Discussion with 802.1 Regarding 802.3at/802.3az use of LLDP

July 2008 Denver Plenary

Wael William Diab – Broadcom

Agenda

- ## Code-point Locations[1]

  - – Associated MIB work

- ## Use of State Machines over LLDP[2]

  - – Review of 802.3 Understanding and Approach with Using of LLDP

  - – Feedback from 802.1

- ## Review of State Machine and Protocol Approaches Discussed in 802.3at[2]

1 applies to 802.3at and 802.3az

2 applies to 802.3at

802.3 Code-points (Subtypes)

Table F-1—IEEE 802.3 Organizationally Specific TLVs

IEEE 802.3 subtype	TLV name	Subclause reference
1	MAC/PHY Configuration/Status	F.2
2	Power Via Medium Dependent Interface (MDI)	F.3
3	Link Aggregation (deprecated)	F.4
4	Maximum Frame Size	F.5
5–255	reserved	—

Table F-1, IEEE P802.1AB-REV/D3.0
(Previously Table G-1, IEEE P802.1AB-2005)

Code-point Location: Options

- We looked at 4 options

  1. Everything in Dot1, tied to AB-REV

  2. Move everything into 802.3

  3. New OUI for 802.3

  4. 802.1 assigns a block of subtypes under the existing OUI to 802.3 to establish an RA within 802.3

     – Subtype assignment to projects at the appropriate time (e.g. Sponsor Ballot)

     – Managed by 802.3 (802.3 Chair or his designated appointee)

Code-point Location: Option 1

- Everything in Dot1, tied to AB-REV

- Advantages
  - Monolithic, same "spot" as before
  - Extension to existing MIB
  - No LoA issues (like .1AX/.3ax)

- Disadvantages
  - Timeline
  - SNMP based MIB only (not 802.3 "Generic" style)
  - Future Maint involves 2 docs/2 WGs/2 PARs

Code-point Location: Option 2

- Move everything into 802.3 (with, perhaps, 802.1 holding back a block for themselves)
- Advantages
  - Monolithic, same "spot" as before in .3.
  - Control our own destiny (i.e. control issuance of our own sub-types)
  - 802.3 "Generic" style
  - Single PAR/Doc/WG for Maint work
- Disadvantages
  - LoA issues (just like .3/.1ax)
  - Timeline/scope (.AB-REV PAR/.3at PAR)
  - Work to convert existing SNMP MIB to .3 Generic style.
  - Would leave LARGE deprecated chunk in the middle of 802.1 MIB

Code-point Location: Option 3

- New OUI for 802.3

- Advantages
  - Monolithic, same "spot" as before.
  - Control our own destiny (i.e. control issuance of our own sub-types)
  - 802.3 "Generic" style
  - Single PAR/Doc/WG for Maint work
  - No LoA issues (like .1AX/.3ax)

- Disadvantages
  - .3at has to do new clause for 802.3
  - How would 802.1 feel about it?
  - Creates 2 address points for what should be the same problem/objective

Code-point Location: Option 4

- 802.1 assigns a block of subtypes under the existing OUI to 802.3 to establish an RA within .3

- Advantages
  - Control our own destiny (i.e. control issuance of our own sub-types)
  - 802.3 "Generic" style
  - Single PAR/Doc/WG for Maint work
  - No LoA issues (like .1AX/.3ax)

- Disadvantages
  - Split MIB and/or MIB extension

Code-point Location: Discussion

- Based on analysis, recommend Option 4

- Other related discussion

  - Update on sub-type assignment for .3at

    - Subtype removed from draft till SA Ballot

  - Inquire regarding what AVB and DCB are doing w.r.t the use of LLDP

    - Are there any similar issues

  - Constraint on "keep alives" in low power mode

    - Can we set a large TTL without sending any frames for a prolonged period of time less than TTL

