Saam-Mail de WH6GGO

User Guide

v1.4

Author: Lawrence Byng.

Date: November 5th, 2022.

Document Version 1.4

Table of Contents

1. What is Saam-Mail?

Saam-Mail was designed primarily for sending fully verified, error corrected forms over ham radio peer to peer or peer to group. Saam-Mail has multiple optimizations and supports many techniques that make the communications resilient even in the most challenging conditions.

Saam-Mail is designed to run as an add-on program to either JS8Call or fldigi. The program is very flexible and has been tested with twenty eight fldigi modes and four JS8Call modes varying from very narrow bandwidth JS8 slow mode at 25Hz to high bandwidth 2600Hz wide PSK modes. Saam-Mail is also open source and cross platform allowing it to be used on many computing platforms. It has been tested with the raspberry pi 4b and the Beelink mini-computing platforms, both of which use very little electrical power and are ideal for off-grid and/or mobile setups.

i. Developers in the Ham Radio Community – Thank You!

Saam-Mail would not exist without fldigi, JS8Call and all the hard work and dedication of the many developers in the amateur radio community. Thank you!

2. Downloading Saam-Mail

The application repository is at https://github.com/gh42lb/Saam-Mail. This repository contains all the necessary files. The most straight forward way to download these is to click on the green ‘Code’ button then to select the ‘Download ZIP’ file option. This will download the latest Saam-Mail files to your computer. If you prefer you can also download the files individually. The above repository is the only approved repository for the Saam-Mail application.

There are three options for download which depend on the chosen computing platform as follows:

1. Download the windows binary file called Saam-Mail.exe from the github repository (link above). The windows binary can be run on the windows platform and has been tested on Windows 10.
2. Download the raspberry pi binary file called ‘Saam-Mail’ from the github repository (link above). The raspberry pi binary has been tested extensively on the raspberry pi 4b using the standard raspbian OS.
3. Download all of the .py files from the github repository (link above). You will also need to install the following modules: PySimpleGUI, sys, threading, json, random, getopt, datetime, socket, time, select, calendar, gps, crc. This can be done using pip3 command for any missing modules e.g. pip3 install pysimplegui. Please note saam_mail is available for python 3 only.

If you have not already done so, you will also need to download the templates. These files can easily be identified as they end with .tpl

There are currently two template files which should be downloaded as follows:

1. ICS_Form_Templates.tpl
2. standard_templates.tpl

I. OK. I’ve Downloaded Saam-Mail. What Are the Next Steps?

Congratulations!

There are a couple of configuration steps required after which Saam-Mail can be started and you can be on the radio transferring forms to a receiving station.

3. Fldigi and JS8Call Configuration

Saam-Mail is run in conjunction with either Fldigi or JS8Call.

I. Saam-Mail Fldigi Configuration

If your station is already setup for digital radio, Saam-Mail should work with fldigi out-of-the-box. Saam-Mail uses the xmlrpc fldigi external API to send and receive messages.

To specify the operating mode you can use the --opmode= command line option (i.e. --opmode=fldigi), however, fldigi mode is the default so it is currently not necessary to specify this if you are running Saam-Mail with fldigi.

To change to xmlrpc port used to communicate with fldigi, use the --fldigi= command line option as follows:

--fldigi=127.0.0.1:7365

the above option will open xmlrpc port 7365 to the fldigi instance running on the local computer 127.0.0.1

To connect with an fldigi instance on another computer over a local network, this can be done as follows:

--fldigi=192.168.53.1:7362

The above option will open xmlrpc port 7362 (primary xmlrpc port in fldigi) to the fldigi instance running on the computer with the ip 192.168.53.1

Currently Saam-Mail supports one instance of fldigi.

II. Saam-Mail JS8Call Configuration

To use Saam-Mail in conjunction with JS8Call there are first a few configuration steps needed in JS8Call as follows:

1. On the ‘Mode’ menu, the ‘Auto Reply’ option should be checked. This option is necessary for the proper functioning of text transfers between Saam-Mail and JS8Call.
2. On the ‘File/Settings/Reporting’ tab find the API section and set as follows:
    - TCP Server Hostname: 127.0.0.1 Enable TCP Server API – Checked
    - TCP Server Port: 2442 Accept TCP Requests - Checked
    - TCP Max Connections: 1 (if using Saam-Mail only) or 2 (if using Saam-Mail and JS8Net concurrently)

When you are ready to send a communication with JS8Call you will need to set the appropriate mode in JS8Call as required (Slow, Normal, Fast, Turbo) and make sure the TX button at the top right of the JS8Call screen is enabled.

To change to JSON port used to communicate with JS8Call, use the --js8call= command line option as follows:

--js8call=127.0.0.1:2447

the above option will open JSON port 2447 to the JS8Call instance running on the local computer 127.0.0.1

To connect with a JS8Call instance on another computer over a local network, this can be done as follows:

--js8call=192.168.53.1:2442

The above option will open JSON port 2442 (default JSON port in JS8Call) to the JS8Call instance running on the computer with the ip 192.168.53.1

Currently Saam-Mail supports one instance of JS8Call.

