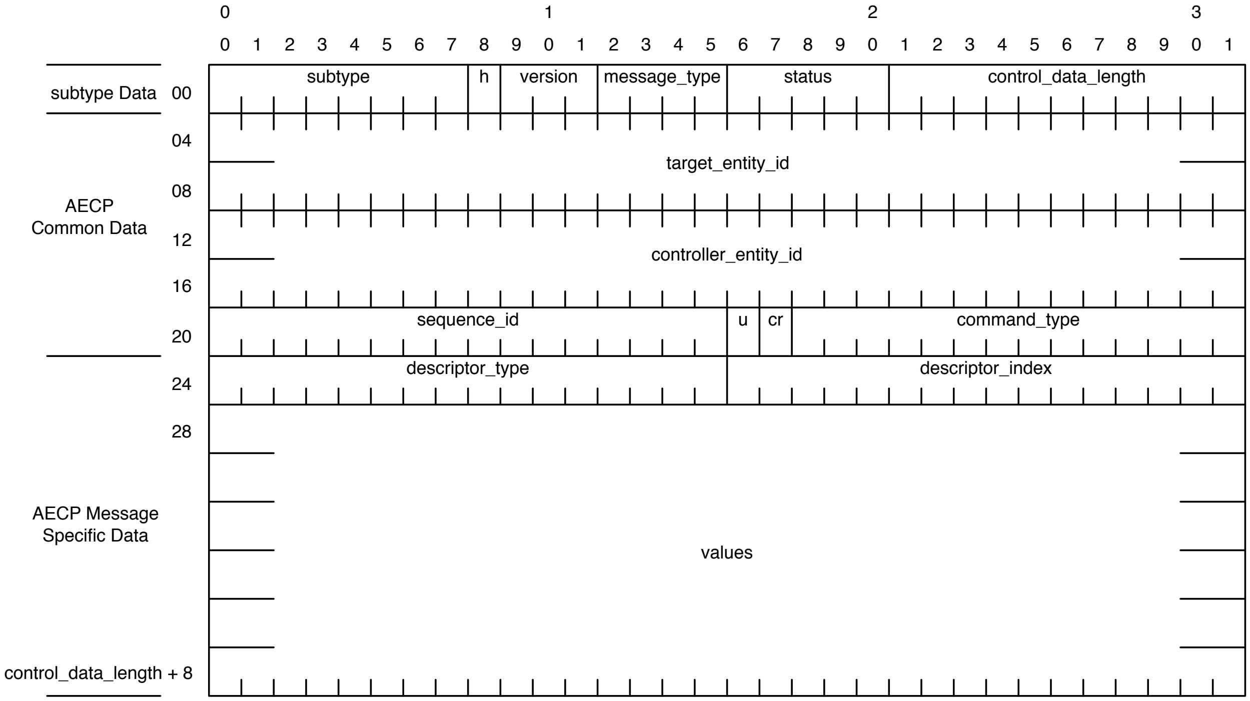

Figure 7-54—**SET_MIXER Command and Response and GET_MIXER Response Format**

The **command_type** field is set to SET_MIXER.

The **descriptor_type** and **descriptor_index** fields are set to the descriptor type and index of the mixer for which the current values are being set. **descriptor_type** is set to MIXER.

The **values** field is set to the new values to be set. The contents of this field are dependent on the control being addressed. The MIXER descriptor with the given index determines how this field is formatted based on the **control_value_type** and **number_of_sources** fields. The **values** field only conveys the current value and not the max, min, default and step. The response always contains the current value, that is it contains the new value if the command succeeds or the old value if it fails.

7.4.31.2. Restrictions

If the ATDECC Entity has been locked or acquired by another ATDECC Controller then the ATDECC Entity responds with an ENTITY_LOCKED or ENTITY_ACQUIRED status response.

The SET_MIXER command may be protected by authentication. If this is the case then the ATDECC Entity responds with a NOT_AUTHENTICATED status response to an unauthenticated ATDECC Controller.

7.4.32. GET_MIXER Command

The GET_MIXER command is used to get the current values of a mixer.

The GET_MIXER command returns the values for the currently active configuration.

7.4.32.1. Command Format

The GET_MIXER command uses the AECPDU format as shown in Figure 7-55.

