IEEE P802.11
Wireless LANs

|  |
| --- |
| SB2 Comment Resolution for Annex G |
| Date: 2016-04-15 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Yongho Seok | NEWRACOM |  |  |  |

Abstract

This submission proposes resolutions of comments received from TGah 2nd Sponsor Recirculation Ballot (TGah Draft 7.0).

* CIDs: 10001, 10022 (2 CID)

Interpretation of a Motion to Adopt

A motion to approve this submission means that the editing instructions and any changed or added material are actioned in the TGah Draft. This introduction is not part of the adopted material.

***Editing instructions formatted like this are intended to be copied into the TGah Draft (i.e. they are instructions to the 802.11 editor on how to merge the text with the baseline documents).***

***TGah Editor: Editing instructions preceded by “TGah Editor” are instructions to the TGah editor to modify existing material in the TGah draft. As a result of adopting the changes, the TGah editor will execute the instructions rather than copy them to the TGah Draft.***

| **CID** | **Page** | **Clause** | **Comment** | **Proposed Change** | **Resolution** |
| --- | --- | --- | --- | --- | --- |
| 10001 | 537.00 | B4.4.2 | I disagree with the resolution of CID 9004. The comment was rejected with the note, the commenter doesn't indicate specific changes to satisfy the comment. This is not true. The specific change is to create Annex G and define the main exchange sequences to be added to the annex. For example the draft identified all types of NDP frames, ACK, RTS, CTS, etc. At least the related seuqnces should be identified and added to the Annex. | Missing a normative Annex doesn't help. As in the comment, please create Annex G and identify the main exchange sequences to be added. | Revised- A current Annex G in the 802.11 spec does not cover all frame exchange sequences. But, as per request from the commenter, TGah BRC accepts to create Annex G for the main exchange sequences based on the NDP frames. TGah Editor makes changes as shown in the as specified in 11-16/0559r0 (URL). |
| 10022 | 611.00 | G | The new frame types created by this amendment are not described in any allowable frame exchange sequence in Annex G. While previous amendments may have failed to modify Annex G, it is a normative Annex and has normative references in the main body text that are necessary for interoperability, so amendments that add frame types must modify Annex G appropriately. (REVmc is likely going to have to clean-up from those amendments that did not do this in the past.) | Add new frame types from this amendment to appropriately modified/new sequences in Annex G. | Revised- A current Annex G in the 802.11 spec does not cover all frame exchange sequences. But, as per request from the commenter, TGah BRC accepts to create Annex G for the main exchange sequences based on the NDP frames. TGah Editor makes changes as shown in the as specified in 11-16/0559r0 (URL). |

**G.2 Basic sequences**

***TGah editor: modify the sub-clause G.2 as the following:***

The allowable frame exchange sequence is defined by the rule frame-exchange-sequence. Except where modified by the *pifs* attribute, frames are separated by a SIFS.

(\* This rule defines all of the allowable frame exchange sequences \*)

frame-exchange-sequence =

([**CTS**] (**Management** +*broadcast* | **Data** +*group*)) |

([**CTS** | **RTS CTS** | **PS-Poll**] {frag-frame **Ack**} last-frame **Ack**) |

(**PS-Poll Ack**) |

([**Beacon** +*DTIM* ] {cf-sequence} [**CF-End** [+*CF-Ack*] ])|

hcf-sequence |

mcf-sequence |
s1g-sequence ;

**G.3 EDCA and HCCA sequences**

***TGah editor: modify the sub-clause G.3 as the following:***

(\* These frames provide acknowledgment to the txop-part-requiring-ack \*)

txop-part-providing-ack=

**Ack** | **NDP-Ack |**

cf-ack-piggybacked-qos-poll-sequence |

(\* An HC responds with a new polled TXOP on expiration of current TXOP \*)

cf-ack-piggybacked-qos-data-sequence |
(\* An HC responds with CF-Ack and its own data on expiration of TXOP \*)

