ARIM Messaging Program v2.8 Help

Introduction

ARIM means "Amateur Radio Instant Messaging" and the ARIM program is a host mode program for the ARDOP TNC being developed by Rick KN6KB and John G8BPQ.

ARIM is written in C and distributed as source code under the terms of the GPL 3.0 license. Developed on Ubuntu Linux, it should compile and run on any modern Linux installation (including Raspbian for Raspberry Pi). Using Microsoft Windows? No problem, ARIM will build and run in the excellent Cygwin environment for Microsoft Windows. Alternatively, on Windows 10, ARIM can be built in the Bash on Ubuntu on Windows environment too. Either of these environments let you build and run ARIM on the same Windows host that ARDOP_Win or ARDOP_2Win are installed on. Compiling the source code is easy, the only build dependency beyond the standard C libraries are the ncurses and zlib development libraries.

Features include:

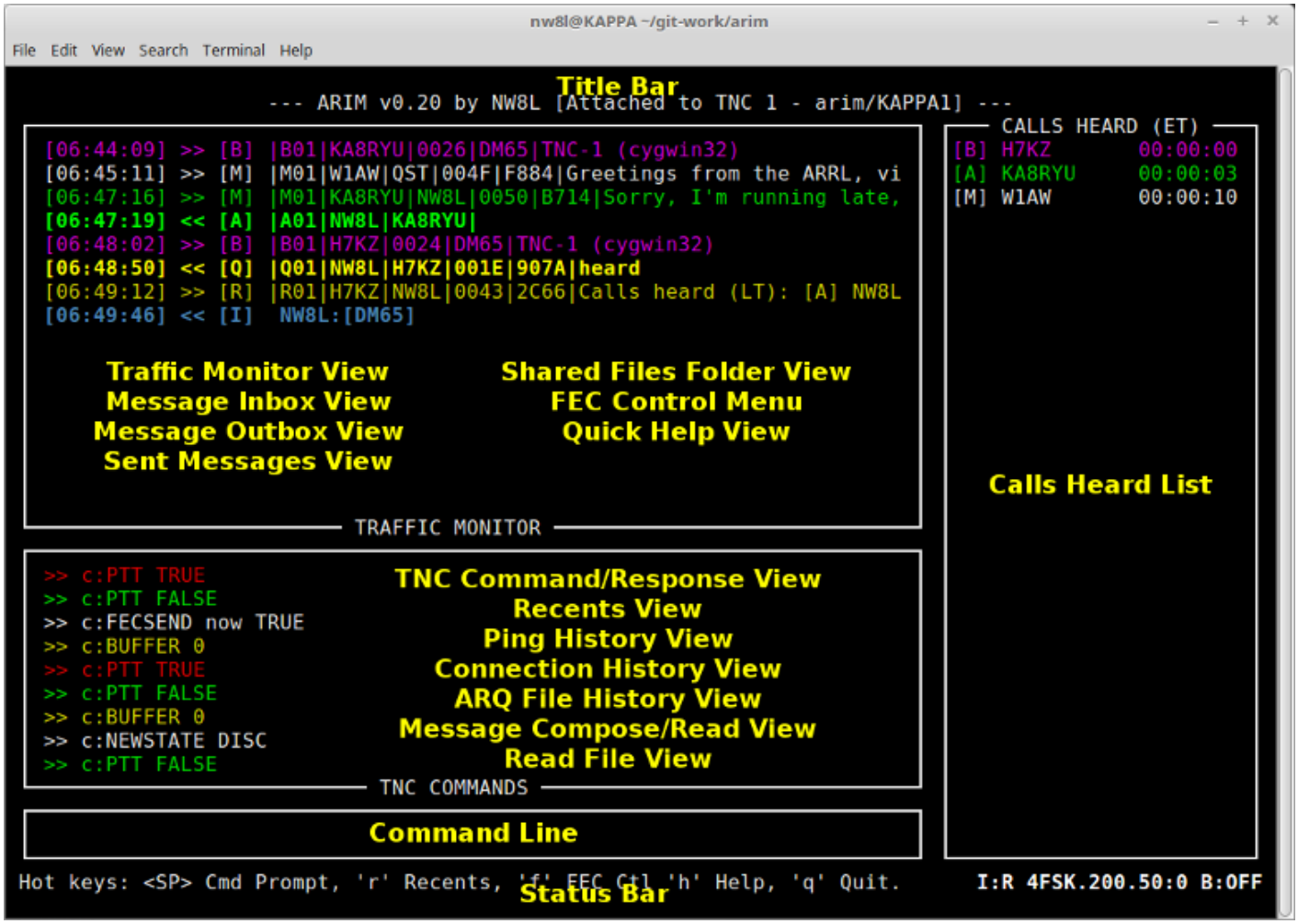
- Uses a TUI (text user interface), suitable for running in an SSH session and using only the keyboard for input. If running in a GUI terminal emulator, ARIM automatically resizes itself when the terminal window is resized.
- Attaches to ARDOP software TNCs running on the local host or remote hosts over a TCP connection. Can attach to a TNC-Pi9K6 hardware TNC mounted on a Raspberry Pi host, over a serial port.
- Uses .ini style configuration file for program settings.
- Allows multiple TNC configurations to be defined in the configuration file. Operator may attach or detach from any of them while ARIM is running. Each TNC has separate station and net call sign list, port name etc.
- Uses a connectionless all-text protocol layered over the ARDOP FEC mode transport, using 8-bit character encoding. Selective call is supported and data payloads are verified by a 16-bit CCIT checksum.
- Allows the operator to send an ARDOP "ping" frame to another station and view the resulting signal report.
- Allows casual keyboard-to-keyboard chat using "unproto" FEC text transmissions.
- Supports ARDOP ARQ mode both as client and as a server for keyboard to keyboard chat and access to BPQ BBSs with ARDOP ports.
- Allows file uploads and downloads in ARQ mode (both text and binary file types), with a compression option. This sends the message or file in RFC 1950 "zlib" format, using the "deflate" compression method documented in RFC 1951. The data is automatically decompressed at the receiving station to recover the original file or message.
- Supports mutual authentication of ARQ sessions using a digest access authentication scheme. This can be used to confirm the identity of the connected station or control access to designated "access-controlled" shared files directories.
- Displays a "calls heard list" view showing stations heard on the air, including timestamp and type of frame last heard.
- Displays a "recents list" view showing recently received messages from other stations.
- Displays a "ping history" view showing ping activity, including signal reports and timestamps, for each station for which pings have been either sent or received.
- Displays a "connection history" view showing ARQ connection activity, including grid square locators, session start and duration times, and number of bytes sent and received.
- Displays an "ARQ file history" view showing ARQ file transfer activity including transfer start and duration times, file size, checksum, file name and whether compression was used.
- Displays a traffic monitor view showing heard ARIM and ARDOP frames, with a 500 line scrollback buffer and optional timestamps.
- Displays a TNC command/response view allowing the operator to issue commands to the TNC and monitor its "async response" stream.
- Supports beaconing with announcement of call sign, grid square and TNC name.
- Allows the operator to send text messages addressed to a specific station call sign or net call sign.
- Confirms message receipt with short ACK frame sent by receiving station to sending station (only when message is addressed to a station call, not a net call). At the receiving station, the message is automatically

stored to the inbox for later reading. At the sending station, if no ACK frame is received, the operator is prompted with a choice to either store the message to the outbox or discard it.
- Allows the operator to forward received text messages to another station or a net, or to save received text messages to a file.
- Supports query of remote stations for information such as ARIM and TNC versions, grid square, calls heard list, shared *text* files and text generated by scripts (dynamic files).
- Displays status flags in message listing to indicate if a message has been read (R), forwarded (F) or saved to file (S).
- Allows messages to be composed "off-line" (no TNC attached) and stored to the message outbox to be sent later. After transmission messages are stored in the "sent messages" mailbox.
- Offers option for automatically purging messages from mailboxes aged beyond a specified number of days. Alternatively, this process can be invoked manually from the message listing views.
- Offers option for automatically repeating attempts to send a message if the result is a NAK or an ACK timeout, with option to progressively "downshift" to a more robust FEC mode each time.
- Offers option for sending "pilot pings" in advance of attempts to send a message or query, or to initiate an ARQ session. If the destination station fails to respond to the pilot pings, the transaction is abandoned to prevent tying up the channel when propagation is poor or the destination station is off the air. In this case, the operator is prompted with a choice to either store the message to the outbox or to discard it.
- Offers option for message tracing using RFC-822 style *Received:* headers. In this way a record of the message's progress through a network is built up as it is forwarded from station to station.
- Offers options for separate message traffic logging and debug trace logging with automatic daily log rotation.
- Offers convenient menu driven control of TNC's FEC mode and frame repeat count settings.
- Includes text file reader for files in the shared files directory as well as the *arim.ini* configuration file.

The ARIM program is a work in progress and I am interested in feedback. I monitor the ARDOP Users group at Groups.io and can be reached there, or at the arim-ham group at Groups.io where files and other information will be posted.

The ARIM screen layout

The ARIM screen is divided into different sections. Some of these can host more than one "view". The current view is identified by the title at the bottom of these sections.



- **Title Bar** This includes:
  - Date and time
  - ARIM version
  - TNC attach status
  - Key TNC parameter data in format **`L:v E:v`**
    Where **`L`** means "listen", **`E`** means "enpingack" and **`v`** is either **`T`** for TRUE or **`F`** for FALSE.
  - New message and file counters (press 'n' to clear)
- **Traffic Monitor View** This usually displays inbound and outbound ARIM and ARDOP frames. The different frame types are identified by a letter code:
  - **`[B]`** indicates an ARIM beacon frame.
  - **`[M]`** indicates an ARIM message frame.
  - **`[Q]`** indicates an ARIM query frame.
  - **`[R]`** indicates an ARIM response frame.
  - **`[A]`** indicates an ARIM ACK frame.
  - **`[N]`** indicates an ARIM NAK frame.
  - **`[I]`** indicates an ARDOP ID frame.
  - **`[P]`** indicates an ARDOP PING frame.
  - **`[p]`** indicates an ARDOP PINGACK frame.
  - **`[U]`** indicates an "unproto" (non-ARIM) ARDOP FEC frame.
  - **`[@]`** indicates an ARDOP ARQ frame.
  - **`[E]`** indicates a bad ARDOP frame.
  - **`[!]`** indicates a bad ARIM frame.

The direction of data flow is shown by the **<<** (outbound) and **>>** (inbound) arrows. A 500 line scrollback buffer allows review of past traffic. Outbound frames sent from ARIM to the TNC are printed in bold text to make them stand out. To enable timestamping of frames in the monitor view set the **mon-timestamp** parameter in the **[ui]** section of the *arim.ini* configuration file to TRUE. This area is also used for:

- Listing the message inbox contents in response to the **li** command.
- Listing the message outbox contents in response to the **lo** command.
- Listing sent messages in response to the **ls** command.
- Listing shared files in response to the **lf** command.
- Displaying the FEC Control Menu in response to the 'f' hot key.
- Displaying the Help contents in response to the 'h' hot key.

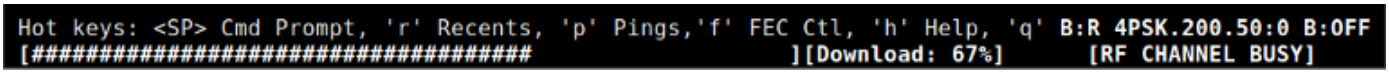
Any of these secondary views can be closed by pressing the 'q' key. When view titles are enabled these views will be identified by their title at the bottom of the monitor view area.

- **TNC Command/Response View** This usually displays the stream of TNC async responses when attached, together with any commands sent to the TNC by the ARIM program, or by the operator. The operator can send TNC commands by pressing SP for the command prompt, and entering the command prefixed by the **!** (bang) character. The direction of data flow is shown by the **<<** (outbound) and **>>** (inbound) arrows. Outbound commands sent from ARIM to the TNC are printed in bold text to make them stand out. This area is also used for:
  - Listing recently received messages. Press 'r' to open this view; press 'r' again to close it.
  - Listing the TNC's ping history. Press 'p' to to open this view; press 'p' again to close it.
  - Listing the TNC's connection history. Press 'c' to to open this view; press 'c' again to close it.
  - Listing the TNC's ARQ file transfer history. Press 'l' to to open this view; press 'l' again to close it.
  - Composing multi-line messages in response to the 'sm' and 'cm' commands. The message is entered line by line at the command prompt and displayed here.
  - Displaying messages from the inbox or outbox in response to the **rm** command.
- **Command Line** Press **<SP>** to open a command prompt here. Commands can be directed to the TNC (if attached) by prefixing them with the '!' character. Text prefixed with the ':' character is sent over the air as "upproto" ARDOP FEC frames if attached to a TNC. Unproto transmissions allow interoperation with stations using ARDOP Chat FEC mode. In ARQ chat mode text entered here is sent to the connected station.
- **Calls Heard List** Here are displayed the call signs of stations heard on the air, listed in reverse chronological order (most recent first). Each entry also shows the frame type and time heard. Time heard can be either *last time* heard, indicated by "(LT)" in the list title, or *elapsed time* since last heard, indicated by "(ET)". The format is HH:MM:SS for last time heard, and DD:HH:MM for elapsed time.
- **Status Bar** A hot key menu appears here normally but status and warning messages appear here temporarily also. At the far right is a group of status indicators.
  In FEC mode they are:
  - **ARIM:TNC status indicator** - format is **I/B:T/R** where ARIM state is either **I** (idle) or **B** (busy) and TNC state is either **T** (transmitting) or **R** (receiving). When busy, new transmissions are blocked. However, pressing the <ESC> key when busy will cancel the current operation and return ARIM to the idle state so that a new message or query can be sent immediately.
  - **FEC mode and repeat count indicators** - format is **fecmode:fecrepeat** . These can be changed by pressing 'f' to open the FEC Control Menu and selecting new values.
  - **Beacon status indicator** - shows the beacon interval time in minutes, or "OFF" if disabled. Use the **btime** command to change the beacon interval time.
  - **Busy channel indicator** - displays **[RF CHANNEL BUSY]** when the TNC reports that the channel is busy. When this is seen, transmissions are inhibited until the channel is quiet again. The **busydet** parameter in the **[tnc]** section of the *arim.ini* configuration file controls the sensitivity of the TNC's channel busy detector. BUSY notifications are *not* sent by the TNC to ARIM during an ARQ session, so this indicator will not appear then.

  In ARQ mode they are:
  - **Input lockout indicator** - format is **!** , displayed when the TNC is busy with a file transfer. When present, text or commands entered at the command prompt are ignored.
  - **Remote station call sign** - format is **ARQ:CALL[+]**. Starting with ARIM v1.2, the optional suffix **+** is appended to the call sign if the ARQ session is authenticated.
  - **Connection bandwidth** - format is **BW** where BW is the maximum bandwidth negotiated when the ARQ connection was made, for example **2000**.
  - **TNC state indicator** - format is **S:STATE** where STATE is usually one of **IRS**, **RtoS** (abbreviation of IRStoISS), **ISS**, **IDLE** or **DISC**.

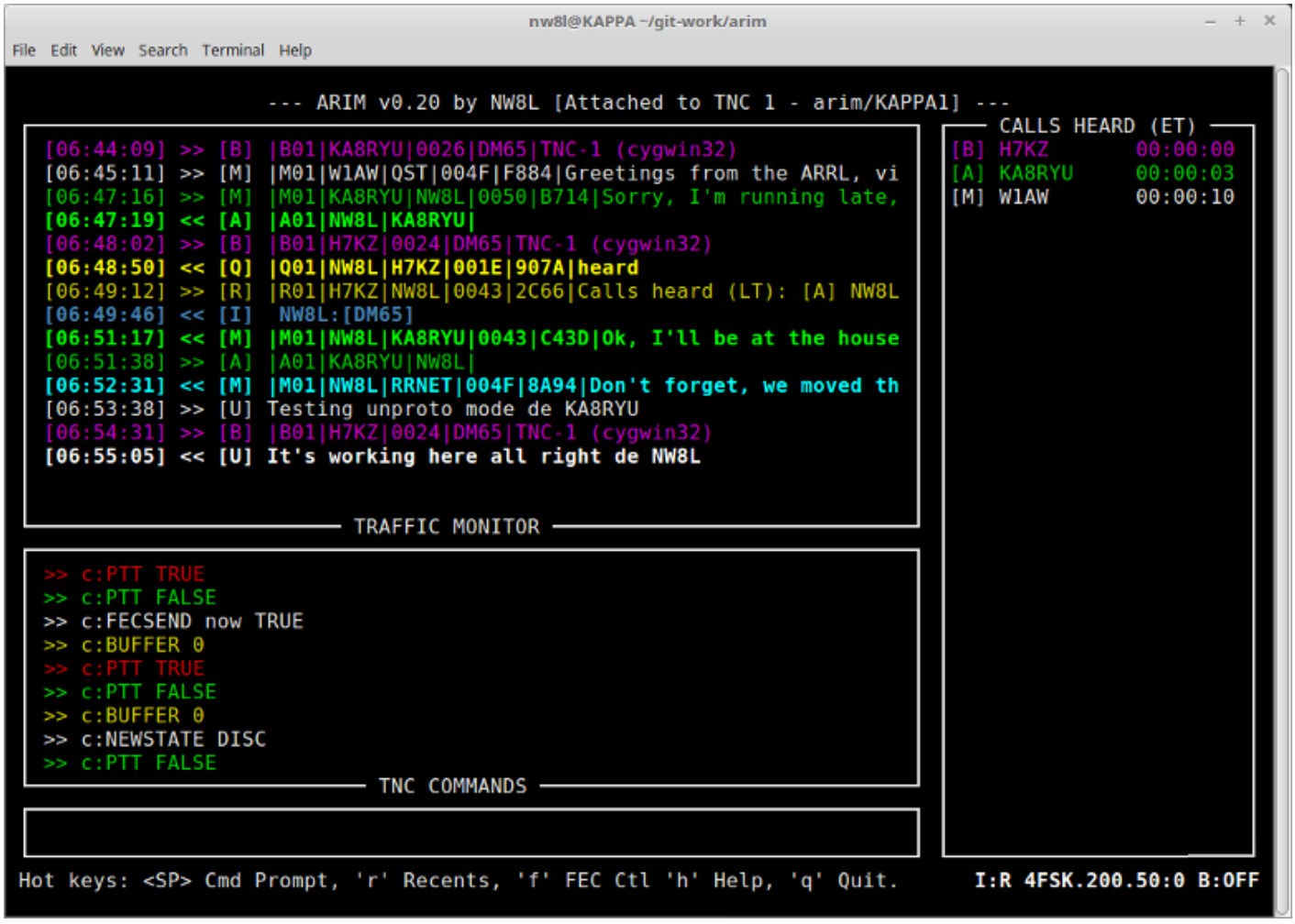
  On the left side, a progress meter appears during many data transfer operations:

- **ARQ message transfers** - Message uploads and downloads (/MPUT, /MGET).
- **ARQ file transfers** - Binary file uploads and downloads (/FPUT, /FGET).
- **ARQ file listing transfers** - File listing uploads and downloads (/FLGET, /FLPUT).
- **FEC message transfers** - Message uploads and downloads (SM).
- **FEC query response transfers** - Query response uploads and downloads (SQ).



Color coded display and UI theme options

Color coding can be enabled by setting the **`color-code`** parameter in the **`[ui]`** section of the *arim.ini* configuration file to TRUE (the default). When enabled, elements in the traffic monitor view, calls heard list and TNC command view are color coded to highlight the different frame/command types and group related traffic flows together. The color scheme is controlled by the theme setting. User defined custom themes can be used in addition to the built-in DARK and LIGHT themes. The default theme is DARK, whose color scheme is described below.



The traffic monitor view and TNC command view show very different types of information so different color coding schemes are used for each. The calls heard view is just a reflection of information in the traffic monitor view and uses the same color coding conventions.

- **Traffic Monitor/Calls Heard** ARIM frame types are color coded according to frame type and the call sign to which they are addressed. This makes it possible to distinguish traffic sent to or from the TNC call or net call from other traffic. Outgoing frames are printed in a bold font to distinguish them from incoming frames.
  - **ARIM Message/Ack frames (mycall)** `[M]` and `[A]` frames sent to or from *the TNC 'mycall' call sign* are colored green.
  - **ARIM Message frames (netcall)** `[M]` frames sent to or from *a TNC 'netcall' call sign* are colored cyan. This makes net traffic stand out in the traffic monitor view.

    - **ARIM Query/Response frames** `[Q]` and `[R]` frames sent to or from *the TNC 'mycall' call sign* are colored yellow.
    - **ARIM Nak frames** `[N]` frames sent to or from *the TNC 'mycall' call sign* are colored red.
    - **ARIM Beacon frames** `[B]` frames sent or received from *any station* are colored magenta.
    - **ARIM Error frames** `[!]` frames addressed to *a TNC 'mycall' or 'netcall' call signs* are colored red.
    - **ARDOP ID frames** `[I]` frames sent or received from *any station* are colored blue.
    - **ARDOP PING/PINGACK frames** `[P]/[p]` frames sent to or from *the TNC 'mycall' call sign* are colored blue.
    - **ARDOP Error frames** `[E]` frames sent or received from *any station* are colored red.
    - **Other frames** All other monitored frames are colored white. This includes various ARDOP frame types (e.g. ARQ frames) and ARIM frames not addressed to the TNC call sign or net call sign.
- **TNC Commands** TNC commands are distinguished by function and direction of travel. This makes it easier to follow the interactions between the ARIM host program and the TNC and recognize TNC state changes announced in the asynchronous response stream.
    - **Commands from Host to TNC** These are bolded and colored cyan.
    - **Asynch Responses from TNC** These are color coded as follows:
        - **`PTT TRUE`** async responses are colored red.
        - **`PTT FALSE`** async responses are colored green.
        - **`BUFFER`** status async responses are colored yellow.
        - **`BUSY`** status async responses are colored magenta.

Themes control the color coding and text attributes of the various elements of the user interface. Two built-in themes, DARK and LIGHT, are always available, and up to 5 custom themes can be defined by the user in the *arim-themes* file. When ARIM starts, the theme set by the **`theme`** parameter of the **`[ui]`** section of the *arim.ini* configuration file is loaded. You can change the UI theme at any time by entering the `theme` command at the command prompt like this:

```
theme name
```

where *name* is the name of the theme. This erases the terminal window and re-draws it using the specified theme. Note: themes work only when the **`color-code`** parameter in the **`[ui]`** section of the *arim.ini* configuration file is TRUE.

Custom themes are defined in the *arim-themes* file, which is read by ARIM at startup. If you are using the portable binary ARIM distribution, this file is located in the same *arim-2.8* directory containing the ARIM executable file. If you compiled and installed ARIM from the source distribution, this file is located in the *arim* data directory in your home directory. Here is an example theme definition:

```
# bluebird is an example custom theme, adjust to taste.
[theme]
name = bluebird
ui-background-color = BLUE
ui-default-color = WHITE
ui-default-attr = NORMAL
ui-status-indicator-color = WHITE
ui-status-indicator-attr = BOLD
ui-status-notify-color = WHITE
ui-status-notify-attr = BOLD
ui-dialog-color = BLACK
ui-dialog-attr = NORMAL
ui-dialog-background-color = YELLOW
ui-clock-color = WHITE
ui-clock-attr = NORMAL
ui-msg-cntr-color = WHITE
ui-msg-cntr-attr = NORMAL
ui-ch-busy-color = WHITE
ui-ch-busy-attr = NORMAL
ui-title-color = WHITE
ui-title-attr = NORMAL
tm-error-color = RED
tm-error-attr = NORMAL
tm-message-color = GREEN
tm-message-attr = NORMAL
tm-query-color = YELLOW
tm-query-attr = NORMAL
```

```
tm-ping-color = BLACK
tm-ping-attr = BOLD
tm-id-color = BLACK
tm-id-attr = BOLD
tm-net-color = CYAN
tm-net-attr = NORMAL
tm-beacon-color = MAGENTA
tm-beacon-attr = NORMAL
tm-arq-color = WHITE
tm-arq-attr = NORMAL
tm-tx-frame-attr = BOLD
tc-cmd-color = CYAN
tc-cmd-attr = BOLD
tc-ptt-true-color = RED
tc-ptt-true-attr = NORMAL
tc-ptt-false-color = GREEN
tc-ptt-false-attr = NORMAL
tc-buffer-color = YELLOW
tc-buffer-attr = NORMAL
tc-ping-color = BLACK
tc-ping-attr = BOLD
tc-busy-color = MAGENTA
tc-busy-attr = NORMAL
tc-newstate-color = YELLOW
tc-newstate-attr = NORMAL
```

arim-themes uses an "INI File" format to store theme definitions. The file is divided into *sections*, each containing a list of *keys* which specify theme properties as name=value pairs, one to a line. Section names occupy a line by themselves and are enclosed in square brackets. Lines beginning with the '#' character are treated as comments. If a key is not present in the file the property is set to its default value. Each theme is configured in a separate **`[theme]`** section. This is a group of color and text attribute *keys* for various UI elements in ARIM. The allowable colors are:

`BLACK, RED, GREEN, YELLOW, BLUE, MAGENTA, CYAN, and WHITE.`

What you see will vary with the color palette used by your terminal emulator. The allowable text attributes are:

`BLINK, BOLD, DIM, ITALIC, NORMAL, REVERSE, STANDOUT and UNDERLINE.`

Not all of these attributes work on a given terminal type; you'll need to experiment with them. Up to 5 themes can be defined. Example themes are included in the distributed *arim-themes* file. Use these as a starting point for your own custom themes. See the arim(5) man page for details.

To view the contents of the *arim-themes* file, open the shared files viewer and enter the **`rt`** command. Do this to learn the names of custom themes for use with the `themes` command.

Downloading and installing

The prerequisite for ARIM is the ARDOP TNC. Both version 1 and version 2 TNCs are supported.

- **Version 1** *ardopc* for Linux, *piardopc* for the Raspbian OS on RPi, and *ARDOP_Win* for Microsoft Windows. Recommended versions are 1.0.4.1b-BPQ for *ardopc* and *piardopc*, and 1.0.4 for *ARDOP_Win*.
- **Version 2** *ardop2* for Linux, *piardop2* for the Raspbian OS on RPi, and *ARDOP_2Win* for Microsoft Windows. Recommended versions are 2.0.3.9-BPQ for *ardopc* and *piardopc*, and 2.0.4 for *ARDOP_Win.*
- **TNC-Pi9K6** *ARDOP_Teensy* or *ARDOP2_Teensy* running on the TNC-Pi9K6 TNC. Use the latest source code, which is available in the TeensyProjects.zip archive on G8PBQ's download site. Learn more about configuring and using the TNC-Pi9K6 here.

ARDOP version 2 introduced changes to the FEC modes and ARQ bandwidth options. If upgrading from ARDOP TNC version 1.0.x, check your *arim.ini* file and make sure that '`fecmode`' parameters in the `[tnc]` sections don't reference one of the FEC modes deleted in ARDOP TNC v2:

- 4FSK.200.50S, 4FSK.500.100S, 4FSK.500.100, 4FSK.2000.600S, 4FSK.2000.600
- 4PSK.200.100S, 4PSK.500.100, 4PSK.1000.100, 4PSK.2000.100
- 8PSK.200.100, 8PSK.500.100, 8PSK.500.100, 8PSK.1000.100, 8PSK.2000.100

- 16QAM.1000.100, 16QAM.2000.100

Also make sure that your '`arq-bandwidth`' parameters in the `[tnc]` sections reference one of these options *only*:

- 200, 500 or 2500

ARIM v2.8 is *not* compatible with ARDOP_2Win TNCs earlier than version 2.0.4. It is compatible with ardop2 and piardop2 v2.0.3.9-BPQ and later.

ARIM v2.8 is *not* compatible with ARIM v1.9 and earlier. It is compatible with gARIM v0.1 and higher.

Information about downloading and installing the ARDOP TNCs is found here:

- ardopc TNC for Linux Operating Systems provides instructions applicable to the *ardopc*, *ardop2*, *piardopc* and *piardop2* TNCs, which are available for download here.
- ARDOP_Win or ARDOP_2Win TNC (part of the ARDOP Chat installation) for Windows Operating Systems. To download the ARDOP Chat installer you must be a member of the ARDOP Users group at groups.io. The installer is found in the Files area of the group.

If you plan to run ARIM in the Cygwin environment on Windows, then Cygwin must be installed first. The installer for this is found on the Installing and Updating Cygwin Packages page. The Cygwin Walkthrough and Beginner's Guide may be helpful in getting started.

If you plan to run ARIM in the Bash on Windows environment on Windows, then the Windows Subsystem for Linux must be installed first. How to Install and Use the Linux Bash Shell on Windows 10 describes installation and use of Bash on Windows. Microsoft publishes a helpful Bash on Windows FAQ which is part of their official Bash on ubuntu on Windows site.

When using Cygwin, you will work in the Cygwin Terminal. A shortcut to the Cygwin Terminal should be placed on your Windows desktop by the Cygwin installer.

When using Bash on Windows, you will work in the Bash Terminal launched by the "Bash on Ubuntu on Windows" Start Menu item.

Next, you need to either build ARIM from source or download a "portable" precompiled binary package.

- To build from source, download the source code tarball:

  Current ARIM Version 2.8 source code and help file

  Enter this command:

  ```
  tar xzvf arim-2.8.tar.gz
  ```

  to unpack the *arim-2.8* directory. This contains source code files and other information including build instructions and this help file.

  To compile and install ARIM, change directory to *arim-2.8* and enter these commands:

  ```
  ./configure
  make
  sudo make install (Linux, Raspbian)
  or
  make install (Cygwin)
  ```

  This installs the *arim* executable, *arim(1)* and *arim(5)* man pages, and additional documentation including this help file. The *arim(1)* man page documents *arim* command line options and the *arim(5)* man page documents the *arim.ini* configuration file (use the command `man 5 arim` to read it).

  To compile from sources you need the GNU compiler, *gcc*, the *make* utility and the *ncurses* and *zlib* development libraries and headers. The *INSTALL* file included with the source distribution contains detailed instructions for installing these and building ARIM from source.

To run ARIM, enter:

```
arim
```

at the command prompt. The first time ARIM runs it will create a directory named *arim* in your home directory containing data files and a model *arim.ini* configuration file. It also contains three subdirectories: *files* for shared files, *doc* with symbolic links to documentation including this Help file, and *log* for log files. Once created, the contents of the *arim* directory will not be overwritten by subsequent installations of ARIM or deleted if ARIM is uninstalled. The links in the *doc* directory will always point to the most recently installed documentation files.

To configure ARIM, open the *arim.ini* file in a text editor and edit as needed. You must define at least one TNC by setting call signs, the grid square locator, and TNC IP address and port number.

  - If editing the file in Windows, use a text editor that understands the Unix text file format, like *Wordpad* or the freeware text editor *Programmer's Notepad*.
  - On hosts with IPv6 enabled, 'localhost' may not work as the address for an ARDOP TNC instance running on the same Windows PC. Use the IPv4 address of the host, e.g. "192.168.1.54", or the IPv4 loopback address "127.0.0.1" instead. The IPv4 address can be discovered by running `ipconfig` from the command line. If running ARDOP_Win or ARDOP2_Win, set this address in the *Virtual TNC Setup* dialog box as well.

- Alternatively, on Red Hat derived Linux systems, you can follow these instructions to build an RPM package from the ARIM source distribution, and use it to install ARIM.

- If you prefer to install precompiled "portable" versions of ARIM, here are download links for 32 and 64-bit Ubuntu Linux (and derivatives like Linux Mint), Raspbian (on Raspberry Pi), and 32 and 64-bit Cygwin for Windows:

Current Version 2.8 binary and help file (32-bit Ubuntu Linux)

Current Version 2.8 binary and help file (64-bit Ubuntu Linux)

Current Version 2.8 binary and help file (Raspbian on Raspberry Pi)

Current Version 2.8 binary and help file (32-bit Cygwin on Windows)

Current Version 2.8 binary and help file (64-bit Cygwin on Windows)

Download the one you need and enter this command:

```
tar xzvf filename
```

to unpack the *arim-2.8* directory. This contains the executable file, data files and a model *arim.ini* file. It also contains three subdirectories: *files* for shared files, *doc* with copies of documentation files including this help file, and *log* for log files. The *arim-2.8* directory can be copied anywhere (e.g. a USB flash stick) and ARIM run from it locally, so it's called the "portable" installation. It's self-contained, with messages, shared files and log files moving with the directory if it's relocated.

To run ARIM, change directory to *arim-2.8* and enter:

```
./arim
```

The `"./"` preceding the file name tells the system to run the copy of arim located in the current directory.

To configure ARIM, open the *arim.ini* file in a text editor and edit as needed. You must define at least one TNC by setting call signs, the grid square locator, and TNC IP address and port number.

  - If editing the file in Windows, use a text editor that understands the Unix text file format, like *Wordpad* or the freeware text editor *Programmer's Notepad*.
  - On hosts with IPv6 enabled, 'localhost' may not work as the address for an ARDOP TNC instance running on the same Windows PC. Use the IPv4 address of the host, e.g. "192.168.1.54", or the IPv4 loopback address "127.0.0.1" instead. The IPv4 address can be discovered by running `ipconfig` from

the command line. If running ARDOP_Win or ARDOP2_Win, set this address in the *Virtual TNC Setup* dialog box as well.

Run arim with the `--help` option to print out command line options:

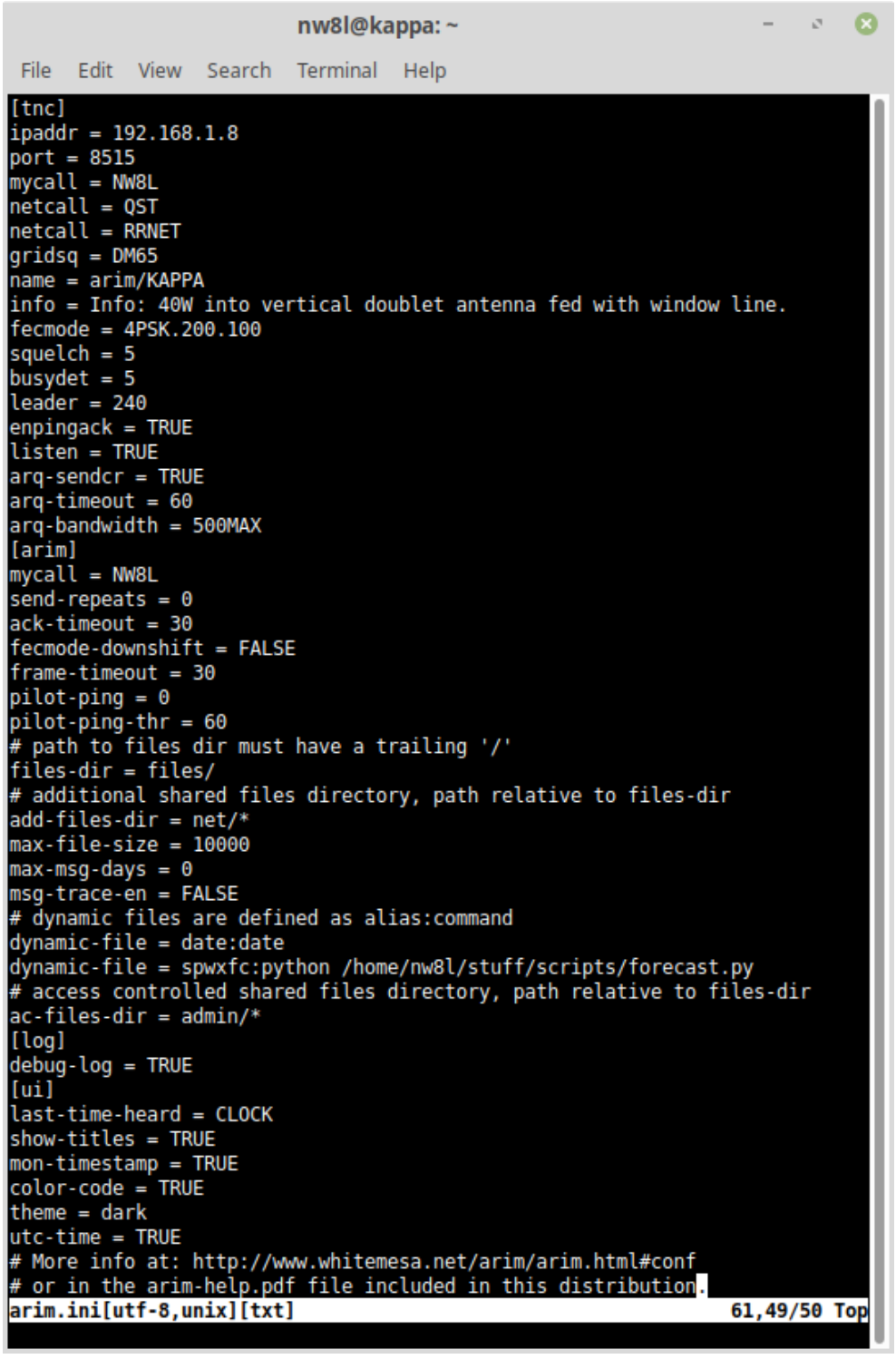
```
Usage: arim [OPTION]
  -v, --version            print version information
  -f, --config-file FILE   use configuration file FILE
  -p, --print-conf FILE    print configuration file listing to FILE
  -h, --help               print this option help message
```

Configuration

By default, ARIM reads an "ini" format configuration file named *arim.ini* at startup. If you are using the portable binary ARIM distribution, this file is located in the same *arim-2.8* directory containing the ARIM executable file and data files. If you compiled and installed ARIM from the source distribution, this file is located in the *arim* data directory in your home directory. To override the default configuration file location, run arim with the `--config-file` command line option, for example:

```
arim --config-file /home/nw8l/HF/arim-net.ini
```

Here is an example configuration file:



Multiple TNCs can be defined. A subset of the ARDOP TNC parameters are initialized here, but these can be overridden from the TNC command line after the program starts. Most options have reasonable default values which are used if they are not found in the .ini file.