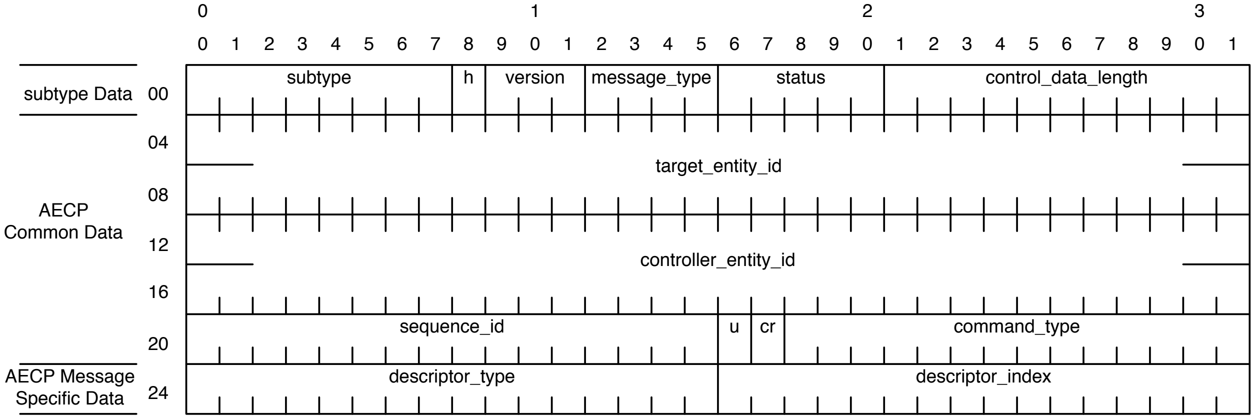

Figure 7-55—**GET_MIXER Command Format**

The **command_type** field is set to GET_MIXER.

The **descriptor_type** and **descriptor_index** fields are set to the descriptor type and index of the mixer for which the current values is being fetched. **descriptor_type** is set to MIXER.

7.4.32.2. Response Format

The GET_MIXER response uses the AECPDU format as shown in Figure 7-54.

The **command_type** field is set to GET_MIXER.

The **descriptor_type** and **descriptor_index** fields are set to the descriptor type and index of the mixer for which the current values is being fetched. **descriptor_type** is set to MIXER.

The **values** field is set to the current values. The contents of this field are dependent on the control being addressed. The MIXER descriptor with the given index determines how this field is formatted based on the **control_value_type** and **number_of_sources** fields. The **values** field only conveys the current value and not the max, min, default and step.

7.4.32.3. Restrictions

The GET_MIXER command may be protected by authentication. If this is the case then the ATDECC Entity responds with a NOT_AUTHENTICATED status response to an unauthenticated ATDECC Controller.

7.4.33. SET_MATRIX Command

The SET_MATRIX command is used to change the values of a control point or points in the matrix. It can be used to fill a matrix or a subregion of a matrix horizontally or vertically with:

- — The same repeating value
- — A list of repeating values
- — A list of non-repeating values
- — A list non-repeating values spanning more than one SET_MATRIX command

The SET_MATRIX command acts on a MATRIX descriptor in the current configuration. An ATDECC Entity may propagate the matrix value changes onto corresponding descriptors in other configurations but an ATDECC Controller cannot assume that this will happen.

On success this command also sends an unsolicited notification.

7.4.33.1. Command Format

The SET_MATRIX command uses the AECPDU format as shown in Figure 7-56.

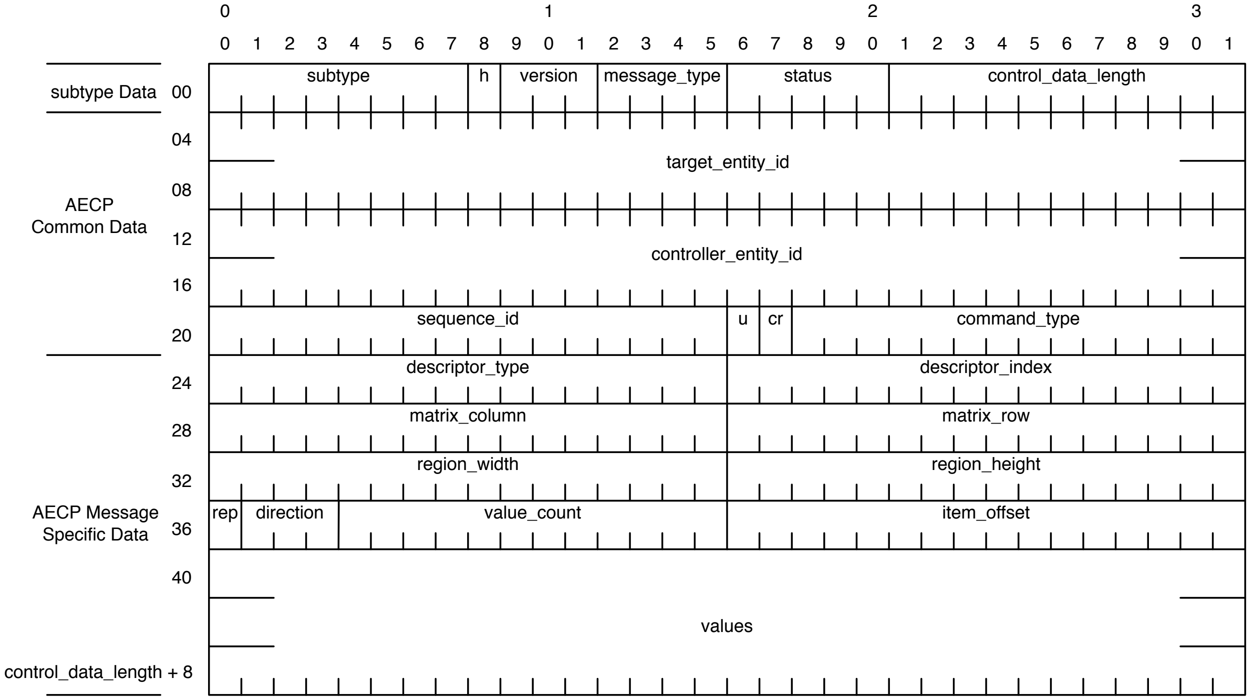

Figure 7-56—**SET_MATRIX Command and Response Format**

The **command_type** field is set to SET_MATRIX.