**Data** +*CF-Ack*;

**G.4 HT and VHT and S1G sequences**

***TGah editor: modify the sub-clause G.4 as the following:***

(\* This is an initiator sequence. The different forms arise from whether the initiator transmits a frame that requires a BlockAck frame, and whether it delivers an RDG. When an RDG is delivered, the response is distinguished according to whether it demands a BlockAck frame response from the initiator. \*)

initiator-sequence = (\* No BlockAck frame expected, no RDG \*)

burst |

(\* block ack request delivered, BlockAck frame expected. No RD \*)

(burst-bar (**BlockAck**[+*HTC*] **| Ack**~~)~~ [+*HTC*] | **NDP-BlockAck | NDP-Ack**) |

(\* No block ack request delivered, RDG \*)

(burst-rd (

burst |

burst-bar initiator-sequence-ba

)

) |

(burst-rd-bar (**BlockAck**[+*HTC*] **| Ack**~~)~~ [+*HTC*]| **NDP-BlockAck | NDP-Ack**) |

(burst-rd-bar (

burst-ba |

burst-ba-bar initiator-sequence-ba

)

) |

ht-ack-sequence |

psmp-burst |

link-adaptation-exchange ;

(\* This is the same as the initiator-sequence, except the initiator is constrained to generate a BlockAck frame response because a previous RD response contained a block ack request \*)

initiator-sequence-ba = burst-ba |

(burst-ba-bar (**BlockAck**[+*HTC*] **|Ack**~~)~~[+*HTC*~~)~~] | **NDP-BlockAck | NDP-Ack**) |

(burst-ba-rd (

burst |

burst-bar initiator-sequence-ba

)

) |

(burst-ba-rd-bar (**BlockAck**[+*HTC*] **|Ack**~~)~~[+*HTC*] | **NDP-BlockAck | NDP-Ack**) |

(burst-ba-rd-bar (

burst-ba |

burst-ba-bar initiator-sequence-ba

)

);

(\* These are sequences that occur within an ht-txop-sequence that have an ack response \*)

ht-ack-sequence = (**BlockAck**+*delayed*[*+HTC*] [*+mu-user-respond* other-users]**Ack**[+*HTC*]) |

(**BlockAckReq**+*delayed*[*+HTC*][*+mu-user-respond* other-users] **Ack**[+*HTC*]) |

**(Data**[*+HTC*]+*individual*[+*null*][+*QoS+normal-ack*][*+mu-user-respond* otherusers]

**(Ack**[+*HTC*] **| NDP-Ack))**;

(\*A PPDU containing BlockAck frame is either a non-A-MPDU BlockAck frame, or an A-MPDU

containing a BlockAck frame, and also containing data that does not carry implicit block ack request. \*)

ppdu-ba= ( **BlockAck**[*+HTC*] | **NDP-BlockAck |**

(

**BlockAck**[*+HTC*]+*a-mpdu*

1{**Data**[*+HTC*]+*QoS+*(*no-ack*|*block-ack*)+*a-mpdu*}

) + *a-mpdu-end;*

) [+*mu-user-respond* other-users];

(\* The VHT beamforming sequence starts with a VHT NDP Announcement frame, followed by a VHT NDP. One of the STAs in the sequence responds immediately with explicit feedback. The VHT AP might transmit Beamforming Report Poll to poll the other STAs to obtain their feedback before generating an MU transmission. The S1G AP might transmit either Beamforming Report Poll or NDP Beamforming Report Poll to poll the other STAs to obtain their feedback before generating an MU transmission. The names of the frames include spaces, so they are delimited using parentheses. \*)

vht-bf =

(**VHT NDP Announcement**) (**VHT NDP**) vht-feedback

{(**Beamforming Report Poll | NDP-Beamforming-Report-Poll**) vht-feedback};