- **[tnc]** Each TNC is configured in a separate [tnc] section. The first [tnc] section in the file defines port 1, the second port 2, and so on. The limit is 10 TNC definitions.
  - **interface** The TNC interface type, either TCP for connecting to a software TNCs like ardop2 or ARDOP_2Win, or SERIAL for connecting to the TNC-Pi9K6 hardware TNC. Default: TCP.
  - **ipaddr** The IPv4 address of the TNC, either in "dotted quad" numerical form or a host name e.g. 'localhost' or 'DELL-1520.example.net'. Max length for host names is 253 characters. Default: 127.0.0.1. NOTE: On hosts with IPv6 enabled, 'localhost' may not work as the address for an ARDOP instance running on the same host. Use the IPv4 address of the host, e.g. "192.168.1.54", or the IPv4 loopback address "127.0.0.1" instead. The IPv4 address can be discovered by running `ipconfig` from

the command line. If running ARDOP_Win or ARDOP2_Win, set this address in the *Virtual TNC Setup* dialog box as well.
- **port** The TCP port on which the TNC is listening. Default: 8515.
- **serial-port** The serial port device name, for example */dev/serial0*, used to connect to the TNC-Pi9K6 hardware TNC on a Raspberry Pi host. Max length for device names is 63 characters. Default: /dev/serial0.
- **serial-baudrate** The baud rate for the serial port used to connect to the TNC-Pi9K6 hardware TNC, either 9600, 19200, 38400, 57600 or 115200. Default: 115200.
- **mycall** The station call sign, e.g. NW8L or NW8L-4, max length 10 characters. Call must be no longer than 7 characters and may have optional SSID in ranges: -A to -Z or -0 to -15. ARIM will respond to queries and messages addressed to this call. Messages will be stored in the inbox. Default: NOCALL.
- **netcall** A net call sign, e.g. RRNET, max length 10 characters. Any printable characters are allowed. When sending a message to this call, no ACK is expected. When receiving, ARIM will recognize messages addressed to this call and store them in the inbox but no ACK will be returned to prevent channel congestion. ARIM will not respond to queries addressed to the net call. Up to 8 net call signs may be defined for a TNC. Default: one netcall, QST.
- **gridsq** The station's grid square locator. It must be a well formed Maidenhead locator, either 4, 6 or 8 characters long. Examples: DM65 or DM65qf or DM65qf15. Letter pairs are not case-sensitive. Default: DM65.
- **btime** The beacon interval time in minutes, or 0 to disable the beacon. Max time is 999 minutes. Default: 0.
- **reset-btime-on-tx** Control whether or not the beacon timer is reset when the station transmits an ARDOP frame. This is useful to prevent beacon transmissions from interfering with traffic between stations, e.g. on a net. This setting has no effect when the beacon is disabled. Set to TRUE to enable beacon timer reset on transmit, FALSE to disable it. Default: FALSE.
- **name** A name assigned to the TNC and advertised by the beacon, e.g. RRNET/ARIM. This is also returned when the TNC receives the ARIM 'pname' query. Max length is 31 characters. Default: Empty.
- **info** Information describing the TNC, returned in response to the 'info' query. Use the character pair **`\n`** to indicate a line break if you want to format the text into multiple lines; this will be converted to a newline character in the response. Max length is 127 characters. Default: Empty.
- **fecmode** The initial ARDOP FECMODE. This is the frame type, in the format *modulation.bandwidth.baudrate*. Available modes are:
  - **ARDOP v1.x** 4FSK.200.50S, 4FSK.500.100S, 4FSK.500.100, 4FSK.2000.600S, 4FSK.2000.600, 4PSK.200.100S, 4PSK.200.100, 8PSK.200.100, 4PSK.500.100, 8PSK.500.100, 4PSK.1000.100, 8PSK.1000.100, 4PSK.2000.100, 8PSK.2000.100, 16QAM.200.100, 16QAM.500.100, 16QAM.1000.100 or 16QAM.2000.100.
  - **ARDOP v2.x** 4PSK.200.50, 4PSK.200.100, 16QAM.200.100, 4FSK.500.50, 4PSK.500.50, 16QAMR.500.100, 16QAM.500.100, 4FSK.1000.50, 4PSKR.2500.50, 4PSK.2500.50, 16QAMR.2500.100 or 16QAM.2500.100.

  Max length is 20 characters. Default: 4PSK.200.50.
- **fecrepeats** The initial ARDOP FECREPEATS value. This is the number of times each FEC frame will be repeated by the sender. This may be useful when propagation is poor, but at the cost of reduced throughput - depending on FEC mode, an ARIM frame may extend across multiple ARDOP FEC frames, each of which will be repeated. Max value is 5. Default: 0.
- **fecid** The initial ARDOP FECID value, which controls whether or not an ARDOP ID frame is sent by the TNC at the start of FEC transmissions. Set to TRUE to enable FECID, FALSE to disable it. Default: FALSE.
- **leader** The initial ARDOP LEADER time in msec. The leader is a special 50 baud two tone signal which precedes data transmission, used by the receiving TNC for synchronization. This may need to be adjusted to compensate for loss of leader due to delays in PTT or VOX keying or audio path latencies in some SDR radios. Range is 120-2500. Default: 240.
- **trailer** The initial ARDOP TRAILER time in msec. Range is 0-200. Non-zero trailer time is only needed for certain SDR radios and is a function of the audio processing latency relative to release of PTT. For these cases try a value of 100-200 msec. Default: 0.
- **squelch** The initial ARDOP SQUELCH setting. This controls the sensitivity of the TNC's leader detector. Lower values mean greater sensitivity but also greater risk of false triggering. Range is 1-10. Default: 5.
- **busydet** The initial ARDOP BUSYDET setting. This controls the sensitivity of the TNC's busy detector. Lower values mean greater sensitivity but also greater risk of false triggering. Setting the value to 0

disables the busy detector. The busy detector should be disabled only in an emergency situation or in very high local noise environments. Range is 0-10. Default: 5.
    - **listen** Control whether or not the TNC listens for ARQ connect requests or pings from other stations. Set to TRUE to enable listening, FALSE to disable it. Default: TRUE.
    - **enpingack** Control whether or not the TNC responds to pings from other stations. Set to TRUE to enable ping ACKs, FALSE to disable them. Default: TRUE.
    - **tnc-init-cmd** Specifies a TNC initialization command to send when ATTACHING to a TNC. For example:
      `tnc-init-cmd = LEADER 300`
      This is useful for passing TNC commands which are not managed by ARIM, such as (future) radio control commands, and for TNC test and development purposes. Commands are send verbatim without validation of the command name or its parameters. Max length is 128 characters. You may define no more than 32 **tnc-init-cmd** parameters. Default: None.
    - **arq-bandwidth** Sets the ARQ connection bandwidth. Available bandwidths are:
        - **ARDOP v1.x** 200MAX, 500MAX, 1000MAX, 2000MAX, 200FORCED, 500FORCED, 1000FORCED or 2000FORCED.
        - **ARDOP v2.x** 200, 500 or 2500.

      Default: 500.
    - **arq-negotiate-bw** Controls whether or not the TNC will negotiate ARQ bandwidth for incoming connections. Set to TRUE to enable bandwidth negotiation, FALSE to disable it. Default: TRUE. *This feature currently supported by ARDOP_2Win TNC version 2.0.4 only.*
    - **arq-timeout** The inactivity timeout for ARQ connections in seconds. Range is 30-600. Default: 120.
    - **arq-sendcr** Control whether or not CRLF line endings are sent in ARQ mode, instead of Unix style LF endings. Set to TRUE to send CR, FALSE to send only LF. Default: TRUE.
    - **log-dir** The directory where log files are located if TNC specific logging is enabled. This can be an absolute path or a relative path rooted in the user's home directory. Max length is 255 characters. Default: the user's home directory.
    - **debug-log** Set to TRUE to enable debug logging for this TNC in the directory specified by the **`log-dir`** parameter, FALSE to disable it. Default: FALSE.
    - **traffic-log** Set to TRUE to enable traffic logging for this TNC in the directory specified by the **`log-dir`** parameter, FALSE to disable it. Default: FALSE.
    - **tncpi9k6-log** Set to TRUE to enable TNC-Pi9K6 logging for this TNC in the directory specified by the **`log-dir`** parameter, FALSE to disable it. Default: FALSE.
- **[arim]** This section holds settings for the ARIM messaging protocol.
    - **mycall** The call sign used as the "from" address for messages. Default: NOCALL.
    - **send-repeats** The number of times an ARIM message will be repeated in the absence of an ACK response from the recipient. It is recommended that this value not exceed 3 to prevent tying up the channel with repeats in poor conditions. Max is 5. Default: 0.
    - **ack-timeout** The maximum time in seconds after sending a message that ARIM will wait for an ACK before repeating it. Applies when **send-repeats** is not 0. Max is 999 seconds. Default: 30.
    - **fecmode-downshift** Control whether or not the FEC mode is progressively "downshifted", or changed to a more robust mode each time an ARIM message is repeated after a NAK or ACK timeout. Set to TRUE to enable, FALSE to disable downshifting. Default: FALSE.
      This works in tandem with the 'send-repeats' parameter. If 'fecmode-downshift' is TRUE and 'send-repeats' is nonzero, then progressively more robust FEC modes are used for re-transmissions after a NAK or timeout. The mode of last resort is 4PSK.200.50. For example, if the initial mode is 4PSK.500.100, then downshifting would progress to 4FSK.500.100, then 16QAM.200.100, and so on. The original FEC mode is restored after the message send operation completes. This is experimental. There are many kinds of channel impairments and no single downshift strategy is best for all. For details look at the FEC mode downshift table in the *arim_proto.c* source code file.
    - **frame-timeout** The time in seconds after which an incomplete ARIM frame will be abandoned and the receive buffer cleared. Because an ARIM frame may be spread over many ARDOP frames, a failure to receive one or more ARDOP frames will cause an ARIM timeout. Max is 999 seconds. Default: 30.
    - **files-dir** The directory in which files available for other stations to read are located. This can be an absolute path or a relative path rooted in the ARIM working directory and must be terminated with a '/' character. Max length is 255 characters. Default: files/
    - **add-files-dir** Specifies an additional shared files directory accessible by remote stations. This must be a path relative to the shared files root directory specified by the **files-dir** parameter. By default, only files in the shared files root directory may be listed or downloaded, and any directories it contains are hidden. If you need to share files organized into multiple directories, use one or more **add-files-dir**

parameters to expose them. For example:
`add-files-dir = forms/`
This allows limited access to the *forms* directory in the shared files root directory. A remote station may list, read or download the files it contains, but any subdirectories are hidden. To grant full access to a directory, including the hierarchy of subdirectories rooted there, append the '*' wildcard character to the end of the path. For example:
`add-files-dir = contests/*`
This grants full access to the *contests* directory in the shared files root directory. This exposes not only the files in *contests*, but also the hierarchy of subdirectories rooted there. Subdirectories such as *contests/2016* or *contests/2016/June* are visible, and a remote station may list and download the files they contain. Max length is 255 characters. NOTE: you may define no more than 16 **add-files-dir** parameters. Default: None.

- **ac-files-dir** Specifies an access-controlled shared files directory, accessible only by authenticated stations in an ARQ session. This must be a path relative to the shared files root directory specified by the **files-dir** parameter. If you need to limit the sharing of certain files to authenticated stations, use one or more **ac-files-dir** parameters to expose them. For example:
`ac-files-dir = admin/`
This allows limited access to the *admin* directory in the shared files root directory. An authenticated station may list, read or download the files it contains, but any subdirectories are hidden. To grant full access to a directory, including the hierarchy of subdirectories rooted there, append the '*' wildcard character to the end of the path. For example:
`ac-files-dir = admin/*`
This grants full access to the *admin* directory in the shared files root directory. This exposes not only the files in *admin*, but also the hierarchy of subdirectories rooted there. Subdirectories such as *admin/2016* or *admin/2016/June* are visible, and a remote station may list and download the files they contain. Max length is 255 characters. NOTE: you may define no more than 16 **ac-files-dir** parameters. Default: None.
- **max-file-size** The maximum size of files that can be transferred in an ARIM message. The output of the **flist** query is filtered in accordance with this limit. To disable access to shared files, set this to 0. Max is 16384 bytes. Default: 4096.
- **max-msg-days** The maximum age, in days, for messages to be kept in the inbox, outbox and sent messages mailbox. Messages that exceed this limit are automatically purged whenever the corresponding message viewer is opened in ARIM (using the 'li', 'lo' or 'ls' commands). Set to 0 to disable the automatic message purge feature. Default: 0.
- **msg-trace-en** Set to TRUE to enable message tracing, FALSE to disable it. Default: FALSE. When enabled, headers like *Received: from KA8RYU by NW8L; Jan 30 2019 05:01:48 UTC* are inserted into messages at the time of receipt. If the message is forwarded to another station with tracing enabled, another *Received:* header is added by the receiving station, and so on. In this way a record of the message's progress through a network is built up as it is forwarded from station to station (read from bottom to top).
- **dynamic-file** A dynamic file definition of the form *alias:command* where *alias* is a "dummy" file name used to invoke the command *command*, with a colon ':' separating the two, for example:
`spwxfc:python /home/nw8l/scripts/forecast.py`
Use absolute paths to script files when ARIM is built from source and installed. Relative paths can be used for "portable" binary installations where the script files are contained in same directory as the *arim* executable file. Dynamic files are used to return the output of a script or system command in response to a file query. *alias* must be unique among any other dynamic file definitions and file names in the shared files root directory. In response to the query **`sq file alias`**, *command* will be executed in a shell and its output returned in the response. *command* can be a batch file, a script invocation like `python myscript` or a system command like `date` or `uname -a`. The output size in bytes is limited by the **max-file-size** parameter. Errors generated by dynamic file scripts are written to a file named *dyn-file-error-YYYYMMDD.log* in the *log* directory. Max length is 128 characters. NOTE: you may define no more than 16 **dynamic-file** parameters. Default: None.
- **pilot-ping** The number of times a *pilot ping* will be repeated in the absence of a PINGACK response from the recipient. It is recommended that this value not exceed 3 to prevent tying up the channel with repeats in poor conditions. Set to 0 to disable pilot pings; otherwise the range is 2-15. Default: 0.
- **pilot-ping-thr** When pilot pings are enabled, this is the threshold by which signal reports from the target station are judged. If the reported constellation quality is above the threshold, the message (or query) send proceeds; if below this threshold it is canceled. It is recommended that this value be 60 or higher; choose a threshold suitable for the FEC mode in use. Min is 50, Max is 100. Default: 60.

- **[log]** Logging settings appear in this section.
  - **debug-log** Set to TRUE to enable debug logging, FALSE to disable it. Default: FALSE.
  - **traffic-log** Set to TRUE to enable traffic logging, FALSE to disable it. Default: TRUE.
  - **tncpi9k6-log** Set to TRUE to enable TNC-Pi9K6 debug logging, FALSE to disable it. Default: TRUE.
- **[ui]** User interface settings appear in this section.
  - **color-code** Set to TRUE for color coding of items in the traffic monitor view, calls heard list and TNC command view according to the scheme discussed in the Color Coded Display and UI Theme Options section. Set to TRUE to enable, FALSE to disable color coding. Default: TRUE.
  - **show-titles** Set to TRUE to show titles for all views (panes) in the UI, FALSE to hide them. Default: TRUE.
  - **last-time-heard** Selects the timestamp format used in various views. Set to CLOCK to indicate time station was last heard, in HH:MM:SS format (either local time or UTC). Set to ELAPSED to indicate elapsed time since station last heard, in DD:HH:MM format. Default: CLOCK.
  - **mon-timestamp** Set to TRUE to enable timestamps in the traffic monitor view. Set to FALSE to disable them. Default: FALSE. Prior to version 0.12 this was located in the **[arim]** section.
  - **utc-time** Selects the time zone used for timestamps in the UI and log output, and for the clock in the title bar. Set to TRUE for UTC, and FALSE for local time. Default: TRUE.
  - **theme** A theme name which selects the UI theme. This controls the color coding and text attributes of various UI elements. Two built-in themes are always available, DARK and LIGHT (DARK is the original ARIM theme). Up to 5 additional named themes may be defined in the *arim-themes* file. Default: DARK. Note: themes work only when the **color-code** parameter is TRUE.

To print the configuration file parameters to a file for analysis, run ARIM with the -p option. For example:

```
arim -p dump.txt
```

In the file is a listing of the parameters and their values *after* processing, which can be helpful for troubleshooting. Any invalid parameter values will be replaced with default values, and parameters with misspelled names will be absent.

Rig Control/PTT

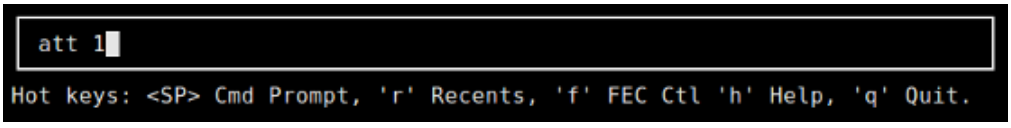
The ARIM program has no rig control features! It depends on the rig control features embedded in the ARDOP TNC.

- **ARDOP_2Win:** Choose "Optional Radio Setup" from the File menu. This opens the "Radio Settings" dialog where you may configure rig control. Be sure to check the "Enable TNC Control of Radio or PTT" box at the bottom of the dialog.
- **ardopc, ardop2:** Supports PTT. Detailed instructions are found in the Running ARDOPC guide by G8BPQ.

I use VOX on my rig and it seems that Signalink cards work too. In both cases the VOX hold or SignaLink DLY must be set to minimum.

Attaching to a TNC

Make sure the ARDOP TNC is running. Press the spacebar to open the command prompt.

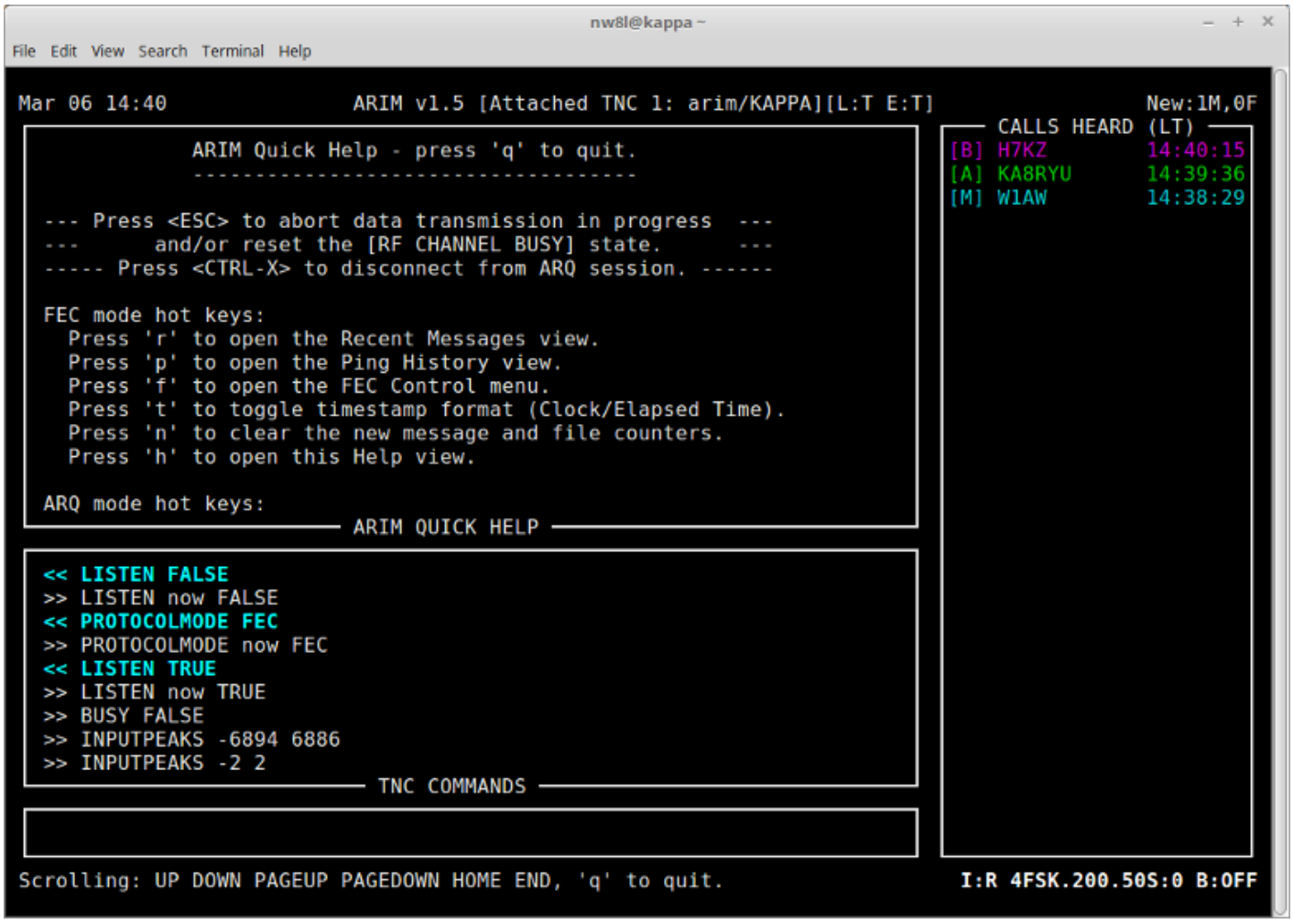


Enter the **`att`** command, a space and the TNC number, e.g. **`att 1`**. The TCP connection will be made and initialization commands from the program to the TNC will scroll by in the TNC Command view. The title bar will display the number and name of the attached TNC.

To detach from the TNC, open the command prompt and enter the **`det`** command.

Press 'h' for Help

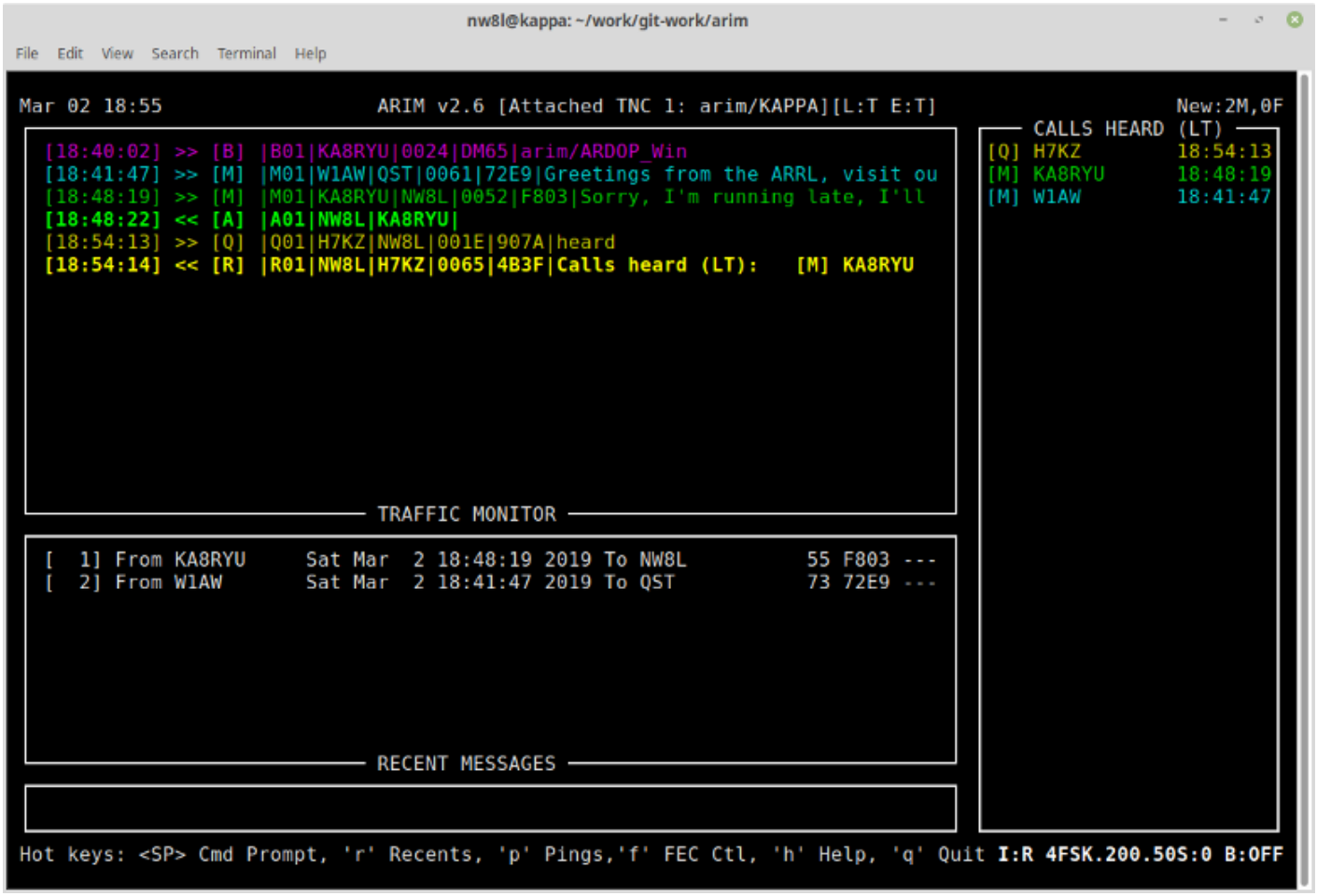
Press the 'h' hot key to open the Help viewer.



Here is a listing of all hot keys and commands. Use the UP and DOWN arrow keys or the PAGEUP, PAGEDOWN, HOME and END keys to scroll through the contents. Press 'q' to quit.

Press 'r' to toggle the Recents view

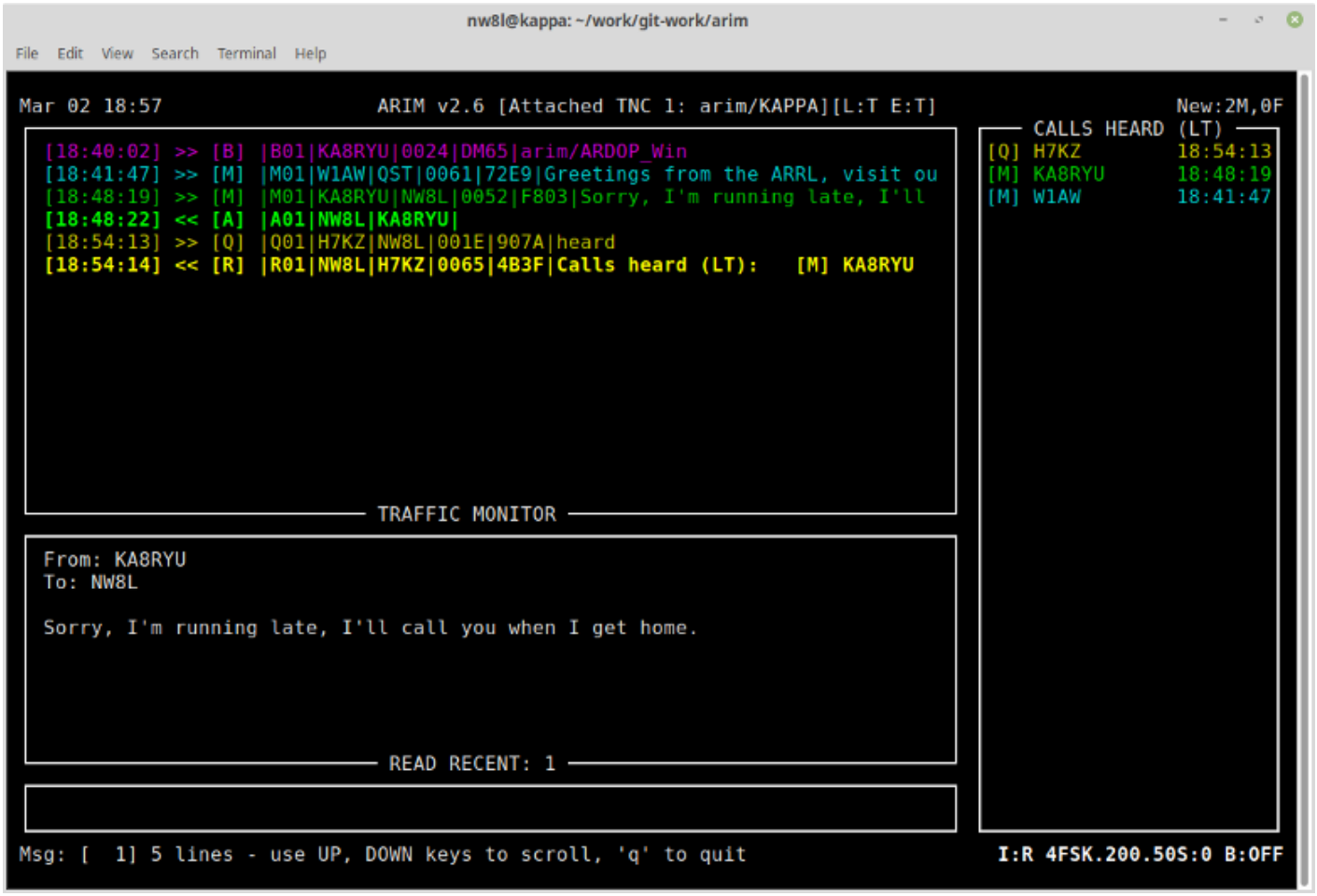
Press the 'r' hot key to toggle on the Recents view.



This is a listing of headers for messages recently received. They are numbered in reverse chronological order (most recent first). Use the UP and DOWN arrow keys or the PAGEUP, PAGEDOWN, HOME and END keys to scroll through the contents. Press 'r' to toggle the Recents view off. Headers contain the following information, from left to right:

- **Message number**, most recent first.
- **Sending station call sign**
- **Date and time**
- **Destination station call sign**
- **Size** over-the-air transfer size in bytes, decimal format (before compression).
- **Checksum** 16-bit CRC, in hexadecimal format.
- **Status flags** where *R* means *read*, *F* means *forwarded*, *S* means *saved* and - means *flag not set*.

Press the 'u' and 'd' keys to scroll the list up and down. To read a message, press the spacebar to open the command prompt and enter the command **`rr nbr`**, where *nbr* is the message number. This opens the message viewer.

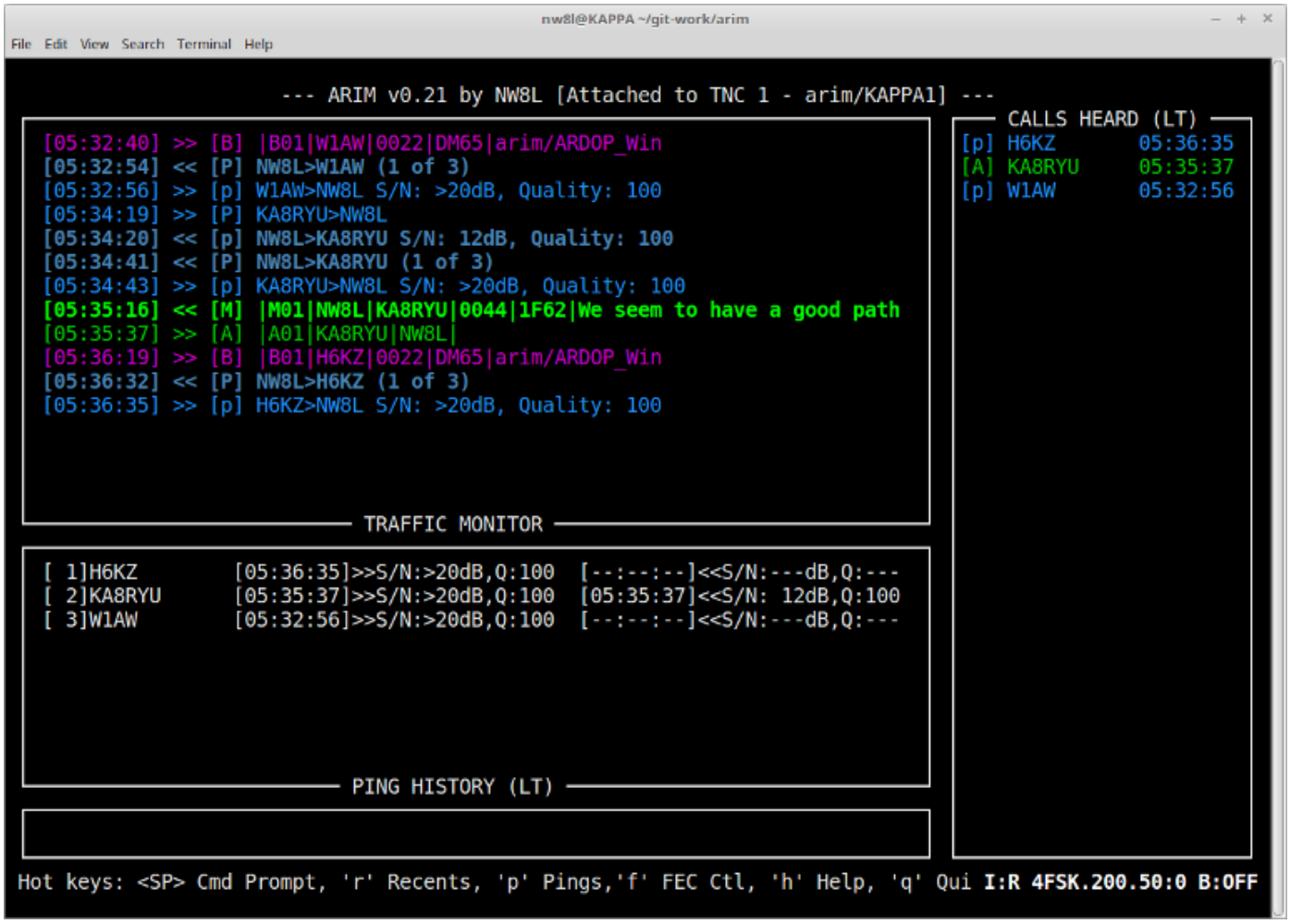


The message number and size in lines are shown in the Status Bar. Use the UP and DOWN arrow keys or the PAGEUP, PAGEDOWN, HOME and END keys to scroll through the message. Press 'q' to quit the message viewer and return to the Recents view.

To clear the view, enter `clrrec` at the command prompt.

Press 'p' to toggle the Ping History view

Press the 'p' hot key to toggle on the Ping History view.