The **descriptor_type** and **descriptor_index** fields are set to the descriptor type and index of the matrix for which the current values are being set. **descriptor_type** is set to MATRIX.

The **matrix_column** field is set to the starting column of the subregion in the matrix.

The **matrix_row** field is set to the starting row of the subregion in the matrix.

The **region_width** field is set to the width (column count) of the subregion in the matrix.

The **region_height** field is set to the height (row count) of the subregion in the matrix.

The **rep** field is set to one (1) if the entire matrix subregion is to be filled with repeating values.

The **direction** field is set to one of the valid values as appropriate from Table 7-146.

Table 7-146—**SET_MATRIX direction field values**

Value	**Name**	**Description**
0	HORIZONTAL	Fill the subregion in the matrix with the values horizontally.
1	VERTICAL	Fill the subregion in the matrix with the values vertically.
2 to 3	—	Reserved

The **value_count** field is set to the number of matrix values in the **values** payload that will be applied to the subregion in the direction specified by the **direction** field.

The **item_offset** field is set to a count of items in the subregion to skip in the direction specified by the **direction** field before applying the **values** payload.

The **values** field is set to the matrix point values that are appropriate for this matrix, as defined by the **control_type** field in 7.2.25. The response always contains the current value, that is it contains the new value if the command succeeds or the old value if it fails.

7.4.33.2. Response Format

The SET_MATRIX response uses the AECPDU format as shown in Figure 7-56.

The **command_type** field is set to SET_MATRIX.

The **descriptor_type** and **descriptor_index** fields are set to the descriptor type and index of the matrix for which the current values are being set. **descriptor_type** is set to MATRIX.

The **matrix_column** field is set to the starting column of the matrix that was set.

The **matrix_row** field is set to the starting row of the subregion in matrix that was set.

The **region_width** field is set to the width (column count) of the subregion in the matrix that was set.

The **region_height** field is set to the height (row count) of the subregion in the matrix that was set.

The **rep** field is set to one (1) if the entire matrix subregion was filled with repeating values.

The **direction** field is set to the one of the valid values as appropriate from Table 7-146 describing the direction of the values that was set.

The **value_count** field is set to the **value_count** field of the corresponding SET_MATRIX command.

The **item_offset** field is set to the **item_offset** field value of the corresponding SET_MATRIX command.

The **values** field is set to the matrix point values that are appropriate for this matrix, as defined by the **control_type** field in 7.2.25. The response always contains the current value, that is it contains the new value if the command succeeds or the old value if it fails.

7.4.33.3. Restrictions

If the ATDECC Entity has been locked or acquired by another ATDECC Controller then the ATDECC Entity responds with an ENTITY_LOCKED or ENTITY_ACQUIRED status response.

The SET_MATRIX command may be protected by authentication. If this is the case then the ATDECC Entity responds with a NOT_AUTHENTICATED status response to an unauthenticated ATDECC Controller.

7.4.34. GET_MATRIX Command

The GET_MATRIX command is used to get the current values of a control point in the matrix. It can be used to get the values of matrix or a values of a subregion of a matrix horizontally or vertically.

The GET_MATRIX command returns the values for the currently active configuration.

7.4.34.1. Command Format

The GET_MATRIX command uses the AECPDU format as shown in Figure 7-57.

