IEEE P802.1Qat/D6.0

Draft Standard for Local and Metropolitan Area Networks—

Virtual Bridged Local Area Networks - Amendment XX: Stream Reservation Protocol (SRP)

Sponsor
LAN/MAN Standards Committee of the IEEE Computer Society

Prepared by the Audio/Video Bridging Task Group of IEEE 802.1

Abstract: This amendment specifies protocols, procedures and managed objects, usable by existing higher layer mechanisms, that allow network resources to be reserved for specific traffic streams traversing a bridged local area network.
Keywords: LANs, local area networks, MAC Bridges, Bridged Local Area Networks, virtual LANs, Virtual Bridged Local Area Networks, Audio/Video Bridging, resource reservation, Multiple Registration Protocol (MRP).

DRAFT STATUS:

Draft generated for 1st Sponsor ballot recirculation.

Changes from D5.0 to D6.0 are: New boundary detection technique via a new SRP attribute, Listeners now use MVRP to add themselves to the VLAN member set for propagation of priority values, updated 802.11 behavior in Annex Q.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

> Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Introduction to IEEE P802.1Qat™

(This introduction is not part of P802.1Qat, IEEE Standards for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks—Stream Reservation Protocol (SRP).)

<<Editor's Note: Standard boilerplate material goes here, such as patent policy etc...>>

This standard contains state-of-the-art material. The area covered by this standard is undergoing evolution. Revisions are anticipated within the next few years to clarify existing material, to correct possible errors, and to incorporate new related material. Information on the current revision state of this and other IEEE 802 standards may be obtained from

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Participants

The following is a list of participants in the Interworking activities of the IEEE 802.1 Working Group during the development of P802.1Qat. Voting members at the time of publication are marked with an asterisk (*).

When the IEEE 802.1 Working Group approved IEEE Std 802.1Qat, it had the following membership:

Tony Jeffree, *Chair and Editor*
Paul Congdon, *Vice Chair*
Michael Johas Teener, *Chair, AV Bridging Task Group*

<<TBA>>

The following members of the balloting committee voted on P802.1Qat. Balloters may have voted for approval, disapproval, or abstention.

<<TBA>>

When the IEEE-SA Standards Board approved this standard on <<TBA>>, it had the following membership:

???, *Chair* **???,** *Vice Chair*
???, *Secretary*

<<TBA>>

Figures

Tables

IEEE P802.1Qat/D6.0

Draft Standard for Local and Metropolitan Area Networks—

Virtual Bridged Local Area Networks — Amendment XX: Stream Reservation Protocol

Editorial Note

This amendment specifies changes to IEEE Std 802.1Q that allow network resources to be reserved for specific traffic streams traversing a bridged local area network. Changes are applied to the base text of IEEE Std 802.1Q-2005, as modified by those amendments that had been approved, but not incorporated into the base text of the standard, at the time that this amendment was approved, namely (in chronological order) IEEE Std 802-1ad, IEEE Std 802.1ak, IEEE Std 802.1ag, IEEE Std 802.1ah, IEEE Std 802.1Q-2005/Cor 1, IEEE Std 802.1ap, IEEE Std 802.1Qaw, IEEE Std Qay, and IEEE Std 802.1Qav. Text shown in bold italics in this amendment defines the editing instructions necessary to make changes to this base text. Three editing instructions are used: ***change***, ***delete***, and ***insert***. ***Change*** is used to make a change to existing material. The editing instruction specifies the location of the change and describes what is being changed. Changes to existing text may be clarified using ~~strikeout~~ markings to indicate removal of old material, and underscore markings to indicate addition of new material). ***Delete*** removes existing material. ***Insert*** adds new material without changing the existing material. Insertions may require renumbering. If so, renumbering instructions are given in the editing instruction. Editorial notes will not be carried over into future editions of IEEE Std 802.1Q.

In particular, this amendment builds upon the Multiple Registration Protocol (MRP) as described in 802.1ak, which defines the MRP applications: MMRP and MVRP. This amendment defines a third MRP application: MSRP. It is highly recommended that the reader gain a thorough understanding of MRP before proceeding with this amendment. Clause numbers referenced by this amendment can be found in P802.1Qat, P802.1ak, P802.1Qav and 802.1Q-2006 (in that order).

1. Overview

Insert the following after the initial paragraphs of clause 1.

This standard specifies protocols, procedures and managed objects, usable by existing higher layer mechanisms, that allow network resources to be reserved for specific traffic streams traversing a bridged local area network. It characterizes resource requirements of traffic streams to a level sufficient for bridges to determine the required resources and provides a mechanism for dynamic maintenance of those resources.

1.1 Scope

Insert the following text and bullets (renumbered appropriately) immediately after the existing text of this clause:

To enable the end-to-end management of resource reservation for QoS guaranteed streams, this standard further specifies protocols, procedures and managed objects, usable by existing higher layer mechanisms, that allow network resources to be reserved for specific traffic streams traversing a bridged local area network. To this end it:

a) Specifies the use of Dynamic Reservation Entries (8.8.7) in the filtering database to control the forwarding of frames associated with a particular Stream.
b) Specifies a Stream Reservation Protocol (SRP). SRP facilitates the registration, de-registration and maintenance of stream reservation information in relevant bridges to establish end-to-end stream paths.

2. References

The following standards contain provisions that, through reference in this document, constitute provisions of this standard. All the standards listed are normative references. Informative references are given in Annex A. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

Insert the following reference at the appropriate point:

RFC 2205: Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification;

RFC 2750: RSVP Extensions for Policy Control

3. Definitions

Insert the following definitions into Clause 3, in appropriate collating sequence, renumbering existing/new definitions as appropriate:

3.1 Designated MSRP Node (DMN): A single station on a shared medium (e.g., 802.11 or a CSN) that controls MSRP access for all other stations on that shared medium.

3.2 Listener: The end station that is the destination, receiver or consumer of a stream.

3.3 Multiple Stream Registration Protocol (MSRP): A protocol designed to provide Quality of Service for streams in bridged networks by reserving resources within each Bridge along the stream’s paths.

3.4 Stream: A unidirectional flow of data (e.g., audio and/or video) from a Talker to one or more Listeners.

3.5 StreamID: A 64-bit field that uniquely identifies a stream.

3.6 Talker: The end station that is the source or producer of a stream.

3.7 Traffic Specification (TSpec): A specification that characterizes the bandwidth that a stream can consume.

4. Abbreviations

Add the following abbreviations, in the appropriate collating sequence.

DMN	Designated MSRP Node
MSRP	Multiple Stream Registration Protocol
MSRPDU	Multiple Stream Registration Protocol Data Unit
QoS	Quality of Service
SR_PVID	Stream Reservation Port VLAN Identifier

5. Conformance

Insert the following new subclause 5.4.3, after subclause 5.4.2 (Multiple VLAN Registration Protocol (MVRP) requirements):

5.4.3 Multiple Stream Registration Protocol (MSRP) requirements

A VLAN-aware Bridge implementation in conformance to the provisions of this standard for the support of MSRP shall:

a) Conform to the operation of the MRP Applicant, Registrar, and LeaveAll state machines, as defined in 10.7.7, 10.7.8, and 10.7.9, as required for one or more of the following variants of the MRP Participant, as defined in 10.6 and 10.7:
   1) The Full Participant.
   2) The Full Participant, point-to-point subset.
b) Exchange MRPDUs as required by those state machines, formatted in accordance with the generic PDU format described in 10.8, and able to carry application-specific information as defined in 35, using the group MAC addresses reserved for use by MRP applications, as defined in Table 10-1.
c) Implement the MSRP Application component as defined in 35.
d) Propagate registration information in accordance with the operation of MAP for the Base Spanning Tree Context, as specified in 10.3.1.
e) Forward, filter or discard MAC frames carrying any MRP Application address as the destination MAC Address in accordance with the requirements of 8.13.6

NOTE—The Periodic Transmission state machine (10.7.10) is specifically excluded from MSRP. A Bridge is allowed to generate an MSRP LeaveAll event to force an immediate redeclaration of all MSRP attributes from its neighbor(s).

Change clause 5.9 as follows:

5.9 End station requirements for MMRP, MVRP and MSRP~~and MVRP~~

This subclause defines the conformance requirements for end station implementations claiming conformance to MVRP, MMRP and MSRP~~and MMRP~~. Although this standard is principally concerned with defining the requirements for VLAN-aware Bridges, the conformance requirements for end station implementations of MVRP, MMRP and MSRP~~and MMRP~~ are included in order to give guidance to such implementations. The PICS Proforma defined in Annex I is concerned with conformance claims with respect to end station implementations.

For the reasons stated in 10.6, it is recommended that end stations that do not require the ability to perform Source Pruning implement the Applicant Only Participant, in preference to the Simple Applicant Participant.

Insert the following new subclause 5.9.3, after subclause 5.9.2 (MVRP requirements and options):

5.9.3 MSRP requirements and options

An end station for which conformance to MSRP is claimed shall:

a) Conform to the operation of the MRP Applicant, Registrar, and LeaveAll state machines, as defined in 10.7.7, 10.7.8, and 10.7.9, as required for one or more of the following variants of the MRP Participant, as defined in 10.6 and 10.7:
   1) The Full Participant.
   2) The Full Participant, point-to-point subset.
   3) The Applicant-only Participant.
   4) The Simple-Applicant Participant.
b) Exchange MPDUs as required by the MRP state machine(s) implemented, formatted in accordance with the generic PDU format described in 10.8, and able to carry application-specific information as defined in 35.2.2, using the MSRP Application address as defined in Table 10-1.

An end station for which conformance to the operation of the Applicant state machine (10.7.7) is claimed may

c) Perform Talker pruning, as defined in 35.2.1.4b, 35.2.3.1 and 35.2.4.3.1.
d) Perform Listener pruning as described in 35.2.3.1.

NOTE—The Periodic Transmission state machine (10.7.10) is specifically excluded from MSRP. An end station is allowed to generate an MSRP LeaveAll event to force an immediate redeclaration of all MSRP attributes from its neighbor(s).

6. Support of MAC Service in VLANs

This amendment makes no changes to clause 6.

9. Tagged frame format

Insert the row for VID 2 in Table 9-2 as shown below:

Table 9-2—Reserved VID values

VID value (hexadecimal)	Meaning/Use
0	The null VLAN ID. Indicates that the tag header contains only priority information; no VLAN identifier is present in the frame. This VID value shall not be configured as a PVID or a member of a VID Set, or configured in any Filtering Database entry, or used in any Management operation.
1	The default PVID value used for classifying frames on ingress through a Bridge Port. The PVID value of a Port can be changed by management.
2	The default SR_PVID value used for SRP (35.2.1.4(i)) Stream related traffic. The SR_PVID value of a Port can be changed by management.
FFF	Reserved for implementation use. This VID value shall not be configured as a PVID or a member of a VID Set, or transmitted in a tag header. This VID value may be used to indicate a wildcard match for the VID in management operations or Filtering Database entries.

10. Multiple Registration Protocol (MRP) and Multiple MAC Registration Protocol (MMRP)

Add the following paragraph to the end of Clause 10:

Clause 35 defines a third MRP application, the Multiple Stream Reservation Protocol (MSRP), that registers data Stream characteristics and reserves Bridge resources as appropriate to provide QoS guarantees.

10.3 MRP Attribute Propagation (MAP)

Change the first paragraph of Clause 10.3 as follows:

~~The MRP Attribute Propagation (MAP) function operates in the same way for all MRP applications and enables propagation of attributes registered on Bridge Ports across the network to other participants.~~

The MRP Attribute Propagation (MAP) function enables propagation of attributes registered on Bridge Ports across the network to other participants. Each MRP application specifies the operation of the MAP function. This subclause specifies the operation of the MAP function for the MMRP application and the MVRP application (11.2.1). The MAP function for the MSRP application is specified in Clause 35.2.4.

10.5 Requirements for interoperability between MRP Participants

Change Table 10-2 to include MSRP EtherType:

Table 10-2—MRP EtherType values

Assignment	Value
MMRP EtherType	88-F6
MVRP EtherType	88-F5
MSRP EtherType	22-EA

10.7 Protocol specification

10.7.9 LeaveAll state machine

Insert the following note in Table 10-5:

Table 10-5—LeaveAll state table

		STATE	
		Active[1]	**Passive**
EVENT	**Begin!**	Start leavealltimer Passive	Start leavealltimer Passive
	tx!	sLA Passive	-x-
	rLA!	Start leavealltimer Passive	Start leavealltimer Passive
	leavealltimer!	Start leavealltimer Active	Start leavealltimer Active

Notes to the table:
[1]Request opportunity to transmit on entry to the Active state.

10.8 Structure and encoding of MRP Protocol Data Units

10.8.1 Structure

10.8.1.2 Structure definition

Change the BNF productions as follows:

MRPDU ::= ProtocolVersion, Message {, Message}, EndMark

ProtocolVersion BYTE ::= *Defined by the specific MRP application*

Message ::= AttributeType, AttributeLength [, AttributeListLength], AttributeList

AttributeType BYTE ::= *Non-zero integer defined by the specific MRP application*

AttributeLength BYTE ::= *Non-zero integer defined by the specific MRP application*

AttributeListLength SHORT ::= *Non-zero integer defined by the specific MRP application*

AttributeList ::= VectorAttribute {, VectorAttribute}, EndMark

VectorAttribute ::= VectorHeader, FirstValue, Vector

VectorHeader SHORT ::= (LeaveAllEvent * 8192) + NumberOfValues

FirstValue ::= Defined by the specific MRP application

Vector ::= ThreePackedEvents {, ThreePackedEvents}
[, FourPackedEvents {, FourPackedEvents}]

ThreePackedEvents BYTE ::= (((((AttributeEvent) *6) + AttributeEvent) *6) + AttributeEvent)

AttributeEvent BYTE ::= New | JoinIn | In | JoinMt | Mt | Lv

FourPackedEvents BYTE ::= ((FourPackedType *64) + (FourPackedType *16)
+ (FourPackedType *4) + FourPackedType)

FourPackedType BYTE ::= *Integer defined by the specific MRP application*

LeaveAllEvent BYTE ::= NullLeaveAllEvent | LeaveAll

NumberOfValues SHORT ::= Number of events encoded in the vector

EndMark SHORT ::= 0x0000 | End of PDU

NullLeaveAllEvent ::= 0

LeaveAll ::= 1

New ::= 0

JoinIn ::= 1

In ::= 2

JoinMt ::= 3

Mt ::= 4

Lv ::= 5

The parameters carried in MRPDUs, as identified in this structure definition, shall be encoded as specified in 10.8.2.

Figure 10-5 illustrates the structure of the MRPDU and its components.

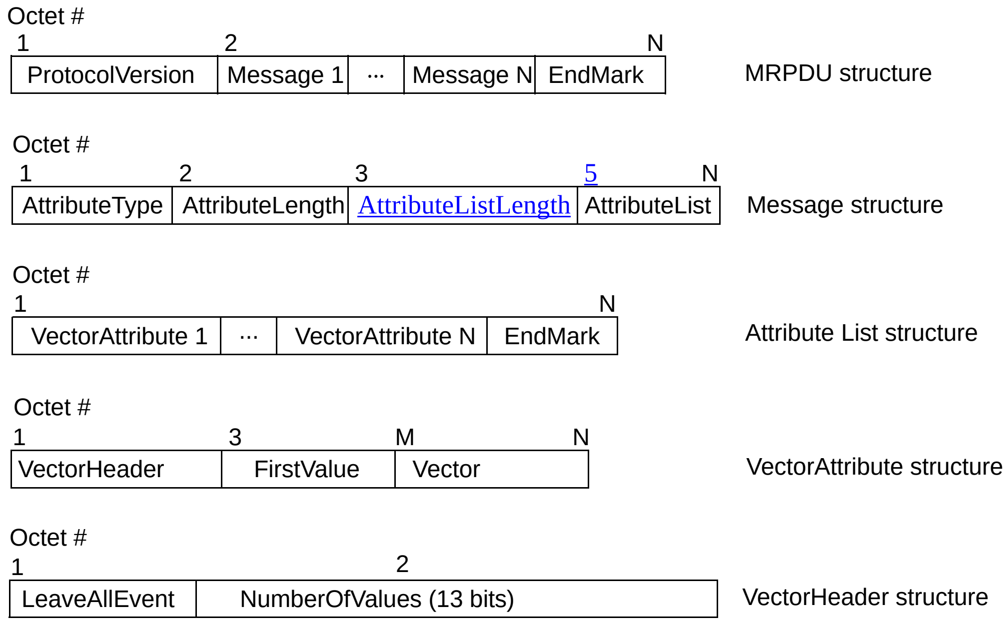

Figure 10-5—Format of the major components of an MRPDU

10.8.2 Encoding of MRPDU parameters

Insert new subclause 10.8.2.4, renumbering and changing subsequent subclauses, as shown:

10.8.2.4 Encoding of AttributeListLength

An AttributeListLength shall be encoded as two octets, taken to represent an unsigned binary number. The AttributeListLength indicates the length, in octets, of the AttributeList field for the Attribute to which the message applies. This field is not present in all MRPDUs. Specifically, MMRPDUs and MVRPDUs do not use this field, whereas MSRPDUs do use this field.

10.8.2.5~~10.8.2.4~~ Encoding of AttributeEvent

An AttributeEvent shall be encoded as an unsigned decimal number in the range 0 through 5. The permitted values and meanings of the AttributeEvent are as follows:

0: New operator
1: JoinIn operator
2: In operator
3: JoinMt operator
4: Mt operator
5: Lv operator

Further values of AttributeEvent are reserved.

The AttributeEvent is interpreted on receipt as a MAD event to be applied to the state machine for the Attribute defined by the AttributeType and AttributeValue to which the AttributeEvent relates.

10.8.2.6~~10.8.2.5~~ Encoding of LeaveAllEvent

A LeaveAllEvent shall be encoded as an unsigned binary number. The permitted values and meanings of LeaveAllEvent are as follows:

0: NullLeaveAllEvent operator
1: LeaveAll operator

Further values of LeaveAllEvent are reserved.

The LeaveAllEvent is interpreted on receipt as a MAD Leave All event to be applied to the state machines for all Attributes of the type defined by the AttributeType field.

The value NullLeaveAllEvent signifies that there is no Leave All event to process, and is included purely for encoding efficiency in the vector attribute structures. Receipt of this value does not cause any event to be applied to any state machine.

10.8.2.7~~10.8.2.6~~ Encoding of FirstValue

A FirstValue is encoded in N octets, taken to be an unsigned binary number, in accordance with the specification for the AttributeType defined by the MRP application concerned. When NumberOfValues is greater than one (1) FirstValue is incremented in a way that is defined by each MRP application. For example MMRP simply increments FirstValue by adding the number one (1) to it, whereas MSRP adds one

(1) to multiple fields within the FirstValue for each increment. Throughout this specification FirstValue incremented by one is denoted as FirstValue + 1, incrementing by two is denoted as FirstValue + 2, etc.

10.8.2.810.8.2.7 Encoding of VectorHeader

The VectorHeader is used to encode both the value of the LeaveAllEvent (10.8.2.4) and the NumberOfValues, the number of AttributeEvent values encoded in a Vector (10.8.2.8). The VectorHeader is taken to be an unsigned binary number, encoded in two octets, as follows:

a) The value of the LeaveAllEvent is multiplied by 8192.
b) The resulting number is added to NumberOfValues.

The range of values that NumberOfValues can take is restricted, such that the following are true:

c) The size of the Vector that is defined by this number will fit in the available space in the PDU.
d) The number of AttributeEvent values, added to the number encoded in FirstValue, Incrementing FirstValue the number of times specified in NumberOfValues, does not exceed the permitted numeric range of FirstValue as defined for the application concerned.
e) The number of AttributeEvent values range from zero (0) to 8191, inclusive. Zero is only valid if LeaveAllEvent is non-zero.

NOTE— If number of AttributeEvents and LeaveAllEvent were both allowed to be zero it would be impossible to differentiate the VectorHeader from an EndMark.

10.8.2.910.8.2.8 Encoding of EndMark

An EndMark shall be encoded as a single octet, taken to represent the unsigned binary number. It takes the numeric value 0.

Further values of EndMark are reserved and shall not be used.

NOTE—As defined by the MRPDU structure definition in 10.8.1, if the end of the MRPDU is encountered, this is taken to be an End Mark from the point of view of processing the PDU contents.

10.8.2.1010.8.2.9 Encoding of Vector

Insert new subclause heading (10.8.2.10.1) as shown:

10.8.2.10.1 Encoding of Vector ThreePackedEvents

The Vector is encoded as onezero or more 8-bit values, each containing a numeric value, ThreePackedEvents, derived from three packed numeric values, each of which represent an AttributeEvent, in the range 0 though 5.

As can be seen from the BNF definition of ThreePackedEvents, each 8-bit value is derived by successively adding an event value and multiplying the result by 6. In order to facilitate the subsequent description, the event values are numbered form *first* to *third*, as follows:

ThreePackedEvents BYTE ::= (((((*first*AttributeEvent) *6) + *second*AttributeEvent)
*6) + *third*AttributeEvent)

The NumberOfValues field in the VectorHeader of the VectorAttribute determines the number of 8-bit ThreePackedEvents values, *E*, that will be present in the vector; hence E is determined by dividing NumberOfValues by 3 and rounding any non-integer answer up to the nearest larger integer.

The FirstValue field of the VectorAttribute determines which of the originator's state machines the *first* AttributeEvent value in the first ThreePackedEvents value relates to. The *second* AttributeEvent value in the first ThreePackedEvents value corresponds to the state machine identified by (FirstValue + 1), and the *third* AttributeEvent value in the first ThreePackedEvents value corresponds to the state machine identified by (FirstValue + 2), and so on, through subsequent packed values.

Where the NumberOfValues field carries a value that is not a multiple of 3, there will be either one or two AttributeEvent values packed in the final ThreePackedEvents that are ignored; these values are encoded as the numeric value 0 on transmission and are ignored on receipt.

NOTE— If NumberOfValues is zero there will be no ThreePackedEvents encoded in the vector.

Insert new subclause 10.8.2.10.2 as shown:

10.8.2.10.2 Encoding of Vector FourPackedEvents

The Vector is encoded as zero or more 8-bit values, each containing a numeric value, FourPackedEvents, derived from four packed numeric values, each of which represent a FourPackedType, in the range 0 through 3. The FourPackedTypes are defined by each MRP application that uses FourPackedEvents. Note that not all MRP applications use FourPackedEvents.

As can be seen from the BNF definition of FourPackedEvents, each 8-bit value is derived by successively adding a FourPackedType value and multiplying the result by 4. In order to facilitate the subsequent description, the event values are numbered from first to fourth, as follows:

FourPackedEvents BYTE ::= ((firstFourPackedType * 64) + (secondFourPackedType * 16) +
(thirdFourPackedType * 4) + (fourthFourPackedType))

The NumberOfValues field in the VectorHeader of the VectorAttribute determines the number of 8-bit FourPackedEvents values, E, that will be present in the vector; hence E is determined by dividing NumberOfValues by 4 and rounding any non-integer answer up to the nearest larger integer.

The FirstValue field of the VectorAttribute determines which of the originator's state machines the first FourPackedType value in the first FourPackedEvents value relates to. The second FourPackedType value in the first FourPackedEvents value corresponds to the state machine identified by (FirstValue + 1). The third FourPackedType value in the first FourPackedEvents value corresponds to the state machne identified by (FirstValue + 2), and the fourth FourPackedType value in the first FourPackedEvents value corresponds to the state machine identified by (FirstValue + 3), and so on, through subsequent packed values.

Where the NumberOfValues field carries a value that is not a multiple of 4, there will be either one, two or three FourPackedType values packed in the final FourPackedEvent that are ignored; these values are encoded as the numeric value 0 on transmission and are ignored on receipt.

NOTE— If NumberOfValues is zero there will be no FourPackedEvents encoded in the vector.

10.12 Definition of the MMRP application

10.12.1 Definition of MRP protocol elements

Change subclause 10.12.1.7 as shown:

10.12.1.7 MMRP FirstValue definitions

The FirstValue field (10.8.2.5) in instances of the MAC Vector Attribute Type shall be encoded in MRPDUs as six octets, each taken to represent an unsigned binary number. The octets are derived from the Hexadecimal Representation of a 48-bit MAC Address (defined in IEEE Std 802) as follows:

a) Each two-digit hexadecimal numeral in the Hexadecimal Representation is taken to represent an unsigned hexadecimal value, in the normal way, i.e., the rightmost digit of each numeral represents the least significant digit of the value, the leftmost digit is the most significant.
b) The first octet of the attribute value encoding is derived from the left-most hexadecimal value in the Hexadecimal Representation of the MAC Address. The least significant bit of the octet (bit 1) is assigned the least significant bit of the hexadecimal value, the next most significant bit is assigned the value of the second significant bit of the hexadecimal value, and so on.
c) The second through sixth octets of the encoding are similarly assigned the value of the second through sixth hexadecimal values in the Hexadecimal Representation of the MAC Address.

