AVBC via SNMP for 1722.1

This is a proposal based on Jeff Koftinoff's AVBC v1.3 connection management and system control proposal. This proposal demonstrates how the core functionality of AVBC can be accomplished using simple network management protocol (SNMP).

Revision history

Version	Date	By	Changes
0.1	25 April 2010	Kevin Gross	Initial revision
1.0	26 April 2010	Kevin Gross	Proofing edits

SNMP

This proposal assumes knowledge of SNMP. *Controlling and Monitoring Audio Systems with Simple Network Management Protocol (SNMP)*[1] is a suggested reference for becoming familiar with SNMP in the context of networked audio devices.

MIB

The management information base (MIB) is a database of management variables arranged in a hierarchical manner. Variables in the database are accessed using object identifiers (OIDs). An OID is a string of integers enumerating the path from root to the variable of interest. An example MIB for an audio device is shown in Figure 1.

[1] Gross, Kevin P.; Holtzen Tom, September 1998. Available via AES E-Library: http://www.aes.org/e-lib/browse.cfm?elib=8420

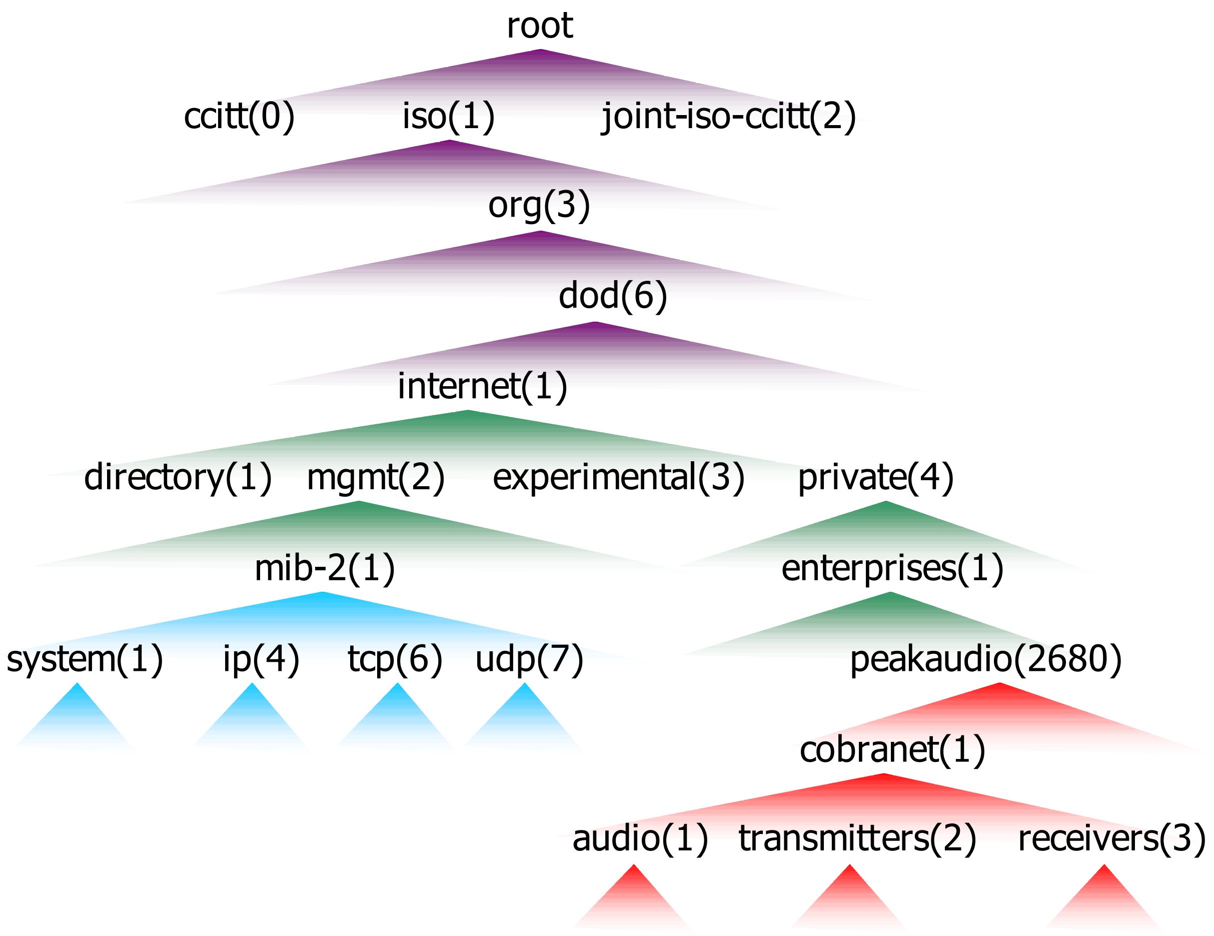

Figure 1: Example MIB for a CobraNet device

Datatypes

The SNMP protocol directly supports three basic data types

1. Integer – 32-bit signed integer
2. Octet string – byte-oriented string of binary data
3. OID – an object identifier used to address a node in a MIB (e.g. 1.3.6.1.2.1.1.3.0)

Other data types such as (e.g. text string, enumeration) are derived from these basic types.

A 64-bit counter type, *Counter64*, was introduced in SNMPv2.

Notably absent is a floating point data type. In SNMP, higher-precision data is typically conveyed in a fixed point format derived from Integers. *sysUpTime*, for instance is defined in units 1/100ths of a second.

Protocol

GetRequest

Retrieve the value of a variable or list of variables. Desired variables specified in variable bindings (values are not used). Retrieval of the specified variable values is to be done as an atomic operation by the agent. A *Response* with current values is returned.

SetRequest

Change the value of a variable or list of variables. Variable bindings specified in the body of the request. Changes to all specified variables are to be made as an atomic operation by the agent. A *Response* with (current) new values for the variables is returned.

GetNextRequest

Returns a *Response* with variable binding for the lexicographically next variable in the MIB. The entire MIB of an agent can be walked by iterative application of *GetNextRequest* starting at OID 0. Rows of a table can be read by specifying column OIDs in the variable bindings of the request.

