

Bridge FW Development January 2023

Copyright © RYSE

Author: MOHAMED Kallel

Revision history:

Date	Author	Description of change
2023/01/03	MOHAMED Kallel	Version 1: WiFi Development
2023/01/19	MOHAMED Kallel	Version 2: Add fix of BCT IPv6 and fix of Continuous Reboot and Appendixes

tags:

WiFi | BCT | IPv6 | FW Upgrade | Bootloader | Onion | MT768

Table of Contents

1. WiFi Feature implementation	3
1.1. Possible scenarios	3
1.2. Feedback from design team	5
1.3. Developed Solution	6
1.4. Enable WiFi driver for MT768	7
1.5. WiFi Configuration	7
1.6. WiFi Limitation	8
1.7. WiFi encryption	9
2. Fixing the issue of BCT IPv6 fail	10
3. Fixing the issue of continuous reboot	12
4. Appendix: Upgrade firmware from boot-loader via web	13
5. Appendix: BCT IPv6	15
6. Appendix: tricks	15

1. WiFi Feature implementation

1.1. Possible scenarios

Solution 1:

Bridge Firmware [WiFi use case]



Solution 1



The first solution consist of using only wifi access to the bridge to configure the WiFi access of the bridge to the gateway

On startup >

If no known WiFi is recorded or known WiFi not found >

Start broadcasting WiFi network "RYSE-SmartBridge_XXXX" (hostname) for 60 sec >

User connects to this network >

Redirected user to webpage >

webpage displays available WiFi networks >

User selects desired SSID and enters username and pass >

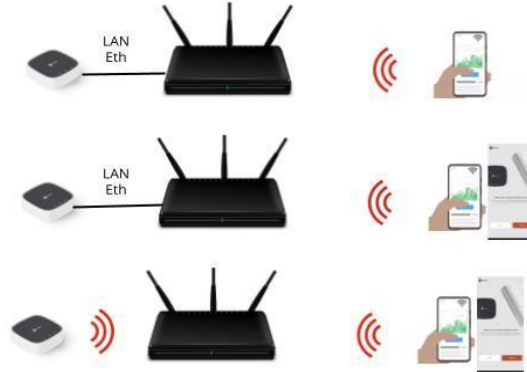
bridge confirms connection >

stops broadcasting SSID

Solution 2:

Bridge Firmware [WiFi use case]

Solution 2



The second solution consist of using (only in the configuration phase) ethernet access between the bridge and the gateway

In any time the user can plugin the bridge to the router with eth cable >

the bridge connect to the LAN and to the cloud>

the user use the ryse mobile app in his mobile to configure WiFi network>

User selects desired SSID and enters username and pass >





bridge confirms connection >

User can unplug the bridge and use the WiFi connection

Comparison:

Bridge Firmware [WiFi use case]



Solution		
	<ul style="list-style-type: none"> - No need Phy access to the Gateway via eth cable 	<ul style="list-style-type: none"> - Need captive portal to configure wifi access - Not simple to implement - Change WiFi Network of the user
	<ul style="list-style-type: none"> - Simple to implement - No Need to Change Wifi Network of the user - Use RYSE App to configure wifi access 	<ul style="list-style-type: none"> - Need Phy access to the gateway via eth cable

Please feel free to share your thoughts!

1.2. Feedback from design team



Saijal 4:57 PM

I prefer the first option as well. Users in general would be more familiar with the first option as it is more of a standard when installing other smart home devices like smart bulbs for example. The physical access bit can also be an issue as it may require the user to locate and move things around to access, since a lot of people store their router at a separate location, especially if they live in a house. It also introduces an additional step and device connection which may require adding further instructions in the app, tutorials etc.

Let me know if I missed something!

Also, Aras and Hannah are off today, so we'll have to wait for them to weigh in once they're back



Aras (R-iss) 8:24 PM

I have a moment to chime in here.

Option 1 sounds better and fits our retrofit solution better. It is similar to other smart device set-up as [@Saijal](#) mentioned above.

I know some people place their routers in obscure locations in general...

[@MOHAMED Kallel](#) is it possible to have the wifi switch back to the gateway wifi automatically after the user inputs their username and password?

