IEEE P802.11
Wireless LANs

|  |
| --- |
| Resolution of TDD BF Related CIDs |
| Date: 2018-1-6 |
| Author(s): |
| Name | Affiliation | Address | Phone | Email |
| Oren Kedem | Intel |  |  | oren.kedem@intel.com |
| Carlos Cordeiro  | Intel  |  |  | carlos.cordeiro@intel.com  |
| Cheng Chen | Intel |  |  | cheng.chen@intel.com |
|  |  |  |  |  |

Abstract

This submission proposes resolutions to 3009, 3011, 3110, 3112, 3338, 3412, 3413, 3415, 3416, 3418, 3420, 3422, 3423, 3424, 3425, 3426, 3433, 3458, 3561, 3632, 3669, 3712, 3715, 3736 CIDs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3009 | 10.26.5.7.1 | Grammar issue | Change "lie" to "are" | Accepted  |

**10.26.6.7 Originator’s behavior**

**10.26.6.7.1 General**

*Change text at P200 L35 as follow*

The originator may transmit QoS Data frames with a TID matching an established block ack agreement in any order provided that their sequence numbers are within the current transmission window.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3011 | 10.26.5.7.2 | grammar using "in case" text should be change to use "If", | Change "In case" to "If"Change "supported, the" to "supported, then the" | Accepted  |

**10.26.6.7 Originator’s behavior**

**10.26.6.7.1 General**

*Change text at P200 L35 as follow*

If recipient memory multiple buffer units capability is not supported, then the parameters *maxMpduInMem* and *mpduSplitInBuffer* are assigned with values 255 and 1, respectively.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3110 | 29.2.2 | Table 43: SCRAMBLER\_INIT\_SETTING, Channel\_BW or CONTROL\_TRAILER can also be used on a single 2.16 GHz channel. It is unclear how NON\_EDMG\_DUP\_C\_MODE captures that, since it specifically says DUP | Replace "In case of NON\_EDMG\_DUP\_C\_MODE" with "In case of NON\_EDMG\_DUP\_C\_MODE or NON\_EDMG\_C\_MODE" | Rejected |
| 3112 | 29.2.2 | Table 43: CH\_BANDWIDTH\_SIGNALING, setting can also be used on a single 2.16 GHz channel. It is unclear how NON\_EDMG\_DUP\_C\_MODE captures that, since it specifically says DUP | Replace "NON\_EDMG\_DUP\_C\_MODE" with "NON\_EDMG\_DUP\_C\_MODE or NON\_EDMG\_C\_MODE" | Rejected  |

**Discussion**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Condition**  | **Value** | **TXVECTOR** | **RXVECTOR** |
| SCRAMBLER\_INIT\_SETTING | FORMAT is EDMG  | Indicates that the PPDU is an EDMG control mode PPDU carrying the EDMG-Header-AEnumerated type:EDMG-Header-A | Y  | Y |
| FORMAT is NON\_EDMG  | Indicates the configuration of the ScramblerInitialization field of a control mode PPDU.Enumerated type:In case of NON\_EDMG\_DUP\_C\_MODE :Channel\_BW or CONTROL\_TRAILEROtherwise: Scrambler | Y  | Y |

There is no such primitive NON\_EDMG\_C\_MODE, relevant primitive is C\_MODE when FORMAT is NON\_EDMG.

Proposed resolution is not accurate since in case of C\_MODE, the option cannot be CONTROL\_TRAILER.

11ay D2.1 definition for the case of C\_MODE, the scrambler will be the scrambler seed (case “Otherwise”) which is the desired behaviour.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3338 | 8.3.4.3 | There is no need to add the antenna list to the IDLE STATE parameter. When the STATE is IDLE it is assumed that all channels are IDLE, It would be best to continue this assumption and also assume all antennas are also IDLE. If any antenna or channel is busy it should be listed as part of the BUSY value. | Change the value field in Table 8-3 for the STATE Parameter to read:(BUSY, [channel-list], [antenna-list])(IDLE) | Accepted  |

*Change text at P62 L6 as follow*

|  |  |  |
| --- | --- | --- |
| **Parameter**  | **Associated primitive**  | **Value**  |
| DATA  | PHY-DATA.request PHY-DATA.indication  | Octet value X'00'–X'FF'  |
| TXVECTOR  | PHY-TXSTART.request  | A set of parameters  |
| STATE  | PHY-CCA.indication  | (BUSY, [channel-list], [antenna-list]) (IDLE)  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3412  | 10.24.2.13 | for 2.16+2.16 GHz, the second 2.16GHz can be any of the secondary, secondar1, secondar2 channels if it is idle | change to 'secondary, secondary1, or secondary2 was idle' | Revised |

