Bluetooth High Speed and Virtual / Soft AMP Controller for mac80211

Andrei Emeltchenko / Intel OTC
Wireless mini-summit
Barcelona
09.11.2012
Contents

• Bluetooth High Speed and how it works
• Types of Bluetooth High Speed Controllers (AMPs)
• Current Bluetooth High Speed stack implementation
• SoftAMP implementation proposals
Bluetooth speed evolution

- **Basic Rate (BR)**
  - Specification versions: 1.0 – 2.1
  - Data rate up to 1 Mbit/s

- **Enhanced Data Rate (EDR)**
  - Specification versions: 2.0+EDR – 2.1+EDR
  - Data rate up to 3 Mbit/s

- **Bluetooth High Speed**
  - Specification versions: 3.0+HS, 4.0
  - Data rate up to 24 Mbit/s
High Speed main ideas

• Utilize secondary radio for fast data transfer
  • 802.11 radio is specified
  • Only data is transferred through HS, negotiation and establishment is done through BR/EDR.

• Using existing HW
  • Usually both Bluetooth and WIFI are on the same chip
  • Cheap to implement for manufacturer

• Transparent to user
  • Support existing Bluetooth profiles
  • No new paradigm
Is this that simple?

• To make Bluetooth High Speed working with obex

...  
+  
+          {  
+              int chan_policy = BT_CHANNEL_POLICY_Amp_PREFERRED;  
+          +  
+              if (setsockopt(sock, SOL_BLUETOOTH, BT_CHANNEL_POLICY,  
+                      &chan_policy, sizeof(chan_policy)) < 0)  
+                      goto failed;  
+          }  
...  

• Currently also need kernel option to enable High Speed

options bluetooth enable_hs=y
Start Bluetooth BR/EDR connection

- Establishing BR/EDR Physical Link and ACL transport
- Query AMP Fixed channel with L2CAP Info Request
- Open L2CAP AMP Manager Fixed Channel
- A2MP AMP Discover Request
- A2MP AMP Discover Response

- AMP Manager supported
- Receive Remote AMP controller list
- Return HCI devices from the HCI dev list

Host & BR/EDR 1  AMP 1  AMP 2  Host & BR/EDR 2
Remote Controller list

• Available remote controllers

• Contains:
  – Controller id
  – Controller type
    • BR/EDR
    • 802.11
  – Controller status
    • AMP status and capacity information

Controller list:
  id 0 type 0 (BR-EDR) status 0x01 (Bluetooth only)
  id 1 type 1 (802.11 AMP) status 0x01 (Bluetooth only)
Get remote AMP controller Info and Assoc

- A2MP Get AMP Info Request
  - HCI Read Local AMP Info
- A2MP Get AMP Info Response
  - A2MP Get AMP Assoc Request
    - HCI Read Local AMP Assoc
- A2MP Get AMP Assoc Response

Contains bandwidths Info, latency and AMP Assoc size

Receive AMP Assoc data

Host & BR/EDR 1  AMP 1  AMP 2  Host & BR/EDR 2
AMP Assoc structure

• Most important data in High Speed communication

• Contains:
  – MAC Address
  – Preferred Channel List
  – Connected Channel List
  – PAL Capabilities
  – PAL version

Assoc data [len 36]:
  MAC: 00:50:43:21:30:F9
  Preferred Chan List (number of triplets 2)
    Country code: US
    Reg ext id 201 reg class 12 coverage class 0
    Channels 1 - 12 max power 20
  PAL CAP: 03 00 00 00
  PAL VER: 01 Comp ID: 0048 SubVer: 0001
Establish High Speed Physical Link

- **A2MP Create Physical Link Request**
  - HCI Create Physical Link
  - HCI Write Remote AMP Assoc
  - Scanning & Start Beaconing
  - HCI Channel Selected
  - HCI Read Local AMP Assoc

- **A2MP Create Physical Link Response**
  - Include Local AMP Assoc in Request
  - Read selected channel and other Assoc info
  - Start Beaconing

- **Host & BR/EDR 1**
- **AMP 1**
- **AMP 2**
- **Host & BR/EDR 2**
High Speed Link Security

PMK is provided in HCI Create Physical Link

HCI Physical Link Completed

4-way handshake: Generating PTK from PMK

802.11 AUTH

802.11 ASSOC

PMK is provided in HCI Accept Physical Link

HCI Physical Link Completed

Host & BR/EDR 1

AMP 1

AMP 2

Host & BR/EDR 2
Create High Speed L2CAP channel and logical link

L2CAP Create Channel Request/Response

L2CAP Configure Request

L2CAP Configure Response (Pending)

HCI Create Logical Link

L2CAP Configure Request

L2CAP Configure Response (Pending)

HCI Create Logical Link

HCI Local Logical Link Completed

L2CAP Configure Response (Success)

L2CAP Configure Response (Success)

L2CAP Socket Ready to send

L2CAP Socket Ready to receive

Host & BR/EDR 1

AMP 1

AMP 2

Host & BR/EDR 2
Send data over High Speed Link

- Host & BR/EDR 1
  - HCI send
  - ACL header
  - L2CAP header
  - DATA

- AMP 1
  - wireless send
  - IEEE802.11 header
  - LLC SNAP header
  - L2CAP header
  - DATA

- AMP 2
  - HCI receive
  - ACL header
  - L2CAP header
  - DATA

- Host & BR/EDR 2
High Speed data packet
## Types of AMP

<table>
<thead>
<tr>
<th></th>
<th>FullAMP</th>
<th>SoftAMP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HW</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMP Controller</td>
<td>PAL</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAC</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PHY</td>
<td></td>
</tr>
<tr>
<td><strong>OS Bluetooth stack</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HCI Interface</strong></td>
<td>USB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UART</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SDIO</td>
<td></td>
</tr>
<tr>
<td><strong>Wireless driver</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>USB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UART</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SDIO</td>
<td></td>
</tr>
<tr>
<td>802.11 wireless card</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bluez stack

