

Amateur kernel networking development

Introducing the HamBSD project

irl

HamBSD

6th December 2019
NorthernRST 2019, Aberdeen, Scotland

\$ whoami

- irl (MM0R0R)

\$ whoami

- irl (MM0ROR)
- Foundation licence: March 2011

\$ whoami

- irl (MM0ROR)
- Foundation licence: March 2011
- MSci Computing Science: August 2014

\$ whoami

- irl (MM0ROR)
- Foundation licence: March 2011
- MSci Computing Science: August 2014
- Intermediate licence: December 2014

\$ whoami

- irl (MM0ROR)
- Foundation licence: March 2011
- MSci Computing Science: August 2014
- Intermediate licence: December 2014
- Debian Developer: July 2015

\$ whoami

- irl (MM0ROR)
- Foundation licence: March 2011
- MSci Computing Science: August 2014
- Intermediate licence: December 2014
- Debian Developer: July 2015
- Full licence: October 2016

\$ whoami

- irl (MM0ROR)
- Foundation licence: March 2011
- MSci Computing Science: August 2014
- Intermediate licence: December 2014
- Debian Developer: July 2015
- Full licence: October 2016
- Scottish Consulate ARC Founding Club Official: November 2019

What is amateur radio?

- A hobby

What is amateur radio?

- A technical hobby

What is amateur radio?

- A technical hobby with strict rules

What is amateur radio?

- A technical hobby with strict rules that requires a licence

What is amateur radio?

- A fun technical hobby with strict rules that requires a licence

What is amateur radio?

- Antennas



<https://en.wikipedia.org/wiki/File:Antenna.jpg> CC BY-SA 3.0

What is amateur radio?

- Antennas
- Electronics



https://en.wikipedia.org/wiki/File:Kit_HEATHKIT_HW8.JPG CC BY-SA 3.0

What is amateur radio?

- Antennas
- Electronics
- History



[https://commons.wikimedia.org/wiki/
File:W6om.png](https://commons.wikimedia.org/wiki/File:W6om.png) CC BY-SA 3.0

What is amateur radio?

- Antennas
- Electronics
- History
- Recreation



[https://commons.wikimedia.org/wiki/
File:Sm0_4s7ab.jpg](https://commons.wikimedia.org/wiki/File:Sm0_4s7ab.jpg) CC BY-SA 3.0

What is amateur radio?

- Antennas
- Electronics
- History
- Recreation
- Sport



[https://en.wikipedia.org/wiki/File:
Frg01_hst.jpg](https://en.wikipedia.org/wiki/File:Frg01_hst.jpg) CC BY-SA 3.0

What is amateur radio?

- Antennas
- Electronics
- History
- Recreation
- Sport
- Space



[https://commons.wikimedia.org/wiki/
File:ISS-24_Doug_Wheelock_uses_ham_
radio_system_1.jpg](https://commons.wikimedia.org/wiki/File:ISS-24_Doug_Wheelock_uses_ham_radio_system_1.jpg) NASA

What is amateur radio?

- Antennas
- Electronics
- History
- Recreation
- Sport
- Space
- Technique



Stone Mountain radio club

What is amateur radio?

- Antennas
- Electronics
- History
- Recreation
- Sport
- Space
- Technique
- EmComm



<https://twitter.com/G1SSR>

What is amateur radio?

- Antennas
- Electronics
- History
- Recreation
- Sport
- Space
- Technique
- EmComm
- Computing



[https://screenshots.debian.net/
package/fldigi](https://screenshots.debian.net/package/fldigi)

What is amateur radio?

- Antennas
- Electronics
- History
- Recreation
- Sport
- Space
- Technique
- EmComm
- Computing
- Packet Networking



<https://ei2gyb.blogspot.com/2018/11/donega-packet-radio-network-part2.html>

What is amateur radio?

- Antennas
- Electronics
- History
- Recreation
- Sport
- Space
- Technique
- EmComm
- Computing
- Packet Networking
- Internetworking



[https://commons.wikimedia.org/wiki/
File:HamNet_DBOMOT.jpg](https://commons.wikimedia.org/wiki/File:HamNet_DBOMOT.jpg) CC BY-SA 4.0

Automatic Packet Reporting System

- real time digital communications

Automatic Packet Reporting System

- real time digital communications
- information of immediate value

Automatic Packet Reporting System

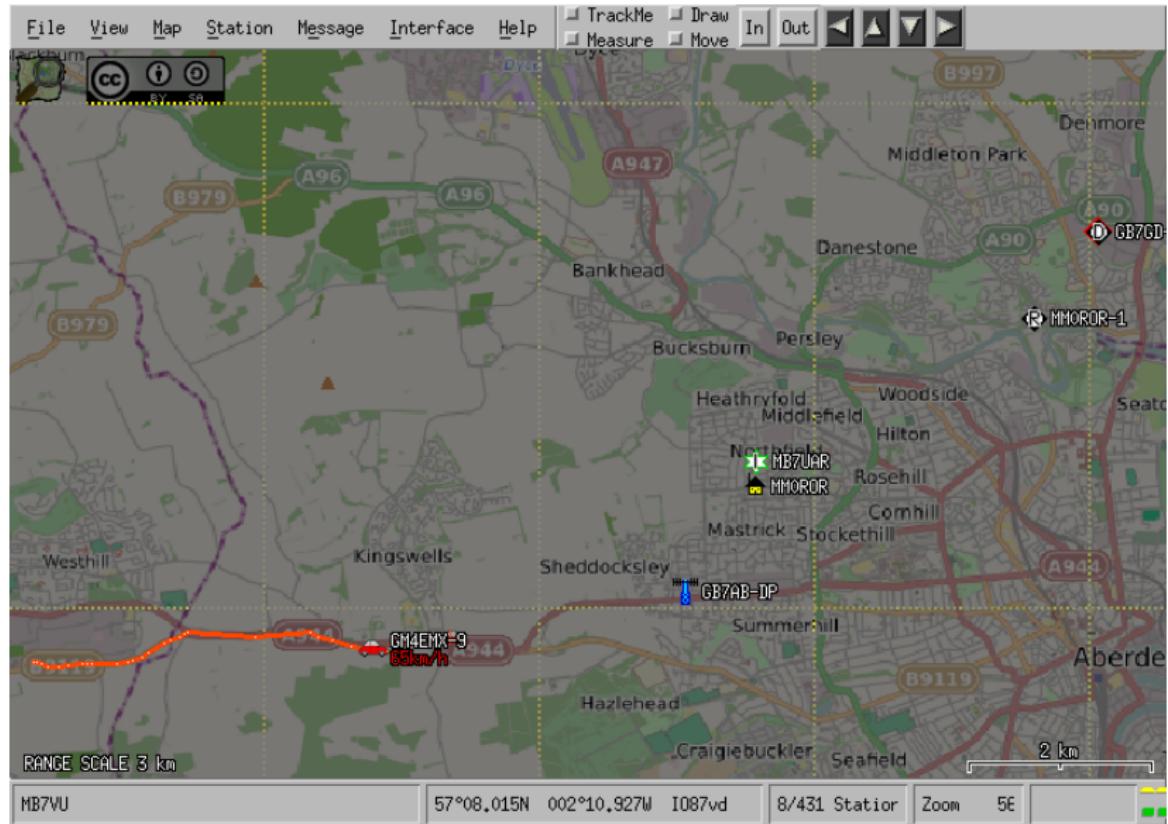
- real time digital communications
- information of immediate value
- regarding the local area