*Change text at P188 L21 as follow*

e) Transmit a 2.16+2.16 GHz PPDU if the secondary, secondary1 or secondary2 channels were idle during an interval of PIFS immediately preceding the start of the TXOP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3413 | 10.26.2 | ... subelement for ...in the ADDBA Request...', but subelements are not allowed in ADDBA Request frame based on p189 L40 | remove the sentence | Rejected  |

**Discussion**

The wording “fields within the Recipient Memory Configuration subelement” relate to the ADDBA Response indicated in the last sentence and doesn’t imply that those are sent in ADDBA Request.

***Text under discussion***

The TID grouping capability is supported in a successfully established block ack agreements if both the originator and recipient set the TID Grouping Capable and the RBUFCAP Quantity Capable subfields to 1 in their respective EDMG Flow Control Extension Configuration elements transmitted in the ADDBA Request and ADDBA Response frames. The Recipient Memory Capabilities field and fields within the Recipient Memory Configuration subelement for TIDs that were set to 1 in TID Grouping subfield shall be identical in the ADDBA Response frames.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3415 | 10.26.4 | Multi-TID BA should also be allowed to be used for acking A-PPDU which carries MPDUs from different TIDs | changes AMPDU to 'AMPDU or A-PPDU' in L15 and L18 | Rejected  |

**Discussion**

Section 10.15 states the following:

“A PPDU within an A-PPDU shall contain an A-MPDU. All MPDUs within A-MPDUs within an A-PPDU shall have the same values for the TA and RA fields. All QoS Data frames within A-MPDUs within an APPDU shall have the same ack policy. If a frame that requires an immediate response is present within an A-PPDU, it shall be transmitted in the last A-MPDU of the A-PPDU

The required behaviour requested by this CID is already allowed in the current draft.

Since A-PPDU is constructed from one/multiple A-MPDUs, the immediate response for A-PPDU can be Back or Multi-TID Back in case supported by the responder.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3416 | 10.26.5.1 | The actual RBUFCAP value is also dlivered by the BA frames | add 'EDMG compressed Block Ack, EDMG multi-TID Block Ack, or' after 'delivered by' | Revised |

*Change text at P193 L33 as follow*

* If the block ack agreement is between a pair of EDMG STAs, the memory occupied by the frames shall not exceed the values computed in Table 38 and in Table 39. The actual RBUFCAP value is delivered by EDMG compressed Block Ack, EDMG multi-TID Block Ack, EDMG Flow Control Extension Configuration element in the ADDBA Response frame or the RBUFCAP update of other TIDs as indicated in TID Grouping field of the Recipient Memory Configuration sublement, whichever comes later. If the ADDBA Response frame does not contain an EDMG Flow Control Extension Configuration element, the relevant originator parameters shall be considered as receiving an RBUFCAP with value Receiver Buffer Empty (9.3.1.8.7).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3418 | 10.26.5.6.3 | Block Ack starting sequence control with MSDU and MPDU is not defined in 9.6.4.2 or in a BAR section, but used in 10.26.5.6.3 | define it | Revised |

**Discussion**

Block Ack Starting Sequence Control subfield within BAR represents only the MSDU sequence number, originator may advance its starting sequence number with complete whole MSDU.

*Edit section as follow*

**10.26 Block acknowledgement (block ack)**

**10.26.1 Introduction**

*Edit section as follow*

Under a block ack agreement using segmentation and reassembly, operations on MSDU Sequence Number and MPDU Sequence Number are performed modulo MSDU\_Modulo and modulo MPDU\_Modulo respectively (see 10.72), where MSDU\_Modulo and MPDU\_Modulo correspond to the values of the MSDU Modulo and MPDU Modulo subfields, respectively, defined in the SAR Configuration element. Operations on the MPDU sequence number and on the MSDU sequence number are performed modulo 2 MPDU\_Modulo and 2MSDU\_Modulo, respectively. Comparisons between MPDU sequence number and MSDU sequence number are circular modulo 2MPDU\_Modulo and 2MSDU\_Modulo, respectively, i.e., the sequence number space is considered divided into two parts, one of which is “old” and one of which is “new,” by means of a boundary created by adding half the sequence number range to the current start of receive window (modulo 2MPDU\_Modulo and 2MSDU\_Modulo, respectively). The Block Ack Starting Sequence Control subfield within the BAR represents the MSDU sequence number. If the block ack agreement does not use segmentation and reassembly, all operations on sequence numbers are performed modulo 212. Comparisons between sequence numbers are circular modulo 212, i.e., the sequence number space is considered divided into two parts, one of which is “old” and one of which is “new,” by means of a boundary created by adding half the sequence number range to the current start of receive window (modulo 212).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3420 | 10.26.5.6.1 | 64 should be 1024? | change to 1024 | Accepted  |