4. Starting Saam-Mail

The application currently supports one instance of Fldigi or JS8Call operating one radio. Testing has shown that it is possible to run fldigi and js8call at the same time side by side on the raspberry pi but this is not currently a supported mode of operation.

I. Saam-Mail using Fldigi

Running the application in fldigi mode is done with the use of the opmode command line parameter:

saam_mail --opmode=fldigi

Upon start-up the main window will display controls that are specific to the fldigi mode of operation.

II. Saam-Mail using JS8Call

Running the application in JS8call mode is done with the use of the opmode command line parameter:

saam_mail --opmode=js8call

Upon start-up the main window will display controls that are specific to the js8call mode of operation.

III. Saam-Mail Configuration

Once Saam-Mail is up and running, go to the ‘My Info’ tab and fill out the information as it relates to your station.

The call sign and group name are required fields and must be filled out in order to use the application. The remaining fields are not as critical however if you wish to use the advanced features of the Dynamic Content Macros these should also be filled out.

5. Sending Your First Form with Saam-Mail

I. Choosing the Form to Send

Click on the ‘Compose Msg’ tab. In the ‘Category’ table select ‘GENERAL’ and to the right of that select the ‘BULLETIN’ form.

II. Selecting the Recipient Stations

To the left of the ‘Category’ table there is a table listing the Call-signs heard. One or more of these can be selected and this will automatically complete the ‘To:’ field with a semicolon delimited list of recipient station call signs.

Alternatively, The recipient station call sign can be entered manually in the ‘To:’ field. If there are multiple recipient station make sure the call signs are separated with the ‘;’ semicolon character.

Now click on the ‘Compose’ Message’ button. This will open a new pop-up window showing the blank form ready to be filled in.

III. Filling In the Form

i. Form Top Section

The top section of the form shows the following information:

1. Form: This is the name of the form selected in the previous step.
2. MSGID: This is the automatically generated ID that will be used to identify the form. The ID is generated from the sending station call sign and the UTC time to the nearest second that the ‘Compose Msg’ button was pressed.
3. Priority: This field sets the priority of the message.
4. To: This field contains the list of call signs that were set in the previous step. If you wish to make any changes this can be done by editing this field.
5. Subject: This field is the subject of the message communication similar to the subject field on an email.

ii. Main Form Section

The main form section is the actual bulletin form. Go ahead and fill out the fields as appropriate. When finished, click on the ‘Post To Out-box’ button at the top of the screen. If you wish to cancel the form, go ahead and click the ‘Cancel’ button.

IV. Sending the Form

Towards the top left of the main application there is a field labelled ‘Connect To:’. If this field is empty it will be flashing red and green. Fill out the field by specifying the call sign of one of the stations that you will be sending the form to. Each of the recipient stations will also need to fill out this field but they should specify your station call sign as they will be connecting to your station to receive the form.

Now go ahead and click on the ‘Out Box’ tab. In the main table you should see the form that you posted to the out-box listed. If you select the message by clicking on it, you will also see a text representation of the form appear in the preview window that is toward the lower part of the page.

There are several buttons on the ‘Out Box’ page which function as follows:

1. View: Clicking on the view button will open a non-editable full colour representation of the bulletin form message that you selected above. You can use this to review the message prior to sending. When done, click on the close button to take you back to the ‘Out Box’ tab.

2. Copy: Clicking on this button will copy the message to the clipboard. This can be used to export the form message to another application such as an email application or winlink or pat winlink or any other application that accepts cut and paste text. The copied message has two parts as follows:
    - The first section of the copied message is the text representation in human readable form that can be used to review the information.
    - The second part of the copied message is at the bottom of the text where you will see ‘SAAM-MAIL-EXPORT=’ followed by the actual communication itself. This export string can be used either separately or in conjunction with the first part described above to export to another application. The form can be imported into Saam-Mail by clicking on the ‘Clipboard Import’ button towards the upper part of the main screen. The import button will import the message by using only the SAAM-MAIL-EXPORT= string and place the message in either the inbox if it was addressed to the station or in the relay-box if not.
3. Edit: Clicking the edit button will open an editable version of the form with a new message ID. If you wish to make changes you can do so on this form and when done post it to the out-box. Please note that both the original message and the revised message are both contained in the out-box. If you wish to delete one of these you can select it and click the delete button.
4. Delete: clicking on the delete button will deleted the selected message from the out-box.
5. Delete All: Clicking ‘Delete All’ will delete all of the messages in the out-box.
6. Ready?: This will send out a transmission to the recipient station detailed in the ‘Connect To’ field to ask if they are ready to receive the message. This is an optional step.
7. Send: Clicking the send button will send the message.

