MSRP Attribute Declaration, Propagation and Bandwidth Allocation

Craig Gunther (cgunther@harman.com)
24 September 2008 (updated 06 November 2008)
Topics

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

SR Station - Talker

Host Application

MSRP Participant

MSRP (MRP Application)

MAD (MRP Attribute Declaration)

REGISTER_STREAM.request(StreamID)

MAD Join.request(TalkerAdvertise)

MSRPDU

PHYSICAL MEDIA

SR Station - Listener

Host Application

MSRP Participant

MSRP (MRP Application)

MAD (MRP Attribute Declaration)

REGISTER_STREAM.indication(StreamID)

MAD Join.indication(TalkerAdvertise)

MSRPDU
Bridge MSRP Propagation

![Diagram of Bridge MSRP Propagation](image-url)
MSRP Service Primitives

**Talker advertises the Stream**

- REGISTER_STREAM.request (StreamID, Advertise)

**Listener #1 successfully attaches**

- REGISTER_ATTACHMENT.indication (StreamID, Ready)

**Listener #2 successfully attaches**

- REGISTER_ATTACHMENT.indication (StreamID, Ready)

**Listener #3 unsuccessfully attaches**

- REGISTER_ATTACHMENT.indication (StreamID, ReadyFailed)

**Bandwidth becomes unavailable**

- REGISTER_ATTACHMENT.indication (StreamID, ReadyFailed or AskingFailed)

Note: Talker sees no MSRP packets related to Listener #2 since Listener #1 has already sent a Listener Ready.

Note: Listener #2 has no MSRP activity related to Listener #2.

Note: Listener #1 sees no MSRP packets related to Listener #2 since Listener #1 never sends each others MSRDPUs. The only exception to this is if a REGISTER_ATTACHMENT.request is sent before the corresponding REGISTER STREAM.request is sent.

See Note: above.

This Stream's bandwidth is given to another Stream.

**Talker MSRPG**

**Listener #1 MSRPG**

**Physical Media**
Talker’s Second Stream Fails

1. REGISTER_STREAM.request (StreamID)
   Status update (Advertise)

2. REGISTER_STREAM.request (StreamID)
   Status update (Advertise)

3. MAD_JOIN.request (TalkerAdvertise)
   MSRPDU (TalkerAdvertise join)

4. MAD_JOIN.indication (ListenerReady)
   MAD_LEAVE.request (TalkerFailed)
   MAD_JOIN.request (TalkerFailed)

SR Station Host Application

Time

MSRP (including bandwidth verification) + MAP

MAD

24 September 2008

IEEE 802.1 Interim - Seoul
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
MRPDU

802.3 MAC header
- MAC DA = 01-80-C2-00-00-2X
- MAC SA = XX-XX-XX-XX-XX-XX
- EtherType = XX-XX

ProtocolVersion
- 8 bits // defined by the specific MRP application

AttributeType
- 8 bits // non-zero integer defined by the specific MRP application

AttributeLength
- 8 bits // Non-zero integer defined by the specific MRP application (length of FirstValue field)

LeaveAllEvent
- 3 bits // LeaveAll = 1

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

FirstValue
- 7 bits // defined by the specific MRP application

VectorHeader

AttributeList

Vector

ThreePacketEvents

ThreePacketEvents

AttributeEvent
- 8-bits of ((AttributeEvent*6)+AttributeEvent)*6+AttributeEvent
- New = 6
- JoinIn = 1
- M = 3
- JoinMT = 3
- Mt = 4
- Lv = 5

EndMark
- 16 bits // 0x0000

EndMark
- 16 bits // 0x0000
MMRPDU Message

- **AttributeName**: 8 bits // MACVectorAttributeType = 2
- **Attribute Length**: 6
- **NumberOfValues**: 13 bits // Number of events encoded in the vector
- **FirstValue**: // MACVector = 48 bits for MAC Address
- **VectorHeader**
- **LeaveAllEvent**: 3 bits // LeaveAll = 1
- **NullLeaveAllEvent**: 0
- **AttributeEvent**
  - // New = 0
  - // JoinIn = 1
  - // In = 2
  - // JoinMT = 3
  - // Mt = 4
  - // Lv = 5

- **ThreePackedEvents**
- **ThreePacketEvents**
- **EndMark**: 16 bits // 0x0000
MSRPDU TalkerFailed Message

MSRPDU

AttributeType: 2
Attribute Length: 36
AttributeListLength: 41
8 bits // TalkerFailed=2
8 bits // Length of FirstValue
16 bits // Length of AttributeList data

VectorAttribute

Message

AttributeList

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

VectorHeader

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

// StreamID (8) // Talker MAC SA Address 48 bits
// Unique ID 16 bits

// Data Frame Parameters (8) // Stream MAC DA 48 bits
// VLAN ID 16 bits

// Bandwidth (in 1000 byte/second granularity) 32 bits
// TSpec (8) // Frame Rate (in 1600 frames/second granularity) 16 bits

// Data Frame Priority (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) 32 bits
// Bridge ID (13.y) of bridge that changed declaration from Advertise to Failed
// 64 bits

// Failure Information (9) // Failure code returned by LAC Service (35.x) 8 bits

AttributeEvent // New = 0
// JoinIn = 1
// In = 2
// JoinMT = 3
// Ml = 4
// LV = 5

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

ThreePackedEvents

ThreePackedEvents

EndMark 16 bits // Ox0000

Talker Failed (36 octets) // AttributeType = 2

FirstValue

802.1 Interim - Seoul

October 2008
MSRPDU (all) Listener Message

MSRPDU

AttributeList

Attribute

VectorAttribute

EndMark

16 bits // 0x0000

EightPackedEvents

SecondDeclarationType 2-bits

FirstDeclarationType 2-bits

ThirdDeclarationType 2-bits

FourDeclarationType 2-bits

FourPackedEvents

... FourPackedEvents

ThreePackedEvents

... ThreePackedEvents

EightPackedEvents

((AttributeEvent*6) + AttributeEvent)*5 + AttributeEvent

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

ListenerReady 3 bits // NullLeaveAllEvent = 0

LeaveAllEvent 3 bits // LeaveAll = 1

LeaveAllEvent

VectorHeader

LeaveAllEvent

FirstValue

Listener AskingFailed (8 octets) // AttributeType = 3

Listener Ready Failed

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

AttributeListLength = 14

AttributeLength = 8

AttributeType = 3

8 bits // Listener = 3

3615

October 2008 IEEE 802.1 Interim - Seoul
Questions?
Acronyms Used in Presentation

- MRP: Multiple Registration Protocol (802.1ak)
- MSRP: Multiple Stream Reservation Protocol (802.1Qat)
- MSRPDU: MSRP Protocol Data Unit (802.1Qat)
- MAD: MRP Attribute Declaration (802.1ak)
- MAP: MRP Attribute Propagation (802.1ak)
- Qat: Stream Reservation Protocol (802.1Qat)
- Qav: Forwarding and Queuing (802.1Qav)