*Change text at P197 L15 as follow*

* *WinEndB* is initialized to *WinStartB* + *WinSizeB* – 1, where *WinSizeB* is set to the smaller of 1024 and the value of the Buffer Size field of the ADDBA Response frame that established the block ack agreement.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3422 | 10.26.5.7.1 | If an originator wants to send a multi-TID AMPDU, which contains TIDs from 2 groups, at start of an TXOP, but 1 TID has No Memory Kept=1 and another TID has No Memory Kept=0, what is the final byte count limit for this multi-TID AMPDU? E,g. 1 TID has (N/A, N/A, rx buffer empty, 0) and another TID has (1, N/A, N/A, 1) based on the columns of table 38 | clarify FlowControlByteCountLimit for multi-TID case | Revised |
| 3423 | 10.26.5.7.1 | should 'frame' be a PSDU? | change to PSDU | Accept |
| 3424 | 10.26.5.7.2 | Algorithm in Fig 115 is not used to compute FlowControlByteCountLimit, which is already calculated by table 38. The algorithm is to compute the number of MPDUs fit in this byte count limit based on receipient memory configuration | change to 'algorithm used to compute the number of MPDUs satisfying a given FlowControlByteCountLimit' | Revised |

**Discussion**

All Flow control calculations relate to byte count on specific TID. In Multi-TID A-MPDU, the initiator should verify that for each TID, *FlowControlByteCountLimit* is not exceeding its limit

*Change text at P201 L32 as follow*

At the start of a data transfer sequence that has an established Block Ack agreement, an originator that is an EDMG STA shall not transmit a PSDU containing data with a size greater than *FlowControlByteCountLimit*, where *FlowControlByteCountLimit* is defined per the configuration obtained during the Block Ack agreement establishment for the respective TID or group of TIDs described in Table 38 and per the computation described in 10.26.6.7.2.

*Change text at P202 L4 as follow*

During a data transfer sequence that has an established Block Ack agreement, an originator that is an EDMG STA shall not transmit a PSDU containing data with a size greater than *FlowControlByteCountLimit*, where *FlowControlByteCountLimit* is defined per the configuration obtained during the Block Ack agreement establishment for the respective TID or group of TIDs described in Table 39 and per the computation described in 10.26.6.7.2.

*Change text at P202 L11 as follow*

**10.26.6.7.2 Number of MPDUs per FlowControlByteCountLimit computation**

Figure 118 specifies the algorithm used to compute the number of MPDUs satisfying a given FlowControlByteCountLimit based on Table 38 and Table 39. In this algorithm, the following apply:

* *numOfMpdusForTx* indicates the number of pending MPDUs in the transmit queue that are within the transmission window.
* In case recipient memory multiple buffer units capability is not supported, the parameters *maxMpduInMem* and *mpduSplitInBuffer* are assigned with values 255 and 1, respectively.
* Parameters *unitBufferSize*, *rbufcap*, *memoryUnitSize*, *maxMpduInMem*, and *mpduSplitInBuffer* correspond to the last EDMG flow control parameters as received from a TID within a group of TIDs and with the respective memory configuration tag.
* *mpduForTx[k]* contains the size of the MPDU at location *k* in the transmit queue with the padding for minimum A-MPDU spacing and A-MPDU delimiter alignment, if required.
* For a multi-TID A-MPDU of TIDs not included in TID group, the algorithm computes the number of MPDUs for each TID using the FlowControlByteCountLimit of that TID. For a multi-TID A-MPDU of TIDs included in TID group, the algorithm computes the number of MPDUs for all the TIDs using the recent calculated FlowControlByteCountLimit for any TID within the TID group.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3425 | 10.26.5.7.1 | Fig 115 does not describe how to do it for multi-TID AMPDU with multiple memory configs | clarify Fig 115 for multi-TID case | Rejected  |

**Discussion**

All Flow control calculations relate to byte count on specific TID. In Multi-TID A-MPDU, the initiator should verify that for each TID, *FlowControlByteCountLimit* is not exceeding its limit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3426 | 10.26.2 | Does the two memory config tag mean there can be two configurations for each TID/TID group, or there are two configuration for the whole system of receipient (i.e. 2 groups of TIDs each using a memory config corresponding to a tag) | please clarify the design with a NOTE | Revised  |

**Discussion**

Two Memory Config Tag allow two configurations for the specific ADDBA agreement hence it is per supported TID or Group TID.