Go ahead and click the send button. If everything is setup correctly, you will see the text sent to fldigi or JS8Call and the process of transferring the form starts. You will also see ‘In Session’ start to flash at the top right of the main form. This indicates that a message transfer session is in process.

V. Advanced Send Message Options

Immediately below the row of buttons on the ‘Out Box’ page you will see a row of options that can be used to fine tune the communications as follows:

1. Pre-Message: This checkbox is used to enable the pre-message option. Currently this is used to send out a stub message for any messages that are waiting to be sent in your out-box. This part of the message is post only and is verified by validating the checksum. It is very useful to communicate to the group if there are any messages pending and waiting to be sent out. If there are multiple pending messages, a single message is chosen at random. Higher priority messages are given a higher probability of selection by the selection algorithm.

2. Repeat Message: This option appears only in the fldigi operation mode. It is used to send one or more additional copies of the message and to interlace these duplicate fragments with the original message. This communication technique is described further in the communication techniques section.

3. Repeat Fragments. This option appears only in the js8call operation mode. It is used to repeat up to the first 3 fragments of the message up to three times. These fragments typically contain the message id and receive list.

4. Include Template: This option will send a copy of the form template used to create the message alongside the content of the form message itself. This can be used by the receiving station to view the message if the receiving station does not have a copy of the form template.

5. Fragment Size: This specifies the number of charters in each message fragment. The default is 30 characters. This can be adjusted depending on band/propagation conditions and also depending on the underlying fldigi or JS8call modulation used. In general, if conditions and modulation mode result in minor errors and a small number of re-transmits, then larger fragment sizes can be used. Conversely if conditions and modulation mode result in a large number of errors and re-transmits, then a smaller fragment size would be more appropriate.

Also above the tabs in the main area of the form you will see additional options as follows:

1. Fldigi Mode: When running in the fldigi opmode, you will see a list of available fldigi modes listed in this combo-box drop-down. The modes are listed from 1 to 28 in the order of speed with mode 1 being the fastest and mode 28 being the slowness. These modes specify the underlying fldigi modulation mode that will be used for the message transfer. The modes available can be filtered by selecting different option on the ‘Span’ combo-box drop-down.

2. Channel: This drop-down lists the channels that are available. There are 16 channels that correspond with 16 different offsets in the passband. Each Channel is 125Hz wide. 125Hz is the bandwidth of the narrowest fldigi mode available so if the selected modulation mode is 125Hz wide then then all 16 channels are available for use. If the modulation mode uses a wider bandwidth such as 500Hz then this would effectively mean that there are 4 Channels available: i.e. 1, 5, 9 and 13. The other aspect of the channel is that it ensures both the transmit and receive stations stay on the precise offset which is required for reliable signal decodes in fldigi. Channel is also present when running in the js8call mode and performs a similar role. The Channels and offset values are identical in js8call mode and fldigi mode.

3. Send To: This drop-down gives a list of radios at your QTH that the message can be sent to. Currently there should be only one rig (radio) listed in this drop-down which will be the one configured for use with Saam-Mail.

4. Span: This drop-down gives options that can be used to filter the fldigi modes available by bandwidth. For example ‘HF – 500’ will filter out any modes that have a bandwidth higher than 500 Hz on the Fldigi Mode drop-down.

6. Advanced Features

I. Dynamic Content Macros

Dynamic Content Macros are used to pre-fill parts of the form message during the compose message phase of the process. The Dynamic content macros relate to the individual fields on the ‘My Info’ tab as well as providing hooks to the current date and time in local time as well as UTC or Zulu time. The current list of available dynamic content macros is as follows:

1. %CALLSIGN%
2. %OPERATORNAME%
3. %OPERATORTITLE%
4. %INCIDENTNAME%
5. %DATE%
6. %TIME%
7. %DATETIME%
8. %LOCALTIME%
9. %ZULUTIME%
10. %UTCTIME%
11. %UTCTIME%
12. %GROUPNAME%
13. %GPSLAT%
14. %GPSLONG%
15. %GRIDSQUARE%
16. %QTHLOCATION%

II. Winlink Integration

On the ‘Winlink’ tab there are two tables, one for the winlink inbox and one for the winlink out-box. In order to setup the integration with winlink you will need to enter the location of the winlink inbox folder in the ‘Winlink Inbox:’ entry field and also the location of the winlink out-box in the ‘Winlink Out-box:’ field. Once this information has been entered, click on the ‘list’ button and this will show all the winlink inbox and out-box messages.

Now select a message from either the winlink inbox or the winlink out-box then click edit. This will open a window that contains the information from winlink. You can now fill out the rest of the fields and click post to out-box and it will be posted to the Saam-Mail out-box ready to send

Please note: The Winlink integration tab is able to read both winlink messages on a windows platform as well as pat winlink messages on a non-windows platform.