FirstValue+1 is defined as adding 1 to FirstValue. There are no restrictions on the range of values that can be represented in this data type.

The FirstValue field in instances of the Service Requirement Attribute Type shall be encoded in MRPDUs (10.8.2.5) as a single octet, taken to represent an unsigned binary number. Only two values of this type are defined:

d) All Groups shall be encoded as the value 0.
e) All Unregistered Groups shall be encoded as the value 1.

The remaining possible values (2 through 255) are reserved.

Insert new subclause 10.12.1.9 as shown:

10.12.1.9 MMRP AttributeListLength definitions

The AttributeListLength field (10.8.2.4) is not present in the MMRPDUs.

Insert new subclause 10.12.1.10 as shown:

10.12.1.10 MMRP Vector definitions

The ThreePackedEvent vectors are encoded as defined in subclause 10.8.2.10.1.

The FourPackedEvent vectors (10.8.2.10.2) are not present in the MMRPDUs.

11. VLAN topology management

11.2 Multiple VLAN Registration Protocol

11.2.3 Definition of the MVRP application

11.2.3.1 Definition of MRP protocol elements

Insert new subclause 11.2.3.1.9 as shown:

11.2.3.1.9 MVRP AttributeListLength definitions

The AttributeListLength field (10.8.2.4) is not present in the MVRPDUs.

Insert new subclause 11.2.3.1.10 as shown:

11.2.3.1.10 MVRP Vector definitions

The ThreePackedEvent vectors are encoded as defined in subclause 10.8.2.10.1.

The FourPackedEvent vectors (10.8.2.10.2) are not present in the MVRPDUs.

12. Bridge management

Insert the following as new subclase 12.22, immediately following subclause 12.21, renumbering subsequent subclauses and table numbers if necessary:

12.22 SRP entities

The Bridge enhancements for support of SRP (Stream Reservation Protocol) are defined in Clause 35.

The objects that comprise this managed resource are

a) The SRP Bridge Base Table (12.22.1)
b) The SRP Bridge Port Table (12.22.2)
c) The SRP Latency Parameter Table (12.22.3)
d) The SRP Stream Table (12.22.4)
e) The SRP Reservations Table (12.22.5)

12.22.1 The SRP Bridge Base Table

There is a set of parameters that configure SRP operation for the entire device. Those parameters are shown in Table 12-4.

Table 12-4—SRP Bridge Base Table row elements

Name	Data type	Operations supported[a]	Conformance[b]	References
msrpEnabledStatus	boolean	RW	B	35.2.1.4(d)
talkerPruning	boolean	RW	B	35.2.1.4(b)
msrpMaxFanInPorts	unsigned integer	RW	B	35.2.1.4(f)
msrpLatencyMaxFrameSize	unsigned integer	RW	B	35.2.1.4(g)

[a]R = Read only access; RW = Read/Write access
[b]B = required for bridge or bridge component support of SRP, E = required for end station support of SRP

12.22.2 The SRP Bridge Port Table

There is one SRP Configuration Parameter Table per Port of a bridge component. Each table row contains a set of parameters for each MSRP entity per port, as detailed in Table 12-2.

12.22.3 The SRP Latency Parameter Table

There is one SRP Latency Parameter Table per Port of a bridge component. Each table row contains a set of parameters for each traffic class supported on a port, as detailed in Table 12-3. Rows in the table can be created or removed dynamically in implementations that support dynamic configuration of ports and components...

Table 12-2—SRP Bridge Port Table row elements

Name	Data type	Operations supported[a]	Conformance[b]	References
msrpPortEnabledStatus	boolean	RW	B	35.2.1.4(e)
MSRP Failed Registrations	counter	R	B	10.7.12.1
MSRP Last PDU Origin	MAC Address	R	B	10.7.12.2
SR_PVID	unsigned integer[1..4094]	RW	B	Table 9-2 35.2.1.4(i)

[a]R = Read only access; RW = Read/Write access
[b]B = required for bridge or bridge component support of SRP, E = required for end station support of SRP

Table 12-3—SRP Latency Parameter Table row elements

Name	Data type	Operations supported[a]	Conformance[b]	References
Traffic class	unsigned integer [0..7]	R	BE	34.3
portTcMaxLatency:	unsigned integer	R	BE	35.2.1.4(a), 35.2.2.8.6

[a]R = Read only access; RW = Read/Write access
[b]B = required for bridge or bridge component support of SRP, E = required for end station support of SRP

12.22.4 The SRP Stream Table

There is one SRP Stream Table per bridge component. Each table contains a set of parameters for each StreamID that is registered on the Bridge, as detailed in Table 12-4. Rows in the table are created and removed dynamically as StreamIDs are registered and deregistered on the Bridge.

12.22.5 The SRP Reservations Table

There is one SRP Reservations Table per reservation direction per port of a bridge component. Each table contains a set of parameters for each Talker or Listener Reservation that is registered on a port of the Bridge, as detailed in Table 12-5. Rows in the table can be created or removed dynamically as Talker and Listener declarations are registered and deregistered on a port of the Bridge..

Table 12-4—SRP Stream Table row elements

Name	Data type	Operations supported[a]	Conformance[b]	References
StreamID	octet string(size(8))	RW	BE	35.2.2.8.2
Stream Destination Address	MAC Address	R	BE	35.2.2.8.3(a)
Stream VLAN ID	unsigned integer [0..4094]	R	BE	35.2.2.8.3(b)
MaxFrameSize	unsigned integer [0..65535]	R	BE	35.2.2.8.4(a)
MaxIntervalFrames	unsigned integer [0..65535]	R	BE	35.2.2.8.4(b)
Data Frame Priority	unsigned integer [0..7]	R	BE	35.2.2.8.5(a)
Rank	unsigned integer [0..1]	R	BE	35.2.2.8.5(b)

[a]R = Read only access; RW = Read/Write access
[b]B = required for bridge or bridge component support of SRP, E = required for end station support of SRP

Table 12-5—SRP Reservations Table row elements

Name	Data type	Operations supported[a]	Conformance[b]	References
StreamID	octet string(size(8))	RW	BE	35.2.2.8.2
Direction	unsigned integer[0..1]	R	BE	35.2.1.2
Declaration Type	unsigned integer [0..4]	R	BE	35.2.1.3
Accumulated Latency	unsigned integer	R	BE	35.2.2.8.6
Failed Bridge ID	BridgeId	R	BE	35.2.2.8.7(a)
Failure Code	unsigned integer [0..16]	R	BE	35.2.2.8.7(b)
Dropped Frames	counter	R	BE	35.2.5.1
Stream Age	unsigned integer	R	BE	35.2.1.4(c)

[a]R = Read only access; RW = Read/Write access
[b]B = required for bridge or bridge component support of SRP, E = required for end station support of SRP

17. Management Information Base (MIB)

17.2 Structure of the MIB

Insert a row at the appropriate position in Table 17-1:

Table 17-1—Structure of the MIB Modules

Module	subclause	Defining standard	Reference	Notes
IEEE8021-SRP MIB	17.7.14	802.1Qat	35	Initial version of 802.1Qat

Insert new subclause 17.2.14, and Table 17-19, as follows, renumbering Table 17-18 if necessary to follow in sequence from any tables referenced in the text prior to 17.2.13. Renumber subsequent tables/subclauses as necessary.

17.2.14 Structure of the IEEE8021-SRP MIB

The IEEE8021-SRP MIB provides objects to configure and manage those aspects of a VLAN Bridge that are related to the Stream Reservation Protocol (SRP).

Objects in this MIB module are arranged into subtrees. Each subtree is organized as a set of related objects. Where appropriate, the corresponding Clause 12 mamagement reference is also included.

Table 17-19 that follows indicates the structure of the IEEE8021-SRP MIB module.

17.3 Relationship to other MIBs

Insert the following new subclause 17.3.14.

17.3.14 Relationship of the IEEE8021-SRP MIB to other MIB modules

The IEEE8021-SRP MIB provides objects that extend the core management functionality of a Bridge, as defined by the IEEE8021-BRIDGE MIB (17.7.2), in order to support the management functionality needed when the Stream Reservation Protocol extensions, as defined in Clause 35, are supported by the Bridge. As support of these objects defined in the IEEE8021-SRP MIB also requires support of the IEEE8021-TC-MIB, IEEE8021-BRIDGE-MIB and IEEE8021-FQTSS-MIB, the provisions of 17.3.2 apply to implementations claming support of the IEEE8021-SRP MIB.

17.4 Security considerations

Insert the following new subclause 17.4.14, as follows, renumbering subsequent subclauses as necessary:

Table 17-19—SRP MIB structure and object cross reference

MIB table	MIB object	References
ieee8021SrpConfiguration subtree		
ieee8021SrpBridgeBaseTable		SRP control and status information for a bridge. Augments ieee8021BridgeBaseEntry.
	ieee8021SrpBridgeBaseMsrpEnabledStatus	Is MSRP enabled on this device? True or False, 12.22.1, 35.2.1.4d
	ieee8021SrpBridgeBaseMsrpTalkerPruning	talkerPruning, 12.22.1, 35.2.1.4b
	ieee8021SrpBridgeBaseMsrpMaxFanInPorts	msrpMaxFanInPorts, 12.22.1, 35.2.1.4(f)
	ieee8021SrpBridgeBaseMsrpLatencyMaxFrameSize	msrpLatencyMaxFrameSize, 12.22.1, 35.2.1.4(g)
ieee8021SrpBridgePortTable		SRP Control and Status information for each port on the Bridge. Augments ieee8021BridgeBasePortEntry.
	ieee8021SrpBridgePortMsrpEnabledStatus	Is MSRP enabled on this port? True or False, 12.22.2, 35.2.1.4e.
	ieee8021SrpBridgePortMsrpFailedRegistrations	How many failed registrations have there been on this port, 10.7.12.1, 12.22.2.
	ieee8021SrpBridgePortMsrpLastPduOrigin	Source MAC Address of last MSRPDU received on this port, 10.7.12.2, 12.22.2.
	ieee8021SrpBridgePortSrPvid	Default VLAN ID for Streams on this port, Table 9-2, Table 12-2, 12.22.2, 35.2.1.4(i)
ieee8021SrpLatency subtree		
ieee8021SrpLatencyTable		Maximum port latency per traffic class Table, 12.22.3, 35.2.2.8.6
	ieee8021SrpTrafficClass	Traffic class (Table index)
	ieee8021SrpPortTcLatency	Maximum port latency for the associated Traffic Class, 12.22.3, 35.2.1.4a, 35.2.2.8.6
ieee8021SrpStreams subtree		
ieee8021SrpStreamTable		Components that define the characteristics of a Stream., 12.22.4, 35.2.2.8.
	ieee8021SrpStreamID	StreamID (Table index), 12.22.4, 35.2.2.8.2.
	ieee8021SrpStreamDestinationAddress	Stream destination MAC address, 12.22.4, 35.2.2.8.3a.
	ieee8021SrpStreamVlanID	VLAN ID for Stream (0=default), 12.22.4,35.2.2.8.3b.

Table 17-19—SRP MIB structure and object cross reference

MIB table	MIB object	References
	ieee8021SrpStreamTspecMaxFrameSize	Maximum frame size sent by Talker, 12.22.4,35.2.2.8.4a.
	ieee8021SrpStreamTspecMaxIntervalFrames	Maximum number of frames sent per class measurement interval, 12.22.4, 35.2.2.8.4b, 34.4.
	ieee8021SrpStreamDataFramePriority	The Priority Code Point (PCP) value that the data Stream will be tagged with, 12.22.4, 35.2.2.8.5a.
	ieee8021SrpStreamRank	Emergency/non-emergency Rank associated with the Stream, 12.22.4, 35.2.2.8.5b.
ieee8021SrpReservations subtree		
ieee8021SrpReservationsTable		A table containing Stream attribute registrations per port, 12.22.5, 35.2.4.
	ieee8021SrpReservationStreamId	StreamID (Table index), 12.22.5, 35.2.2.8.2.
	ieee8021SrpReservationDirection	Talker or Listener (Table index), 12.22.5, 35.2.1.2.
	ieee8021SrpReservationDeclarationType	Advertise or Failed for Talkers. Asking Failed, Ready or Ready Failed for Listeners.12.22.5, 35.2.1.3.
	ieee8021SrpReservationAccumulatedLatency	Latency at ingress port for Talker registrations, or latency at end of egress media for Listener Declarations, 12.22.5, 35.2.2.8.6.
	ieee8021SrpReservationFailureBridgeId	Bridge ID of Bridge that change Talker Advertise to Talker Failed, 12.22.5, 35.2.2.8.7a.
	ieee8021SrpReservationFailureCode	Failure Code associated with Bridge that changed Talker Advertise to Talker Failed, 12.22.5, 35.2.2.8.7b.
	ieee8021SrpReservationDroppedStreamFrames	Number of stream data frames (not MSRPDUs) that have been dropped for this stream on this port, 12.22.5, 35.2.5.1.
	ieee8021SrpReservationStreamAge	Number of seconds since reservation was established, 12.22.5, 35.2.1.4c.

17.4.14 Security considerations of the IEEE8021-SRP MIB

The purpose of MSRP is to create reservations for various types of data streams, including Audio/Video content. Access to the objects within the IEEE8021-SRP MIB module, whether they have MAX-ACCESS of read-write, read-create, or read-only, may reveal sensitive information in some network environments. Very serious health and safety situations could arise if MSRP was involved in configuring network resources for an emergency public safety announcement and the MSRP behavior of the bridged network was allowed to be modified unexpectedly.

With these consideration in mind it is thus important to control all types of access (including GET and/or NOTIFY) to these objects and possibly even encrypt the their values when sending them over the network via SNMP.

The following tables and objects in the IEEE8021-SRP MIB can be manipulated to interfere with the operation of the stream reservation mechanisms in a manner that would be detrimental to the transmission of the associated stream data:

ieee8021SrpBridgeBaseMsrpEnabledStatus
ieee8021SrpBridgeBaseMsrpTalkerPruning
ieee8021SrpBridgeBaseMsrpMaxFanInPorts
ieee8021SrpBridgeBaseMsrpLatencyMaxFrameSize
ieee8021SrpBridgePortMsrpEnabledStatus
ieee8021SrpBridgePortSrPvid

a) ieee8021SrpBridgeBaseMsrpEnabledStatus can be manipulated to enable or disable MSRP protocol operations for the entire Bridge.
b) ieee8021SrpBridgeBaseMsrpTalkerPruning can be manipulated to stop the propagation of Talker attributes if Listeners aren't configured to support Talker Pruning.
c) ieee8021SrpBridgeBaseMsrpMaxFanInPorts can be manipulated to set the number of ingress ports supporting streaming down to one, which would stop Talkers streams from coming in on any other port.
d) ieee8021SrpBridgeBaseMsrpLatencyMaxFrameSize can be manipulates to set a frame size that is so large or so small that it causes the Bridge to calculate unreasonable maximum latency.
e) ieee8021SrpBridgePortMsrpEnabledStatus can be manipulated to enable or disable MSRP on a particular port, perhaps allowing sensitive stream data to be sent to unacceptable devices.
f) ieee8021SrpBridgePortSrPvid can be manipulated to move Streams to a VLAN that has been blocked by management, thus disabling reception of the Stream by one or more Listeners.

17.7 MIB modules

Insert the following new subclause 17.7.14, as follows, renumbering subsequent subclauses as necessary:

17.7.14 Definitions of the IEEE8021-SRP MIB module

```
IEEE8021-SRP-MIB DEFINITIONS ::= BEGIN

-- =============================================================
-- MIB for support of 802.1Qat Stream Reservation Protocol
-- (SRP) in 802.1Q Bridges.
-- =============================================================

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Counter64,
    Unsigned32
        FROM SNMPv2-SMI
    MacAddress,
    TEXTUAL-CONVENTION,
    TruthValue
        FROM SNMPv2-TC
    MODULE-COMPLIANCE,
```

```
    OBJECT-GROUP
        FROM SNMPv2-CONF
    ieee802dot1mibs,
    IEEE8021PriorityCodePoint,
    IEEE8021VlanIndex
        FROM IEEE8021-TC-MIB
    IEEE8021FqtssTrafficClassValue
        FROM IEEE8021-FQTSS-MIB
    ieee8021BridgeBaseComponentId,
    ieee8021BridgeBaseEntry,
    ieee8021BridgeBasePort,
    ieee8021BridgeBasePortEntry
        FROM IEEE8021-BRIDGE-MIB
    BridgeId
        FROM BRIDGE-MIB
    ;

ieee8021SrpMib MODULE-IDENTITY
    LAST-UPDATED "201004190000Z" -- April 19, 2010
    ORGANIZATION "IEEE 802.1 Working Group"
    CONTACT-INFO
        "WG-URL: http://grouper.ieee.org/groups/802/1/index.html
         WG-EMail: stds-802-1@ieee.org

          Contact: Craig Gunther
           Postal: Harman International Industries
                   8760 S. Sandy Parkway
                   Sandy, Utah 84070
                   United States

           E-mail: craig.gunther@harman.com"
    DESCRIPTION
        "The Bridge MIB module for managing devices that support
        the IEEE 802.1Qat Stream Reservation Protocol.

        Unless otherwise indicated, the references in this MIB
        module are to IEEE Std 802.1Q-2005/Cor 1 as amended by
        IEEE Std 802.1ad, IEEE Std 802.1ak, IEEE Std 802.1ag,
        IEEE Std 802.1ah, IEEE Std 802.1ap, IEEE Std 802.1aw,
        IEEE Std 802.1ay, IEEE Std 802.1Qav, and IEEE Std 802.1Qat.

        Copyright (C) IEEE.
        This version of this MIB module is part of IEEE802.1Q;
        see the draft itself for full legal notices."
    REVISION     "201004190000Z" -- April 19, 2010
    DESCRIPTION
         "Initial revision, included in IEEE 802.1Qat"
    ::= { ieee802dot1mibs 19 }

-- =============================================================
-- Textual Conventions
-- =============================================================

IEEE8021SrpStreamRankValue ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "An 802.1 SRP Stream Rank value. This is an integer,
         with the following interpretation placed on the value:
```

```
          0: Emergency, high-rank stream,
          1: Non-emergency stream."
    REFERENCE    "35.2.2.8.5b"
    SYNTAX       INTEGER {
                     emergency(0),
                     nonEmergency(1)
                 }


IEEE8021SrpStreamIdValue ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "1x:1x:1x:1x:1x:1x.1x:1x"
    STATUS       current
    DESCRIPTION
            "Represents an SRP Stream ID, which is often defined
            as a MAC Address followed by a unique 16-bit ID."
    SYNTAX       OCTET STRING (SIZE (8))

IEEE8021SrpReservationDirectionValue ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "An 802.1 SRP Stream Reservation Direction value. This is
         an integer, with the following interpretation placed on
         the value:

         0: Talker registrations,
         1: Listener registrations."
    REFERENCE    "35.2.1.2"
    SYNTAX       INTEGER {
                     talkerRegistrations(0),
                     listenerRegistrations(1)
                 }


IEEE8021SrpReservationDeclarationTypeValue ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "An 802.1 SRP Stream Reservation Declaration Type value.
         This is an integer, with the following interpretation
         placed on the value:

         0: Talker Advertise,
         1: Talker Failed,
         2: Listener Asking Failed,
         3: Listener Ready,
         4: Listener Ready Failed."
    REFERENCE    "35.2.1.3"
    SYNTAX       INTEGER {
                     talkerAdvertise(0),
                     talkerFailed(1),
                     listenerAskingFailed(2),
                     listenerReady(3),
                     listenerReadyFailed(4)
                 }


IEEE8021SrpReservationFailureCodeValue ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "An 802.1 SRP Stream Reservation Failure Code value.
```

```
            This is an integer, with the following interpretation
            placed on the value:

            0: No failure,
            1: Insufficient bandwidth,
            2: Insufficient Bridge resources,
            3: Insufficient bandwidth for Traffic Class,
            4: StreamID in use by another Talker,
            5: Stream destination address already in use,
            6: Stream pre-empted by higher rank,
            7: Reported latency has changed,
            8: Egress port is not AVBCapable,
            9: Use a different destination_address,
            10: Out of MSRP resources,
            11: Out of MMRP resources,
            12: Cannot store destination_address,
            13: Requested priority is not an SR Class priority,
            14: MaxFrameSize is too large for media,
            15: maxFanInPorts limit has been reached,
            16: Changes in FirstValue for a registered StreamID,
            17: VLAN is blocked on this egress port (Registration Forbidden),
            18: VLAN tagging is disabled on this egress port (untagged set),
            19: SR class priority mismatch."
    REFERENCE    "35.2.2.8.7"
    SYNTAX       INTEGER {
                     noFailure(0),
                     insufficientBandwidth(1),
                     insufficientResources(2),
                     insufficientTrafficClassBandwidth(3),
                     streamIDInUse(4),
                     streamDestinationAddressInUse(5),
                     streamPreemptedByHigherRank(6),
                     latencyHasChanged(7),
                     egressPortNotAVBCapable(8),
                     useDifferentDestinationAddress(9),
                     outOfMSRPResources(10),
                     outOfMMRPResources(11),
                     cannotStoreDestinationAddress(12),
                     priorityIsNoAnSRCLass(13),
                     maxFrameSizeTooLarge(14),
                     maxFanInPortsLimitReached(15),
                     firstValueChangedForStreamID(16),
                     vlanBlockedOnEgress(17),
                     vlanTaggingDisabledOnEgress(18),
                     srClassPriorityMismatch(19)
                 }


-- ===============================================================
-- subtrees in the SRP MIB
-- ===============================================================

ieee8021SrpNotifications
    OBJECT IDENTIFIER ::= { ieee8021SrpMib 0 }

ieee8021SrpObjects
    OBJECT IDENTIFIER ::= { ieee8021SrpMib 1 }

ieee8021SrpConformance
```

```
    OBJECT IDENTIFIER ::= { ieee8021SrpMib 2 }

ieee8021SrpConfiguration
    OBJECT IDENTIFIER ::= { ieee8021SrpObjects 1 }

ieee8021SrpLatency
    OBJECT IDENTIFIER ::= { ieee8021SrpObjects 2 }

ieee8021SrpStreams
    OBJECT IDENTIFIER ::= { ieee8021SrpObjects 3 }

ieee8021SrpReservations
    OBJECT IDENTIFIER ::= { ieee8021SrpObjects 4 }


-- =============================================================
-- The ieee8021SrpConfiguration subtree
-- This subtree defines the objects necessary for the
-- operational management of SRP.
-- =============================================================

ieee8021SrpBridgeBaseTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021SrpBridgeBaseEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table for SRP main control and status information.
        All writeable objects in this table must be persistent
        over power up restart/reboot.  These objects augment
        the ieee8021BridgeBasePortTable."
    ::= { ieee8021SrpConfiguration 1 }

ieee8021SrpBridgeBaseEntry OBJECT-TYPE
    SYNTAX      Ieee8021SrpBridgeBaseEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "SRP control and status information for a bridge."
    AUGMENTS { ieee8021BridgeBaseEntry }
    ::= { ieee8021SrpBridgeBaseTable 1 }

Ieee8021SrpBridgeBaseEntry ::=
    SEQUENCE {
         ieee8021SrpBridgeBaseMsrpEnabledStatus
            TruthValue,
         ieee8021SrpBridgeBaseMsrpTalkerPruning
            TruthValue,
         ieee8021SrpBridgeBaseMsrpMaxFanInPorts
            Unsigned32,
         ieee8021SrpBridgeBaseMsrpLatencyMaxFrameSize
            Unsigned32
    }

ieee8021SrpBridgeBaseMsrpEnabledStatus OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The administrative status requested by management for
```

```
            MSRP.  The value true(1) indicates that MSRP should
            be enabled on this device, in all VLANs, on all ports
            for which it has not been specifically disabled.  When
            false(2), MSRP is disabled, in all VLANs and on all
            ports, and all MSRP frames will be forwarded
            transparently.  This objects affects both Applicant and
            Registrar state machines.  A transition from false(2)
            to true(1) will cause a reset of all MSRP state
            machines on all ports.

            This object may be modified while the corresponding
            instance of ieee8021BridgeBaseRowStatus is active(1).

            The value of this object MUST be retained across
            reinitializations of the management system."
        REFERENCE   "35.2.1.4d"
        DEFVAL      { true }
        ::= { ieee8021SrpBridgeBaseEntry 1 }

    ieee8021SrpBridgeBaseMsrpTalkerPruning OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The value of the talkerPruning parameter which
            controls the propagation of Talker declarations.
            The value true(1) indicates that Talker attributes
            are only declared on ports that have the Stream
            destination_address registered in the MMRP MAC
            Address Registration Entries.  When false(2),
            Talker attribute are declared on all egress ports
            in the active topology.

            The value of this object MUST be retained across
            reinitializations of the management system."
        REFERENCE   "12.22.1, 35.2.1.4b, 35.2.4.3.1"
        DEFVAL      { false }
        ::= { ieee8021SrpBridgeBaseEntry 2 }

    ieee8021SrpBridgeBaseMsrpMaxFanInPorts OBJECT-TYPE
        SYNTAX      Unsigned32
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The value of the msrpMaxFanInPorts parameter which
            limits the total number of ports on a Bridge that
            are allowed to establish reservations for inbound
            Streams.  A value of zero (0) indicates no fan-in
            limit is being specified and calculations involving
            fan-in will only be limited by the number of MSRP
            enabled ports.

            The value of this object MUST be retained across
            reinitializations of the management system."
        REFERENCE   "12.22.1, 35.2.1.4f"
        DEFVAL      { 0 }
        ::= { ieee8021SrpBridgeBaseEntry 3 }

    ieee8021SrpBridgeBaseMsrpLatencyMaxFrameSize OBJECT-TYPE
```

```
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The value of msrpLatencyMaxFrameSize parameter
        which is used in the calculation of the maximum
        latency through a bridge.  The maximum size is
        defined to be 2000 octets by default, but may be
        set to a smaller or larger value dependent on the
        particular Bridge configuration. This parameter
        does not imply any type of policing of frame size,
        it is only used in the latency calculations.

        The value of this object MUST be retained across
        reinitializations of the management system."
    REFERENCE   "12.22.1, 35.2.1.4g"
    DEFVAL      { 2000 }
    ::= { ieee8021SrpBridgeBaseEntry 4 }


ieee8021SrpBridgePortTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021SrpBridgePortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table for SRP control and status information about
        every bridge port.  Augments the ieee8021BridgeBasePortTable."
    ::= { ieee8021SrpConfiguration 2 }

ieee8021SrpBridgePortEntry OBJECT-TYPE
    SYNTAX      Ieee8021SrpBridgePortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "SRP control and status information for a bridge port."
    AUGMENTS { ieee8021BridgeBasePortEntry }
    ::= { ieee8021SrpBridgePortTable 1 }

Ieee8021SrpBridgePortEntry ::=
    SEQUENCE {
        ieee8021SrpBridgePortMsrpEnabledStatus
            TruthValue,
        ieee8021SrpBridgePortMsrpFailedRegistrations
            Counter64,
        ieee8021SrpBridgePortMsrpLastPduOrigin
            MacAddress,
        ieee8021SrpBridgePortSrPvid
            IEEE8021VlanIndex
    }

ieee8021SrpBridgePortMsrpEnabledStatus OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The administrative state of MSRP operation on this port.  The
        value true(1) indicates that MSRP is enabled on this port
        in all VLANs as long as ieee8021BridgeMsrpEnabledStatus is
        also true(1).  A value of false(2) indicates that MSRP is
```

```
            disabled on this port in all VLANs: any MSRP frames received
            will be silently discarded, and no MSRP registrations will be
            propagated from other ports.  Setting this to a value of
            true(1) will be stored by the agent but will only take
            effect on the MSRP protocol operation if
            ieee8021BridgeMsrpEnabledStatus
            also indicates the value true(1).  This object affects
            all MSRP Applicant and Registrar state machines on this
            port.  A transition from false(2) to true(1) will
            cause a reset of all MSRP state machines on this port.

            The value of this object MUST be retained across
            reinitializations of the management system."
        REFERENCE   "35.2.1.4e"
        DEFVAL      { true }
        ::= { ieee8021SrpBridgePortEntry 1 }

    ieee8021SrpBridgePortMsrpFailedRegistrations OBJECT-TYPE
        SYNTAX      Counter64
        UNITS       "failed MSRP registrations"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The total number of failed MSRP registrations, for any
            reason, in all VLANs, on this port.

            Discontinuities in the value of the counter can occur at
            re-initialization of the management system, and at other
            times as indicated by the value of ifCounterDiscontinuityTime
            object of the associated interface (if any)."
        REFERENCE   "10.7.12.1"
        ::= { ieee8021SrpBridgePortEntry 2 }

    ieee8021SrpBridgePortMsrpLastPduOrigin OBJECT-TYPE
        SYNTAX      MacAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The Source MAC Address of the last MSRP message
            received on this port."
        REFERENCE   "10.7.12.2"
        ::= { ieee8021SrpBridgePortEntry 3 }

    ieee8021SrpBridgePortSrPvid OBJECT-TYPE
        SYNTAX      IEEE8021VlanIndex
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The default VLAN ID that Streams are assigned to.
            Talkers learn this VID from the SRP Domain attribute
            and tag Streams accordingly.

            The value of this object MUST be retained across
            reinitializations of the management system."
        REFERENCE   "35.2.2.8.3b"
        DEFVAL      { 2 }
        ::= { ieee8021SrpBridgePortEntry 4}
```

```
-- =============================================================
-- The ieee8021SrpLatency subtree
-- This subtree defines the objects necessary for retrieving
-- the latency of the various traffic classes on a port.
-- =============================================================

-- =============================================================
-- the ieee8021SrpLatencyTable
-- =============================================================
ieee8021SrpLatencyTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021SrpLatencyEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing a set of latency measurement
        parameters for each traffic class."
    REFERENCE   "35.2.2.8.6"
    ::= { ieee8021SrpLatency 1 }

ieee8021SrpLatencyEntry OBJECT-TYPE
    SYNTAX      Ieee8021SrpLatencyEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects containing latency information
        for each traffic class. Rows in the table are
        automatically created for ports that are not an
        SRP domain boundary port (i.e. SRPdomainBoundaryPort
        is FALSE). Refer to Clause 6.6.4, 8.8.2, 12.21.3."
    INDEX  { ieee8021BridgeBaseComponentId,
             ieee8021BridgeBasePort,
             ieee8021SrpTrafficClass  }
    ::= { ieee8021SrpLatencyTable 1 }

Ieee8021SrpLatencyEntry ::=
    SEQUENCE {
        ieee8021SrpTrafficClass
            IEEE8021FqtssTrafficClassValue,
        ieee8021SrpPortTcLatency
            Unsigned32
        }

ieee8021SrpTrafficClass OBJECT-TYPE
    SYNTAX      IEEE8021FqtssTrafficClassValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The traffic class number associated with the
        row of the table.

        Rows in the table are automatically created for
        ports that are not an SRP domain boundary port
        (i.e. SRPdomainBoundaryPort is FALSE)."
    REFERENCE   "6.6.4, 8.8.2, 12.21.3"
    ::= { ieee8021SrpLatencyEntry 1 }


ieee8021SrpPortTcLatency OBJECT-TYPE
    SYNTAX      Unsigned32
```

```
    UNITS       "nano-seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The value of the portTcMaxLatency parameter for the
        traffic class.  This value is expressed in
        nano-seconds."
    REFERENCE   "35.2.1.4, 35.2.2.8.6"
    ::= { ieee8021SrpLatencyEntry 2}


-- =============================================================
-- The ieee8021SrpStreams subtree
-- This subtree defines the objects necessary for retrieving
-- the characteristics of the various Streams currently registered.
-- =============================================================

-- =============================================================
-- the ieee8021SrpStreamTable
-- =============================================================
ieee8021SrpStreamTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021SrpStreamEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing a set of characteristics
        for each registered Stream."
    REFERENCE   "35.2.2.8"
    ::= { ieee8021SrpStreams 1 }

ieee8021SrpStreamEntry OBJECT-TYPE
    SYNTAX      Ieee8021SrpStreamEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects containing characteristics
        for each registered Stream. Rows in the table are
        automatically created for Streams registered on any
        port of a bridge."
    INDEX  { ieee8021SrpStreamId }
    ::= { ieee8021SrpStreamTable 1 }

Ieee8021SrpStreamEntry ::=
    SEQUENCE {
        ieee8021SrpStreamId
            IEEE8021SrpStreamIdValue,
        ieee8021SrpStreamDestinationAddress
            MacAddress,
        ieee8021SrpStreamVlanId
            IEEE8021VlanIndex,
        ieee8021SrpStreamTspecMaxFrameSize
            Unsigned32,
        ieee8021SrpStreamTspecMaxIntervalFrames
            Unsigned32,
        ieee8021SrpStreamDataFramePriority
            IEEE8021PriorityCodePoint,
        ieee8021SrpStreamRank
            IEEE8021SrpStreamRankValue
        }
```

```
ieee8021SrpStreamId OBJECT-TYPE
    SYNTAX      IEEE8021SrpStreamIdValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Stream ID associated with the row of the table.

        Rows in the table are automatically created when
        Streams are registered via MSRP."
    REFERENCE   "35.2.2.8.2"
    ::= { ieee8021SrpStreamEntry 1 }

ieee8021SrpStreamDestinationAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The MAC destination address for the Stream described
        by this reservation."
    REFERENCE   "35.2.2.8.3a"
    ::= { ieee8021SrpStreamEntry 2}

ieee8021SrpStreamVlanId OBJECT-TYPE
    SYNTAX      IEEE8021VlanIndex
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The VLAN ID associated with the MSRP registration
        for this Stream."
    REFERENCE   "35.2.2.8.3b"
    ::= { ieee8021SrpStreamEntry 3}

ieee8021SrpStreamTspecMaxFrameSize OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The maximum size frame that will be sent by
        a Talker for this Stream. This value is part
        of the Traffic Specification for the Stream."
    REFERENCE   "35.2.2.8.4a"
    ::= { ieee8021SrpStreamEntry 4}

ieee8021SrpStreamTspecMaxIntervalFrames OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The maximum number of frame that will be sent
        during a class measurement interval (L.2).  This
        value is part of the Traffic Specification for
        the Stream."
    REFERENCE   "35.2.2.8.4b, L.2"
    ::= { ieee8021SrpStreamEntry 5}

ieee8021SrpStreamDataFramePriority OBJECT-TYPE
    SYNTAX      IEEE8021PriorityCodePoint
    MAX-ACCESS  read-only
```

```
    STATUS      current
    DESCRIPTION
        "The Priority Code Point (PCP) value that the
        referenced Stream will be tagged with. This value
        is used to distinguish Class A and Class B traffic."
    REFERENCE   "35.2.2.8.5a"
    ::= { ieee8021SrpStreamEntry 6}

ieee8021SrpStreamRank OBJECT-TYPE
    SYNTAX      IEEE8021SrpStreamRankValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "SRP supports emergency and non-emergency.
        Emergency traffic will interrupt non-emergency
        traffic if there is insufficient bandwidth or
        resources available for the emergency traffic."
    REFERENCE   "35.2.2.8.5b"
    ::= { ieee8021SrpStreamEntry 7}


-- ===============================================================
-- The ieee8021SrpReservations subtree
-- This subtree defines the objects necessary for retrieving
-- the Stream attribute registrations on each port of a Bridge.
-- ===============================================================

-- ===============================================================
-- the ieee8021SrpReservationsTable
-- ===============================================================
ieee8021SrpReservationsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF Ieee8021SrpReservationsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table containing Stream attribute
        registrations per port."
    REFERENCE   "35.2.4"
    ::= { ieee8021SrpReservations 1 }

ieee8021SrpReservationsEntry OBJECT-TYPE
    SYNTAX      Ieee8021SrpReservationsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of objects containing Stream attribute
        registrations per port. Rows in the table are
        automatically created for Streams registered on any
        port of a bridge."
    INDEX  { ieee8021SrpReservationStreamId,
             ieee8021SrpReservationDirection,
             ieee8021BridgeBaseComponentId,
             ieee8021BridgeBasePort }
    ::= { ieee8021SrpReservationsTable 1 }

Ieee8021SrpReservationsEntry ::=
    SEQUENCE {
        ieee8021SrpReservationStreamId
            IEEE8021SrpStreamIdValue,
```

```
        ieee8021SrpReservationDirection
            IEEE8021SrpReservationDirectionValue,
        ieee8021SrpReservationDeclarationType
            IEEE8021SrpReservationDeclarationTypeValue,
        ieee8021SrpReservationAccumulatedLatency
            Unsigned32,
        ieee8021SrpReservationFailureBridgeId
            BridgeId,
        ieee8021SrpReservationFailureCode
            IEEE8021SrpReservationFailureCodeValue,
        ieee8021SrpReservationDroppedStreamFrames
            Counter64,
        ieee8021SrpReservationStreamAge
            Unsigned32
        }

ieee8021SrpReservationStreamId OBJECT-TYPE
    SYNTAX      IEEE8021SrpStreamIdValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The Stream ID associated with the row of the table.

        Rows in the table are automatically created when
        Streams are registered via MSRP."
    REFERENCE   "35.2.2.8.2"
    ::= { ieee8021SrpReservationsEntry 1 }

ieee8021SrpReservationDirection OBJECT-TYPE
    SYNTAX      IEEE8021SrpReservationDirectionValue
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The source of this Stream registration, either
        Talker or Listener."
    REFERENCE   "35.2.1.2"
    ::= { ieee8021SrpReservationsEntry 2 }

ieee8021SrpReservationDeclarationType OBJECT-TYPE
    SYNTAX      IEEE8021SrpReservationDeclarationTypeValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The type of Talker or Listener registration."
    REFERENCE   "35.2.1.3"
    ::= { ieee8021SrpReservationsEntry 3 }

ieee8021SrpReservationAccumulatedLatency OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "nano-seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Accumulated Latency associated with the current
        registration.

        For Talker registrations this represents the accumulated
        latency from the Talker to the ingress port of this
        Bridge.
```

```
            For Listener registrations this represents the accumulated
            latency to the ingress port of the neighbor Bridge or
            end stations. This include the latency of the media
            attached to this egress port."
        REFERENCE   "35.2.2.8.6"
        ::= { ieee8021SrpReservationsEntry 4 }

    ieee8021SrpReservationFailureBridgeId OBJECT-TYPE
        SYNTAX      BridgeId
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The first Bridge that changes a Talker Advertise to a
            Talker Failed registration will report its Bridge
            Identification in this field. That single Bridge
            Identification is then propagated from Bridge to Bridge."
        REFERENCE   "35.2.2.8.7a"
        ::= { ieee8021SrpReservationsEntry 5 }

    ieee8021SrpReservationFailureCode OBJECT-TYPE
        SYNTAX      IEEE8021SrpReservationFailureCodeValue
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The first Bridge that changes a Talker Advertise to a
            Talker Failed registration will report the Failure Code
            in this field. That single Failure Code is then propagated
            from Bridge to Bridge."
        REFERENCE   "35.2.2.8.7b"
        ::= { ieee8021SrpReservationsEntry 6 }

    ieee8021SrpReservationDroppedStreamFrames OBJECT-TYPE
        SYNTAX      Counter64
        UNITS       "frames"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "A count of the number of data stream frames that have
            been dropped for whatever reason.  These are not MSRP
            frames, but the stream data frames that are carried by
            the MSRP Reservation.

            Discontinuities in the value of the counter can occur at
            re-initialization of the management system, and at other
            times as indicated by the value of ifCounterDiscontinuityTime
            object of the associated interface (if any)."
         REFERENCE   "35.2.5.1"
        ::= { ieee8021SrpReservationsEntry 7 }

    ieee8021SrpReservationStreamAge OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "seconds"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The number of seconds since the reservation was established
            on this port.  Talkers shall report this as the seconds
            since the first receipt of the Talker Advertise or Talker
```

```
            Failed.  Listeners shall report this as the number of
            seconds since the destination_address was first added to
            the Dynamic Reservations Entries."
        REFERENCE   "35.2.1.4c"
        ::= { ieee8021SrpReservationsEntry 8 }


    -- ==============================================================
    -- IEEE8021 SRP MIB - Conformance Information
    -- ==============================================================

    ieee8021SrpCompliances
        OBJECT IDENTIFIER ::= { ieee8021SrpConformance 1 }
    ieee8021SrpGroups
        OBJECT IDENTIFIER ::= { ieee8021SrpConformance 2 }

    -- ==============================================================
    -- units of conformance
    -- ==============================================================

    -- ==============================================================
    -- the ieee8021SrpConfiguration group
    -- ==============================================================

    ieee8021SrpConfigurationGroup OBJECT-GROUP
        OBJECTS {
            ieee8021SrpBridgeBaseMsrpEnabledStatus,
            ieee8021SrpBridgeBaseMsrpTalkerPruning,
            ieee8021SrpBridgeBaseMsrpMaxFanInPorts,
            ieee8021SrpBridgeBaseMsrpLatencyMaxFrameSize,
            ieee8021SrpBridgePortMsrpEnabledStatus,
            ieee8021SrpBridgePortMsrpFailedRegistrations,
            ieee8021SrpBridgePortMsrpLastPduOrigin,
            ieee8021SrpBridgePortSrPvid
        }
        STATUS      current
        DESCRIPTION
            "Objects that define configuration of SRP."
        ::= { ieee8021SrpGroups 1 }

    -- ==============================================================
    -- the ieee8021SrpLatency group
    -- ==============================================================

    ieee8021SrpLatencyGroup OBJECT-GROUP
        OBJECTS {
            ieee8021SrpPortTcLatency
        }
        STATUS      current
        DESCRIPTION
            "Objects that define latency for SRP."
        ::= { ieee8021SrpGroups 2 }

    -- ==============================================================
    -- the ieee8021SrpStreams group
    -- ==============================================================

    ieee8021SrpStreamsGroup OBJECT-GROUP
        OBJECTS {
```

```
            -- ieee8021SrpStreamId,
            ieee8021SrpStreamDestinationAddress,
            ieee8021SrpStreamVlanId,
            ieee8021SrpStreamTspecMaxFrameSize,
            ieee8021SrpStreamTspecMaxIntervalFrames,
            ieee8021SrpStreamDataFramePriority,
            ieee8021SrpStreamRank
        }
        STATUS      current
        DESCRIPTION
            "Objects that define Streams for SRP."
        ::= { ieee8021SrpGroups 3 }

    -- =============================================================
    -- the ieee8021SrpReservations group
    -- =============================================================

    ieee8021SrpReservationsGroup OBJECT-GROUP
        OBJECTS {
            -- ieee8021SrpReservationStreamId,
            -- ieee8021SrpReservationDirection,
            ieee8021SrpReservationDeclarationType,
            ieee8021SrpReservationAccumulatedLatency,
            ieee8021SrpReservationFailureBridgeId,
            ieee8021SrpReservationFailureCode,
            ieee8021SrpReservationDroppedStreamFrames,
            ieee8021SrpReservationStreamAge
        }
        STATUS      current
        DESCRIPTION
            "Objects that define Stream Reservations for SRP."
        ::= { ieee8021SrpGroups 4 }


    -- =============================================================
    -- compliance statements
    -- =============================================================

    ieee8021SrpCompliance MODULE-COMPLIANCE
        STATUS      current
        DESCRIPTION
            "The compliance statement for devices supporting
            Stream Reservation Protocol.

            Support of the objects defined in the IEEE8021-SRP MIB
            also requires support of the IEEE8021-BRIDGE-MIB; the
            provisions of 17.3.2 apply to implementations claiming
            support of the IEEE8021-SRP MIB."

        MODULE -- this module
            MANDATORY-GROUPS {
                ieee8021SrpConfigurationGroup,
                ieee8021SrpLatencyGroup,
                ieee8021SrpStreamsGroup,
                ieee8021SrpReservationsGroup
            }

        ::= { ieee8021SrpCompliances 1 }
```

END

34. Forwarding and queuing for time-sensitive streams

34.2 Detection of SRP domains

Change the following NOTE:

NOTE 1—SRP domain detection is based on the assumption that a device connected to a Port is either SRP capable for a given SR class, or is not SRP capable for that SR class. ~~The detection is based on current reservation activity~~SRP provides a boundary detection mechansim through the exchange of MSRPDUs; the boundary of a domain will therefore expand to include Ports as SRP attributes are declared. The position of the domain boundary has no effect on the transmission of SRP frames; rather, it reflects where SRP activity is occurring. Ports are removed from the SRP domain when they are removed from the active topology.

Insert the following Clause after Clause 34:

35. Stream Registration Protocol (SRP)

The Stream Reservation Protocol (SRP) utilizes three signaling protocols, MMRP (10.9), MVRP (11.) and MSRP (35.1) to establish stream reservations across a bridged network.

Within SRP the Multiple MAC Registration Protocol (MMRP) is optionally used to control the propagation of Talker registrations throughout the bridged network (35.2.4.3.1).

The Multiple VLAN Registration Protocol (MVRP) is used by end stations and Bridges to declare membership in a VLAN where a Stream is being sourced. This allows the Data Frame Priority (35.2.2.8.5(a)) to be propagated along the path from Talker to Listener(s) in tagged frames. MSRP will not allow Streams to be established across Bridge Ports that are members of the untagged set (8.8.2) for the related VLAN ID.

The Multiple Stream Registration Protocol (MSRP) is a signaling protocol that provides end stations with the ability to reserve network resources that will guarantee the transmission and reception of data streams across a network with the requested quality of service. These end stations are referred to as Talkers (devices that produce data streams) and Listeners (devices that consume data streams).

Talkers declare attributes that define the stream characteristics so Bridges have the necessary information available when they need to allocate resources for a Stream. Listeners declare attributes that request reception of those same streams. Bridges along the path from Talker to Listener process, possibly adjust, and then forward these MSRP attribute declarations. Bridges associate Talker and Listener attributes via the StreamID present in each of those attributes, which result in changes to the extended filtering services and allocation of internal resources when streams are “brought up”.

In order to establish the SRP domain boundaries, Bridges exchange SR class characteristics with each other and with end stations via MSRP. Neighboring devices that have identical SR class characteristics are considered to be in the same SRP domain and streams may be established between those devices.

MSRP provides a limited error reporting capability that is utilized when a Listener’s request to receive a stream cannot be honored because of some resource constraint within the network.

MSRP also supports the concept of data stream importance. For example, an emergency announcement would be flagged with a more important “rank” than a stream providing background music. This ranking ability allows the bridges to replace less important streams with more important streams without requiring intervention from the end stations.

There is a considerable body of experience in supplying data streams with guarantees for quality of service parameters such as latency, latency variation, or bandwidth. In particular, routers and hosts use the Internet Protocol and the Resource Reservation Protocol (RSVP, IETF RFC 2205 and RFC 2750) to achieve such guarantees. Supplying guarantees to a data stream requires two components:

a) A definition of the resources to be allocated and configured, by end stations and network nodes, for the support of a data stream; and
b) A protocol for end stations to signal to the network nodes their data streams’ requirements, for network nodes to distribute those requirements among each other, and for the network nodes to signal the success or failure of the attempt to reserve resources to support the guarantees.

RSVP supplies the signaling protocol for routers to support data streams in routed networks. This and the following clauses define a protocol to support data streams in bridged networks.

35.1 Multiple Stream Registration Protocol (MSRP)

MSRP supports the reservation of resources for streams, each destined for one or more Listeners, and each from a single source, across a bridged network. Transmitted data that conforms to a successful stream reservation will not be discarded by any Bridge due to congestion on a LAN. In order to propagate requests for reservations, MSRP defines an *MRP application* that provides the Stream resource registration service defined in 35.2.3. MSRP makes use of the MRP Attribute Declaration (MAD) function, which provides the common state machine descriptions defined for use in MRP-based applications. The MRP architecture, and MAD are defined in Clause 10. MSRP defines a new MRP Attribute Propagation (MAP) function, to provide an attribute propagation mechanism.

MSRP propagates registrations for stream reservations in a manner similar to the operation of MMRP (10.9) and MVRP (11.2), which are used for registering Group membership and individual MAC address information, and VLAN membership, respectively. Unlike MMRP and MVRP, however, the registered attributes can be combined, discarded, or otherwise altered, as they are propagated by the participating Bridges.

In order to make and keep quality of service guarantees all devices in a bridged network must participate in the signaling and queuing operations required of Bridges. For example, this would include IEEE Std 802.11 wireless media access points and stations. Thus, MSRP provides a means for Bridges or end stations running MSRP to cooperate both with higher network layers, such as routers or hosts running RSVP, and with lower network layers, such as wireless media.

MSRP is also responsible for establishing the SRP domain boundary for a particular SR class. All systems that support a particular SR class are in the same SRP domain if they use the same priority. An SRP domain boundary exists for an SR class when neighboring devices use different priorities for the SR class.

Figure 35-1 illustrates the architecture of MSRP in the case of a two-Port Bridge and an end station.

35.1.1 MSRP and Shared Media

Classic shared media, such as IEEE Std 802.3 half-duplex Carrier Sense Multiple Access with Collision Detect (CSMA/CD), cannot provide latency or bandwidth guarantees, because their operation depends on random timers. Such media are, therefore, not supported by MSRP.

There are other shared media where one node on the medium exercises control over access to the medium by the other nodes. For example, an IEEE Std 802.11 wireless medium has a single Access Point (AP) that controls access by the AP and the stations attached to the wireless medium so that some guarantee of latency and bandwidth can be made, subject to frame loss caused by data corruption errors. Similarly, an IEEE Std 802.3 Ethernet Passive Optical Network has a single Optical Line Terminal (OLT) that controls access to the optical medium by itself and some number of Optical Network Units (ONUs).

