Proposed DPP Burst Structure for Applications Characterized by Short Messages and Low duty Cycle (Single PDU per Burst)

# Introduction

This document outlines a proposed DPP Burst structure optimized for applications characterized by short messages and low duty cycle resulting in a single PDU per burst. The IEEE802.16t DPP burst structure is shown below:



## Gain Adjustment Sequence and Synchronisation sequence/Preamble length.

The IEEE802.16t DPP burst Gain Adjustment sequence is currently fixed at 54 bits, while the preamble length is set to 63 bits. Simulation analyses are required across various MCS and repetition configurations to determine the optimal sequence lengths to make sure these sequences are robust enough.

## CTRL-MSG optimisation

The IEEE802.16t DPP CTRL-MSG is a fixed-length message. Based on the “Control Message Type” field, it describes various usage of the message. The proposed structure for short message and duty cycle application, is described below.

|  |
| --- |
| * CTRL Message
 |
| Syntax | Size (bits) | Notes |
| Control Message () { | --- | --- |
| Control Message Type | 2 | This field indicates the type of CTRL MSG based on what description it is carrying.Value 0: DPP PDU   1: RTS   2: CTS   3: ACK |
| Relay Status | 1 | 0: Original transmission, 1: Relay transmission |
| Relay Option | 2 | Value 0: Direct transmission only, No Relay   1: Relay   2: Relay based on ACK failure |
| AUTHI | 1 | Authentication. 0: Disabled, 1: Valid CMAC/HMAC is present |
| If (AUTHI == 0) { |  |  |
|  If (small Packet Size == 1) { |  |  |
|  Sender ID | 8 | Unique ID assigned to the Sender SS. |
|  Receiver ID | 8 | Unique ID assigned to the Receiver SS. |
|  } else { |  |  |
|  Sender ID | 48 | MAC address of the Sender SS |
|  Receiver ID  | 48 | MAC address of the Receiver SS |
|  } |  |  |
| } |  |  |
| if (control message type == 1) { |  |  |
|  If (small Packet Size == 1) { |  |  |
|  Requested Bytes | 8 | Total bytes to transmit, including DPP PDU and SDU overheads |
|   } else {Requested Bytes | 16 | Total bytes to transmit, including DPP PDU and SDU overheads |
|  } |  |  |
|  Reserved | ~~3~~10 |  |
| } |  |  |
| else if (control message type == 3) { |  |  |
|  If (small Packet Size == 1) { |  |  |
|   ACK bitmap | 8 | LSB applies to first DPP PDU and MSB to last.Bit value 1 indicates ACK. Maximum number of DPP PDUs in a burst shall not exceed 8. |
|   }else {ACK bitmap | 16 | LSB applies to first DPP PDU and MSB to last.Bit value 1 indicates ACK. Maximum number of DPP PDUs in a burst shall not exceed 16. |
|  } |  |  |
|  Reserved | ~~3~~10 |  |
| } else { |  |  |
|  MCS | 4 | MCS includes the Repetition. Refer to Table 18-3 |
|  ACK1 | 1 | ACK Indication. 0: disabled, 1: enabled |
|  If (small Packet Size == 1) { |  |  |
|   Number of Bytes | 8 | Number of Bytes allocated (in CTS/PDU) post CTRL MSG. |
|   else {Number of slots | 12 | Number of slots allocated (for CTS/PDU) post CTRL MSG. |
|   Reserved | 1 |  |
|  } |  |  |
| } |  |  |
| ~~Nonce~~ | ~~8~~ | ~~Bits derived from the truncation of the previous HMAC~~ |
| ~~}~~ |  |  |
|  If (small Packet Size == 1) { |  |  |
|  PDU Header Type | 1 | 0: Management DPP PDU 1: Data DPP PDU |
|  PHS index | 3 | Values 0: PHS disabled, 1 to 7 PHS rules. |
|  Reserved |  1 | Set to 0 |
|  } |  |  |
| CRC | 8 | CRC for the above bytes computed per 6.3.3.5. |
| if (AUTH == 1) { |  |  |
|  Nonce | 8 | Bits derived from the truncation of the previous HMAC |
| HMAC Digest | *variable* | HMAC is a keyed hash. This is calculated over CTRL-MSG and all PDUs in the burst, excluding the HMAC field. If AUTHI is 0 then this field is not transmitted, when AUTHI is set to 1 this will be present after the CRC.  |
| } |  |  |

The CTRL-MSG without HMAC Digest is 6 Bytes, MCS QPSK1/2, one slot is sufficient to transmit the CTRL-MSG.

## PDU overhead optimisations

All the header parameters needed to decode the PDU are embedded in the CTRL-MSG. Due to smaller packet sizes, instead of a 4-byte CRC, a 2-byte CRC can be used for the PDU and when Authentication is enabled HMAC can be used instead of CRC to check the data integrity.