III. Import & Export - Clipboard Copy & Paste

Towards the center of the main screen, you will see a button labelled ‘Clipboard Import’. This button enables importing messages directly from the clipboard into Saam-Mail. The message to be imported must contain a SAAM-MAIL-EXPORT= tag along with the form information immediately following the tag for the import to succeed.

The inbox, out-box and relay-box have a ‘Copy’ button. This ‘Copy’ button can be used to export the selected message and to copy it to the clipboard. The exported message will have two parts as follows:

1. The first part of the exported message is a text representation of the information contained on the form. This is included by default but is optional. One reason that you may wish to include a text version of the message relates to whether the receiving station has Saam-Mail or not. Another would be that the message is being exported to be sent over the internet or other fast connection and bandwidth is not an issue and the text is included for message preview purposes.

2. The second part of the exported message is the export string and this begins with the text ‘SAAM-MAIL-EXPORT=’. This export string is critical if the message is to be successfully imported into Saam-Mail at the receiving station.

Important Note: When viewing the text version of an exported message, you must use a mono-spaced font such as Times New Roman or Helvetica and also turn word-wrap off. This will enable the text portion to be viewed correctly.

IV. Customizing Colors

The ‘Colors’ tab enables many of the default color settings in Saam-Mail to be changed. The process of changing a defult color involves selecting the new color in the relevant field then clicking the ‘Update’ button. The different options for changing default colors in Saam-Mail are as follows:

1. Notebook Tabs. Each of the notebook tabs can be set to a different color. This could be useful to give a visual indication for the currently selected tab. Please note that after clicking the ‘Update’ button, the application must be closed then re-opened for the tab colors to reflect the change.

2. Main Heading Background and Text. These two selections allow the default color of the Main Heading used on the forms to be changed. The setting affects only your station as does not change any of the information actually sent with the form.

3. Sub Heading Background and Text. These two selections allow the default color of the Sub Heading used on the forms to be changed. The setting affects only your station as does not change any of the information actually sent with the form.

4. Numbered Section Background and Text. These two selections allow the default color of the Numbered Section used on the forms to be changed. The setting affects only your station as does not change any of the information actually sent with the form.

5. Table Header Background and Text. These two selections allow the default color of the Table Header used on the forms to be changed. The setting affects only your station as does not change any of the information actually sent with the form.

V. Customized Table Control

Saam-Mail utilizes many standard ‘Graphical User Interface’ (GUI) controls however in order to provide the functionality for sending forms it was necessary to create a customized table control so that it could be edited. The functionality of the custom control is still being refined. The main aspects of the custom editable table are as follows:

1. Click on table cell. Clicking on one of the cells in the table allows the field to be edited. When the edit is complete, go ahead and click the return or enter key to complete the update of the field. Make sure that focus is in the editable field when you click the enter key.

2. ‘Tab’ key. First click on the table cell to begin editing. Once you have updated the field, you can click the ‘Tab’ key to update the value and move to the next field. This is very useful if there is a large amount of data to be entered into the table.

3. ‘Add Row’ button. When you click on a cell to start editing, a button will appear that allows you to add additional blank rows onto the end of the table as required.

7. Critical Message Communication Techniques

These techniques are used for the transfer of fully verified error corrected messages as would be the case for critical messages.

I. Single Station to Single Station (Peer to Peer)

Sending a verified, error corrected message from one station to another using the peer to peer approach is done simply by specifying only a single receive station in the ‘send to’ list when the message is composed.

Peer to peer communications of forms is optimized for shorter message length, faster transfer speeds and increased resilience to adverse band conditions.

II. Single Station to Multiple Stations (Peer to Group)

This can be achieved by entering a list of recipient stations in the ‘To:’ field when composing a message. All specified receiving stations will need to be on frequency and listening for the incoming message.

During the send process, the station sending the form will verify with each receiving station in turn that they have received the message in full without error and if not will re-transmit any missing parts. At the end of this process each receiving station should have a fully verified, error corrected copy of the original message.

Please note: during the re-transmit phase, the sending station will make multiple attempts to resend the missing parts up-to a preset number of retries before moving to the next station on the list. If the message cannot be transferred within the specified number of retries, there are additional options available to complete the message transfer process as follows:

1. Following completion of the transfer session, any station that has not received a fully verified, error corrected copy of the message can send out a pull request to the sending station to pull the message.
2. The sending station can make additional attempts at sending the message to the stations that did not verify during the original session using the message push approach.
3. The receiving station can request missing fragments from a relay station.

Please note, if there are a large number or receiving stations specified (i.e. more than 10 or so) on the receive list, then this technique can still be used however another technique may be more appropriate such as that discussed in the next section.

i. Reducing Re-Transmits: Duplicate Fragments

This can be done by sending additional copies of the full message (full set of fragments) or sending additional copies of specified fragments such is the first 3 which contain message ID and receive list.