Different kinds of shared media use different techniques to allocate opportunities to transmit, and these techniques can have various dependencies on frame sizes, station-to-station vs. station-to-head data paths, or other factors. Rather than introducing the complexities of every such medium into MSRP, this standard takes advantage of the presence of a controlling entity to map medium-specific characteristics to the capabilities of MSRP.

MSRP defines and requires the existence of a Designated MSRP Node (DMN) on any shared medium. This DMN provides the MSRP services for the shared medium and determines each station’s ability to receive the

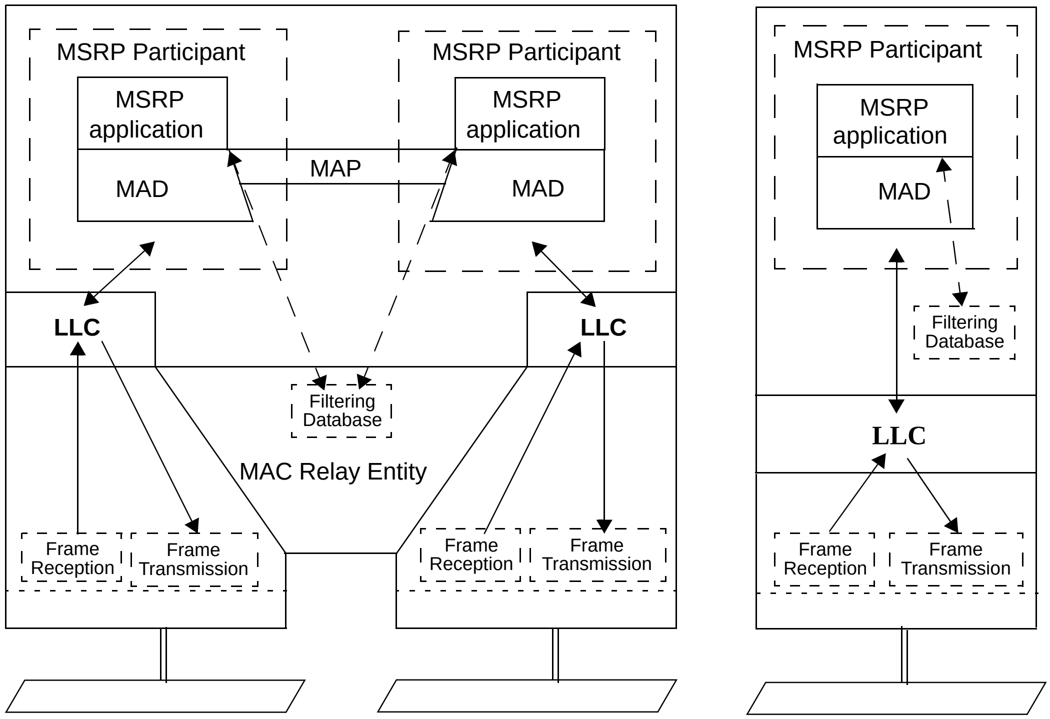

Figure 35-1—Operation of MSRP

MSRPDUs transmitted by other stations on the medium. A Non-DMN Port shall be configured to only process the MSRPDUs transmitted by DMN Ports, and ignore the MSRPDUs transmitted by other Non-DMN Ports. Furthermore, the DMN has the absolute control of the resource allocation on the shared medium. Given these two facts, a DMN can effectively control which reservations are and are not successful on the medium it controls.

Annex Q: DMN (Designated MSRP Node) Implementations provides examples on various shared media.

35.1.2 Behavior of end stations

35.1.2.1 Talkers

To announce the Streams that can be supplied and their characteristics, Talkers use the MAD_Join.request primitive (10.2) to make the Talker Declarations (35.2.1.3). To indicate the Streams that are no longer supplied, Talkers use the MAD_Leave.request primitive (10.2) to withdraw their Talker Declarations.

Talker Declarations are propagated by MSRP such that the Listeners and Bridges are aware of the presence of Talkers and the Streams that are offered. Talker Declarations are also used to gather QoS information along their paths. Based on the gathered QoS information, Talker Declarations are classified as follows:

a) **Talker Advertise:** An advertisement for a Stream that has not encountered any bandwidth or other network contraints along the network path from the Talker. Listeners that request attachment to this Stream are likely to create a reservation with the described QoS. A Talker Advertise will continue to be declared as long as the resources continue to be available.

b) **Talker Failed:** An advertisement for a Stream that is not available to the Listener because of bandwidth contraints or other limitations somewhere along the path from the Talker.

Talkers respond to the registration and de-registration events of Listener Declarations (35.2.1.3), signalled by MAD as follows:

On receipt of a MAD_Join.indication for a Listener Declaration, the Talker first merges (35.2.4.4.3) the Listener Declarations that it has registered for the same Stream. Then the Talker examines the StreamID (35.2.2.8.2) and Declaration Type (35.2.1.3) of the merged Listener Declaration. If the merged Listener Declaration is associated with a Stream that the Talker can supply, and the DeclarationType is either Ready or Ready Failed (i.e. one or more Listeners can receive the Stream), the Talker can start the transmission for this Stream immediately. If the merged Listener Declaration is an Asking Failed, the Talker shall stop the transmission for the Stream, if it is transmitting.

On receipt of a MAD_Leave.indication for a Listener Declaration, if the StreamID of the Declaration matches a Stream that the Talker is transmitting, then the Talker shall stop the transmission for this Stream, if it is transmitting.

35.1.2.2 Listeners

To indicate what Streams they want to receive, Listeners use the MAD_Join.request primitive (10.2) to make the Listener Declarations (35.2.1.3). To indicate the Streams that are no longer wanted, Listeners use the MAD_Leave.request primitive (10.2) to withdraw their Listener Declarations.

The Listener Declaration also conveys the results of the bandwidth and resource allocation along its path back to the Talker. Based on those results, Listener declarations are classified as follows:

a) **Listener Ready:** One or more Listeners are requesting attachment to the Stream. There is sufficient bandwidth and resources available along the path(s) back to the Talker for all Listeners to receive the Stream.

b) **Listener Ready Failed:** Two or more Listeners are requesting attachment to the Stream. At least one of those Listeners has sufficient bandwidth and resources along the path to receive the Stream, but one or more other Listeners are unable to receive the stream because of network bandwidth or resource allocation problems.

c) **Listener Asking Failed:** One or more Listeners are requesting attachment to the Stream. None of those Listeners are able to receive the Stream because of network bandwidth or resource allocation problems.

NOTE—The reader will notice that the Talker response to Ready and Ready Failed declarations is the same: the Talker can begin transmitting the Stream. Talkers might choose to pass the Ready Failed response to a higher layer protocol which could notify the user that the Stream is flowing, but not all Listeners are receiving it. It would be the responsibility of that higher layer to respond to this information as appropriate.

When there is a Talker Declaration registered on an interested Listener end station, the Listener shall create a Listener Declaration as follows:

If the Listener receives a Talker Advertise declaration, and the Listener is ready to receive the Stream, the Listener shall issue a Listener Ready declaration for the Stream. The Listener shall also issue an MVRP VLAN membership request for the vlan_identifier contained in the Talker Advertise DataFrameParameters (35.2.2.8.3(b)) so the neighboring bridge will add the associated Bridge Port to the member set for the VLAN.

If the Listener receives a Talker Failed declaration, and the Listener is ready to receive the Stream, the Listener shall issue a Listener Asking Failed declaration for the Stream.

There is no requirement for the order in which the Talker and Listener declarations, or VLAN membership request are communicated. The Listener Declaration can be made before the Listener receives an associated Talker Declaration, in which case the Listener shall issue a Listener Asking Failed declaration.

35.1.3 Behavior of Bridges

MSRP-aware Bridges register and de-register Talker and Listener declarations on the Bridge Ports according to the procedures defined in MRP (10.), and automatically generate de-registration of stale registrations. Any changes in the state of registration are processed by the MSRP Attribute Propagation (35.2.4) function, and disseminated in the network by making or withdrawing Talker and Listener declarations as defined in the Talker attribute propagation (35.2.4.3) and Listener attribute propagation (35.2.4.4).

In general, Talker declarations are propagated to all other Bridge Ports. There is a *talkerPruning* option (35.2.1.4(b)) that limits the scope of Talker declaration propagation. Listener declarations are only propagated to the Bridge Port with the associated Talker declaration (i.e. matching StreamID). If there is no associated Talker declaration registered on any Bridge Port then Listener declaration will not be propagated.

35.1.3.1 Blocked Declarations

For the purposes of MSRP Attribute Propagation (35.2.4), a Declaration is said to be “blocked” on a Bridge Port if the state of the Spanning Tree Instance identified by the vlan_identifier in the DataFrameParameters (35.2.2.8.3) of the Declaration, on that Bridge Port, has any value other than Forwarding. In a station’s Participant, no Declaration is ever blocked.

35.2 Definition of the MSRP application

MSRP maintains two categories of variables. The first category is used internally by the application state machines. These are defined in detail, below.

MSRP also defines another category of variables identified as MRP protocol elements that are communicated in MSRPDUs between stations on a network. These protocol elements include the MRP frame addressing and other fields defined in the MRP Protocol Data Units. The MSRP FirstValue fields, which are used to exchange the MSRP attributes, are also defined here.

35.2.1 Definition of internal state variables

The following variables and parameters are utilized by various state machines within MSRP:

a) Port Media Type (35.2.1.1);
b) Direction (35.2.1.2);
c) Declaration Type (35.2.1.3);
d) SRP parameters (35.2.1.4);

35.2.1.1 Port Media Type

MSRPDU processing on a port is handled differently depending on the type of medium the port is attached to. For example, the DMN on a shared medium port that receives MSRPDUs from one station shall update and retransmit those attributes so that all stations on that medium are updated appropriately. The possible values are:

a) **Access Control Port:** Transmitter controls access to the medium on which it is sending, so is either the DMN for a shared medium, or is a port on a full-duplex point-to-point medium;

b) **Non-DMN shared medium Port:** Transmitter is attached to a shared medium, but does not control access to the medium.

35.2.1.2 Direction

The Direction field is derived from the MSRP AttributeType definitions (35.2.2.4). The Direction indicates whether this is a Talker or a Listener MSRP Declaration, and takes one of two values:

a) **Talker:** MSRP AttributeType definitions of type Talker Advertise Vector Attribute Type (35.2.2.4(a)) or Talker Failed Vector Attribute Type (35.2.2.4(b)). Set Direction to zero for Talker attributes.

b) **Listener:** MSRP AttributeType definitions of type Listener Vector Attribute Type (35.2.2.4(c)). Set Direction to one for Listener attributes.

35.2.1.3 Declaration Type

The Declaration Type field is derived from the MSRP AttributeType definitions (35.2.2.4) and the MSRP FourPackedEvents (35.2.2.7.2). The Declaration Type indicates the specific type of the Talker or Listener MSRP Declaration.

For a Talker, the value of the Declaration Type component is either:

a) **Advertise:** MSRP AttributeType definitions of Talker Advertise Vector Attribute Type (35.2.2.4(a)).

b) **Failed:** MSRP AttributeType definitions of Talker Failed Vector Attribute Type (35.2.2.4(b)).

For a Listener, the value of the Declaration Type component is either:

c) **Asking Failed:** MSRP AttributeType definitions of Listener Vector Attribute Type (35.2.2.4(c)) with MSRP FourPackedType equal to Asking Failed (35.2.2.7.2(b)).

d) **Ready:** MSRP AttributeType definitions of Listener Vector Attribute Type with MSRP FourPackedType equal to Ready (35.2.2.7.2(c)).

e) **Ready Failed:** MSRP AttributeType definitions of Listener Vector Attribute Type with MSRP FourPackedType equal to Ready Failed (35.2.2.7.2(d)).

35.2.1.4 SRP parameters

The following parameters are used by SRP:

a) **portTcMaxLatency:** The maximum per-port per-traffic class latency, expressed in nanoseconds, a frame may experience through the underlying MAC service. There may be different latency numbers for different traffic classes on the same port.

b) ***talkerPruning:*** Enabling this parameter on the Bridge will limit the Talker declarations to ports that have the Streams destination_address (35.2.2.8.3(a)) in the MMRP MAC Address Registration Entries.

c) ***streamAge:*** A per-port per-stream 32-bit unsigned value used to represent the time, in seconds, since a stream's destination_address was first added to the Dynamic Reservations Entries (8.8(k)) for the associated port. This value is used when determining which streams have been configured the longest. Streams with a numerically larger *streamAge* are considered to be configured earlier than other streams, and therefore carry a higher implicit importance.

NOTE 1—A 32-bit unsigned value allows for expressing a streamAge of up to 136 years!

d) **msrpEnabledStatus:** MSRP shall have the ability to be enabled (true) or disabled (false) on a device. When MSRP is enabled on a device it shall cause a reset of all MSRP state machines on all ports. This affects the Applicant and Registrar state machines. The state of this parameter shall be persistent over power up restart/reboot.

e) **msrpPortEnabledStatus:** MSRP shall have the ability to be enabled (true) or disabled (false) on the ports of a device. When MSRP is enabled or disabled on a port of a device it shall cause MAP to be rerun on all MSRP enabled ports so existing attributes can be propagated to the port just enabled. This affects the Applicant and Registrar state machines. The state of this parameter shall be persistent over power up restart/reboot.

f) **msrpMaxFanInPorts:** The total number of ports on a Bridge that are allowed to establish reservations for inbound Streams. This number may be less than the total number of ports with msrpPortEnabledStatus set TRUE, which will result in lower maximum latency because of limits on the amount of possible interferring traffic. A value of zero (0) indicates no fan-in limit is being specified and calculations involving fan-in will only be limited by the number of MSRP enabled ports. Example calculations of delay associated with fan-in can be found in the paper “Calculating the Delay Added by Qav Stream Queue” [B29].

g) **msrpLatencyMaxFrameSize:** Calculation of the maximum latency through a bridge is in-part related to the maximum size of an interfering frame. The maximum size is defined to be 2000 octets by default. This parameter allows a smaller or larger value to be used in the latency calculations for the particular Bridge implementation. msrpLatencyMaxFrameSize does not imply any type of policing of frame size, it is only used in the latency calculations.

h) **SRPdomainBoundaryPort:** A per-port, per-SR class, boolean parameter that contains the value TRUE if the port is an SRP Domain Boundary Port, otherwise it contains the value FALSE. The parameter for a given SR class and Port shall be set to TRUE if any of the following conditions are met:

   1) The port is declaring an MSRP Domain attribute for that SR class, and the port has no MSRP Domain attribute registrations for that SR class, or;

   2) The port is declaring an MSRP Domain attribute for that SR class, and the port has at least one MSRP Domain attribute registration for that SR class with a different priority, or;

   3) One or more ports which support that SR class are declaring MSRP Domain attributes for that SR class, and this port does not support that SR class.

   In all other cases the parameter shall be set to FALSE.

i) **SR_PVID:** The Stream Reservation Port VLAN Identifier (SR_PVID) is a per-port parameter that contains the default VLAN ID for Stream related traffic. It shall contain a valid VID value (Table 9-2) and may be configured by management. If the value has not been explicitly configured, the SR_PVID shall assume the default SR_PVID defined in Table 9-2. This value is passed to the Talker via the SRclassVID (35.2.2.9.4) contained in the MSRP Domain attribute.

35.2.2 Definition of MRP protocol elements

35.2.2.1 MSRP application address

The group MAC address used as the destination address for MRPDUs destined for MSRP Participants shall be the group MAC address for “Individual LAN Scope group address, Nearest Bridge group address” as specified in Table 8-1, 8-2 and 8-3 (C-VLAN, S-VLAN and TPMR component Reserved addresses, respectively).

NOTE—Using this address will guarantee that MSRPDUs are never forwarded by an 802.1 Bridge, although MSRP aware Bridges do propagate the MSRP attributes.

35.2.2.2 MSRP application EtherType

The EtherType used for MRPDUs destined for MSRP Participants shall be the MSRP EtherType identified in Table 10-2.

35.2.2.3 MSRP ProtocolVersion

The ProtocolVersion for the version of MSRP defined in this standard takes the hexadecimal value 0x00.

35.2.2.4 MSRP AttributeType definitions

MSRP defines four AttributeTypes (10.8.2.2) that are carried in MRP protocol exchanges. The numeric values for the AttributeType are shown in Table 35-1 and their use is defined below:

a) **Talker Advertise Vector Attribute Type:** Attributes identified by the Talker Advertise Vector Attribute Type are instances of VectorAttributes (10.8.1), used to identify a sequence of values of Talker advertisements for related Streams that have not been constrained by insufficient bandwidth or resources.

b) **Talker Failed Vector Attribute Type:** Attributes identified by the Talker Failed Vector Attribute Type are instances of VectorAttributes, used to identify a sequence of values of Talker advertisements for related Streams that have been constrained by insufficient bandwidth or resources.

c) **Listener Vector Attribute Type:** Attributes identified by the Listener Vector Attribute Type are instances of VectorAttributes, used to identify a sequence of values of Listener requests for related Streams regardless of bandwidth constraints. Listener Vector Attribute Types are subdivided into individual Declaration Types via the MSRP FourPackedEvents (35.2.2.7.2).

d) **Domain Vector Attribute Type:** Attributes identified by the Domain Vector Attribute Type are instances of VectorAttributes, used to identify a sequence of values that describe the characteristics of an SR class.

Table 35-1—AttributeType Values

AttributeType	Value
Talker Advertise Vector	1
Talker Failed Vector	2
Listener Vector	3
Domain Vector	4

35.2.2.5 MSRP AttributeLength definitions

The AttributeLength field (10.12.1.8) in instances of the Talker Advertise Vector Attribute Type shall be encoded in MRPDUs (10.8) as an unsigned binary number, equal to the value shown in Table 35-2 (AttributeLength Values).

The AttributeLength field in instances of the Talker Failed Vector Attribute Type shall be encoded in MRPDUs as an unsigned binary number, equal to the value shown in Table 35-2.

The AttributeLength field in instances of the Listener Vector Attribute Type shall be encoded in MRPDUs as an unsigned binary number, equal to the value shown in Table 35-2.

The AttributeLength field in instances of the Domain Vector Attribute Type shall be encoded in MRPDUs as an unsigned binary number, equal to the value shown in Table 35-2.

Table 35-2—AttributeLength Values

AttributeType	Value
Talker Advertise Vector	25 (0x19)
Talker Failed Vector	34 (0x22)
Listener Vector	8
Domain Vector	4

35.2.2.6 MSRP AttributeListLength definitions

The AttributeListLength field (10.12.1.9) shall be encoded in MRPDUs (10.8) as an unsigned binary number, equal to the number of octets contained within the AttributeList. This field can be used when calculating the number of octets to skip to proceed to the next Message (or Message EndMark) in the MRPDU.

35.2.2.7 MSRP Vector definitions

35.2.2.7.1 MSRP ThreePackedEvents

The ThreePackedEvent vectors are encoded as defined in 10.8.2.10.1.

35.2.2.7.2 MSRP FourPackedEvents

MSRP FourPackedEvents are only used by the Listener Vector Attribute Type (35.2.2.4(c)). Within the FourPackedEvent, there are four possible values for the FourPackedType (10.8.2.10.2) as explained below. The numeric values for the MSRP FourPackedEvents are shown in Table 35-3.

a) **Ignore:** The StreamID referenced by FirstValue+n is not defined in this MSRPDU. When using this FourPackedType, the AttributeEvent (10.8.2.10.1) value, encoded in the ThreePackedEvent, shall be set to zero on transmit and ignored on receive

b) **Asking Failed:** The StreamID referenced by FirstValue+n has a declaration type of Listener Asking Failed.

c) **Ready:** The StreamID referenced by FirstValue+n has a declaration type of Listener Ready.

d) **Ready Failed:** The StreamID referenced by FirstValue+n has a declaration of type Listener Ready Failed.

Table 35-3—FourPackedEvent Values

FourPackedType	Value
Ignore	0
Asking Failed	1
Ready	2
Ready Failed	3

NOTE—In terms of efficent use of octets within an MSRP packet, placing nineteen (19) Ignores between two Listener declarations uses less octets than using two VectorAttributes to declare the Listener attributes separately. A single Listener VectorAttribute takes 12 octets, two attributes would take 24 octets. Declaring 21 attributes (two valid attributes with 19 Ignores inbetween) takes 12 octets, plus 6 additional ThreePackedEvents, plus 5 additional FourPackedEvents for a total of 23 octets.

35.2.2.8 MSRP FirstValue definitions (Stream reservations)

There are four Attribute Declarations defined for MSRP (35.2.2.4), three of which are related to stream reservations: Talker Advertise, Talker Failed, and Listener. The fourth attribute type, Domain, is used to discover the SRP domain, and is described in clause 35.2.2.9.

The Talker Advertise attribute contains all the characteristics that a Bridge needs in order to understand the resource requirements and importance of the referenced stream.

The Talker Failed attribute contains all the fields carried in the Talker Advertise Attribute, plus additional information regarding resource or bandwidth availability failures.

Listener attributes carry three subtypes: Ready, Ready Failed, and Asking Failed. These Listener subtypes are encoded in the FourPackedEvent (35.2.2.7.2).

FirstValue shall be incremented one or more times when NumberOfValues is greater than one. Incrementing FirstValue within MSRP is defined as follows:

a) Add 1 to Unique ID (35.2.2.8.2(b)), and
b) Add 1 to Stream destination_address (35.2.2.8.3(a))

The example shown in Table 35-4 illustrates the use of FirstValue and NumberOfValues within MSRP. This example shows four Streams (a, b, c and d) to be declared. In order to use the efficient packing techniques of MRP it would be preferable to assign these Streams sequential StreamIDs and destination_addresses as shown. Notice that StreamID yy-yy-yy-yy-yy-yy:00-04 is missing from this table (between Stream “c” and “d”). MSRP allows declaration of all four StreamIDs in a single VectorAttribute by setting NumberOfValues=5, with the StreamID = yy-yy-yy-yy-yy-yy:00-01 and a destination_address of xx-xx-xx-xx-xx-25. Setting the fourth FourPackedEvent to Ignore (35.2.2.7.2(a)) notifies MSRP that the StreamID between “c” and “d” is not in use.

The FirstValue field within MSRP is comprised of several components: StreamID (35.2.2.8.2), DataFrameParameters (35.2.2.8.3), TSpec (35.2.2.8.4), PriorityAndRank (35.2.2.8.5), Accumulated Latency (35.2.2.8.6), and FailureInformation (35.2.2.8.7). MSRP does not support changes in any of the FirstValue fields for an existing StreamID. If a Talker wishes to tear an old Stream down and bring a new Stream up, with a different FirstValue, utilizing the same StreamID, there must be at least two LeaveAllTime (Table 10-7) time periods between when the Talker removes the existing Stream registration and declares the new

Table 35-4—MSRP FirstValue NumberOfValues example

Stream	StreamID	destination_address
a	yy-yy-yy-yy-yy-yy:00-01	xx-xx-xx-xx-xx-25
b	yy-yy-yy-yy-yy-yy:00-02	xx-xx-xx-xx-xx-26
c	yy-yy-yy-yy-yy-yy:00-03	xx-xx-xx-xx-xx-27
d	yy-yy-yy-yy-yy-yy:00-05	xx-xx-xx-xx-xx-29

Stream. This guarantees that MRP has enough time to remove the current attribute from all devices in the network. If the new declaration were to occur too quickly the associated Streaming data could be corrupted because the filtering database may allow the new Stream data to start flowing while the old Stream bandwidth constraints are still configured. When the Bridge detects this occuring it will fail the Talker Advertise with the appropriate FailureInformation (35.2.2.8.7). Talkers may tear a Stream down and bring the same Stream back up immediately, as long as the FirstValue has not changed.

35.2.2.8.1 Structure definition

The FirstValue for Talker Advertise, Talker Failed and Listener attributes in MSRPDUs exchanged according to the protocol specified in this subclause shall have the following structure:

a) The first eight octets contain the *StreamID* (35.2.2.8.2).

This is the end of the Listener attribute. If this is a Talker Advertise or Talker Failed attribute continue as follows:

b) Following the StreamID are eight octets containing the *DataFrameParameters* (35.2.2.8.3).
c) Following the DataFrameParameters are four octets containing the *TSpec* (35.2.2.8.4).
d) Following the TSpec is one octet containing the *PriorityAndRank* (35.2.2.8.5).
e) Following the PriorityAndRank are four octets containing the *AccumulatedLatency* (35.2.2.8.6).

This is the end of the Talker Advertise attribute. If this is a Talker Failed attribute continue as follows:

f) Following the AccumulatedLatency are nine octets containing the *FailureInformation* (35.2.2.8.7).