1.3. Developed Solution



MOHAMED Kallel 9:53 AM

@Aras (R-iss) only if we play all the scenario from our RYSE mob app

```
>the User will use a specific menu from our RYSE app to configure WiFi  
access of the bridge to the gateway  
>the RYSE app will look for the WiFi Network RYSE_SmartBridge_XXXX for 60sec  
>then it will connect automatically to this network (if it's found)  
>then the app will return a list of available WiFi networks in the area  
>then the user select the WiFi network of the gateway and input password  
>then the bridge connect to the gateway via WiFi  
>then the RYSE app switch back to the gateway WiFi network
```



Jakub 5:43 PM

It's kind of like a WiFi speaker if I recall. Something like this I think up until about a minute in (strictly from an app interface)

<https://www.orbitsound.com/en-gb/support/articles/i-need-help-setting-up-my-wi-fi-speaker/>

Orbitsound

I need help setting up my Wi-Fi speaker

Use this page if you need help setting up Wi-Fi on your Orbitsound Wi-Fi enabled speaker (123 kB) ▾



The link:

<https://www.orbitsound.com/en-gb/support/articles/i-need-help-setting-up-my-wi-fi-speaker/>

ticket added about this implementation in bitbucket (admin repo) :

https://bitbucket.org/axislife/bridge_admin/issues/50/supporting-wifi

1.4. Enable WiFi driver for MT768

In make menuconfig, we need to select:

```
.config - OpenWrt Configuration
> Kernel modules > Wireless Drivers -

Wireless Drivers
Arrow keys navigate the menu. <Enter> selects submenus ---- (or empty submenus ----). Highlighted letters are hotkey
excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [
module capable

^(-)
-* kmod-mac80211..... Linux 802.11 Wireless Networking Stack
< > kmod-mac80211-hwsim..... mac80211 HW simulation device
< > kmod-mt76..... MediaTek MT76x2/MT7603 wireless driver (metapackage)
< > kmod-mt7601u..... MT7601U-based USB dongles Wireless Driver
-* kmod-mt7603..... MediaTek MT7603 wireless driver
< > kmod-mt76x0e..... MediaTek MT76x0E wireless driver
< > kmod-mt76x0u..... MediaTek MT76x0U wireless driver
-* kmod-mt76x2..... MediaTek MT76x2 wireless driver
< > kmod-mt76x2u..... MediaTek MT76x2U wireless driver
< > kmod-mwifiex-pcie
< > kmod-mwifiex-sdio
```

diff of .conf and config.seed:

1.5. WiFi Configuration

The adequate WiFi architecture for the chosen solution is to have:

- WiFi AP interface (Access Point) attached to the lan interface(network config). The lan will contain physically only wifi AP interface and not bridged. The WiFi AP interface will be enabled only in the first 60 seconds of the start time of the bridge then it will be kept disabled for the running time of the bridge.
- WiFi STA interface (Station/client) attached to the wwan interface(network config). The wwan will contains physically only wifi STA interface and not bridged. The WiFi STA interface will be disabled by default and enabled only when configured by the Mob user
- Keep the same config of the other interfaces in network file as usual

Wireless config

```
config wifi-device 'radio0'
    option type 'mac80211'
    option channel 'auto'
    option hwmode '11g'
    option path 'platform/10300000.wmac'
    option htmode 'HT20'
    option disabled '0'

config wifi-iface 'default_radio0'
    option device 'radio0'
    option network 'lan'
    option mode 'ap'
    option encryption 'psk2'
    option ssid 'RYSE-SmartBridge-F412'
    option key '123456789'
```

```
option disabled '0'

config wifi-iface 'ifacegw'
    option device 'radio0'
    option network 'wwan'
    option mode 'sta'
    option disabled '0'
    option ssid 'OpenWrt'
    option key '123456789'
    option encryption 'psk'
```

Network config

```
config interface 'loopback'
    option ifname 'lo'
    option proto 'static'
    option ipaddr '127.0.0.1'
    option netmask '255.0.0.0'

config globals 'globals'
    option ula_prefix 'fd8b:ac45:9889::/48'

config interface 'lan'
    option proto 'static'
    option netmask '255.255.255.0'
    option ip6assign '60'
    option ipaddr '10.17.19.1'

config interface 'wan'
    option proto 'dhcp'
    option ifname 'eth0'
    option ip6assign '60'

config interface 'wwan'
    option proto 'dhcp'
    option ip6assign '60'
```

1.6. WiFi Limitation

If the credentials of the WiFi STA are wrong or the STA interface is unable to connect to the target SSID, then the STA interface will keep the channel busy in order to proceed to the retry of connection and thus the AP will be kept down and we can't see any more the SSID RYSE-SmartBridge-XXXX.