***TGah editor: add the following at the end of the sub-clause G.4***

(\* An s1g-sequence represents additional sequences that may be generated by an S1G STA using EDCA channel access.\*)

s1g-sequence =

([**CTS** | **NDP-CTS**] 1{(**Data** +*group* +*QoS* | **Management** +*broadcast*) +pifs} |

([**CTS** | **NDP-CTS** | **RTS** (**CTS** | **NDP-CTS**)| **PS-Poll** | **NDP-PS-Poll**] {frag-frame (**Ack | NDP-Ack**)} last-frame (**Ack | NDP-Ack**)) |

(**PS-Poll** **Ack**) | (**NDP-PS-Poll NDP-PS-Poll-Ack**) |

([**CTS | NDP-CTS**] 1{s1g-txop-sequence});

(\* A TXOP may be filled with s1g-txop-sequences, which are initiated by the S1G TXOP holder. \*)

s1g-txop-sequence =

(((**RTS** (**CTS** | **NDP-CTS**)) | **CTS**+*self*| **NDP-CTS** +*self*) **Data** +*individual* +*QoS* +(*block-ack* | *no-ack*)) |

[**RTS** (**CTS | NDP-CTS**)] (txop-part-requiring-ack txop-part-providing-ack )|

[**RTS** (**CTS | NDP-CTS**)] **(Management** | (**Data** +*S1GAP*)) +*individual* (**Ack | NDP-ACK**)|

[**RTS** (**CTS | NDP-CTS**)] **(BlockAckReq** (**BlockAck | NDP-BlockAck**)) |

s1g-nav-protected-sequence |

1{initiator-sequence};

(\* an s1g-nav-protected sequence consists of setting the NAV, performing one or more initiator-sequences and then resetting the NAV if time permits \*)

s1g-nav-protected-sequence = s1g-nav-set 1{initiator-sequence} [s1g-resync-sequence] ;

(\* These are the series of frames that establish NAV protection for an S1G sequence \*)

s1g-nav-set = (**RTS**[+HTC] (**CTS**[+HTC]) | **NDP-CTS)** |

**CTS**+*self* **| NDP-CTS**+*self* |

**(Data**[+HTC]+*individual*[+*null*][+*QoS*+*normal-ack*] (**Ack | NDP-Ack))** |

**Data**[+HTC]+individual+QoS[+(*no-ack*|*block-ack*)] |

**Data**+group[+null]+QoS |

(1{ **Data**[+HTC]+*individual*+*QoS*+*implicit-bar*+*a-mpdu*}+*a-mpdu-end* (**BlockAck**[+HTC] | **NDP-BlockAck**)) |

(**BlockAckReq**[+HTC] (**BlockAck**[+HTC] | **Ack**[+HTC]) | **NDP-BlockAck | NDP-Ack**)) |

1{s1g-rts-cts};

(\* The s1g-rts-cts term applies to RTS transmitted by an S1G STA to another S1G STA. When the RTS is transmitted using an S1G non-duplicate PPDU or S1G 2 MHz duplicate PPDU, the transmission of the RTS is delayed so that at least a PIFS has elapsed since the previous frame exchange sequence (see 10.22.2.7 (Multiple frame transmission in an EDCA TXOP)) and the RTS is transmitted with a Dynamic Indication field set to 1 (see 10.3.2.6 (VHT and S1G RTS procedure)). \*)

s1g-rts-cts = **RTS**+*pifs* [+*HTC*] (**CTS**[+*HTC*] | **NDP-CTS**);

s1g-resync-sequence = **CF-End | (CF-End+***non-S1GAP* **CF-End+** *S1GAP***) | NDP-CF-End | (NDP-CF-End+***non- S1GAP* **NDP-CF-End+** *S1GAP***) |** ;

***TGah editor: add the following attibutes on Table G-1***

**Table G-1—Attributes applicable to frame exchange sequence definition**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| … | … |
| *non-S1GAP* | Frame is transmitted by a non-AP S1G STA. |
| *S1GAP* | Frame is transmitted by an S1G AP. |