The following partial BNF production gives the formal description of the MSRPDU FirstValue structure for the Talker Advertise attribute:

FirstValue ::= StreamID, DataFrameParameters, TSpec, PriorityAndRank, AccumulatedLatency

The following partial BNF production gives the formal description of the MSRPDU FirstValue structure for the Talker Failed attribute:

FirstValue ::= StreamID, DataFrameParameters, TSpec, PriorityAndRank, AccumulatedLatency,
FailureInformation

The following partial BNF production gives the formal description of the MSRPDU FirstValue structure for the Listener attribute:

FirstValue ::= StreamID

Figure 35-2 illustrates the structure of the MSRPDU FirstValue components. Each MSRP Attribute shall only use those structures as defined by the partial BNF productions described above. The octet #'s shown represent the octet location within the FirstValue field.

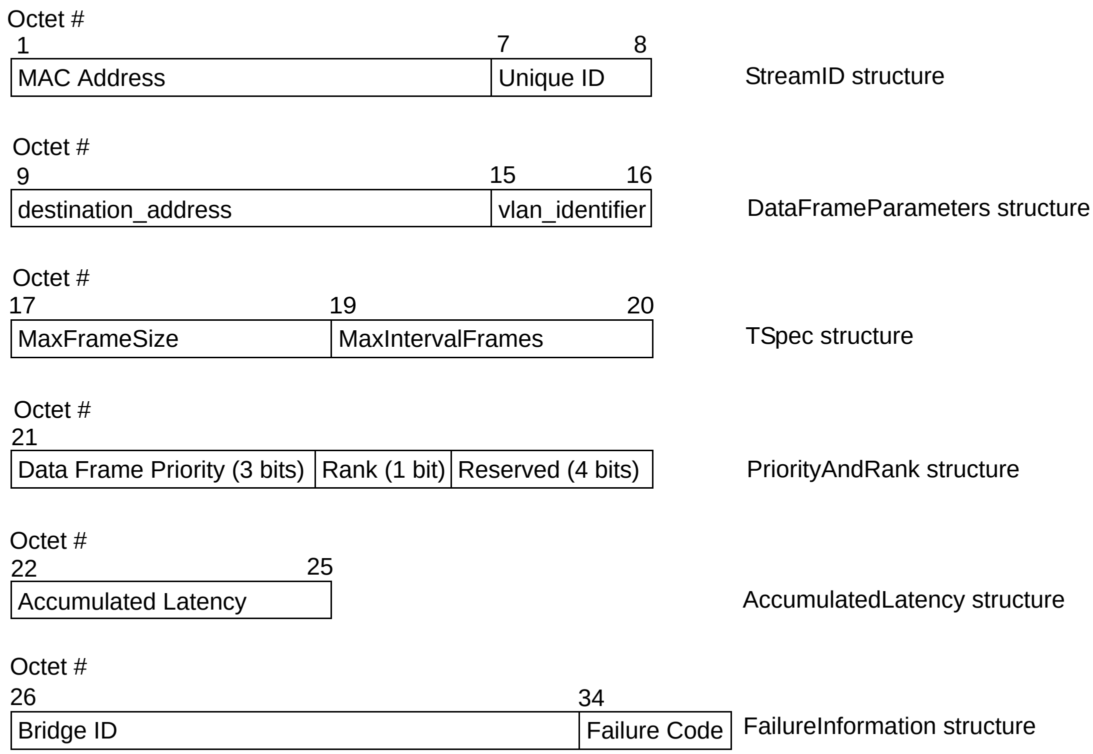

Figure 35-2—Format of the components of the reservation FirstValue fields

35.2.2.8.2 StreamID

The 64-bit StreamID is used to match Talker registrations with their corresponding Listener registrations (35.2.4). The StreamID comprises two subcomponents:

a) A 48-bit MAC Address associated with the System sourcing the stream to the bridged network. The entire range of 48-bit addresses are acceptable.

b) A 16-bit unsigned integer value, Unique ID, used to distinguish among multiple streams sourced by the same System.

StreamIDs are unique across the entire bridged network and are generated by the system offering the stream, or possibly a device controlling that system. A system reserving resources for more than one stream in the same bridged network shall use a StreamID that is unique among all StreamIDs in that bridged network. The combination of these two subcomponents ensure that such an assignment is possible.

NOTE 1—The Spanning Tree Protocol ensures that there can be at most one path from a Talker to its Listener(s). Multiple Declarations for the same StreamID can therefore occur briefly, during changes in the active topology of the bridged network.

NOTE 2—The MAC address component of the StreamID can, but does not necessarily, have the same value as the source_address parameter of any frame in the actual data stream.

35.2.2.8.3 DataFrameParameters

The DataFrameParameters component of the MSRP Attribute specifies the EISS parameters that are common to all frames belonging to the data stream for which this MAD is reserving resources. This information is used by Bridges to create Dynamic Reservation Entries (8.8(k)). The parameters are:

a) The destination_address; and
b) The vlan_identifier.

The destination_address specifies the destination MAC address of the streaming data packets. Only one Talker is allowed per destination_address. MSRP does not describe the actual streaming data, only the bandwidth associated with that stream.

Use of destination_address for both a Stream and “best effort” traffic (34.5) is outside the scope of SRP. SRP only supports destination_addresses that are multicast or locally administered addresses.

NOTE—MSRP enforces reserved bandwidth guarantees by filtering Stream destination addresses (35.2.4.4.2) for Streams that do not have a reservation. This blocks all traffic to that destination address. If that destination address was also being used for “best effort” traffic that device would no longer be reachable.

Systems that are not VLAN aware shall use the value SRclassVID (35.2.2.9.4) for the vlan_identifier in the DataFrameParameters. VLAN aware systems may use any valid VID (1 through 4094).

35.2.2.8.4 TSpec

The 32-bit TSpec component is the Traffic Specification associated with a Stream. It consists of the following two elements (which are encoded as described in 10.8.1.1):

a) **MaxFrameSize:** The 16-bit unsigned MaxFrameSize component is used to allocate resources and adjust queue selection parameters in order to supply the quality of service requested by an MSRP Talker Declaration. It represents the maximum frame size that the Talker will produce, excluding any overhead for media specific framing (e.g., preamble, IEEE Std 802.3 header, Priority/VID tag, CRC, interframe gap). As the Talker or Bridge determines the amount of bandwidth to reserve on the egress port it will calculate the media specific framing overhead on that port and add it to the number specified in the MaxFrameSize field.

b) **MaxIntervalFrames:** The 16-bit unsigned MaxIntervalFrames component is used to allocate resources and adjust queue selection parameters in order to supply the quality of service requested by an MSRP Talker Declaration. It represents the maximum number of frames that the Talker may transmit in one “class measurement interval” (34.4).

NOTE—Consider the example of a Class A 48kHz stereo audio stream encapsulated in an ethernet frame [B32]. The audio data within the frame would contain two sets of six 32-bit samples, plus a 32-octet header, for a total of 80 octets per frame sent once every class measurement interval (34.4). Therefore, MaxFrameSize=80, and MaxIntervalFrames=1. An IEEE Std 802.3 port on a Bridge would also add 42 octets of media specific framing overhead (8-octet preamble, 14-octet IEEE Std 802.3 header, 4-octet IEEE Std 802.1Q priority/VID Tag, 4-octet CRC, 12-octet IFG). When the Bridge calculates the amount of bandwidth to reserve it would combine 42 octets of media specific framing overhead with the MaxFrameSize of 80 octets, to arrive at a total frame size of 122 octets per class measurement interval. This represents a total bandwidth of approximately 7.7Mbit/second (122 octets * 8 bits per octet * 8000 frames per second).

Table 35-5 contains some examples of various forms of audio and video Streams with their associated TSpec components.

Table 35-5—TSpec Components Examples

Source	Raw Bit Rate	Media Specific Framing Overhead	TSpec MaxFrameSize	TSpec MaxIntervalFrames
48kHz stereo audio stream (32-bit samples) Class A [B32]	~3 Mbit/sec	~4.7 Mbit/sec	80	1 (8,000 frames/sec)
96kHz stereo audio stream (32-bit samples) Class A [B32]	~6 Mbit/sec	~4.7 Mbit/sec	128	1 (8,000 frames/sec)
MPEG2-TS video Class B [B33]	~24 MBit/sec	~2.5 Mbit/sec	786	1 (4,000 frames/sec)[1]
SD SDI (Level C) uncompressed Class A [B34]	270 Mbit/sec	~15 Mbit/sec	1442	3 (24,000 frames/sec)
SD SDI (Level D) uncompressed Class A [B34]	360 Mbit/sec	~20 Mbit/sec	1442	4 (32,000 frames/sec)
HD SDI 1080i uncompressed Class A [B35]	1.485 Gbit/sec	~80 Mbit/sec	1486	16 (128,000 frames/sec)
HD SDI 1080p uncompressed Class A [B36]	2.97 Gbit/sec	~160 Mbit/sec	1486	32 (256,000 frames/sec)

[1]The MPEG-2 TS entry in this table (third row) is shown as Class B traffic which runs at a default frame rate of 250 micro-seconds per frame, or 4,000 frames per second. Class A traffic runs at a default rate of 8,000 frames per second.

35.2.2.8.5 PriorityAndRank

a) **Data Frame Priority:** The 3-bit Data Frame Priority component specifies the priority value used to generate the Priority Code Point (PCP) the referenced data streams will be tagged with. It indicates the priority EISS or ISS parameter that will be used in all frames belonging to the data stream for which this MAD is reserving resources. This parameter determines which queue the frame is placed into on an output Bridge Port. In accordance with 10.8.1.1 these three bits are in the most significant positions (8, 7 and 6). The priority specified here is associated with the SR Classes as described in 34.5.

b) **Rank:** The single-bit Rank component is used by systems to decide which streams can and cannot be served, when the MSRP registrations exceed the capacity of a Port to carry the corresponding data streams. If a Bridge becomes oversubscribed (e.g., network reconfiguration, IEEE Std 802.11

bandwidth reduction) the Rank will also be used to help determine which Stream or Streams can be dropped. A lower numeric value is more important than a higher numeric value. In accordance with 10.8.1.1 this bit is in position 5.

For streams that carry emergency data such as North America 911 emergency services telephone calls, or fire safety announcements, the Rank shall be 0. Non-emergency traffic shall set this bit to a 1.

NOTE—It is expected that higher layer applications and protocols can use the Rank to indicate the relative importance of streams based on user preferences expressed by means beyond the scope of this standard. The values and defaults provided by this Standard are sufficient to order streams on a first-come-first-served basis, with special priority provided for emergency services.

c) **Reserved:** This 4-bit field shall be zero filled on transmit and ignored on receive. In accordance with 10.8.1.1 these four bits are in the least significant positions (4, 3, 2 and 1).

35.2.2.8.6 Accumulated Latency

The 32-bit unsigned Accumulated Latency component is used to determine the worst-case latency that a Stream can encounter in its path from the Talker to a given Listener. The latency reported here is not intended to increase during the life of the reservation. If some event occurs that would increase the latency beyond the original guarantee, MSRP will change the Talker Advertise to a Talker Failed and report Failure Code=7 (Table 35-6).

NOTE 1—An example of how latency could increase is if the speed of the underlying media where to decrease, such as one might see on a wireless link.

The initial value sent by the Talker is set to *portTcMaxLatency:* plus any amounts specified in the REGISTER_STREAM.request, and its value is increased by each Bridge as the Talker Declaration propagates through the network.

The portTcMaxLatency: per hop is equal to the sum of:

a) (equal or higher priority traffic) the time required to empty the queue in which frames of that priority are placed, if that queue and all higher priority queues are full;
b) (lower priority traffic) the time required to transfer one lower priority frame of maximum size that could have just started transmitting as the current priority frame was queued up;
c) (internal processing) the worst-case time required by the Bridge to transfer a received frame from the input port to the output queue; and
d) (wire propagation time) the time required for the first bit of the frame to propagate from the output port to the receiving device;
e) (media access delay) the time required to wait for the media to become available for transmission.

For item a) the total number of ports with msrpPortEnabledStatus set TRUE, and the msrpMaxFanInPorts will effect these calculations[1].

For item a) and item b) the maximum size of the interfering traffic is limited to msrpLatencyMaxFrameSize octets[2].

For item d, the propagation time, in the absence of better information, a value of 500 ns shall be used.

NOTE 2—This implies that no type of frame flow control can be used on the associated data stream packets.

[1]Example calculations for latency are contained in the paper “Calculating the Delay Added by Qav Stream Queue” [B29]
[2]Example calculations for latency are contained in the paper “Calculating the Delay Added by Qav Stream Queue” [B29]

The Listener can use this information to decide if the Latency is too large for acceptable presentation of the stream. The Accumulated Latency component is in units of nanoseconds.

35.2.2.8.7 FailureInformation

At the point when a Talker Advertise Declaration is transformed into a Talker Failed Declaration, the Bridge making the transformation adds information that indicates, to the Listeners registering the Talker Failed Declaration, the cause of the failure, and the identity of the Bridge and Bridge Port at which the failure occurred. The subcomponents of the FailureInformation include:

a) The Bridge ID (13(y)) of the Bridge that changed the Declaration Type from Advertise to Failed.
b) The Reservation Failure Code which is represented by a single octet containing the value shown in Table 35-6.

Table 35-6—Reservation Failure Codes

Failure Code	Description of cause
1	Insufficient bandwidth
2	Insufficient Bridge resources
3	Insufficient bandwidth for Traffic Class.
4	StreamID in use by another Talker
5	Stream destination_address already in use
6	Stream preempted by higher rank
7	Reported latency has changed
8	Egress port is not AVB capable[1]
9	Use a different destination_address (i.e. MAC DA hash table full)

Table 35-6—Reservation Failure Codes

Failure Code	Description of cause
10	Out of MSRP resources
11	Out of MMRP resources
12	Cannot store destination_address (i.e. Bridge is out of MAC DA resources)
13	Requested priority is not an SR Class (3.3) priority
14	MaxFrameSize (35.2.2.8.4(a)) is too large for media
15	msrpMaxFanInPorts (35.2.1.4(f)) limit has been reached
16	Changes in FirstValue for a registered StreamID.
17	VLAN is blocked on this egress port (Registration Forbidden)[2]
18	VLAN tagging is disabled on this egress port (untagged set)
19	SR class priority mismatch

[1]A device could choose to use the asCapable variable from P802.1AS [B30], clause 10.2.7.1, to help determine if its neighboring device is AVB capable. If the asCapable variable is FALSE for a particular port, then the neighboring device is not a time-aware system, and therefore not AVB capable.

[2]This Failure Code is never declared in a Talker Failed message since Talker attributes are not propagated on egress ports that have the associated VLAN blocked. The Bridge can still be queried by other means to learn why the Talker attribute was not declared.

35.2.2.9 MSRP FirstValue definitions (Domain discovery)

The Domain attribute contains all the information that a Bridge Port needs in order to determine the location of the SRP domain boundary (35.2.1.4(h)).

FirstValue shall be incremented one or more times when NumberOfValues is greater than one. Incrementing FirstValue for the MSRP Domain attribute is defined as follows:

a) Add 1 to SRclassID (35.2.2.9.2), and
b) Add 1 to SRclassPriority (35.2.2.9.3)

The choice of encoding and incrementing with this rule means that if one class (e.g. class B) is supported, then the FirstValue will be {5,2,VID} (class B, priority 2, and a VID) and the NumberOfValues field will be set to 1. If class A and class B are supported, with the default values, the FirstValue will again be {5,2,VID}, but the NumberOfValues fields will be set to 2. Applying the above incrementing rule to {5,2,VID} generates the value {6,3,VID}, i.e., class A, priority 3, and a VID, which is what we need for the default case.

35.2.2.9.1 Structure definition

The FirstValue for the Domain attribute in MSRPDUs exchanged according to the protocol specified in this subclause shall have the following structure:

a) The first octet contains the *SRclassID* (35.2.2.9.2).
b) Following the SRclassID is an octet containing the *SRclassPriority* (35.2.2.9.3).
c) Following the SRclassPriority are two octets containing the *SRclassVID* (35.2.2.9.4).

The following BNF production gives the formal description of the MSRPDU FirstValue structure for the Domain attribute:

FirstValue ::= SRclassID, SRclassPriority, SRclassVID

Figure 35-3 illustrates the structure of the MSRPDU FirstValue components for the Domain attribute.

Figure 35-3—Format of the components of the Domain FirstValue

35.2.2.9.2 SRclassID

SRclassID is a numeric representation of the SR classes that are supported by a particluar Bridge Port. The mapping for the first two SR classes is shown in Table 35-7.

Table 35-7—SR class ID

SR class	SR class ID
A	6
B	5

35.2.2.9.3 SRclassPriority

This fields holds the Data Frame Priority (35.2.2.8.5(a)) value that will be used for streams that belong to the associated SR class. End stations may initally declare the SR class default priority (Table 6-6) in order to learn the SRclassPriority that their neighboring system is declaring, and then use that learned value in subsequent declarations.

35.2.2.9.4 SRclassVID

This field contains the SR_PVID (35.2.1.4(i)) that the associated streams will be tagged with by the Talker. End stations may initally declare the default SR_PVID value (Table 9-2) in order to learn the SRclassVID that their neighboring system is declaring, and then use that learned value in subsequent declarations.

35.2.3 Provision and support of Stream registration service

35.2.3.1 Initiating MSRP registration and de-registration

MSRP utilizes the following five declaration types (35.2.1.3) to communicate: Talker Advertise, Talker Failed, Listener Ready, Listener Ready Failed and Listener Asking Failed.

An SR Station behaving as a Talker will send a Talker Advertise declaration to inform the network about the characteristics (35.2.2.8) of the Stream it can provide. Bridges register this declaration, update some of the information contained within the Talker declaration, and forward it out the non-blocked (35.1.3.1) ports on the Bridge. If *talkerPruning* is enabled and the Bridge has the Stream’s destination_address registered on one or more ports (via MMRP) it shall only forward the declarations out those ports. Eventually the Talker declaration will be registered by other SR stations.

If *talkerPruning* is enabled and the Stream’s destination_address is not registered on any ports the Talker declaration shall not be forwarded.

An SR Station behaving as a Listener will receive the Talker Advertise declaration and register it. If the Listener is interested in receiving that Stream it will send a Listener Ready declaration back towards the Talker. The Bridge’s MSRP MAP function will use the StreamID (35.2.2.8.2) to associate the Listener Ready with the Talker Advertise and forward the Listener Ready declaration only on the port that registered the Talker Advertise. This is referred to as Listener Pruning since the declarations are not forwarded out any other ports. The Bridge will also reserve the required bandwidth and configure its queues and the Filtering Database.

If any Bridge along the path from Talker to Listener does not have sufficient bandwidth or resources available its MSRP MAP function will change the Talker Advertise declaration to a Talker Failed declaration before forwarding it. Similarly a Listener Ready declaration will be changed to a Listener Asking Failed declaration if there is not sufficient bandwidth or resources available. This way both the Talker and Listener will know whether the reservation was successful or not.

In the case where there is a Talker attribute and Listener attribute(s) registered within a Bridge for a StreamID and a MAD_Leave.request is received for the Talker attribute, the Bridge shall act as a proxy for the Listener(s) and automatically generate a MAD_Leave.request back toward the Talker for those Listener attributes. This is a special case of the behavior described in 35.2.4.4.1.

Finally, there is the Listener Ready Failed declaration. This is used when there are two or more Listeners for a Stream. To simplify the explanation assume there are only two Listeners and one has sufficient bandwidth back to the Talker (signified by a Listener Ready), and the other does not (signified by a Listener Asking Failed). At some point in the network topology there will be a single Bridge that will receive the Listener Ready on one port from the first Listener and the Listener Asking Failed on another port from the second Listener. That Bridge’s MSRP MAP function will merge (35.2.4.4.3) those two declarations into a single Listener Ready Failed declaration that will be forwarded to the Talker. When the Talker receives the Listener Ready Failed it will know there are one or more Listeners that want the Stream and can receive it, and there are one or more Listeners that want the Stream but have insufficient bandwidth or resources somewhere along the path to receive it. The Talker may still send the Stream, but it will realize that not all Listeners are going to receive it.

MSRP provides a set of Service Primitives which control the declarations of the attributes defined above. The primitives associated with the Talker application entities are summarized in Table 35-8. Listener

Table 35-8—Summary of Talker primitives

Name	Request	Indication
REGISTER_STREAM	35.2.3.1.1	-
DEREGISTER_STREAM	35.2.3.1.3	-
REGISTER_ATTACH	-	35.2.3.1.6
DEREGISTER_ATTACH	-	35.2.3.1.8

application entities have a corresponding set of primitives summarized in Table 35-9.

Table 35-9—Summary of Listener primitives

Name	Request	Indication
REGISTER_STREAM	-	35.2.3.1.2
DEREGISTER_STREAM	-	35.2.3.1.4
REGISTER_ATTACH	35.2.3.1.5	-
DEREGISTER_ATTACH	35.2.3.1.7	-

35.2.3.1.1 REGISTER_STREAM.request

A Talker application entity shall issue a REGISTER_STREAM.request to the MSRP Participant to initiate the advertisement of an available Stream.

A Talker may choose to enter a non-zero value in the Accumulated Latency that indicates the amount of latency in nanoseconds that a Stream will encounter before being passed to the MAC service interface.

On receipt of a REGISTER_STREAM.request the MSRP Participant shall issue a MAD_Join.request service primitive (10.2, 10.3). The attribute_type (10.2) parameter of the request shall carry the value of Talker Advertise Vector Attribute Type (35.2.2.4(a)) or Talker Failed Vector Attribute Type (35.2.2.4(b)), depending on the Declaration Type. The attribute_value (10.2) parameter shall carry the values from the REGISTER_STREAM.request primitive.

```
REGISTER_STREAM.request    (
                            StreamID,
                            Declaration Type,
                            DataFrameParameters,
                            TSpec,
                            Data Frame Priority,
                            Rank,
```

Accumulated Latency,
FailureInformation
)

35.2.3.1.2 REGISTER_STREAM.indication

A REGISTER_STREAM.indication notifies the Listener application entity that the referenced Stream is being advertised by a Talker somewhere on the attached network.

On receipt of a MAD_Join.indication service primitive (10.2, 10.3) with an attribute_type of Talker Advertise Vector Attribute Type or Talker Failed Vector Attribute Type the MSRP application shall issue a REGISTER_STREAM.indication to the Listener application entity. The REGISTER_STREAM.indication shall carry the values from the attribute_value parameter.

REGISTER_STREAM.indication (
StreamID,
Declaration Type,
DataFrameParameters,
TSpec,
Data Frame Priority,
Rank,
Accumulated Latency,
FailureInformation
)

35.2.3.1.3 DEREGISTER_STREAM.request

A Talker application entity shall issue a DEREGISTER_STREAM.request to the MSRP Participant to remove the Talker’s advertisement declaration, and thus remove the advertisement of a Stream, from the network.

On receipt of a DEREGISTER_STREAM.request the MSRP Participant shall issue a MAD_Leave.request service primitive (10.2, 10.3) with the attribute_type set to the Declaration Type currently associated with the StreamID. The attribute_value parameter shall carry the StreamID and other values that were in the associated REGISTER_STREAM.request primitive.

DEREGISTER_STREAM.request (
StreamID
)

35.2.3.1.4 DEREGISTER_STREAM.indication

A DEREGISTER_STREAM.indication notifies the Listener application entity that the referenced Stream is no longer being advertised by a Talker.

On receipt of a MAD_Leave.indication service primitive (10.2, 10.3) with an attribute_type of Talker Advertise Vector Attribute Type or Talker Failed Vector Attribute Type the MSRP application shall issue a DEREGISTER_STREAM.indication to the Listener application entity.

DEREGISTER_STREAM.indication (
StreamID
)

35.2.3.1.5 REGISTER_ATTACH.request

A Listener application entity shall issue a REGISTER_ATTACH.request to the MSRP Participant to request attachment to the referenced Stream.

On receipt of a REGISTER_ATTACH.request the MSRP Participant shall issue a MAD_Join.request service primitive (10.2, 10.3). The attribute_type parameter of the request shall carry the value of Listener Vector Attribute Type (35.2.2.4(c)). The attribute_value shall contain the StreamID and the Declaration Type.

```
REGISTER_ATTACH.request     (
                            StreamID,
                            Declaration Type
                            )
```

35.2.3.1.6 REGISTER_ATTACH.indication

A REGISTER_ATTACH.indication notifies the Talker application entity that the referenced Stream is being requested by one or more Listeners.

On receipt of a MAD_Join.indication service primitive (10.2, 10.3) with an attribute_type of Listener Vector Attribute Type the MSRP application shall issue a REGISTER_ATTACH.indication to the Talker application entity. The REGISTER_ATTACH.indication shall carry the values from the attribute_value parameter.

```
REGISTER_ATTACH.indication  (
                            StreamID,
                            Declaration Type
                            )
```

35.2.3.1.7 DEREGISTER_ATTACH.request

A Listener application entity shall issue a DEREGISTER_ATTACH.request to the MSRP Participant to remove the request to attach to the referenced Stream.