If the Mob user enters the wrong credentials of the Router WiFi network then the user will lost the WiFi connection to the bridge.

This limitation is normal since we have only 1 radio interface in the bridge.

To fix this limitation, we will add in the bridge side a timeout of 30 sec when the user enter the STA credentials. If there is no connection in the WiFi STA interface after the timeout, then disable the STA and then the AP could be back to UP and then can see back the SSID

RYSE-SmartBridge-XXXX. From Mob App side, we need to add also a timeout when the user request for connection of the bridge to the router and then reconnect

1.7. WiFi encryption

The possible WiFi encryption configuration are stipulated in the following table:

Value	AP	Ciphers	STA
none	no authentication	none	none
sae	WPA3 Personal (SAE)	CCMP	
sae-mixed	WPA2/WPA3 Personal (PSK/SAE) mixed mode	CCMP	psk2
psk2+tkip+ccmp	WPA2 Personal (PSK)	TKIP, CCMP	psk2
psk2+tkip+aes	WPA2 Personal (PSK)	TKIP, CCMP	psk2
psk2+tkip	WPA2 Personal (PSK)	TKIP	not working psk2 & psk2+tkip
psk2+ccmp	WPA2 Personal (PSK)	CCMP	psk2
psk2+aes	WPA2 Personal (PSK)	CCMP	psk2
psk2	WPA2 Personal (PSK)	CCMP	psk2
psk+tkip+ccmp	WPA Personal (PSK)	TKIP, CCMP	psk
psk+tkip+aes	WPA Personal (PSK)	TKIP, CCMP	psk
psk+tkip	WPA Personal (PSK)	TKIP	not working psk & psk+tkip
psk+ccmp	WPA Personal (PSK)	CCMP	psk
psk+aes	WPA Personal (PSK)	CCMP	psk
psk	WPA Personal (PSK)	CCMP	psk
psk-mixed+tkip+ccmp	WPA/WPA2 Personal (PSK) mixed mode	TKIP, CCMP	psk2
psk-mixed+tkip+aes	WPA/WPA2 Personal (PSK) mixed mode	TKIP, CCMP	psk2
psk-mixed+tkip	WPA/WPA2 Personal (PSK) mixed mode	TKIP	not working psk2 & psk2+tkip neither with psk
psk-mixed+ccmp	WPA/WPA2 Personal (PSK) mixed mode	CCMP	psk2
psk-mixed+aes	WPA/WPA2 Personal (PSK) mixed mode	CCMP	psk2
psk-mixed	WPA/WPA2 Personal (PSK) mixed mode	CCMP	psk2
wep	defaults to "open system" authentication aka wep+open	RC4	
wep+open	"open system" authentication	RC4	
wep+shared	"shared key" authentication	RC4	
wpa3	WPA3 Enterprise	CCMP	
wpa3-mixed	WPA3/WPA2 Enterprise	CCMP	
wpa2+tkip+ccmp	WPA2 Enterprise	TKIP, CCMP	
wpa2+tkip+aes	WPA2 Enterprise	TKIP, CCMP	
wpa2+ccmp	WPA2 Enterprise	CCMP	
wpa2+aes'	WPA2 Enterprise	CCMP	
wpa2	WPA2 Enterprise	CCMP	
wpa2+tkip	WPA2 Enterprise	TKIP	
wpa+tkip+ccmp	WPA Enterprise	TKIP, CCMP	
wpa+tkip+aes	WPA Enterprise	TKIP, AES	
wpa+ccmp	WPA Enterprise	CCMP	
wpa+aes	WPA Enterprise	CCMP	
wpa+tkip	WPA Enterprise	TKIP	
wpa	WPA Enterprise	CCMP	
wpa-mixed+tkip+ccmp	WPA/WPA2 Enterprise mixed mode	TKIP, CCMP	
wpa-mixed+tkip+aes	WPA/WPA2 Enterprise mixed mode	TKIP, CCMP	
wpa-mixed+tkip	WPA/WPA2 Enterprise mixed mode	TKIP	
wpa-mixed+ccmp	WPA/WPA2 Enterprise mixed mode	CCMP	
wpa-mixed+aes	WPA/WPA2 Enterprise mixed mode	CCMP	
wpa-mixed	WPA/WPA2 Enterprise mixed mode	CCMP	
owe	Opportunistic Wireless Encryption (OWE)	CCMP	

