Raspberry Pi Hamvoip Allstar
A Method for Switching Wireless Access points

The hamvoip.org wireless setup has the ability to setup numerous wireless access points which are stored in the wpa_supplicant config file. This works fine when moving between physically different access points. For instance you might have a portable node that you carry between a home and vacation location where each location has wireless access. You would setup the home location using wireless and then when you went to the vacation location you would also setup wireless access. In this scenario once setup you could move the portable node back and forth at will and it would find and connect to the wireless access points at each location upon boot.

At times however you may want to force a connect to a specific wireless access point where more than one physical access point is within reach of your location. If all the access points were defined in the wpa_supplicant file it is not always clear which one would take precedence, and it is not always the first defined access point in the file. You can setup a priority using commands in the file but this still does not give you absolute control over the selected access point. Another way to do this would be a method using scripts to dynamically change the wpa_supplicant file and the access point connection. This would also facilitate changes using DTMF functions from your radio. A typical scenario would be if you had access to your home wireless and your cellphone hotspot at the same physical location and you wanted to be able to switch between them at will. Here is an example.

The first thing you would do is setup the access points using the Hamvoip Admin Menu. For each Wi-Fi you wish to set up:

1. Select Menu Item No. 7 “Configure the WiFi Interface Networking”

2. From the Wireless Menu, Select Item 1 “Setup Wireless SSID and passphrase”.

3. Be sure to say YES to Clear the wpa-supplicant_file.

4. Scan and selection your wifi access point from the SSID list. Enter Passphrase.

5. Reboot the system. Disconnect the wired cat6 connector and verify that you can login using the wi-fi access point.

6. We now have one entry in the **wpa_supplicant_custom-wlan0.conf** file. It would look something like this -

```
network={
        ssid="A-fixed"
        #psk="some-pw" psk=some_random_number
        }
```

The ssid and #psk line would be the SSID and Passphrase you entered. The psk line is a long line of hex numbers generated by the software.

7. Now we need to copy this to a separate file with a name that reflects what this wifi access point is. For example “A-fixed.wifi”:

```
cp  /etc/wpa_supplicant/wpa_supplicant_custom-wlan0.conf  /etc/asterisk/local/A-fixed.wifi
```

8. Repeat Steps 1 through 7 for each separate wifi access point you have.

When done, depending on how many access points you have you will have several files in the /etc/asterick/local directory with each referencing a different wifi access point. For three access points, these might be called:

A-fixed.wifi
B-fixed.wifi
C-portable.wifi

The next step is to create a script in **/etc/asterisk/local** that will swap the wifi access files as required using touch tone codes from your radio. Let’s call the script **wireless-change.sh**

It would look like this:

```
#  !/bin/bash
# Script to change wpa_supplicant files on the fly

if [ -z "$1" ]
then
echo "Wireless definition file parameter missing"
exit 1
else
if ! [ -e "/etc/asterisk/local/$1" ]
```

```
then
echo "Wireless definition file \"$1\" not found"
exit 1
  fi
fi
systemctl stop wpa_supplicant@wlan0.service
cp /etc/asterisk/local/$1 /etc/wpa_supplicant/wpa_supplicant_custom-wlan0.conf
systemctl start wpa_supplicant@wlan0.service
echo "$1 copied to wpa_supplicant@wlan0.service"
if [ $# -eq 2 ]
then
  sleep 10
  /usr/bin/asterisk -rx "rpt fun $2 *A1"
   sleep 7
  /usr/bin/asterisk -rx "rpt fun $2 *A3"
fi

# end of script
```

Save the script and make it executable - **chmod 750 wireless-change.sh**

To execute the script you would call it like this -

/etc/asterisk/local/wireless-change.sh A-fixed.wifi <node> # or B-fixed.wifi or C-portable.wifi

The <node> is the node number WITHOUT the <> and is optional and would say the assigned local IP and then the public IP for the new connection to that node or nothing if left off. The /etc/asterisk/local directory is not in the search path so you would have to use the entire path as shown above.

To call it with DTMF from your radio via Allstar you would put the following in the function section of **/etc/asterisk/rpt.conf** :

[functions]

A20=cmd,/etc/asterisk/local/wireless-change.sh A-fixed.wifi <node>
A21=cmd,/etc/asterisk/local/wireless-change.sh B-fixed.wifi <node>
A22=cmd,/etc/asterisk/local/wireless-change.sh C-portable.wifi <node>

So a DTMF of *A20 would select A-fixed, *A21 B-fixed, and *A22 C-portable.

Note that the way Asterisk Allstar accepts DTMF commands you cannot have both A2 and A21 functions defined. The processor stops at the first defined function code (A2) and would never find (A21) This goes for any defined codes. In this case A2 is probably not defined so this would work but a word of caution when defining.

Example: if a function 85 was defined then any function defined with digits after 85 would be ignored. If a function 853 was defined and then a function 8539 was defined it would not work because the 853 function would be executed first.

The optional <node> is the node number you want to say the local and public IP's on **WITHOUT the <>**

If you had the A1 "say local IP" and A3 "say public IP" functions uncommented in the functions section of rpt.conf a DTMF *A1 on your radio would tell you what local IP address was assigned and what public IP address was assigned after a change. The last four lines in the above script do this at each access point change but would do nothing if the A1 and A3 functions were not uncommented in rpt.conf.

Keep in mind that function codes, file names, script names are all fictitiously picked for this example. You can use your own scheme.

Also note that when bringing up nodes with new public IP addresses, which would happen when changing from a home wifi to a cell hotspot, you will not be able to connect to another node until your new IP address has propagated to that node. If your node uses Hamvoip registration this will happen very quickly but if not it can take up to 10 minutes. Until both ends have each other's correct IP address you will get "connection failed" messages. If you are connected somewhere using a permanent (*73) connect before you make the change it will reconnect automatically using the new path when the circuit becomes available.

While this example works it is meant to be a starting point for experimentation. Feel free to customize and add your own ideas and please share your thoughts.

Doug, WA3DSP