Automatic Packet Reporting System

- real time digital communications
- information of immediate value
- regarding the local area



Automatic Packet Reporting System



Automatic Packet Reporting System

One Way Trackers

- takes GPS data



https://commons.wikimedia.org/wiki/File:Byonics_AI0.jpg

Automatic Packet Reporting System

One Way Trackers

- takes GPS data
- converts to APRS packet



https://commons.wikimedia.org/wiki/File:Byonics_AI0.jpg

Automatic Packet Reporting System

One Way Trackers

- takes GPS data
- converts to APRS packet
- transmit on pre-agreed frequency (144.8MHz)



https://commons.wikimedia.org/wiki/File:Byonics_AI0.jpg

Automatic Packet Reporting System

One Way Trackers

- takes GPS data
- converts to APRS packet
- transmit on pre-agreed frequency (144.8MHz)
- other amateurs see where you are



https://commons.wikimedia.org/wiki/File:Byonics_AI0.jpg

Automatic Packet Reporting System

Two Way Trackers

- takes GPS data



Automatic Packet Reporting System

Two Way Trackers

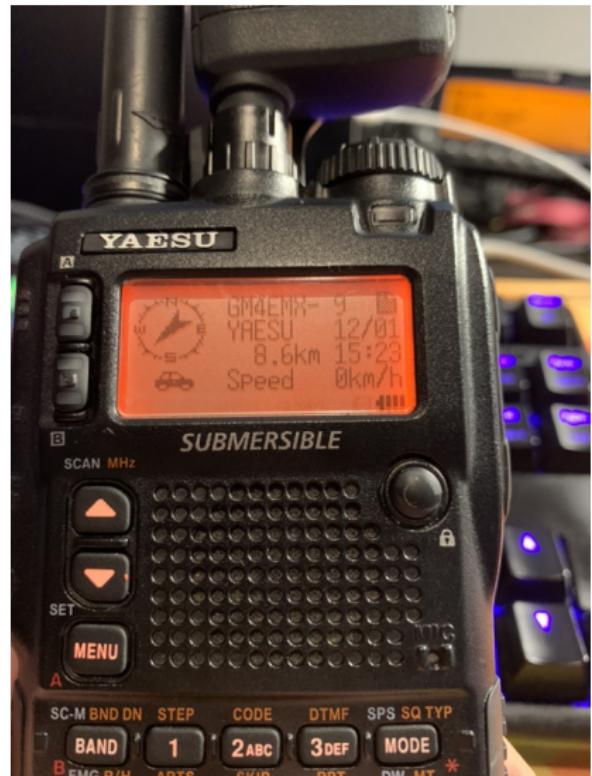
- takes GPS data
- converts to APRS packet



Automatic Packet Reporting System

Two Way Trackers

- takes GPS data
- converts to APRS packet
- transmit on pre-agreed frequency (144.8MHz)



Automatic Packet Reporting System

Two Way Trackers

- takes GPS data
- converts to APRS packet
- transmit on pre-agreed frequency (144.8MHz)
- also receives packets from others



Automatic Packet Reporting System

Two Way Trackers

- takes GPS data
- converts to APRS packet
- transmit on pre-agreed frequency (144.8MHz)
- also receives packets from others
- shows where other amateurs are in relation to you



Automatic Packet Reporting System

Two Way Messaging

- takes a message



Automatic Packet Reporting System

Two Way Messaging

- takes a message
- converts to APRS packet



Automatic Packet Reporting System

Two Way Messaging

- takes a message
- converts to APRS packet
- transmit on pre-agreed frequency (144.8MHz)



Automatic Packet Reporting System

Two Way Messaging

- takes a message
- converts to APRS packet
- transmit on pre-agreed frequency (144.8MHz)
- also receives packets from others



Automatic Packet Reporting System

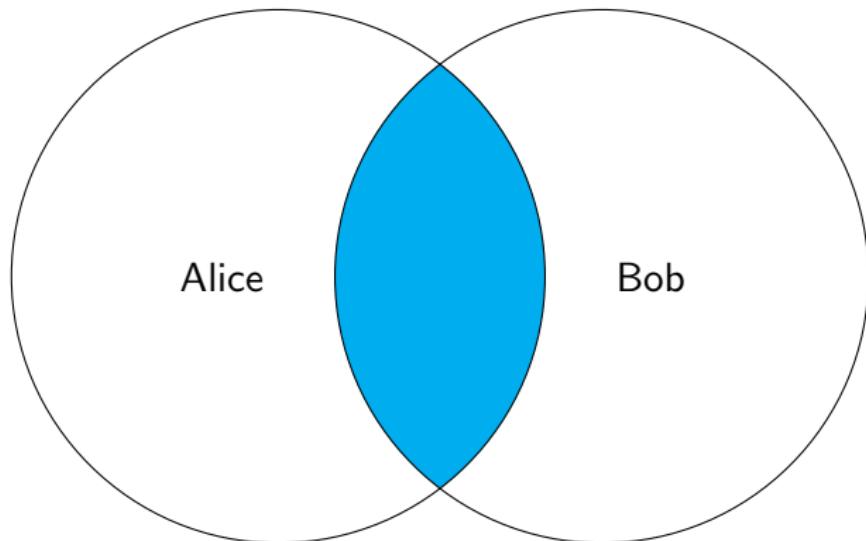
Two Way Messaging

- takes a message
- converts to APRS packet
- transmit on pre-agreed frequency (144.8MHz)
- also receives packets from others
- enables two-way text messaging



