

Raspberry Pi3 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 in one 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 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 wireless-setup.sh script. So lets say you have three access points you want to define. We will call them A-fixed, B-fixed, and C-portable and assume all of these points would show up in the scanned SSID list using those names. The names used here are just examples and you would have different SSID names in your own scheme. In this example the names would define two fixed wireless access points and a portable hotspot. Lets also assume for this example that all three are at the same physical location and accessible.

Before using wireless-setup.sh we would first clear the wpa_supplicant file by issuing the command -

```
clear_wpa_passwd_file.sh
```

Answer 'yes' to clear the file.

Then run **wireless-setup.sh** and pick the first SSID – 'A-fixed' in this example from the list. Enter the passphrase, save the setting and ensure that it works.

You now have an 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.

So now we have our first SSID – 'A-fixed' defined. Now we need to copy this to a separate file.

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

You would then repeat the above steps, clearing the file, setting up the wireless for the 'B-fixed' and 'C-portable' SSID's and copy each one to a separate file, B-fixed.wifi, C-portable.wifi.

When done you would have the files A-fixed.wifi, B-fixed.wifi, and C-portable.wifi in **/etc/asterisk/local**. It is best to save your own custom files in the **/etc/asterisk/local** directory.

Now you would create a script in **/etc/asterisk/local**. Lets 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"  
      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 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.

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

If you had the A1 “say local IP” function uncommented in the functions section of rpt.conf a DTMF *A1 on your radio would tell you what local IP address was assigned after a change. The last two lines in the above script to do this at each access point change but would do nothing if the A1 function was 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 which can take up to 10 minutes. Until both ends have each others 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