Intelligent IoT Gateway on OpenWrt

Andrzej Wieczorek    Bartosz Markowski
Introduction

IoT

Gateways

Standards

IoT home alone concept
IoT is big! Infinite!

„Bigger than the biggest thing ever and then some. Much bigger than that in fact, really amazingly immense, a totally stunning size, "wow, that's big", time. Infinity is just so big that by comparison, bigness itself looks really titchy.”

[Douglas Adams, The Restaurant at the End of the Universe]
IoT structure

Edge “things”

Aggregation gateways

Network Access / Backhaul / Core Network

Cloud / Backend / IT
Gateways role evolves (home)

From single application to ecosystem enabler
Ecosystem applications

Purpose build apps [Vertical]

Ecosystem enabled apps [Horizontal]

„Hot today!“
IoT Alliances and Consortia
Connectivity complexity
Smart home applications

Application Layer Profiles
- AllSeen
- OIC
- IoTivity
- LW M2M
- Home Kit
- Weave

Network Layer
- Bluetooth Host Stack (optionally 6LoWPAN)
- TCP / UDP
- IPv4 / IPv6
- 6LoWPAN

Transport Layer
- Bluetooth Link Layer
- IEEE 802.11ac (vht rate)
- IEEE 802.15.4g
- IEEE 802.15.4 (low power)

Physical/ Link Layer (PHY/ MAC)
- Bluetooth Link Layer
- IEEE 802.11ac (vht rate)
- IEEE 802.15.4g
- IEEE 802.15.4 (low power)
# Smart Home – concerns

## Top 3 General Concerns About Smart Home Devices (Q4/14)

Among All BB HHs, n=4,991, ±1.39%

<table>
<thead>
<tr>
<th>Concern</th>
<th>% Citing Concern Among the Top 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Private information is stolen by hackers</td>
<td>73%</td>
</tr>
<tr>
<td>Installation of the devices is expensive</td>
<td>57%</td>
</tr>
<tr>
<td>Doors are accidentally locked or unlocked because the lock or associated app malfunctions</td>
<td>57%</td>
</tr>
<tr>
<td>Installation of the devices is difficult</td>
<td>34%</td>
</tr>
<tr>
<td>Electricity is wasted because automated settings are not optimal</td>
<td>29%</td>
</tr>
<tr>
<td>Garage door unexpectedly closes because the garage door or associated app malfunctions</td>
<td>28%</td>
</tr>
<tr>
<td>Pipes freeze because the thermostat or associated app malfunctions and turns the heat down too much</td>
<td>22%</td>
</tr>
</tbody>
</table>

---

"Q2911. What are your greatest concerns about connecting these types of devices to the Internet?" | Source: American Broadband Households and Their Technologies Q4 2014 | N=10,000, ±0.98% | © 2015 Parks Associates
Smart Home – key drivers

Personal and family security

Cost savings

Programming (yes!)

Source:
„One to rule them all”

Intelligent IoT Gateway on OpenWrt

- Self learning
- Scalability
- Interoperability
- Freedom
Intelligent IoT Gateway concept

Ecosystem enabler for horizontal applications

Multiradio Linux/ mbed Gateway + Intelligence + Virtualization

Ecosystem enabler for horizontal applications
Implementation

platform
radio(s)
layers
security
apps
demo
Platform Specification

- Platform: mid-range router / ~$100 / + we have it at home
  - HW: TP-LINK Archer C5 v1.20
  - CPU: Qualcomm Atheros QCA9558 (720 MHz) / mips arch
  - RAM: 128 MiB / 16MiB Flash
  - Networking: Wi-Fi (dual-band / 2.4GHz 11n + 5GHz 11ac) + eth switch
  - USB ports

- OpenWrt ready:
  - Target System (Atheros AR7xxx/AR9xxx)
  - Target Profile (TP-LINK Archer C5/C7)
  - Chaos Calmer, trunk r46693
1. Prepare development environment
   • git clone trunk OpenWrt sources (git clone git://git.openwrt.org/openwrt.git)
   • from menuconfig pick target system/profile + utilities at your own preference

   Target System (Atheros AR7xxx/AR9xxx)
   Target Profile (TP-LINK Archer C5/C7)

2. Reflash the router from the TP-LINK web UI
   • for later easy-use it’s good to enable luci feeds in OpenWrt

   luci-mod-admin-full. LuCI Administration - full-featured for full control
Adding PHY/MAC

- Application Layer Profiles:
  - AllSeen
  - OIC
  - IoTivity
  - LW M2M
  - Home Kit
  - Weave
  - ...