1. Sending Multiple Copies of the full message. This option is currently available only with the fldigi opmode. The additional message copies are interlaced with each other to increase resilience to any impacts to the send stream. Sending multiple copies of the message may be a good strategy if there are a large number of receiving stations as the extra time to send an extra copy could well be less than the time required to go station by station and re-transmit when only a single initial copy is sent out.
2. Sending additional copies of first 3 fragments. This option is currently available only with the JS8Call/JS8 operating mode. The purpose for doing this is to ensure the first three fragments that contain the most critical information are sent several times. Once received, these initial fragments will provide key information such as message ID and recipient list that the receiving station can then use to determine if a full copy is required and if so to have the message ID that can be used to request the full copy.

ii. Pulling Missing Fragments

This technique can be used if the message transfer was not successful during the original transfer session. There are two approaches to pulling any missing fragments as follows:

1. Request missing parts from sending station. The station that sent out the original message is known to have a full copy of that message so the process is quite simple. If you have a message in the inbox or relay box that shows ‘partial’ then you can send a request to the sending station by selecting the message in the table in the inbox or relay box and then clicking the ‘Request Msg’ button. This will pull the message from the original sending station and attempt to provide a fully verified, error corrected copy of the message using a similar process as before.

2. Request missing parts from a relay station station. In order to use this approach you will first need to send out a query to determine if there are any other stations on frequency that have a copy of the full message or specified missing fragments. This can be done by selecting the message in the inbox or relay box then click on the ‘Query Msg’ button. This will send a query to the group of stations with the message id. If a relay station has a copy and replies back to confirm, then you can initiate the pull request in the same manner as step 1 above, the only difference is that you will change the ‘Connect To:’ field to be the call sign of the relay station. Then click ‘Request Msg’ button and the transfer of missing fragments will begin.

iii. Send then Pull

This technique could be used for a large number of stations. The technique involves sending the message to the group then having members of the group request any missing parts using the pull technique. For this process to be effective it would make sense to send multiple copies of the message in the initial send to reduce the number of missing fragments at the receiving stations.

iv. Pending Messages Notification

Notifying one or more stations of pending message can be done by using the pre-message option. This will send notification for any messages waiting in the out-box or the relay-box. The notification is achieved by sending (posting) a very short check-summed message that contains the message ID and the receive list only. This is sufficient information for any station that receives the stub to determine if they wish to pull the rest of the message.

v. Requesting Message from Stub

1. Inbox. Any station that receives a stub notification message will have a message appear in the inbox or in the relay box with the word ‘stub’ in the ‘completion’ column. Any stub message can be selected and then the ‘Request Msg’ button pressed to initiate the transfer for the full message.

2. Relay Box. If the stub message appears in the relay box, a similar technique can be used as described above. Hen complete the full message will show ‘Verified’ in the ‘completion’ column. Verified messages also appear with a green colouring in the messages list.

III. Receiving Multiple Simultaneous Messages (Group to Peer)

The key aspect to Group to Peer communications has to do with the technique of simultaneous parallel communication.

This enables a modem application to be able to listen to multiple different offsets relative to the given frequency at the same time. This technique is used extensively in the WSJT-X and JS8Call applications for FT8 mode and JS8 mode respectively.

If the modem has this ability to listen to multiple channels simultaneously then Saam-Mail is able to utilize this for receiving messages using the passive receive approach from multiple stations simultaneously.

This could be very useful if there is a large amount of message traffic going back and forth between stations. In the event that some parts of the message are not transferred successfully during this passive mode message transfer, the receiving station can initiate a pull request to the sending station or to a relay station as described in the earlier sections.

Please note: Due to current limitations, Group to Peer communications are only available using JS8Call.

IV. Multiple Stations to/from Multiple Stations (Group to Group)

The key aspect to Group to Group communications also has to do with the modem application being able to listen to multiple different offsets relative to the given frequency at any time. Additionally there will be multiple sending stations spanning one or more groups as well as multiple listening stations also spanning one or more groups. In this way, group to group communication is now possible.

Please note: Due to current limitations, Group to Group communications are only available using JS8Call.

8. Saam-Mail Station Roles

In any given situation stations are often assigned a different role for facilitating message transfers. Saam-Mail is very flexible and integrates with multiple other platforms to allow use of a wide variety of communication techniques. Saam-Mail offers additional capabilities that enhance the line-up of techniques available.

Some of the different station roles that can be used in conjunction with Saam-Mail are as follows:

1. Message relay station for store and forward
2. Communications hub station
3. Ad-hoc mesh node station
4. Message Repository station

These different roles are described in the following sections.

Please note: During a disaster situation the internet may not be reliable. If portions of the internet are still usable then the internet based modes described in the following sections might still be viable. However, when utilizing internet based communication during a disaster receiving a confirmation reply via the same communication channels will be necessary to gauge the effectiveness of that particular mode of operation. This would apply to internet connected gateways, regular email and other internet based techniques.

