
n1uro@n1uro.ampr.org
Current message 3 of 7
=> l
St Num From          Subject          Date          Size
U  1 postmaster@portal.ampr.or [AMPRNet] Encap file  Mon, 16 Apr 2018 49295
U  2 uro-jokes-owner@n1uro.amp Uro-jokes post from lu9d Sun, 15 Apr 2018 6705
>K 3 root@asterisk.n1uro.ampr. Cron <root@asterisk> /va Mon, 16 Apr 2018 1144
U  4 root@asterisk.n1uro.ampr. Cron <root@asterisk> /va Mon, 16 Apr 2018 1144
U  5 root@asterisk.n1uro.ampr. Cron <root@asterisk> /va Mon, 16 Apr 2018 1144
U  6 root@n1uro.ampr.org (Cron Cron <root@n1uro> /usr/l Mon, 16 Apr 2018 7166
N  7 postmaster@portal.ampr.or [AMPRNet] Encap file  Mon, 16 Apr 2018 49280
n1uro@n1uro.ampr.org
Current message 3 of 7
```

But I changed my mind as it may be a server report I want to save:

```
=> u 3
Message 3 unkilld.
n1uro@n1uro.ampr.org
Current message 3 of 7
```

Here I sent myself a mail and requested both receipts to be generated. Creating a read receipt on axMail-FAX is as simple as reading the mail:

N 8 Brian <n1uro@n1uro.ampr.o test Mon, 16 Apr 2018 609
N 9 MAILER-DAEMON@n1uro.ampr. Successful Mail Delivery Mon, 16 Apr 2018 2318
n1uro@n1uro.ampr.org
Current message 0 of 9

=> r 8

Message 8: (New)

From: Brian <n1uro@n1uro.ampr.org>
To: n1uro@n1uro.ampr.org
Subject: test
Date: Mon, 16 Apr 2018 00:33:54 -0400 (EDT)
test.

73 de Brian, N1URO - supporting packet radio since 1995.

sent via axMail-FAX by N1URO.

--- end of message #8 ---

A receipt was asked for, do we send one? (y/N):

Commands in Uppercase are hotkeys and/or defaults. If I chose to NOT send one all I would have to do is hit enter. N or n is not required, but here I want to be nice and send one:

y

Receipt going to: Brian <n1uro@n1uro.ampr.org>

Read receipt sent.

n1uro@n1uro.ampr.org

Current message 8 of 9

=>

Can't get any easier than that!

Now I want to read my server receipt:

r 9

Message 9: (New)

Date: Mon, 16 Apr 2018 00:33:54 -0400 (EDT)
From: MAILER-DAEMON@n1uro.ampr.org (Mail Delivery System)
Subject: Successful Mail Delivery Report
To: n1uro@n1uro.ampr.org
This is a MIME-encapsulated message.

--69E882111F.1523853234/n1uro.ampr.org

Content-Description: Notification

Content-Type: text/plain; charset=us-ascii

This is the mail system at host n1uro.ampr.org.

Your message was successfully delivered to the destination(s) listed below. If the message was delivered to mailbox you will receive no further notifications. Otherwise you may still receive notifications of mail delivery errors from other systems.

The mail system

<n1uro@n1uro.ampr.org> (expanded from <n1uro>): delivery via local: delivered
to command: /usr/bin/procmail -a "\$EXTENSION" -p /etc/procmailrc

--69E882111F.1523853234/n1uro.ampr.org
Content-Description: Delivery report
Content-Type: message/delivery-status

Reporting-MTA: dns; n1uro.ampr.org
X-Postfix-Queue-ID: 69E882111F
X-Postfix-Sender: rfc822; n1uro@n1uro.ampr.org
Arrival-Date: Mon, 16 Apr 2018 00:33:54 -0400 (EDT)

Final-Recipient: rfc822; n1uro@n1uro.ampr.org
Original-Recipient: rfc822; n1uro
Action: delivered
Status: 2.0.0
Diagnostic-Code: X-Postfix; delivery via local: delivered to command:
/usr/bin/procmail -a "\$EXTENSION" -p /etc/procmailrc

--69E882111F.1523853234/n1uro.ampr.org
Content-Description: Message Headers
Content-Type: text/rfc822-headers

Return-Path: <n1uro@n1uro.ampr.org>
Received: by n1uro.ampr.org (Postfix, from userid 1000)
id 69E882111F; Mon, 16 Apr 2018 00:33:54 -0400 (EDT)
From: Brian <n1uro@n1uro.ampr.org>
To: n1uro@n1uro.ampr.org
X-Priority: 3 (Normal)
Disposition-Notification-To: Brian <n1uro@n1uro.ampr.org>
Subject: test
Message-Id: <20180416043354.69E882111F@n1uro.ampr.org>
Date: Mon, 16 Apr 2018 00:33:54 -0400 (EDT)

--69E882111F.1523853234/n1uro.ampr.org--

--- end of message #9 ---

n1uro@n1uro.ampr.org

Current message 9 of 9

=>

Now I can't say I didn't get the mail as there's physical proof otherwise to argue such a statement.

For grins I want to read my read receipt:

r 10

Message 10: (New)

From: Brian <n1uro@n1uro.ampr.org>

To: Brian <n1uro@n1uro.ampr.org>

Subject: axMail-FAX read receipt for "test"

Date: Mon, 16 Apr 2018 00:35:21 -0400 (EDT)

Your mail to Brian <n1uro@n1uro.ampr.org> about "test"

written on Mon, 16 Apr 2018 00:33:54 -0400 (EDT) has been read.

--- end of message #10 ---
n1uro@n1uro.ampr.org
Current message 10 of 10
=>

Very **VERY** simple to use and without any special needs just a dumb terminal and you're off and running.

IV – Other “fax”:

One big question I get is “Can it handle file attachments?” The answer is quite simple: Yes. File attachments, as much as they appear to the eye as a separate entity, are often an encode of a file which is placed within the body of a mail message. The client then determines where and how it's located and then extracted upon receipt. This can be accomplished by using an external encode/decode utility such as UUCP or MIME. Say I have a file with some sort of wire labeling I want to send. I'd run uuencode to parse it and then copy/paste the file in email:

```
n1uro@n1uro:~$ cat foo
1-red
2-brown
3-white
4-black
5-green
```

or...

```
1-white
2-brown
3-red
4-green
5-black
n1uro@n1uro:~$ uuencode foo foo.txt
begin 644 foo.txt
M,2UR960*,BUB<FJW;@HS+7=H:71E"C0M8FQA8VL*-2UG<F5E;@H*;W(N+BX*
G"C$M=VAI=&4*,BUB<FJW;@HS+7)E9`HT+6=R965N"C4M8FQA8VL*
,
end
```

Enter message text (end with "/ex" or "." on a line by itself):

```
uuencode this buddy
begin 644 foo.txt
M,2UR960*,BUB<FJW;@HS+7=H:71E"C0M8FQA8VL*-2UG<F5E;@H*;W(N+BX*
G"C$M=VAI=&4*,BUB<FJW;@HS+7)E9`HT+6=R965N"C4M8FQA8VL*
,
end
```

Deliver (Y/n/c/?):

Request a delivery receipt? (y/N):

Message sent.

Now I received it and I paste it into a file called foo.uu:

```
n1uro@n1uro:~$ nano foo.txt
```

Now I decode it:

```
n1uro@n1uro:~$ uudecode -o foo.uu foo.txt
```

and I read it as it was originally:

```
n1uro@n1uro:~$ cat foo.uu
```

```
1-red  
2-brown  
3-white  
4-black  
5-green
```

or...

