IEEE P802.11  
Wireless LANs

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Some MAC/PHY Layering Issues | | | | | | Date: 2021-10-28 | | | | | | Author(s): | | | | | | Name | Affiliation | Address | Phone | email | | Brian Hart | Cisco Systems |  |  | [brianh@cisco.com](mailto:brianh@cisco.com) | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |

Abstract

This submission proposes resolutions for the following comments from comment collection on P802.11be D1.0:

4627, 4628, and 4629

The baseline used in this document is D1.2.

NOTE – Set the Track Changes Viewing Option in the MS Word to “All Markup” to clearly see the proposed text edits.

**Revision History:**

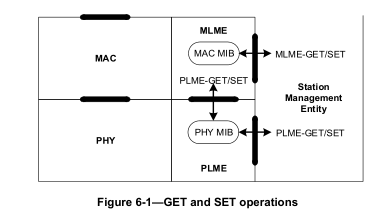
R0: Initial version.

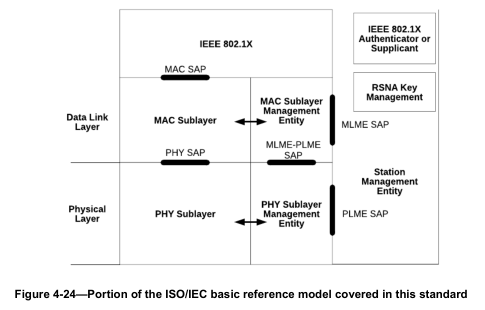
R1: Reduced MAC changes, added OMI parameters.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4627 | 36.3.3.1.1 | 371.28 | This para has the 802.11 arch back to front, and leads to circular logic. What should happen: Step 1) PHY declares its capabilities via a MIB variable. Step 2) MLME reads the PHY's capabilities. Step 3) MLME may opt to prune PHY capabilities according to policy; Step 4: the MLE/MAC advertises this (pruned) list as this STA's PHY capabilities to peer STAs. What is happening here: the MAC is magically discovers what the PHY is capable of, and then magically lets the PHY know. | 1) If not already present in Table 36-68, define a MIB variable so the PHY can express if the PHY is capable of this particular feature or not. This is required. 2) If we really think that the MLME may want the PHY to disable this particular feature(!?), then give the MAC a MIB variable to use to control the PHY to disable/enable this particular feature. Or not. Add language connecting the dots. | Revised. See changes in 21/xxxxR<motionedRevision> that substantially implement the commenter’s proposal. |
| 4628 | 36.3.3.2.2 | 372.41 | This para has the 802.11 arch back to front, and leads to circular logic. What should happen: Step 1) PHY declares its capabilities via a MIB variable. Step 2) MLME reads the PHY's capabilities. Step 3) MLME may opt to prune PHY capabilities according to policy; Step 4: the MLE/MAC advertises this (pruned) list as this STA's PHY capabilities to peer STAs. What is happening here: the MAC is magically discovers what the PHY is capable of, and then magically lets the PHY know. | 1)If not already present in table 36-68, define a MIB variable so the PHY can express if the PHY is capable of this particular feature or not. This is required. 2) If we really think that the MLME may want the PHY to disable this particular feature(!?), then give the MAC a MIB variable to use to control the PHY to disable/enable this particular feature. Or not. Repeatedly apply 1) and 2) to each instance of "Capabilties" in this section until we don't see "Capabilties" anywhere in this section (currently 9x). Add language connecting the dots. | Revised. See changes in 21/xxxxR<motionedRevision> that substantially implement the commenter’s proposal. |
| 4629 | 36.3.3.2.4 | 373.04 | This para has the 802.11 arch back to front, and leads to circular logic. What should happen: Step 1) PHY declares its capabilities via a MIB variable. Step 2) MLME reads the PHY's capabilities. Step 3) MLME may opt to prune PHY capabilities according to policy; Step 4: the MLE/MAC advertises this (pruned) list as this STA's PHY capabilities to peer STAs. What is happening here: the MAC is magically discovers what the PHY is capable of, and then magically lets the PHY know. | 1) If not already present in Table 36-68, define a MIB variable so the PHY can express if the PHY is capable of this particular feature or not. This is required. 2) If we really think that the MLME may want the PHY to disable this particular feature(!?), then give the MAC a MIB variable to use to control the PHY to disable/enable this particular feature. Or not. Add language connecting the dots. | Revised. See changes in 21/xxxxR<motionedRevision> that substantially implement the commenter’s proposal. |