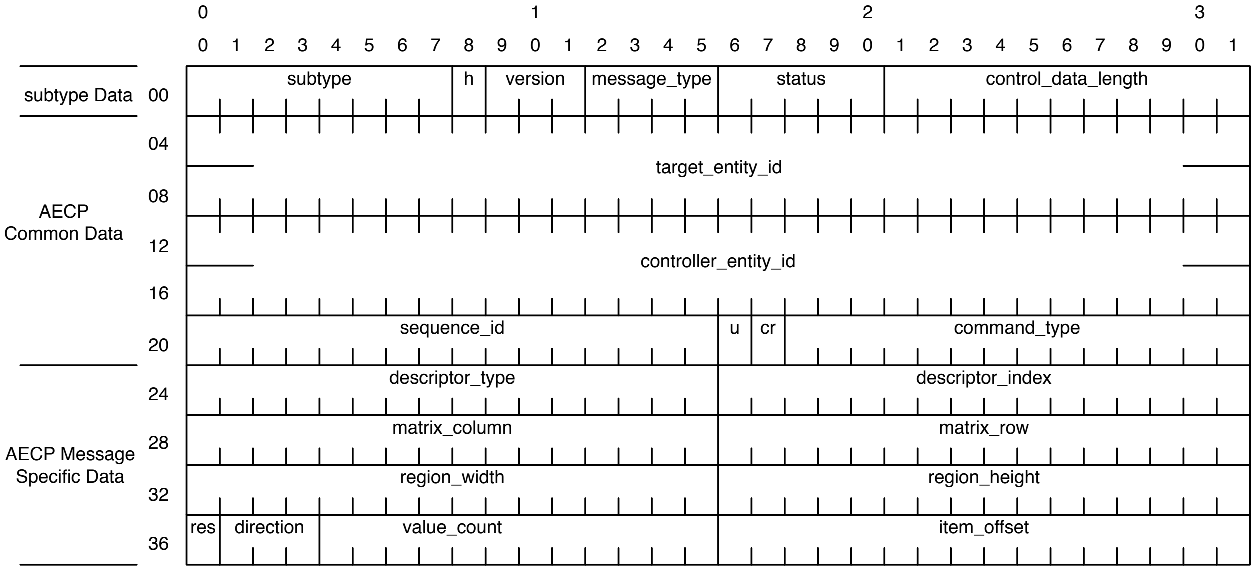

Figure 7-57—**GET_MATRIX Command Format**

The **command_type** field is set to GET_MATRIX.

The **descriptor_type** and **descriptor_index** fields are set to the descriptor type and index of the matrix for which the current values are being fetched. **descriptor_type** is set to MATRIX.

The **matrix_column** field is set to the starting column of the subregion in the matrix.

The **matrix_row** field is set to the starting row of the subregion in the matrix that is being requested.

The **region_width** field is set to the width (column count) of the subregion in the matrix that is being requested.

The **region_height** field is set to the height (row count) of the subregion in the matrix that is being requested.

The **res** is reserved and is set to zero (0).

The **direction** field is set to one of the valid values as appropriate from Table 7-146.

The **value_count** field is set to the count of the values being requested, after the **item_offset** skip.

The **item_offset** field is set to a count of items to skip in the direction specified by the **direction** field.

7.4.34.2. Response Format

The GET_MATRIX response uses the AECPDU format as shown in Figure 7-58.

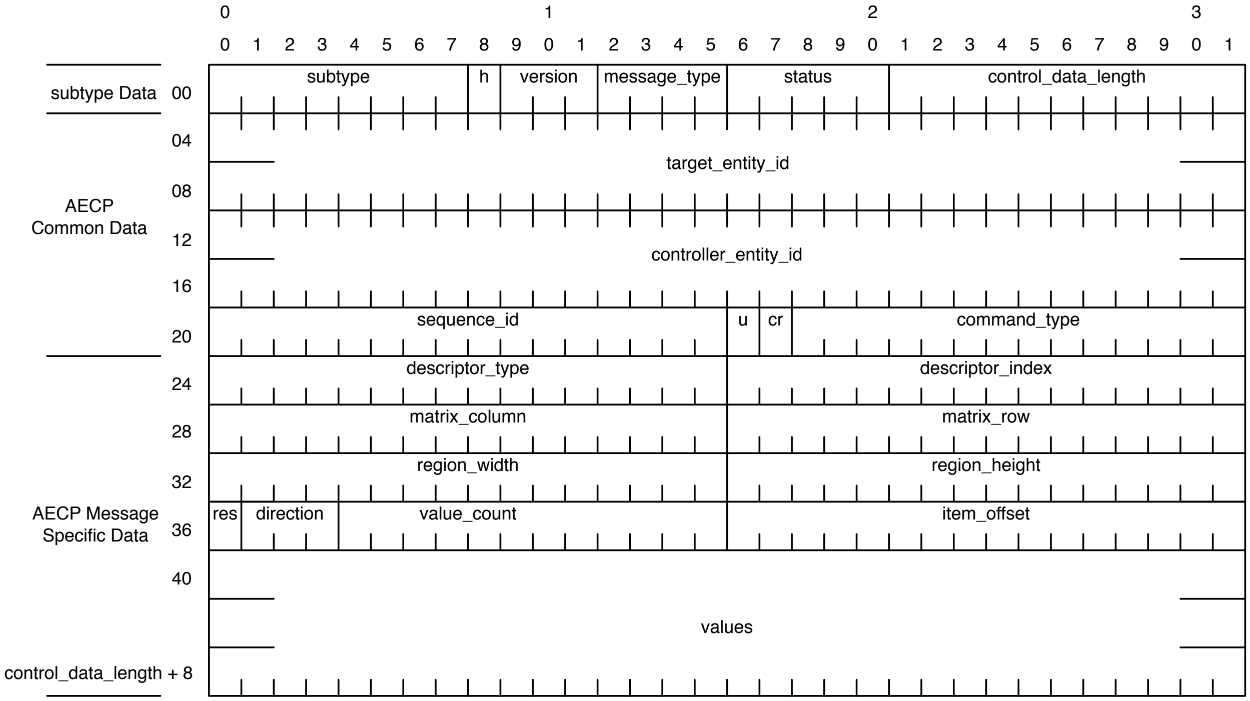

Figure 7-58—**GET_MATRIX Response Format**

The **command_type** field is set to GET_MATRIX.