Take in consideration hidden SSID and WPS

2. Fixing the issue of BCT IPv6 fail

the BCT IPv6 fails from the beginning of the test. The initial IPv6 MDNS probe was not sent by the bridge upon the start of the device.

by debugging the avahi-daemon and avahi-autoipd we can see the following error messages in the start of avahi-daemon:

```
[root@RYSE-SmartBridge-F412:~# logread | grep avahi
Thu Jan 19 17:36:37 2023 daemon.info avahi-daemon[1198]: Found user 'nobody' (UID 65534) and group 'nogroup' (GID 65534)
Thu Jan 19 17:36:37 2023 daemon.info avahi-daemon[1198]: Successfully dropped root privileges.
Thu Jan 19 17:36:37 2023 daemon.info avahi-daemon[1198]: avahi-daemon 0.6.32 starting up.
Thu Jan 19 17:36:37 2023 daemon.warn avahi-daemon[1198]: WARNING: No NSS support for mDNS detected, consider installing nss
Thu Jan 19 17:36:37 2023 daemon.info avahi-daemon[1198]: No service file found in /etc/avahi/services.
Thu Jan 19 17:36:37 2023 daemon.info avahi-daemon[1198]: Joining mDNS multicast group on interface eth0.IPv6 with address fe80::42a3:6bff:
Thu Jan 19 17:36:37 2023 daemon.info avahi-daemon[1198]: New relevant interface eth0.IPv6 for mDNS.
Thu Jan 19 17:36:37 2023 daemon.info avahi-daemon[1198]: Network interface enumeration completed.
Thu Jan 19 17:36:37 2023 daemon.info avahi-daemon[1198]: Registering new address record for fe80::42a3:6bff:fec7:f412 on interface eth0.
Thu Jan 19 17:36:37 2023 daemon.info avahi-daemon[1198]: Registering HINFO record with values 'MIPS'/'LINUX'.
Thu Jan 19 17:36:37 2023 daemon.debug avahi-daemon[1198]: sendmsg() to 0:0:ff02:: failed: Invalid argument
Thu Jan 19 17:36:38 2023 daemon.debug avahi-daemon[1198]: sendmsg() to 0:0:ff02:: failed: Invalid argument
Thu Jan 19 17:36:38 2023 daemon.info avahi-autoipd(eth0)[1235]: Found user 'nobody' (UID 65534) and group 'nogroup' (GID 65534)
Thu Jan 19 17:36:38 2023 daemon.info avahi-autoipd(eth0)[1235]: Successfully called chroot().
Thu Jan 19 17:36:38 2023 daemon.info avahi-autoipd(eth0)[1235]: Successfully dropped root privileges.
Thu Jan 19 17:36:38 2023 daemon.debug avahi-autoipd(eth0)[1235]: State transition START-0 -> START-0
Thu Jan 19 17:36:38 2023 daemon.info avahi-autoipd(eth0)[1235]: Starting with address 169.254.8.249
Thu Jan 19 17:36:38 2023 daemon.debug avahi-autoipd(eth0)[1235]: State transition START-0 -> WAITING_PROBE-0
Thu Jan 19 17:36:38 2023 daemon.debug avahi-autoipd(eth0)[1235]: sleeping 246ms
Thu Jan 19 17:36:38 2023 daemon.debug avahi-daemon[1198]: sendmsg() to 0:0:ff02:: failed: Invalid argument
```