On receipt of a DEREGISTER_ATTACH.request the MSRP Participant shall issue a MAD_Leave.request service primitive (10.2, 10.3) with the attribute_type set to the Listener Vector Attribute Type. The attribute_value parameter shall carry the StreamID and the Declaration Type currently associated with the StreamID.

```
DEREGISTER_ATTACH.request   (
                            StreamID
                            )
```

35.2.3.1.8 DEREGISTER_ATTACH.indication

A DEREGISTER_ATTACH.indication notifies the Talker application entity that the referenced Stream is no longer being requested by any Listeners.

On receipt of a MAD_Leave.indication service primitive (10.2, 10.3) with an attribute_type of Listener Vector Attribute Type the MSRP application shall issue a DEREGISTER_ATTACH.indication to the Talker application entity. The DEREGISTER_ATTACH.indication shall contain the StreamID.

```
DEREGISTER_ATTACH.indication        (
```

StreamID
)

35.2.4 MSRP Attribute Propagation

There is no MSRP MAP function for Domain attributes. MSRP simply declares the characteristics of the SR classes that are supported on the Bridge Port regardless of what has been learned from Domain registrations on other Bridge Ports.

For the Talker and Listener attributes MSRP propagates attributes in a manner different from that described in 10.3 for MMRP and MVRP. In principle, the MAP performs MSRP Attribute Propagation every time a MAD_Join.indication adds a new attribute to MAD (with the *new* parameter, 10.2, set to TRUE) or MAD_Leave.indication is issued by the MAD, and every time an internal application declaration or withdrawal is made in a station, and when the bandwidth of the underlying media changes (see *bandwidthAvailabilityChanged* notification in 34.3.2). MAP should also be run when a port becomes an SRP domain core port (3.5).

The MSRP MAP function is responsible for adjusting and propagating Talker and Listener attributes throughout the bridged network. It also updates the Dynamic Reservation Entries (8.8(k)) to specify which Streams shall be filtered and which shall be forwarded, along with updating the associated *streamAge* (35.2.1.4(c)). The bandwidth associated with those Streams is reported to the queuing algorithms via the *operIdleSlope(N)* parameter (34.3(d)) on a per-port per-Traffic Class basis.

Streams of higher importance are given available bandwidth before streams of lower importance.

If insufficient bandwidth or resources are available the streams destination_address will be filtered and the failure will be noted in the Talker and Listener attributes declared from that Bridge.

A port shall only forward MSRP declarations for SR classes it supports. This will eliminate unnecessary priority remapping for traffic related to unsupported SR classes.

The following subclauses describe what the MSRP MAP function shall accomplish.

35.2.4.1 Stream importance

MSRP utilizes the stream Rank (35.2.2.8.5(b)) to decide which stream is more important than another.

In the case where two streams have the same Rank, the *streamAge* will be compared. If the Ranks are identical and the *streamAge*s are identical the StreamIDs (35.2.2.8.2) will be compared and the numerically lower StreamID is considered to be more important.

35.2.4.2 Stream bandwidth calculations

As referenced in Table 35-5, the bandwidth requirements of a Stream include more than just the amount specified in the REGISTER_STREAM.request (35.2.3.1.1). SRP shall add the *perFrameOverhead* (34.4) associated with the media attached to the port.

If this port is on a Shared Media (35.1.1) the bandwidth requirements may need to be further increased. For example, a Stream transmitted from one station to another may have to be sent across the media twice. One time from the Talker station to the DMN, and a second time from the DMN to the Listener station. In this case the bandwidth requirements would need to be doubled.

The *totalFrameSize* for a stream on an outbound Port is therefore the sum of the following three amounts (doubled if each frame is transmitted on the media twice):

a) MaxFrameSize (35.2.2.8.4(a));
b) *perFrameOverhead* (34.4) associated with the media attached to the port;
c) one (1) additional octet to account for slight differences (up to 200 ppm) in the class measurement interval between neighboring devices.

Multiply *totalFrameSize* by MaxIntervalFrames (35.2.2.8.4(b)) to arrive at the associated bandwidth (in bits per second), which is then used to update *operIdleSlope(N)* as shown in Table 35-13.

Streams that are in the Listener Ready or Listener Ready Failed state reduce the amount of bandwidth available to other Streams. Streams that have no Listeners, or the Listeners are in the Asking Failed state, do not reduce available bandwidth for other Streams since the Stream data will not be flowing through this outbound port. These details are considered when calculating how many Streams can flow through a particular port. The total amount of bandwidth available to a particular Traffic Class on a port is represented by *deltaBandwidth(N)* (34.3(b1)). Subclause 34.3.1 describes the relationship of available bandwidth between Traffic Classes.

35.2.4.3 Talker attribute propagation

Table 35-10 describes the propagation of Talker attributes for a StreamID from one port of a Bridge to

Table 35-10—Talker Attribute Propagation per port

		Talker		
		(none)	**Advertise**	**Failed**
Listener	**(none)**	(none)	Talker Advertise or Talker Failed	Talker Failed
	Ready	(none)	Talker Advertise or Talker Failed	Talker Failed
	Ready Failed	(none)	Talker Advertise or Talker Failed	Talker Failed
	Asking Failed	(none)	Talker Advertise or Talker Failed	Talker Failed

another. If no Talker attributes are registered for a StreamID then no Talker attributes for that StreamID will be declared on any other port of the Bridge. If a Talker Failed is registered then it will be propagated as a Talker Failed out all other non-blocked (35.1.3.1) ports on the Bridge.

Talker Advertise registrations require further processing by the MAP function. MAP will analyze available bandwidth and other factors to determine if the outbound port has enough resources available to support the Stream. MAP will also verify msrpMaxFanInPorts, if non-zero, will not be exceeded. If there are sufficient resources and bandwidth available MAP will declare a Talker Advertise. Otherwise, MAP will declare a Talker Failed on the outbound port and add appropriate FailureInformation.

35.2.4.3.1 Talker Pruning

By default Talker declarations are sent out all non-blocked ports. If *talkerPruning* is enabled and the destination_address (35.2.2.8.3(a)) of the Stream is found in the MAC Address Registration Entries (8.8.4) for the port, the declaration shall be forwarded. If the destination_address is not found the declaration shall be blocked and no Talker declaration of any type shall be forwarded.

35.2.4.4 Listener attribute propagation

Listener Attributes, unlike Talker Attributes, can be merged (35.2.4.4.3) from several Listeners on different ports into a single Listener declaration. There are two steps involved:

1) Processing of incoming Listener attribute registration on a port based upon the status of the associated Talker attribute registration,
2) Merging all the individual ports Listener attributes gathered in step 1, above, into a single Listener attribute to be declared on the non-blocked (35.1.3.1) outbound port which has the associated Talker registration. If no Talker attribute is registered within the Bridge for the StreamID associated with the Listener, the Listener attribute will not be propagated.

35.2.4.4.1 Incoming Listener attribute processing

Table 35-11 describes how Listener attributes are propagated from the incoming ports.

If no Listener attributes are registered on a particular port then no Listener attribute will be propagated to the Listener attribute merging (35.2.4.4.3) from that port.

If a Listener Asking Failed is registered on a port then it will be propagated as a Listener Asking Failed and merged with other Listener Attributes from other ports.

If no Talker attributes are associated with the Listener attribute, the Listener attribute will not be propagated.

If a Talker Failed is registered on another port for the associated Listener Ready or Listener Ready Failed then a Listener Asking Failed will be propagated and merged with other Listener attributes from other ports.

If a Talker Advertise is registered on another port for the StreamID associated with a Listener Ready or

Table 35-11—Incoming Listener Attribute Propagation per port

		Talker		
		(none)	**Advertise**	**Failed**
Listener	**(none)**	(none)	(none)	(none)
	Ready	(none)	Listener Ready	Listener Asking Failed
	Ready Failed	(none)	Listener Ready Failed	Listener Asking Failed
	Asking Failed	(none)	Listener Asking Failed	Listener Asking Failed

Listener Ready Failed then that Listener attribute will be forwarded as-is and merged with Listener Attributes from other ports.

35.2.4.4.2 Updating Queuing and Forwarding information

When Incoming Listener attribute processing (35.2.4.4.1) has been completed for a port the Dynamic Reservation Entries (8.8(k)) shall be updated as shown in Table 35-12.

Table 35-12—Updating Dynamic Reservation Entries

		Talker		
		(none)	**Advertise**	**Failed**
Listener	**(none)**	(no entry)	Filtering	Filtering
	Ready	Filtering	Forwarding	Filtering
	Ready Failed	Filtering	Forwarding	Filtering
	Asking Failed	Filtering	Filtering	Filtering

The *operIdleSlope(N)* (34.3(d)) shall be updated as shown in Table 35-13, dependant on the change to the Dynamic Reservation Entries. MSRP MAP processing can occur at any time (35.2.4), resulting in a change to which Streams are filtered and which streams are forwarded. These changes in bandwidth requirements shall be reflected in the *operIdleSlope(N)* variable. Streams that have had their bandwidth removed shall decrease *operIdleSlope(N)*. Streams that have just been allocated bandwidth shall increase *operIdleSlope(N)*.

Table 35-13—Updating *operIdleSlope(N)*

		Dynamic Reservation Entries prior to MSRP MAP running		
		(none)	**Filtering**	**Forwarding**
Dynamic Reservation Entries after MSRP MAP running	**(none)**	(no change)	(no change)	Decrease *operIdleSlope(N)*
	Filtering	(no change)	(no change)	Decrease *operIdleSlope(N)*
	Forwarding	Increase *operIdleSlope(N)*	Increase *operIdleSlope(N)*	(no change)

NOTE—Care should be taken when updating the Dynamic Reservation Entries and the *operIdleSlope(N)* in order to not allow any associated streaming packets to be dropped. If the bandwidth utilization of a port is going to be increased (i.e. a Stream is going to be forwarded) the *operIdleSlope(N)* shall be updated before the Dynamic Reservation Entries. If the bandwidth utilization of a port is going to be decreased (i.e. a Stream is going to be filtered) the Dynamic Reservation Entries shall be updated before the *operIdleSlope(N)*.

35.2.4.4.3 Merge Listener Declarations

Listener Registrations with the same StreamID shall be merged into a single Listener Declaration that will be declared on the port with the associated Talker registration. If no such Talker registration exists the Listener attribute is not declared. When this Declaration is sent:

a) the Direction (35.2.1.2) is Listener;
b) the Declaration Type (35.2.1.3) is determined according to Table 35-14;
c) the StreamID (35.2.2.8.2) is that of the Listener Registrations.

Table 35-14—Listener Declaration Type Summation

First Declaration Type	Second Declaration Type	Resultant Declaration Type
Ready	none or Ready	Ready
	Ready Failed or Asking Failed	Ready Failed
Ready Failed	any	Ready Failed
Asking Failed	Ready or Ready Failed	Ready Failed
	none or Asking Failed	Asking Failed

35.2.4.5 MAP Context for MSRP

MSRPDUs can carry information about Streams in multiple VLANs, which in an MST environment, can be in different Spanning Tree Instances. Queue resources, however, are allocated and used according to priority parameters, not according to VLAN ID. Furthermore, on a shared medium, Streams can be using the shared medium even on VLANs that are blocked on the Bridge’s Port to that shared medium (e.g., consider an IEEE Std 802.11 AP that transmit packets between two stations that are on a VLAN that is blocked on the AP). Therefore there is a single context for MSRP attribute propagation that includes all Bridge Ports. The Declarations are filtered according to the state of the spanning tree, as described in 35.2.4.

All MSRPDUs sent and received by MSRP Participants in SST Bridges are transmitted as untagged frames.

35.2.5 Operational reporting and statistics

35.2.5.1 Dropped Stream Frame Counter

An implementation may support the ability to maintain a per-port per-traffic class count of data stream frames that are dropped for any reason. These are not MRP frames, but the data stream frames that flow between Talker and Listener(s) through the reservations established by MSRP.

35.2.6 Encoding

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 the Attribute Type (35.2.2.4) or FourPackedEvent (35.2.2.7.2) has changed, then the Bridge should behave as though an **rLv!** event (with immediate leavetimer expiration in the Registrar state table) was generated for the MAD in the Received MSRP Attribute Declarations before the **rJoinIn!** or

rJoinMt! event for the attribute in the received message is processed. This allows an Applicant to indicate a change in a stream reservation, e.g., a change from a Talker Failed to a Talker Advertise registration, without having to issue both a withdrawal of the old attribute, and a declaration of the new. A Listener attribute is also updated this way, for example, when changing from a Listener Ready to a Listener Ready Failed.

NOTE—This rule ensures that there is at most one Listener Declaration or one Talker Declaration for any given value of StreamID on any given port. In the unlikely situation where a Talker Advertise and a Talker Failed are received for the same Stream on the same port, the Talker Failed declaration takes precedence.

The following examples will help clarify the intent of this subclause. The first example describes the behavior when one attribute (Talker Advertise) is replaced by another (Talker Failed). The second example describes the behavior when the MSRP FourPackedEvents changes with a single Listener attribute.

This example illustrates processing of Talker Declaration changes. Assume a Bridge is receiving Talker Advertise declarations on an inbound Port. An emergency situation occurs and a 911 call is being placed via an entirely different Stream. The bandwidth that was available for the first Stream is now no longer available so the Bridge begins receiving Talker Failed declarations for that original stream. The MAP function realizes the Talker declaration has changed for the Stream and generate an internal leave event for the Talker Advertise and a join event for the Talker Failed. This behavior guarantees there will not be a declaration for a Talker Advertise and a Talker Failed for a single Stream existing within the Bridge at the same time.

As another example assume the same situation has occured as described in the example above (a 911 call). For this scenario consider the Listener declarations flowing in the opposite direction. When there was bandwidth available the Bridge was declaring a Listener Ready, which was then changed to a Listener Asking Failed as soon as the 911 call came through. The other Bridge receiving these Listener declarations realized that the Listener attribute MSRP FourPackedEvents had changed and acted as if the Listener declaration had been withdrawn and replaced by the updated Listener declaration.

35.2.7 Attribute value support requirements

Implementations of MSRP shall maintain state information for all attribute values that support the Stream registrations (35.2.2.8).

Implementations of MSRP shall be capable of supporting any attribute value in the range of possible values that can be registered using Stream registrations (35.2.2.8); however, the maximum number of attribute values for which the implementation is able to maintain current state information is an implementation decision, and may be different for Talker attributes and Listener attributes. The number of values that the implementation can support shall be stated in the PICS.

Annex A (normative)

PICS proforma - Bridge Implementations

A.5 Major capabilities

Insert the following row at the end of Table A.5:

Item	Feature	Status	Reference	Support
SRP	Does the implementation support the Stream Reservation Protocol?	O	35	Yes [ ] No [ ]

A.14 Bridge management

Insert the follow row at the end of Table A.14, re-numbering item MGT-98 if necessary:

Item	Feature	Status	Reference	Support
MGT-98	Does the implementation support the management entities defined in 12.22?	SRP: O	12.22	Yes [ ] N/A [ ]

A.24 Management Information Base (MIB)

Insert the following row at the end of Table A.24, re-numbering MIB-22 if necessary:

Item	Feature	Status	Reference	Support
MIB-22	Is the IEEE8021-SRP-MIB module fully supported (per its MODULE-COMPLIANCE)?	SRP: O	17.2, 35	Yes [ ] N/A [ ]

Insert new Table A.32 as shown:

A.32 Stream Reservation Protocol

Item	Feature	Status	Reference	Support
	If Stream Reservation Protocol (SRP in Table A.32) is not supported, mark N/A and ignore the remainder of this table.			N/A [ ]
SRP-1	Does the implementation support the exchange of MSRPDUs, using the generic MRPDU format defined in 10.8 to exchange MSRP-specific information, as defined in 35.2.2.8.1?	M	10.8, 35.2.2.8.1	Yes [ ]
SRP-2	Is the MSRP Application supported as defined in 35?	M	35	Yes [ ]
SRP-3	Does the implementation propagate registration information in accordance with the operation of MAP for the Base Spanning Tree Context, as specified in 35.2.4?	M	35.2.4	Yes [ ]
SRP-4	Does the implementation forward, filter or discard MAC frames carrying any MRP Application address as the destination MAC address in accordance with the requirements of 8.13.6?	M	8.13.6	Yes [ ]
SRP-5	Is the group MAC Address used as the destination address for MRPDUs destined for MSRP Participants the group MAC address identified in Tables 8-1, 8-2 and 8-3 as “Individual LAN Scope group address, Nearest Bridge group address”?	M	35.2.2.1, Table 8-1, Table 8-2, Table 8-3	Yes [ ]
SRP-6	Is the EtherType used for MRPDUs destined for MSRP Participants the MSRP EtherType identified in Table 10-2?	M	35.2.2.2, Table 10-2	Yes [ ]
SRP-7	Does the ProtocolVersion used for the implementation of MSRP take the hexadecimal value 0x00?	M	35.2.2.3	Yes [ ]
SRP-8	Are the Attribute Type values used in the implementation as specified in 35.2.2.4 and Table 35-1?	M	35.2.2.4, Table 35-1	Yes [ ]
SRP-9	Are the Attribute Length values used in the implementation as specified in 35.2.2.5 and Table 35-2?	M	35.2.2.5, Table 35-2	Yes [ ]

Item	Feature	Status	Reference	Support
SRP-10	Are the MSRP Vector Four-PackedEvents values used in the implementation as specified in 35.2.2.7.2 and Table 35-3?	M	35.2.2.7.2, Table 35-3	Yes [ ]
SRP-11	Does the implementation encode the values in FirstValue fields in accordance with the definition in 35.2.2.8?	M	35.2.2.8	Yes [ ]
SRP-12	Does the implementation update Accumulated Latency as the Talker attributes propagate through the Bridge?	M	35.2.2.8.6	Yes [ ]
SRP-13	Does the implementation update the Failure Information Bridge ID and Code in the event of insufficient bandwidth or resources through a Bridge?	M	35.2.2.8.7	Yes [ ]
SRP-14	Does the implementaiton propagate a Talker Advertise as a Talker Failed in the event of insufficient bandwidth or resources through a Bridge?	M	35.2.4.3, Table 35-9	Yes [ ]
SRP-15	Is talkerPruning and MMRP supported?	O	35.2.4.3.1	Yes [ ] No [ ]
SRP-16	Are Listener attributes merged and propagated as described in 35.2.4.4?	M	35.2.4.4, Table 35-10, Table 35-13	Yes [ ]
SRP-17	Does the implementation support updates to the Dynamic Reservation Entries as described in 35.2.4.4.2?	M	8.8k, 35.2.2.4.2, Table 35-11	Yes [ ]
SRP-18	Does the implementation support updates to operIdleSlope as defined in IEEE Std 802.1Qav?	M	35.2.4.4.3, Table 35-12	Yes [ ]
SRP-19	Does the implementation support the automatic modifications to Stream reservations as described in 35.2.5?	M	35.2.5	Yes [ ] No [ ]
SRP-20	Are MSRPDUs transmitted on all ports?	M	34.2	Yes [ ]
SRP-21	State the number of Talker registrations values that can be supported on each Port.	M	35.2.7	Number _____
SRP-22	State the number of Listener registrations values that can be supported on each Port.	M	35.2.7	Number _____
SRP-23	Does the device issue an appropriate MVRP VLAN membership request when attaching to or detaching from a Stream?	M	35.1.2.2	Yes [ ]

Item	Feature	Status	Reference	Support
SRP-24	Can the SR_PVID for any Port be assigned the value of the null VLAN ID or VLAN ID FFF (hexadecimal)?	X	35.2.1.4(i), Table 9-2	No [ ]
SRP-25	Does the device support changing the SR_PVID value from the default value shown in Table 9-2?	O	35.2.1.4(i), Table 9-2	Yes [ ] No [ ]
SRPMDCSN	Does this device support media dependendent Coordinated Shared Networking (CSN) functionality on one or more ports?	SRPMDMOCA:M OR: SRPMDDOT11:M	Q	Yes [ ] No [ ]
SRPMDMOCA	Does this device support media dependendent MoCA functionality on one or more ports?	O:1	Q.2	Yes [ ] No [ ]
SRPMDDOT11	Does this device support media dependendent IEEE Std 802.11 Access Point functionality on one or more ports?	O:1	Q.3	Yes [ ] No [ ]
SRPMDCSN-1	Does this device support a single Designated MSRP node (DMN)?	SRPMDCSN:M	35.1.1	Yes [ ]
SRPMDMOCA-1	Does this device support DMN Device Attribute Information Element to L2ME message?	SRPMDMOCA:M	Q.2.1.2	Yes [ ]
SRPMDMOCA-2	Does this device support DMN selection?	SRPMDMOCA:M	Q.2.1.3	Yes [ ]
SRPMDMOCA-3	Does this device support MSRP Attribute Declaration as specified in Table Q-2?	SRPMDMOCA:M	Q.2.2, Table Q-2	Yes [ ]
SRPMDDOT11-1	Does this device support EDCA-AC?	SRPMDDOT11:M	Table Q-6	Yes [ ]
SRPMDDOT11-2	Is the DMN and the QAP of the IEEE Std 802.11 BSS co-located in the same device?	SRPMDDOT11:M	Q.3.2	Yes [ ]
SRPMDDOT11-3	Does the device support MLME primitives specified in Table Q-5?	SRPMDDOT11:M	Q.3.3, Table Q-5	Yes [ ]
SRPMDDOT11-4	Does the device support VLAN tag encapsulation/de-encapsulation on the 802.11 interface?	SRPMDDOT11:M	Q.3.3.1	Yes [ ]
SRPMDDOT11-5	Is the reservation process an atomic operation?	SRPMDDOT11:M	Q.3.1, Figure Q-10, Figure Q-11, Figure Q-12,	Yes [ ]

Annex H (informative)

Bibliography

Insert the following references as appropriate.

[B32] IEEE P1722, Draft Standard for Layer 2 Transport Protocol for Time Sensitive Applications in Bridged Local Area Networks, 2008-08-25. See Figure 5.3 -- AVBTP common stream data header format. Also see IEC 61883-6, Consumer audio/video equipment - Digital Interface -- Part 6: Audio and music data transmission protocol, Second edition, 2005-10.

[B33] IEEE P1722, Draft Standard for Layer 2 Transport Protocol for Time Sensitive Applications in Bridged Local Area Networks, 2008-08-25. See Figure 5.3 -- AVBTP common stream data header format. Also see IEC 61883-4, Consumer audio/video equipment - Digital Interface - Part 4: MPEG2-TS data transmission, Second edition, 2004-08.

[B34] SMPTE 259M-2008, SMPTE Standard for Television - SDTV Digital Signal/Data -- Serial Digital Interface, Society of Motion Picture and Television Engineers, 2008. See clause 8.

[B35] SMPTE 292M-2008, SMPTE Standard 1.5 Gb/s Signal/Data Serial Interface, Society of Motion Picture and Television Engineers, 2008. See subclause 4.3.

[B36] SMPTE 424M-2008, SMPTE Standard for Television - 3 Gb/s Signal/Data Serial Interface, Society of Motion Picture and Television Engineers, 2008. See subclause 4.3.

[B37] MoCA MAC/PHY SPECIFICATION v1.0, MoCA-M/P-SPEC-V1.0-07122009, Multimedia over Coax Alliance (MoCA), July 12, 2009 (www.mocalliance.org).

[B38] MoCA MAC/PHY SPECIFICATION EXTENSIONS v1.1, MoCA-M/P-SPEC-V1.1-06162009, Multimedia over Coax Alliance (MoCA), June 16, 2009 (www.mocalliance.org).

[B39] MoCA MAC/PHY SPECIFICATION v2.0 - Draft (MoCA-M/P-SPEC-V2.0-MMDDYYYY), Multimedia over Coax Alliance (www.mocalliance.org).

Annex I (informative)

PICS proforma - End station implementations

I.5 Major capabilities

Change Table I-5 as shown (only the first 6 rows of the Table are shown):

Item	Feature	Status	Reference	Support	
MRPAP	Does the implementation support any MRP applications? If “No” is marked, continue at FQTSSE	O	5.12.1	Yes [ ]	No [ ]
MMRP	Is the operation of MMRP supported?	O.1	5.12.1, I.9	Yes [ ]	No [ ]
MVRP	Is automatic configuration and management of VLAN topology using MVRP supported?	O.1	5.12.1, I.7	Yes [ ]	No [ ]
MSRP	Is the operation of MSRP supported?	O.1	5.21.1, I.10	Yes [ ]	No [ ]
MRP	Is the Multiple Attribute Registration Protocol (MRP) implemented in support of MRP Applications?	M	10 I.9, I.7, I.8	Yes [ ]	
SPRU	Does the implementation support Source Pruning?	O	5.12, 10.10.3, 11.2.1.1	Yes [ ]	No [ ]
...	...	...	...	...	...

Insert the following clause after clause I.9.