***Discussion***

The 802.11 architecture is well captured in the following two figures.





The benefits of the ISO Seven Layer Model is:

* One layer (or sublayer) can be swapped out for another layer (e.g., 802.11 for 802.3)
* Layering conforms to good software development practice
  + Each layer/sublayer can be thought of an object, with public/private constants, variables, and methods
  + Each method has a clear **and complete** interface
  + No method needs to “silently reach into another object’s private variables” to complete its task
* A “divide and conquer” approach is possible:
  + **During standardization, relatively independent teams can work in parallel on their own layer/sublayer**
  + During product development, relatively independent teams can work in parallel on their own layer/sublayer
  + These teams might even reside in different companies so that the complete product is composed of components from different companies

For instance, when this design approach is applied to capability advertisement, it operates as follows:

* The PLME declares its capabilities via PHY MIB variables
  + i.e., the PHY is the source of truth for the PHY
* The MLME reads the PHY capabilities from the PHY MIB via the PLME-GET primitive
* The MLME applies any policy (e.g., perhaps does not advertise its maximum bandwidth or spatial stream capability to save power)
* The MLME populates the PHY-related fields of the EHT Capabilities element with the capabilities learnt from the PHY MIB and after MAC policy is applied.
* The MLME uses the MAC and PHY data plane to transmit its EHT Capabilities element in an MMPDU in an MPDU in a PSDU in a PPDU

Corollaries:

* Main takeaway: the PLME/PHY is (presently) unaware of the PHY-related fields of the EHT Capabilities element.
  + The PHY data plane does transmit the EHT Capabilities element, but only understands it as an opaque PSDU.
* For the PLME to be aware of the PHY-related fields of the EHT Capabilities element, the MLME must notify the PLME of the values of these parameters explicitly
  + Via a standardized MAC/PHY interface.

The main standardized information flows between MAC and PHY are:

* PLME-CHARACTERISTICS.request/confirm so that, upon request, PLME can send its constants to MLME
* Read only parameters in the PHY MIB so the PHY can report its capabilities
* **PHY-CONFIG.request(PHYCONFIG\_VECTOR) so the MAC can dynamically configure the PHY**
* Read/write parameters in the PHY MIB (to allow an external mgmt. entity to manage the STA; or *possibly* the MAC)
  + But better is to use PHY-CONFIG.request(PHYCONFIG\_VECTOR)
  + To complete the story we should add an extra read-only MIB variable to alert the external mgmt. system that the PHY is operating at lower than its maximum capability
    - but since the ASN.1 MIB is rarely implemented, that external mgmt. system has much bigger problems, so this would be pointless standards make-work.
* PHY-TXSTART(TXVECTOR) and PHY-RXSTART(RXVECTOR) for per PPDU parameters
  + Also TRIG\_VECTOR, but this seems to be missing a primitive (needs a new REVme comment).

For instance, if the MLME determines to send a OMN frame to a peer STA modify its own **operating bandwidth**, then the MLME also needs to notify its collocated PHY using the PHY-CONFIG.request(PHYCONFIG\_VECTOR) primitive.