*Change text at P135 L12 as follow*

The Two Memory Config Tag Capable subfield is set to 1 to indicate capability to support two Memory Configuration Tag values per TID or TID group. (Figure 78) and is set to 0 otherwise.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3433 | 10.40.1 | the secondary channel can also be the secondary1 channel | change to 'secobdary or secondary1 channels' | Accepted  |

**Discussion**

A-BFT is indicated relative to Primary channels hence might be on each of the secondary channel(s).

*Change text at P210 L23 as follow*

* The A-BFT shall be present on the primary channel of the BSS and can also be present on the secondary or secondary1 channel of the BSS (see 10.43.5.1).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3458 | 29.2.2 | What is the RXVECTOR representing the BW in a received CT?If not defined, CH\_BANDWIDTH\_SIGNALING should also indicates the BW in CT, because of the way it is used in p181 L35 | add BW field in CT in the definition of CH\_BANDWIDTH\_SIGNALING | Revised  |

**Discussion**

CH\_BANDWIDTH\_SIGNALING is represented in 4 bits of the scrambler seed field hence it cannot represent the whole BW field which is 8 bits. Rule in p181 L35 indicates that if EDMG frame is sent in response to non-EDMG duplicate frame that incorporate the CH\_BANDWIDTH\_SIGNALING but doesn’t incorporate the BW field. The response frame BW field should indicate the channels were represented in CH\_BANDWIDTH\_SIGNALING of the frame that elicit the response.

The channel bandwidth information is always presented in the RTS, DMG CTS and DTS frames there the information is critical to establish TXOP. The bandwidth information is not presented in the BAR and BA frames sent in the NON\_EDMG\_DUP there the information is less critical for the interacting devices that the TXOP is already established.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Condition**  | **Value** | **TXVECTOR** | **RXVECTOR** |
| CH\_BANDWIDTH\_SIGNALING | FORMAT is NON\_EDMG,NON\_EDMG\_MODULATION isNON\_EDMG\_DUP\_C\_MODE | Indicates the value of the Scrambler Initializationfield, as defined in Table 47, of the PPDU transmittedin NON\_EDMG\_DUP\_C\_MODE. | O  | Y |

*Change section as follow*

**10.6.7.6 Channel Width selection for Control frames transmitted by EDMG STAs**

The rules in this subclause, combined with the rules in 10.6.7.2, determine the format of control frames that are not RTS, DMG CTS, DMG DTS or CF-End frames.

Channel width selection rules for DMG CTS and DMG DTS frames are specified in 10.3.2.9.

Channel width selection rules for RTS and CF-End frames are specified in 10.3.2.18.

An EDMG STA that sends a Control frame in response to a frame carried in an EDMG PPDU shall set the TXVECTOR parameter CH\_BANDWIDTH to the value indicated by the RXVECTOR parameter CH\_BANDWIDTH of the frame eliciting the response.

An EDMG STA that sends a Control frame in response to a frame carried in a non-EDMG duplicate PPDU shall set the TXVECTOR parameter CH\_BANDWIDTH as follows:

* If the frame that elicited the response includes the channel bandwidth information in the RXVECTOR parameter CH\_BANDWIDTH\_SIGNALING or BW\_IN\_CT, CH\_BANDWIDTH shall be set to a value that represents the equivalent channels indicated by the CH\_BANDWIDTH\_SIGNALING or BW\_IN\_CT parameter;
* Otherwise if the STA received at least one EDMG PPDU as part of the current frame exchange sequence, CH\_BANDWIDTH shall be set to the value of the RXVECTOR parameter CH\_BANDWIDTH of the last received EDMG PPDU in the current frame exchange sequence.
* Otherwise if the STA transmitted at least one EDMG PPDU or non-EDMG duplicate PPDU as part of the current frame exchange sequence, CH\_BANDWIDTH shall be set to the value of the TXVECTOR parameter CH\_BANDWIDTH of the last EDMG PPDU or non-EDMG duplicate PPDU, whichever came later, transmitted by the STA in the current frame exchange sequence.
* Otherwise, CH\_BANDWIDTH shall be set to the estimated value of the RXVECTOR parameter CH\_BANDWIDTH of the frame that elicited the response.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3561 | 11.5.2.4 | Table 42 is not accurate: there is no single indication fro EDMG Flow control as indicated in "ADDBA Condition" column. In second option mandatory flow control is maintained | change ADDBA condition to include real capability indications. Change third column to include mandatory EDMG Flow Control and Optional EDMG Flow Control features. Add EDMG Multi-TID BlockAck as an option with and without flow control | Accepted  |

*Change Table 42 at P321 as follow*

**Table 42 — Types of block ack agreement based on capabilities and ADDBA conditions for EDMG STAs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Capabilitiesconditions** | **ADDBA condition**  | **Type of BlockAckReq andBlockAck variant** | **Type of block ackagreement** |
| One of the STAs is anon-EDMG STA | Per Table 11-5  | Per Table 11-5  | Per Table 11-5 |
| Both STAs are EDMGSTAs | At least one STA indication of EDMG FlowControl support is 0  | Compressed BlockAckReq andEDMG Compressed BlockAck | HT-Immediate + Mandatory EDMG Flow Control  |
| Both STAs are EDMGSTAs | Both STAs indication of EDMG FlowControl support is set to 1 | Compressed BlockAckReq andEDMG Compressed BlockAck | HT-Immediate + Mandatory and Optional EDMG Flow Control |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3632 | 6.3.119 | There is no reference to the normative subclause 11.36.4 TDD beam measurement that uses the primitives | Add reference to 11.36.4 in the commented line | Revised |

*Change P52 L17as follow*

On receipt of this primitive, the MLME invokes the MAC sublayer TDD beam measurement procedures defined in 10.43.11 and 11.36.4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3669 | 9.3.1.8.7 | Where is the "EDMG Compressed BlockAck frame" defined? It took me quite some time to work out that this is a BlockAck frame with the EDMG Compressed variant. I think the terminology in the text is very confusing, as it's not possible to find the definition of a "EDMG Compressed BlockAck frame". | Change "EDMG Compressed BlockAck frame" to "BlockAck frame EDMG Compressed variant". Incidentally, I realise that this would result in similar changes to Draft P802.11REVmd\_D1.2.pdf. Talking of which, there's an error in Draft P802.11REVmd\_D1.2.pdf, Page 802, Line 60 (Clause 9.3.1.8.4), as the text should refer to "Extended Compressed BlockAck frame", or as I would prefer it "BlockAck frame Extended Compressed variant". | Accepted  |

*Change P70 L2 as follow*

The TID\_INFO subfield of the BA Control field of the BlockAck frame EDMG Compressed variant contains the TID for which a BlockAck frame is requested.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3712 | 3.1 | Segmentation and reassembly are similar to fragmentation and defragmentation. Why do we need two similar mechanisms? | Forbid fragmentation and only allow segmentation between EDMG STAs. Make the More Fragments subfield reserved for EDMG STAs. | Rejected  |

**Discussion**

Segmentation and fragmentation are two different mechanisms. While fragmentation enables the over the air partitioning of MPDU to smaller fragments, segmentation enables the over the air partitioning of large host datagrams into MPDUs done solely by the MAC layer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3715 | 9.3.1.8.1 | Will the free memory space be 100% not kept? Won't there be any probability that the situation may change? | Change to "... in the last RBUFCAP sufbfield is not guaranteed to be kept at the start of the next frame exchange sequence; ...". | Revised |

*Change P69 L8 as follow*

The No Memory Kept subfield set to one indicates that the free memory space indicated in the last RBUFCAP subfield may not kept at the start of the next frame exchange sequence; otherwise if set to zero, free memory space indicated by the RBUFCAP subfield is kept by the receiver for the next frame exchange sequence for the of the corresponding TID(s).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CID** | **Clause** | **Comment** | **Proposed change** | **Resolution**  |
| 3736 | 8.3.5.12.2 | secondary2 in the channel-list parameter elements has been used in IEEE 802.11ah-2016.Please use the different terminology. | As in comment. |  |

*Change P69 L9 as follow*

**SP/M:** Do you accept the resolutions given in this document?