GetBulkRequest

Optimized version of *GetNextRequest*. Requests multiple iterations of *GetNextRequest* and returns a *Response* with multiple variable bindings walked from the variable binding or bindings in the request.

Response

Returns variable bindings and acknowledgement for *GetRequest, SetRequest, GetNextRequest* and *GetBulkRequest*. Error reporting is provided by *error-status* and *error-index* fields.

Trap

Asynchronous notification from agent to manager. Includes *sysUpTime*, an OID identifying the type of trap and optional variable bindings. Destination addressing for traps is determined in an application specific manner typically through trap configuration variables in the MIB.

InformRequest

Acknowledged asynchronous notification from manager to manager. Same format as Trap. Receiver replies with Response parroting all information in the *InformRequest*.

AVBC mapping

This section demonstrates how the proposed AVBC capabilities may be mapped onto an SNMP implementation.

Schema (5.1.4)

This schema gives a means for a manager to determine what variables exist on an agent. Under SNMP this facility is provided by *GetNextRequest* and *GetBulkRequest*. Under AVBC, variable names are returned. Under SNMP variable OIDs are returned. While AVBC’s variables are more human readable, the SNMP OIDs reference an associated MIB definition file where variables are described in human and machine readable detail.

Limits (5.1.5)

AVBC provides a means for a manager to retrieve variable attributes including data type, minimum and maximum allowed values and description. Variables retrieved through SNMP indicate their basic data type – Integer, (octet) string or OID. The additional attributes for SNMP variables are available by referencing the MIB definition file.

While the requirement for an external reference to the MIB definition file for retrieving this information may at first seem cumbersome, presenting it in this way saves overhead in protocol operations that would be required to retrieve it and memory space on the agent that would be required to store it.

Subscriptions (5.1.6)

AVBC proposes a comprehensive mechanism for notification of changes to specified variables. Notification, however, does not appear to be a critical requirement for enumeration and connection management. Requirements can likely be handled by notification directed polling. Three AVB-specific traps are proposed.