- Network Layer
  - Bluetooth Host Stack (optionally 6LoWPAN)
  - TCP / UDP
  - IPv4 / IPv6
  - 6LoWPAN

- Transport Layer
  - ZigBee

- Physical/Link Layer (PHY/MAC)
  - Bluetooth Link Layer
  - IEEE 802.11ac (vht rate)
  - IEEE 802.15.4g
  - IEEE 802.15.4 (low power)
Boosting WiFi

MU-MIMO

• „Wave 2” devices
• Spatial Division Multiplexing (SDM)
• Advanced form of beamforming
• Simultaneous AP-to-multiple-clients transmission
Boosting WiFi
MU-MIMO

• Benefits
  • Spectrum efficiency
    • improved combined DL throughput
  • Lower latencies

• Limitations?
  • Clients also have to support MU-MIMO (beamformees)
  • Downstream only – From AP to clients (sophisticated antenna systems and signal processing)
  • Limited number of clients can be supported (NSS-1)
Boosting WiFi
MU-MIMO – upgrading router

• PHY: Replace wifi NICs hardware
  • CUS223 (3x3 11ac) ---> WLE1200V5-22 CUS239 (4x4 11ac + mu-MIMO)

• SW: Upgrade ath10k driver (QCA99X0 support in 4.3-rc1)

• FW: ath10k 10.4.1.00007-1
Boosting WiFi
MU-MIMO – how to

• Create your own backports package from e.g. ath.git / linux-next or …*
  • It’s just important to make sure the QCA99X0 support is there
• Cross-compile the backports to get the wireless LKM
  toolchain-mips_34kc_gcc-4.8-linaro_musl-1.1.10/initial/bin/mips-openwrt-
  linux-musl-
• Upload (replace) the backports *.ko modules and firmware to FS, depmod etc.

• Problems?
  • Firmware for QCA99X0 crashes during bootup on BE mach with ath10k… (under
    investigation)
  • Processing power of the QCA9558 platform (do not expect maximum performance)
Adding BLE

- One of the most popular local connectivity protocols
- Many small cheap multi-purpose devices
- Low and ultra-low energy
- Easy for building various use cases

- 6LowPAN supported
  - IPv6 networking over BLE link
  - BT SIG: ISPS 1.0
  - Since kernel 3.17 (OpenWrt trunk for ar71xx has 4.1 now)
CUS223 -> CUS239 (Compex WLE1200V5-22)

LogiLink Bluetooth 4.0 (BT0015)
Adding BLE support

How to do it

• HW: Extend AP by plugging in BLE USB dongle
• SW:
  • Fetch and install: bluez-utils, bluez-libs, ip feeds
  • Enable kernel modules: kmod-bluetooth, kmod-bluetooth_6lowpan
• Compile new openwrt-ar71xx-generic-archer-c5-squashfs-*\.bin and flash the router
Adding BLE support
How to enable it

- Consider btmgmt ctrl tool (new tool)
- Check for bt0 interface now
- Play around with networking interface and firewall settings for packet forwarding

```
root@OpenWrt:# modprobe bluetooth_6lowpan
root@OpenWrt:# echo 1 > /sys/kernel/debug/bluetooth/6lowpan_enable
root@OpenWrt:# hciconfig hci0 reset
root@OpenWrt:# hcitool lescan
root@OpenWrt:# echo "connect 00:1B:DC:07:32:7E 1" > /sys/kernel/debug/bluetooth/6lowpan_control
```
Adding BLE support

Debugging

• use dynamic_debugs for bt kernel messages
  • enable it from menuconfig:
    • Global build settings ---> [*] Compile the kernel with dynamic printk

• check readlog –f
• check if you have all required crypto modules
Extending with Thread

- New standard designed for Smart Home/ IoT, all-in-one
  - mesh
  - IPv6
  - low energy
  - security
- First devices coming soon (beginning 2016)
- Same radio as in ZigBee
- Thread Group with big names: Nest, Samsung, Freescale, Qualcomm, …
Extending with Thread

How to do it

• Add **Thread Border Router** (mbed OS) over Ethernet
• Static IPv6 addressing to communicate with OpenWrt br-lan interface
• **Alternative way**: implement **Thread stack and Border Router in Linux/ OpenWrt** + connect **802.15.4 radio**
About Thread

• Based on 802.15.4 radio, IPv6 and low power networking (6loWPAN) standard (existing IEEE and RFC documents)

• Designed mainly for Connected Home apps with high impact on security aspects.

• Mesh topology with No Single Point of Failure to guarantee reliability

• Device types: Boarder Router, Router, REED (router-eligible end device), End Device
Unifying transport

- **Application Layer Profiles**
  - AllSeen
  - OIC
  - IoTivity
  - LW M2M
  - Home Kit
  - Weave

- **Network Layer**
  - Bluetooth Host Stack (optionally 6LoWPAN)
  - TCP / UDP
  - IPv4 / IPv6
  - 6LoWPAN
  - ZigBee

- **Physical/Link Layer (PHY/MAC)**
  - Bluetooth Link Layer
  - IEEE 802.11ac (vht rate)
  - IEEE 802.15.4g
  - IEEE 802.15.4 (low power)
IPv6 to unify

Application 0  Application n

Application Protocol  Application Protocol

IPv6 - A unified Convergence Layer for the home

802.15.4  802.11  Bluetooth 4.x
6LoWPAN - focus on low energy

- Fragmentation and reassembling
  - 1280 bytes IPv6 packets fragmented to fit into 127 bytes 802.15.4 frames

