Creating BT PAN/RNDIS router using OpenWrt

Koichi Okamoto, Ishikawa Masayuki / let’s dive into their mechanism!
Biography (Koichi Okamoto)

• Architect of network team (IP layer or higher) of our Video & Sound platform based on Linux at Sony Home Entertainment and Sound Products Inc.

See the below URL for more detailed with Sony products related to me

https://ossna19.sched.com/speaker/koichiokamot
Table of Contents (Two parts)

- **Off-the-self hardware using OpenWrt extends BT PAN in addition to USB RNDIS**
  - What is Bluetooth (BT) PAN
  - Learning Linux BT mechanisms
  - Overview
  - Hardware/Software Requirement
  - Create OpenWrt Firmware
  - Write OpenWrt firmware
  - `bluetoothctl` utility flexibility
  - How to control BT PAN on BlueZ
  - Persistence of Pairing Information
  - Make NAP work at cold start

- **Introduction to low-end devices running NuttX with OpenWrt router**
  - Introduction to NuttX networking features
  - How to run Bluetooth on NuttX
  - Software stack for Bluetooth networking
  - BTstack log example
  - MP3 streaming via Bluetooth
  - Running RNDIS on NuttX
  - Working with Wi-Fi on Spresense
  - Use case for Webserver via Wi-Fi
Off-the-self hardware using OpenWrt extends BT PAN in addition to USB RNDIS
What is Bluetooth (BT) PAN

- **PAN stands for Personal Area Networking**
  - The following roles are defined within the PAN profile.
    - Network Access Point (NAP) and NAP service
    - Group Ad-hoc Network (GN) and GN service
    - PAN User (PANU) and PANU service

![Diagram of PAN network structure](https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=6554)
Learning Linux BT mechanisms (1/2)

Dell OptiPlex 7040
Core i7-6700 3.4GHz
RAM 8GByte
HDD 1TByte

Ubuntu 16.04.5 LTS (xenial)

BT USB Dongle (BSBT4D09BK)
NAP role

https://kernel.googlesource.com/pub/scm/bluetooth/bluez.git
bluez/test (python script)
- test-nap
Network server hierarchy
========================

Service          org.bluez
Interface        org.bluez.NetworkServer1
Object path      /org/bluez/{hci0,hci1,...}

Methods         void Register(string uuid, string bridge)

Register server for the provided UUID. Every new connection to this server will be added the bridge.

This API is the essence of NAP role for BT PAN connection.

'dbus-send' command can invoke this API.
Overview

SMP and Networking Support on NuttX/LC823450

OpenIoT2018 NA Portland

NuttX is POSIX-compliant oriented RTOS such as Tiny-Linux.

We have an update of RNDIS in second part

192.168.1.XX/24
(LAN network segment)

192.168.68.XX/24
(WAN network segment)

USB2.0

BT PAN
Profile connection

BT PAN

USB3.0 HUB
(U3H-T410BK)

BT Dongle
(BSBT4D09BK)

NAP role

USB RNDIS

Xperia XZ (Android7.0)

PANU role

BUFFALO WiFi Router
WZR-HP-G300NH

ssh/telnet console

Ethernet

Internet
(WAN side)

The Linux Foundation
Hardware Requirement

• OpenWrt supporting hardware with USB port
  – BUFFALO WZR-HP-G300NH (USB 2.0 port)
    You can see https://openwrt.org/toh/start

• BT USB Dongle supporting PAN profile
  – BUFFALO BT USB Dongle BSBT4D09BK
Software Requirement

- OpenWrt Firmware for WZR-HP-G300NH with USB RNDIS configuration

I used the below commit at that time.

https://git.opensrt.org/openwrt/openwrt.git/commit?name=HEAD
Author: Christoph Krapp <achterin@googlemail.com>
Date: Tue Jul 3 11:06:20 2018 +0200

Commit 8722c52b41d551e768b3cc46049af6657099d59
Author: Christoph Krapp <achterin@googlemail.com>
Date: Tue Jul 3 11:06:20 2018 +0200

ath79: remove bs-partition ro-flag for UniFi AC devices

This removes the read-only flag from the bs (bootselect) partition
on UniFi AC devices. This allows to correct the indicator from which
partition the device is booting its kernel from.

