**WARP Protocol**

**Version: 1.5**

(Updated 31st October 2014 by Hoai Phuoc Truong)

The protocol below aims to support internal two way communication between WARP board and PC Engine over the Ethernet. Modification to the protocol should ensure modularity and extensibility to support future developments and additional features. Also, this design avoids fields with variable lengths for simple implementation in WARP.

**I) General architecture of WARP Protocol:**

A typical packet from PC Engine to WARP (and vice versa) will have the following:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ethernet | WARP | | | Encapsulated dot11 data  (if applicable) |
| WARP Header | | WARP Data |
| Type | Subtype | (See below) |

**II) WARP Header:**

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Subtype | Expected WARP Element | Expected data at the end |
| 0 (Ignore) (Default value) | Any | None | No |
| 1 (Transmit) | 0 | Transmit Element | Yes |
| 1 |
| 2 (Control) | Any | Control Element | No |

|  |  |  |
| --- | --- | --- |
| Type | Subtype | Meaning |
| 0 (Ignore) (Default value) | Any | Ignore (for testing purpose) |
| 1 (Transmit) | Any | Management frame transmit |
| 2 (Control) | 0 | Ignore |
| 1 | Fetch hardware configurations |
| 4 | MAC address database management (see MAC Address Management) |
| 8 | Transmission database configuration (see Transmission Config) |
| 12 | Station management (see BSSID Control Element) |

**III) WARP Elements:**

**1) Transmit Elements:**

a) WARP Transmit Element:

|  |  |
| --- | --- |
| Field | Data length |
| Size (byte) | 2 |
| Default | 0 |
| Note | \* |

\* First byte represents most significant 8 bits of the length. Second byte represents least significant 8 bits of the length.

**2) Control Elements:**

a) Fetch Hardware Configurations Element:

Not defined yet

b) WARP BSSID Control Element:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | WARP BSSID Control Element | | | |
| Total number of elements | bssid | Operation code | Station MAC address |
| Size (byte) | 1 | 6 | 1 | 6 |
| Default | 1 | 00:00:00:00:00:00 | 0 | 00:00:00:00:00:00 |
| Note | \* |  |  | \*\* |

\* If total number of elements is not 1, there would be n = total number of elements station MAC addresses in the control element. WARP would apply the operation using operation code to all station MAC address.

\*\* For certain operation code (see below), this field would not be present.

|  |  |  |
| --- | --- | --- |
| Operation code (PC Engine to WARP) | Meaning | Note |
| NOTHING (0) | Do nothing |  |
| STATION\_ASSOCIATE(D) (1) | Add |  |
| SYNC (16) | Send current database back to PC Engine for synchronization | \* |
| STATION \_DISASSOCIATE(D) (32) | Remove |  |
| STATION\_CLEAR(ED) (33) | Remove all stations in database | \* |
| STATION\_CHECK\_IF\_EXIST (64) | Check if exist (from PC Engine to WARP) |  |

.

|  |  |  |
| --- | --- | --- |
| Operation code (WARP to PC Engine) | Meaning | Note |
| STATION\_LIST\_FULL (2) | Cannot associate new station because the station list is full |  |
| STATION\_EXISTED (65) | Existing association |  |
| STATION\_NOT\_EXISTED (66) | Association not existed |  |
| BSSID\_STATION\_CONFLICT (70) | Conflict request and existing data of (bssid, station) pair |  |
| BSSID\_NOT\_EXISTED (75) | Required operation cannot be executed since bssid does not exists in WARP | \* |

\* For these operation codes, WARP/PC Engine expects no station MAC address

c) WARP Transmission Control Element:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field |  | | WARP Transmission Control Element | | | | | | |
| Total number of elements | Operation code | | bssid | Disabled | TxPower | Channel | Rate | HW mode |
| Size (byte) | 1 | 1 | | 6 | 1 | 1 | 1 | 1 | 1 |
| Default | 1 | 0 | | 00:00:00:00:00:00 | 0 | 0 | 2 | 1 | 0 |
| Note | \* |  | | \*\* |  |  |  |  | Not defined yet |

\* For total number of elements is n, there would be n bssid(s) with their configurations followed.

\*\* This bssid should have been added previously by the PC Engine to MAC low. WARP will verify that the bssid is in the database. If not, WARP simply ignores this packet.

PC Engine would send the control element to WARP. WARP will attempt to modify/ add the configuration to fit the requirement and then reply to the PC Engine using the same format as the control message received initially.

|  |  |
| --- | --- |
| Operation code (PC Engine to WARP) | Meaning |
| NOTHING (0) | Do nothing |
| CONFIGURE (1) | Configure/ Add new configuration for a bssid |
| VERIFY (4) | Verify for inconsistency in the configuration of this radio |

|  |  |
| --- | --- |
| Operation code (WARP to PC Engine) | Meaning |
| VERIFIED (5) | Configuration verified and there is no inconsistency |
| INCONSITENT (6) | Inconsistency in the WARP database and the sent configuration |
| CONFIGURE\_SUCCESSFUL (8) | Existing (reply from WARP to PC Engine) |
| CONFIGURE\_FAIL (16) | Not existed (reply from WARP to PC Engine) |

d) WARP MAC Address Management Element:

|  |  |  |
| --- | --- | --- |
| Field | WARP MAC Address Control Element | |
| Operation code | MAC Address |
| Size (byte) | 1 | 6 |
| Default | 0 | 00:00:00:00:00:00 |

|  |  |
| --- | --- |
| Operation code (PC Engine to WARP) | Meaning |
| NOTHING (0) | Do nothing |
| MAC\_ADD(ED)(1) | Add |
| MAC\_REMOVE(D) (32) | Remove |
| MAC\_CLEAR(ED) (33) | Remove all MACs in database |
| MAC\_CHECK\_IF\_EXIST (64) | Check if exist (from PC Engine to WARP) |

|  |  |
| --- | --- |
| Operation code (WARP to PC Engine) | Meaning |
| MAC\_EXISTED (65) | Existing (reply from WARP to PC Engine) |
| MAC\_NOT\_EXISTED (66) | Not existed (reply from WARP to PC Engine) |