I.10 SRP (Stream Reservation Protocol)

Item	Feature	Status	Reference	Support
	If SRP is not supported, mark N/A and ignore the remainder of this table.			N/A [ ]
SRP-1	Does the implementation support the exchange of MSRPDUs, using the generic MRPDU format defined in 10.8 to exchange MSRP-specific information, as defined in 35.2.2.8.1?	M	5.4.3, 10.8, 35	Yes [ ]
SRP-2	Is the MSRP Application supported as defined in 35?	M	5.4.3, 35	Yes [ ]
SRP-3	Is the group MAC Address used as the destination address for MRPDUs destined for MSRP Participants the group MAC address identified in Tables 8-1, 8-2 and 8-3 as “Individual LAN Scope group address, Nearest Bridge group address”?	M	35.2.2.1, Table 8-1, Table 8-2, Table 8-3	Yes [ ]
SRP-4	Is the EtherType used for MRPDUs destined for MSRP Participants the MSRP EtherType identified in Table 10-2?	M	35.2.2.2, Table 10-2	Yes [ ]
SRP-5	Does the ProtocolVersion used for the implementation of MSRP take the hexadecimal value 0x00?	M	35.2.2.3	Yes [ ]
SRP-6	Are the Attribute Type values used in the implementation as specified in 35.2.2.4 and Table 35-1?	M	35.2.2.5, Table 35-1	Yes [ ]
SRP-7	Are the Attribute Length values used in the implementation as specified in 35.2.2.5 and Table 35-2?	M	35.2.2.5, Table 35-2	Yes [ ]
SRP-8	Are the MSRP Vector FourPackedEvents values used in the implementation as specified in 35.2.2.7.2 and Table 35-3?	M	35.2.2.7.2, Table 35-3	Yes [ ]
SRP-9	Does the implementation encode the values in FirstValue fields in accordance with the definition in 35.2.2.8?	M	35.2.2.8	Yes [ ]
SRP-10	Does the Talker implementation populate the Accumulated Latency with a reasonable, non-zero value?	M	35.2.2.8.6	Yes [ ]

Item	Feature	Status	Reference	Support
SRP-11	Does the implementation update the Failure Information Bridge ID and Code in the event of insufficient bandwidth or resources through a Bridge?	M	35.2.2.8.7	Yes [ ]
SRP-12	Does the implementaiton create a Talker Failed in the event of insufficient bandwidth or resources through a Bridge?	M	35.2.4.3, Table 35-9	Yes [ ]
SRP-13	Is talkerPruning and MMRP supported?	O	35.2.4.3.1	Yes [ ] No [ ]
SRP-14	Are MSRPDUs transmitted on all ports?	M	34.2	Yes [ ]
SRP-15	State the number of Talker registration values that can be supported on each Port.	M	35.2.6	Number _____
SRP-16	State the number of Listener registration values that can be supported on each Port.	M	35.2.6	Number _____
SRP-17	Does the Listener issue an appropriate MVRP VLAN membership request when attaching to or detaching from a Stream?	M	35.1.2.2	Yes [ ]
SRP-18	Does the device support use of the SR class default priority to discover the SRclassPriority of the neighboring system?	O	35.2.2.9.3, Table 6-6	Yes [ ] No [ ]
SRP-19	Does the device support use of the default SR_PVID value to discover the SRclassVID of the neighboring system?	O	35.2.2.9.4, Table 9-2	Yes [ ] No [ ]
SRPMDCSN	Does this device support media dependendent Coordinated Shared Networking (CSN) functionality on one or more ports?	SRPMDMOCA:M OR: SRPMDDOT11:M	Q	Yes [ ] No [ ]
SRPMDMOCA	Does this device support media dependendent MoCA functionality on one or more ports?	O:1	Q.2	Yes [ ] No [ ]

Item	Feature	Status	Reference	Support
SRPMDDOT11	Does this device support media dependendent IEEE Std 802.11 Access Point functionality on one or more ports?	O:1	Q.3	Yes [ ] No [ ]
SRPMDCSN-1	Does this device support a single Designated MSRP node (DMN)?	SRPMDCSN:M	35.1.1	Yes [ ]
SRPMDMOCA-1	Does this device support DMN Device Attribute Information Element to L2ME message?	SRPMDMOCA:M	Q.2.1.2	Yes [ ]
SRPMDMOCA-2	Does this device support DMN selection?	SRPMDMOCA:M	Q.2.1.3	Yes [ ]
SRPMDMOCA-3	Does this device support MSRP Attribute Declaration as specified in Table Q-2?	SRPMDMOCA:M	Q.2.2, Table Q-2	Yes [ ]
SRPMDDOT11-1	Does this device support EDCA-AC?	SRPMDDOT11:M	Table Q-6	Yes [ ]
SRPMDDOT11-2	Is the DMN and the QAP of the IEEE Std 802.11 BSS co-located in the same device?	SRPMDDOT11:M	Q.3.2	Yes [ ]
SRPMDDOT11-3	Does the device support MLME primitives specified in Table Q-5?	SRPMDDOT11:M	Q.3.3, Table Q-5	Yes [ ]
SRPMDDOT11-4	Does the device support VLAN tag encapsulation/de-encapsulation on the 802.11 interface?	SRPMDDOT11:M	Q.3.3.1	Yes [ ]
SRPMDDOT11-5	Is the reservation process an atomic operation?	SRPMDDOT11:M	Q.3.1, Figure Q-10, Figure Q-11, Figure Q-12,	Yes [ ]

Annex Q (normative)

DMN (Designated MSRP Node) Implementations

This annex describes the DMN implementation on an IEEE Std 802.11 Network and Coordinated Shared Networks (CSNs)

Q.1 Designated MSRP nodes on CSNs

A CSN is a contention-free, time-division multiplexed-access network, supporting reserved bandwidth based on priority or flow (QoS). One of the nodes of the CSN acts as the Network Coordinator (NC) node, granting transmission opportunities to the other nodes of the network. The NC node also acts as the bandwidth resource manager of the network.

Q.1.1 Coordinated Shared Network (CSN) characteristics

CSNs support two types of transmissions: unicast transmission for node-to-node transmission and multicast/broadcast transmission for one-node-to-other/all-nodes transmission. Each node-to-node link has its own bandwidth characteristics which could change over time due to the periodic ranging of the link. The multicast/broadcast transmission characteristics are the lowest common characteristics of multiple/all the links of the network.

A CSN network is physically a shared network, in that a CSN node has a single physical port connected to the half-duplex medium, but is also a logically fully-connected one-hop mesh network, in that every node could transmit to every other node using its own profile over the shared medium.

Figure Q-1 illustrates a CSN network acting as a backbone interconnecting AV systems.

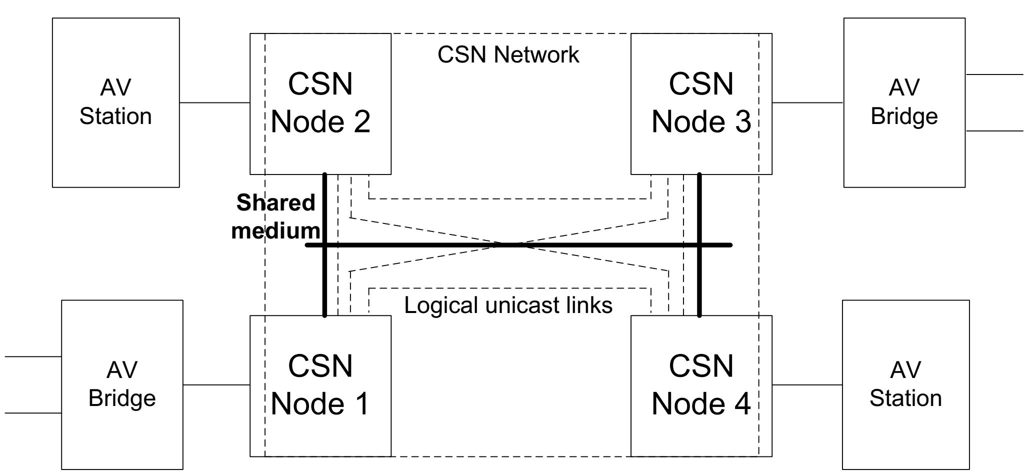

Figure Q-1—CSN Backbone

Depending on the CSN technology, the Network Coordinator node may either be a fixed node or may be dynamically selected during normal operation.

Q.1.2 Designated MSRP Node handling on CSN

From the bandwidth reservation stand point a CSN network is modeled as a Bridge as illustrated by Figure Q-2. Each node-to-node link is equivalent to the path from an input to an output Bridge's port.

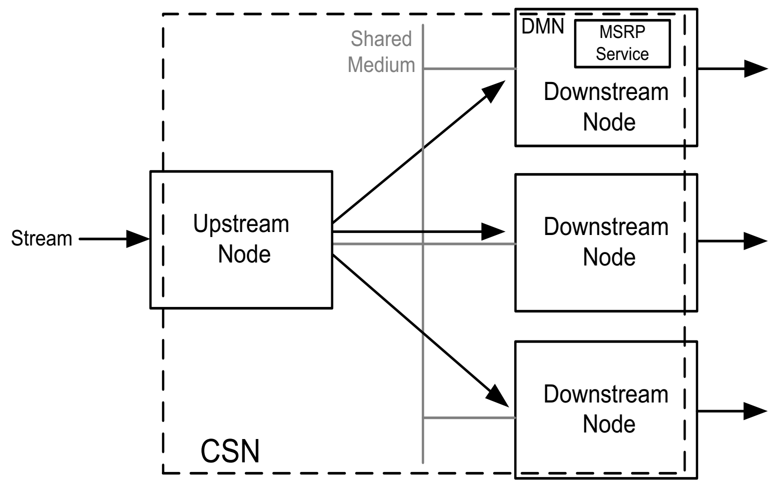

Figure Q-2—Bridge's CSN Model for Bandwidth Reservation

A CSN shall provide a single entity called the Designated MSRP Node or DMN (35.1.1) which communicates with the MSRP Service to manage the CSN bandwidth resources for the MSRP streams.

Q.1.2.1 DMN Selection and Migration

Depending on the CSN technology, the DMN might correspond to a static node or dynamically migrate between nodes during normal operation. The DMN selection is network specific and described in Q.2.1 and Q.3.2.

Over time the DMN constructs its database by handling the MSRP Talker and Listener Declarations generated by the nodes of the CSN. If the DMN migrates, the new DMN broadcasts an MRP LeaveAll message to all the nodes of the CSN which will force its neighbors to re-declare their attributes. The MSRP Participant nodes answer the MRP LeaveAll message by sending an MRP JoinIn message consumed by the DMN as an MRP Re-Declare! message. These re-declarations permit the new DMN to immediately build its database.

Q.1.3 MSRPDU handling on a CSN

Figure Q-3 and Figure Q-4 illustrate the flow and handling of MSRPDU messages on a CSN.

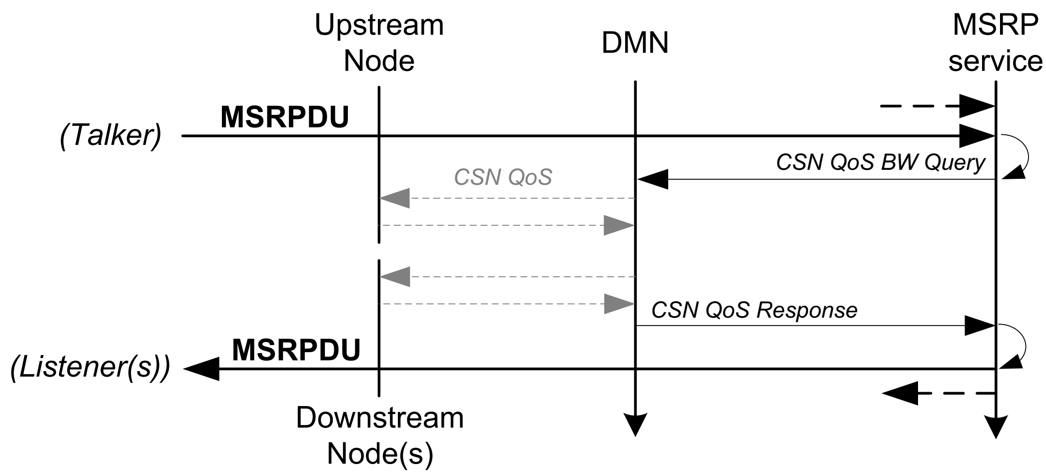

Figure Q-3—Talker MSRPDU flow

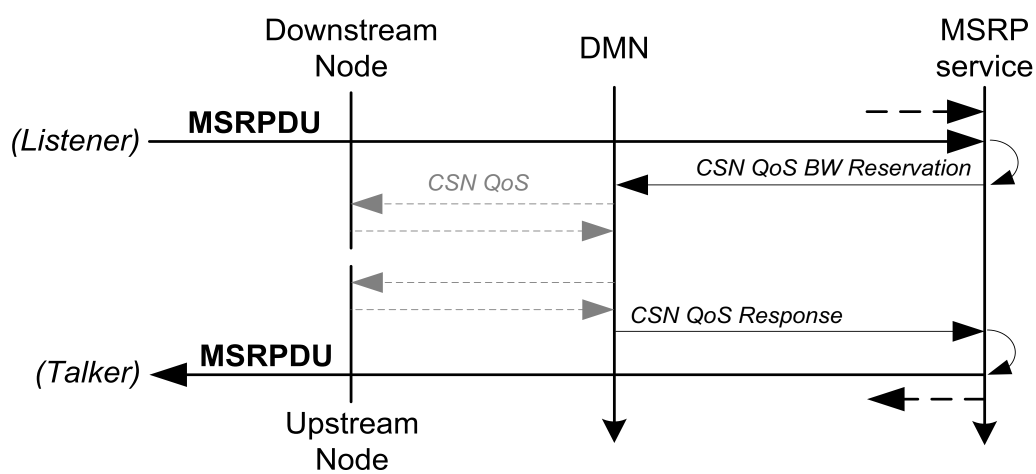

Figure Q-4—Listener MSRPDU flow

1) Non-DMN CSN nodes identify MSRPDUs by their Group Destination Address (35.2.2.1) and EtherType (35.2.2.2), and encapsulate them in regular CSN data frames, then send them, as appropriate, over the CSN.

2) The DMN delivers the MSRPDUs to the MSRP service.

3) The DMN translates the MSRP TSpec parameters into CSN QoS parameters and invokes the CSN's Protocol Specific QoS transactions with the CSN Network Coordinator (Q.2.2, Q.3.3) as follows:

   a) When the DMN receives a Talker Advertise message originated from an upstream CSN node, the DMN invokes a bandwidth query transaction with the CSN Network Coordinator to check whether or not the bandwidth advertised in the message's TSpec is available on each upstream to downstream node link of the CSN network. In addition the DMN maps the MSRP TSpec with the message's StreamID.

   b) When the DMN receives a Listener Ready message originated from a downstream CSN node, the DMN invokes a bandwidth reservation transaction with the CSN QoS manager to reserve the bandwidth associated with the message's StreamID on the downstream to upstream CSN node link.

4) After the DMN completes the CSN QoS transactions, the DMN behaves as an MSRP application on a Bridge and propagates MSRP attributes (35.2).

Q.1.4 CSN bandwidth fluctuations

Bandwidth on a CSN may fluctuate depending on interference experienced by the media. The MSRP Attribute Propagation (35.2.4) supports a *bandwidthAvailabilityChanged* notification when bandwidth increases or decreases across the media. When a bandwidth changed is encountered, MAP will reassess reservation requests to maintain appropriate bandwidth utilization.

Q.2 Designated MSRP Node on MoCA

NOTE—The discussion that follows is based on terminology found in the MoCA Specifications [B37][B38][B39].

Q.2.1 DMN Selection on MoCA Network

Q.2.1.1 DMN capable node discovery

A DMN capable node shall append the *IEEE DMN Device Attribute Information Element* (Q.2.1.2) to the L2ME payload of the Device Discovery Protocol SUBMIT L2ME transaction message specified in the MoCA v2.0 Specification [B39].

Upon completion of the L2ME Device Discovery transaction all the DMN-capable nodes of the MoCA network share the same information:

1) which MoCA nodes are DMN capable,
2) which MoCA node is selected as the DMN.

If no DMN is selected, the DMN selection shall be performed (Q.2.1.3).

Q.2.1.2 IEEE DMN Device Attribute IE

Q.2.1.2.1 General

The fields of the IEEE DMN Device Attribute IE are specified in Table Q-1 and Q.2.1.2.2 through Q.2.1.2.7. The general format of the Device Attribute Information Element is described in the MoCA v2.0 Specification [B39].

Bits (8 7 6 5 4 3 2 1)	Octets	Offset From Start of IE
ATTRIBUTE_ID	1	0
LENGTH	1	1
VENDOR_ID	2	2
TLV_TYPE	1	4
TLV_LENGTH	1	5
TLV_VALUE	2	6

Table Q-1—IEEE DMN Device Attribute IE

Q.2.1.2.2 ATTRIBUTE_ID (Enumeration8)

The value of the ATTRIBUTE_ID is 0xFF.

Q.2.1.2.3 LENGTH (UInteger8)

The value of the LENGTH is 1.

NOTE—The actual length of the Attribute IE in bits is (LENGTH + 1) * 32.

Q.2.1.2.4 VENDOR_ID (Enumeration16)

The value of VENDOR_ID is 0x0090 (IEEE 802.1 AVB).

Q.2.1.2.5 TLV_TYPE (Enumeration16)

The value of the TLV_TYPE is 1 (SRP).

Q.2.1.2.6 TLV_LENGTH (UInteger8)

The length of the TLV in octets is 4.

Q.2.1.2.7 TLV_VALUE (Octet2)

The meaning of the field of the TLV_VALUE are specified as described in Q.2.1.2.7.1 through Q.2.1.2.7.3.

Q.2.1.2.7.1 DMNcapable (Bit 0 - Boolean)

A value of 1 indicates the node is capable of acting as the DMN of the network. A value of 0 indicates the node is not capable to act as a DMN.

Q.2.1.2.7.2 DMNselected (Bit 1 - Boolean)

A value of 1 indicates the node has been selected as the DMN of the network. A value of 0 indicates the node is not the selected DMN.

Q.2.1.2.7.3 Reserved (Bit 2..15)

These bits are reserved for future usage.

Q.2.1.3 DMN selection and confirmation

If either 1) the *NODE_BITMASK* field into MAP frames indicates that the selected DMN has been removed from the network (due to failure, power state/down, etc.) or 2) the DMN node discovery (Q.2.1.1) does not indicate a DMN selected node, the DMN capable node with the lowest node ID will start acting as the DMN and confirm the selection to the other DMN capable nodes by generating a L2ME DMN Confirmation Transaction (Q.2.1.3.1).

NOTE—”MAP frame” is defined in the MoCA v1.0 Specification [B37].

Q.2.1.3.1 L2ME DMN Confirmation Transaction

Q.2.1.3.1.1 Overview of DMN ConfirmationTransaction

Figure Q-5 provides an overview of the signals exchanged among the nodes during an L2ME DMN Confirmation Transaction. As shown in the figure, the NC node starts the transaction either when it receives a Submit L2ME Frame from a node (called the DMN Selected Entry Node for that Transaction) or on its own. The transaction includes two L2ME Waves. The details of each message exchanged during the DMN Confirmation Transaction are provided below.

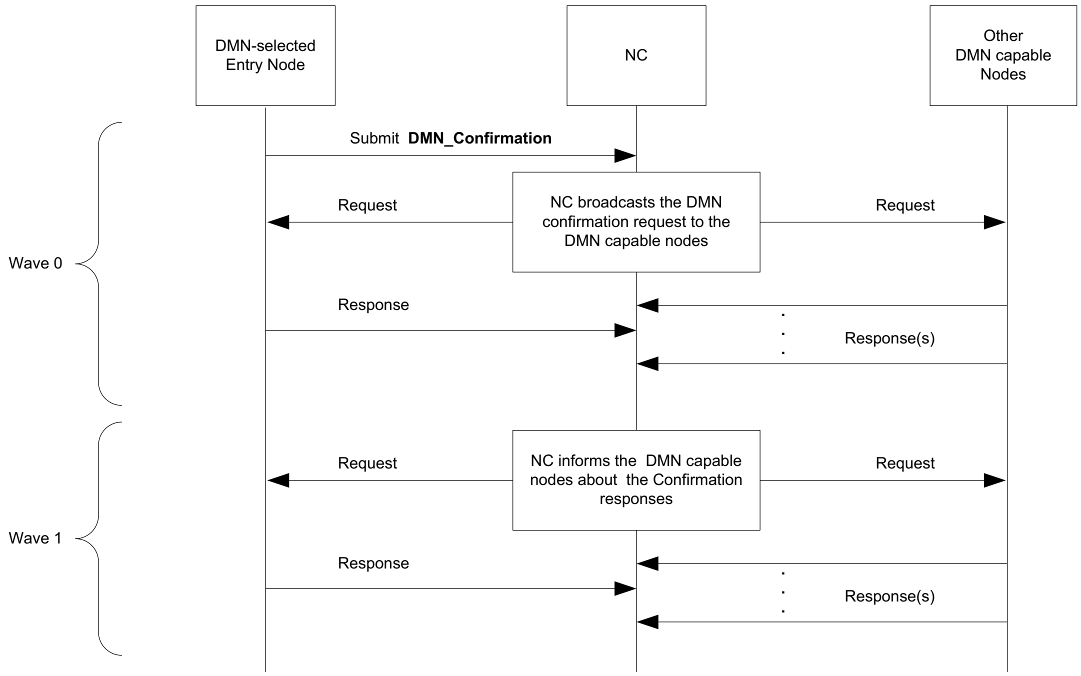

Figure Q-5—DMN ConfirmationTransaction

Q.2.1.3.1.2 DMN Confirmation Submit

The DMN Confirmation Transaction starts when the DMN Selected Entry Node sends a Submit L2ME Frame to the NC.

The general fields of the Submit L2ME frame are as specified in Section 2.2.3.1. of the MoCA v1.1 Specification [B38]. The parameters in the Submit L2ME Frame are set as follows:

a) VENDOR_ID (see Q.2.1.2.4)
b) TRANS_TYPE = 0x1 (SRP)
c) TRANS_SUBTYPE = 0x1 (DMN Confirmation)
d) WAVE0_NODEMASK = (DMN capable nodes)
e) MSG_PRIORITY = 0xF0
f) TXN_LAST_WAVE_NUM = 0x1
g) L2ME_PAYLOAD = 0 bytes

Q.2.1.3.1.3 Request L2ME Frame of Wave 0 of DMN Confirmation Transaction

In the Wave 0 that follows the DMN Confirmation Submit message, the NC node initiates a Request L2ME Frame based on the Submit L2ME Frame.

The general fields of the Resquest L2ME frame are as specified in the in Section 2.2.3.2. of the MoCA v1.1 Specification [B38].

Q.2.1.3.1.4 Response L2ME Frame of Wave 0 of DMN Confirmation Transaction

Each of the requested DMN capable nodes sends a Response L2ME Frame to acknowledge the confirmation request.

The general fields of the Response L2ME frame are as specified in Section 2.2.3.3. of the MoCA v1.1 Specification [B38]. The parameters in the Response L2ME Frame are set as follows:

a) RESP_STATUS <INTERPRETED> = ’1’
b) RESP_STATUS <IN_NEXT_WAVE> = ’1’
c) L2ME_PAYLOAD = 0 bytes

Q.2.1.3.1.5 Request L2ME Frame of Wave 1 of DMN Confirmation Transaction

In Wave 1, the NC node informs the DMN capable nodes about the acknowledgement results from Wave 0. The NC node initiates Wave 1 using a Request L2ME Frame with the “concatenated” type of L2ME_PAYLOAD.

The general fields of the Request L2ME frame are as specified in Section 2.2.3.1. of the MoCA v1.1 Specification [B38]. The “concatenated“ L2ME_Payload is as specified in Table 2-4 of the MoCA v1.1 Specification [B38].

Q.2.1.3.1.6 Response L2ME Frame of Wave 1 of DMN Confirmation Transaction

The DMN Confirmation Transaction is completed when the DMN Selected Entry node and the other DMN capable nodes send their final Response L2ME Frame to the NC node.

The general fields of the Response L2ME frame are as specified in Section 2.2.3.3. of the MoCA v1.1 Specification [B38]. The parameters in the Response L2ME Frame are set as follows:

a) RESP_STATUS <INTERPRETED> = ’1’
b) RESP_STATUS <IN_NEXT_WAVE> = ’0’
c) L2ME_PAYLOAD = 0 bytes

Q.2.2 MoCA network bandwidth management

The MSRP service within the MoCA network manages the MoCA bandwidth for the MSRP streams by invoking the MoCA native PQoS transactions. The DMN shall map the MSRP Attribute Declaration and the resultant MAD declarations as described in Table Q-2.

Table Q-3 describes the mapping between SRP TSpec components and MoCA PQoS TSPEC parameters.