The **descriptor_type** and **descriptor_index** fields are set to the descriptor type and index of the matrix for which the current values are being fetched. **descriptor_type** is set to MATRIX.

The **matrix_column** field is set to the starting column of the subregion in the matrix.

The **matrix_row** field is set to the starting row of the subregion in the matrix that is being requested.

The **region_width** field is set to the width (column count) of the subregion in the matrix that is being requested.

The **region_height** field is set to the height (row count) of the subregion in the matrix that is being requested.

The **res** is reserved and is set to zero (0).

The **direction** field is set to one of the valid values as appropriate from Table 7-146.

The **value_count** field contains the number of matrix point values in the **values** payload field.

The **item_offset** field is set to a count of items to skip in the direction specified by the **direction** field.

7.4.34.3. Restrictions

The GET_MATRIX command may be protected by authentication. If this is the case then the ATDECC Entity responds with a NOT_AUTHENTICATED status response to an unauthenticated ATDECC Controller.

7.4.35. START_STREAMING Command

The START_STREAMING command is used to start an already connected stream that was connected via ACMP with the STREAMING_WAIT flag set or which has previously been stopped with the STOP_STREAMING command.

The START_STREAMING command acts on a STREAM_INPUT or STREAM_OUTPUT descriptor in the current configuration. An ATDECC Entity may propagate the streaming state change onto corresponding descriptors in other configurations but an ATDECC Controller cannot assume that this will happen.

On success this command also sends an unsolicited notification.

7.4.35.1. Command and Response Format

The START_STREAMING command and response share the same AECPDU format as shown in Figure 7-59.

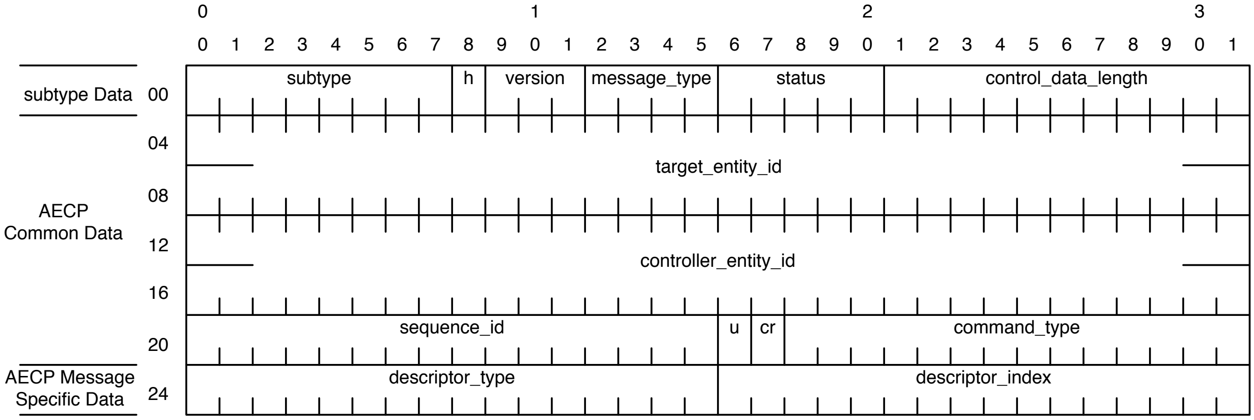

Figure 7-59—**START_STREAMING and STOP_STREAMING Command and Response Format**

The **command_type** field is set to START_STREAMING.

The **descriptor_type** and **descriptor_index** fields are set to the descriptor type and index of the stream which is being started. **descriptor_type** is set to either STREAM_INPUT or STREAM_OUTPUT.

7.4.35.2. Restrictions

If the ATDECC Entity is locked or acquired by another ATDECC Controller then the ATDECC Entity responds with an ENTITY_LOCKED or ENTITY_ACQUIRED status response to the ATDECC Controller.

The START_STREAMING command may be protected by authentication. If this is the case then the ATDECC Entity responds with a NOT_AUTHENTICATED status response to an unauthenticated ATDECC Controller.

7.4.36. STOP_STREAMING Command

The STOP_STREAMING command is used to stop a connected stream from streaming media. It does not disconnect the stream.

The STOP_STREAMING command acts on a STREAM_INPUT or STREAM_OUTPUT descriptor in the current configuration. An ATDECC Entity may propagate the streaming state change onto corresponding descriptors in other configurations but an ATDECC Controller cannot assume that this will happen.

On success this command also sends an unsolicited notification.

7.4.36.1. Command and Response Format

The STOP_STREAMING command and response share the same AECPDU format as shown in Figure 7-59.

The **command_type** field is set to STOP_STREAMING.

