Cl **35**   *SC* **35.2.6**   *P* **65**   *L* **47**   # 20

Zhihong Yu (Don Pannell)   Marvell

Comment Type   **GR**   *Comment Status*   **X**

The procedure described in D3.2 will not work, due to following reasons:
1) rMt! Is an event which is meaningless to MRP Registrar and will be ignored;
2) rLv! Will change Registrar state to LV and consequently will be changed to IN by origianl rJoinIn! or rJoinMt!, but this transition process will not lead to Lv action which issues MAD_Leave.indication and will not lead to Join action which issues MAD_Join.indication;
3) The peer MRP Applicant may send JoinIn and JoinMt for various reasons, such as by its own state machine, by rLv! or rLA!, or by periodic! or redecalre!;
4) For exactly same attribute, the procedure stated in 35.2.6 must be avoided. Otherwise state machine flip-flop or unnecessary network transaction will occur;

SuggestedRemedy

The proposed modification to the fist paragraph of section 35.2.6 is as follows:

If an MSRP message is received from a Port with an event value (35.2.6) specifying the JoinIn or JoinMt message, and if the StreamID (35.2.2.8.2), and Direction (35.2.1.2) all match those of an attribute already registered on that Port, and if either AttributeType (35.2.2.4), or MSPR FourPackedEvents (35.2.2.7.2) is different with that message, and if there are any other such discrepancies determined by specific implementation, then an Flush! event is generated for the MAD in the Received MSRP Attribute Declarations before the rJoinIn! or rJoinMt! event for attribute in the received message is processed.

Proposed Response   *Response Status*   **W**

DISCUSS.

Cl **00**   *SC* **0**   *P*   *L*   # 21

Zhihong Yu (Don Pannell)   Marvell

Comment Type   **G**   *Comment Status*   **X**

5) There are additional prerequisites needed to determine whether or not to apply the procedure stated in 35.2.6. There must be some differences exist between current registered value and the incoming message. The AttributeType (for Talker messages) and FourPackedEvent (for Listener messages) must be considered. Other differences in Talker message such as TSpec should also be considered;
6) Instead of rLv! or rMt!, I suggest to use Flush! as the artificial event before deliver rJoinIn! or rJoinMt! to registrar. Flush! will change the state to MT and will lead to Join event when processing rJoin! or rJoinMt!;
7) Flush! processing in MRP need to be fixed too. Please see next comment.

SuggestedRemedy

Proposed Response   *Response Status*   **W**

DISCUSS

Cl **00**   *SC* **0**   *P*   *L*   # 22

Zhihong Yu (Don Pannell)   Marvell

Comment Type   **G**   *Comment Status*   **X**

In 802.1ak-2007 on page 45, in section 10.7.8 "Registrar state machine", Flush! Processing need further discussion.

If received Flush! in IN state, there is no Lv action taken. This can be okay since this transition is not due to rLv!. But considering the Join action when receiving next rJoinIn! in MT state, it could be okay too to add Lv action for Flush!.

For state machine completeness, it could be bettwe for Flush! to stop leavetimer.

SuggestedRemedy

The proposed Flush! State machine is as below:

```
_____|_IN_|_LV_|_MT_
Flush!|_*1_|_*1_|_MT_
```

Due to space limitation, the *1 is as below:

```
 Stop leavetimer
 Lv
 MT
```

Proposed Response   *Response Status*   **W**

DISCUSS. This is really an MRP issue, but we have addressed other MRP issues (e.g. Table 10-5) within MSRP. This is definitely outside the scope of MSRP but could be addressed if agreed to by the WG.

TYPE: TR/technical required ER/editorial required GR/general required T/technical E/editorial G/general
COMMENT STATUS: D/dispatched A/accepted R/rejected   RESPONSE STATUS: O/open W/written C/closed U/unsatisfied Z/withdrawn
SORT ORDER: Comment ID

Page 1 of 2

Comment ID # **22**   9/23/2009 10:10:02 AM

Cl **35**   *SC* **35.2.2.8.5**   *P* **55**   *L* **8**   # 29
John Nels Fuller   None entered

Comment Type   **TR**   *Comment Status* **D**

The only valid values for Data Frame Priority are those associated with an SR class. Thus this value spcifies the SR class of the stream.
Note that if the ingress port remaps priorities into the SR priorities they must also be remapped here (before checking for the valid values).

SuggestedRemedy

Add note to this effect

Proposed Response   *Response Status* **W**

DISCUSS. What could be said here? SRP reserves by Priority Code Point (0-7) not SR Class (A,B).

We need to discuss your comment about remapping on ingress. Currently SR Class A & B are defaulted to 5 & 4 with no override capabilities defined at this time (FYI: during Interim MJT noted that we need to default to 3 & 2). Or, perhaps you are referring to the concept of the bridge that is joining two SRP domains that use different Traffic Classes for the same SR class?

Cl **35**   *SC* **35.2.2.8.6**   *P* **55**   *L* **33**   # 31
John Nels Fuller   None entered

Comment Type   **T**   *Comment Status* **X**

portTcMaxLatency seems to be a number that could be put on a data sheet. In reality a better (smaller) number can be calculated knowing the amount of bandwidth reserved for the described stream, both at the talker and at each bridge hop. Refer to av-fuller-queue-delay-calculation-0809-v02.pdf for a start at these calculations.

SuggestedRemedy

Discuss and make appropriate changes.

Proposed Response   *Response Status* **W**

DISCUSS. Don Pannell suggested that we put the final formula from Qav Annex L here, then refer to Annex L (which is informative) for those who want to understand its derivation.

Cl **35**   *SC* **35.2.4.2**   *P* **61**   *L* **29**   # 33
John Nels Fuller   None entered

Comment Type   **TR**   *Comment Status* **D**

supported protocols (e.g. MACSec) also affect perFrameOverhead.

SuggestedRemedy

Add text to this effect.

Proposed Response   *Response Status* **W**

PROPOSED REJECT. perFrameOverhead is defined in Qav. In an effort to keep the entire definition in one location, any additional text should be added there (34.4).

Cl **35**   *SC* **35.2.4.2**   *P* **61**   *L* **31-3**   # 60
Craig Gunther   Harman International

Comment Type   **TR**   *Comment Status* **D**

Remove Editor's Note

SuggestedRemedy

Discuss. Do we want to add a reservation fudge factor or not?

Proposed Response   *Response Status* **W**

PROPOSED ACCEPT IN PRINCIPLE. During the Interim we had decided to simple remove this note about adding a slight amount of overhead to the reservations. While discussing Qav we brought the subject back up and decided we need to do something to handle a bridge that has a reservation for the _exact_ amount of bandwidth in the media packets provided by the Talker, but the Talker class measurement interval is running slightly faster than the bridges interval. Eventually the Talker will feed an extra packet to the bridge and the bridge will drop it. The solution appears to be that the Talker should always ask for a bit more than it needs. HOW MUCH IS A BIT MORE?

TYPE: TR/technical required ER/editorial required GR/general required T/technical E/editorial G/general
COMMENT STATUS: D/dispatched A/accepted R/rejected   RESPONSE STATUS: O/open   W/written   C/closed   U/unsatisfied Z/withdrawn
SORT ORDER: Comment ID

Page 2 of 2

Comment ID # **60**

9/23/2009 10:10:02 AM