Table Q-4 describes the mapping of the SRP StreamID to MoCA PQoS Flow transactions. The MoCA Flow parameters includes a 32-bit field that allows an application, like SRP, to store application specific information. SRP will use this field (FLOW_TAG) to store the 16-bit unique ID portion of the StreamID.

Q.3 Designated MSRP Nodes on IEEE Std 802.11 media

NOTE—Even though IEEE Std 802.11 is not a CSN, it uses the same DMN concepts described below.

From the bandwidth reservation standpoint an IEEE Std 802.11 BSS network is modeled as a Bridge as illustrated by Figure Q-6, Figure Q-7 and Figure Q-8. Each STA-AP link, STA-AP-STA link and optional STA-STA DLS direct link is equivalent to the path from an input to an output Bridge’s port.

Table Q-2—SRP to MoCA PQoS Transaction mapping

MSRP Attribute	MAD Primitive	MoCA PQoS Transactions	Description
Talker Advertise	MAD_Join.request(new)	Create PQoS Flow	Query bandwidth without reservation NOTE—MoCA PQoS APIs do not include a Bandwidth Query specific API. Therefore, bandwidth is queried by invoking a CreatePQoSFlow reservation for more bandwidth than the network could provide. The request will fail and CreatePQoSFlow will then return a failure status which includes the bandwidth available for reservations.
Listener Ready or Listener Ready Failed	MAD_Join.request(new)	Create PQoS Flow	Reserve bandwidth for a stream
Listener Ready or Listener Ready Failed	MAD_Join.request()	Update PQoS Flow	Renew the bandwidth reservation (leased time) for a stream
Talker or Listener Leave	MAD_Leave.request()	Delete PQoS Flow	Free bandwidth associated with a stream

Table Q-3—SRP TSpec to MoCA TSPEC mapping

SRP TSpec	MoCA PQoS TSPEC
MaxFrameSize	Max Packet Size
MaxFrameSize * MaxIntervalFrames	Peak Data Rate
MaxFrameSize * MaxIntervalFrames * Class B class measurement interval (34.4)	(Max) Burst Size

Table Q-4—SRP StreamID to MoCA PQoS Flow transaction mapping

SRP StreamID	MoCA PQoS Flow Transaction		
	Transaction	L2ME Payload	Field
48-bit MAC Address	Create PQoS Flow Update PQoS Flow	Submit	PACKET_DA
	Query PQoS Flow	Response	
16-bit Unique ID	Create PQoS Flow Update PQoS Flow	Submit	FLOW_TAG
	Query PQoS Flow	Response	

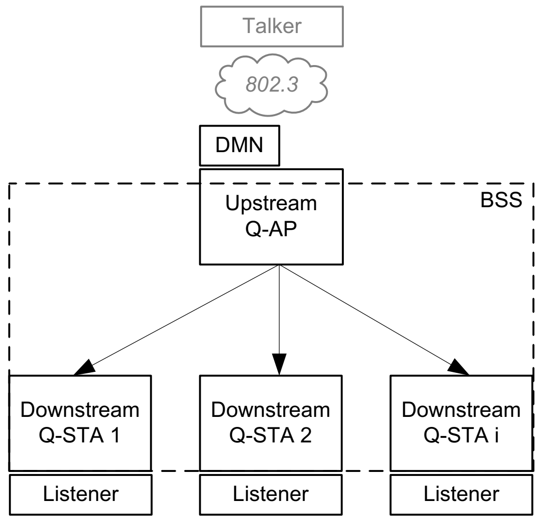

Figure Q-6—Bandwidth Reservation - Bridge Model for IEEE 802.11 BSS (STA Downstream Port)

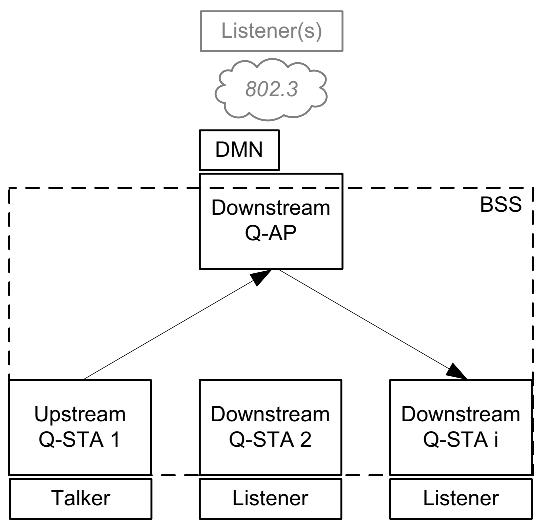

Figure Q-7—Bandwidth Reservation - Bridge Model for IEEE 802.11 BSS (STA Upstream Port)

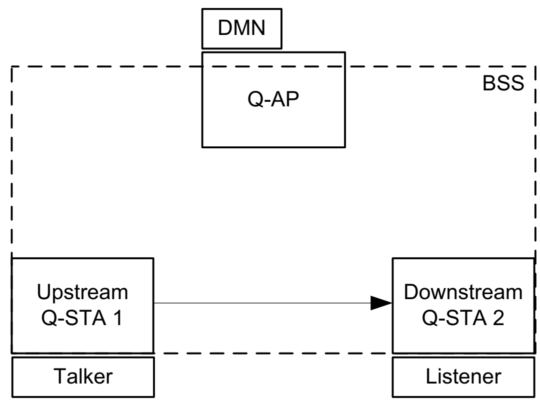

Figure Q-8—Bandwidth Reservation - Bridge Model for IEEE 802.11 BSS (Direct Link Setup)

An IEEE Std 802.11 BSS provides a single entity called the Designated MSRP Node (DMN) (35.1.1) to manage the BSS bandwidth resources for the MSRP streams.

Q.3.1 MSRP Handling

MSRPDUs are transparently transported by the Std 802.11 BSS network and delivered to the DMN.

The DMN maps the MSRP commands into IEEE Std 802.11 MLME TS commands and interacts with the AP through the AP's MLME SAP.

Figure Q-9 through Figure Q-12 describe the flow of information between the MSRP and IEEE Std 802.11 entities, and corresponding over the air IEEE Std 802.11 frames. Figure Q-9 is an example of the IEEE Std 802.11 bandwidth query process associated with an MSRP Talker Advertise.

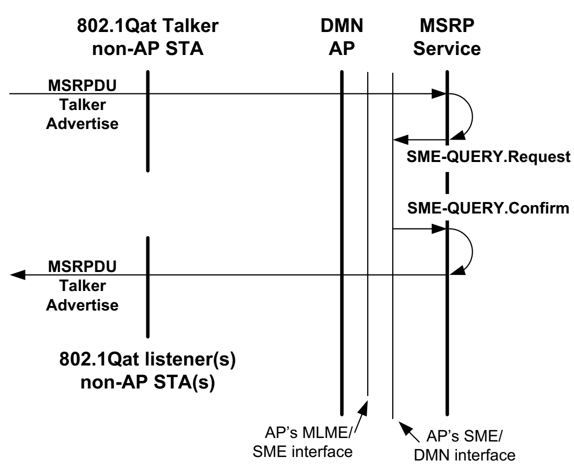

Figure Q-9—MSRP/802.11 Query Flows

Figure Q-10 is an example of the reservation process associated with a Listener non-AP STA requesting a Stream from the Talker non-AP STA. The diagram at the left of the figure shows the Listener non-AP STA sending a Listener Ready (A) through the AP's MSRP Service, which then propagates that Listener Ready (B) to the Talker non-AP STA. The message flow on the right of the figure shows the corresponding IEEE Std 802.11 message exchanges associated with the two Listener Readys (A & B).

There are two reservations (A & B) required to allow the Stream to flow from Talker to Listener. In Figure Q-10, Figure Q-11 and Figure Q-12 the reservation process must be an atomic operation so if either reservation fails then both reservations shall be removed and the MSRP attributes updated accordingly.

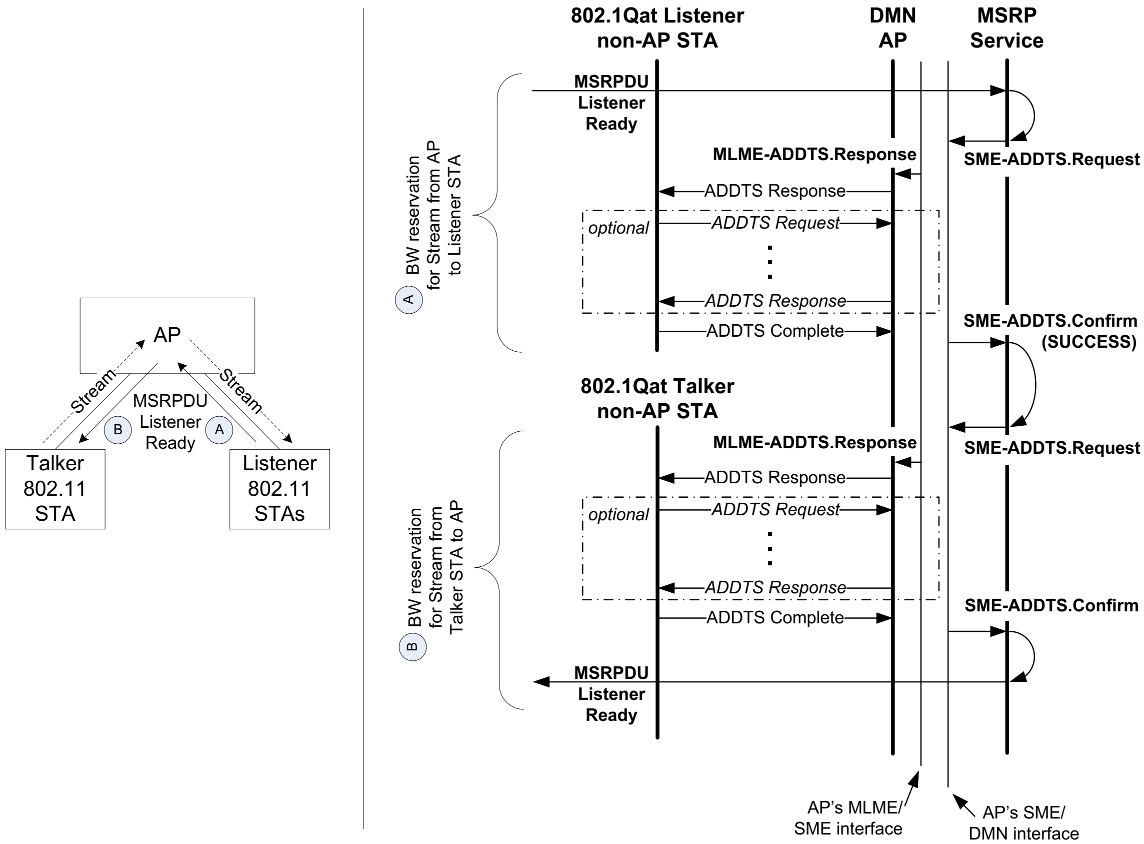

Figure Q-10—MSRP/802.11 Talker STA to Listener STA Reservation Flows

Figure Q-11 is an example of the reservation process associated with a “Bridged” Listener requesting a Stream from a Talker non-AP STA. The Listener sends a Listener Ready (A) from the “Bridged” side of the network and bandwidth is reserved as explained in SRP (35.). The IEEE Std 802.11 message exchanges associated with the Listener Ready (B) propagating to the Talker non-AP STA are also shown.

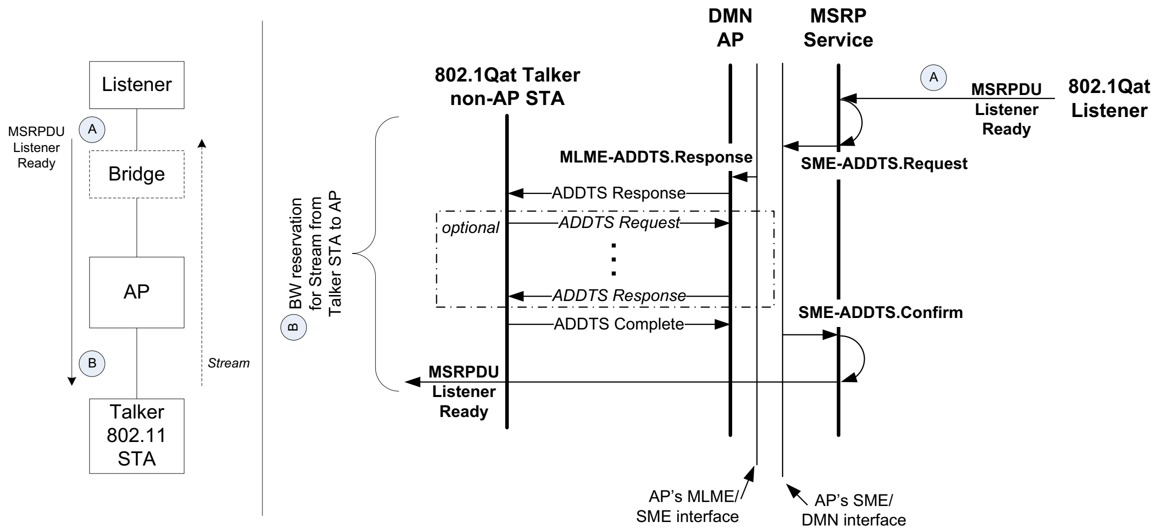

Figure Q-11—MSRP/802.11 “Bridged” Listener to Talker STA Reservation Flows

Figure Q-12 is an example of the reservation process associated with a Listener non-AP STA requesting a Stream from a “Bridged” Talker. The Listener non-AP STA sends a Listener Ready (A) through the AP’s MSRP Service, resulting in the IEEE Std 802.11 message exchanges shown. The reservation process associated with the Listener Ready (B) sent to the “Bridged” Talker is explained in SRP (35.).

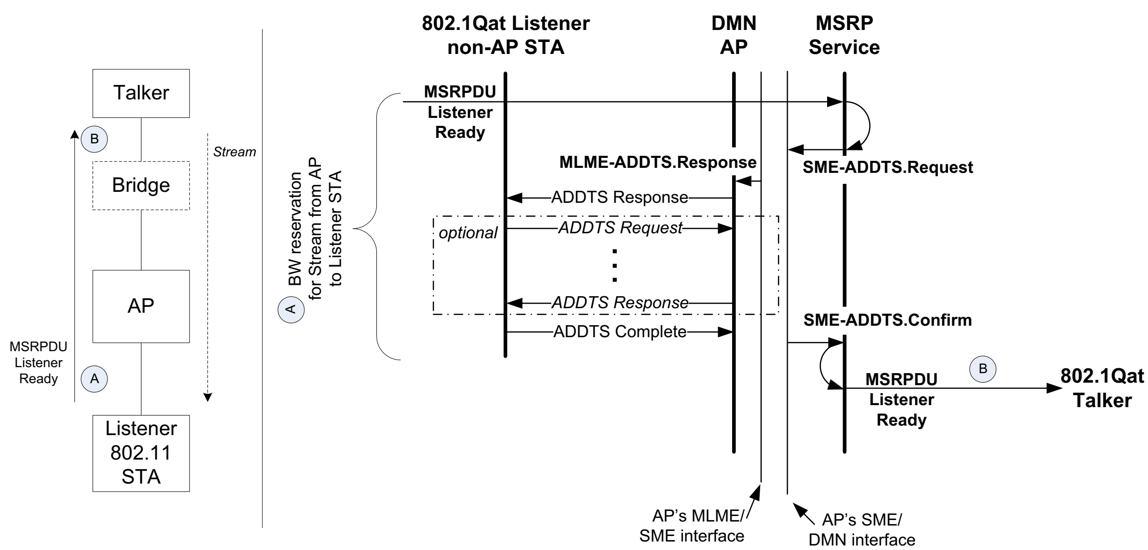

Figure Q-12—MSRP/802.11 Listener STA to “Bridged” Talker Reservation Flows

The IEEE Std 802.11 message exchanges shown in the preceeding three figures are explained in more detail here:

1) BBS nodes identify MSRPDUs by their Group Destination Address (35.2.2.1) and EtherType (35.2.2.2) and send these PDUs to the AP.
2) The AP forwards the MSRPDUs to the MSRP service on the DMN.
3) The DMN translates the MSRP TSpec parameters into an equivalent IEEE Std 802.11 TSPEC and invokes DMN-SME interface primitives with the AP as follows:
   a) When the DMN receives a Talker Advertise message originated from an upstream BSS node, the DMN invokes QoS Query transactions (SME-QUERY.Request) with the BSS QoS Manager to check whether or not the bandwidth advertised in the message's TSpec is available on each upstream to downstream node link of the BSS. In addition the DMN maps the MSRPDU's TSpec with the message's StreamID.
   b) When the DMN receives a Listener Ready message originated from a downstream BSS node, the DMN invokes a QoS Reservation transaction (SME-ADDTS.Request) with the BSS QoS manager to reserve the bandwidth associated with the message's StreamID on the downstream to upstream BSS node link.

   The IEEE Std 802.11 QoS AP on receipt of a SME-ADDTS.Request from the DMN shall make a determination about whether to accept the request or deny the request. The algorithm to be used by the QoS AP to make this determination is an implementation detail.

   If the QoS AP decides to accept the request, the AP shall derive a medium time value from the parameters specified in the SME-ADDTS.Request. The QoS AP shall then generate an autonomous ADDTS Response frame in which the medium time value is included and transmit it to the appropriate SRP Talker (BSS upstream) and Listener (BSS downstream) nodes.

   If the QoS AP decides to reject the request, it shall respond to the DMN with SME-ADDTS.confirm with a ResultCode of Rejected. The confirm primitive may also include a TSPEC which the QoS AP can accept, if specified in a subsequent SME-ADDTS.request.

NOTE—The TSPEC included in the SME-ADDTS.confirm is based on the result of the negotiations (labeled optional in Figure Q-10 through Figure Q-12) with the upstream BSS node. As a result the TSPEC included in the SME-ADDTS.confirm may be different from the one in the SME-ADDTS.request from the DMN.

4) After the DMN completes the BSS QoS transactions (SME-QUERY.Confirm or SME-ADDTS.Confirm as appropriate), the DMN behaves as an MSRP application on a Bridge and propagates MSRP attributes (35.2).

Q.3.2 BSS DMN selection

The DMN shall be located with the device that supports the QAP function in the BSS.

Q.3.3 BSS network bandwidth management

The MSRP service within the IEEE Std 802.11 network manages the BSS bandwidth for the MSRP streams by invoking the MLME QoS services. The DMN shall map the MLME services as described in Table Q-5.

Q.3.3.1 MSRPDU Encapsulation/De-encapsulation

In order to preserve the priority of an MSRPDU when traversing through a IEEE Std 802.11 network, the priority shall be encapsulated while the MSRPDU is in the IEEE Std 802.11 network and shall be de-encapsulated as it exits. See IEEE 802.11-2007 Annex-M for additional information.

Table Q-5—SRP to MLME QoS Services mapping

MSRP Attribute	MAD Primitive	SME QoS Services	Description
Talker Advertise	MAD_Join.request(new)	SME-QUERY	Query bandwidth without reservation
Listener Ready or Listener Ready Failed	MAD_Join.request(new)	SME-ADDTS	Reserve bandwidth for a stream
Listener Ready or Listener Ready Failed	MAD_Join.request()	SME-ADDTS[a]	Renew the bandwidth reservation (leased time) for a stream
Talker or Listener Leave	MAD_Leave.request()	SME-DELTS	Free bandwidth associated with a stream

[a]Bandwidth renewal is not required as long as the reservation is already established.

NOTE—For example if the User Priority of the MSRPDU is 4, CFI=0 and VLAN ID = 1893 the equivalent VLAN tag field (32 bits) is 81-00-87-65. When the frame enters the 802.11 network, the encapsulated 802.11 LLC header is AA-AA-03-00-00-00-81-00-87-65-AA-AA-03-00-00-00-08-00, where AA-AA-03-00-00-00-00-81-00-87-65 is the SNAP encoded VLAN header. When the frame exits the 802.11 network a de-encapsulation operation is performed and the resulting VLAN tag field is 81-00-87-65.

Q.3.3.2 QoS Maintenance Report

An SRP DMN may obtain QoS Maintenance Report using IEEE Std 802.11 Transmit Stream/Category Measurement Requests and processing the corresponding Transmit Stream/Category Measurement Reports. The Transmit Stream/Category Measurement Request is sent to both the SRP Talker (BSS upstream) and the SRP Listener (BSS downstream) nodes. Triggers are set on appropriate conditions such that Transmit Stream/Category Measurement Reports are generated only when predefined thresholds are breached. See IEE802.11-2007 Cl. 7.3.2.21.10 Transmit Stream/Category Measurement Request for details.

NOTE—This provides an indication to SRP (35.2.4) that bandwidth has changed.

Q.3.3.3 SRP TSpec to IEEE Std 802.11 TSPEC mapping

SR Class B traffic has three parameters associated with it, namely delay budget per IEEE Std 802.11 hop (20msecs), **MaxFrameSize** (<=1500 bytes) and **MaxIntervalFrames** (units of 4000 frames per second). IEEE Std 802.11 TSPECs include a Minimum PHY rate that is derived from the SR Class B parameters as described below:

1) Overhead = 10 byte VLAN tag + 8 byte Protocol definition = 18 bytes.
2) Mean Data Rate = (SRP TSpec **MaxFrameSize** + overhead) * 4000 * SRP TSpec **MaxIntervalFrames** bytes/sec.
3) The Mean Data Rate is also the Max Data Rate (since we assume MSDU size is fixed).
4) Assuming 70% * efficiency between the MAC and the PHY this translates into:
   (10/7) * (SRP TSpec **MaxFrameSize** + overhead) * 4000 * SRP TSpec **MaxIntervalFrames** bytes/sec, or
   (10/7) * 8 * (SRP TSpec **MaxFrameSize** + overhead) * 4000 * SRP TSpec **MaxIntervalFrames** bits/sec.
5) Minimum PHY Rate is therefore:
   (10/7) * 8 * (SRP TSpec **MaxFrameSize** + overhead) * 4000 * SRP TSpec **MaxIntervalFrames** bits/sec

NOTE—For example, with 1500 and 1 for **MaxFrameSize** and **MaxIntervalFrames** the above turns into 69.394285 Mbps. Or, with 64 and 1 for **MaxFrameSize** and **MaxIntervalFrames** the above turns into 3.748571 Mbps

Table Q-6 describes the mapping between SRP TSpec components and IEEE Std 802.11 QoS TSPEC parameters for the mandatory EDCA-AC mode.

Table Q-6—EDCA-AC for AV Streams

TSPEC Parameter		**SR Class B**
TSINFO	TID	5
	Direction	Up, Down
	Access Policy	10 (EDCA)
	ACK Policy	10 (No Ack) / 11 (Block Ack)
	APSD	0
	Aggregation	Yes
	Priority	**5**
Nominal MSDU Size[a]		**SRP TSpec MaxFrameSize + 18 (also set bit 15 to a 1)**
Maximum MSDU Size		**SRP TSpec MaxFrameSize + 18**
Peak Data Rate		**(SRP TSpec MaxFrameSize + 18) * 4000 * SRP TSpec MaxIntervalFrames**
Minimum PHY Rate[b]		**(10/7) * 8 * (SRP TSpec MaxFrameSize + 18) * 4000 * SRP TSpec MaxIntervalFrames**
Delay Bound		**20 msecs**
Surplus Bandwidth Allowance[c]		1.2

[a]bit15 set to indicate that the MSDU size is fixed
[b]Assuming 70% efficiency between the MAC and the PHY
[c]20% surplus allocation

Table Q-7 describes the mapping between SRP **TSpec** components and IEEE Std 802.11 QoS TSPEC parameters for the optional HCCA mode.

Table Q-7—HCCA for AV Streams

TSPEC Parameter		SR Class B
TSINFO	TID	5
	Direction	Up, Down
	Access Policy	01 (HCCA)
	ACK Policy	10 (No Ack) / 11 (Block Ack)
	APSD	0
	Aggregation	Yes
	Priority	**5**
Nominal MSDU Size[a]		**0**
Maximum MSDU Size		**SRP TSpec MaxFrameSize + 18**
Minimum Service Interval		10 msec
Maximum Service Interval		10 msec
Inactivity Interval		0
Minimum Data Rate		0
Mean Data Rate		**0**
Maximum Burst Size		**(SRP TSpec MaxFrameSize + 18) * 4000 * SRP TSpec MaxIntervalFrames * 10^{-2}**
Minimum PHY Rate[b]		**(10/7) * 8 * (SRP TSpec MaxFrameSize + 18) * 4000 * SRP TSpec MaxIntervalFrames**
Peak Data Rate		**(SRP TSpec MaxFrameSize + 18) * 4000 * SRP TSpec MaxIntervalFrames**
Delay Bound		20 msecs
Surplus Bandwidth Allowance[c]		1.2

[a]bit15 set to indicate that the MSDU size is fixed.
[b]Assuming 70% efficiency between the MAC and the PHY
[c]20% surplus allocation