By investigating the source code of avahi and wireshark, I noticed that avahi was unable to send some broadcast IPv6 message in the start of avahi-daemon and the avahi did not retry to send these messages. But later the avahi-daemon was able to send the next IPv6 multicast packets:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fe80::42a3:6bff:fec7:f412	ff02::2	ICMPv6	70	Router Solicitation from 40:a3:6b:c7:f4:12
2	14.179803	::	ff02::16	ICMPv6	130	Multicast Listener Report Message v2
3	14.309620	::	ff02::16	ICMPv6	130	Multicast Listener Report Message v2
4	14.619633	::	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
5	15.167284	::	ff02::1:fffc:f412	ICMPv6	86	Neighbor Solicitation for fe80::42a3:6bff:fec7:f412
6	15.439575	::	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
7	15.637698	Onion_07:f4:12	Broadcast	ARP	60	Who has 169.254.8.249? (ARP Probe)
8	16.447633	fe80::42a3:6bff:fec7:f412	ff02::16	ICMPv6	170	Multicast Listener Report Message v2
9	16.469515	fe80::42a3:6bff:fec7:f412	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
10	16.799558	fe80::42a3:6bff:fec7:f412	ff02::16	ICMPv6	170	Multicast Listener Report Message v2
11	16.920417	fe80::42a3:6bff:fec7:f412	ff02::fb	MDNS	238	Standard query response 0x0000 PTR, cache flush RYSE-SmartBridge-F412.local HINFO,
12	17.069544	fe80::42a3:6bff:fec7:f412	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
13	17.176459	Onion_07:f4:12	Broadcast	ARP	60	Who has 169.254.8.249? (ARP Probe)
14	18.792257	Onion_07:f4:12	Broadcast	ARP	60	Who has 169.254.8.249? (ARP Probe)
15	19.170507	fe80::42a3:6bff:fec7:f412	ff02::fb	MDNS	238	Standard query response 0x0000 PTR, cache flush RYSE-SmartBridge-F412.local HINFO,
16	20.793632	Onion_07:f4:12	Broadcast	ARP	60	ARP Announcement for 169.254.8.249

The first idea that I tried to fix the issue is to add a sleep before starting the avahi daemon. Maybe something is not yet initiated in the startup and it causes the failure of sending the IPv6 packets. this idea fix the issue and I can see in the logread of avahi and in the wireshark capture that the MDNS IPv6 messages are sent:

Time	Source	Destination	Protocol	Length	Info
1 0.000000	fe80::42a3:6bff:fec7:f412	ff02::12	ICMPv6	70	Router Solicitation from 40:a3:6b:c7:f4:12
2 14.079690	::	ff02::16	ICMPv6	130	Multicast Listener Report Message v2
3 14.619497	::	ff02::1:ffc7:f412	ICMPv6	86	Neighbor Solicitation for fe80::42a3:6bff:fec7:f412
4 14.959570	::	ff02::16	ICMPv6	130	Multicast Listener Report Message v2
5 15.679671	fe80::42a3:6bff:fec7:f412	ff02::16	ICMPv6	150	Multicast Listener Report Message v2
6 15.709460	fe80::42a3:6bff:fec7:f412	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
7 15.799427	fe80::42a3:6bff:fec7:f412	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
8 16.029553	fe80::42a3:6bff:fec7:f412	ff02::16	ICMPv6	150	Multicast Listener Report Message v2
9 17.339470	fe80::42a3:6bff:fec7:f412	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
10 17.450203	fe80::42a3:6bff:fec7:f412	ff02::fb	MDNS	250	Standard query 0x0000 ANY 2.1.4.f.7.c.e.f.f.b.6.3.a.2.4.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa,
11 17.629484	fe80::42a3:6bff:fec7:f412	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
12 17.700992	fe80::42a3:6bff:fec7:f412	ff02::fb	MDNS	250	Standard query 0x0000 ANY 2.1.4.f.7.c.e.f.f.b.6.3.a.2.4.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa,
13 18.007396	fe80::42a3:6bff:fec7:f412	ff02::fb	MDNS	250	Standard query 0x0000 ANY 2.1.4.f.7.c.e.f.f.b.6.3.a.2.4.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa,
14 18.260560	fe80::42a3:6bff:fec7:f412	ff02::fb	MDNS	238	Standard query response 0x0000 PTR, cache flush RYSE-SmartBridge-F412.local HINFO, cache flush MIPS
15 18.895364	Onion_07:f4:12	Broadcast	ARP	60	Who has 169.254.8.249? (ARP Probe)
16 19.310243	fe80::42a3:6bff:fec7:f412	ff02::fb	MDNS	215	Standard query response 0x0000 PTR, cache flush RYSE-SmartBridge-F412.local AAAA, cache flush fe80::
17 19.418618	fe80::42a3:6bff:fec7:f412	ff02::fb	MDNS	124	Standard query response 0x0000 HINFO, cache flush MIPS LINUX
18 20.168829	Onion_07:f4:12	Broadcast	ARP	60	Who has 169.254.8.249? (ARP Probe)



The issue is fixed by adding a sleep of 10 seconds before launching the avahi-daemon, but this is a work around, and we need to know the real cause and fix it with the right solution.