Here is a listing of signal reports resulting from ARDOP pings sent *to other stations* by the TNC or from pings sent *to the TNC* by other stations. The history shows a line for each remote station, sorted in reverse chronological order (most recent on top). Each line includes, from left to right:

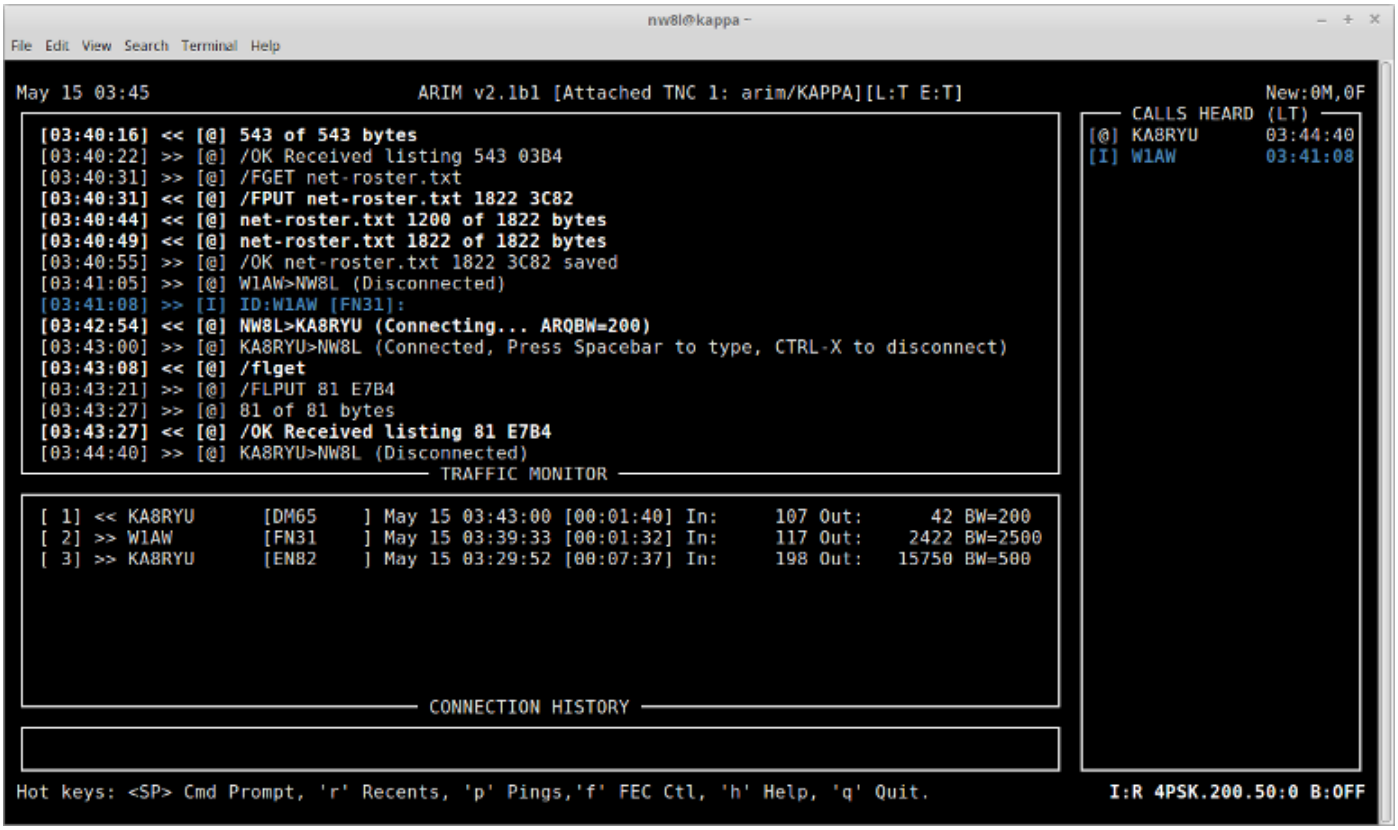
- The call sign of the remote station
- The most recent report *from* that station *to* the TNC, showing signal-to-noise ratio, the constellation quality and a timestamp
- The most recent report *to* that station *from* the TNC, showing signal-to-noise ratio, the constellation quality and a timestamp

Press 't' to toggle the timestamp format between elapsed time or the clock time. Press the 'u' and 'd' keys to scroll the list up and down.

To clear the view, enter `clrping` at the command prompt.

Press 'c' to toggle the Connection History view

Press the 'c' hot key to toggle on the Connection History view.



Here is a listing of connection reports resulting from both inbound and outbound ARQ connection requests. The history shows a line for each connection, sorted in reverse chronological order (most recent on top). Each line includes, from left to right:

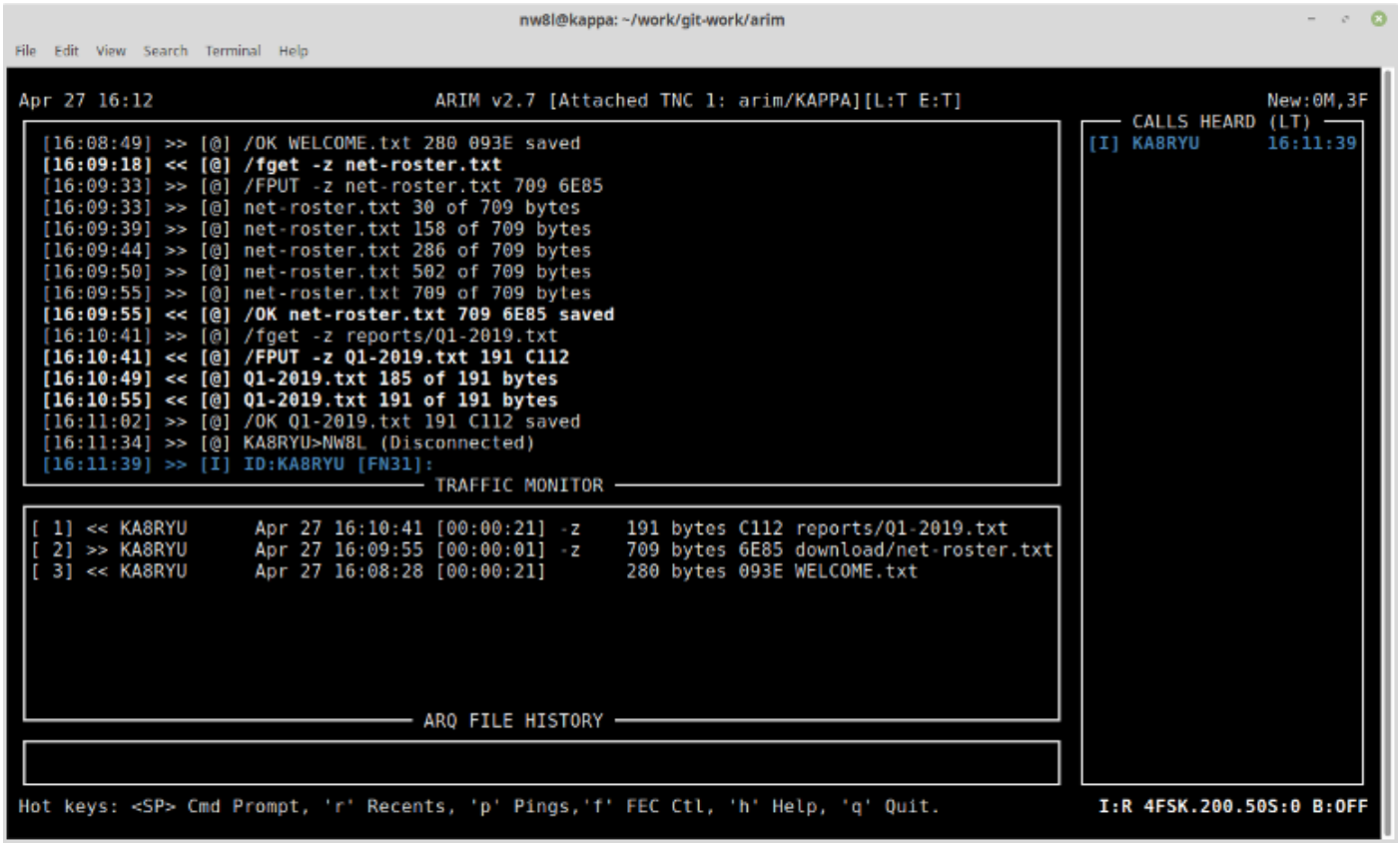
- The direction of the connection request, inbound (>>) or outbound (<<)
- The call sign of the remote station
- The grid square locator of the calling (client) station
- The ARQ session start date and time in format: `Mon DD HH:MM:SS`
- The ARQ session time duration in format: `HH:MM:SS`
- The number of bytes received
- The number of bytes transmitted
- The ARQ bandwidth

Press the 'u' and 'd' keys to scroll the list up and down.

To clear the view, enter `clrconn` at the command prompt.

Press 'l' to toggle the ARQ File History view

Press the 'l' (as in "Lima") hot key to toggle on the ARQ File History view.



Here is a listing of ARQ file transfer reports. The history shows a line for each transfer, sorted in reverse chronological order (most recent on top). Each line includes, from left to right:

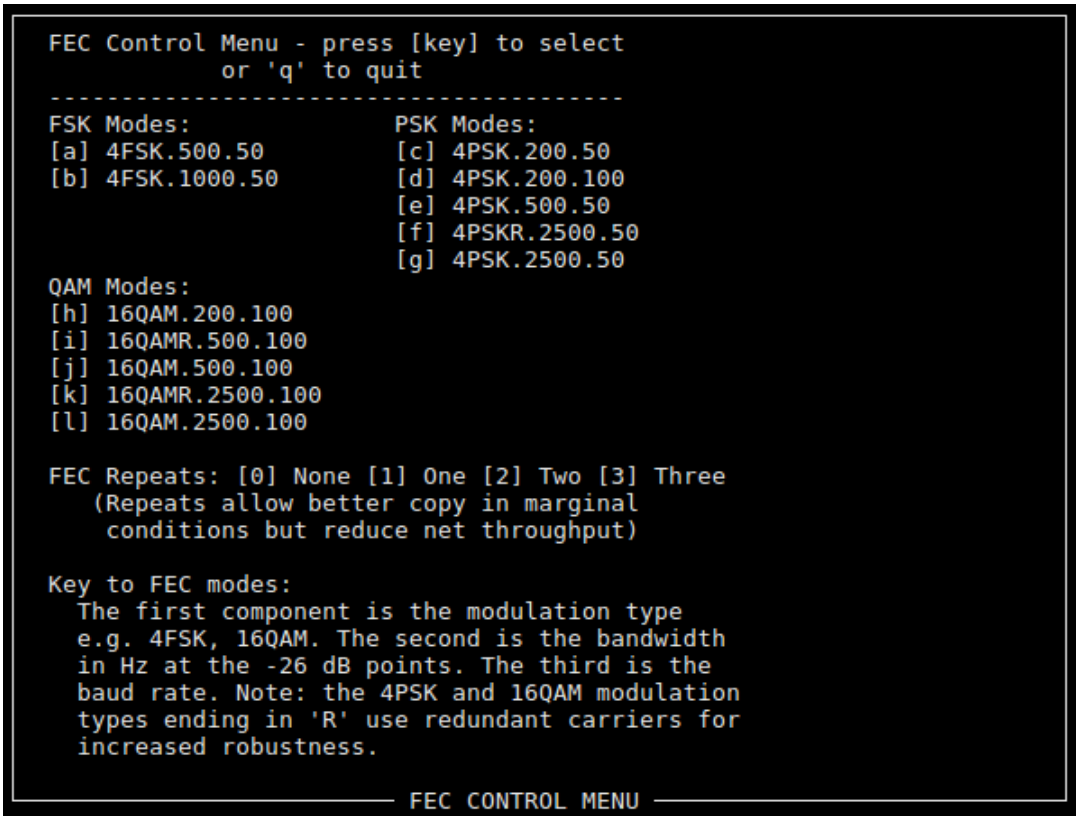
- Direction of the file transfer, inbound (>>) or outbound (<<)
- Call sign of the remote station
- Transfer start date and time in format: `Mon DD HH:MM:SS`
- Transfer time duration in format: `HH:MM:SS`
- Compression flag, `-z` if compression was invoked, or blank if not
- File size in bytes
- File payload checksum
- File name and path, relative to the *local* shared files directory

Press the 'u' and 'd' keys to scroll the list up and down.

To clear the view, enter `clrfile` at the command prompt.

Press 'f' to set FEC mode and repeat count

Press the 'f' hot key to open the FEC Control Menu.



Press a key to select one of the options. All FEC modes are listed here, as well as options to set the FEC frame repeat count to either 0, 1, 2 or 3. Use the UP and DOWN arrow keys or the PAGEUP, PAGEDOWN, HOME and END keys to scroll through the menu contents. Press 'q' to quit. Note that these are ARDOP settings; the ARIM message repeat count found in the *arim.ini* configuration file is a different thing. The ARDOP FEC repeat count affects all FEC transmissions by the TNC.

NOTE: In all views, the current FEC mode and repeat count are indicated in the status bar, to the left of the beacon time indicator. The format is *fecmode*:*fecrepeat*.

Press 't' to toggle the timestamp format

Press the 't' hot key to switch the timestamp format between:

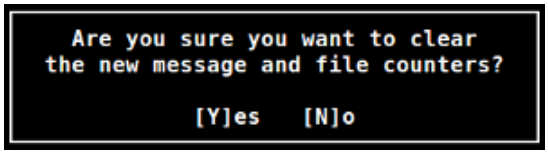
- Time last heard (LT) in format HH:MM:SS.
- Elapsed time since last heard (ET) in format DD:HH:MM.

This affects the timestamps shown in the Calls Heard list and Ping History view.

Press 'n' to clear the new message and file counters

Press the 'n' hot key to clear the new message and new file counters, located on the right-hand side of the Title Bar.

If either counter is non-zero you'll be prompted to confirm the operation:



Press 'y' to proceed or 'n' to cancel. The counters increment when a new message or file is received. They are helpful as indicators of activity during periods of unattended operation. Note that files received in FEC mode are encapsulated in a message or query response frame, so the message counter is incremented, not the file counter. New messages are listed in the convenient Recent Messages view; press 'r' to toggle this on and off. New message and file counter values are lost when the program is closed.

Press <ESC> to abort send or receive

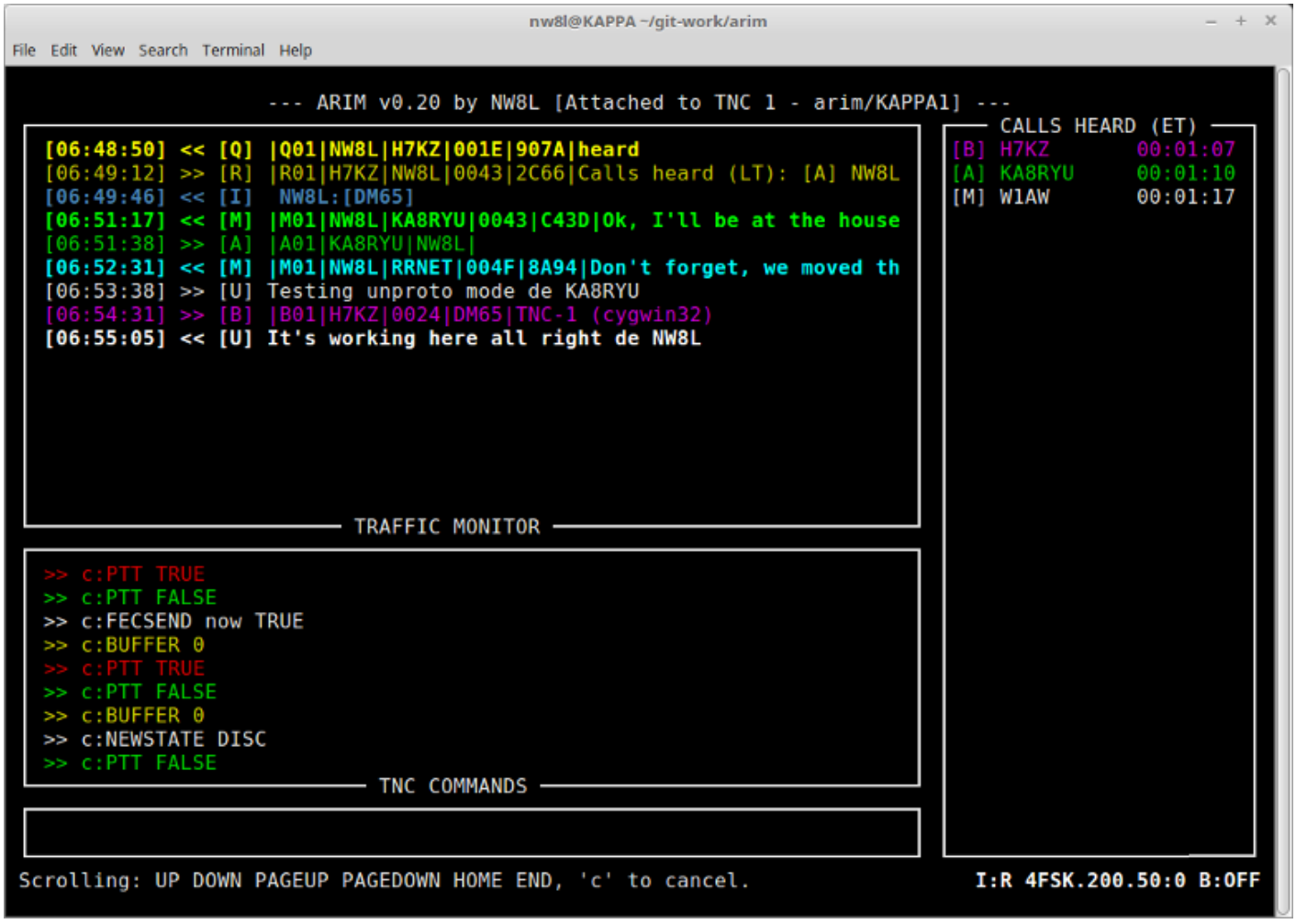
Sometimes it's necessary to abort a transmission in progress or halt a receive operation instead of waiting for it to time out. To do this press the <ESC> key to cancel the operation and return ARIM to the idle state. When transmitting, this makes the TNC stop sending, flushes the transmit buffer, and cancels any pending message repeats. When receiving (or waiting for) an ARIM frame the time out counters are reset and any pending response is canceled, making the TNC available for a new send operation immediately. A confirmation message will appear briefly on the Status Bar and the Busy/Idle indicator will reflect the change.

In FEC mode, pressing the <ESC> key also resets the `[RF CHANNEL BUSY]` state. This may be necessary in rare cases where ARIM's busy state falls out of sync with the TNC's busy detector state.

In ARQ mode, pressing the <ESC> key to abort a connection results in a "dirty disconnect". Use CTRL-X to attempt a clean disconnect before resorting to <ESC>.

Scrolling back the Monitor View

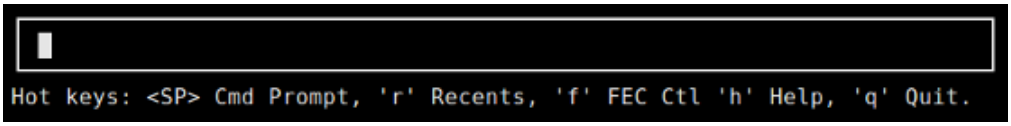
A 500 line scrollback buffer allows review of past traffic. Press the UP or DOWN arrow keys or the PAGEUP, PAGEDOWN, HOME or END keys to start scrolling.



The status bar is updated to indicate that scrolling is active. This state persists unless you stop using the movement keys for more than 30 seconds or press 'e' to cancel. Until then the view won't auto-scroll to the end when new traffic is posted.

Using the command line

Open the program and press the spacebar to open the command prompt.



There are multiple uses for the command line.

- **TNC commands** These are prefixed with the '!' character and are sent directly to the TNC e.g. **`!SENDID`**
- **Unproto messages** These are prefixed with the ':' character and are sent out over the air as raw ARDOP FEC frames, rather than being formatted as an ARIM message; e.g. **`:Hello John`**
- **ARQ chat** In ARQ chat mode text entered here is sent out over the air to the connected station.
- **ARIM commands** These manage TNCs, beacons, shared files, messages and the UI.
  - TNC commands
    - **`att`** ***`nbr`*** Attach to a TNC. *nbr* indicates the TNC number.
    - **`det`** Detach from the currently attached TNC.
    - **`btime`** ***`nbr`*** Set the beacon interval time and start it. *nbr* indicates the beacon interval in minutes, setting it to '0' disables beaconing.
    - **`btest`** Test the beacon.
    - **`srpts`** ***`nbr`*** Set the number of times a message send will be repeated in the absence of an ACK response from the recipient. Setting it to '0' disables send repeats.
    - **`ackto`** ***`nbr`*** Set the message ACK timeout in seconds.
    - **`fecds`** ***`TRUE or FALSE`*** - Enable or disables FEC mode downshifting when repeating messages.
    - **`srset`** Open a pop-up dialog showing the current message send repeat settings.
    - **`mycall`** ***`call`*** Set the TNC call sign to *call*.
    - **`netcall add`** ***`call`*** Add net call sign *call* to the list.
    - **`netcall del`** ***`call`*** Delete net call sign *call* from the list.
    - **`gridsq`** ***`loc`*** Set the grid square locator to *loc*.
    - **`pname`** ***`name`*** Set the TNC (port) name to *name*. This appears in ARIM beacon frames.
    - **`listen`** ***`TRUE or FALSE`*** - Control whether or not the TNC listens for ARQ connect requests or pings from another station.
    - **`enpingack`** ***`TRUE or FALSE`*** - Control whether or not the TNC responds to pings from another station.
    - **`tncset`** Open a pop-up dialog showing the current TNC settings.
  - Ping commands
    - **`ping`** ***`call nbr`*** Send ping to remote station *call*. *nbr* is the number of pings to send to *call* in the absence of a response; it must be in the range 2-15.
    - **`pping`** ***`nbr`*** Enable/disable the ARIM *pilot ping* feature. If enabled, ARDOP pings are sent in advance of a message or query transmission to check if the RF path to the target station is adequate. The signal report contained in the PINGACK response is compared to the *pilot ping threshold* setting and the transmission of the message or query proceeds only if the reported signal constellation quality meets or exceeds the threshold; otherwise it is canceled. *nbr* sets the number of ping repeats sent in the absence of a response from the target station; it must be in the range 2-5. If *nbr* is 0, pilot pings are disabled.
    - **`ppthr`** ***`nbr`*** Set the ARDOP signal constellation quality threshold for the pilot ping feature. This must be in the range 60-100.
    - **`ppset`** Open a pop-up dialog showing the current pilot ping settings.
  - ARQ mode commands
    - **`conn`** ***`call nbr [bw]`*** Initiate ARQ connection with remote station *call*. *nbr* is the number of connect requests to send to*call* in the absence of a response; it must be in the range 3-15. *bw* is an optional connection bandwidth (ARQBW) specifier. It can be one of *200*, *500*, *2500* or *any*. If *any*, ARIM will attempt to connect using each ARQBW setting in succession, starting with the

current ARQBW setting (set in the ARIM configuration file or by using the `arqbw` command. This is useful if the ARQBW setting of the remote station is unknown.
    - **`arqto nbr`** Set the ARQ connection inactivity timeout. *nbr* is the time in seconds.
    - **`arqbw bw`** Set the ARQ connection bandwidth. *bw* is a specifier which must be one of: 200, 500 or 2500.
    - **`arqnegbw TRUE or FALSE`** - Controls whether or not the TNC will negotiate ARQ bandwidth for incoming connections. *This feature currently supported by ARDOP_2Win TNC version 2.0.4 only.*
    - **`arqset`** Open a pop-up dialog showing the current ARQ settings.
- Messaging commands at main prompt
    - **`sm call [msg]`** Send message to station *call*. The message text may follow the call sign, but if not given then the message composer opens for input. If attached to a TNC the message is sent over the air; if not it is stored in the outbox for sending later.
    - **`cm call`** Compose message for station *call*. The message composer opens for input. When done the message is stored in the outbox for sending later.
    - **`sq call query`** Send query *query* to station *call*. See the Query another station for information topic for a list of supported queries. Station *call* will return a message containing the requested information which is stored in the message inbox.
- Message viewer commands
    - **`li`** Open the message inbox view.
    - **`lo`** Open the message outbox view.
    - **`ls`** Open the sent message view.
    - **`rm nbr`** Read message number *nbr* in the message viewer. Only available in the message listing views.
    - **`rr nbr`** Read *recent* message number *nbr* in the message viewer. Only available in the recent messages view.
    - **`fm [-z] nbr call`** Forward message number *nbr* to station *call*. In ARQ mode *only*, the optional *-z* switch compresses the message to minimize transfer time. Only available from the message inbox and sent messages views when attached to a TNC.
    - **`sv nbr fname`** Save message *nbr* to file *fname*. *fname* can be an absolute path like */home/nw8l/foo.txt* or a relative path like *foo.txt* or *files/foo.txt*. Relative paths are relative to the ARIM working directory (the current directory when *arim* was invoked). Only available in the message inbox and sent messages views.
    - **`sm [-z] nbr`** Send message number *nbr* to destination station. In ARQ mode *only*, the optional *-z* switch compresses the message to minimize transfer time. Only available from the message outbox viewer when attached to a TNC.
    - **`km nbr`** Kill (delete) message number *nbr*. Only available in the message listing views.
    - **`pm nbr`** Purge (delete) all messages older than *nbr* days. Only available in the message listing views.
    - **`cf nbr flag`** Clear flag *flag* on message number *nbr*. *flag* is one of *R* (read), *F* (forwarded), *S* (saved) or * (all flags). Only available in the message listing views.
- Shared files viewer commands
    - **`lf`** Open the shared files viewer.
    - **`rf nbr`** Read file number *nbr* in the file reader. Only available in the shared files viewer.
    - **`sf nbr call`** (FEC Mode) Send file number *nbr* to station *call*. Only available in the shared files viewer.
    - **`sf [-z] nbr [dir]`** (ARQ Mode) Send file number *nbr* to the remote station. The optional *-z* switch compresses the the file to minimize transfer time. The optional *dir* parameter specifies the destination directory at the remote station, which is a path relative to the shared files root directory. Only available in the shared files viewer.
    - **`ri`** Read the *arim.ini* configuration file. Only available in the shared files viewer.
    - **`rp`** Read the *arim-digest* password digest file. Only available in the shared files viewer.
    - **`rt`** Read the *arim-themes* UI themes file. Only available in the shared files viewer.
    - **`cd nbr`** Open directory number *nbr* and list its files. Only available in the shared files viewer.
- Remote shared files viewer commands
    - **`rf nbr`** (ARQ Mode) Read file number *nbr* in the traffic monitor view. Only available in the remote shared files viewer.
    - **`gf [-z] nbr [dir]`** (ARQ Mode) Download file number *nbr* to the local station. The optional *-z* switch compresses the the file to minimize transfer time. The optional *dir* parameter specifies the

destination directory at the local station, which is a path relative to the shared files root directory. Only available in the remote shared files viewer.
    - **`cd -z nbr`** (ARQ Mode) Open directory number *nbr* and list its files. The optional *-z* switch compresses the listing to minimize transfer time. Only available in the remote shared files viewer.
- ARQ mode messaging commands at main prompt
    - **`/sm [-z] [msg]`** Send message to the connected station. The *-z* option compresses the message to minimize transfer time. The message text may follow, but if not given then the message composer opens for input.
    - **`/mlist`** Return a list of messages in the remote station's outbox that are addressed to your station. *Requires that your station be authenticated with the remote station.* If not, then the remote station will respond with an authentication challenge.
    - **`/mget [-z] [nbr]`** Download messages addressed to your station from the remote station's outbox to your inbox. The *-z* option compresses the message to minimize transfer time. *nbr* optionally specifies the maximum number of messages to download. The default is 10. *Requires that your station be authenticated with the remote station.* If not, then the remote station will respond with an authentication challenge.
- ARQ mode file transfer commands at main prompt
    - **`/flist [dir]`** Return a list of shared files and directories. *dir* is an optional directory path, relative to the shared files root directory, e.g. `contests/2016/FOBB`. If *dir* is not given, then the the shared files root directory is listed; otherwise the specified directory is listed. By default only files in the shared files root directory and dynamic files are listed. Subdirectories are not listed unless they are made visible by an *add-files-dir* configuration file parameter. Access-controlled subdirectories defined by the *ac-files-dir* configuration file parameter are indicated with the **!** (bang) character like this: `!DIR`.
    - **`/flget [-z] [dir]`** Downloads a list of shared files and directories, then displays it in the remote shared files viewer. The *-z* option compresses the file to minimize transfer time. *dir* is an optional directory path, relative to the shared files root directory, e.g. `contests/2016/FOBB`. If *dir* is not given, then the the shared files root directory is listed; otherwise the specified directory is listed. By default only files in the shared files root directory and dynamic files are listed. Subdirectories are not listed unless they are made visible by an *add-files-dir* configuration file parameter. Access-controlled subdirectories defined by the *ac-files-dir* configuration file parameter are indicated with the **!** (bang) character like this: `!DIR`. The viewer shows a numbered list of files and directories, making it easy to read or download files without typing in lengthy file names. Learn more about using `/flget` here.
    - **`/file fn`** Print a file to the traffic monitor view. *fn* is the name of a file in the shared files directory, or a file path relative to it, e.g. `contests/2016/FOBB.log`
    - **`/fget [-z] fn [> dir]`** Download a file to the local station. The *-z* option compresses the file to minimize transfer time. *fn* is the name of a file in the remote station's shared files directory, or a file path relative to it, e.g. `contests/2016/FOBB.log`. The optional *dir* parameter specifies the destination directory at the local station, relative to the shared files root directory, e.g. `contests/2016`. The destination directory must made visible by an *add-files-dir* configuration file parameter at the local station. If *dir* is not specified, the file is stored in the *download* subdirectory in the local station's shared files root directory. ARIM will create the destination directory if it doesn't already exist.
    - **`/fput [-z] fn [> dir]`** Upload a file to the remote station. The *-z* option compresses the file to minimize transfer time. *fn* is the name of a file in the local station's shared files directory, or a file path relative to it, e.g. `contests/2016/FOBB.log`. The optional *dir* parameter specifies the destination directory at the remote station, relative to the shared files root directory, e.g. `contests/2016`. If *dir* is not specified, the file is stored in the *download* subdirectory in the remote station's shared files root directory. The destination directory must made visible by an *add-files-dir* configuration file parameter at the remote station. ARIM will create the destination directory if it doesn't already exist.
- Digest authentication commands
    - **`/auth`** Manually trigger the mutual authentication process.
- Password management commands
    - **`passwd client-call server-call password`** Create or change a password in the local password digest file. *client-call* is the call sign of the *client* station. This is a station whose operator will connect to another station, the *server* station, and use the password to authenticate when challenged. *server-call* is the call sign of the TNC at the *server* station to which the client station connects. This is the call sign set by the `mycall` parameter in the corresponding `[tnc]` section in

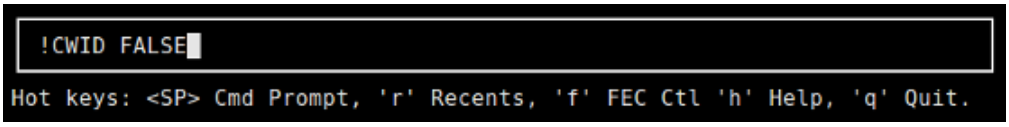
the *arim.ini* configuration file at the *server* station. *password* is the password, which may include any printable character, and has a maximum length of 32 characters. Note: *passwords are case sensitive!*
        - **`delpass client-call server-call`** Delete a password in the local password digest file. *client-call* is the call sign of the *client* station, and *server-call* is the call sign of the *server* station.
    - Housekeeping commands
        - **`clrmon`** Clear the Traffic Monitor view
        - **`clrheard`** Clear the Calls Heard view
        - **`clrrec`** Clear the Recent Messages view
        - **`clrping`** Clear the Ping History view
        - **`clrconn`** Clear the Connection History view
        - **`clrfile`** Clear the ARQ File History view
    - UI theme control
        - **`theme name`** Change the UI theme. *name* identifies the theme, which may be one of the built-in themes DARK and LIGHT, or a custom theme defined in the *arim-theme* file. A theme controls the color coding and text attributes of various UI elements. The effect of this command is immediate; the terminal window is erased and re-drawn using the specified theme. Note: theme names are *not* case sensitive.

Basic line editing features are available on the command prompt. The left and right arrow keys, HOME, END, DEL and backspace keys work as expected. This makes it easy to correct mistakes in typing, whether a entering a command or composing a message. For those familiar with the Emacs style line editing in Linux: CTRL-A, CTRL-E, CTRL-F, CTRL-B, CTRL-D, CTRL-K and CTRL-U can be used instead.

The command prompt has a history buffer which holds the last 15 unique commands entered. Use the UP and DOWN keys to browse command history when you want to reuse a previously sent command. For those familiar with Linux command history navigation, the CTRL-P and CTRL-N keys can be used instead.

Sending commands to the TNC

A TNC must be attached. Press the spacebar to open the command prompt.



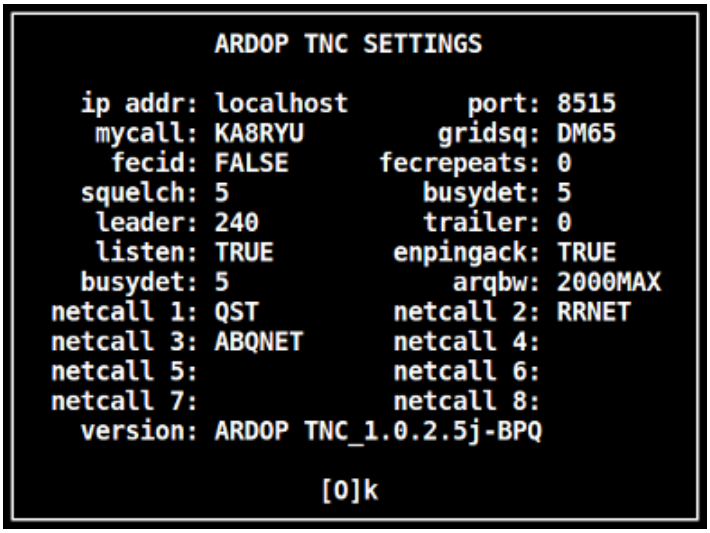
Prefix the message with the '!' character. Press ENTER to send the command to the TNC. The command trace will appear in the TNC command view, followed by the TNC's response if it was a query. To set a TNC parameter, follow the command with the parameter, e.g. **`!CWID TRUE`**. To query a TNC parameter, send the command without any argument, e.g. **`!CWID`**. Complete control over the TNC is available using this interface. ARDOP TNC commands are documented in the *Host Interface Spec for the ARDOP TNC* by Rick Muething, KN6KB and available in the files area of the ARDOP users group at groups.io.

Viewing and changing TNC settings

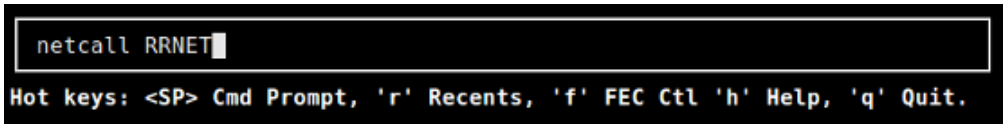
TNC parameters are set in the **`[tnc]`** section of the *arim.ini* configuration file and loaded when attaching to a TNC. However, some of these settings can be modified on-the-fly from the ARIM command prompt with these commands:

- **`mycall`** Set the TNC call sign. Useful for testing or changing SSID quickly. Example: `mycall NW8L-4`
- **`netcall add`** Add a net call sign to the list. Useful for testing or quick net changes. Example: `netcall add RRNET`
- **`netcall del`** Delete a net call sign from the list. Useful for testing or quick net changes. Example: `netcall del RRNET`
- **`gridsq`** Set the grid square locator, which is contained in ARIM beacon frames. Example: `gridsq DM65qf`
- **`pname`** Set the TNC name, which is advertised in ARIM beacon frames, and also returned in response to the ARIM 'pname' query. Example: `ARIM/ardop2`

- **`listen`** Control whether or not the TNC responds to ARQ connect requests or pings. Value must be TRUE or FALSE. Example: `listen TRUE`
- **`enpingack`** Control whether or not the TNC responds to pings from another station.
- **`tncset`** Open a pop-up dialog showing the TNC settings including IP address, port number, and other parameters. Example: `tncset`



To change one of these, press the spacebar to open the command prompt and enter the command like this:

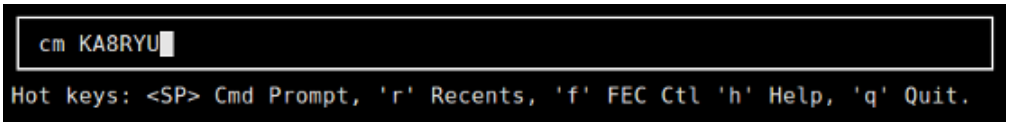


The change will be confirmed by a message on the Status Bar.