User space

Bluetooth Profiles
Socket Interface

Linux Kernel

MGMT
SCO
ACL
L2CAP
RFCOMM
HCI core

HCI Interface
USB
UART
SDIO

Firmware & HW

BR/EDR Controller
BlueZ stack with High Speed support

User space
- Bluetooth Profiles
- Socket Interface
- RFCOMM
- ACL
- SCO
- MGMT
- L2CAP
- A2MP

Linux Kernel
- HCI core
- HCI Interface
- USB
- UART
- SDIO

HW
- BR/EDR Controller

Configuration SW
- Socket Interface
- RFCOMM
- ACL
- SCO
- MGMT
- L2CAP
- A2MP

Configuration SW
- Socket Interface
- RFCOMM
- ACL
- SCO
- MGMT
- L2CAP
- A2MP

BlueZ stack with High Speed support
- BlueZ stack
- High Speed support
- USB
- UART
- SDIO
- AMP Controller
Upstream support

- A2MP – kernel 3.5
- Basic AMP configuration – kernel 3.7
- HCI core and L2CAP – kernel 3.8 (currently in bluetooth-next)
- L2CAP channel move full support – kernel 3.8 – 3.9
BlueZ and SoftMAC wireless card

User space:
- Bluetooth Profiles
- Socket Interface

Linux Kernel:
- HCI core
- HCI Interface
- USB
- UART
- SDIO

HW:
- BR/EDR Controller

Configuration SW:
- Socket Interface

User space:
- nl80211
- SOFTMAC WIFI

User space:
- User space

Linux Kernel:
- HCI core
- HCI Interface
- cfg80211
- cfg80211_ops
- mac80211

Software:
- A2MP
- L2CAP

Configuration SW:
- NO AMP

User space:
- Configuration SW
- User space

User space:
- User space

User space:
- User space

User space:
- User space

User space:
- User space
SoftAMP tasks

• Implement registration of Bluetooth HCI AMP device
  • Need to provide convenient way to start/stop SoftAMP.

• Handle HCI commands and events.
  • Process HCI commands and send events.

• Security – auth, assoc, 4 way handshake

• Data management
  • Send: remove HCI header, add LLC/SNAP, add IEEE802.11 header.
  • Receive: remove IEEE802.11 header, remove LLC/SNAP header, add HCI header.

• Channel management and bandwidth sharing
Security

• 4 way handshake derives PTK from PMK

• PMK is derived from Bluetooth Link Key (LK)
  • Derive Generic AMP Key (GAMP) from LK
    • \( \text{GAMP}_LK = \text{HMAC-SHA-256}(LK||LK, 'gamp', 32) \)
  • Derive Dedicated AMP Link Key which is PMK
    • \( \text{PMK} = \text{HMAC-SHA-256}(\text{GAMP}_LK, '802b', 32) \)
Methods of implementing SoftAMP

- In-kernel
- User-space driven
Common parts of SoftAMP

• Make SoftAMP a part of mac80211 (selectable?)
  • Any SoftMAC card might be used

• Implemented as a new virtual interface type in mac80211
  • SoftMAC drivers shall support this new type

• Data packets goes directly between bluetooth and wireless stacks
In-kernel SoftAMP

- As much as possible is done in the kernel
  - Even 4-way handshake?

- User-space knows very little that High Speed is used
  - Used mostly for enabling / disabling.

- SoftAMP can be created with any standard tool like:
  $ iw phy phy0 interface add softamp type softamp
In-kernel SoftAMP interfaces

User space
- Bluetooth Profiles
- Socket Interface
- Configuration SW
- User space
- SoftMAC WIFI card

Linux Kernel
- HCI core
- HCI Interface
- USB
- UART
- SDIO
- A2MP
- L2CAP
- ACL
- MGMT
- SCO

HW
- BR/EDR Controller
- SoftAMP
- SoftMAC WIFI card
User-space driven SoftAMP

• User space is controlling many aspects of High Speed setup

• SoftAMP is created and managed through special nl80211 API messages.

• All data except for data frames are sent to wireless user space components (wpa_supplicant).
User-space driven SoftAMP. Johannes Berg ideas:

• `cfg80211` add/remove interface can get a new interface type, to also create/remove the hci

• `ifup/down` will be hci up/down

• new `nl80211` API to start/stop the AMP (start/stop beaoning)

• new `nl80211` API to RX/TX HCI messages so userspace can handle parts of the HCI protocol

• new `nl80211` API to RX/TX AMP PDUs, with the given `AMP_PROTO`, so userspace can handle 4-way-handshake etc.
User space driven SoftAMP interfaces

User space
- Bluetooth Profiles
- Socket Interface
- Configuration SW
- Controlling SW
- User space

Linux Kernel
- MGMT
- SCO
- ACL
- HCI core
- HCI Interface
- USB
- UART
- SDIO
- A2MP
- L2CAP
- ACL
- HCI core
- HCI Interface
- SoftAMP

HW
- BR/EDR Controller
- SoftMAC WIFI card
- mac80211
- cfg80211
- cfg80211_ops
Abbreviation

- A2MP – AMP Manager Protocol
- ACL – Asynchronous Connection-oriented link
- AMP – Alternate MAC / PHY Controller
- L2CAP – Logical Link Control and Adaptation Protocol
- PAL – Protocol Adaptation Layer
- SCO – Synchronous Connection-Oriented link
Questions?