Maybe the problem is that the avahi-daemon tried to send IPv6 messages before an IPv6 is assigned to the eth0 interface by the process of neighboring solicitation by Linux. I added a function in the avahi-daemon init script that the check that there is IPv6 assigned to the eth0 interface before launching avahi, but the problem persist I can see that the avahi is started after the IPv6 was assigned to the eth0 but the bug still persist.

The second suspicion may be the issue is related to the IPv6 routing table. Maybe the routing table is not filled yet before the send of the first IPv6 multicast packets.

I added a command to show the IPv6 route table in the start of avahi-daemon and I found that the Linux did not finish yet the filling of IPv6 routing table (even after the assignment of IPv6 address to the eth0):

```
root@RYSE-SmartBridge-F412:~# route -A inet6
Kernel IPv6 routing table
Destination                                Next Hop                                Flags Metric Ref    Use Iface
::/0                                       ::                                     !n     -1    1      59 lo
::/0                                       ::                                     !n     -1    1      59 lo
```

I added a check of IPv6 routing table in the /etc/init.d/avahi daemon before the start of the avahi-daemon and the issue is gone:

```
check_ipv6_assigned() {
    local r6=""
    local i=0
    local useip6=$(cat /etc/avahi/avahi-daemon.conf | grep "use-ipv6.*yes")
    [ "$useip6" = "" ] && return
    while [ "$r6" = "" -a $((i++)) -lt 50 ]; do
        r6=$(route -A inet6 | grep "[fF][eE]80:[A-Fa-f0-9:][A-Fa-f0-9].*/128")
        sleep 1
    done
}
```

this fix was pushed to bridge_firmware git repository in Jan 13 2023

fix the issue of BCT ipv6 failing

13 Jan 2023 16:50

MOHAMED Kallel

fd4d3b5b

Commit: fd4d3b5b0fbee4fb59c6c74d5ad3bd8004726143

Parents: [482820a5a0297d88c5b0630eebcb012e271b298c](#)

Author: MOHAMED Kallel <mohamed@helloryse.com>

Committer: MOHAMED Kallel <mohamed@helloryse.com>

Date: Fri Jan 13 2023 16:50:38 GMT+0100 (Central European Standard Time)

fix the issue of BCT ipv6 failing

files / etc

init.d

avahi-daemon

uci-defaults

15_network (+1 | -1)

3. Fixing the issue of continuous reboot

The fix of BCT IPv6 issue caused a delay in the avahi-daemon init script and this delay caused a continuous reboot of the bridge.

The issue happens only after resetting the bridge

We got the following error in the console log:

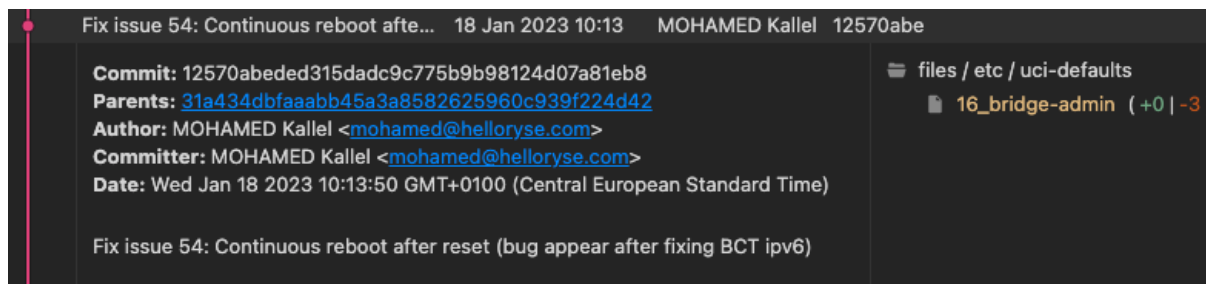
```
[ 38.237031] Removing MTD device #6 (rootfs_data) with use count 1
[ 38.270341] reboot: Restarting system
```