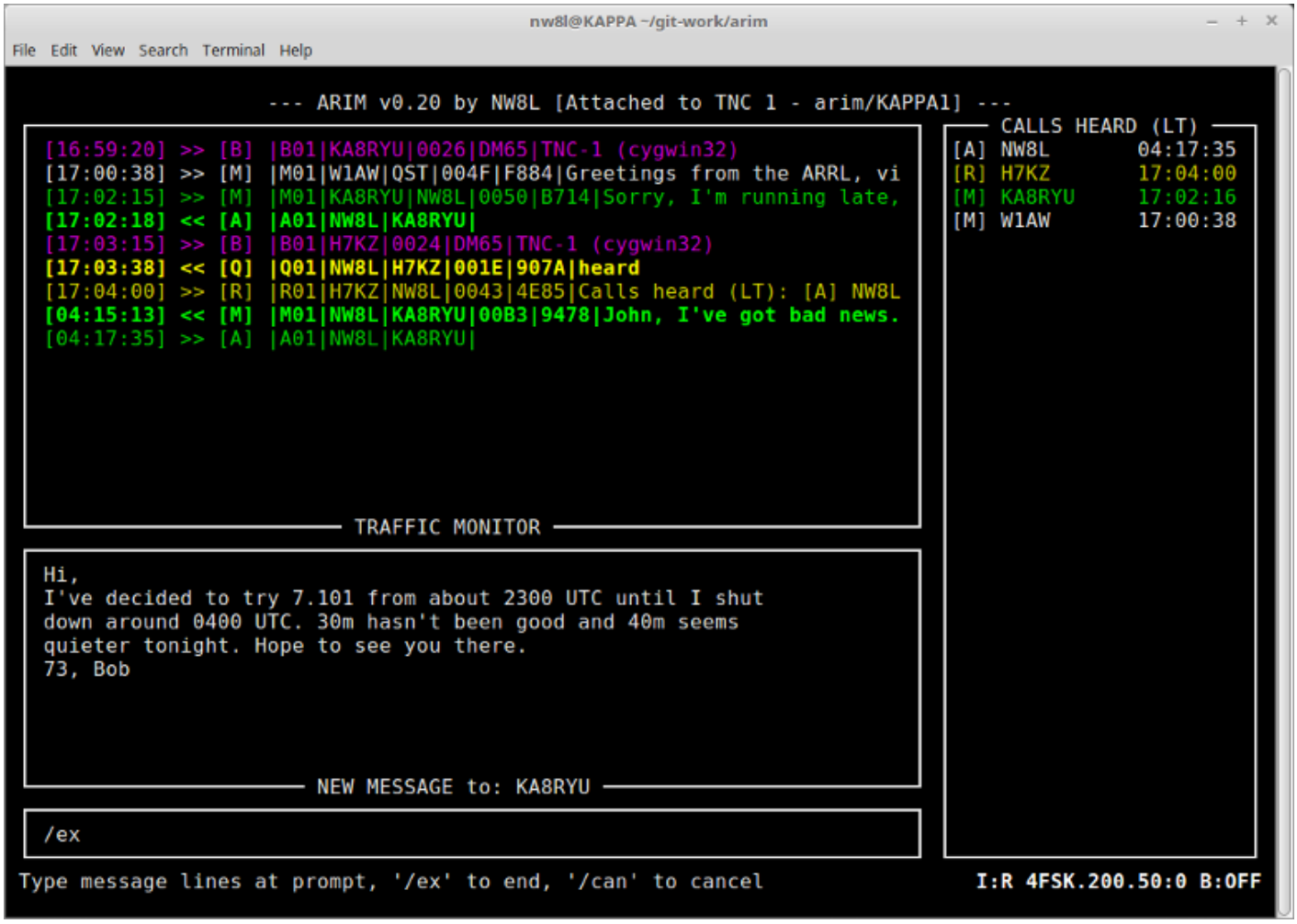
When attached to a TNC, any ARDOP parameter may be queried or changed at the command prompt using *bang commands*. These are prefixed with the '!' character and are sent directly to the TNC. For example, **`!SENDID`** queries the SENDID parameter, and **`!BUSYDET 5`** sets the BUSYDET parameter to '5'. ARDOP TNC commands are described in the *ARDOP_2 Protocol Native TNC Commands* document by KN6KB. The current version is located in the files section of the ARDOP Developers Group.

Composing a message (to outbox)

Whether or not attached to a TNC, messages can be composed and stored in the message outbox using the **`cm`** ***`call`*** command where *call* is the call sign of the recipient. Press the spacebar to open the command prompt and enter the command like this:



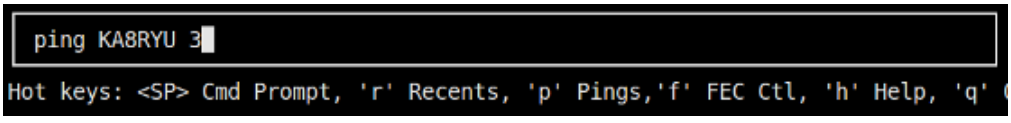
The message composer opens.



Enter the message line-by-line at the command prompt, then enter **`/ex`** at the start of a new line to close the message composer view and send the message to the outbox. To cancel a message enter **`/can`** at the start of a new line. This will discard the message and close the composer view.

Pinging another station

A TNC must be attached and the RF channel must not be *busy*. Press the spacebar to open the command prompt and enter the `ping` command like this:



The format is `ping` *`call nbr`*. The ping repeat count must follow the target call sign. This is the number of times the PING packet will be sent in the absence of a response from the target station before the TNC "gives up". The minimum repeat count is 2 and the maximum is 15. The response from the target station will include a signal report including the signal-to-noise ratio and a constellation quality figure ranging from 0-100%.

If the RF channel is busy, ARIM won't send the ping, and an error message will be shown on the status bar.

Using Pilot Pings

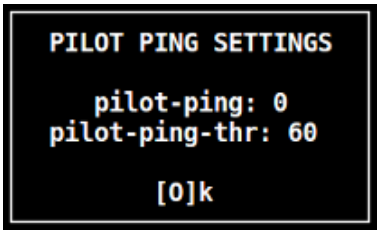
Pilot pings are ARDOP PING frames automatically sent in advance of an ARIM message or query transmission or ARQ connect request to test the RF path to the target station. If the signal constellation quality report from the target station meets a preset threshold, the message transmission proceeds; if not then it is canceled. This prevents hopeless attempts at message transmission from tying up the frequency in marginal conditions. The PING/PINGACK cycle is very short compared to the length of a typical FEC message transmission so the

additional overhead is low, but the potential improvement in efficiency of channel use is high. Pilot pings are never sent in advance of ARIM messages directed to the net call. These parameters control pilot ping behavior:

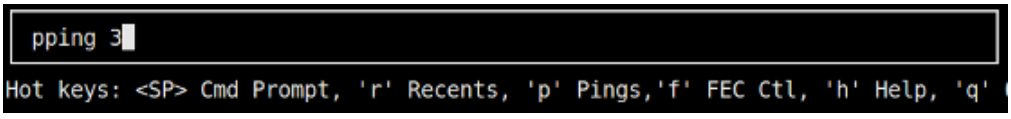
- **pilot-ping** The number of times a pilot ping will be repeated in the absence of a response from the recipient. It is recommended that this value not exceed 3 to prevent tying up the channel with repeats in poor conditions. Set to 0 to disable pilot pings; otherwise the range is 2-5. Default: 0.
- **pilot-ping-thr** When pilot pings are enabled, this is the threshold by which signal reports from the target station are judged. If the reported constellation quality is above the threshold, message (or query) transmission proceeds; if below this threshold it is canceled. It is recommended that this value be 60 or higher; choose a threshold suitable for the FEC mode in use. Min is 50, Max is 100. Default: 60.

These parameters are set in the **`[arim]`** section of the *arim.ini* configuration file, but can also be modified on-the-fly from the ARIM command prompt so that the operator an adapt to changing conditions. Here are the commands you can use from the command prompt:

- **`pping`** Set the number of times the pilot ping will be repeated in the absence of a response from the target station. Value must be in the range 2-5. Setting it to '0' disables pilot pings. Example: `pping 3`
- **`ppthr`** Set the pilot ping signal report quality threshold. Value must be in the range 60-100. Example: `ppthr 75`
- **`ppset`** Open a pop-up dialog showing the pilot ping settings. Example: `ppset`



To change one of these, press the spacebar to open the command prompt and enter the command like this:

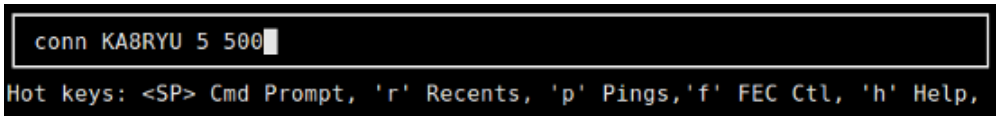


The change will be confirmed by a message on the Status Bar. When in doubt, use the `ppset` command to check settings.

ARQ: Connecting to a remote station

ARIM can act as an ARQ client or server, interoperating with ARDOP Chat or another ARIM station for keyboard to keyboard chat. ARIM may also be used as an ARQ client to connect to a BPQ BBS with a ARDOP port.

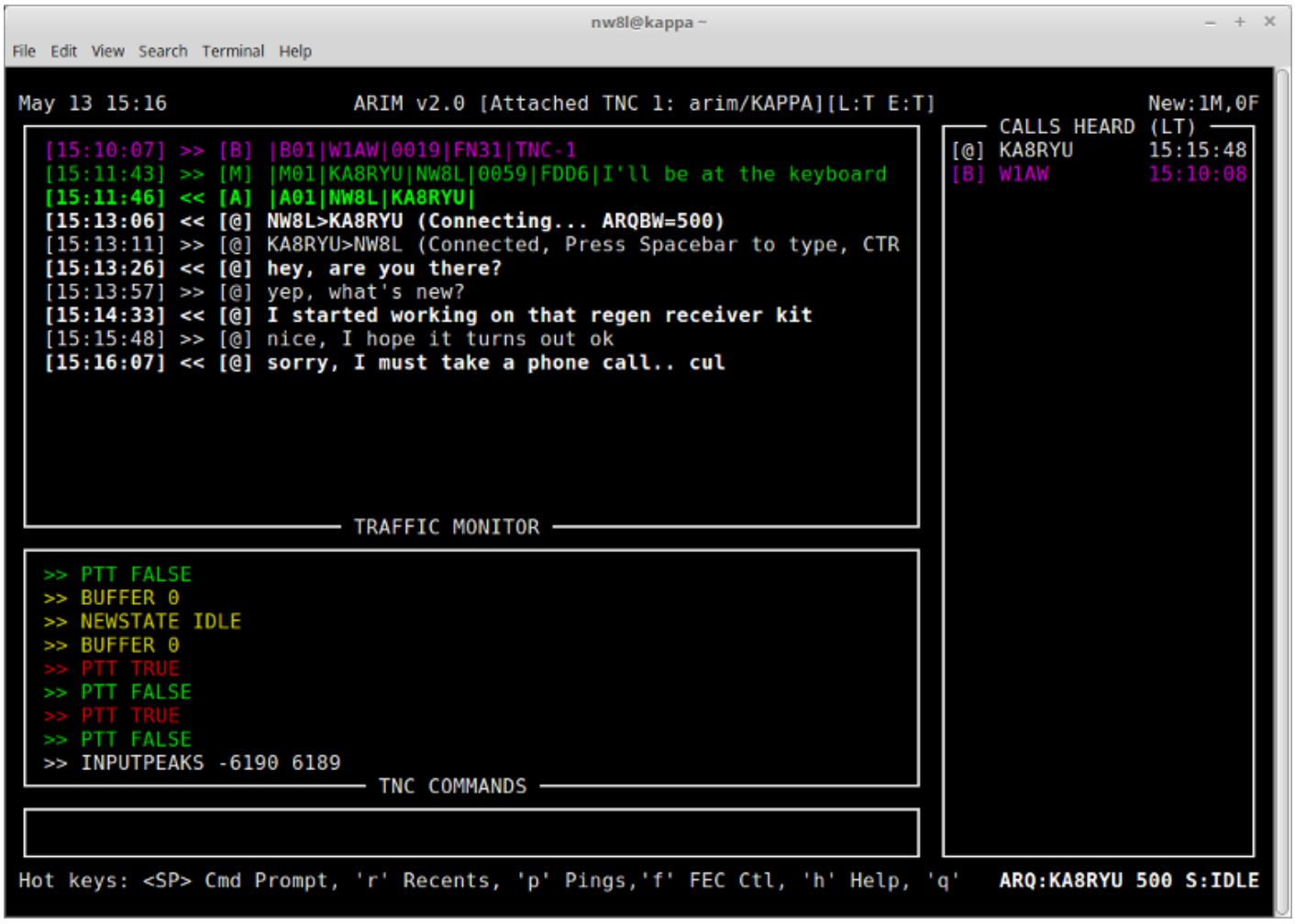
A TNC must be attached and the RF channel must not be *busy*. Press the spacebar to open the command prompt and enter the `conn` command like this:



The format is `conn` *`call nbr [bw]`* where *call* is the call sign of the target station, and *nbr* is the request repeat count. This is the number of times the ARQCALL packet will be sent in the absence of a response from the target station before the TNC "gives up". The minimum repeat count is 3 and the maximum is 15. *bw* is an optional connection bandwidth (ARQBW) specifier. It can be one of *200*, *500*, *2500* or *any*. If *any*, ARIM will attempt to connect using each ARQBW setting in succession, starting with the current ARQBW setting (set in the ARIM configuration file or by using the `arqbw` command. This is useful if the ARQBW setting of the remote station is unknown. Connect requests and responses are displayed in the monitor view.

ARDOP 2 TNCs reject connection requests if the ARQBW setting of the calling station exceeds that of the remote station. For this reason, the ARIM 2.x `conn` command optionally lets the operator specify an ARQ connection

bandwidth to override the local ARQBW setting and allow the connection. Alternatively, the operator can use the `any` specifier to make ARIM discover a workable ARQBW setting automatically (i.e. negotiating the connection bandwidth itself). In this example the proper ARQBW is known to be **500**.



If the RF channel is busy, ARIM won't send the connection request, and an error message will be shown on the status bar.

When the connection is made, ARIM will be in ARQ mode, where text entered at the command prompt is sent directly to the remote station rather than processed as an ARIM command. The line editing and history recall features of the command prompt work as usual but the '!' prefix won't work to send commands directly to the TNC, and the ':' prefix won't work to send unproto FEC transmissions.

All text sent and received will be displayed on the Traffic Monitor, and the status bar will show the call sign of the remote station, the negotiated max connection bandwidth and the current TNC state. Note: the Recent Messages and Ping History views can't be opened during an ARQ session.

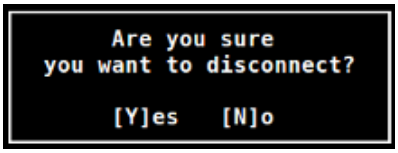
The shared files, message inbox, message outbox and sent messages viewers can be opened during an ARQ session by using these hot keys:

- **`'f'`** Open the shared files viewer.
- **`'i'`** Open the message inbox viewer.
- **`'o'`** Open the message outbox viewer.
- **`'s'`** Open the sent messages viewer.

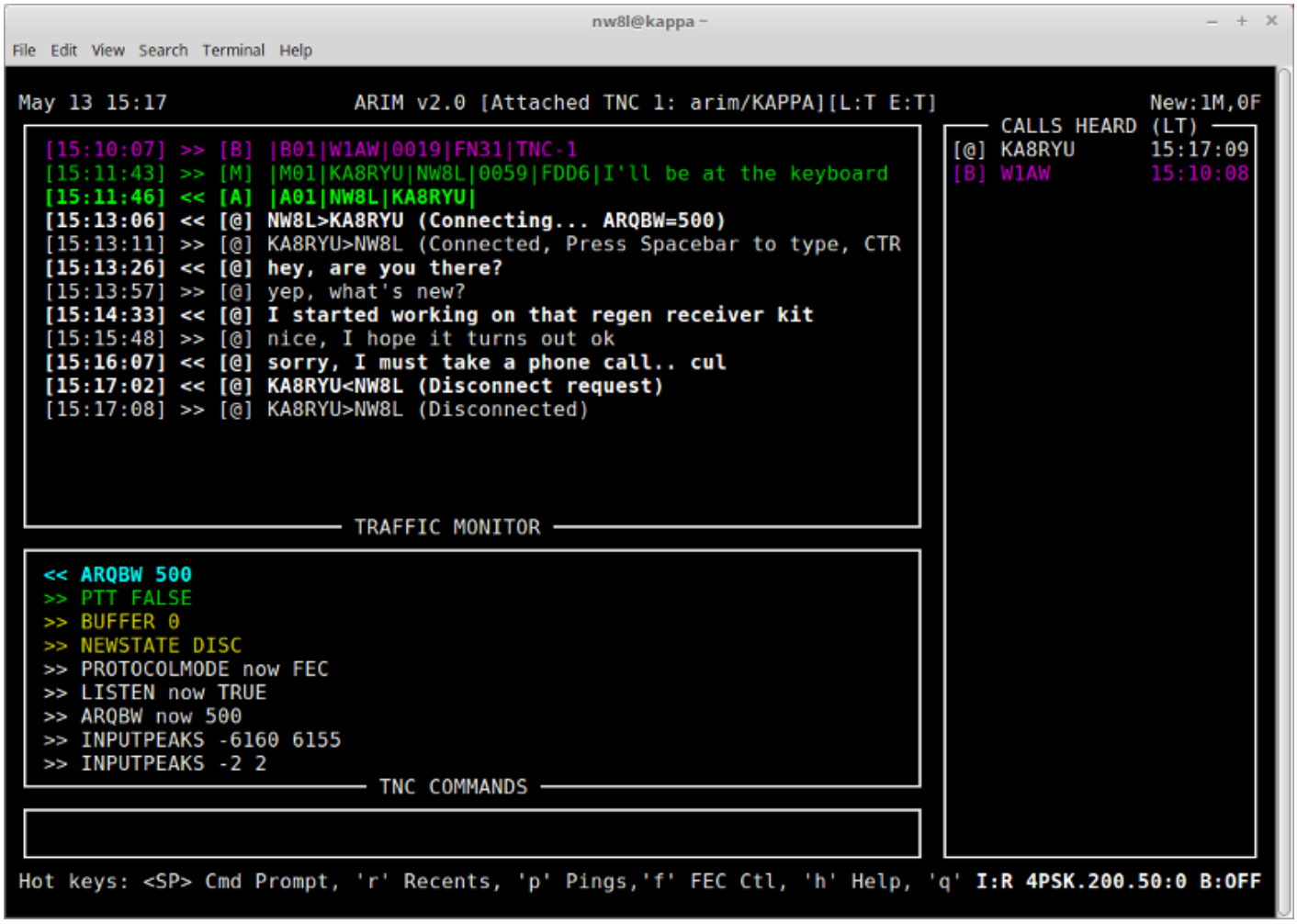
These work as usual in ARQ mode. Files can be uploaded to the remote station from the shared files viewer with the **`sf [-z] n [dir]`** command. Previously composed messages can be sent to the remote station from the message outbox viewer with the **`sm n`** command. Messages can be forwarded to the remote station from the message inbox and sent messages viewers with the **`fm n`** commands.

To disconnect, press CTRL-X or enter the special command '/dis' at the command prompt and press ENTER.

You'll be prompted to confirm:



Type 'y' and a clean disconnect will be attempted.



ARQ behavior is controlled by these parameters in the **[tnc]** section of the *arim.ini* configuration file:

- **listen** Control whether or not the TNC responds to ARQ connect requests or pings. Default: TRUE.
- **arq-bandwidth** Set the bandwidth for ARQ connections. The value must be one of: 200, 500 or 2500. Default is 500.
- **arq-negotiate-bw** Controls whether or not the TNC will negotiate ARQ bandwidth for incoming connections. Default: TRUE. *This feature currently supported by ARDOP_2Win TNC version 2.0.4 only.*
- **arq-timeout** Set the inactivity timeout for ARQ connections, in seconds. The value must be in the range 30 to 600. Default is 120.
- **arq-sendcr** Control whether or not CRLF line endings are sent in ARQ mode, instead of Unix style LF endings. Default: TRUE.

These parameters can also be modified on-the-fly from the ARIM command prompt by the operator. The following commands are available:

- **`listen`** Control whether or not the TNC responds to ARQ connect requests or pings. Value must be TRUE or FALSE. Example: `listen TRUE`
- **`arqto`** Set the ARQ inactivity timeout in seconds. Value must be in the range 30-600. Example: `arqto 120`
- **`arqbw`** Set the ARQ connection bandwidth. The bandwidth must be specified as one of: 200, 500 or 500. Example: `arqbw 500`
- **`arqnegbw`** Controls whether or not the TNC will negotiate ARQ bandwidth for incoming connections. Value must be TRUE or FALSE. Example: `arqnegbw TRUE` *This feature currently supported by ARDOP_2Win TNC version 2.0.4 only.*
- **`arqset`** Open a pop-up dialog showing the ARQ settings. Example: `arqset`

If pilot pings are enabled they will apply to ARQ connection requests, so that the RF path to the remote station can be tested in advance. If the pilot pings fail, the connection attempt will be canceled.

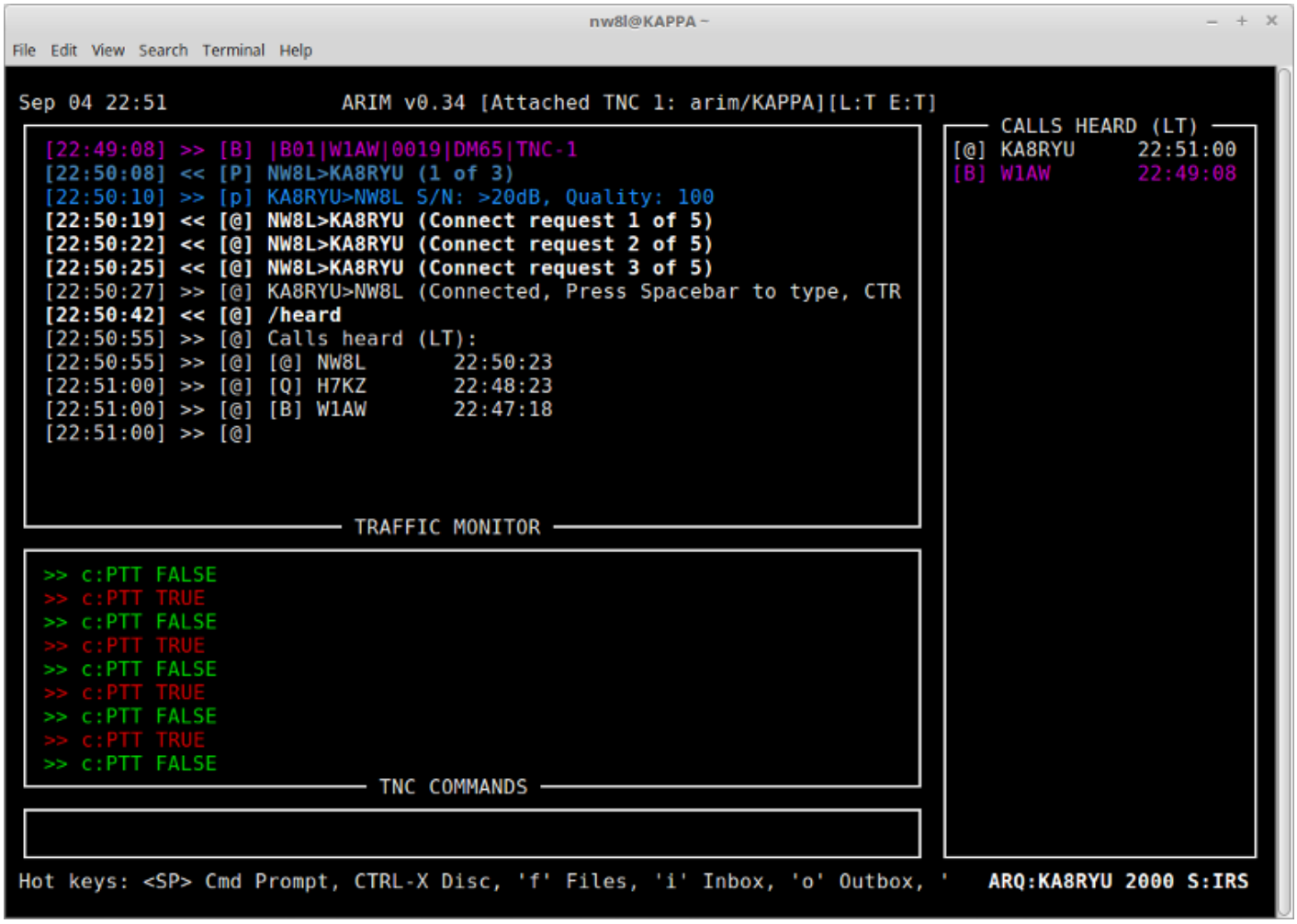
ARQ: Sending queries to the remote station

When connected to an ARIM station, the following query commands, prefixed by the '/' (slash) character, may be entered at the prompt to retrieve information from the remote station:

- **`/version`** Return the software version numbers for ARIM and the ARDOP TNC.
- **`/gridsq`** Return the Maidenhead locator (gridsquare).
- **`/info`** Return the info statement.
- **`/pname`** Return the port name for the TNC.
- **`/heard`** Return the calls heard list.
- **`/netcalls`** Return the net call signs list.
- **`/flist [dir]`** Return a list of shared files and directories. *dir* is an optional directory path, relative to the shared files root directory, e.g. `contests/2016/FOBB`. If *dir* is not given, then the the shared files root directory is listed; otherwise the specified directory is listed. By default only files in the shared files root directory and dynamic files are listed. Subdirectories are not listed unless they are made visible by an *add-files-dir* configuration file parameter. Starting with ARIM version 1.2, access-controlled subdirectories defined by the *ac-files-dir* configuration file parameter are indicated with the **!** (bang) character like this: `!DIR`.
- **`/file fn`**, Print a file to the traffic monitor view. *fn* is the name of a file in the shared files root directory at the remote station, or a file path relative to it, e.g. `contests/2016/FOBB.log`.

Press the spacebar to open the command prompt and enter the command at the start of the line like this:

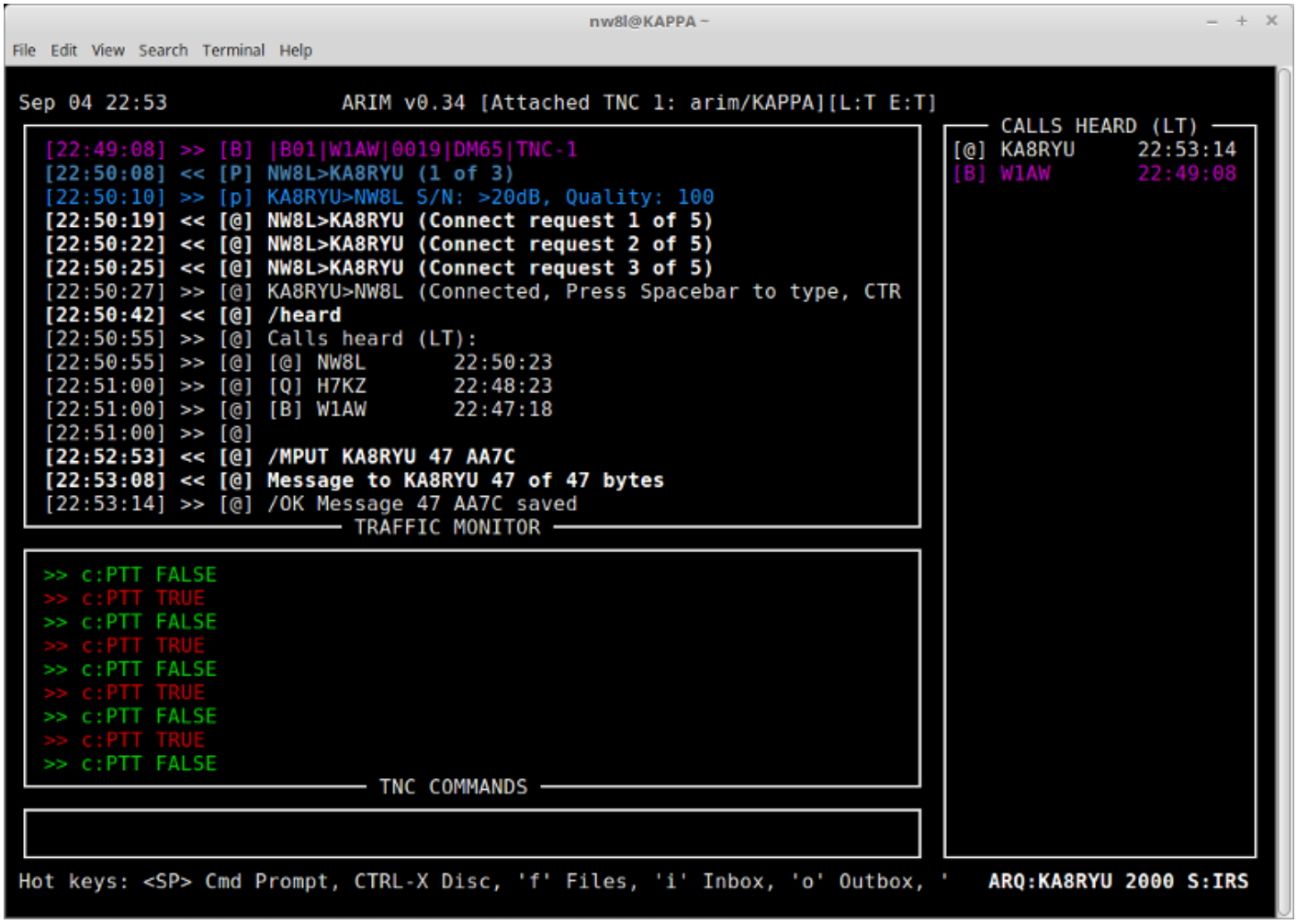
Here we see the response:



ARQ: Sending messages to the remote station

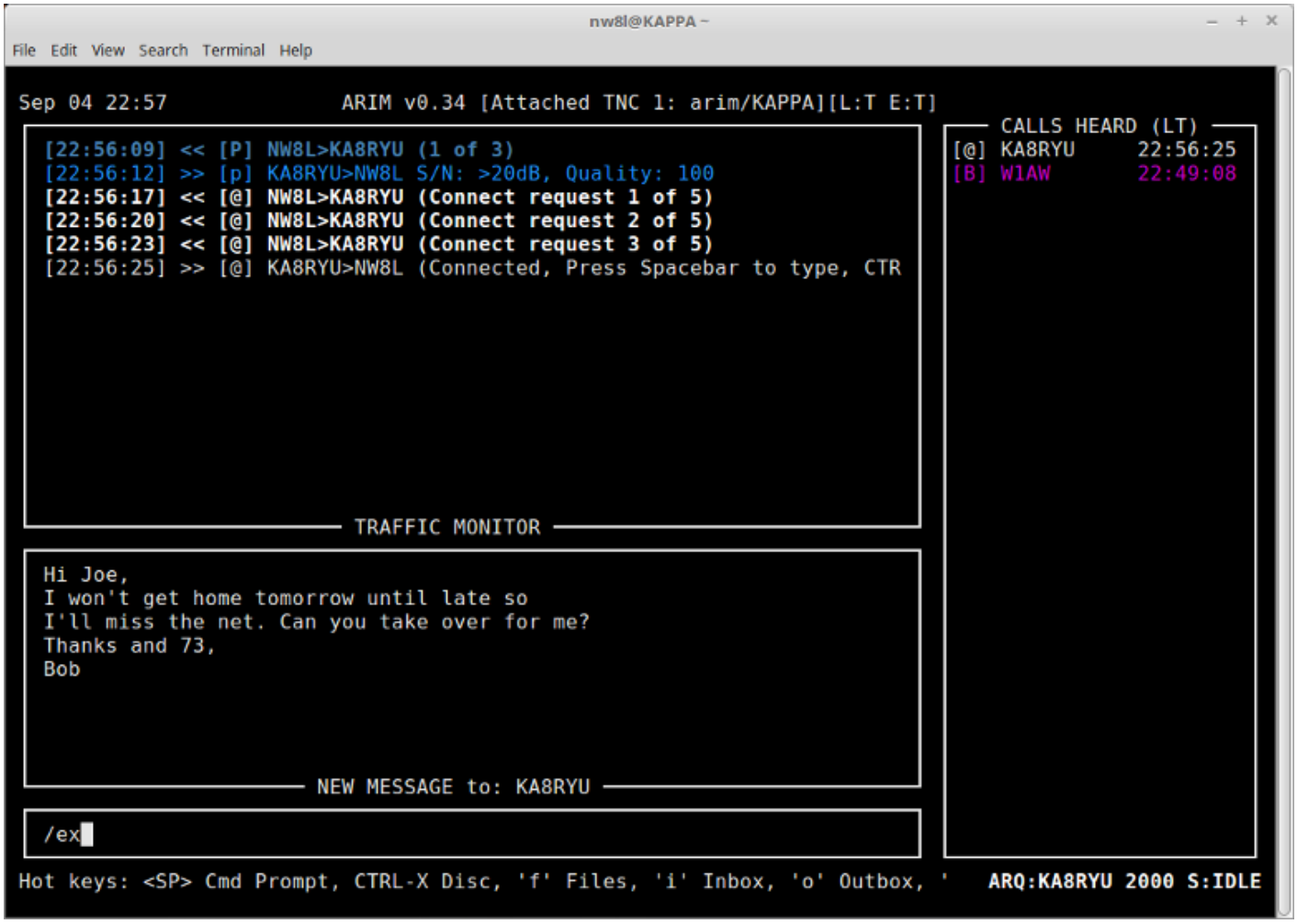
When connected to an ARIM station, single-line messages can be sent using the **`/sm [-z][msg]`** command where the *-z* option invokes compression, and *msg* is the message text. This is convenient when the message is short and you wish to bypass the message composer. Press the spacebar to open the command prompt and enter the command like this:

The remote station saves the message to its inbox and returns an `/OK` response to confirm receipt. A copy will be stored in your sent messages mailbox.

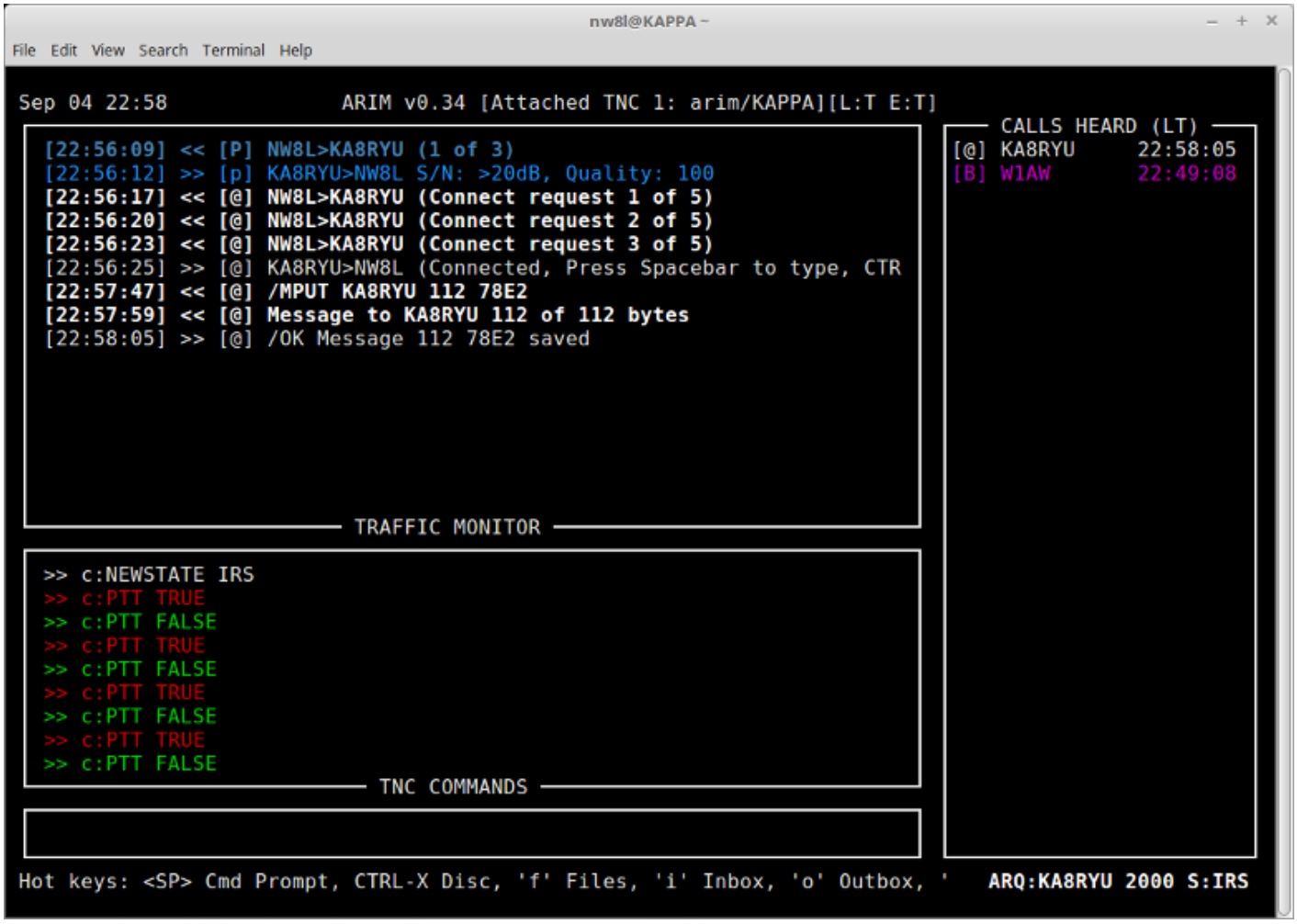


For longer, multi-line messages, press the spacebar to open the command prompt and enter the command like this:

Enter the message line-by-line at the command prompt. When done, enter **`/ex`** at the start of a new line to close the message composer view and send the message on its way.