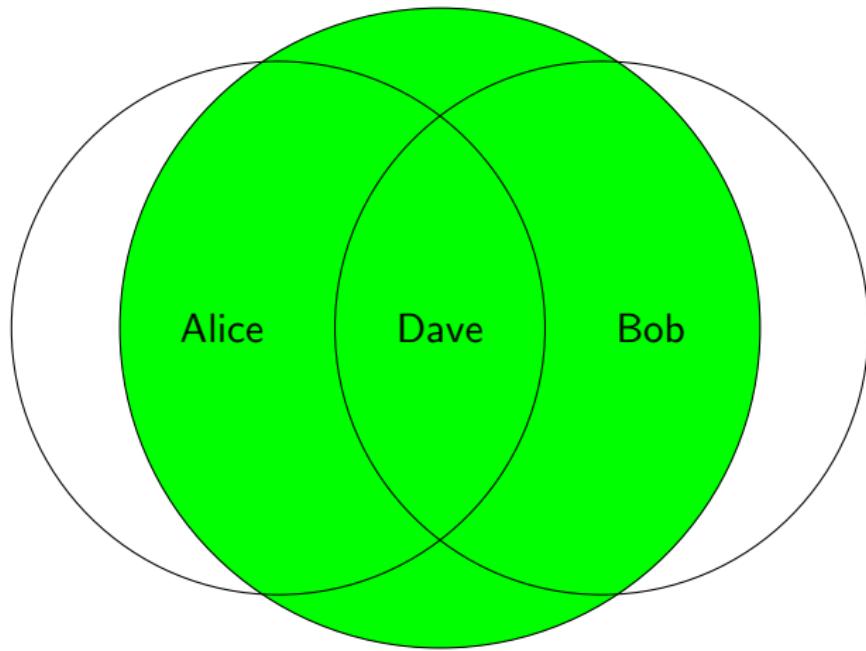
Automatic Packet Reporting System

Digipeating



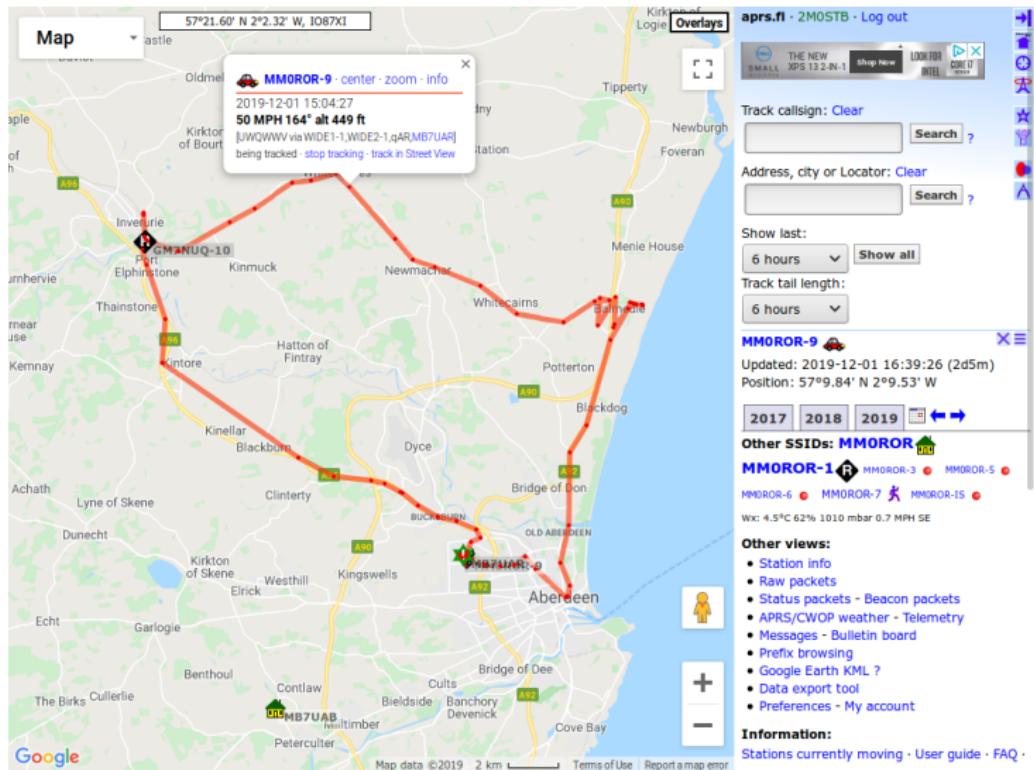
Automatic Packet Reporting System

Digipeating



Automatic Packet Reporting System

APRS-IS and Internet Gateways (IGates)



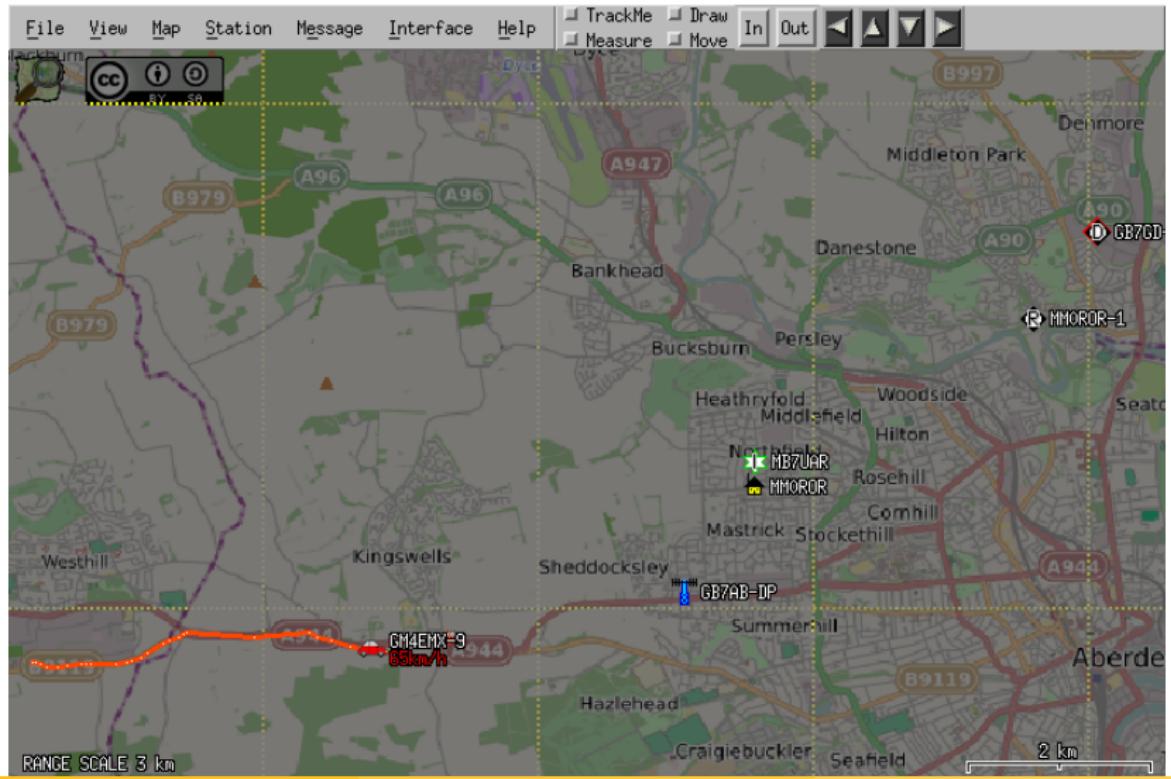
Automatic Packet Reporting System

APRS-IS and Internet Gateways (IGates)