Code-point Discussion: Summary

- Consensus on Option 2

  – Steps identified on next slide

- Other related discussion

  – Constraint on "keep alives" in low power mode

    - Can we set a large TTL without sending any frames for a prolonged period of time less than TTL
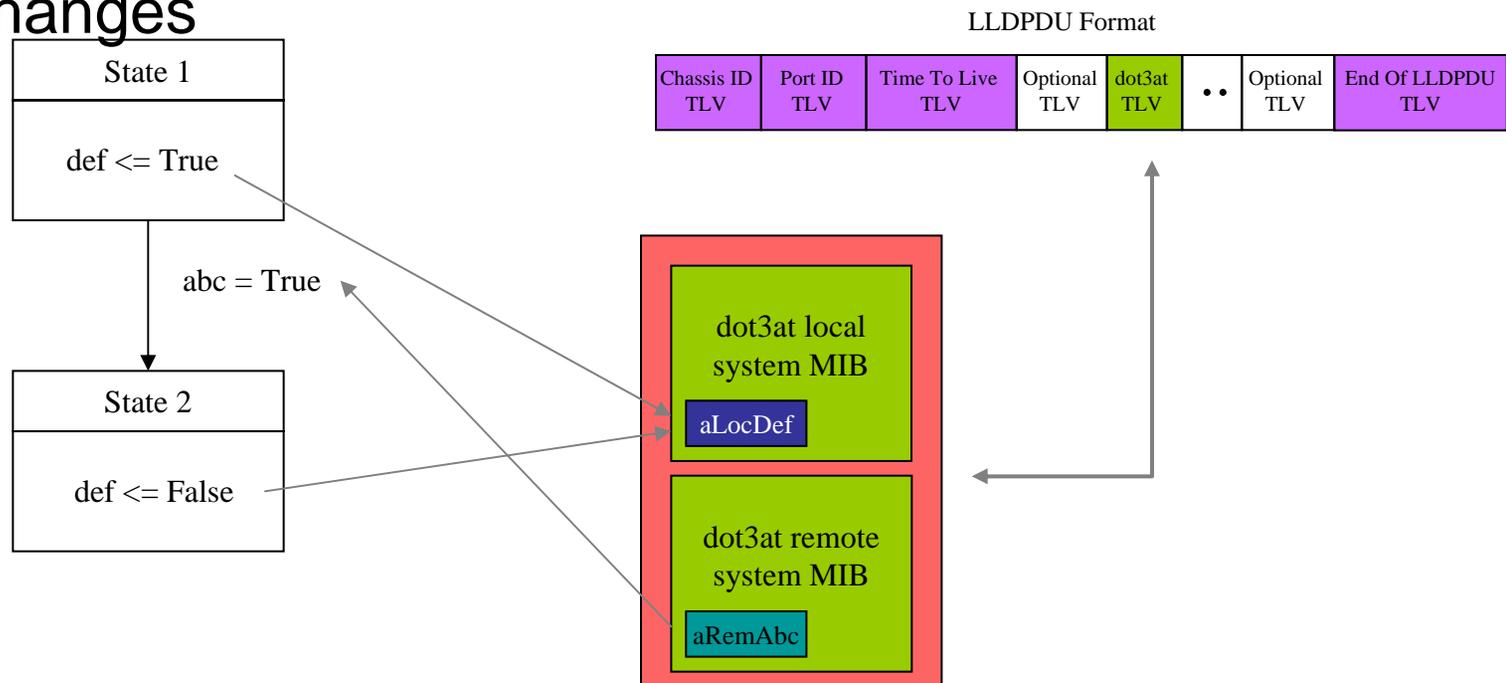
    - Answer: YES

Steps to Move Annex F to .3

- Scope modification of 802.3at

- Make Patcom aware and explain to PatCom why the situation differs from .3ax/.1AX

- Steps in 802.3
  - Discuss scope modification and work in .3at
  - Discuss maintaining once in .3 in maintenance

- Consider steps in 802.1AB-REV

LLDP and State diagrams

- Can't map directly to TLV contents
  - Map through objects in dot3at local and remote MIB
  - Define MIB attribute to variable mapping
  - Allows .3 layers to take action based on variable changes

LLDPDU Format

Chassis ID TLV	Port ID TLV	Time To Live TLV	Optional TLV	dot3at TLV	..	Optional TLV	End Of LLDPDU TLV

State 1

def <= True

abc = True

State 2

def <= False

dot3at local system MIB

aLocDef

dot3at remote system MIB

aRemAbc

Use of State Machine with LLDP

- 802.3at's Understanding of LLDP
  - LLDP is an advertise *only* mechanism
  - Idea is whatever is in one MIB will be reflected to a copy (mirrored) in a MIB on the other side of the link
  - Was not *originally* intended for a request-response protocol

- Request for Feedback from 802.1
  - Any concerns with building a State Machine on top of LLDP?
  - If so, what are the concerns?
  - If not, any restrictions?

Review of Protocol Approaches Discussed in 802.3at

- Protocol to budget power for PoE

- 802.3at considered two approaches

  - Near identical functionality

  - Initial approach had an implicit ACK/NACK that was sent within the TLVs

  - Revised approach reverted to advertising changes in the parameters and simplified diagrams

- Does 802.1 care about what approach 802.3at uses for their protocol and State Machine?

Guidance for use of a State Machine/Protocol over LLDP

- No fundamental problem to do State Machine
- Preferably don't do ACK/NACKs, if you do, you need serial numbers
  - Look at DCB proposal as an example of serial numbers. Has not been examined in .1 yet
- Don't make it too chatty
  - LLDP may be running other protocols
  - Minimize the number of frames transmitted
- 802.1 expertise may be available to help
- Opportunity for 802.1 members to ballot in WG on 802.3at
  - Request based system
  - Same for 802.3az