The remote station saves the message to its inbox and returns an `/OK` response to confirm receipt. A copy will be stored in your sent messages mailbox.



To cancel a message you've started to send, enter **`/can`** at the start of a new line to close the message composer view and discard the message.

While the upload is in progress no text or commands are accepted from the command prompt. This condition is indicated by a **`!`** character seen in the status bar indicator area:

```
ARIM Busy: ARQ message upload in progress                    ! ARQ:KA8RYU 500 S:ISS
[##################################                ][  Upload: 63%]
```

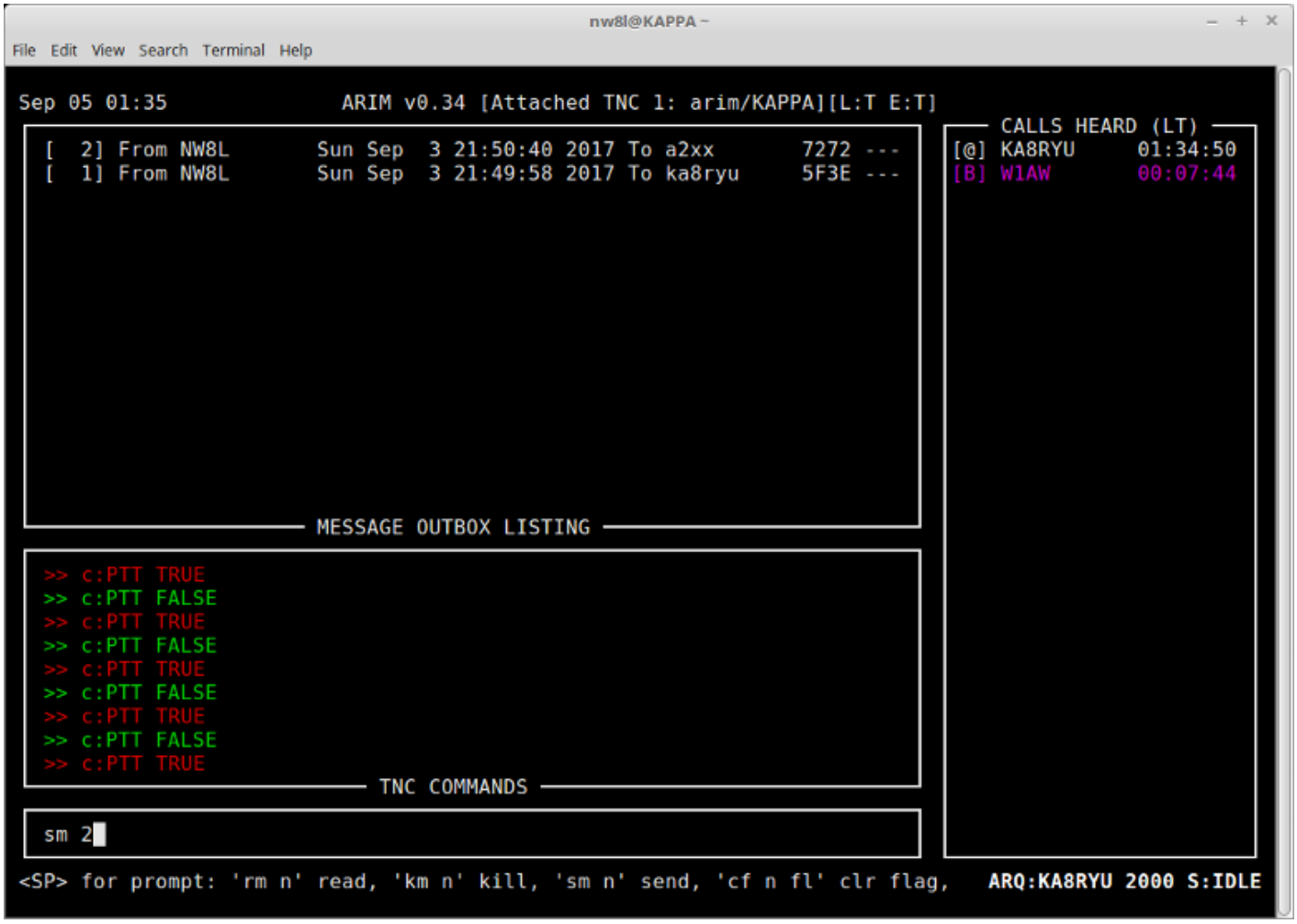
If the remote ARIM station encounters an error saving the message, it returns an `/ERROR` response and the message is discarded.

Use the **`-z`** option to compress the message to reduce transfer time. The message will be automatically decompressed at the receiving station before it is saved to the inbox there.

Existing messages in a local mailbox can also be sent to the remote station from one of the message viewers. During an ARQ session these are opened not by entering a commmand at the prompt, but by pressing a hot key:

- **`'i'`** Open the message inbox viewer.
- **`'o'`** Open the message outbox viewer.
- **`'s'`** Open the sent messages viewer.

Previously composed messages stored in the outbox can be sent to the remote station. Press 'o' to open the message outbox viewer:



Enter the **`sm n`** command to start the upload. This is a more efficient method than composing the messages during the ARQ session itself. Alternatively, use the **`sm -z n`** command to compress the message before it is sent. This can reduce transfer time for larger messages.

Messages in the inbox can be forwarded to the remote station. Press 'i' to open the message inbox viewer and enter the **`fm n`** command to start the upload. Alternatively, use the **`fm -z n`** command to compress the message before it is sent. This can reduce transfer time for larger messages.

Messages in the sent messages mailbox can also be forwarded to the remote station. Press 's' to open the sent messages viewer and enter the **`fm n`** command to start the upload. Alternatively, use the **`fm -z n`** command to compress the message before it is sent. This can reduce transfer time for larger messages.

If the remote ARIM station encounters an error saving the message, it returns an `/ERROR` response and the message is discarded.

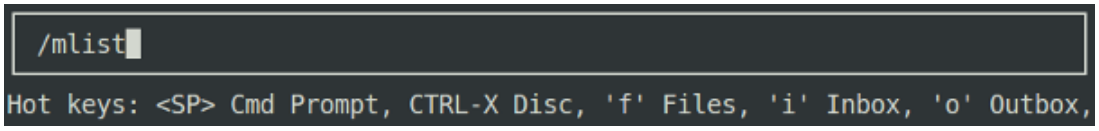
ARQ: Downloading messages from the remote station

In an authenticated ARQ session, you can retrieve messages addressed to your station that are queued in the remote station's outbox. This is useful if you are on the air only intermittently; the operator at the remote station can compose a message when convenient, leaving it in the outbox for you to download later. Use these commands to list and download messages:

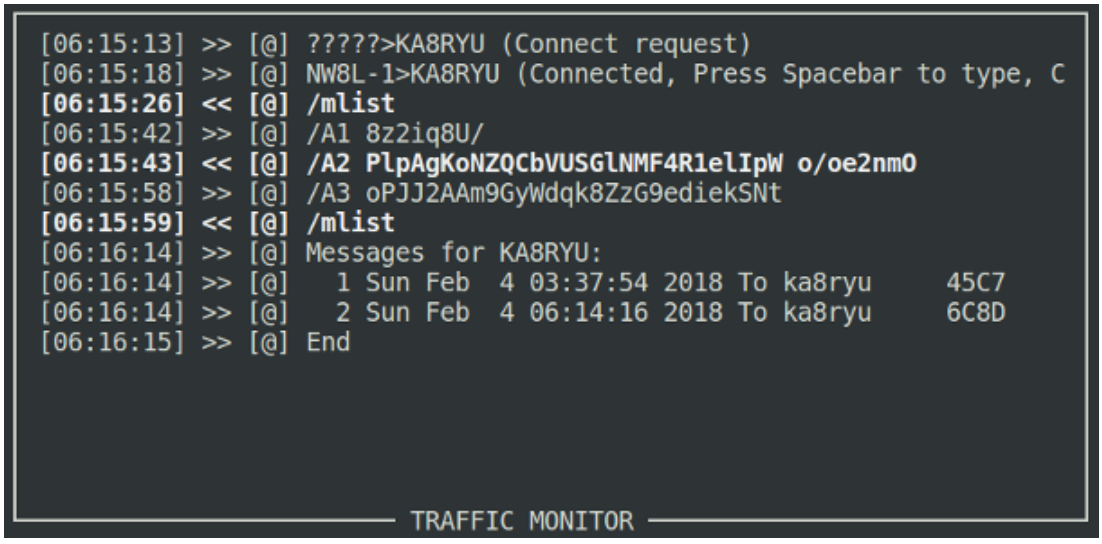
- **`/mlist`** Return a list of messages in the remote station's outbox that are addressed to your station's call sign. *Requires that your station be authenticated with the remote station.* If not, then the remote station will respond with an authentication challenge.
- **`/mget [-z] [nbr]`** Download messages addressed to the your station's call sign, *from* the remote station's outbox *to* your station's inbox. The *-z* option compresses messages to minimize transfer time. *nbr* optionally specifies the maximum number of messages to download. The default is 10. *Requires that your station be*

authenticated with the remote station. If not, then the remote station will respond with an authentication challenge.

To list your messages, press the spacebar to open the command prompt and enter the **`/mlist`** command at the start of the line like this:

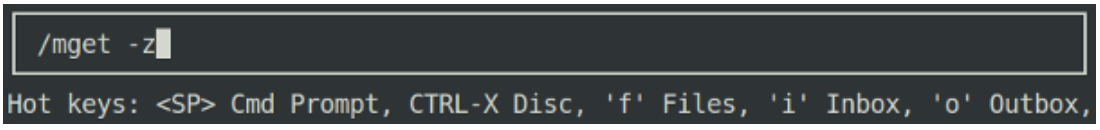


The remote station responds:

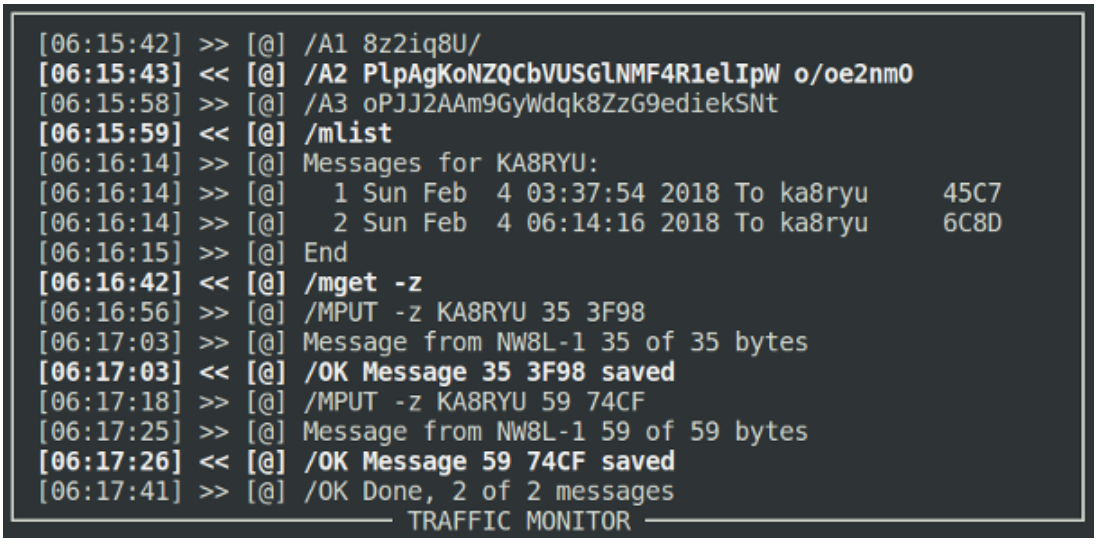


In this example, the command triggers the mutual authentication process automatically. After that, the listing is received. The messages are listed in chronological order (oldest first).

To download the listed messages, press the spacebar to open the command prompt and enter the **`/mget [-z] [nbr]`** command at the start of the line like this:



In this example, the the **`-z`** option is used to invoke compression. The download will begin:



The progress of the download is reported as each message is received. As each message is received, the receiving station stores the message in its inbox and sends a `/OK` response to confirm receipt. The sending station then deletes the message from its outbox. If the either station encounters an error, it sends an `/ERROR` response to the other and the download is canceled. In that case the message is *not* deleted from the outbox at the sending station.

While the download is in progress no text or commands are accepted from the command prompt. This condition is indicated by a `!` character seen in the status bar indicator area:

To cancel the download, press CTRL-X to close the ARQ connection.

Use the **`/mget -z`** command to compress the message before it is sent. This can reduce transfer time for large messages.

Use the **`/mget nbr`** command to limit the number of messages downloaded. If less than the number of messages available, then the oldest *nbr* are downloaded. The minimum value of *nbr* is 1 and the maximum is 10. This option can be combined with the *-z* option.

ARQ: Listing shared files on the remote station

When connected to an ARIM station a listing of any shared files it offers can be viewed using one of these commands:

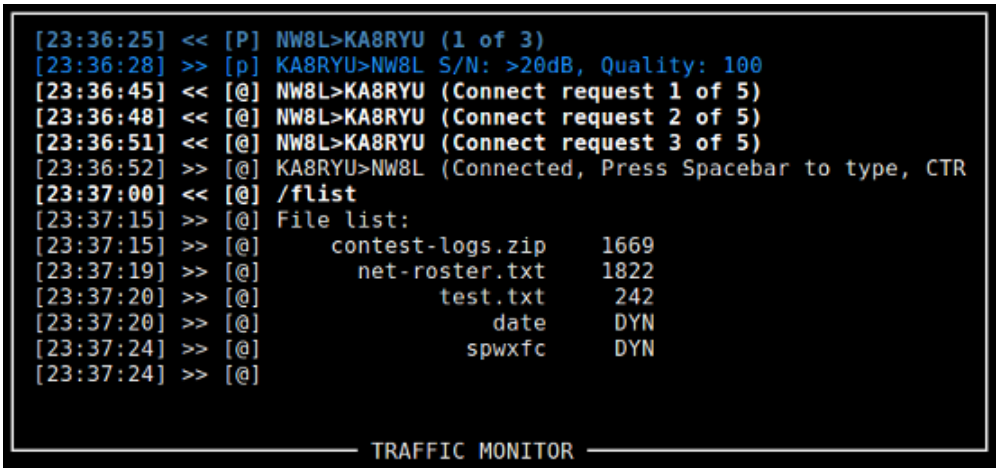
- **`/flist [dir]`** Return a list of shared files and directories. *dir* is an optional directory path, relative to the shared files root directory, e.g. `contests/2016/FOBB`. If *dir* is not given, then the the shared files root directory is listed; otherwise the specified directory is listed.
- **`/flget [-z] [dir]`** Downloads a list of shared files and directories, then displays it in the remote shared files viewer. The *-z* option compresses the file to minimize transfer time. *dir* is an optional directory path, relative to the shared files root directory, e.g. `contests/2016/FOBB`. If *dir* is not given, then the the shared files root directory is listed; otherwise the specified directory is listed. The viewer shows a numbered list of files and directories, making it easy to read or download files without typing in lengthy file names.

By default, only files in the shared files root directory and dynamic files are listed. Subdirectories are not listed unless they are made visible by an *add-files-dir* configuration file parameter. Access-controlled subdirectories defined by the *ac-files-dir* configuration file parameter are indicated with the **!** (bang) character like this: `!DIR`.

Examples:

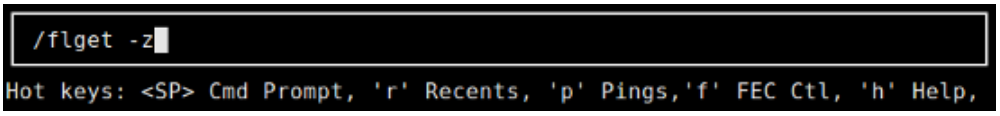
- In this example `/flist` is used to list the files in the shared files root directory. Press the spacebar to open the command prompt and enter the **`/flist`** command at the start of the line like this:

  The remote station responds:



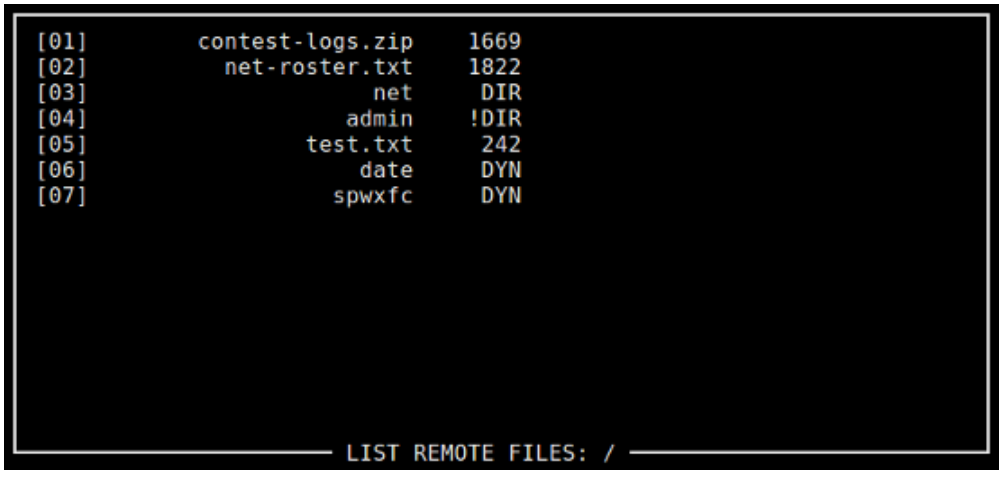
The contents of the directory are listed, one to a line. For files, the name and size in bytes are shown. For subdirectories, the name is shown. Access-controlled subdirectories are indicated with the **!** (bang) character like this: `!DIR`.

- In this example `/flget` is used to list the files in the shared files root directory, with the compression option enabled. Press the spacebar to open the command prompt and enter the **`/flget -z`** command at the start of the line like this:



The file listing is downloaded. While the download is in progress no text or commands are accepted from the command prompt. This condition is indicated by a `!` character seen in the status bar indicator area:

The remote shared files viewer opens:



The remote directory path is shown in the title of the viewer window.

The contents of the directory are listed, one to a line. For files, the name and size in bytes are shown. For subdirectories, the name is shown. Access-controlled subdirectories are indicated with the **!** (bang) character like this: `!DIR`.

Use the UP and DOWN arrow keys or the PAGEUP, PAGEDOWN, HOME and END keys to scroll through the listing. Each file or directory is given a number for use with commands. A list of available commands is printed in the Status Bar:

- **`rf nbr`** Read file number *nbr* in the file viewer.
- **`gf [-z] nbr [dir]`** Download file number *nbr*. Use the *-z* switch to enable compression to reduce transfer time. The optional *dir* parameter specifies the destination directory at the local station, relative to the shared files root directory, e.g. contests/2016.
- **`cd [-z] nbr`** Download a file listing for directory *nbr* in the shared files viewer and display it in the remote shared files viewer. Use the *-z* switch to enable compression to reduce transfer time.

Press 'q' to quit the viewer.

The remote shared files viewer automatically closes when a command is invoked. However, *the file listing is not lost*; press the hot key '#' to reopen the viewer if you need to download another file from the directory.

The mutual authentication process will be triggered automatically if `/flist` or `/flget` are used to list access-controlled directories when the session is not yet authenticated.

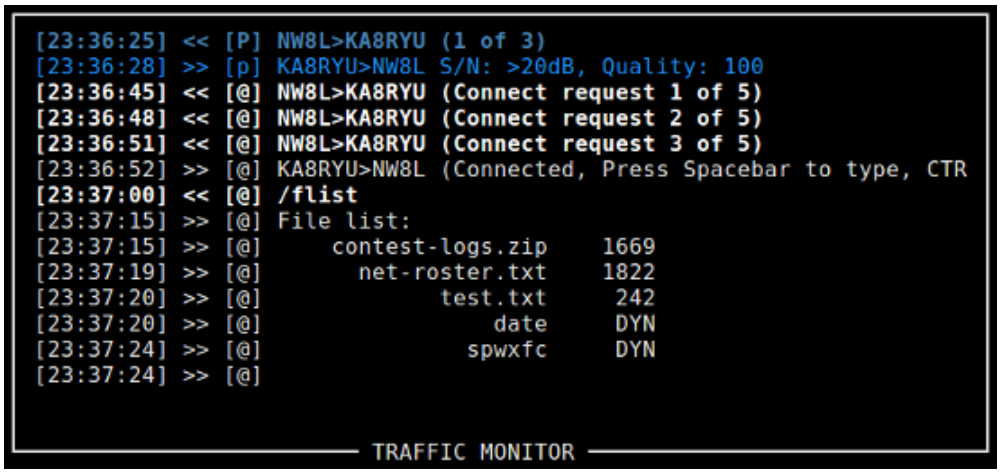
ARQ: Reading files from the remote station

When connected to an ARIM station any shared files it offers can be listed and read in the Traffic Monitor view. Only text files and dynamic files with text output can be printed to the screen. Use these commands to list available files and read them:

- **`/flist [dir]`** Return a list of shared files and directories. *dir* is an optional directory path, relative to the shared files root directory, e.g. `contests/2016/FOBB`. If *dir* is not given, then the the shared files root directory is listed; otherwise the specified directory is listed. By default only files in the shared files root directory and dynamic files are listed. Subdirectories are not listed unless they are made visible by an *add-files-dir* configuration file parameter. Access-controlled subdirectories defined by the *ac-files-dir* configuration file parameter are indicated with the **!** (bang) character like this: `!DIR`.
- **`/flget [-z] [dir]`** Downloads a list of shared files and directories, then displays it in the remote shared files viewer. The *-z* option compresses the file to minimize transfer time. *dir* is an optional directory path, relative to the shared files root directory, e.g. `contests/2016/FOBB`. If *dir* is not given, then the the shared files root directory is listed; otherwise the specified directory is listed. The viewer shows a numbered list of files and directories, making it easy to read or download files without typing in lengthy file names. Learn more about using `/flget` here.
- **`/file fn`**, Print a file to the traffic monitor view. *fn* is the name of a file in the shared files directory, or a file path relative to it, e.g. `contests/2016/FOBB.log`

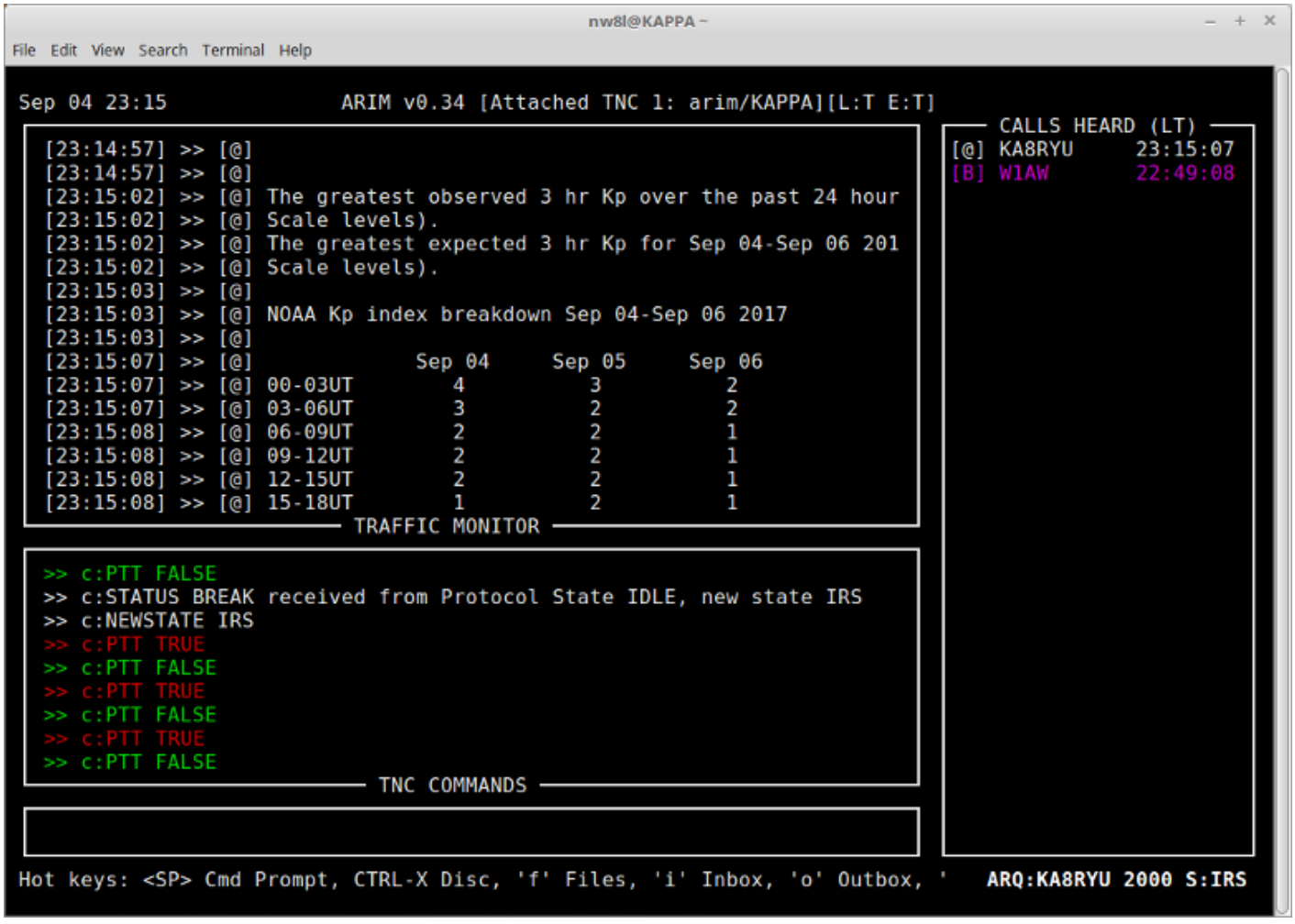
In this example `/flist` is used to list the files in the shared files root directory. Press the spacebar to open the command prompt and enter the **`/flist`** command at the start of the line like this:

The remote station responds:



Choose a file, press the spacebar to open the command prompt and enter the **`/file fn`** command at the start of the line like this:

The file is read and streamed to the local station:



The text is printed the the Traffic Monitor view line by line as it is received.

ARQ: Downloading files from the remote station

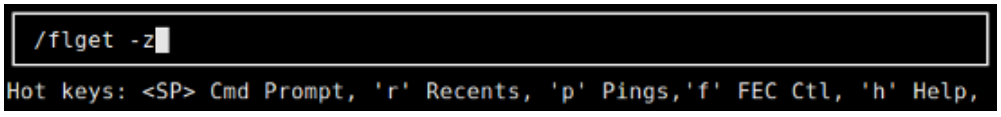
When connected to an ARIM station any shared files it offers can be listed and downloaded. Both text and binary file types can be transferred over the ARQ connection. A compression option can be invoked to reduce transfer time for many file types, and you can specify a destination directory at the local station other than the default. Use these commands to list and download files:

- **`/flist [dir]`** Return a list of shared files and directories. *dir* is an optional directory path, relative to the shared files root directory, e.g. `contests/2016/FOBB`. If *dir* is not given, then the the shared files root directory is listed; otherwise the specified directory is listed. By default only files in the shared files root directory and dynamic files are listed. Subdirectories are not listed unless they are made visible by an *add-files-dir* configuration file parameter at the remote station. Access-controlled subdirectories defined by the *ac-files-dir* configuration file parameter are indicated with the **!** (bang) character like this: `!DIR`.
- **`/flget [-z] [dir]`** Downloads a list of shared files and directories, then displays it in the remote shared files viewer. The *-z* option compresses the file to minimize transfer time. *dir* is an optional directory path, relative to the shared files root directory, e.g. `contests/2016/FOBB`. If *dir* is not given, then the the shared files

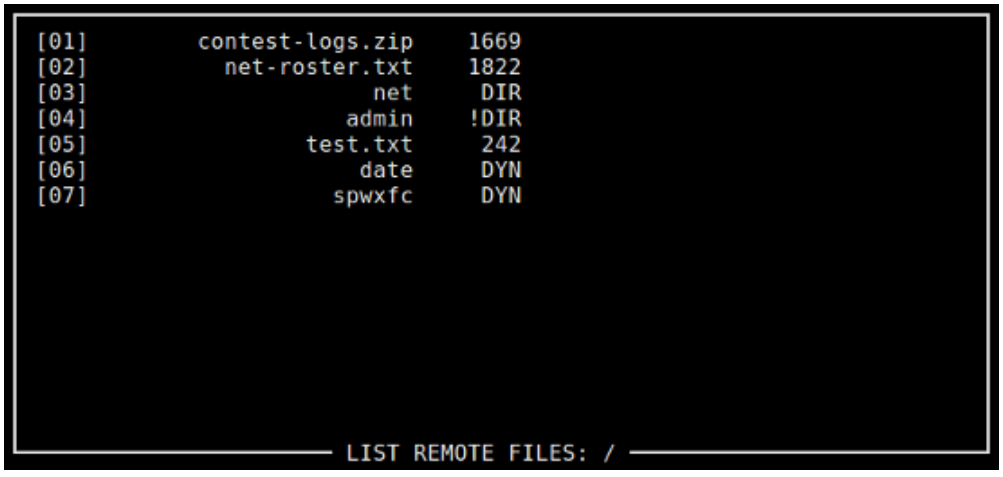
root directory is listed; otherwise the specified directory is listed. The viewer lets you read or download files without typing in lengthy file names. Learn more about using `/flget` here.

- **`/fget [-z] fn [> dir]`** Download a file to the local station. The *-z* option compresses the file to minimize transfer time. *fn* is the name of a file in the remote station's shared files directory, or a file path relative to it, e.g. `contests/2016/FOBB.log`. The optional *dir* parameter specifies the destination directory at the local station, relative the the shared files root directory, e.g. `contests/2016`. The destination directory must made visible by an *add-files-dir* configuration file parameter at the local station. If *dir* is not specified, the file is stored in the *download* subdirectory in the local station's shared files root directory. ARIM will create the destination directory if it doesn't already exist.

In this example `/flget` is used to list the files in the shared files root directory, with the compression option enabled. Press the spacebar to open the command prompt and enter the **`/flget -z`** command at the start of the line like this:

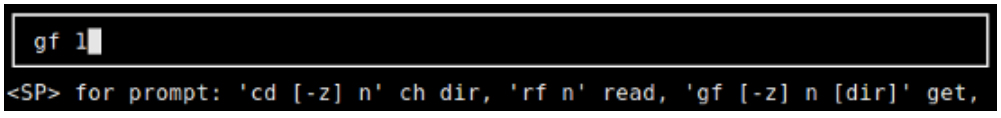


The file listing is downloaded and the remote shared files viewer opens:

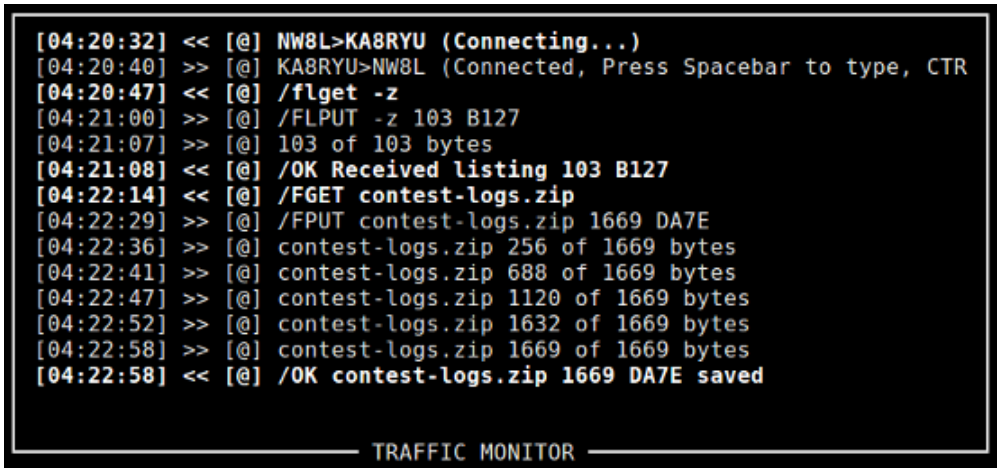


The directory path is shown in the title of the viewer window. The contents of the directory are listed, one to a line. For files, the name and size in bytes are shown. For subdirectories, the name is shown. Access-controlled subdirectories are indicated with the **!** (bang) character like this: `!DIR`.

In this example we'll download file #1, *contest-logs.zip*, which is already compressed. Press the spacebar to open the command prompt and enter the **`gf 1`** command at the start of the line like this:



The viewer closes and the download begins:



The progress of the download is reported as each data frame is received. When it's complete an `/OK` response is sent to the remote station to confirm receipt. If the remote ARIM station encounters an error sending the file, it returns an `/ERROR` response and the download is canceled.

By default, downloaded files are stored in the *download* subdirectory in the local station's shared files root directory. ARIM will create this if it doesn't already exist.

While the download is in progress no text or commands are accepted from the command prompt. This condition is indicated by a `!` character seen in the status bar indicator area:

To cancel the download, press CTRL-X to close the ARQ connection.

Use the **gf *-z nbr*** command to compress the file before it is sent. This can reduce transfer time for larger files. Note that compression isn't useful for very small files, or files that are already compressed (e.g. "zip" files or many image file types). In these cases the zlib format overhead can make the file size increase, not decrease.

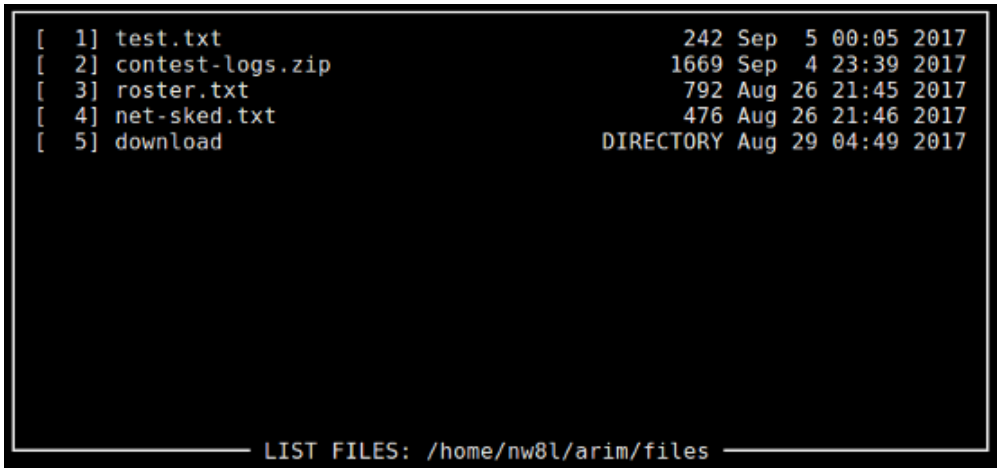
Use the **gf *[-z] nbr dir*** command to store the file to a location other than the default download directory. The *dir* parameter specifies the destination directory as a path relative to the shared files root directory at the local station, e.g. *contests/2016*. This option can be combined with the *-z* option.

The remote shared files viewer automatically closes when a command is invoked. However, *the file listing is not lost*; press the hot key '#' to reopen the viewer if you need to download another file from the directory.

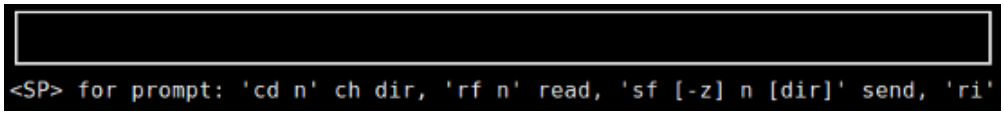
ARQ: Uploading files to the remote station

When connected to an ARIM station local files can be uploaded to the remote station. Both text and binary file types can be transferred over the ARQ connection. A compression option can be invoked to reduce transfer time for many file types, and you can specify a destination directory at the remote station other than the default. If you do, the destination directory must made visible by an *add-files-dir* configuration file parameter at the remote station. If not specified, the file is stored in the *download* subdirectory in the remote station's shared files root directory.