After investigation the issue and checking many scenarios by changing the init scripts of avahi and admin, I found that the admin app is starting in early phase and I found that the admin is launched by the uci-default script `16_bridge-admin`:

```
1 #!/bin/sh
2
3- /etc/init.d/bridge-admin enable
4- /etc/init.d/bridge-admin start
5-
```

I removed these lines and the issue is fixed.

this fix was pushed to bridge_firmware git repository in Jan 13 2023



The screenshot shows a git commit interface. The commit message is "Fix issue 54: Continuous reboot after reset (bug appear after fixing BCT ipv6)". The commit hash is 12570abed315dadc9c775b9b98124d07a81eb8. The commit was made by MOHAMED Kallel on Wed Jan 18 2023. The diff shows changes to the file `etc/uci-defaults/16_bridge-admin`, with a change of +0 lines and -3 lines.

This fix could fix systematically the issues #9 and #16

The fix of issues #9 and #16 was made by adding a reboot after the reset or fw upgrade. And it's not a real fix but a work around.

We need to re-investigate these issue and see if this fix is fixing these issues (without adding reboot)

#9: https://bitbucket.org/axislife/bridge_admin/issues/9/

(Second device firmware update over the air fails)

#16: https://bitbucket.org/axislife/bridge_admin/issues/16/

(OTA after factory reset fails Create issue)



4. Appendix: Upgrade firmware from boot-loader via web

In case it's unable to access the bridge fw with mqtt or ssh or local api.
Or in case the firmware keep crashing / rebooting in the start of the firmware
Or in case there is panic in the start of the fw of the bridge
Then we need to upgrade the bridge from the boot-loader.

- 1- Connect to the bridge with serial
- 2- power off the bridge
- 3- keep pushing the reset button
- 4- power on the bridge
- 5- got the following menu on the console:

```
ASIC 7628_MP (Port5<->None)
DRAM component: 512 Mbits DDR, width 16
DRAM bus: 16 bit
Total memory: 64 MBytes
Flash component: SPI Flash
Date:May 12 2020   Time:21:55:10
=====
icache: sets:512, ways:4, linesz:32 ,total:65536
dcache: sets:256, ways:4, linesz:32 ,total:32768
CPU freq = 575 MHZ
Estimated memory size = 64 Mbytes
Resetting MT7628 PHY.
Initializing MT7688 GPIO system.
wifi mac address = 40A36BC7F411.

*****
* Hold Reset button for more options *
*****

You have 40 seconds left to select a menu option...

Please select option:
[ Enter ]: Boot Omega2.
[ 0 ]: Start Web recovery mode.
[ 1 ]: Start command line mode.
[ 2 ]: Flash firmware from USB storage.

Option [0] selected.
```