The **descriptor_type** and **descriptor_index** fields are set to the descriptor type and index of the stream which is being stopped. **descriptor_type** is set to either STREAM_INPUT or STREAM_OUTPUT.

7.4.36.2. Restrictions

If the ATDECC Entity is locked or acquired by another ATDECC Controller then the ATDECC Entity responds with an ENTITY_LOCKED or ENTITY_ACQUIRED status response to the ATDECC Controller.

The STOP_STREAMING command may be protected by authentication. If this is the case then the ATDECC Entity responds with a NOT_AUTHENTICATED status response to an unauthenticated ATDECC Controller.

7.4.37. REGISTER_UNSOLICITED_NOTIFICATION Command

The REGISTER_UNSOLICITED_NOTIFICATION command is used to add the ATDECC Controller as being interested in receiving unsolicited response notifications.

7.4.37.1. Command and Response Format

The REGISTER_UNSOLICITED_NOTIFICATION command and response share the same AECPDU format as shown in Figure 7-60.

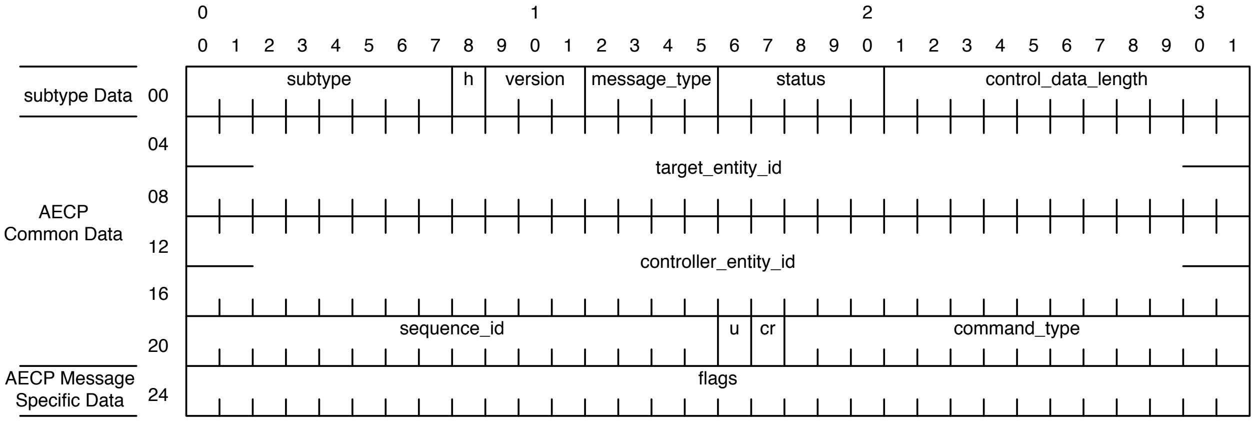

Figure 7-60—**REGISTER_UNSOLICITED_NOTIFICATION Command and Response Format**

The **command_type** field is set to REGISTER_UNSOLICITED_NOTIFICATION.

The **flags** field is set to a combination of values as appropriate from Table 7-147 or zero (0).

Table 7-147—**REGISTER_UNSOLICITED_NOTIFICATION Flags**

Bit	Field Value	Name	Description
31	00000001_{16}	TIME_LIMITED	The registration will automatically timeout and be removed if it is not renewed.
0-30	—	—	Reserved for future use.

NOTE—The REGISTER_UNSOLICITED_NOTIFICATION command and response has additional fields when compared to the REGISTER_UNSOLICITED_NOTIFICATION response from IEEE Std 1722.1-2013. A controller needs to be prepared to receive a response that does not include the **flags** field and has a shorter length when receiving a response from an ATDECC Entity compliant with the earlier version of the standard.

An ATDECC Entity shall accept an REGISTER_UNSOLICITED_NOTIFICATION command both with or without the new **flags** field, i.e., both the format defined here and the format defined in IEEE Std 1722.1-2013.

This allows both forward and backward compatibility of the REGISTER_UNSOLICITED_NOTIFICATION command.

When an ATDECC Entity receives a command with the old format (without the **flags** field) it is treated as thought the **flags** field is set to 0.

7.4.37.2. Time limiting

When the TIME_LIMITED flag is set this by the ATDECC Controller this indicates that it will periodically re-register for notifications by sending a REGISTER_UNSOLICITED_NOTIFICATION command every 100 seconds.

Upon receiving and registering (or reregistering) the ATDECC Controller, the ATDECC Entity will start a timeout of 300 seconds for the registration. If the timeout expires (i.e., there was no re-registration) then the ATDECC Entity shall send a DEREGISTER_UNSOLICITED_NOTIFICATION unsolicited response for the ATDECC Controller and remove the registration.

NOTE—These timeout values have been chosen based on the default dynamic filtering entry ageing defined in 8.7.3 of IEEE Std 802.1Q-2018. They have been chosen to limit the amount of time that unicast flooding will occur on the network before the entity stops sending. However the ageing value can be changed by management and as such these values can represent a longer time of unicast flooding if the ageing time is reduced.