See also:
- freifunk-gluon/gluon#1301

Signed-off-by: Christoph Krapp <achterin@googlemail.com>
$ git clone https://git.openwrt.org/openwrt/openwrt.git/
$ cd openwrt
$ ./scripts/feeds update -a
$ ./scripts/feeds install -a
$ cp enable_btpan_usbrndis_for_wzr-hz-g300nh.seed .config
$ make defconfig
$ make

These two command should execute once so that a needed software package can be selected.

I provide config.seed file at the bottom of this slide

Generated firmware exists on ./bin/targets/ar71xx/generic/ directory
Write OpenWrt firmware

• Firmware update is done from Web GUI
  – BUFFALO’s original firmware Web GUI can accept “openwrt-ar71xx-generic-wzr-hp-g300nh-squashfs-factory.bin” image

  – After OpenWrt firmware is run, firmware update is done from the OpenWrt Web GUI (Luci) using “openwrt-ar71xx-generic-wzr-hp-g300nh-squashfs-sysupgrade.bin” image
bluetoothctl utility flexibility

Please note that “bluetoothctl” is the file name of Bluetooth utility program.

Interactive mode

```
root@OpenWrt:~# bluetoothctl
Agent registered
[bluetooth]# power on
[CHG] Controller 00:1B:DC:06:61:D4 Class: 0x00020000
Changing power on succeeded
[CHG] Controller 00:1B:DC:06:61:D4 Powered: yes
[bluetooth]# quit
root@OpenWrt:~#
```

“root@OpenWrt:~#” is shell prompt. “[Bluetooth]#” is the prompt of bluetoothctl.

Single command line

```
root@OpenWrt:~# bluetoothctl power on
[CHG] Controller 00:1B:DC:06:61:D4 Class: 0x00020000
Changing power on succeeded
[CHG] Controller 00:1B:DC:06:61:D4 Powered: yes
root@OpenWrt:~#
```

I impressed bluetoothctl utility consideration for both use cases. I applied my sample control application in my company after I knew that.
How to connect BT PAN by hand (1/6)

- ssh login to OpenWrt and set NAP UUID to `bluetoothd`

```
root@OpenWrt:~# dbus-send --system --dest=org.bluez /org/bluez/hci0 --type=method_call org.bluez.NetworkServer1.Register string:"00001116-0000-1000-8000-00805f9b34fb" string:"br-lan"
root@OpenWrt:~#
```
Now `bluetoothd` supports NAP role (2/6)

```
[bluetooth]# show
Controller 00:1B:DC:06:61:D4 (public)
   Name: BlueZ 5.49
   Alias: BlueZ 5.49
   Class: 0x00020000
   Powered: yes
   Discoverable: no
   Pairable: yes
   UUID: Generic Attribute Profile (00001801-0000-1000-8000-00805f9b34fb)
   UUID: NAP                       (00001116-0000-1000-8000-00805f9b34fb)
   UUID: A/V Remote Control        (0000110e-0000-1000-8000-00805f9b34fb)
   UUID: PnP Information           (00001200-0000-1000-8000-00805f9b34fb)
   UUID: A/V Remote Control Target (0000110c-0000-1000-8000-00805f9b34fb)
   UUID: Generic Access Profile    (00001800-0000-1000-8000-00805f9b34fb)
   Modalias: usb:v1D6Bp0246d0531
   Discovering: no

[bluetooth]#
```

Please note that “bluetoothd” is the file name of Bluetooth Daemon program.

Service Discovery

Set trusted device in advance (3/6)

root@OpenWrt:~# bluetoothctl
[bluetooth]# trust 00:02:5B:00:A5:A5
[CHG] Device 00:02:5B:00:A5:A5 Trusted: yes
Changing 00:02:5B:00:A5:A5 trust succeeded
[bluetooth]#

Specify the MAC address of BT PANU role device to be connected (e.g: low-end device such as second part)
Connect your device to BT on OpenWrt (4/6)

- Make BT discoverable on OpenWrt router
- Pairing your device with OpenWrt router
- Establish BT PAN connection
When PANU role device connects to router, bnep0 interface appears on BT PAN router (NAP role). This node is added “br-lan” bridge interface to which ethernet LAN port connects.