I. Message Relay Station: Store and Forward

When a station is operating as a message relay the following Saam-Mail features can be used

1. Relay Box. The relay box will store any messages received by the station that do not include the relay station in the receive list. To forward a message that is stored in the Relay box, select the message in the relay messages table then click on the ‘Add to Out-box’ button. This will queue the message up in the out-box. Now specify the station you wish to send the message to in the ‘Connect To:’ field then click on the out-box tab, select the message that you just queued and then click send.
2. The relay station can listen for requests directly to pull a specific message with a given message ID or requests for which stations have copies of a given message stored. A relay station can respond to both of these requests and either initiate a send or respond to a pull request.

II. Communications Hub Station

A hub station typically receives messages from one or more sources and then sends these out.

The Saam-Mail features used by a hub station include:

1. Saam-Mail has the ability to export Saam-Mail messages so that they can be sent via other applications. Any messages can be exported into an export format by clicking on the copy button. This will produce a text version of the message along with a SAAM-MAIL-EXPORT= tag at the end that contains all the necessary information to reproduce a full copy of the message in Saam-Mail. Any messages received that contain a SAAM-MAIL-EXPORT= tag can be copied to the clipboard and then imported directly into Saam-Mail by clicking on the ‘Clipboard Import’ button. The message will then be added either to the inbox or the relay box and from there it can be posted to the out-box for sending out via Saam-Mail.
2. Winlink. To utilize this feature follow the section that describes winlink integration in the advanced features section. By using this approach, messages can be sourced from winlink or pat-winlink and sent out via Saam-Mail. Additionally any winlink messages received that contain a SAAM-MAIL-EXPORT= tag can be imported directly using the clipboard as described above then sent out.
3. Email. Similarly any emails received by the hub station that have the SAAM-MAIL-EXPORT= tag can be processed in a similar manner to the above.
4. Voice. Stations receiving message via voice can use the ‘Compose Msg’ tab to create a message then post to the out-box to be sent out via Saam-Mail.

III. Ad-hoc Gateway Station

Using similar techniques to those already described, it would be possible without any special configuration to set up a station as an Ad-hoc Gateway simply by starting up the Saam-Mail application. The integration with internet based applications currently requires a manual cut and paste approach as described in the earlier sections however this functionality opens up the possibility of setting up an Ad-hoc Gateway to handle message traffic.

IV. Ad-hoc Mesh Node Station

An Ad-hoc mesh node could utilize a combination of all the features described above. In addition the following Saam-Mail features may also be utilized:

1. Send Saam-Mail messages to a Winlink gateway. This can be achieved by using the Saam-Mail export function as described above. Copy the Saam-Mail message to the clipboard using the ‘copy’ button and then paste this into a new winlink email. This can then be sent out to a winlink gateway using winlink.
2. Sending via email. In a similar manner a message can be created in email using the same cut and paste approach then sent out over email.

V. Message Repository Station

A message repository station could be used to store many messages that can be queried and pulled by any other station. There will typically be two types of message in this category as follows:

1. Messages addressed to a group: By using this approach group messages can be stored and retrieved by any member of that group. This could be a very useful method to disseminate information to a group of stations.
2. Individually addressed messages: This is similar to the relay role described above. These types of messages are available for any station to pull as required.

9. Real-Time Simultaneous Parallel Communication

The technique of Real-Time Simultaneous Parallel Communication relates to the ability for a single digital station to receive and decode multiple digital transmissions simultaneously on a single frequency, as well as the ability for a single digital station to send messages to a specific group of digital stations simultaneously. This technique has been around at least since WSJT-X / FT8 and JS8Call / JS8 arrived on the ham radio scene. WSJT-X for example can typically decode 30 or more 50Hz wide FT8 stations simultaneously on a single HF (3kHz wide) frequency. The technique was further enhanced by JS8Call with the ability to send messages to a specific group of digital stations. The technique has proved to be extremely popular in the context of making a QSO for station to station communication but has not been applied to any great degree to communications to/from groups of stations i.e. peer to group, group to peer and group to group. Digital nets do exist and are slowly becoming more popular but at present these are more of a niche area that is still evolving.

When utilized for transferring messages among groups of stations, Real-Time Simultaneous Parallel Communication has enormous potential. Included in this section are several examples and scenarios giving details of how performance improvements can be realized. Key to understanding this is something called a multiplier effect as described in the following section.

I. Multiplier Effect

To illustrate the multiplier effect, two scenarios are provided for comparison. The first relates to peer to group communications and the second relates to group to peer communications. Each scenario compares the gateway approach to the real-time simultaneous parallel communication approach. The first scenario involves sending a form from one station to five recipients (peer to group). The second involves sending five forms, one from each of five sending stations, to one recipient station (group to peer). In both scenarios, message transfer is achieved using radio based communications only.

i. Scenario 1a: Gateway

To send a form to five recipients, the sending station must first contact the gateway and send the message. Following the initial message upload, each of the recipient stations must then contact the gateway to retrieve the message. The stations have no knowledge that there is a message waiting for them on the gateway so they must check back periodically by connecting to the gateway to see if there are any messages waiting. When a recipient station sees there is a message on the gateway for them, they must then retrieve that message. The same process is also required for the other four recipient stations. At the end of this process each recipient station will have a copy of the message.