7.4.37.3. Restrictions

The REGISTER_UNSOLICITED_NOTIFICATION command may be protected by authentication. If this is the case then the ATDECC Entity responds with a NOT_AUTHENTICATED status response to an unauthenticated ATDECC Controller.

7.4.38. DEREGISTER_UNSOLICITED_NOTIFICATION Command

The DEREGISTER_UNSOLICITED_NOTIFICATION command is used to remove the ATDECC Controller from receiving unsolicited response notifications.

7.4.38.1. Command and Response Format

The DEREGISTER_UNSOLICITED_NOTIFICATION command uses the base AEM AECPDU as defined in Figure 9-2.

The **command_type** field is set to DEREGISTER_UNSOLICITED_NOTIFICATION.

The **command_specific_data** field is zero length.

7.4.38.2. Restrictions

The DEREGISTER_UNSOLICITED_NOTIFICATION command may be protected by authentication. If this is the case then the ATDECC Entity responds with a NOT_AUTHENTICATED state response to an unauthenticated ATDECC Controller.

7.4.39. IDENTIFY_NOTIFICATION Unsolicited Response

The IDENTIFY_NOTIFICATION unsolicited response is used by the identification notifications to signal an identification button press via a user on an ATDECC Entity. IDENTIFY_NOTIFICATION shall never be sent as a command.

7.4.39.1. Unsolicited Response Format

The IDENTIFY_NOTIFICATION unsolicited response use the AECPDU format as shown in Figure 7-61.

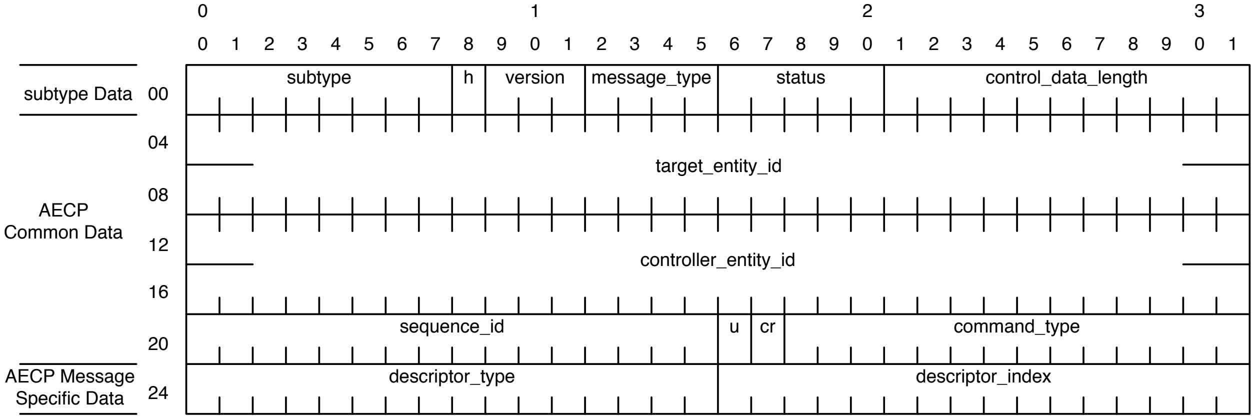

Figure 7-61—**IDENTIFY_NOTIFICATION Unsolicited Response Format**

The **u** field is set to one (1).

The **command_type** field is set to IDENTIFY_NOTIFICATION.

The **descriptor_type** field is set to CONTROL.

The **descriptor_index** field is set to the index of the IDENTIFY control generating the unsolicited response.

7.4.39.2. Restrictions

IDENTIFY_NOTIFICATION is only ever sent as an unsolicited response by the ATDECC Entity. If an ATDECC Entity ever receives this as a command then it shall return a response with the status code BAD_ARGUMENTS.

7.4.40. GET_AVB_INFO Command

The GET_AVB_INFO command is used to get the dynamic AVB information for an AVB_INTERFACE in the current configuration.

7.4.40.1. Command Format

The GET_AVB_INFO command has the AECPDU format as shown in Figure 7-62.