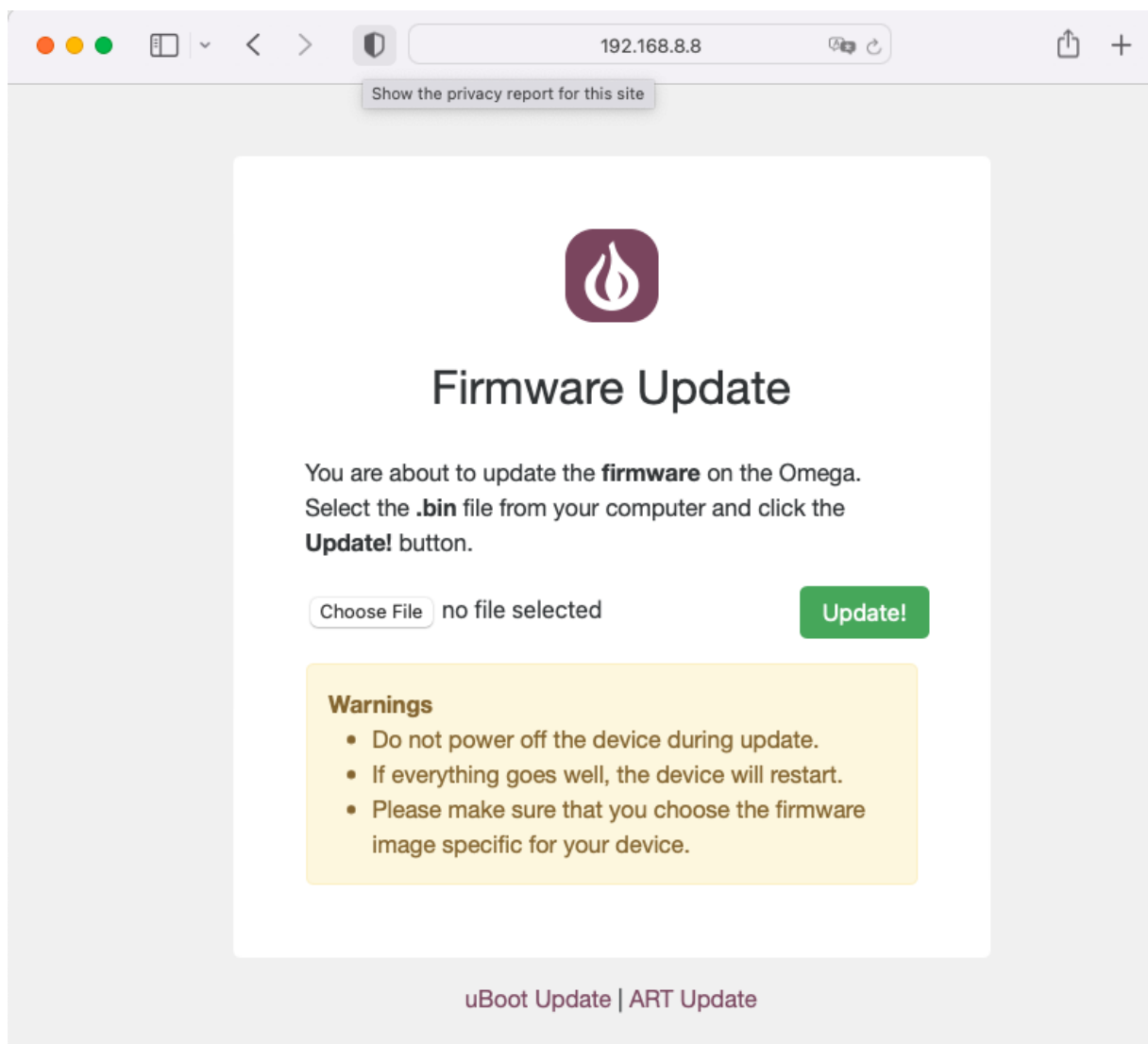
6- select 0

7- configure your pc network interface to have address in the subnet 192.168.8.0/24
for mac os I can add a secondary ip address to my interface with:

```
sudo ifconfig en4 alias 192.168.8.19 255.255.255.0
```

8- From the PC web browser connect to the IP address of the bridge (bootloader address)

9- and then upgrade the new firmware





5. Appendix: BCT IPv6

The platform of BCT is the same as described in the document []

The same for the platform configuration (MacBook and Router), the configuration is exactly the same as the IPv4 platform configuration.

To launch the BCT IPv6 tests:

```
sudo ./BonjourConformanceTest -I en4 -6 -M -D -F BCT_IPv6_brief_log_HK6.3_fwvers2.0.2-SWAUTH-20230119-3.txt  
-Aip fe80::02e0:4cff:fee9:3052 -Amac 00:e0:4c:e9:30:52
```

The fe80::02e0:4cff:fee9:3052 and 00:e0:4c:e9:30:52 are the IPv6 address and the mac address of the router. And they can be retrieved easily from the web interface of the router or by using wireshark

6. Appendix: tricks

Interface index

```
Linux# ip link show  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN  
qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP100> mtu 1500 qdisc fq_codel state  
UNKNOWN qlen 1000  
    link/ether 40:a3:6b:c7:f4:12 brd ff:ff:ff:ff:ff:ff
```

the interface index is the first element in the line

error in onion openwrt build

When building the firmware, if we got error related to cert certificate of Ubuntu:

```
fatal: unable to access 'http://git.openwrt.org/project/iwinfo.git/': server  
certificate verification failed. CAfile: /etc/ssl/certs/ca-certificates.crt  
CRLfile: none
```

then we need to update the certificate:

```
apt-get install --reinstall ca-certificates
```