It means BT PANU device and ethernet LAN connected device have same LAN side IP network.
“bnep0” interface belongs to “br-lan” bridge (6/6)

LAN is “br-lan” bridge interface

WAN is “eth1” interface

4 ethernet port interface is “eth0.1”

NAP role
BT PAN connection

PANU role

WAN is “eth1” interface
Persistence of Pairing Information

- Nothing to do for BT pairing
- Once BT pairing is completed by hand, `bluetoothd` (BlueZ daemon) will save its connection data.
Make NAP work at cold start

- Two things need to be executed at startup time on OpenWrt router to enable BT NAP role
  - Set NAP role to `bluetoothd`
  - BT Dongle Power ON
- “Local Startup” is used for the above operation as shell scripts.
  - “Local Startup” locates Web GUI’s “System” tab -> “Startup” screen on the bottom.
Introduction to low-end devices running NuttX with OpenWrt router
Senior Software Engineer
at Sony Home Entertainment and Sound Products Inc.

Technical background
- 3D graphics, home networking, Internet-to-Home, Embedded Systems

Product development
- Portable Media Player (Linux/Android)
- Digital voice recorder, music player (NuttX)

Public talks
- Arm Techcon 2016, ELC2017NA, OpenIoT2018NA, NuttX2019
Introduction to NuttX networking features

- **Motivation**
  - Confirm interoperability between OpenWrt and NuttX by running network applications on resource limited devices

- **Features**
  - Ethernet and IEEE 802.11 Full MAC
  - 6LoWPAN for IEEE 802.15.4 MAC
  - USB RNDIS, CDC-ECM
  - SLIP, TUN/PPP, loopback devices
  - IPv4, IPv6, TCP, UDP, ARP, ICMP, ICMPv6, IGMPv2
  - BSD compatible socket layer
  - DNS name resolution / NetDB
  - User socket

- NuttX 7.31 + H/W (e.g. Cortex-M based)
How to run Bluetooth on NuttX

- Port the BTstack by Bluekitchen to NuttX
  - Based on posix-h4 with H/W flow control
  - UART speed: 921600 baud
  - Free for non-commercial use

- Add TAP mode to the NuttX tun driver
  - TAP mode is used for network bridge
  - NOTE: TUN mode is used for network routing

- HCI_RESET issue in SMP mode
  - CSR’s mode change with HCI_RESET is tricky
  - Still unstable in SMP mode
Software stack for Bluetooth networking

*BNEP: Bluetooth Network Encapsulation Protocol
BTstack log example

H4 device: /dev/ttvS1
[2019-06-27 12:12:42.280] LOG -- hci.c.3797: BTSTACK EVENT_STATE 1
[2019-06-27 12:12:42.810] LOG -- hci.c.1878: Manufacturer: 0x000a

Local version information:
- HCI Version 0x0006
- HCI Revision 0x2031

[2019-06-27 12:12:57.070] LOG -- bnep.c.79: BNEP EVENT_CHANNEL_OPENED status 0x00 bd addr: BNEP connection open succeeded to 00:1B:DC:06:86:59 source UUID 0x1115 dest UUID: 0x1116,

Network Interface bnep0 activated
MP3 streaming via Bluetooth

Realtime Traffic

<table>
<thead>
<tr>
<th>Interface</th>
<th>Status</th>
<th>Inbound</th>
<th>Average</th>
<th>Outbound</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>bne0</td>
<td></td>
<td>4.59 kbit/s (0.57 kB/s)</td>
<td>4.17 kbit/s (0.52 kB/s)</td>
<td>181.04 kbit/s (22.63 kB/s)</td>
<td>157 kbit/s (19.63 kB/s)</td>
</tr>
</tbody>
</table>

```
mp3 streaming via Bluetooth

Music #1

Music #2
```
Running RNDIS on NuttX

LC823450XGEVK + LEDE (RNDIS) at OpenIoT 2018

Spresense + OpenWrt (RNDIS) at ELC2019NA
Working with Wi-Fi on Spresense

- Wi-Fi module: Telit GS2200M
  - Radio protocols: 802.11b/g/n (2.4GHz)
  - Interface: SPI 10MHz with DMA