To summarize this, the sending station and each of the five recipient stations has contacted the gateway at least one time to retrieve the full message. This effectively ties up this particular channel (frequency + offset) for six separate communication sessions. So for any given communications mode, the total time = upload transfer time + (retrieval transfer time x 5) = message transfer time x 6.

ii. Scenario 1b: Real-Time Simultaneous Parallel Communication

When distributing communications to many stations simultaneously in real-time there are many advantages. Not only does the message get received in real-time so the station does not have to keep checking back, but the message can be transferred more quickly. Using this approach, the sending station will send out the message once and each of the receiving stations will receive that message as it is being sent out. If each of the receiving stations were to receive the message on the first attempt, this will have significantly shortened the amount of radio time required to transfer the message on that channel (frequency + offset). In this case for a given communications mode, total time = message transfer time x 1.

iii. Scenario 1 Comparison

So to compare scenario 1a and 1b above, you can see that scenario 1b is six times faster than scenario 1a. Or to put this another way, scenario 1b can use a modulation mode that is six times slower than that used by scenario 1a and still transfer the messages in the same amount of time with the added benefit that the stations do not have to keep calling the sender to check if there is a message waiting. This is a multiplier effect as it relates to real-time simultaneous parallel communication in a peer to group setting.

This multiplier effect for a group of stations on frequency and listening all at the same time makes this a highly efficient mode for transferring messages to groups of stations.

In addition there is a second multiplier effect as it relates to real-time simultaneous parallel communication in a group to peer setting. This scenario involves five stations sending a single form to a single recipient station:

iv. Scenario 2a: Gateway

Each of the five sending stations must first contact the gateway to upload the message. The recipient station must contact the gateway periodically and then retrieve the messages. Regardless of the fact of weather this retrieval is spread over multiple sessions or all in one session, the receiving station will still be utilizing the air time for five message transfers. So to summarize this the total time that a given channel (frequency + offset) is in use using this approach is total time = (5 x message upload time) + (5 x message download time) = 10 x message transfer time

v. Scenario 2b: Real-Time Simultaneous Parallel Communication -Single Channel

Each of the five sending stations will use a single channel (frequency + offset) to send a single message to a receiving station. Because this is being done using a single channel, each of the sending stations must take turns in using this channel in the same way as if they were sending to a gateway. The receiving station will receive these messages as they are being sent. To summarize this, the total time on air for this transfer of the five messages using this approach is total time = 5 x message transfer time.

vi. Scenario 2c: Real-Time Simultaneous Parallel Communication -Multi Channel

Each of the five sending stations will use a different channel (frequency + offset) to send a single message to a receiving station. As they are using multiple channels, the messages can all be sent simultaneously. As long as the receiving station is able to receive multiple message transfers simultaneously, as is the case when using Saam-Mail with the js8call opmode, the receiving station will receive these message as they are being sent. To summarize this, the total time on air for this transfer of the five messages using this approach is total time = 1 x message transfer time.

vii. Scenario 2 Comparison

To compare these approaches, scenario 2b completes the message transfer twice as quickly as scenario 2a. Or to put this another way, scenario 2b can use a modulation mode that is half the speed of that used by 2a and still transfer the messages in the same amount of time.

Now compare 2a with 2c. Scenario 2c has the added benefit that it is able to utilize multiple channels simultaneously in addition to the other benefits of real-time simultaneous parallel communication. The net effect of scenario 2c compared to scenario 2a is that of a ten times multiplier; scenario 2c is 10x faster than scenario 2a. Or to put this another way, scenario 2c can use a modulation mode that is ten times slower than that used in scenario 2a and still transfer the messages in the same amount of time.

viii. Comparison Summary

The above scenarios help to illustrate the degree to which the technique of real-time simultaneous parallel communication when used in a group setting is more efficient. In the examples given, when using a single channel the speed multiplier is between 2x and 6x times faster. When using multiple channels this speed multiplier is 10x times faster.

When these improvements are factored together with the Saam-Mail optimizations of sending content only, dictionary compression and run length encoding, this linear performance improvement multiplier becomes exponential. This multiplier relates to aspects of communication other than the different digital modulation modes themselves.

This gives a backdrop where it is clear there is plenty of headroom for the operator to select the digital mode that is appropriate, whether it is JS8 slow mode or Olivia or PSK or something else and not have to be unduly concerned about any net-negative impacts to the message transfer speed. The often stated criticism that “it’s too slow, it’s like watching paint dry” clearly does not apply in this context.