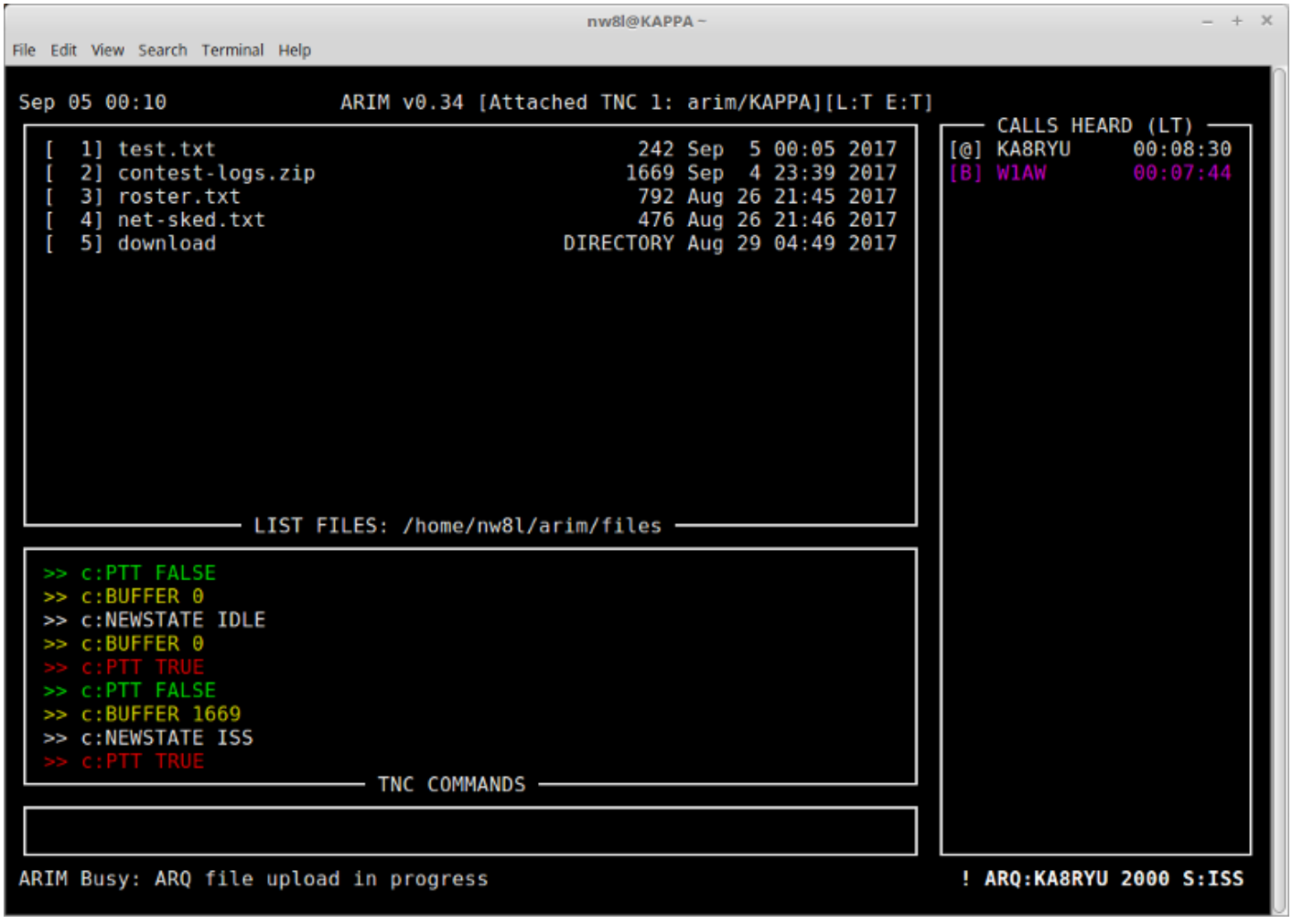
Press the 'f' hot key in ARQ mode to open the shared files viewer and locate the file you wish to send.



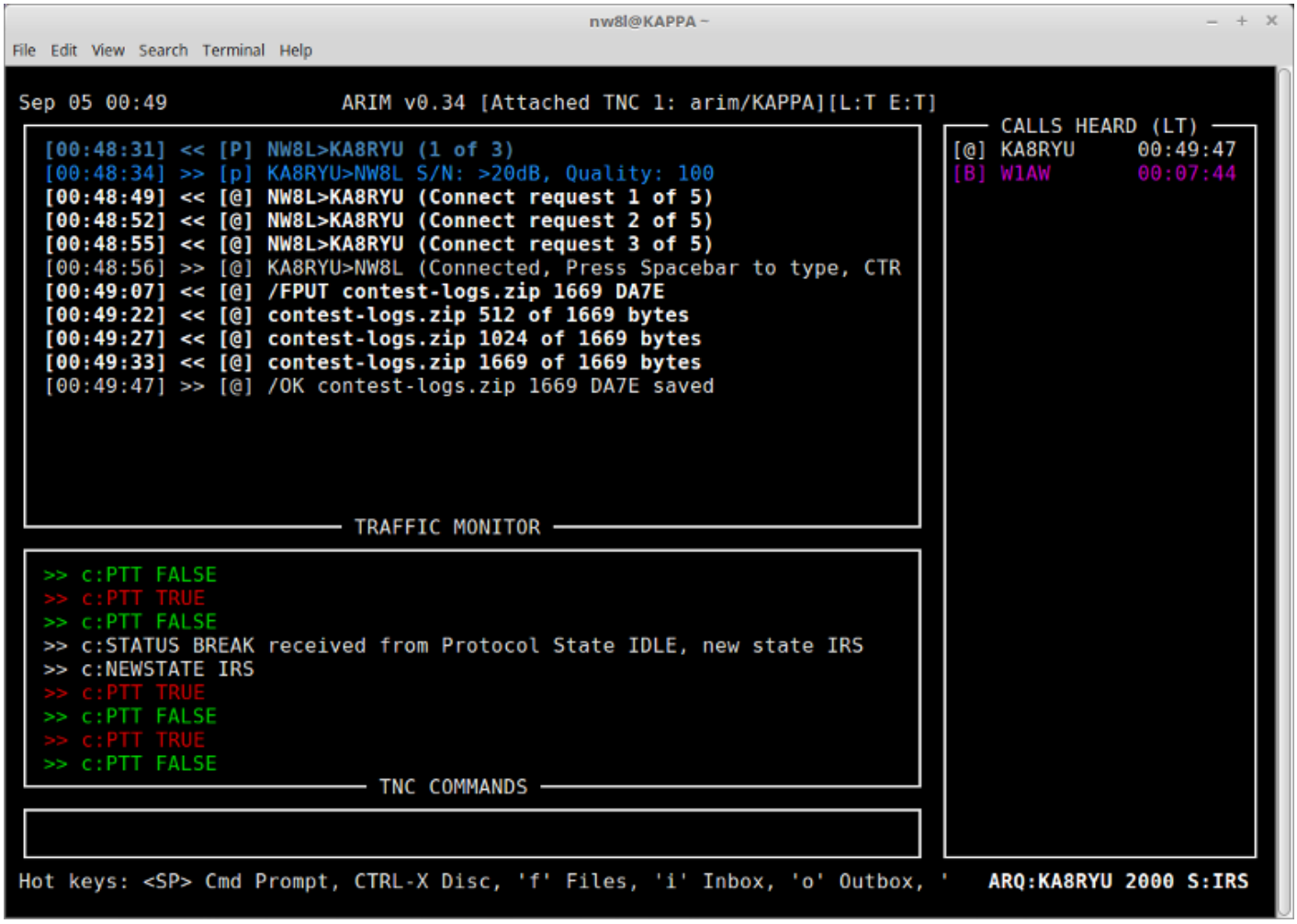
To upload a file, press the spacebar to open the command prompt and enter the **`sf` *`nbr`*** command like this:



The upload will begin:



Press 'q' to quit the shared files viewer.



The progress of the upload is reported as each data frame is sent. When it's complete an `/OK` response is received from the remote station to confirm the file was saved. If the remote ARIM station encounters an error receiving the file, it returns an `/ERROR` response and the upload is canceled.

Uploaded files are stored in the destination directory on the remote station. ARIM will create this if it doesn't already exist.

While the upload is in progress no text or commands are accepted from the command prompt. This condition is indicated by a `!` character seen in the status bar indicator area:

```
ARIM Idle: ARQ file upload complete                                  ! ARQ:KA8RYU 500 S:IDLE
[##########################################################][  Upload: 100%]
```

To cancel the upload, press CTRL-X to close the ARQ connection.

Use the **`sf -z n`** command to compress the file before it is sent. This can reduce transfer time for larger files. Note that compression isn't useful for very small files, or files that are already compressed (e.g. "zip" files or many image file types). In these cases the zlib format overhead can make the file size increase, not decrease.

Use the **`sf [-z] n dir`** command to store the file to a location other than the default download directory. The *dir* parameter specifies the destination directory as a path relative to the shared files root directory at the remote station, e.g. *contests/2016*. If you do, the destination directory must made visible by an *add-files-dir* configuration file parameter at the remote station. This option can be combined with the *-z* option.

ARQ: Session Authentication

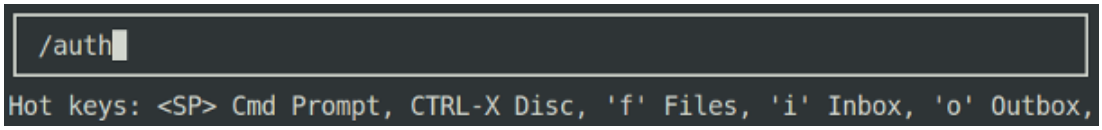
The optional Mutual Authentication feature provides a way to verify the identities of the stations in an ARQ session. This works by making each station prove to the other that it possesses a shared secret (password)

known only to the two stations involved. This is done using a Digest Access Authentication scheme very similar to that specified in RFC 2069 with the exception that the MD5 hash function is replaced by the more modern and secure Blake2 hash function. The exchange of proofs is accomplished without sending the password itself over the air. The process is automatic and requires no operator intervention. This feature can be used to:

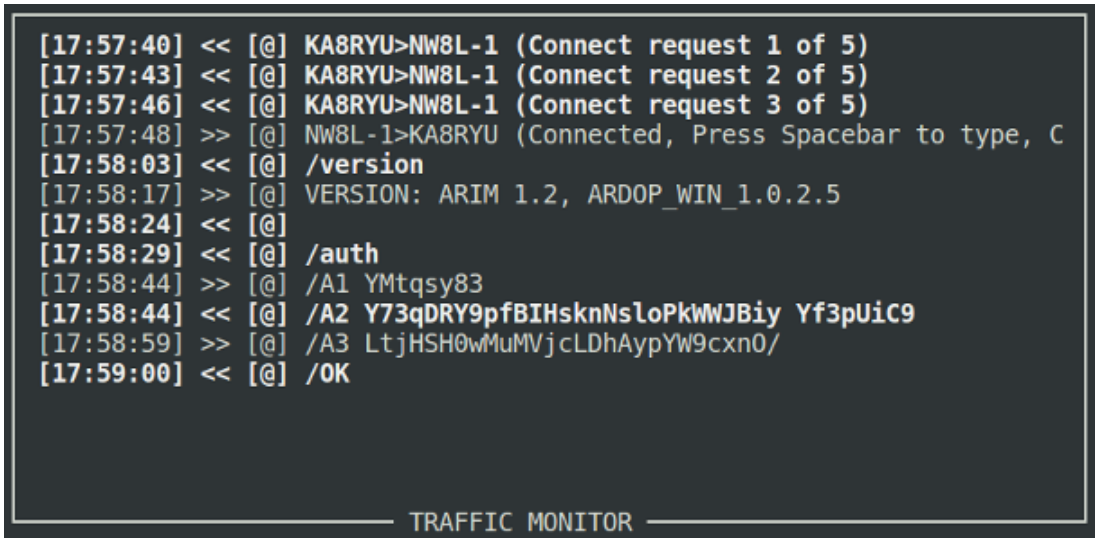
- verify the identity of the station you connect to in ARQ mode
- grant another station access to controlled resources located on your station

Once an ARQ connection is established, the operator can invoke the mutual authentication process manually using the `/auth` command. If it succeeds, the identity of the remote station is proven and the operator can read, upload or download files with confidence. If it fails, because one station or the other doesn't know the shared secret (password), the operator is presented with a warning message and given the choice of either continuing or terminating the ARQ session.

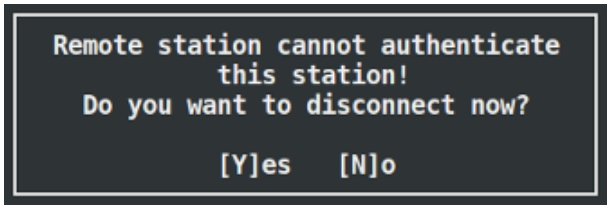
For example, to authenticate pre-emptively, press the spacebar to open the command prompt and enter the`/auth` command:



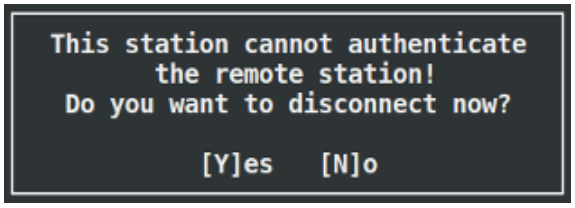
The remote station sends a challenge which is the first of 3 messages (A1, A2 and A3) exchanged during the authentication process.



If the remote station cannot authenticate your station, you are alerted and asked if you want to disconnect:



Likewise, if your station cannot authenticate the remote station, you are alerted and asked if you want to disconnect:



In either case the problem is likely to be missing or incorrect password digest file entries.

The authentication challenge/response protocol

1. The process begines when the operator at one station (the *client*), connects to another station (the *server*) and:
   - attempts to access a controlled resource for the first time

   or:
   - sends the `/auth` command
2. At the server station, ARIM receives the command and recognizes that authentication is required. It searches its password digest file for an entry with the client station's call sign as the *client* and its own call sign as the *server*. If it cannot find such an entry, it responds with `/EAUTH` and the operator at the client station is informed that authentication has failed. If ARIM does find such an entry, it responds with an authentication *challenge* in the form of the `/A1` command. The challenge includes an opaque *nonce*, a token whose content it controls and which must be incorporated into the response made by the client station.
3. At the client station, ARIM receives the `/A1` command. It searches its password digest file for an entry with its own call sign as the *client* and the server station's call sign as the *server*. If it cannot find such an entry, it responds with '/EAUTH' and the operator at the client station is informed that authentication has failed. If the needed entry is found, ARIM computes a response token which proves that it knows the shared secret. The response is:
   - **H(HA1:nonce:HA2)**

   where:
   - **H(x)** is the Blake2 hash of an array of data 'x'.
   - **HA1** is **H(client_call:server_call:password)**, the password hash token stored in the password digest file at the client station.
   - **nonce** is the token sent by the challenger to influence the response hash value.
   - **HA2** is **H(method:path)** where **method**' is `FPUT`, `FGET`, `FLIST` or `FILE` and **path** is the directory or file path referenced by the command.

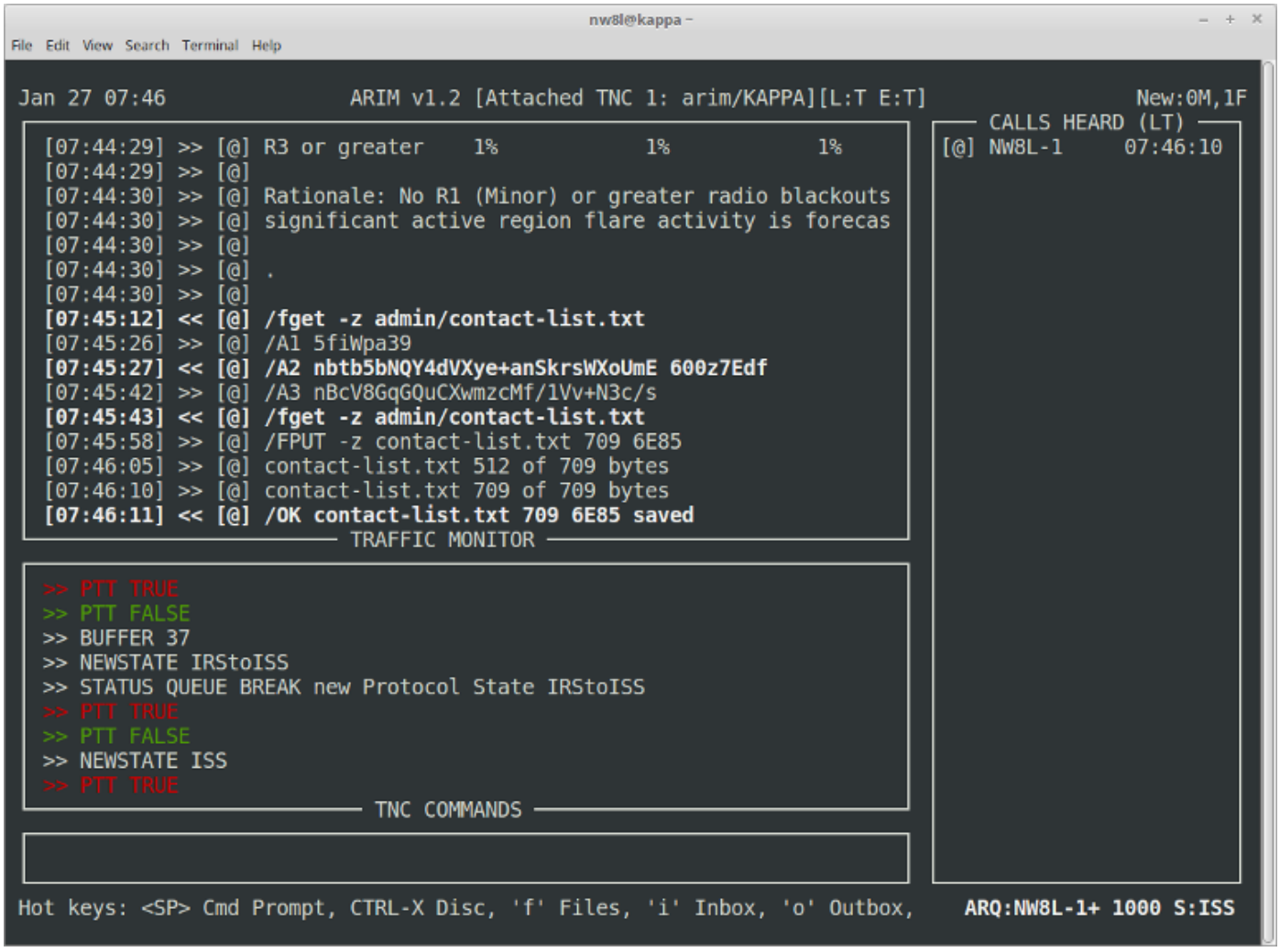
   Thus the response to each challenge is unique, in a way dependent on the nature of the request and the nonce issued by the challenger. This makes "replay" attacks difficult to mount and recovery of the password from the response token computationally infeasible. ARIM sends the response token plus a challenge nonce to the server station in the form of the `/A2` command.
4. At the server station, ARIM receives the `/A2` command and checks the challenge response token it contains. It does this by computing the response token it expects using the password digest token (HA1) stored in the password digest file *it* owns. It compares this with the response token received from the client station. If they don't match, then the client station must not have the same password digest that ARIM has stored locally. In this case ARIM will send `/EAUTH` to the client station where the operator will be informed that authentication has failed. If they match, then the client station has proven its authenticity and ARIM will send a response to the challenge contained in the `/A2` command. As before, this is computed as **H(HA1:nonce:HA2)**, using the challenge nonce received from the client station. It sends this response to the client station in the form of the `/A3` command.
5. At the client station, ARIM receives the `/A3` command and checks the challenge response token it contains. It does this by computing the response token it expects using the password digest token (HA1) stored in the password digest file *it* owns. It compares this with the response token received from the server station. If they don't match, the server station must not have the same password digest that ARIM has stored locally. In this case ARIM will inform the operator that authentication has failed. If they match, then the server station has proven its authenticity and ARIM re-sends the command that triggered the process, unless it was the `/auth` command. In that case there is no work to do, so ARIM simply sends the `/OK` response to signal that the mutual authentication is complete.

To strike a reasonable balance between security and the "air time" cost of the challenge/response data exchanges, challenge nonces contain 6 bytes of data and the Blake2 response hash is truncated to 21 bytes, or 168 bits (big enough for excellent *collision resistance*). These are sent in base64 encoded form making the over-the-air sizes 8 and 28 bytes respectively.

ARQ: File access control

Shared files directories at a station can be designated as *access-controlled* with the **ac-files-dir** parameter in the *arim.ini* configuration file. These resources are available only to stations that successfully authenticate themselves

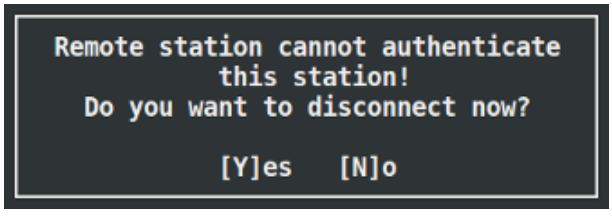
in an ARQ session. By default, the authentication process is triggered automatically on the first attempt by a station to access a controlled resource. Here is an example:



The operator at KA8RYU connects to NW8L-1 and reads the space weather report. Next he attempts to download a controlled file, *admin/contact-list.txt* with the `/fget` command. This triggers an authentication challenge from NW8L-1 in the form of the `/A1` command. The mutual authentication succeeds, so the `/fget` command is automatically repeated and the file is downloaded. Note the **+** character appended to NW8L-1's call sign in the status bar. This indicates that the ARQ session is authenticated.

If mutual authentication succeeds, subsequent accesses of controlled resources on the remote station proceed normally.

Alternatively, the operator at KA8RYU can send the `/auth` command pre-emptively to authenticate with the remote station *before* attempting to access controlled directories and their contents. On receiving this, if station NW8L-1 is unable to find a password digest file entry for KA8RYU, it sends the `/EAUTH` response and the operator at KA8RYU is alerted:



Otherwise the authentication process proceeds as usual.

Note: shared files directories defined by **add-files-dir** parameters in the *arim.ini* configuration file are *not* access-controlled. These are accessible by any station in ARQ and FEC mode operations. Likewise, the root shared files

directory is always accessible by any station for file listing, uploads and downloads.

ARQ: Password management

Passwords are created at the ARIM command prompt with the **`passwd`** command and stored in digest ("hashed") form in a file. Passwords are hashed with a *salt* (the call signs of the corresponding stations), so that any given password results in unique hash for different call sign pairs. This hash is identical to the **HA1** term used in the digest authentication scheme, so it can be read out of the password digest file when needed, making it unnecessary to store the original password. Mathematically, the Blake2 hash function is a strong *one way* function; it is computationally infeasible to recover the password from the hash. Because the password hash is stored in a file the authentication process can operate without operator intervention; there's no need to remember and enter the password each time.

In the context of an ARQ session, the term *client* refers to the station where an operator triggers the authentication process by:

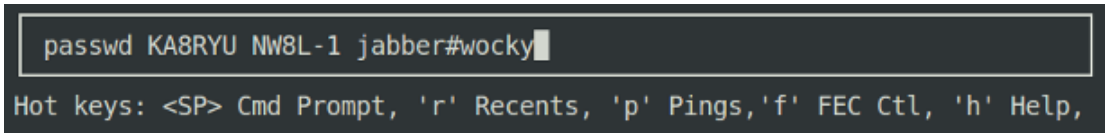
- sending the `/auth` command to the other station
- attempting to access a controlled resource at the other when the ARQ session is not yet authenticated.

When this happens the other station assumes the role of *server* and responds with an authentication challenge. These definitions imply that an operator is always present at the *client* station, but not at the *server* station, which is assumed to be operating automatically. However, once an ARQ connection is established between two stations, the operator at either end can seize the *client* role by being the first to trigger the mutual authentication process.

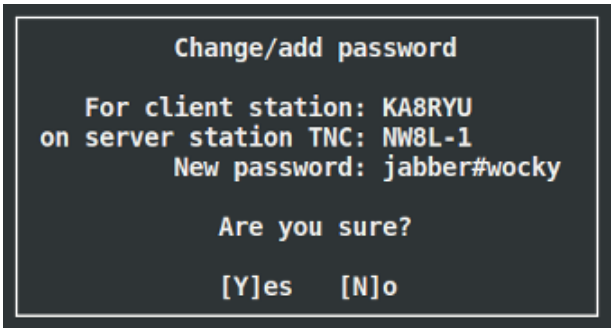
The **`passwd`** command must be invoked at both stations like this:

```
passwd client_call server_call some-password
```

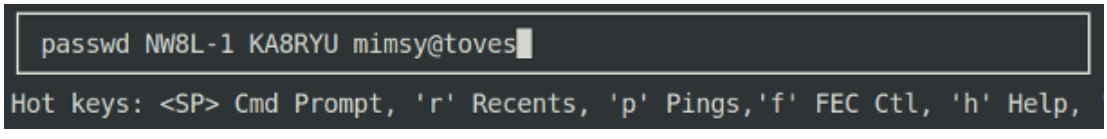
Let's assume that `client-call` is KA8RYU, and `server-call` is NW8L-1. Press the spacebar to open the command prompt, and enter the `passwd` command like this:



You will be prompted to confirm the password.



Press 'y' to store the password digest into the file. *This command must be executed at both stations to store the password digest on each.* If this is done, KA8RYU can connect to NW8L-1 and successfully authenticate if challenged. However, this doesn't mean that the reverse is true: NW8L-1 can't connect to KA8RYU and do the same. To make this possible the `passwd` command must be invoked again at each station with NW8L-1 specified as the *client*:



Now either station can connect to the other and play the role of *client* to the other station's role of *server* in the authentication process. Note that a different password is used; this is the best practice. If we read the *arim-digest* password digest file at either station using the **`rp`** command in the Shared Files viewer we see this:

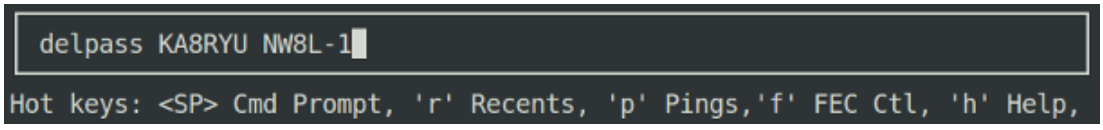
There are two entries, one for KA8RYU as *client* and NW8L-1 as *server*, and another for NW8L-1 as *client* and KA8RYU as *server*. This allows both one-way and two-way relationships to be defined. For instance, a station serving as a document repository can issue passwords to multiple stations who connect to it as clients and access the controlled files it offers. However, no such password is issued to the repository station by the client stations; it cannot connect to any of them in the client role and authenticate for the purpose of accessing any controlled resources they publish.

To protect against accidental disclosure of password digest file contents, any file named 'arim-digest', or variations like 'arim-digest.bak', no matter where located in the file system, are protected against access by a remote station. Avoid easy-to-guess passwords, and never include the call signs of the client or server stations in the password. While the base64 encoded Blake2 digest (HA1) may look quite random, it contains only as much *entropy* as the original password, so choose passwords well. Note: *passwords are case sensitive!*

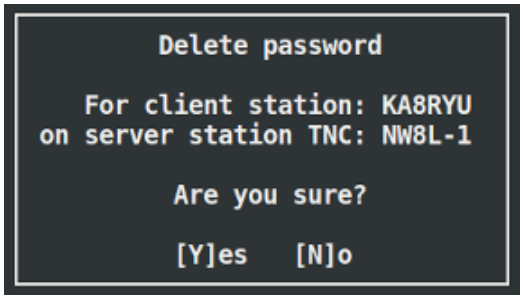
Use the **`delpass`** command to delete a password digest from the *arim-digest* file:

```
delpass client_call server_call
```

Let's assume that `client-call` is KA8RYU, and `server-call` is NW8L-1. Press the spacebar to open the command prompt, and enter the `delpass` command like this:



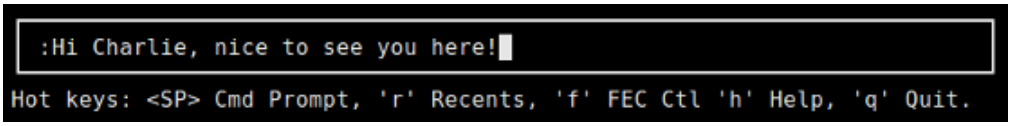
You will be prompted to confirm the deletion.



Press 'y' to delete the password digest from the file.

FEC: Sending an unproto message

A TNC must be attached and the RF channel must not be *busy*. Press the spacebar to open the command prompt.

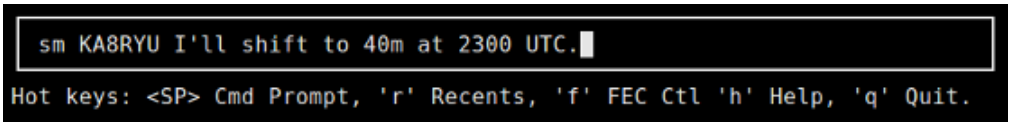


Prefix the message with the ':' character. Press ENTER to send the message out over the air as raw ARDOP FEC frames, rather than being formatted as an ARIM message. The message will appear in the monitor view tagged with the **`[U]`** indicator.

If the RF channel is busy, ARIM won't send the unproto message, and an error message will be shown on the status bar.

FEC: Sending a message (to TNC)

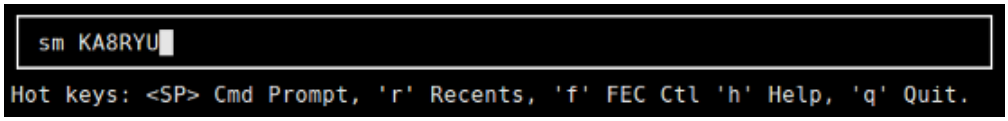
When attached to a TNC, single-line messages can be sent using the **`sm`** ***`call [msg]`*** command, where *call* is the call sign of the recipient and *msg* is the message text. This is useful when the message is short and you wish to bypass the message composer. Press the spacebar to open the command prompt and enter the command like this:



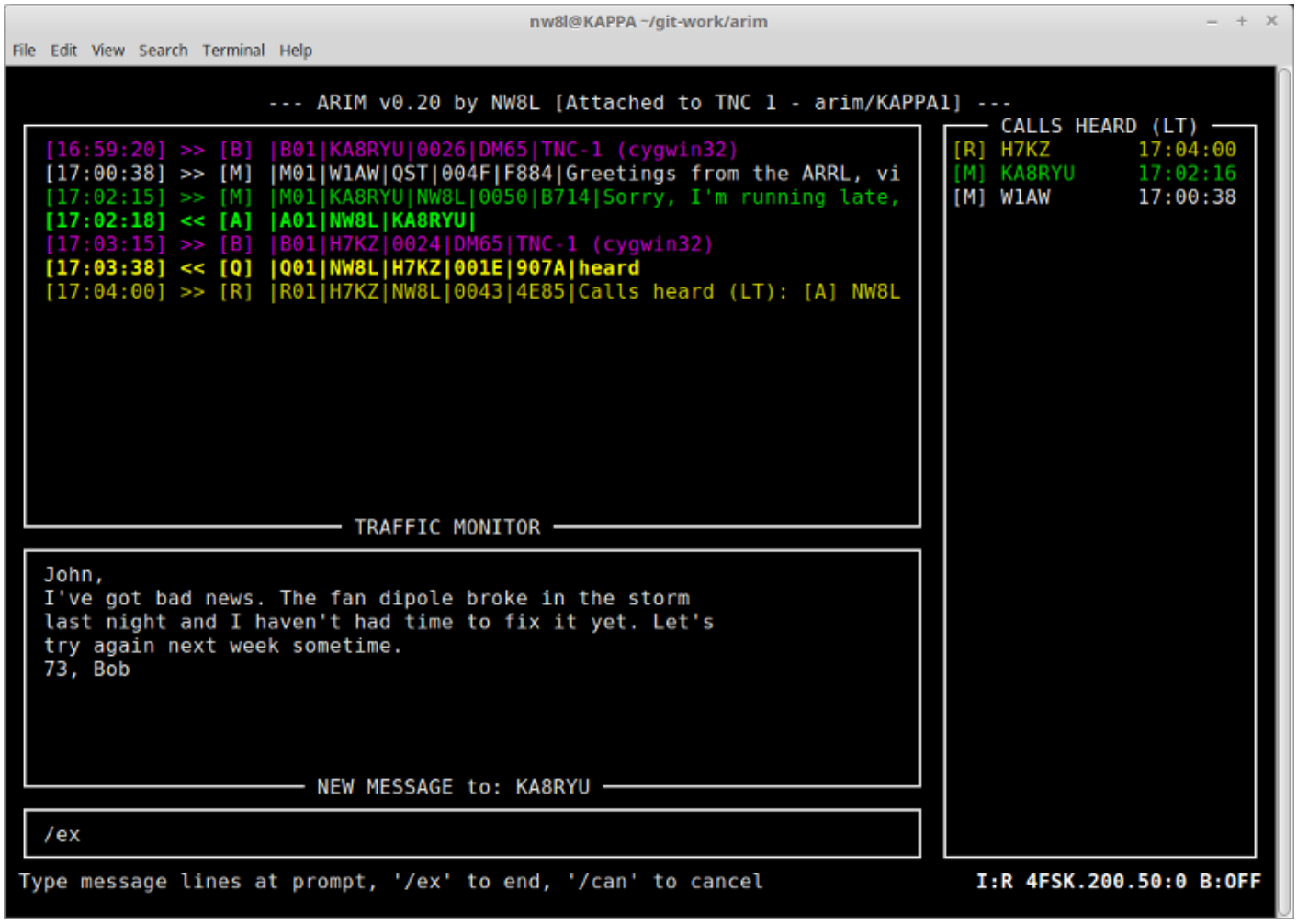
The message will be sent immediately after pressing the ENTER key to complete the command. A copy will be stored in your sent messages mailbox.

If the RF channel is busy, ARIM won't send the message, and an error message will be shown on the status bar.

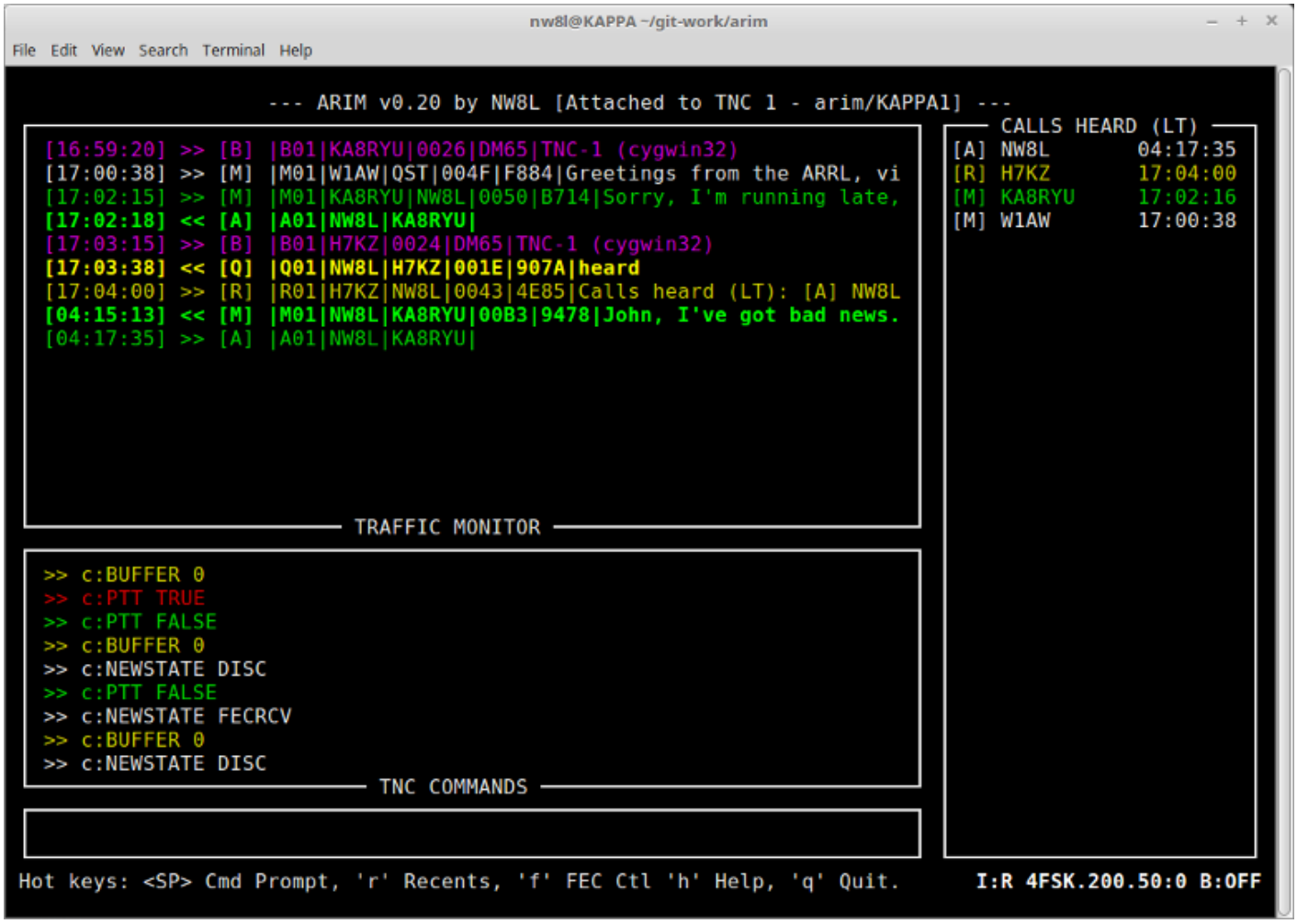
For longer messages, press the spacebar to open the command prompt and enter the command like this:



The message composer opens.