- Implement GS2200M driver from scratch*
  - Based on the NuttX usrsock
  - Both STA and AP modes are supported
  - Fix cxd56_gpioint.c for interrupt handling
  - TCP and UDP are supported

- Modify the uIP webserver app for NuttX
  - Add a directory listing feature

* The code is available at https://bitbucket.org/nuttx/nuttx
Use case for Webserver via Wi-Fi

Network applications

- DHCP client
- DNS client
- telnetd
- webserver
- gs220m daemon

/dev/mmcasd0 /dev/urssock /dev/gs2200m

SDHCI Interrupt SPI

micro SD GS2200M

The latest NuttX upstream + Spresense

Wi-Fi

OpenWrt

WZR-HP-G300NH

VirtualBox + Ubuntu

Web browsing with Firefox

Index of /sd0

- [ ]
- [ ]
- [ ]
- [ ]
- [ ]
- [ ]
- [ ]

audio/
ps.txt
.Seventh/Spotlight.V100/
-spresence_header_code.jpg
.Trashes/
hello.txt

allP web server (NuttX 7.30 d228d33 Jul 5 2019 17:49:54)
Conclusion

• This work shows how to easily extend an OpenWrt router
• Also, NuttX networking is feasible on resource limited devices
Reference (1/3)

- I knew about LEDE at ELC2017 session in Portland

  OpenWrt/LEDE: when two become one, presented by Florian Fainelli, ([https://sched.co/9luP](https://sched.co/9luP))

- [https://elinux.org/images/0/0a/ELC_OpenWrt_LEDE.pdf](https://elinux.org/images/0/0a/ELC_OpenWrt_LEDE.pdf)

- [http://events17.linuxfoundation.org/events/embedded-linux-conference/program/slides](http://events17.linuxfoundation.org/events/embedded-linux-conference/program/slides)
Developing Audio Products with Cortex-M3/NuttX/C++11
(https://sched.co/9O0s)

(ELC2017 North America)

SMP and Networking Support on NuttX/LC823450
(https://sched.co/DYML)

(OpenIoT2018 North America)
OpenWrt documents

- Table of Hardware (supported hardware list)
  https://openwrt.org/toh/start

- Official Documents starting point
  https://openwrt.org/docs/start

- Developer Guide
  https://openwrt.org/docs/guide-developer/start

- Creating packages
  https://openwrt.org/docs/guide-developer/packages
  OpenWrt manages software components on a package basis. For this reason, it is better to create a package to import (Porting software component to OpenWrt). This URL explains about it. The template of Makefile realizes easy porting explanation by the example of bridge package.

- Quick Image Building Guide
  https://openwrt.org/docs/guide-developer/quickstart-build-images

BUFFALO product information

- Wi-Fi router
  http://buffalo.jp/products/catalog/network/wzr-hp-g300nh/

- BT Dongle
  http://buffalo.jp/product/peripheral/wireless-adapter/bsbl4d09bk/

IBM WATSON IOT iot-raspberrypi
https://github.com/ibm-watson-iot/device-raspberrypi

dbus
https://www.freedesktop.org/wiki/Software/dbus/

dbus-monitor
https://dbus.freedesktop.org/doc/dbus-monitor.1.html
dump D-Bus message
Any Questions?
Thank you for your participation and interest
Supplemental Material

- What’s OpenWrt
- Next two slides come from our OpenIoT2018 North America “SMP and Networking support on NuttX / LC823450”.

THE LINUX FOUNDATION
Introduction to LEDE

- **Motivation**
  - Build a shareable network testing environment for NuttX

- **Software**
  - LEDE project as of ELC2017 session
  - The project was forked from OpenWRT that is famous OSS for the router world as a turn key solution but they became one again (at the beginning of 2018)

- **Hardware**
  - WZR-HP-G300NH (buffalo) Wi-Fi router with USB 2.0 port

![WZR-HP-G300NH](image_url)
Support RNDIS on LEDE

- How to setup
  - Modify configuration
  - Add network USB0 (RNDIS) via LuCi
  - Change the network setting of USB0