Automatic Packet Reporting System

Software - Xastir



Automatic Packet Reporting System

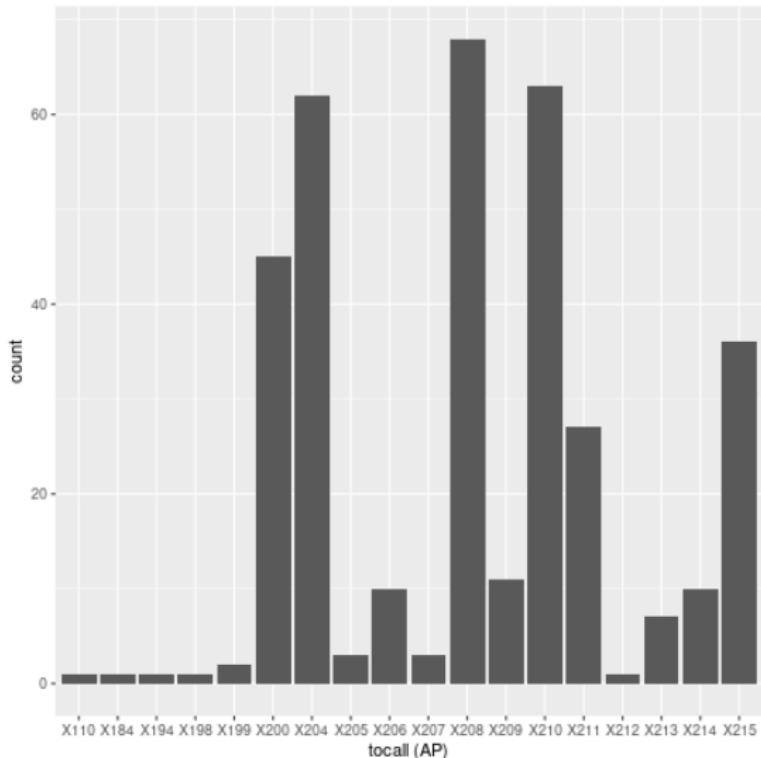
Software - Dire Wolf

```
● direwolf.service - DireWolf is a software "soundcard" modem/TNC and APRS decoder
  Loaded: loaded (/lib/systemd/system/direwolf.service; disabled; vendor preset: enabled)
  Active: active (running) since Fri 2019-11-22 13:55:33 GMT; 1 weeks 4 days ago
    Docs: man:direwolf
 Main PID: 3075 (direwolf)
   Tasks: 12 (limit: 2200)
  Memory: 1.3M
 CGroup: /system.slice/direwolf.service
         └─3075 /usr/bin/direwolf -c /etc/direwolf.conf

Dec 03 18:30:35 raspberrypi direwolf[3075]: [ig] MB7UAR>APDW14:;GB3GN      *111111z5701.06N/0022
Dec 03 18:30:35 raspberrypi direwolf[3075]: [ig] MB7UAR>APDW14:;GB3NG      *111111z5736.13N/0020
Dec 03 18:40:35 raspberrypi direwolf[3075]: [ig] MB7UAR>APDW14:!5709.89NI00209.67W#Northfield
Dec 03 18:40:35 raspberrypi direwolf[3075]: [ig] MB7UAR>APDW14:;GB3GN      *111111z5701.06N/0022
Dec 03 18:40:35 raspberrypi direwolf[3075]: [ig] MB7UAR>APDW14:;GB3NG      *111111z5736.13N/0020
Dec 03 18:40:35 raspberrypi direwolf[3075]: [0L] MB7UAR>APDW14,WIDE1-1,WIDE2-1:;GB3GN      *1111
Dec 03 18:50:34 raspberrypi direwolf[3075]: [0L] MB7UAR>APDW14,WIDE1-1,WIDE2-1:;GB3NG      *1111
Dec 03 18:50:35 raspberrypi direwolf[3075]: [ig] MB7UAR>APDW14:!5709.89NI00209.67W#Northfield
Dec 03 18:50:35 raspberrypi direwolf[3075]: [ig] MB7UAR>APDW14:;GB3GN      *111111z5701.06N/0022
Dec 03 18:50:35 raspberrypi direwolf[3075]: [ig] MB7UAR>APDW14:;GB3NG      *111111z5736.13N/0020
lines 1-20/20 (END)
```

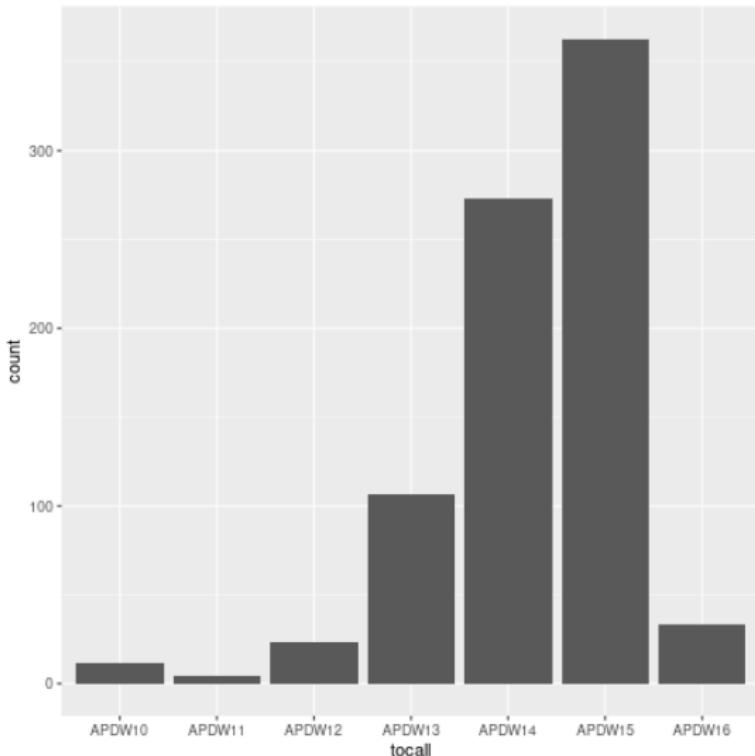
Automatic Packet Reporting System

Versions of Xastir seen in about 30 minutes one evening in December



Automatic Packet Reporting System

Versions of Dire Wolf seen in about 30 minutes one evening in December



Automatic Packet Reporting System

Not Internet Safe

- reminds me of SS7 or BGP

Automatic Packet Reporting System

Not Internet Safe

- reminds me of SS7 or BGP
- originally no authentication built-in

Automatic Packet Reporting System

Not Internet Safe

- reminds me of SS7 or BGP
- originally no authentication built-in
- generally passes on anything received

Automatic Packet Reporting System

Not Internet Safe

- reminds me of SS7 or BGP
- originally no authentication built-in
- generally passes on anything received
- minimal filtering

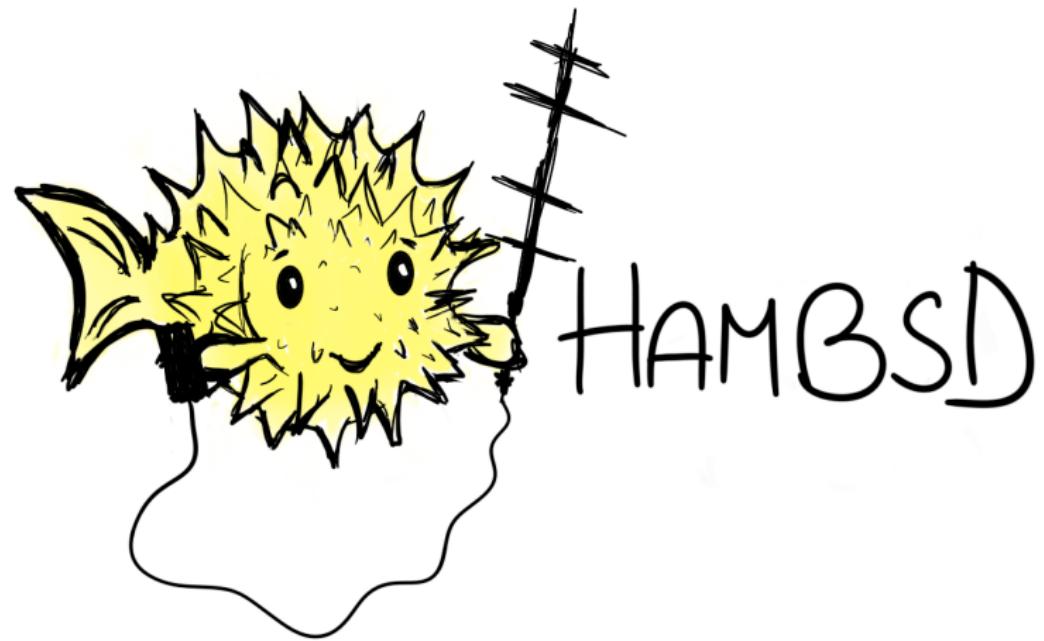
Automatic Packet Reporting System

Not Internet Safe

- reminds me of SS7 or BGP
- originally no authentication built-in
- generally passes on anything received
- minimal filtering
- plaintext transport (though TLS is supported on *some* servers)

HamBSD

Can I do better?