```
1-white  
2-brown  
3-red  
4-green  
5-black
```

Typically your email client software will do this for you when you 'attach a file'. This however separates you from an appliance operator. It's always best to know how something works vs being held captive to the unknown.

Another question I get asked often is “what software if I chose to use a GUI client does this take?.. is there an axMail-express or similar?” There's what I consider 2 levels of axMail-FAX:

```
1 – Basic  
2 – Suite
```

The basic is more designed for the simple 1200 baud or lower node and while it may support IP to the end user, it's not the ideal scenario so just basic ax25 connections are mainly used.

The suite typically contains a pop-before-smtp utility that grants temporary SMTP access for dynamic Ips and can be used strictly on amprnet OR any other internet device. Also a modified pop3 daemon is used which can be had from my FTP server. Because of this, NO SPECIAL SOFTWARE is required. You can use Outlook, Thunderbird, whatever email client software you desire to use. On my mobile devices I like to use Aqua-mail and for a client I use Evolution which is a linux client by Novell that acts and feels like Outlook. ANY client not designed for a specific mail system is fine.

“What about webmail services?” This is one of the key reasons for stricter password security. It's been known that other systems which require the end user to share their password IN PLAIN TEXT is so dangerous. Even those who challenge passwords are easily cracked and can be tested via webmail – and to the wrong user (listening is allowed by EVERYONE not just hams) they can collect usernames and passwords from RF and sell them to spam outlets. Since axMail-FAX 2.9, there is never a password to be sent via RF. If the user is pirating a callsign, that's an FCC matter as it is anyway not ours and sending mail would then lead a trail to the pirate who'd be looking for replies. Sometimes we over-process things to the point where security by obscurity makes the best sense. Using Lets Encrypt, apache2, Squirrel-mail, and Dovecot in ssl-imap mode, full SSL/TLS webmail is very doable and also can be done via a mobile device. Obviously not to be run at a 1200 baud nodesite but for those on HamWan/802.11 or even 9600 baud systems (the typical speed of a router's console) this is quite feasible and is actually in use on EastNet at various locations.

“Why you over them?” It's not a matter of who/what/where/when/how it's a matter of what's proven to last. During Hurricane Irma, it was reported to me that the center part of Florida and out to the coasts were without internet. The excessive winds also ate HF antennas like it was having a midnight snack – basically rendering EOC/ARES mail in some parts totally useless. Little does everyone know, the amprnet and axMail-FAX exchanges between Melbourne and Vero Beach was 100% functional 100% of the time. This unfortunately is the result of when politics takes president over functionality – innocent lives get harmed.

The FCC asked ALL communication providers amateur and commercial to provide them with a listing of infrastructures during all major hurricane events in 2017 and the biggest failures were HF amateur communications and wired internet services. High winds were way too much even for the best of HF antenna structures, along with the dangling wires of telcom/cable providers. Those that had the lesser amounts of damage were the cellular networks and short-hop RF services which VHF/UHF/802.11 provide. The proof positive of this is the fact W4OT<>W4MLB continued to be able to exchange mail the entire time. The real question here is why is it buried under the rug? Two things you can't accept in times of emergency: failed communications transport and suffering a blue screen of death.

“How do we get a demo or training?” Simple: ask! The U.S. Air Force and U.S. Army MARS programs are migrating away from everything else to axMail-FAX, don't you think the EOCs and ARES folks should as well? Go to <https://eastnetpacket.org> and write to the webmaster. I'm sure arrangements can be made somehow.

“We're out of IPv4 now, what about IPv6 can you handle that?” Absolutely! <https://uronode.n1uro.com> has a listing of URONode sites that are already IPv6, to which axMail also is IPv6. You'll find these sites to be dual stacked.

“This guy says he can support 100 users at once, can you?” First, I want him to get 100 radios with 100 TNCs and make 100 concurrent connects to the same single node at 1200 baud. If he told you also that he can spit \$10.00 GOLD coins out of his ears would you believe that too? Both make equal sense to each other. In reality yes we can support more than 100 concurrent users to axMail-FAX... and the more sites that set up with it, the more users and improved IP routing we'll have on amateur radio.

V – Closing:

I thank you for reading this and for your interest in the axMail-FAX system. Most importantly for making use of our packet resources and keeping it alive and well. Use axMail-FAX and tell your ham friends about the ease of use, the volume of features (especially those that help you C.Y.A.) Both axMail-FAX and URONode are available from SourceForge, and from <ftp://n1uro.ampr.org> and are both given support by TAPR.