MSRP Attribute Declaration, Propagation and Bandwidth Allocation

Craig Gunther (cgunther@harman.com)
24 September 2008 (updated 04 November 2008)
Topics

• MSRP Attribute Propagation
• Triggers for Bandwidth Reallocation
• Bandwidth Allocation Algorithm
• Listener Attribute Propagation
• Qat/Qav - What runs where?
• MSRPDU Talker Attributes
MSRP Attribute Propagation
MSRP Attribute Propagation

SR Station - Talker

Host Application

MSRP Participant

MSRP (MRP Application)

MAD (MRP Attribute Declaration)

MAD_Join.request(TalkerAdvertise)

MSRPDU

PHYSICAL MEDIA (CAT-5e)

SR Station - Listener

Host Application

MSRP Participant

MSRP (MRP Application)

MAD (MRP Attribute Declaration)

MAD_Join.indication(TalkerAdvertise)

MSRPDU

PHYSICAL MEDIA (CAT-5e)
Bridge MSRP Propagation
MSRP Service Primitives

**Bandwidth Scenario 1:**
This Stream's bandwidth is given to another Stream by the network.

**Bandwidth Scenario 2:**
Talker runs out of bandwidth on its network link for this Stream.

**Bandwidth Scenario 3:**
Listener runs out of bandwidth on its network link for this Stream.

- **REGISTER_STREAM.request** (StreamID)
- **REGISTER_STREAM.STATUS.indication** (StreamID, Advertise)
- **REGISTER_ATTACH.request** (StreamID)
- **REGISTER_ATTACH.STATUS.indication** (StreamID, Ready)
- **REGISTER_ATTACH.indication** (StreamID, Ready/Failed)
- **REGISTER_ATTACH.indication** (StreamID, Ready/AskingFailed)
- **REGISTER_STREAM.indication** (StreamID, Failed)
- **REGISTER_STREAM_STATUS.indication** (StreamID, Failed)

**Physical Media (CAT-5e)**
Talker’s Second Stream Fails
Bridge Forwarding a Stream
Triggers for Bandwidth Reallocation
Bandwidth Reallocation

Actions that might cause a bandwidth allocation change

- TalkerAdvertise
- TalkerFailed
- ListenerReady
- ListenerReadyFailed
- ListenerAskingFailed
- Media bandwidth changed
Bandwidth Reallocation - 2

WARNINGS:

In normal operation you would expect to see Talker declarations registered before Listener declarations, although this is not a requirement.

You might also see unexpected interaction in MSRP attribute processing for a short period of time after a Spanning Tree reconfiguration.
Bandwidth Allocation Algorithm
Definitions & Notes

• Bandwidth reservation is done on the TX side
• Definition: Active Listener = Listener Ready or Listener Ready Failed (bandwidth will be consumed if this stream is Approved)
• On Non-shared media: Talker Advertise from another port is associated with Active Listener on the current port
• On Shared media: the Talker Advertise may also come from the current port (e.g. 802.11)
Definitions & Notes - 2

- Merging Listener Declarations:

<table>
<thead>
<tr>
<th>First Declaration Type</th>
<th>Second Declaration Type</th>
<th>Resultant Declaration Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ready</td>
<td>Ready</td>
<td>Ready</td>
</tr>
<tr>
<td></td>
<td>Ready Failed or Asking Failed</td>
<td>Ready Failed</td>
</tr>
<tr>
<td>Ready Failed</td>
<td>Any</td>
<td>Ready Failed</td>
</tr>
<tr>
<td>Asking Failed</td>
<td>Ready or Ready Failed</td>
<td>Ready Failed</td>
</tr>
<tr>
<td></td>
<td>Asking Failed</td>
<td>Asking Failed</td>
</tr>
</tbody>
</table>
Per-Port Bandwidth Analysis

- Do the following for each TX port (completing the analysis for each port before proceeding to the next port):
  - Create a list of Talker declarations from any other port that could be sent out this port
  - This list is divided into two sections:
    - Streams with Active Listeners on this port are at the top of the list
    - Streams without Active Listeners on this port are at the bottom of the list
  - Sort each section by Talker Rank + StreamID
Per-Port Bandwidth Analysis - 2

• Include the following information with each Talker declaration in the list:
  – Stream ID
  – Talker Attribute (Advertise or Failed)
  – Talker Traffic Class (A or B)
  – TSpec
  – Talker is on same port “flag” (True or False)
    • Bandwidth requirements double if True
  – Stream MAC DA
  – Stream VID
  – Approved/Disapproved “flag” (set by Qav)
Qav Bandwidth Analysis

• Ask Qav to do a Trial Reservation based on information in the list
  – Qav will set Approved/Disapproved flag
  – Qav may adjust Talker attribute

• Qav processes the list as follows:
  – Work from top of list to bottom
  – Calculate bandwidth requirements based on:
    • TSpec + additional media overhead
    • Double bandwidth requirement if Talker is also on this port
Qav Bandwidth Analysis - 2

• Qav continues processing as follows:
  – If bandwidth is not available for this stream with this traffic class, or attribute declaration is Talker Failed:
    • Flag list entry as Disapproved
    • Set attribute in list to Talker Failed on this port
    • Move to next entry in the list
Qav Bandwidth Analysis - 3

• Qav continues processing as follows (There is sufficient bandwidth for this stream):
  – If there are Active Listeners on this port:
    • Flag list entry as Approved
    • Reduce available bandwidth for remaining streams
    • Move to next entry in the list
  – If there are no Active Listeners on this port:
    • Flag list entry as Approved
    • Do not reduce available bandwidth (no stream data will flow)
    • Move to next entry in the list
Bandwidth Allocation