HamBSD

Built on OpenBSD

- forked from NetBSD in 2005

HamBSD

Built on OpenBSD

- forked from NetBSD in 2005
- secure by default

HamBSD

Built on OpenBSD

- forked from NetBSD in 2005
- secure by default
- emphasis on code quality

HamBSD

Built on OpenBSD

- forked from NetBSD in 2005
- secure by default
- emphasis on code quality
- known for its high-quality documentation

HamBSD

Goals

- KISS TNC support

HamBSD

Goals

- KISS TNC support
- AX.25 networking support

HamBSD

Goals

- KISS TNC support
- AX.25 networking support
- APRS application support

HamBSD

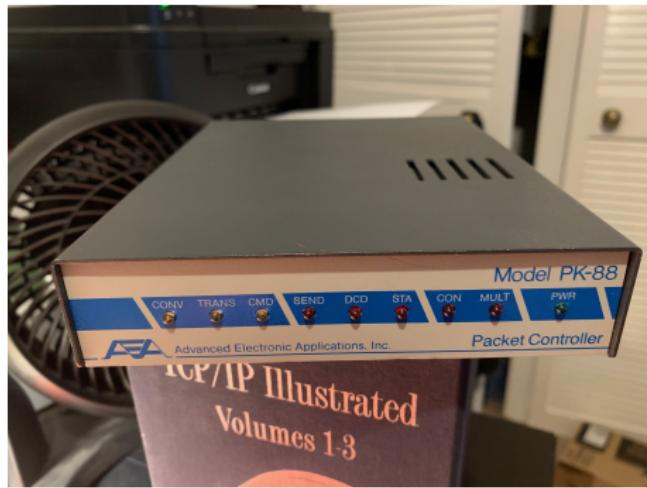
Goals

- KISS TNC support
- AX.25 networking support
- APRS application support
- APRS-IS compatibility

HamBSD

kiss(4): KISS TNC Support¹

- simple framing protocol
- four special characters:
 - FEND (frame end)
 - FESC (frame escape)
 - TFEND (transposed frame end)
 - TFESC (transposed frame escape)
- frames are sent separated by FEND

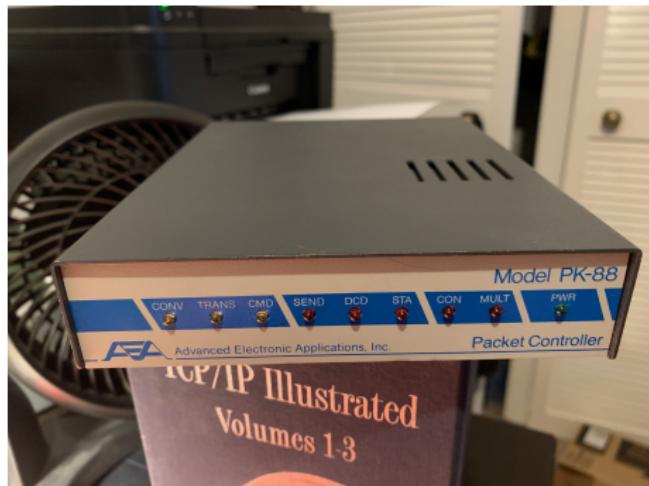


¹Mike Chepponis, K3MC and Phil Karn, KA9Q, The KISS TNC: A simple Host-to-TNC communications protocol, ARRL 6th Computer Networking Conference, Redondo Beach, CA, United States., 1987.

HamBSD

kiss(4): KISS TNC Support¹

- if FEND or FESC appear in the frame, instead send a two byte sequence:
 - FEND is replaced with FESC TFEND
 - FESC is replaced with FESC TFESC
- FEND always means the end of the frame
- FESC always means the start of an escape sequence

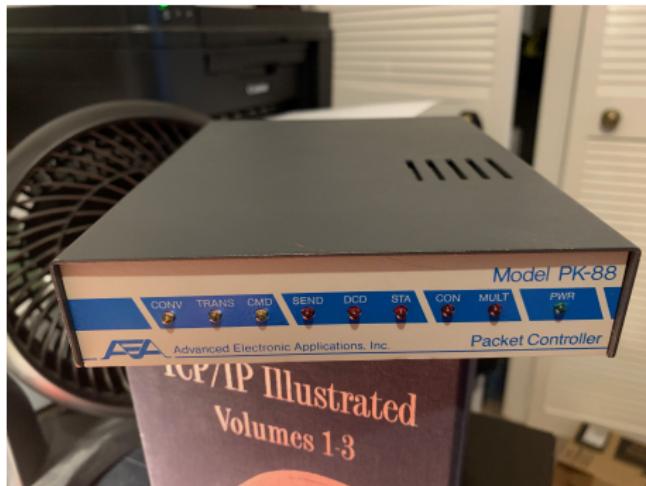


¹Mike Chepponis, K3MC and Phil Karn, KA9Q, The KISS TNC: A simple Host-to-TNC communications protocol, ARRL 6th Computer Networking Conference, Redondo Beach, CA, United States., 1987.

HamBSD

kiss(4): KISS TNC Support¹

- the start of every frame is a command
- we will only look at the "data" command, which means the frame contains packet data



¹Mike Cheponis, K3MC and Phil Karn, KA9Q, The KISS TNC: A simple Host-to-TNC communications protocol, ARRL 6th Computer Networking Conference, Redondo Beach, CA, United States., 1987.

HamBSD

kiss(4): KISS TNC Support

```
1 function sendpacket( packetdata ):
2     put(FEND)
3     put(CMDDATA)
4     for byte in packetdata:
5         switch (byte):
6             case FEND:
7                 put(FESC)
8                 put(TFEND)
9             case FESC:
10                put(FESC)
11                put(TFESC)
12            default:
13                put( byte )
14
15    putc(FEND)
```

```
2      putc(KISSFEND, &tp->t_outq);
3      putc(KISSCMD_DATA, &tp->t_outq); /* implicitly port 0 */
4
5      while (m) {
6          register u_char *ep;
7
8          cp = mtod(m, u_char *); ep = cp + m->m_len;
9          while (cp < ep) {
10             /*
11              * Find out how many bytes in the string we can
12              * handle without doing something special.
13             */
14             register u_char *bp = cp;
15
16             while (cp < ep) {
17                 switch (*cp++) {
18                     case KISSFESC:
19                     case KISSFEND:
20                         --cp;
21                         goto out;
22                     }
23                 }
24             }
25         }
26     }
27 }
```

HamBSD

kiss(4): KISS TNC Support

```
out:  
2   if (cp > bp) {  
3     /*  
4      * Put n characters at once  
5      * into the tty output queue.  
6      */  
7     if (b_to_q((char *)bp, cp - bp,  
8                 &tp->t_outq))  
9       break;  
10    sc->sc_if.if_oBYTES += cp - bp;  
 }
```

HamBSD

kiss(4): KISS TNC Support

```
1  /*
2   * If there are characters left in the mbuf,
3   * the first one must be special..
4   * Put it out in a different form.
5   */
6  if (cp < ep) {
7      if (putc(KISSFESC, &tp->t_outq))
8          break;
9      if (*cp++ == KISSFESC ?
10          KISSTFESC : KISSTFEND,
11          &tp->t_outq)) {
12          (void) unputc(&tp->t_outq);
13          break;
14      }
15      sc->sc_if.if_oBYTES += 2;
16  }
```

HamBSD

kiss(4): KISS TNC Support

```
1     m = m_free(m);  
}  
}
```

HamBSD

kiss(4): KISS TNC Support

- sending frames

HamBSD

kiss(4): KISS TNC Support

- sending frames
- receiving frames (slightly trickier, but not much)

HamBSD

kiss(4): KISS TNC Support

- sending frames
- receiving frames (slightly trickier, but not much)
- how to handle frames?