Enter the message line-by-line at the command prompt. When done, enter **`/ex`** at the start of a new line to close the message composer view and send the message on its way. A copy will be stored in your sent messages mailbox.

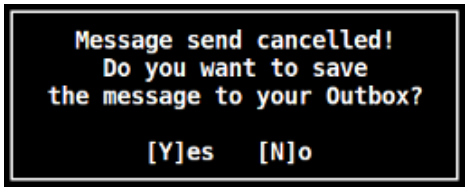


The monitor view shows the outbound message and the inbound ACK from the recipient station.

To cancel a message you've started to send, enter **`/can`** at the start of a new line to close the message composer view and discard the message.

If the TNC is *busy* when the message is sent, the message will be stored in the outbox automatically. If the message send *fails* for some reason (NAK, ACK timeout, etc.) then this dialog will appear:

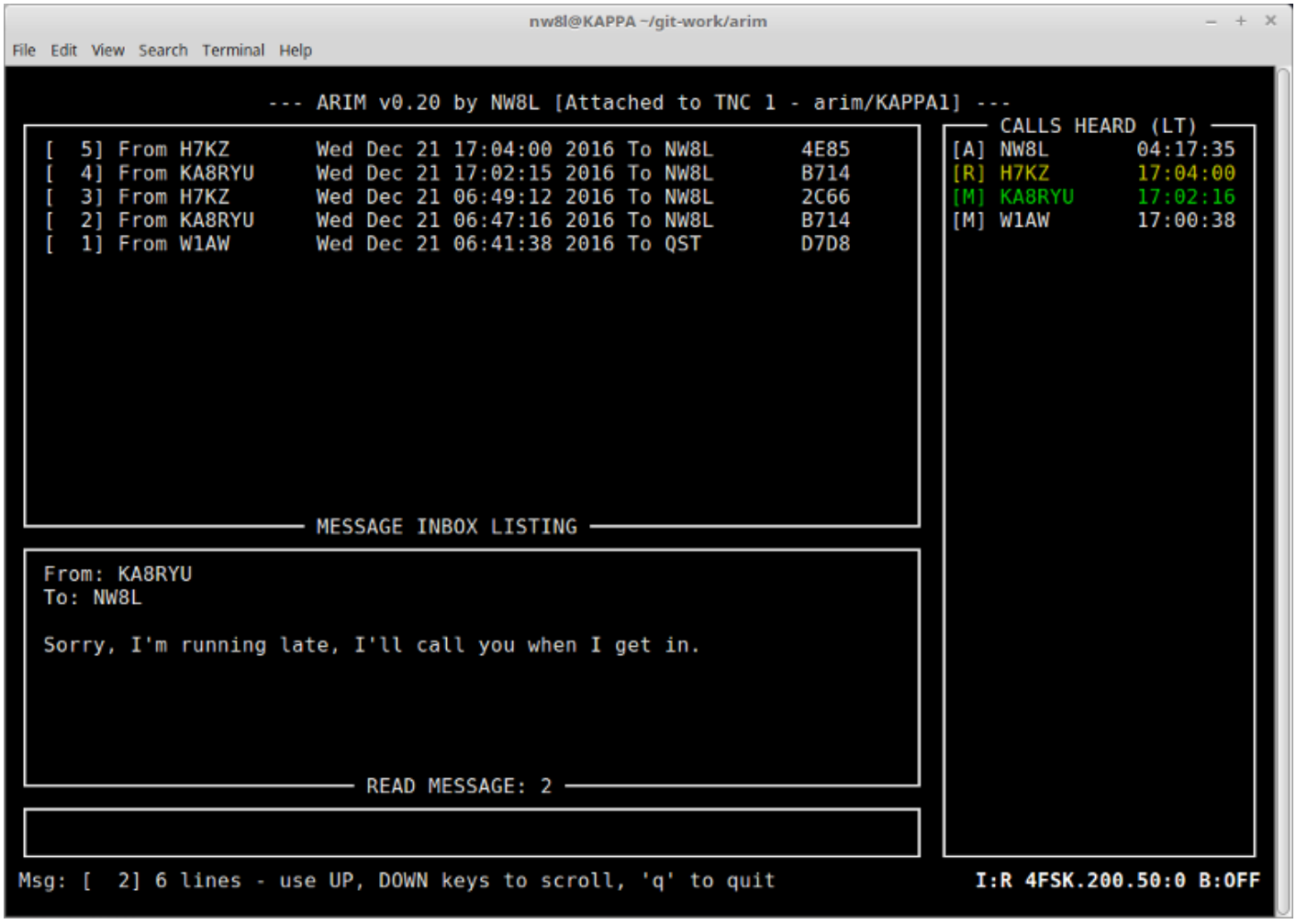
Press 'y' to store the message to the outbox, 'n' to discard it. If you *cancel* the message send then this dialog will appear:



Press 'y' to store the message to the outbox, 'n' to discard it.

FEC: Reading messages

When in the inbox or outbox viewer, press the spacebar to open the command prompt and enter the **`rm nbr`** command where *nbr* is the message number. This opens the message viewer.

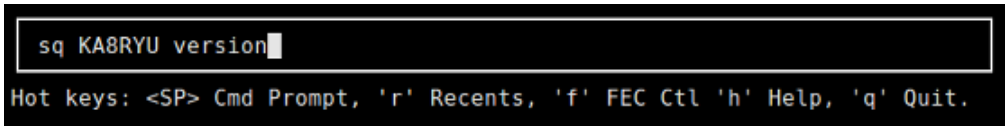


The message number and size in lines are shown in the Status Bar. Use the UP and DOWN arrow keys or the PAGEUP, PAGEDOWN, HOME and END keys to scroll through the message. Press 'q' to quit the message viewer and return to the message header view.

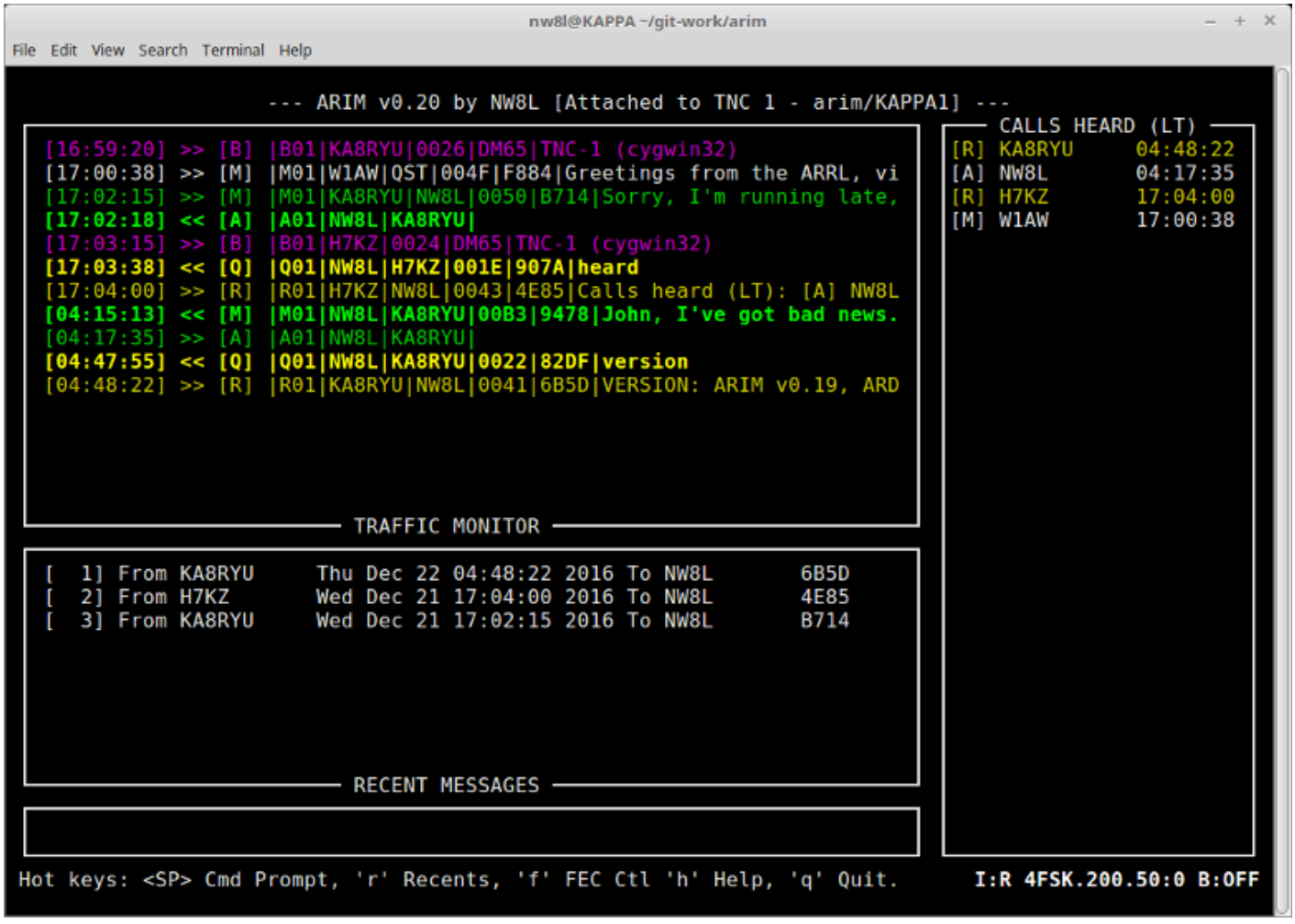
FEC: Query another station for information

A TNC must be attached and the RF channel must not be *busy*. Press the spacebar to open the command prompt and enter the **`sq call query`** command where *call* is the call sign of the target station and *query* is one of these:

- **`version`** Query version of ARIM program and attached ARDOP TNC.
- **`gridsq`** Query gridsquare of target station (from *arim.ini* configuration file).
- **`pname`** Query name of attached ARDOP TNC (from *arim.ini* configuration file).
- **`info`** Query info text for the attached ARDOP TNC (from *arim.ini* configuration file).
- **`heard`** Query calls heard list of target station.
- **`netcalls`** Query net call signs list of target station.
- **`flist [dir]`** Query shared files listing from target station. *dir* is an optional directory path, relative to the shared files root directory, e.g. `contests/2016/FOBB`. If *dir* is not given, then the the shared files root directory is listed; otherwise the specified directory is listed. By default only files in the shared files root directory and dynamic files are listed. Subdirectories are not listed unless they are made visible by an *add-files-dir* configuration file parameter. Access-controlled subdirectories defined by the *ac-files-dir* configuration file parameter are indicated with the **!** (bang) character like this: `!DIR`.
- **`file fn`** Query file *fn*, where *fn* is the name of a file in the shared files directory, or a file path relative to it, e.g. `contests/2016/FOBB.log`.



The target station will send the requested information in a response message.

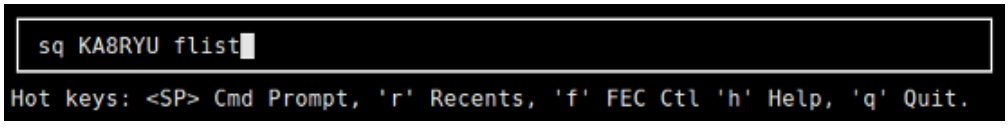


The response message will appear in the monitor view tagged with the **`[R]`** indicator. It will be stored in the message inbox for later viewing, and also appear in the Recents list.

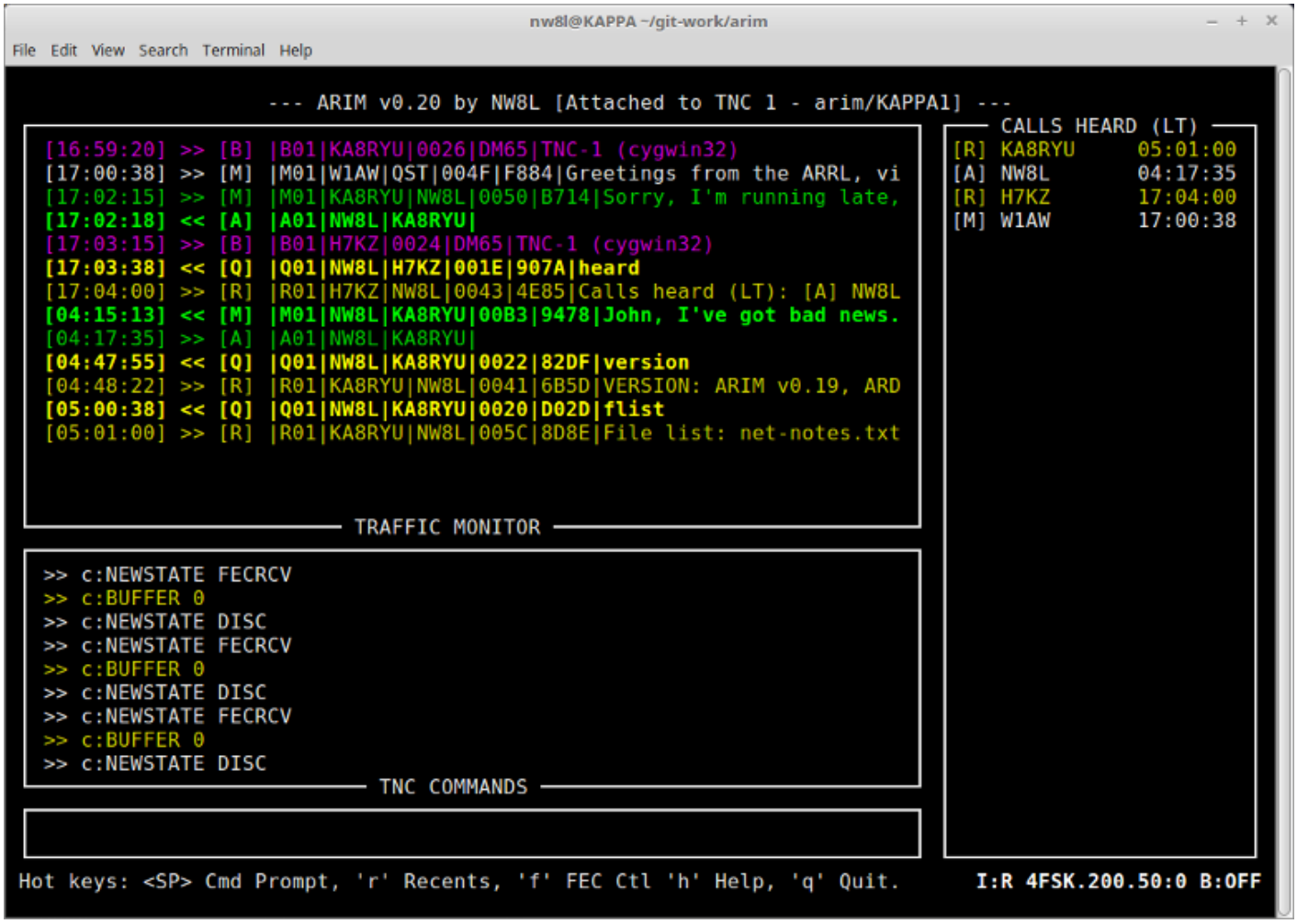
If the RF channel is busy, ARIM won't send the query, and an error message will be shown on the status bar.

FEC: Retrieving files from another station

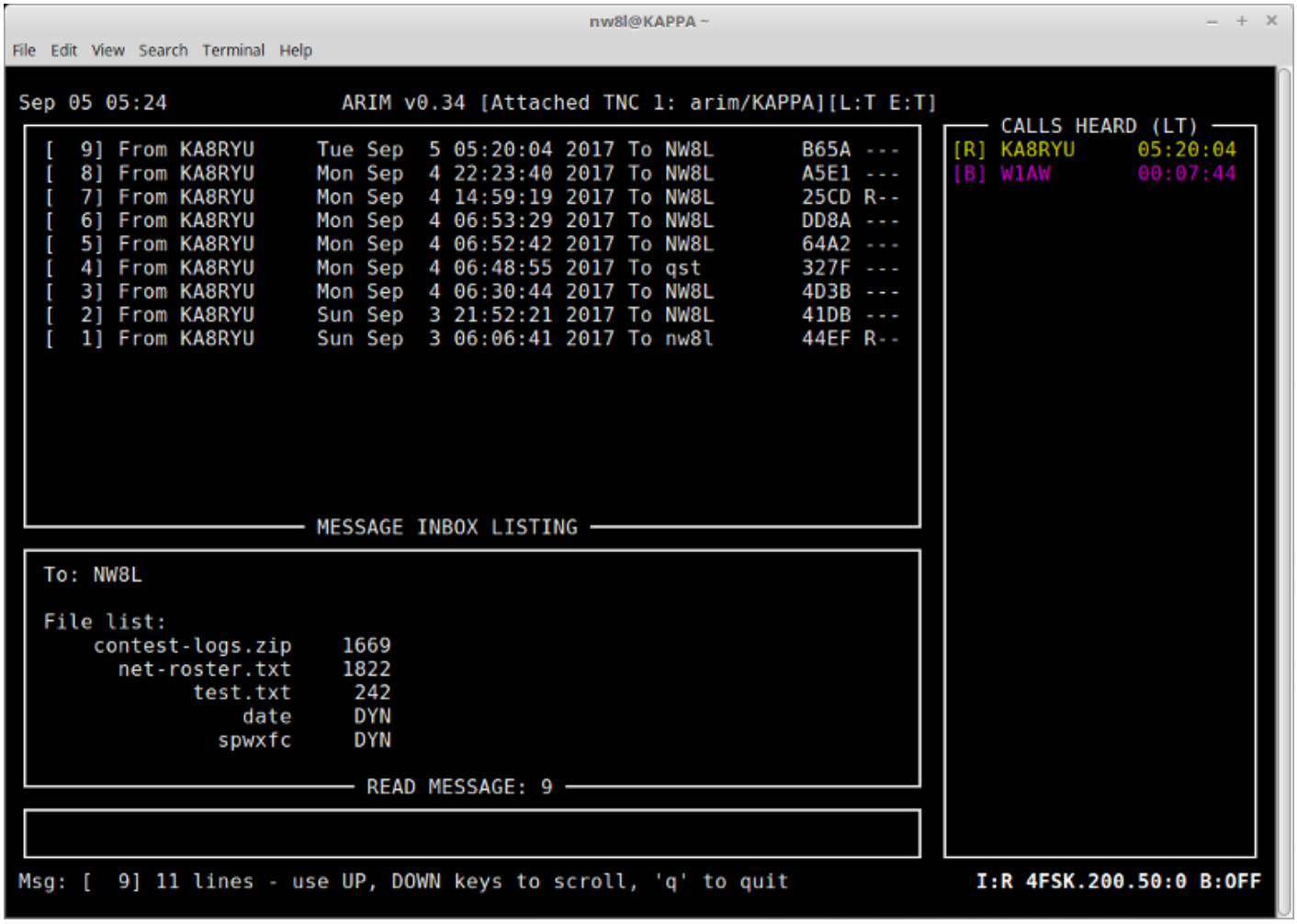
Use the **`sq` *`call`* `flist` *`[dir]`*** query to get a list of shared files from station *call* where *dir* is an optional directory path, relative to the shared files root directory, e.g. `contests/2016/FOBB`.



The list will be returned in the response message which is stored into the inbox.



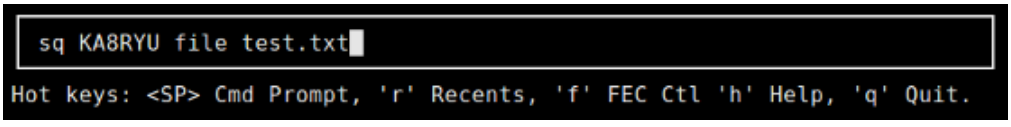
To see it, open the message inbox viewer with the **`li`** command, then use the **`rm` *`nbr`*** command to read the file list message:



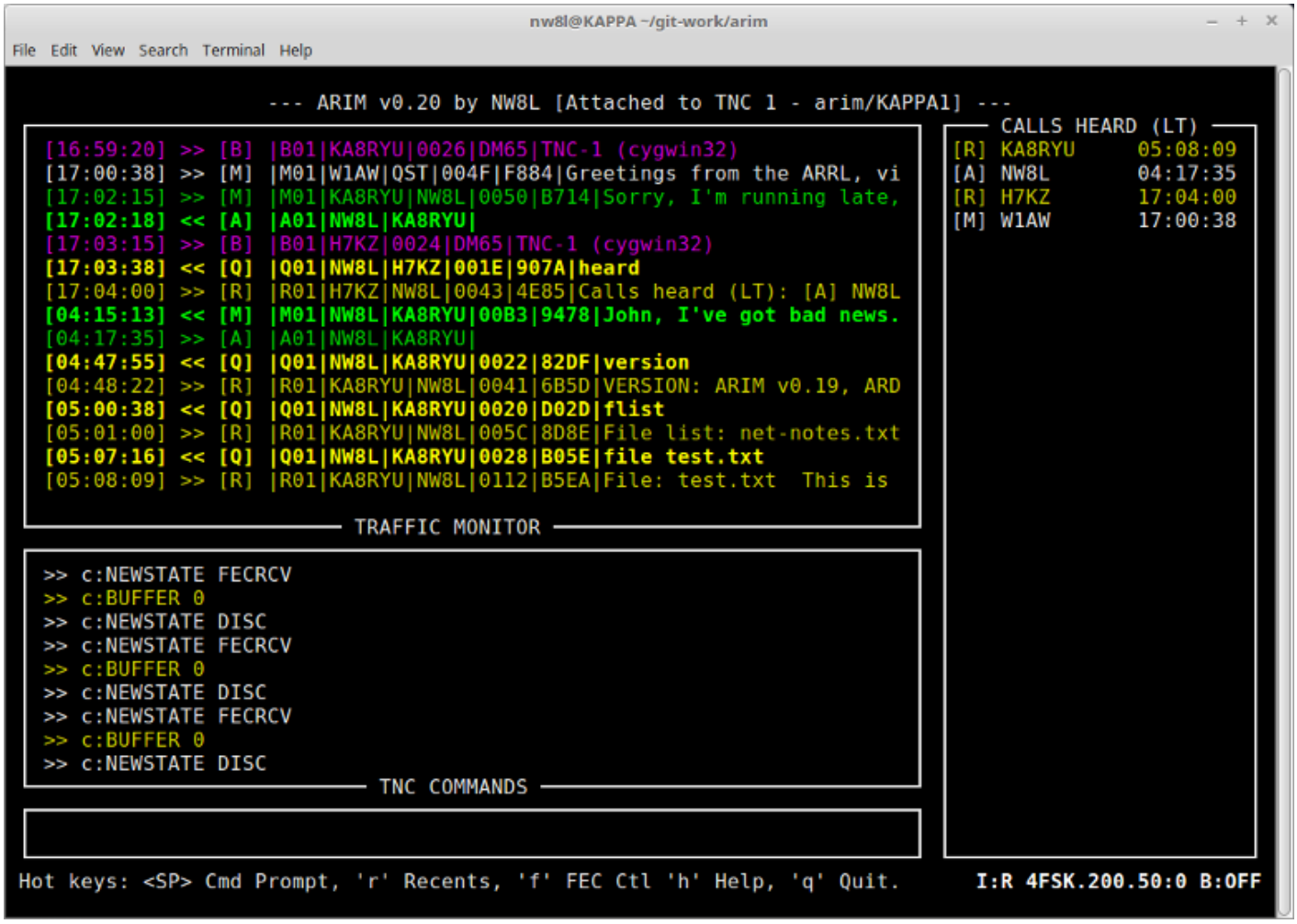
Note the name of the file you wish to get. NOTE: Static files have their size in bytes listed with the file name. Dynamic files have the `DYN` label listed with the file name to indicate that their size is variable.

Press 'q' to close the message viewer and 'q' again to close the message inbox view.

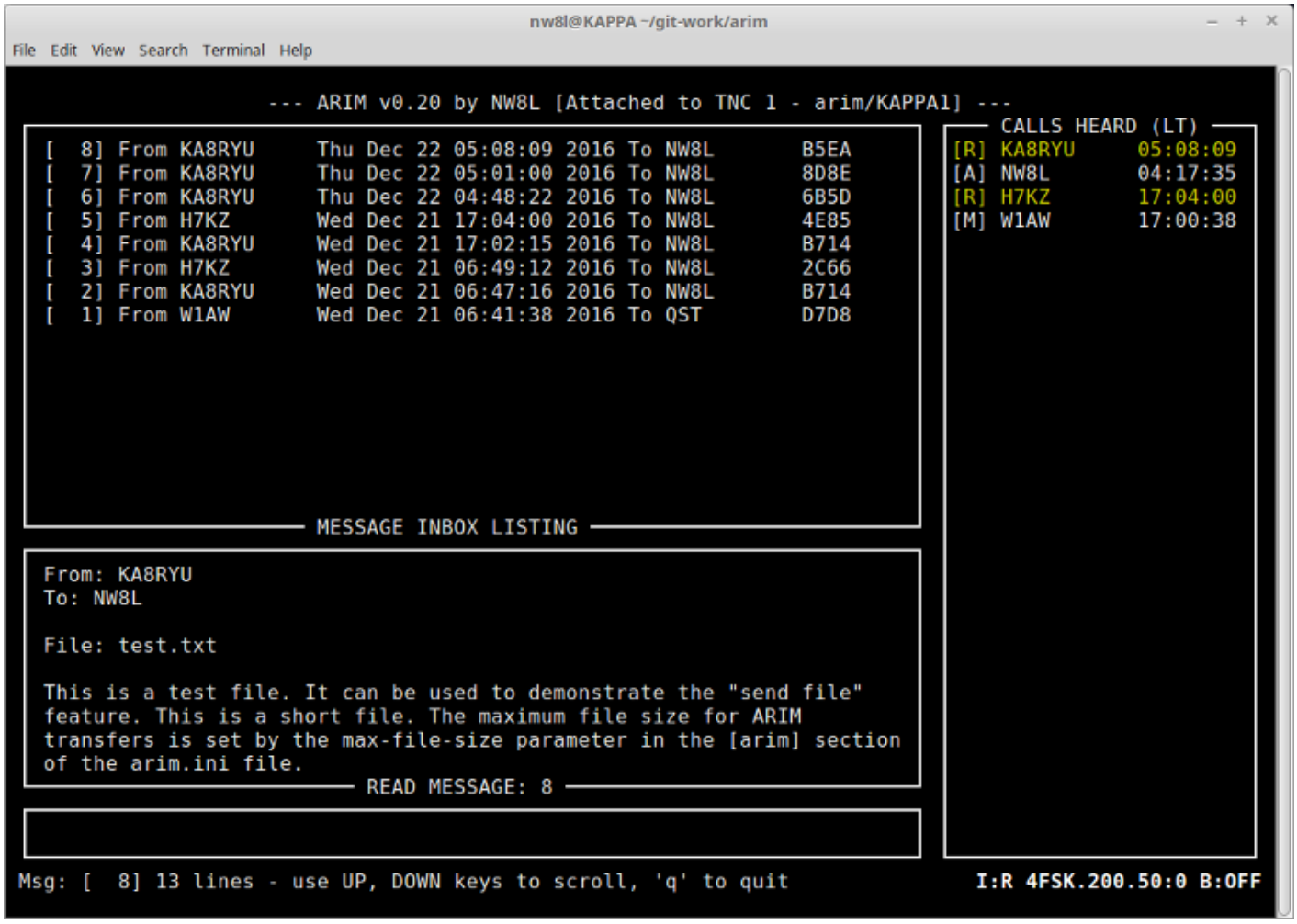
Now use the **`sq` *`call file fname`*** command to query the file *fname* from station *call*. NOTE: file names are case sensitive!



The file will be returned in the response message.



The message is stored into the inbox where it can be viewed or saved to disk.



NOTE: the maximum file size is set by the **`max-file-size`** parameter in the **`[arim]`** section of the *arim.ini* configuration file. The file list returned by the **`flist`** query is filtered so that only static files less than the max size are listed, and the output of dynamic files is limited to the max size. At this time the absolute maximum allowed is 16384 bytes. The location of the shared files root directory is set by the **`files-dir`** parameter in the **`[arim]`** section of the *arim.ini* configuration file.

FEC: Using message send repeats

In difficult conditions it may be helpful to enable automatic repeats of messages if they are nak'd or an ACK timeout occurs. Three parameters control message send repeat behavior:

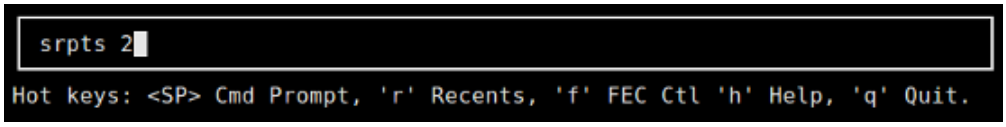
- **send repeats** - this is the number of times a message send will be repeated in the absence of an ACK response from the recipient.
- **ack timeout period** - after sending a message, this is the maximum time in seconds that ARIM will wait for an ACK before repeating it.
- **FEC mode downshifting** - if enabled, downshifting causes each message repeat to be made in a more robust FEC mode than the previous repeat in hopes of getting through in poor conditions. The starting FEC mode is automatically restored at the conclusion of the message repeats.

These parameters are set in the **`[arim]`** section of the *arim.ini* configuration file, but can also be modified on-the-fly from the ARIM command prompt so that the operator an adapt to changing conditions. Here are the commands you can use from the command prompt:

- **`srpts`** - sets the number of times a message send will be repeated in the absence of an ACK response from the recipient. Value must be in the range 0-5. Setting it to '0' disables send repeats. Example: `srpts 3`
- **`ackto`** - sets the message ACK timeout in seconds. Value must be in the range 10-999. Example: `ackto 30`
- **`fecds`** - enables or disables FEC mode downshifting when repeating messages. Value must be either TRUE or FALSE (not case sensitive, T or F are ok). Examples: `fecds TRUE` or `fecds t`

- **`srset`** - Open a pop-up dialog showing the message send repeat settings. Example: `srset`

To change one of these, press the spacebar to open the command prompt and enter the command like this:



The change will be confirmed by a message on the Status Bar. When in doubt, use the `srset` command to check settings.

FEC: Beaconing

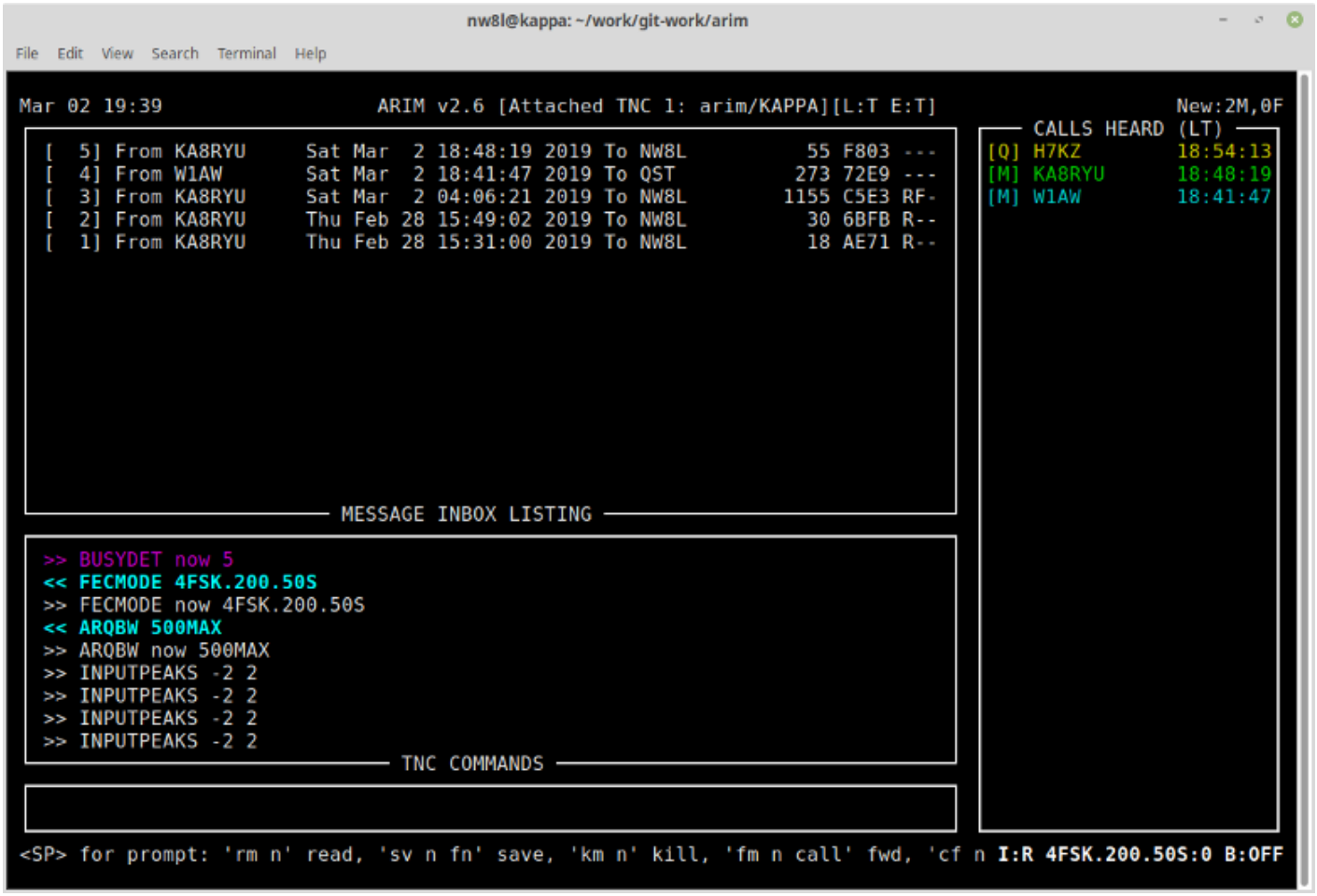
Beaconing can be configured for each TNC by the **`btime`** parameter in the relevant **`[tnc]`** section of the *arim.ini* configuration file. This parameter defines the beacon interval time in minutes, with '0' meaning that beaconing is disabled.

When attached to a TNC, beaconing can be managed using the **`btime`** and **`btest`** commands. Use **`btime`** ***`nbr`*** to set the beacon interval to *nbr* minutes, and **`btest`** to test the beacon. The beacon status indicator in the Status Bar shows the interval time when beaconing is enabled, or "OFF" when disabled.

If the RF channel is busy, ARIM won't send the beacon, and an error message will be shown on the status bar. In the case of beacons scheduled every **`btime`** seconds, ARIM will try again once, 2 minutes later. If the RF channel remains busy, ARIM gives up and schedules the next beacon **`btime`** seconds later.

Working with the message inbox

Press the spacebar to open the command prompt and enter the **`li`** command to open the message inbox viewer.



Use the UP and DOWN arrow keys or the PAGEUP, PAGEDOWN, HOME and END keys to scroll through the message header list. Headers contain the following information, from left to right:

- **Message number**, most recent first.
- **Sending station call sign**
- **Date and time**
- **Destination station call sign**
- **Size** over-the-air transfer size in bytes, decimal format (before compression).
- **Checksum** 16-bit CRC, in hexadecimal format.
- **Status flags** where *R* means *read*, *F* means *forwarded*, *S* means *saved* and - means *flag not set*.

