IEEE P802.11
Wireless LANs

|  |
| --- |
| MIB Attributes Design Pattern Background |
| Date: 2014-08-25 |
| Author(s): |
| Name | Affiliation | Address | Phone | email |
| Mark Hamilton | Spectralink, Corp | 2560 55th StBoulder, CO 80301 USA | +1-303-441-7553 | mark.hamilton@spectralink.com |
|  |  |  |  |  |

Abstract

This document contains background material for the ARC SC discussion, in response to the WG Chair’s request, to look into MIB attribute usage in REVmc (and active TGs, perhaps).

*From Adrian Stephens:*

Please excuse the long analysis here.  I think the discussion in TGai highlights flaws in REVmc. *[An outcome of this discussion is captured at the bottom of this document.]*

I’m going to task ARC to do the analysis and create a design pattern, which we can use in REVmc to highlight and fix flaws in the use of …Implemented and …Activated MIB variables.

The characteristics we generally have in REVmc are:

“named” features only have a dot11xxxxImplemented MIB variable.

The presence of the feature is represented by a capability field or capability element.

The presence of this field is static.

Some features that are not named are represented by both “Implemented” and “Activated”.

My intuition tells me these should have the following properties:

1.       Do not relate to a “named” feature

2.       If the peer needs to understand the state of activity of this feature to use it (e.g. with Rx LDPC),  then the “activated” needs to be signalled on-the-air.

a.       If it is signalled in a “capability” field,  then the MIB variable “changes take effect” will be limited to when a STA next joins or starts a BSS.

OK,  so how does REVmc stack up against this.    Let’s take LDPC as an example.

1.       It is a feature represented by both “implemented” and “activated”.

dot11LDPCCodingOptionImplemented OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This is a capability variable.

Its value is determined by device capabilities.

This attribute, when true, indicates that the LDPC coding option is implemented."

DEFVAL { false }

::= { dot11PhyHTEntry 13 }

dot11LDPCCodingOptionActivated OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This is a control variable.

It is written by an external management entity.

Changes take effect as soon as practical in the implementation.

This attribute, when true, indicates that the LDPC coding option is

enabled."

DEFVAL { false }

::= { dot11PhyHTEntry 14 }

2.       The peer does need to know this state.  The transmit logic is as follows:

9.16 LDPC operation

An HT STA shall not transmit a frame with the TXVECTOR parameter FORMAT set to HT\_MF or HT\_GF and the TXVECTOR parameter FEC\_CODING set to LDPC\_CODING unless the RA of the frame corresponds to a STA for which the LDPC Coding Capability subfield of the HT Capabilities element received from that STA contained a value of 1 and **dot11LDPCCodingOptionActivated** is true.

And at 871.12:



This doesn’t match my intuition in a couple of ways:

1.       There is no limitation on when the “activated” variable can change, no on-the-air signalling to tell a peer when this has changed.

2.       The description of the on-the-air signalling leaves it completely ambiguous as to whether it reflects the “implemented” or “activated”.   I suppose that those implementers who understand how it should work will signal the “activated” variable, and those who don’t think too deeply will signal the “implemented” variable.

This was the first dynamic feature I checked.   I did check others and found apparent flaws in them all.

So, at this point, I think we (i.e., ARC) need to create a “design pattern” (see Wikipedia) for how these variables are used, and make it a job for REVmc to adapt existing dynamic features to this design pattern (during sponsor ballot).

*From discussion on the TGai reflector:*

Based on the discussion and straw polls taken today during the TGai slot PM2, the group settled on the **following basis for defining a FILS STA**:

            dot11FILSImpleted equals TRUE

and

            dot11FILSActivated equals TRUE

Note, that this modifies the current definition which only defines a FIlS STA as a "STA that supports FILS" STA, i.e. dot11FILSImplemented equals TRUE" without specifying the state of dot11FILSActivated.