HamBSD

ax25(4): AX.25 Networking Support

- use the kernel networking subsystem

HamBSD

ax25(4): AX.25 Networking Support

- use the kernel networking subsystem
- need a network interface: axkiss(4)

HamBSD

ax25(4): AX.25 Networking Support

- use the kernel networking subsystem
- need a network interface: axkiss(4)

```
# ifconfig axkiss0
2 flags=8843<UP,BROADCAST,RUNNING,SIMPLEX> mtu 610
4 lladdr MM0ROR-6
6 index 2 priority 0 llprio 3
 media: KISS TNC (1200 baud)
 status: active
```

HamBSD

ax25(4): AX.25 Networking Support

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	MM0R0R-7	UWQPWV	APRS	47	MIC-E Lat: 5710.76N Lo
2	180.800960	MM0R0R-7	UWQPWU	APRS	47	MIC-E Lat: 5710.75N Lo
3	361.557527	MM0R0R-7	UWQPWU	APRS	47	MIC-E Lat: 5710.75N Lo
4	1084.155021	MM0R0R-7	UWQPWV	APRS	47	MIC-E Lat: 5710.76N Lo
5	1264.738934	MM0R0R-7	UWQPWV	APRS	47	MIC-E Lat: 5710.76N Lo
6	1445.436443	MM0R0R-7	UWQPWV	APRS	47	MIC-E Lat: 5710.76N Lo
7	1626.122182	MM0R0R-7	UWQPWV	APRS	47	MIC-E Lat: 5710.76N Lo

Frame 1: 47 bytes on wire (376 bits), 47 bytes captured (376 bits) on interface eka0
AX.25, Src: MM0R0R-7, Dst: UWQPWV, Ver: V2.0+
AX.25 No Layer 3 - (APRS)
Current Mic-E Data (not used in TM-D700)
Data Type Indicator:
Current Mic-E: Lat: 5710.76N Long: 00206.65W, Cse: 730, Spd: 0, SSID: 0, Msg
Destination Address: Lat 5710.76, Msg A 1, Msg B 1, Msg C 1, N/S N, Long 0
Longitude degrees: 0x78
Longitude minutes: 0x5e
Longitude hundredths of minutes: 0x5d
Speed (hundreds & tens): 0x6c
Speed (tens), Course (hundreds): 0x20

0000	aa ae a2 a0 ae ac	60	9a	9a	60	a4	9e	a4	ee	ae	92						
0010	88	8a	62	40	62	ae	92	88	8a	64	40	63	03	f0	60	78	..b@b..	d@c..	x
0020	5e	5d	6c	20	65	5b	2f	60	22	34	5f	7d	5f	20	0d	^]1	e[/	"4_	.

HamBSD

ax25(4): AX.25 Networking Support

BPF(4)

Device Drivers Manual

BPF(4)

NAME

bpf — Berkeley Packet Filter

SYNOPSIS

pseudo-device **bpfilter**

DESCRIPTION

The Berkeley Packet Filter provides a raw interface to data link layers in a protocol-independent fashion. All packets on the network, even those destined for other hosts, are accessible through this mechanism.

The packet filter appears as a character special device, `/dev/bpf`. After opening the device, the file descriptor must be bound to a specific network interface with the `IOSETIF` [ioctl\(2\)](#). A given interface can be shared between multiple listeners, and the filter underlying each descriptor will see an identical packet stream.

Associated with each open instance of a **bpf** file is a user-settable packet filter. Whenever a packet is received by an interface, all file descriptors listening on that interface apply their filter. Each descriptor that accepts the packet receives its own copy.

Reads from these files return the next group of packets that have matched the filter. To improve performance, the buffer passed to read must be the same size as the buffers used internally by **bpf**. This size is returned by the `BIOCGBLEN` [ioctl\(2\)](#) and can be set with `BIOCSBLEN`. Note that an individual packet larger than this size is necessarily truncated.

A packet can be sent out on the network by writing to a **bpf** file descriptor. Each descriptor can also have a user-settable filter for controlling the writes. Only packets matching the filter are sent out of the interface. The writes are unbuffered, meaning only one packet can be processed per write.

Once a descriptor is configured, further changes to the configuration can be prevented using the `BIOCLOCK` [ioctl\(2\)](#).

HamBSD

ax25(4): AX.25 Networking Support

BPF(4)

Device Drivers Manual

BPF(4)

NAME

bpf — Berkeley Packet Filter

SYNOPSIS

pseudo-device **bpfilter**

DESCRIPTION

The Berkeley Packet Filter provides a raw interface to data link layers in a protocol-independent fashion. All packets on the network, even those destined for other hosts, are accessible through this mechanism.

The packet filter appears as a character special device, `/dev/bpf`. After opening the device, the file descriptor must be bound to a specific network interface with the `IOSETIF` [ioctl\(2\)](#). A given interface can be shared between multiple listeners, and the filter underlying each descriptor will see an identical packet stream.

Associated with each open instance of a **bpf** file is a user-settable packet filter. Whenever a packet is received by an interface, all file descriptors listening on that interface apply their filter. Each descriptor that accepts the packet receives its own copy.

Reads from these files return the next group of packets that have matched the filter. To improve performance, the buffer passed to read must be the same size as the buffers used internally by **bpf**. This size is returned by the `BIOCGBLEN` [ioctl\(2\)](#) and can be set with `BIOCSBLEN`. Note that an individual packet larger than this size is necessarily truncated.

A packet can be sent out on the network by writing to a **bpf** file descriptor. Each descriptor can also have a user-settable filter for controlling the writes. Only packets matching the filter are sent out of the interface. The writes are unbuffered, meaning only one packet can be processed per write.

Once a descriptor is configured, further changes to the configuration can be prevented using the `BIOCLOCK` [ioctl\(2\)](#).

HamBSD

ax25(4): AX.25 Networking Support

BPF(4)

Device Drivers Manual

BPF(4)

NAME

bpf — Berkeley Packet Filter

SYNOPSIS

pseudo-device **bpfilter**

DESCRIPTION

The Berkeley Packet Filter provides a raw interface to data link layers in a protocol-independent fashion. All packets on the network, even those destined for other hosts, are accessible through this mechanism.

The packet filter appears as a character special device, `/dev/bpf`. After opening the device, the file descriptor must be bound to a specific network interface with the `ioctl(2)`. A given interface can be shared between multiple listeners, and the filter underlying each descriptor will see an identical packet stream.