• Qav has finished processing this port’s list
• Compare Approved/Disapproved flag for every stream against current reservations, as specified on following pages
Bandwidth Allocation - 2

• Process list from top to bottom looking for Disapproved streams:
  – If stream is currently Approved:
    • Set port to filter Stream MAC DA
    • Remove bandwidth allocation from Qav
    • Deregister Talker Advertise from this port
  – Register a Talker Failed on this port
  – If there is a Listener attribute registered on this port declare a pending Listener Asking Failed on the port the Talker is registered on (this may require merging other Listener attributes)
  – Move to next entry in the list
Bandwidth Allocation - 3

• Reprocess list from top to bottom looking for Approved streams:
  – Deregister Talker Failed if it was declared for this port
  – Register a Talker Advertise for this port
  – If stream is currently not Approved allocate bandwidth via Qav
  – Set port to forward Stream MAC DA
  – Declare a pending Listener attribute on the port the Talker is registered on that is identical to this port’s Listener attribute (this may require merging other Listener attributes)
  – Move to next entry in the list
Bandwidth Allocation - 4

• If there are more ports on the bridge:
  – Build a new list for the next port by returning to the “Per-Port Bandwidth Analysis” slide #14

• If all ports have been processed:
  – Proceed to the next slide and process the “pending Listener attributes”
Listener Attribute Propagation
Listener Attribute Propagation

• Bandwidth Allocation for all ports in the bridge has been completed

• Process each port’s pending Listener attributes
  – If pending Listener attribute does not match current Listener attribute
    • Deregister current Listener attribute
    • Register pending Listener attribute
Qat/Qav
What runs where?
What runs in the Talker?

• 802.1Qav for:
  – Bandwidth query and reservation
  – Queuing and shaping traffic

• 802.1Qat for:
  – TalkerAdvertise request and ListenerReady indication processing
  – Bandwidth related Advertise/Failed changes
What runs in the Bridge?

• 802.1Qav for:
  – Bandwidth query and reservation
  – Queuing and shaping traffic

• 802.1Qat for:
  – Bandwidth related Advertise/Ready/Failed changes
  – Talker & Listener attribute propagation

• 802.1ak (MMRP) for:
  – Optional Talker pruning
What runs in the Listener?

• **802.1Qat** for:
  - TalkerAdvertise indication and ListenerReady request processing

• **802.1ak (MMRP)** for:
  - Optional Talker pruning
MSRPDU Talker Attributes
MMRPDU Message

AttributeType = 2 // MACVectorAttributeType = 2

Attribute Length = 6

8 bits // Length of FirstValue field = 6

LeaveAllEvent 3 bits // NullLeaveAllEvent = 0
// LeaveAll = 1

VectorHeader

NumberOfValues 13 bits // Number of events encoded in the vector

FirstValue // MACVector = 48 bits for MAC Address

AttributeEvent

// New = 0
// JoinIn = 1
// In = 2
// JoinMT = 3
// Mt = 4
// Lv = 5

ThreePackedEvents

8-bits of ((AttributeEvent*8)+AttributeEvent)*8+AttributeEvent

EndMark 16 bits // 0x0000

24 September 2008
MSRPDU TalkerFailed Message

- **AttributeType**: 2 bits
  - **TalkerFailed**: 2

- **Attribute Length**: 36 bits
- **AttributeListLength**: 41 bits
- **Length of AttributeList data**: 16 bits

- **LeaveAllEvent**: 3 bits
  - **LeaveAll = 1**

- **VectorHeader**
  - **NumberOfValues**: 13 bits

- **Talker Failed (36 octets)**
  - **AttributeType**: 2

- **FirstValue**

- **VectorAttribute**

- **AttributeList**

- **NullLeaveAllEvent**: 0

- **StreamID**: 8 bits
  - **Unique ID**: 16 bits

- **Data Frame Parameters**: 8 bits
  - **Stream MAC**: 48 bits
    - **VLAN ID**: 16 bits
    - **Bandwidth (in 1000 byte/second granularity)**: 32 bits
    - **TSpec**: 6 bits
      - **Frame Rate (in 1000 frames/second granularity)**: 16 bits
    - **Data Frame Frontend (Class A=5, Class B=4)**: 3 bits
    - **Rank Stream Importance**: 5 bits (only define emergency plus high & low priority for now)
    - **Accumulated Latency**: 4 bits
    - **Bridge ID**: 13 bits
      - **Change from Advertise to Failed**: 64 bits
      - **Failure Information**: 9 bits
        - **Failure code returned by LAC Service (35.x)**: 8 bits

- **ThreePackedEvents**

- **EndMark**: 16 bits
  - **0x0000**
MSRPDU (all) Listener Message

**MSRPDU**

- **AttributeType** = 3
- **Attribute Length** = 8
- **AttributeListLength** = 14

**VectorHeader**
- **LeaveAllEvent** = 3 bits
  - **NullLeaveAllEvent** = 0
  - **LeaveAll** = 1

**VectorAttribute**
- **Listener Ready**
- **Listener Ready Failed**
- **ListenerAskingFailed** (8 octets)
  - **AttributeType** = 3

**VectorThreePacked**
- **ThreePackedEvents**
- **...**

**VectorFourPacked**
- **FourPackedEvents**
- **...**

**EndMark** = 16 bits (0x0000)
Questions?