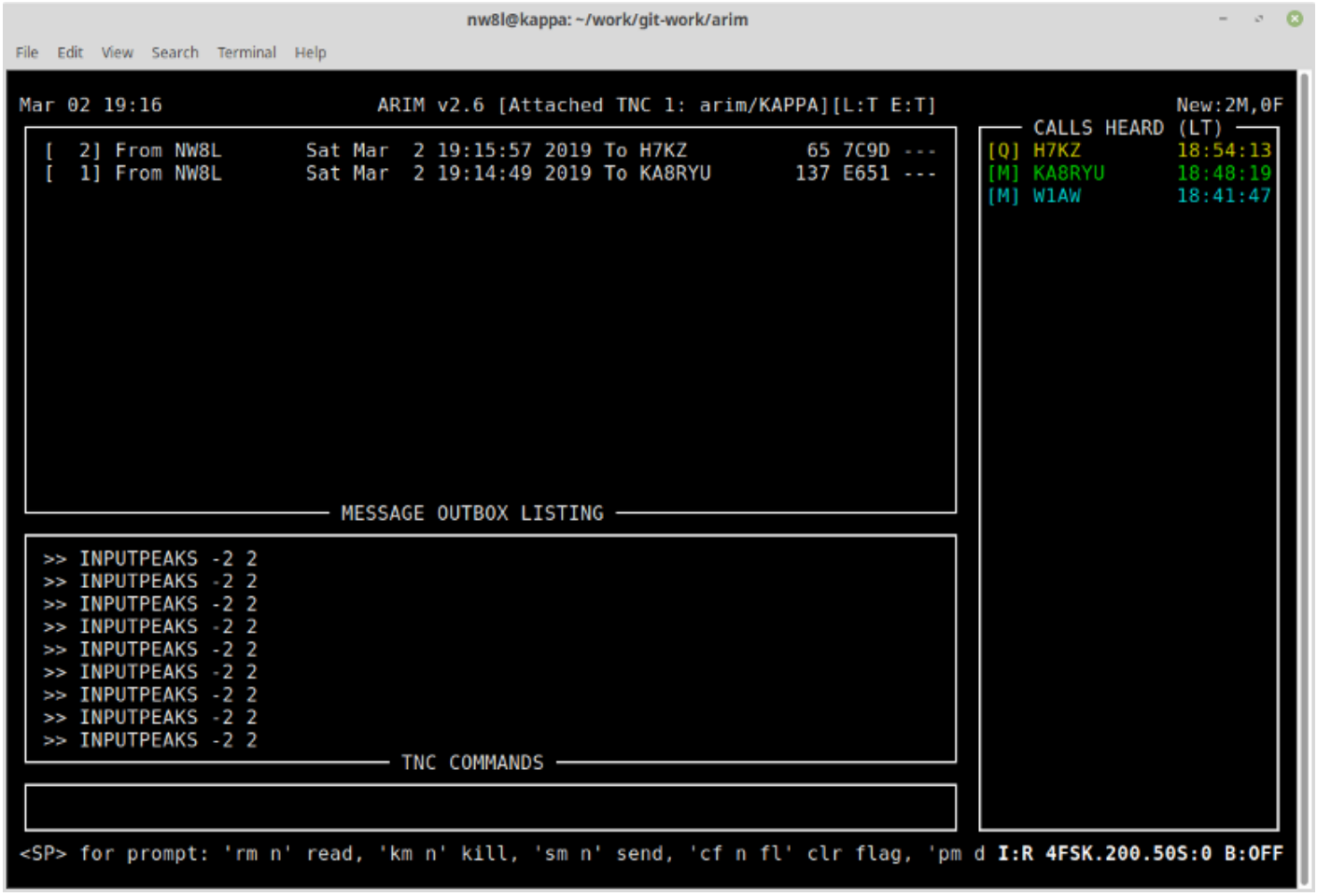
The following commands are available:

- **`rm` *`nbr`*** Read message number *nbr* in the message viewer.
- **`km` *`nbr`*** Kill (delete) message number *nbr*.
- **`pm` *`nbr`*** Purge (delete) all messages older than *nbr* days.
- **`sv` *`nbr fname`*** Save message *nbr* to file *fname*. *fname* can be an absolute path like */home/nw8l/foo.txt* or a relative path like *foo.txt* or *files/foo.txt*. Relative paths are relative to the ARIM working directory (the current directory when *arim* was invoked).
- **`fm` *`nbr call`*** Forward message number *nbr* to station *call*. Only available when attached to a TNC.
- **`cf` *`nbr flag`*** Clear flag *flag* on message number *nbr*. *flag* is one of *R* (read), *F* (forwarded), *S* (saved) or * (all flags).

Press 'q' to quit. Note: the automatic message purge process runs whenever this view is opened. This discards messages whose age in days exceeds the limit set by the **`max-msg-days`** parameter in the **`[arim]`** section of the *arim.ini* configuration file. Set it to **`0`** to disable automatic message purging. The default setting is 0.

Working with the message outbox

Previously composed messages and messages saved after a failed send operation are stored here. Press the spacebar to open the command prompt and enter the **`lo`** command to open the message outbox viewer.



Use the UP and DOWN arrow keys or the PAGEUP, PAGEDOWN, HOME and END keys to scroll through the message header list. Headers contain the following information, from left to right:

- **Message number**, most recent first.
- **Sending station call sign**
- **Date and time**
- **Destination station call sign**
- **Size** over-the-air transfer size in bytes, decimal format (before compression).
- **Checksum** 16-bit CRC, in hexadecimal format.
- **Status flags** where *R* means *read*, *F* means *forwarded*, *S* means *saved* and - means *flag not set*.

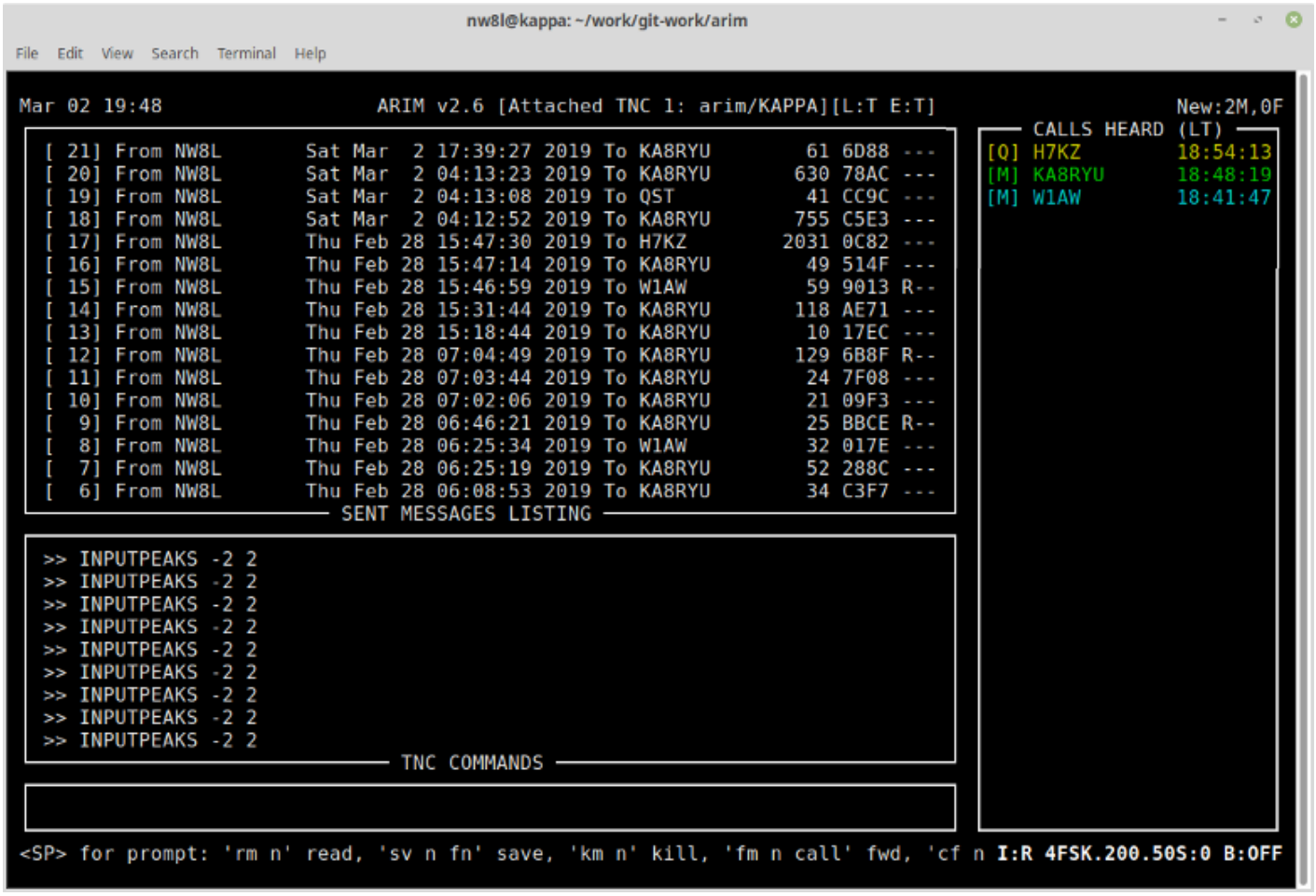
The following commands are available:

- **`rm`** ***`nbr`*** Read message number *nbr* in the message viewer.
- **`km`** ***`nbr`*** Kill (delete) message number *nbr*.
- **`pm`** ***`nbr`*** Purge (delete) all messages older than *nbr* days.
- **`sm`** ***`nbr`*** Send message *nbr* if attached to a TNC.
- **`cf`** ***`nbr flag`*** Clear flag *flag* on message number *nbr*. *flag* is one of *R* (read), *F* (forwarded), *S* (saved) or * (all flags).

Press 'q' to quit. Note: the automatic message purge process runs whenever this view is opened. This discards messages whose age in days exceeds the limit set by the **`max-msg-days`** parameter in the **`[arim]`** section of the *arim.ini* configuration file. Set it to **`0`** to disable automatic message purging. The default setting is 0.

Working with sent messages

Press the spacebar to open the command prompt and enter the **`ls`** command to open the sent message viewer.



Use the UP and DOWN arrow keys or the PAGEUP, PAGEDOWN, HOME and END keys to scroll through the message header list. Headers contain the following information, from left to right:

- **Message number**, most recent first.
- **Sending station call sign**
- **Date and time**
- **Destination station call sign**
- **Size** over-the-air transfer size in bytes, decimal format (before compression).
- **Checksum** 16-bit CRC, in hexadecimal format.
- **Status flags** where *R* means *read*, *F* means *forwarded*, *S* means *saved* and - means *flag not set*.

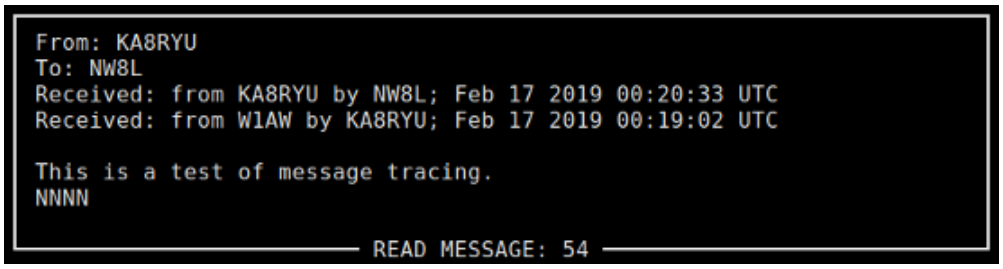
The following commands are available:

- **`rm`** ***`nbr`*** Read message number *nbr* in the message viewer.
- **`km`** ***`nbr`*** Kill (delete) message number *nbr*.
- **`pm`** ***`nbr`*** Purge (delete) all messages older than *nbr* days.
- **`sv`** ***`nbr fname`*** Save message *nbr* to file *fname*. *fname* can be an absolute path like */home/nw8l/foo.txt* or a relative path like *foo.txt* or *files/foo.txt*. Relative paths are relative to the ARIM working directory (the current directory when *arim* was invoked).
- **`fm`** ***`nbr call`*** Forward message number *nbr* to station *call*. Only available when attached to a TNC.
- **`cf`** ***`nbr flag`*** Clear flag *flag* on message number *nbr*. *flag* is one of *R* (read), *F* (forwarded), *S* (saved) or * (all flags).

Press 'q' to quit. Note: the automatic message purge process runs whenever this view is opened. This discards messages whose age in days exceeds the limit set by the **`max-msg-days`** parameter in the **`[arim]`** section of the *arim.ini* configuration file. Set it to **`0`** to disable automatic message purging. The default setting is 0.

Message tracing

The optional message tracing feature inserts headers like *Received: from KA8RYU by NW8L; Jan 30 2019 05:01:48 UTC* into received messages. If the message is forwarded to another station with tracing enabled, another *Received:* header is added by the receiving station, and so on. In this way a record of the message's progress through a network is built up as it is forwarded from station to station (read from bottom to top).



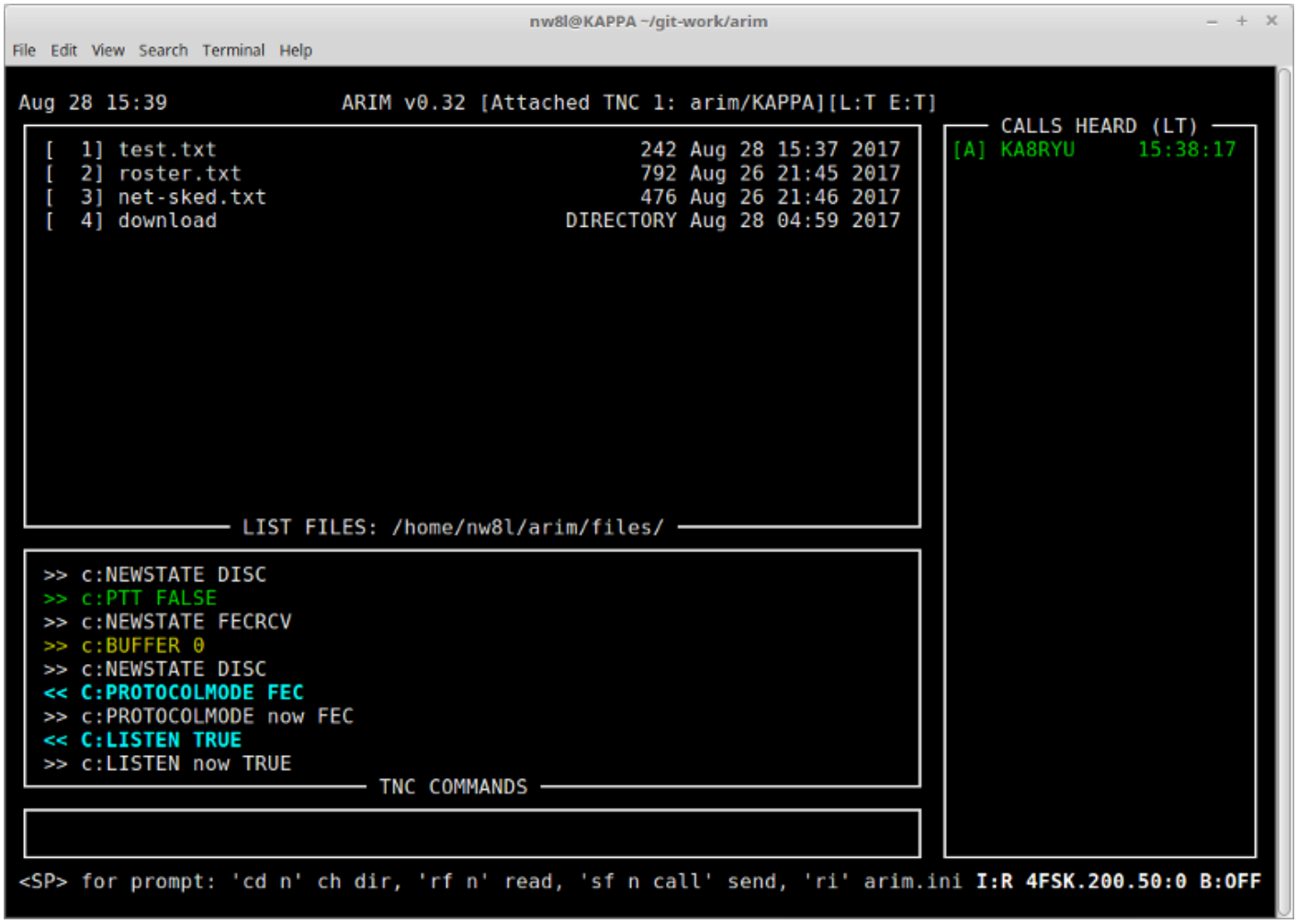
The *Received:* headers are visible in the message viewer together with the *From:* and *To:* headers. However, being message header lines, they are not visible and cannot be changed when sending or forwarding a message. This feature is compatible with previous versions of gARIM and ARIM.

The timestamp included in the *Received:* headers indicates time of receipt, either local time or UTC time at the receiving station, as set by the **`utc-time`** parameter in the **`[ui]`** section of the *arim.ini* configuration file. No time zone is included for stations configured for local time stamps (`utc-time=FALSE`).

Message tracing is enabled or disabled by the value of the **`msg-trace-en`** parameter in the **`[arim]`** section of the configuration file. Set to TRUE to enable tracing or FALSE to disable it. The default value is FALSE.

Working with the shared files viewer

Press the spacebar to open the command prompt and enter the **`lf`** command to open the shared files viewer.



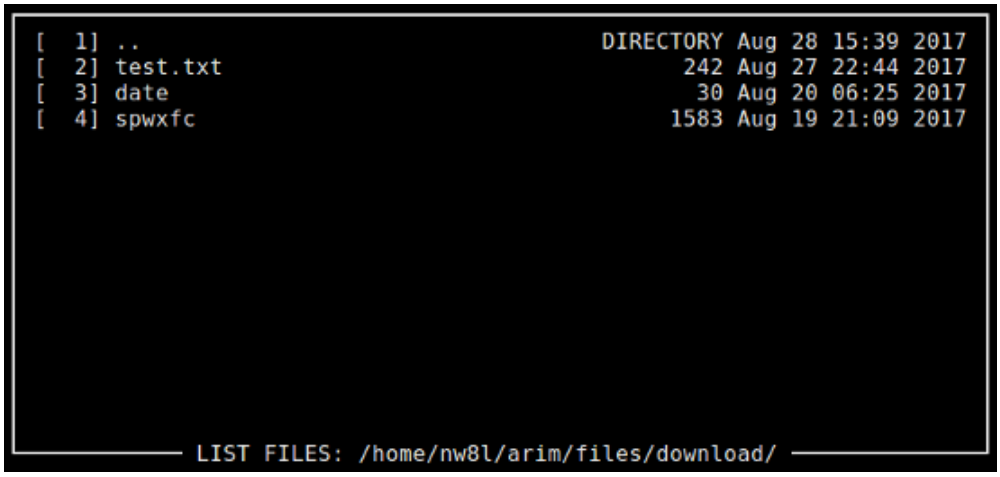
The viewer opens in the shared files root directory defined by the **`files-dir`** parameter in the **`[arim]`** section of the *arim.ini* configuration file. The directory path is shown in the title of the viewer window.

The contents of the directory are listed, one to a line. For files, the name, size in bytes and the last-modified date are shown. For subdirectories, the name and last-modified date are shown. Access-controlled subdirectories defined by the *ac-files-dir* configuration file parameter are indicated with the **!** (bang) character like this: `!DIRECTORY`.

Use the UP and DOWN arrow keys or the PAGEUP, PAGEDOWN, HOME and END keys to scroll through the listing. Each file or directory is given a number for use with commands. A list of available commands is printed in the Status Bar:

- **`rf`** ***`nbr`*** Read file number *nbr* in the file viewer.
- **`sf`** ***`nbr call`*** Send file number *nbr* to station *call* if attached to a TNC.
- **`ri`** Read the *arim.ini* configuration file, no matter where it is located.
- **`rp`** Read the *arim-digest* password digest file.
- **`rt`** Read the *arim-themes* UI themes file.
- **`cd`** ***`nbr`*** Open directory *nbr* in the shared files viewer and list its files.

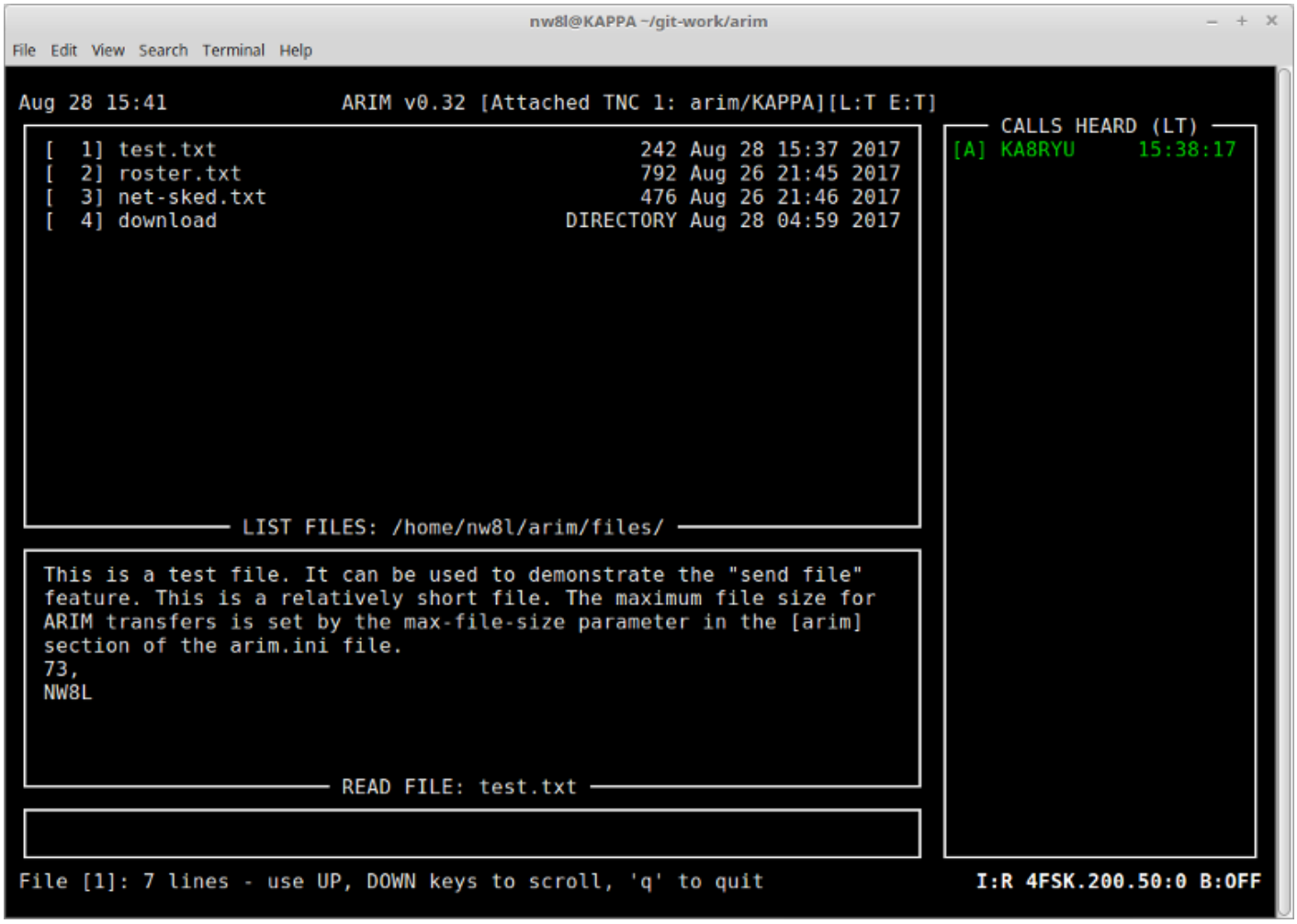
When you open a directory with the **`cd`** command, the listing includes the parent directory (**`..`**) as item number 1.



Enter the command **`cd 1`** to return to it. Press 'q' to quit the file viewer.

Reading files

Text files listed in the shared files viewer can be opened for reading. Press the spacebar to open the command prompt and enter the **`rm`** ***`nbr`*** command where *nbr* is the message number. This opens the file reader.

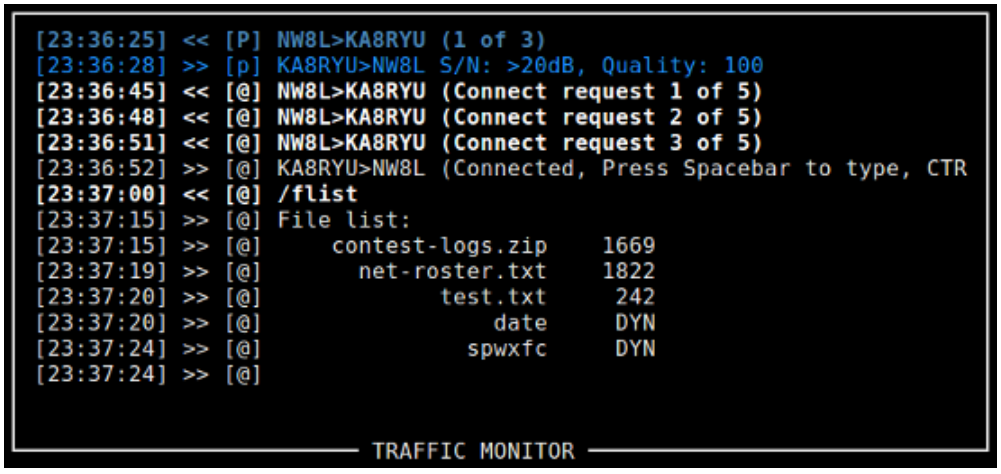


The file number and size in lines are shown in the Status Bar. Use the UP and DOWN arrow keys or the PAGEUP, PAGEDOWN, HOME and END keys to scroll through the file. Press 'q' to quit the file reader and return to the shared files viewer.

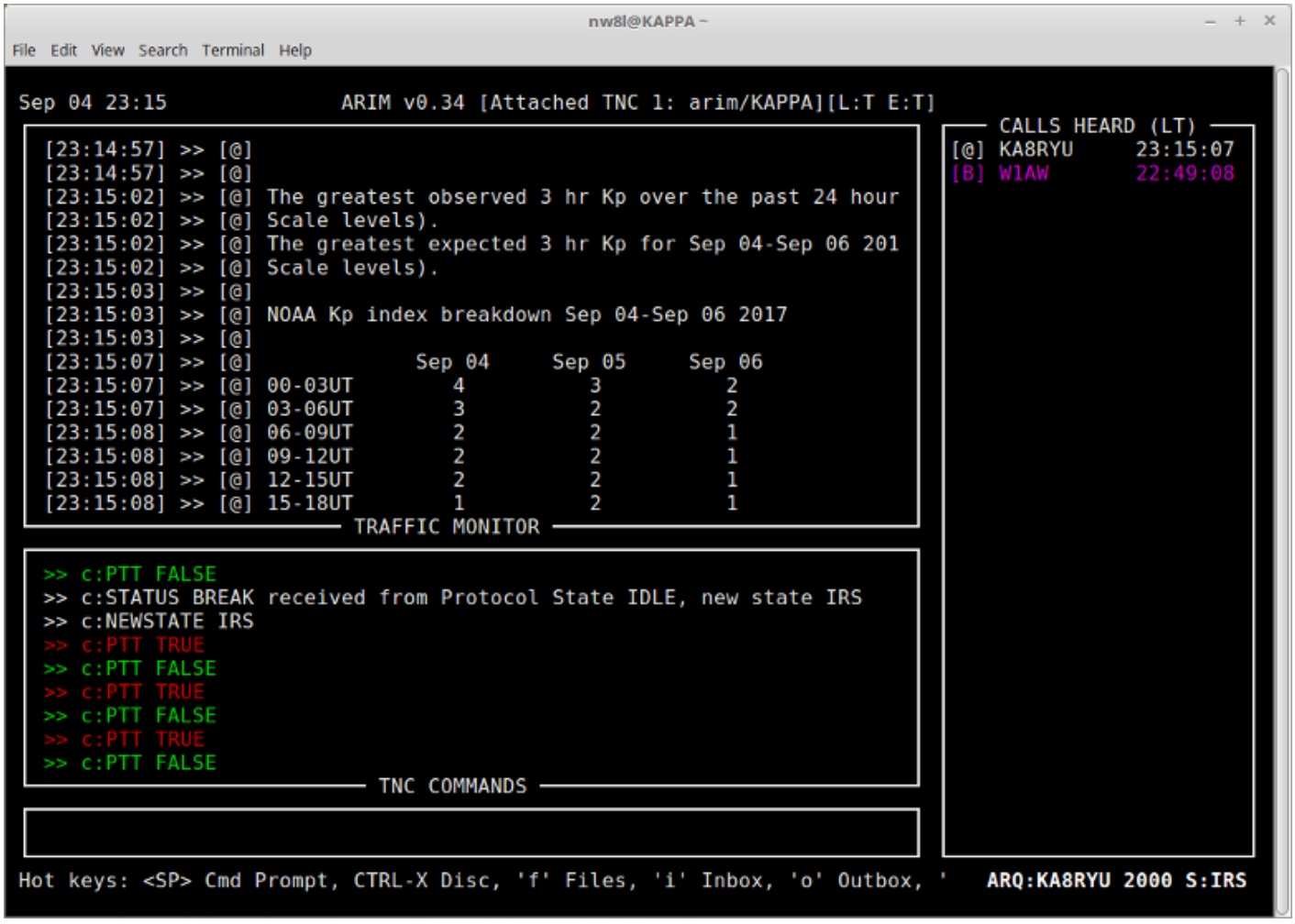
Serving dynamic files

Dynamic files are used to return output from a script or system command executed by ARIM in response to a file query.

Dynamic file aliases are listed in the output of the **`flist`** query for the shared files root directory. Because the size of the output is not known in advance, no file size is shown; instead the `DYN` identifier appears, like this:



Here's a sample of the **`spwxfc`** dynamic file output as seen in an ARQ session:



Dynamic file aliases are defined in the **`[arim]`** section of the *arim.ini* configuration file. The format is *alias:command* where *alias* is the name used to access the file and *command* is the invocation of the command or script, separated by a **`:`** (colon) character. For example:

```
dynamic-file = spwxfc:python /home/nw8l/scripts/forecast.py
```

Make sure that *alias* is unique among the other dynamic file definitions and file names in the shared files directory. Use absolute paths to script files when ARIM is built from source and installed. Relative paths can be used for "portable" binary installations where the script files are contained in same directory as the *arim* executable file. You may define no more than 16 dynamic files.

In response to the query **`sq file alias`**, *command* will be executed in a shell and its output returned in the response. *command* can be a batch file, a script invocation like `python myscript` or a system command like `date` or `uname -a`. The output size in bytes is limited by the **max-file-size** parameter in the **`[arim]`** section of the *arim.ini* configuration file. Errors generated by dynamic file scripts are written to a file named *dyn-file-error.log* in the *log* directory.

RF channel busy detect

ARIM listens for `BUSY` notifications from the ARDOP TNC, which are sent to the host program when a signal is detected in the RF channel. When the channel is busy, ARIM displays the `[RF CHANNEL BUSY]` indicator below the status bar:

ARIM will check the RF channel busy status when the operator initiates any of these actions:

- Send a message to another station or the net
- Send a query to another station
- Ping another station
- Connect to another station for an ARQ session
- Send an unproto message
- Send a beacon

If the RF channel is busy, ARIM will cancel the operation and an error message will be shown on the status bar. If sending a message, the operator is prompted to save it to the outbox to be sent later. This helps prevent collisions between transmissions on a busy channel, for example during net operations.

The TNC does *not* send `BUSY` notifications to the host program during an ARQ session.

The ARDOP TNC's busy detect feature is influenced by these configuration parameters:

- **`busydet`** sets the threshold for the busy detector. The range is 0 to 10, where 0 means *disabled*. Lower values make the detector more sensitive, and higher values make it less sensitive. The default value is 5, which works well for most installations.
- **`arqbw`** sets the bandwidth for ARQ mode. This, to a certain extent, controls the bandwidth of the busy detector. Based on testing and my reading of the source code for the ARDOP TNCs, the 200Hz and 500Hz ARQ bandwidths result in a busy detector bandwidth of a little less than 1kHz; the 2500Hz ARQ bandwidth results in a a busy detector bandwidth of a little over 2.5kHz.

In some cases, ARIM may not receive a `BUSY FALSE` notification when one is expected. This can leave ARIM stuck in the `[RF CHANNEL BUSY]` state, and attempts to start a new operation will fail. If this happens, the operator can recover by pressing the <ESC> key to reset ARIM's busy state.

Logging

Logging can be enabled or disabled in the **`[log]`** section of the *arim.ini* configuration file. These are the traffic log, the debug log and the log for the TNC-Pi9K6 serial TNC.

- The traffic log is enabled by default and logs the content of all ARIM and ARDOP data frames sent and received by the TNC, with timestamping of each entry.
- The debug log is disabled by default. It logs all TNC commands sent and received by the ARIM program as well as exceptions encountered in operation like data wait timeouts, communication errors etc.

- The TNC-Pi9K6 log is disabled by default. It logs the serial communications between ARIM and the serial TNC.

These log files are created in the *log* subdirectory of the ARIM working directory. By default, logging output for all TNCs is directed here.

However, this arrangement doesn't work well if more than one instance of ARIM is running at the same time. In this case, logging output from each instance can be directed to separate directories with settings in the **[tnc]** sections for each TNC in the *arim.ini* configuration file. Settings made here override the default settings in the **[log]** section when that TNC is attached. This prevents intermingling of the logging output from different TNCs in the default log files.

- To configure default log settings in the **[log]** section of the *arim.ini* configuration file:

  Set the **traffic-log** parameter to TRUE to enable traffic logging and to FALSE to disable it.

  Set the **debug-log** parameter to TRUE to enable debug logging and to FALSE to disable it.

  Set the **tncpi9k6-log** parameter to TRUE to enable TNC-Pi9K6 logging and to FALSE to disable it.

- To configure TNC specific log settings in the **[tnc]** section of the *arim.ini* configuration file:

  Use the **log-dir** parameter to specify the directory where log files are located if TNC specific logging is enabled. This can be an absolute path or a relative path rooted in the user's home directory. Max length is 255 characters. Default: the user's home directory.

  Set the **traffic-log** parameter to TRUE to enable traffic logging and to FALSE to disable it.

  Set the **debug-log** parameter to TRUE to enable debug logging and to FALSE to disable it.

  Set the **tncpi9k6-log** parameter to TRUE to enable TNC-Pi9K6 logging and to FALSE to disable it.

  If one or more of **traffic-log**, **debug-log** or **tncpi9k6-log** are set to TRUE, then logging output for this TNC is directed to the directory specified by **log-dir**. If **traffic-log**, **debug-log** and **tncpi9k6-log** are all absent or all set to FALSE, then the global settings in the **[log]** section are used, and logging output is directed to the default log directory.

Another log file, the dynamic file error log, is always enabled. It logs error messages from scripts or programs that generate dynamic file text output.

By default, logs are kept in the *log* subdirectory of the ARIM working directory (or the *$HOME/arim* directory if ARIM is installed) unless directed to a different location for a particular TNC by parameters in the **[tnc]** section of the *arim.ini* configuration file. Logs are automatically rotated every 24 hours. The file name format is *name-YYYYMMDD.log* where *name* is either *traffic*, *debug*, *tncpi9k6* or *dyn-file-error*, followed by the date, e.g. *traffic-20161114.log*.

Using the TNC-Pi9K6 for RPi

When using the TNC-Pi9K6, the TNC must be programmed with the ARDOP Teensy firmware, and the host Raspberry Pi must be configured to assign the proper serial port hardware to the TNC.

Configure the Raspberry Pi: Follow the instructions in the TNC-Pi9K6 TNC-Pi9K6 User Guide to use the *raspi-config* program to enable the serial port. You will address the port as */dev/serial0* in the *arim.ini* configuration file.

Program the TNC-Pi9K6: You must compile and install either the *ARDOP_Teensy* or *ARDOP2_Teensy* firmware into the TNC-Pi9K6. Follow the instructions in the TNC-Pi9K6 User Guide to install the *Arduino IDE* and the *Teensyduino* add-on, compile from source code and program the TNC.

Please note the following:

- Don't use the *TeensyProjects.zip* file from the *tnc-x* website. Instead, download the latest, which is available in the TeensyProjects.zip archive on G8PBQ's download site. This is the up-to-date version.
- To enable TNC-Pi9K6 debug logging, you must comment out line 79 of the *TeensyProjects/libraries/TeensyConfig/TeensyConfigARDOP.h* file. Make `#define MONPORT Serial` look like `//#define MONPORT Serial` before compiling the program. Set the *tncpi9k6-log* parameter in the *arim.ini* configuration file to *TRUE* to enable or *FALSE* to disable logging.
- Normally, the serial port on the RPi GPIO header is used. It's also possible to connect the USB device port on the TNC-Pi9K6 to one of the USB host ports on the RPi. For this to work line 75 of the *TeensyConfigARDOP.h* file must be changed. Make `#define HOSTPORT Serial1` look like `#define HOSTPORT Serial` before compiling. If you do this, you must address serial port */dev/ttyACM0* in the *arim.ini* configuration file.
- The I2C interface with TNC-Pi9K6 is not supported by ARIM at present.