- Header compression mechanism
  - IPv6 header is 40 bytes long! – reducing transmission overhead

- Link layer packet forwarding
  - Thread is using IP layer routing with link layer packet forwarding
Archer C7 (OpenWrt Linux) + Mbed 6LoWPAN (thread gateway router) + BLE Gateway

IPv6 networking
Selecting application layer

Network Layer
Transport Layer

Bluetooth Host Stack
(optionally 6LoWPAN)

TCP / UDP
IPv4 / IPv6
6LoWPAN

ZigBee

Physical/ Link Layer
(PHY/ MAC)

Bluetooth Link
Layer
IEEE 802.11ac
(vht rate)
IEEE 802.15.4g
IEEE 802.15.4 (low power)
App transfer protocols

IoT

CoRE
CoAP
 ...

Web/Internet

{ REST }
HTTP
What is CoAP?

“The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things.

The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation.” (res. http://coap.technology/)

• Open standard
• REST model (resource access)
• GET, PUT, POST, DELETE, …methods to
CoAP over Thread – demo

console

Router – „Device Server"

Thread Border Router

Thread Node

CoAP over Thread – demo

IEEE 802.11

Router – „Device Server”

Thread Border Router

Thread Node

IEEE 802.15.4/6LoWPAN
(Thread)

IEEE 802.15.4/6LoWPAN
(Thread)

POST

CoAP ACK

coop-client -m post coap://[fd00:0ff1:ce0b:a5e0:fcc2:3d00:4:a5ee]/resources/led

GET

RESP (counter)

cnt=$(root/coap-client/coap://[fd00:0ff1:ce0b:a5e0:fcc2:3d00:4:a5ee]/resources/button)
Applications and containers

Examples

- Spook Home (smart home + AI)
- Smart Home (routing, notifications, rules)
- Hello world
- Hello world
- Docker
- ZigBee

Network Layer
- Bluetooth Host Stack (optionally 6LoWPAN)
- TCP / UDP
- IPv4 / IPv6
- 6LoWPAN

Transport Layer
- Bluetooth Link Layer
- IEEE 802.11ac (vht rate)
- IEEE 802.15.4g
- IEEE 802.15.4 (low power)

ApplicationLayer Profiles
- AllSeen
- OIC
- IoTivity
- LW M2M
- Home Kit
- Weave
AI in applications

- Mining and analysis for aggregated data
- Learning trends = habits
- Levels of notifications based on probability
- Fuzzy logic
- Cost saving (water, electricity)
- Security improvement
- Feedback based learning

- Example below
Containers

OPKG
Native apps
packages

LXC
(available via feeds)

...
<table>
<thead>
<tr>
<th>Security</th>
<th>Application</th>
<th>System</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fine grained security</td>
<td>Security holes like Heartblead, ShellShock or backdoors in devices</td>
<td>Security services for link layer (auth, data integrity, confidentiality, reply protection)</td>
</tr>
<tr>
<td></td>
<td>Data aggregation and privacy protection (for cloud)</td>
<td>Keep your system up to date - security on the level of last patch</td>
<td>IP networking based security mechanisms (access control, firewalls, ...)</td>
</tr>
<tr>
<td></td>
<td>Information integrity check by AI (data correlation, trends analysis etc)</td>
<td>Open source</td>
<td>Keep to standards, e.g. DTLS (RFC6347 v1.2)</td>
</tr>
</tbody>
</table>
Demo - „Spook House“

example smart application for lighting

- Light control system
- Learning light usage trends
- Notifications based on deviations and likeliness
- Focused on cost saving (i.e. recommendations) and security (notifications when out of trend)
Demo - „Spook House”

Description

• running on IOT Gateway and receives data from the IoT network sensors
• monitor usage of home lights and detect unusual events. For instance if someone forgets to turn off the light.
• Application gathers statistical information about how often and how long light are turned on.
• Since users activity changes during the week (for instance on weekends) the algorithm takes into account not only time of day but also day of week.
• Application will notice if lights in some room are turned on for unusually long time.
Demo - „Spook House”

Internals

- **StateTracker**
  - Receives data in form of events about state change
    - "Device": "light3", "EventType": "SHStateChange", "State": "On", "Timestamp": 122848.71204376257
    - "Device": "light3", "EventType": "SHStateChange", "State": "Off", "Timestamp": 123907.1497501683
  - Communicate with TrendMonitor and external modules

- **TrendMonitor (learning)**
  - Keeps a data base with statistics
  - Implements algorithm to calculate and detect unusual situations (deviations) and notify ActionManager about such.

- **ActionManager (learning)**
  - Receives 'SHActionRequests' about deviations and makes decision on what to do with it (e.g. notify user)
  - Options: User feedback loop

- **Clock or internal interrupt to throw ’SHtimer’ event – check the House state.**
Wrap-up
Intelligent IoT Gateway on OpenWrt

Thank you!

Andrzej Wieczorek  Bartosz Markowski