Associated with each open instance of a **bpf** file is a user-settable packet filter. Whenever a packet is received by an interface, all file descriptors listening on that interface apply their filter. Each descriptor that accepts the packet receives its own copy.

Reads from these files return the next group of packets that have matched the filter. To improve performance, the buffer passed to read must be the same size as the buffers used internally by **bpf**. This size is returned by the `BIOCGBLEN ioctl(2)` and can be set with `BIOCBLEN`. Note that an individual packet larger than this size is necessarily truncated.

A packet can be sent out on the network by writing to a **bpf** file descriptor. Each descriptor can also have a user-settable filter for controlling the writes. Only packets matching the filter are sent out of the interface. The writes are unbuffered, meaning only one packet can be processed per write.

Once a descriptor is configured, further changes to the configuration can be prevented using the `BIOCLOCK ioctl(2)`.

HamBSD

aprsd(8): Automatic Packet Reporting System Daemon

- send position reports

HamBSD

aprsd(8): Automatic Packet Reporting System Daemon

- send position reports
 - OpenBSD already has GPS support

HamBSD

aprsd(8): Automatic Packet Reporting System Daemon

- send position reports
 - OpenBSD already has GPS support
- listen for reports

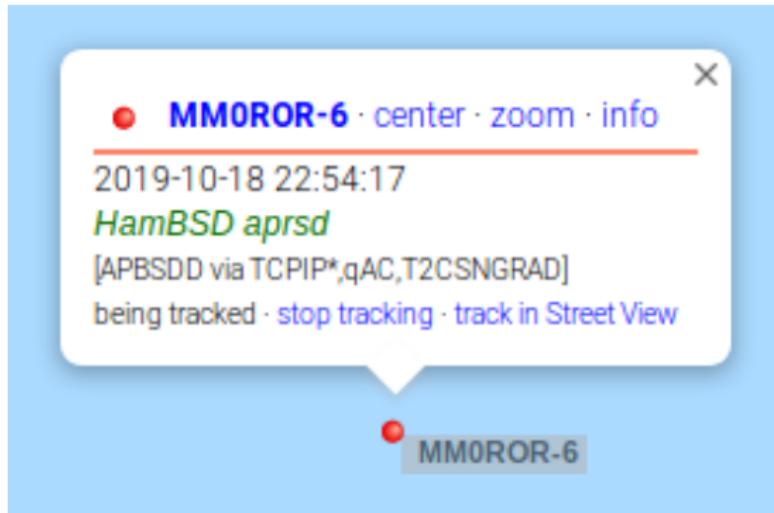
HamBSD

aprsd(8): Automatic Packet Reporting System Daemon

- send position reports
 - OpenBSD already has GPS support
- listen for reports
- digipeat reports

HamBSD

aprsd(8): Automatic Packet Reporting System Daemon



HamBSD

aprsd(8): Automatic Packet Reporting System Daemon

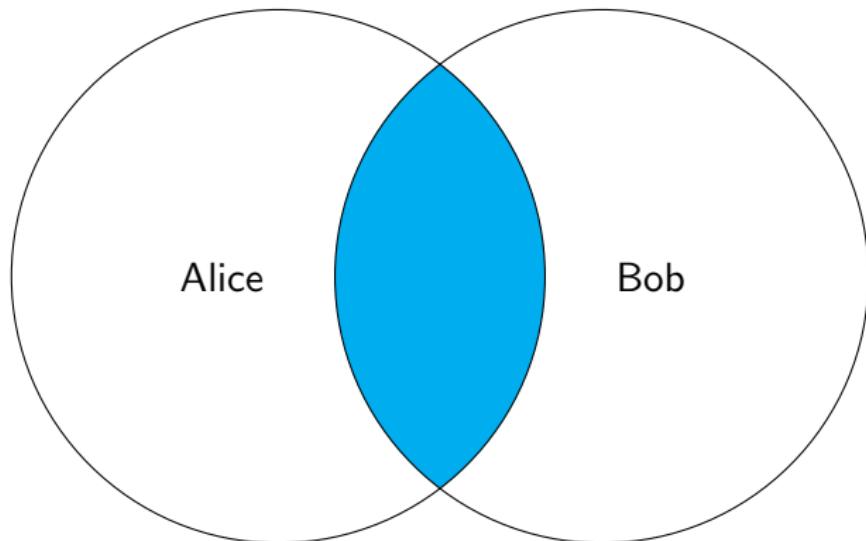
/etc/aprsd.conf:

```
gps="nmea0"
beacon position sensor $gps comment "HamBSD aprsd"
```

```
# rcctl set aprsd flags -i axkiss0
# rcctl start aprsd
```

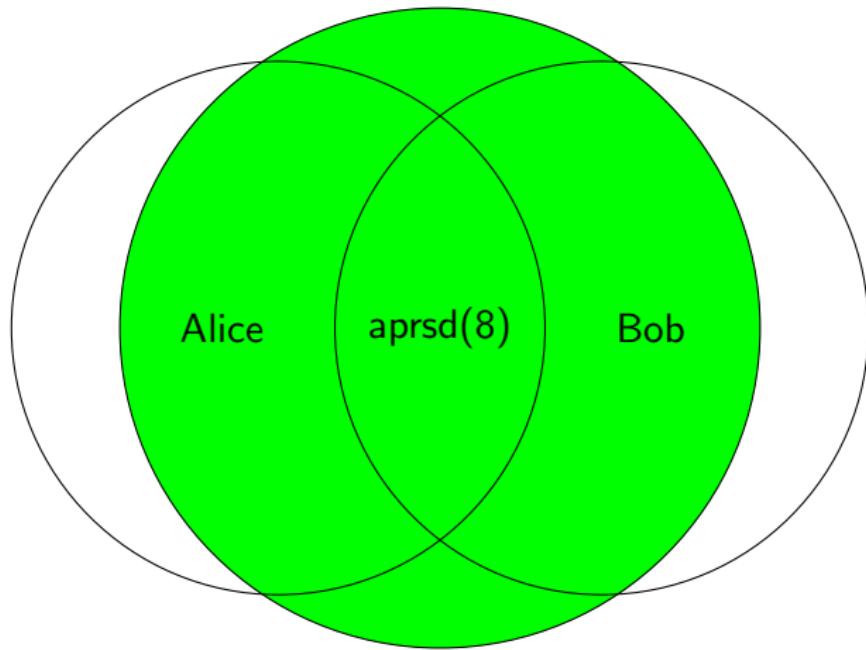
HamBSD

aprsd(8): Automatic Packet Reporting System Daemon



HamBSD

aprsd(8): Automatic Packet Reporting System Daemon



HamBSD

aprsd(8): Automatic Packet Reporting System Daemon

/etc/aprsd.conf:

```
2 gps="nmea0"
beacon position sensor $gps comment "HamBSD aprsd"

4 digipeat {
    5     pass in on axkiss0
    6     pass out on axkiss0
}
```

HamBSD

aprsisd(8): APRS-IS Daemon

- using a new axtap(4) network interface

HamBSD

aprsisd(8): APRS-IS Daemon

- using a new axtap(4) network interface
 - complete rip-off of tun(4)/tap(4) driver using the ax25(4) functions instead of Ethernet

HamBSD

aprsis(8): APRS-IS Daemon

- using a new axtap(4) network interface
 - complete rip-off of tun(4)/tap(4) driver using the ax25(4) functions instead of Ethernet
- make a TCP connection to APRS-IS