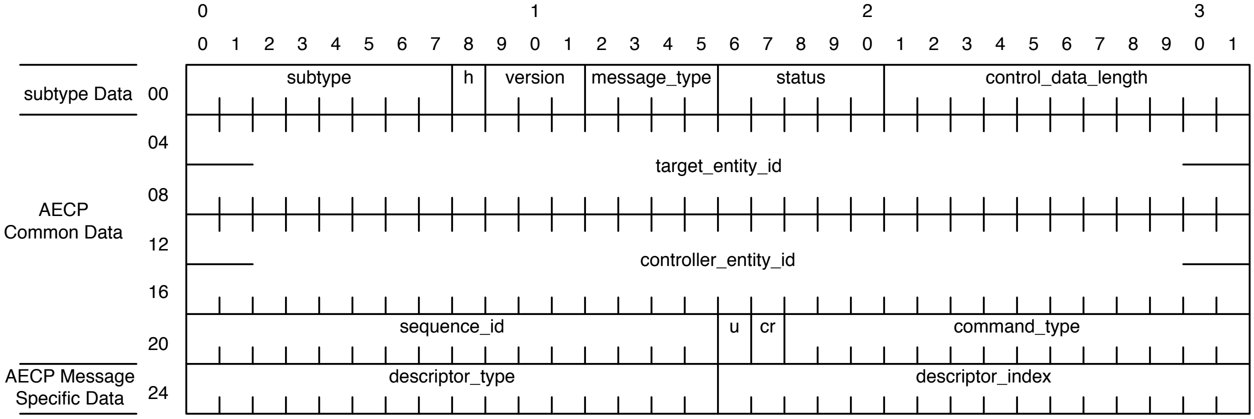

Figure 7-62—**GET_AVB_INFO Command Format**

The **command_type** field is set to GET_AVB_INFO.

The **descriptor_type** and **descriptor_index** fields are set to the descriptor type and index of the AVB Interface the information is being fetched from. **descriptor_type** is AVB_INTERFACE.

7.4.40.2. Response Format

The GET_AVB_INFO response has the AECPDU format as shown in Figure 7-63.

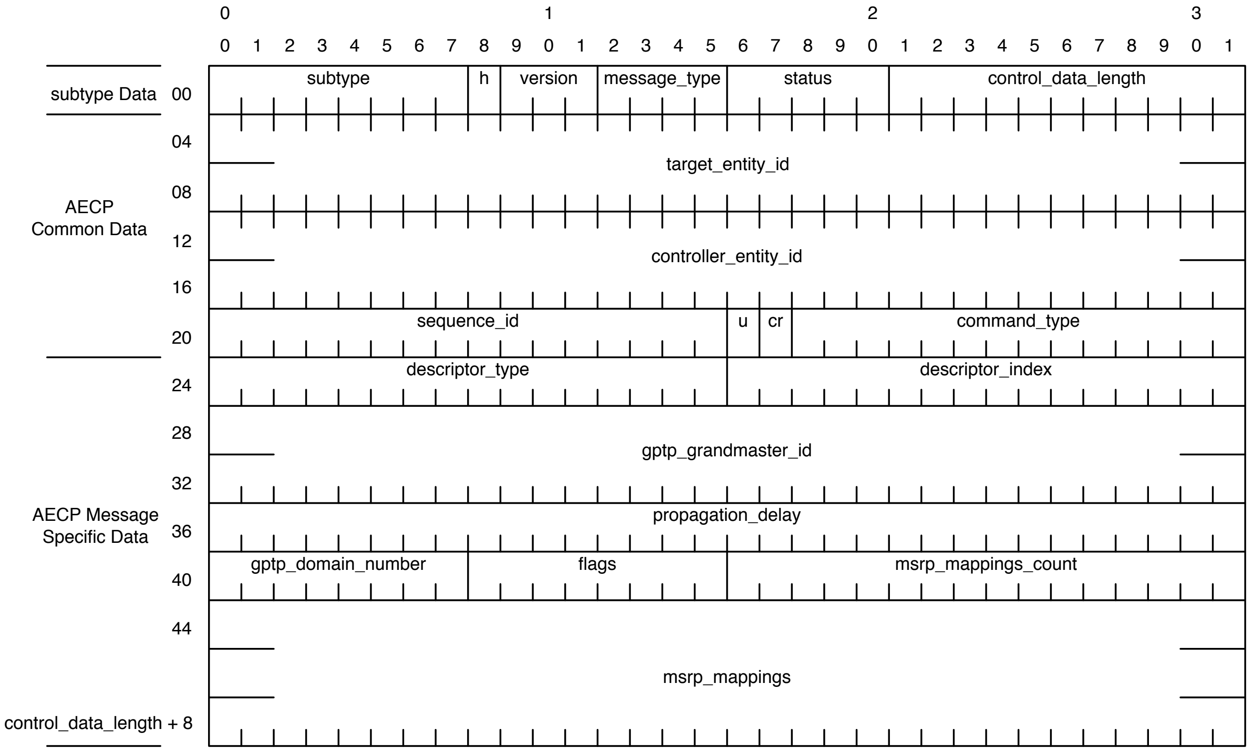

Figure 7-63—**GET_AVB_INFO Response Format**

The **command_type** field is set to GET_AVB_INFO.

The **descriptor_type** and **descriptor_index** fields are set to the descriptor type and index of the AVB Interface the information is being fetched from. **descriptor_type** is AVB_INTERFACE.

The **gptp_grandmaster_id** field is set to the ClockIdentity of the current IEEE Std 802.1AS-2020 grandmaster as elected on this AVB interface.

The **propagation_delay** field is set to the propagation delay in nanoseconds as reported by the IEEE Std 802.1AS-2020 pDelay mechanism.

The **gptp_domain_number** field is set to the domainNumber of the current IEEE Std 802.1AS-2020 grandmaster as elected on this AVB interface.

The **flags** field is set to a combination of values as appropriate from Table 7-148 or zero (0).