In reality, there may be some additional effects that come into play, that may adjust these multipliers up or down by some amount but it is clear that by using Saam-Mail in a group setting along with real-time simultaneous parallel communication techniques, transferring messages between the stations is extremely efficient. These multiplier effects can be used to real benefit and may help address the challenges related to transferring messages via gateway stations.

Having described the many benefits of real-time simultaneous parallel communication, the other question relating to this technique is the overall manner in which it is used. That is described in the next sections.

II. Digital Net Setting

Real-time simultaneous parallel communication can be used in a net setting. The idea is that a digital net would provide the coordination at a high level and then the individual stations would then move to a different channel to do the actual message transfer then return back to the net. Alternatively all stations could remain on a single channel throughout the entire process and the net and the individual message transfers would proceed one after the other.

III. Free-for-all

The other approach to doing this is currently hypothetical. This would look something like the many FT8 stations going back and forth on one of their main frequencies such as 7.074. The idea here is that each individual station would manage its own activities with regard to message transfer...in essence a free-for-all approach. The many different message transfer modes available in Saam-Mail make this type of approach feasible.

10. Single Mode Digital vs Multi Mode Digital

Saam-Mail can be used in conjunction with a digital net, either using the same digital mode as that of the message transfer (single mode) or using a different digital mode (mixed mode). Having a digital net running at the same time that forms are being transferred is a way to coordinate the message traffic and may enable a more efficient message transfer process.

I. Single Mode Digital

An example of this would be running a JS8 net and using Saam-Mail to send forms to the group over JS8.

One big advantage of using JS8 and JS8Call is that the narrow bandwidth of the JS8 modes allows for many stations to be present on frequency at the same time. This enables a single station to receive multiple messages from multiple stations at the same time. This multiplier effect increases message transfer efficiency significantly.

II. Mixed Mode or Multi-Mode Digital

Multimode digital requires the ability to use two different digital modes at the same time. When multiple digital modes are used it is possible to receive and decode the different modes of communication simultaneously. Only one of the modes is used to transmit at any given time.

One reason for doing multi mode digital is that some modes are better suited for running a net and other modes are better suited for passing message traffic. When running a net and passing message traffic on the same frequency, the use of two different modes may be desirable and also provides additional flexibility.

There are several approaches to doing multimode digital as follows:

1. Using two different fldigi modes in one application on one computer using one radio. One for running a net and one for sending message traffic
2. Using two different js8call modes in one application on one computer using one radio. One for running a net and one for sending message traffic
3. Running two separate programs such as fldigi and js8call side by side on the same computer using one radio. Js8call could be used for running the net and fldigi for passing message traffic.
4. Running two separate programs such as fldigi and js8call on two different computers using two radios.

Option 1 above can be achieved simply by typing directly into fldigi to run a net using a particular mode then when it is time to transfer messages, switch to Saam-Mail and let Saam-Mail handle the message transfer using the mode selected within Saam-Mail.

Option 2 above can also be realized with the following. Running a js8 net using the js8-net program on github (https://github.com/gh42lb/js8-net). Saam-Mail can be run alongside js8-net. The only accommodation that needs to be made is in the js8call external API settings to allow at least 2 ports to use the JSON interface instead of 1.

Option 3 above. Although this is experimental, I have confirmed that this is quite possible at least on the raspberry pi 4b. In order to use this approach there are several pre-requisites as follows:

1. Configure JS8Call to use the digital interface directly. I use a signalink and the JS8 audio settings are configured to point to the alsa_input.usb-BurrBrown_from_Texas_Instruments_USB_AUDIO_CODEC-00.analog-stereo and alsa_output.usb-BurrBrown_from_Texas_Instruments_USB_AUDIO_CODEC-00.analog-stereo devices.
2. Configure fldigi to use the default input and output.
3. Set the audio pop-up menu at the lower right of the raspberry pi screen to USB AUDIO CODEC for both input and output.
4. You must be using a VOX based setup either through a vox setting on your rig that detects audio signals and engages ptt accordingly or you can use an external device such as the Signalink for this purpose.

Using the above setup JS8Call and fldigi can be run at the same time and both applications can decode the respective traffic simultaneously in real time on the pi. Performance also appears to be acceptable.

Using this approach, multiple JS8 signals and a single fldigi signal can be received and decoded simultaneously. It would also be possible to use either JS8 or fldigi to transmit to other stations using a single digital mode such as fldigi for example to verify accurate message receipt. As soon as the message is transferred then both stations will continue to receive any transmissions for a net or other activity that may be going on either on JS8 or fldigi.

Option 4 above. This is also possible simply by running the net from one computer and running Saam-Mail on another computer. Participation on the net can be done from the first computer. When it is time to transfer forms, the operator would make sure all transmits regarding the net are complete, then pause everything on the first computer, then switch to the second computer to transfer the message. Once message transfer and all transmits are complete, the operator would then switch back to the first computer to continue on the net.