HamBSD

aprsisd(8): APRS-IS Daemon

- using a new axtap(4) network interface
 - complete rip-off of tun(4)/tap(4) driver using the ax25(4) functions instead of Ethernet
- make a TCP connection to APRS-IS
 - later this will only support TLS

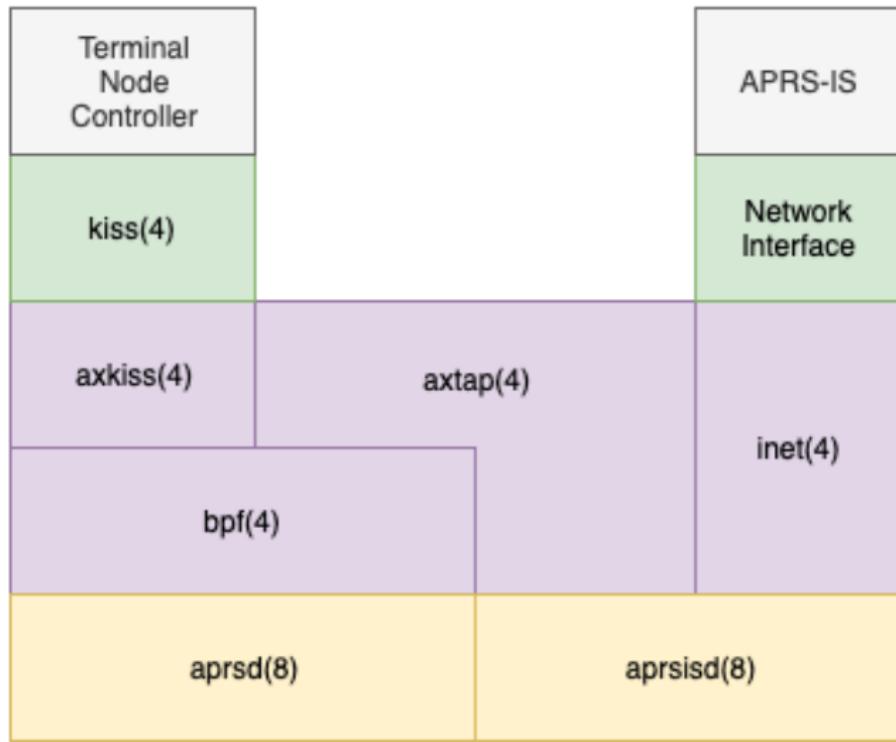
HamBSD

aprsisd(8): APRS-IS Daemon

- using a new axtap(4) network interface
 - complete rip-off of tun(4)/tap(4) driver using the ax25(4) functions instead of Ethernet
- make a TCP connection to APRS-IS
 - later this will only support TLS
- feeds packets through the axtap(4) interface, like a VPN would

HamBSD

aprsisd(8): APRS-IS Daemon



HamBSD

What next?

- rock-solid digipeating

HamBSD

What next?

- rock-solid digipeating
- recording heard station reports

HamBSD

What next?

- rock-solid digipeating
- recording heard station reports
- privilege separation for parsing

HamBSD

What next?

- rock-solid digipeating
- recording heard station reports
- privilege separation for parsing
- room for innovation

APRS Looking Glass

mb7uar.hamsd.org

show heard

Station	Timestamp	Position	Message
MM0ROR-6	2019-11-13 08:24:28	57°10.74N 2°06.63W	Bridge of Don rx-only IGate
GM4EMX-9	2019-11-13 08:19:22	57°07.45N 2°08.25W	Sysop Kit /m Aberdeen area
MM0ROR-7	2019-11-12 21:09:02	57°10.74N 2°06.62W	---

aprslg is an APRS looking glass for [HamBSD](#).

Copyright (C) 2019 Iain R. Learmonth.

HamBSD

Beyond APRS

- IPv4 over AX.25

HamBSD

Beyond APRS

- IPv4 over AX.25
- 6LoWPAN over AX.25

HamBSD

Beyond APRS

- IPv4 over AX.25
- 6LoWPAN over AX.25
- TCP over AX.25

HamBSD

Beyond APRS

- IPv4 over AX.25
- 6LoWPAN over AX.25
- TCP over AX.25
- Amateur Wi-Fi

HamBSD

Beyond APRS

- IPv4 over AX.25
- 6LoWPAN over AX.25
- TCP over AX.25
- Amateur Wi-Fi
- Amateur LoRaWAN

HamBSD

Beyond APRS

- IPv4 over AX.25
- 6LoWPAN over AX.25
- TCP over AX.25
- Amateur Wi-Fi
- Amateur LoRaWAN
- Amateur Pagers

HamBSD

Beyond APRS

- IPv4 over AX.25
- 6LoWPAN over AX.25
- TCP over AX.25
- Amateur Wi-Fi
- Amateur LoRaWAN
- Amateur Pagers
- Amateur Sattelite Comms

HamBSD

How Can I Help?



Iain R. Learmonth

is creating free, secure, robust, and open source amateur radio software

SELECT A MEMBERSHIP LEVEL

\$3 or more per month

\$3

PER MONTH

Let me know that you think what I'm doing is useful, to motivate me to do more of it. You'll get a monthly update on what it is that I've been doing.

\$10 or more per month

\$10

PER MONTH

Let me know that you think what I'm doing is really useful, to really motivate me to do more of it. You'll get a monthly update on what it is that I've been doing.

\$25 or more per month

\$25

PER MONTH

Let me know that you think what I'm doing is really useful, to really motivate me to do more of it. You'll get a monthly update on what it is that I've been doing, and once you've been a patron for a

<https://www.patreon.com/fossirl>



HamBSD

How Can I Help?

HamBSD Hardware Wanted

The following list outlines hardware that the HamBSD project could use. There are a number of reasons why developers need hardware, and it is important to first of all realize these reasons:

- To support a specific device, normally one which is more rare or new.
- To be part of their development environment.
- Some of these devices may have quirks that require some investigation and workaround in order to function correctly.
- That said, space or power may be at a premium for some people.

If you do not own these devices, but want to help us, we recommend you search on [eBay](#) for the devices. If you do the bidding and then get the device shipped to us, it really helps us. It is better if developers develop, and skip the bidding process, since any time saved can be spent on improving HamBSD instead.

Contact [Iain Learmonth](#) who will be able to tell you if something is useful, and if so, where to send it.

Please be very clear as to where you are located!! Unless an item is particularly rare, it is probably uneconomical to ship from anywhere outside Europe. (All development currently happens in *Scotland*.)

General Requirements

- Terminal Node Controllers (RS232 or USB)
- USB sound card interfaces
- All-mode transcievers (e.g. FT-897, FT-857 and Icom 706)
- RS232 protocol analyser

<https://hamsd.org/want.html>

HamBSD

How Can I Help?

Come hang out on IRC
<irc://chat.freenode.net/hamsd>

GB1RST

Special Event Station: NorthernRST Hackathon

- 7th December - 8th December
- we'll be operating a special event station from the NorthernRST hackathon
- we'll be around on 2 meters and also (hopefully) on HF
- check APRS for up-to-date information on which band(s) we're on

MS0SCZ

Scottish Consulate ARC Meeting

- 8th December afternoon
- we're a new club
- not sure what we're doing yet

Thanks

<https://hambsd.org/>