For instance, if the MLME determines to send a OMN frame to a peer STA modify its own number of **spatial streams**, then the MLME also needs to notify its collocated PHY. But … there is no parameter for this in the PHY-CONFIG.request(PHYCONFIG\_VECTOR) primitive nor a writable MIB parameter. So … today, either a) fail (needs a new REVme comment) or b) apparently the PHY continues to operate in a mode where it can TX/RX the maximum number of spatial streams declared in its MIB (!?).

***Text changes***

***TGbe editor, please make the following changes under CID 4627 as shown by Word track changes***

35.13.2a EHT PHY Capabilities Information Contents

The EHT MAC determines the capabilities of its EHT PHY by using the PLME-GET primitive to read the EHT PHY MIB attributes (see Table 36-68—EHT PHY MIB attributes). The subfields of the EHT PHY Capabilities Information field in the EHT Capabilities element shall not signal greater capability than indicated by the EHT PHY MIB.

36.2.4 PHY CONFIG\_VECTOR

The PHYCONFIG\_VECTOR carried in a PHY-CONFIG.request primitive for an EHT PHY contains an OPERATING\_CHANNEL parameter, which identifies the operating or primary channel. The PHY shall set dot11CurrentPrimaryChannel to the value of this parameter.

The PHYCONFIG\_VECTOR carried in a PHY-CONFIG.request primitive for an EHT PHY contains a CHANNEL\_WIDTH parameter, which identifies the operating channel width and takes one of the values 20 MHz, 40 MHz, 80 MHz, 160 MHz, and 320 MHz. The PHY shall set dot11CurrentChannelWidth to the value of this parameter. The PHY shall set dot11EHTCurrentChannelWidthSet to a value that is obtained from the Supported Channel Width Set subfield of a transmitted EHT Capabilities element (see 9.4.2.295c (EHT Capabilities element))(#1540).

The PHYCONFIG\_VECTOR carried in a PHY-CONFIG.request primitive for an EHT PHY contains a CENTER\_FREQUENCY\_SEGMENT parameter, which identifies the center frequency of the channel and takes a value between 1 and 255. The PHY shall set dot11CurrentChannelCenterFrequencyIndex0 to the value of this parameter.

The PHYCONFIG\_VECTOR carried in a PHY-CONFIG.request primitive for an EHT PHY further contains an EHT\_PHY\_CAPABILITIES\_INFORMATION parameter and a SUPPORTED\_EHT\_MCS\_AND\_NSS\_SET parameter. The EHT\_PHY\_CAPABILITIES\_INFORMATION and SUPPORTED\_EHT\_MCS\_AND\_NSS\_SET parameters contain the same fields as the EHT PHY Capabilities Information and Supported EHT-MCS And NSS Set fields defined in the EHT Capabilities element.

36.3.3.1.1 Supported RU/MRU sizes in DL MU-MIMO(#2699)

When the Partial Bandwidth DL MU-MIMO subfield of the EHT\_PHY\_CAPABILITIES\_INFORMATION parameter in the latest PHYCONFIG\_VECTOR parameter equals 1, the STA shall support receiving an RU/MRU in an EHT PPDU where MU-MIMO is employed in the RU/MRU, the RU/MRU size being greater than or equal to 242 tones, and where there are multiple RUs/MRUs within the PPDU bandwidth.

36.3.3.2.2 Supported RU sizes in UL MU-MIMO

When the Partial Bandwidth UL MU-MIMO subfield of the EHT\_PHY\_CAPABILITIES\_INFORMATION parameter in the latest PHYCONFIG\_VECTOR parameter equals 1, the STA shall support transmitting an RU/MRU in an EHT TB PPDU where UL MU-MIMO is employed in the RU/MRU, the RU/MRU size being greater than or equal to 242 tones, and where there are multiple RUs/MRUs within the PPDU bandwidth.

36.3.3.2.4 Maximum number of spatial streams in UL MU-MIMO

The maximum number of spatial streams supported by a STA for SU transmissions is indicated in the SUPPORTED\_EHT\_MCS\_AND\_NSS\_SET in the latest PHYCONFIG\_VECTOR parameter.