1. Change in device capabilities
2. Successful stream connection – indicates affected sources or sinks by index
3. Loss of stream connection – indicates affected sources or sinks by index

Device (5.2)

The *system* branch in the SNMP MIB-2 provides the following device identifying variables which appear to cover proposed AVBC functionality.

- sysObjectID (RO) – globally unique numeric manufacturer (/device/identity/vendor_id) and product identifier
- sysDescr (RO) – textual description of the device. Often includes manufacturer (/device/identity/vendor) and product (/device/identity/product) designations, firmware version (/device/identity/version) and serial number (/device/identity/serial).
- sysName (RW) – administratively assigned network name (/device/name) of device
- sysLocation[2] (RW) – administratively assigned (RW) device location. Propose to use to indicate system name (/device/system).

Connection management

Streams

A device needs to indicate formats it is able to receive and transmit. Under AVBC, these are indicated by */avb/source/formats* and */avb/sink/formats*. Formats are described in textually. Overall details are slim. A numeric, string or OID representation is possible in the SNMP implementation. Under SNMP using OIDs is preferred for extensibility.

[2] *sysContact* is an additional option for providing device context

AVBC allows for dynamic creation and destruction of AVB stream sources and sinks. There is precedent for these sorts of operations within the SNMP framework (e.g. entries are dynamically added to and removed from IP routing tables).

Arrangement of sources and sinks in tables with a fixed number of rows corresponding to the maximum possible number of supported sources or sinks on the device is suggested as starting point for discussion.

Stream sources (5.3.1)

A table of stream sources (talkers) with the following columns is proposed.

Name	Access	Data type	Description	AVBC mapping
name	RW	String	Textual name of stream sourced. No stream is sourced if name is zero length or same name as another stream source on the device.	/avb/source/create (stream name)
id	RO	octet string [8]	AVB assigned StreamID for sourced stream	/avb/source/*/id
mmac	RO	physAddress	MAAP assigned multicast destination address for sourced stream	/avb/source/*/mmac
state	RO	Enum	State of stream source: 0 – potential, 1 – reserved, 2 - active	/avb/source/*/state
format	RW	Enum	Selected transmission format for sourced stream[3]	/avb/source/*/format
channels	RW	Integer	Number of channels in sourced stream	/avb/source/*/channels
map	RW	Sub-table of integers, one row per channel	Upper 16 bits defines media ID. Lower 16 bits defines a channel within that source.	/avb/source/*/map
presentation	RW	Integer[4]	Offset from origination time to presentation time in nanoseconds[5]	/avb/source/*/presentation

Stream sinks (5.3.2)

A table of stream sinks (listeners) with the following columns is proposed.

[3] Does format need to be selectable on a per-channel basis?

[4] Integer is suggested by AVBC but this should probably look the 802.1AS MIB for a real data type

[5] *presentation* variable is available at both source and sink. Semantics are not well documented in the AVBC proposal.

Name	Access	Data type	Description	AVBC mapping
name	RW	String	Textual name of desired stream. No stream is received if name is zero length	/avb/sink/*/create (stream name), /avb/sink/*/name
device	RW	String	Network name of device sourcing desired stream	/avb/sink/*/create (device name), /avb/sink/*/device
id	RW	octet string [8]	AVB assigned StreamID for desired stream	/avb/sink/*/create (stream id), /avb/sink/*/id
mmac	RW	physAddress	Multicast destination address for desired stream	/avb/sink/*/create (multicast mac address), /avb/sink/*/mmac
state	RO	Enum	State of stream sink: 0 – potential, 1 – reserved, 2 - active	/avb/sink/*/state
format	RO	Enum	Format of received stream[6]	/avb/sink/*/format
channels	RO	Integer	Number of channels in received stream	/avb/sink/*/channels
presentation	RW	Integer[7]	Offset from indicated presentation time to playback time in nanoseconds[8]	/avb/sink/*/presentation
map	RW	Sub-table of integers, one row per channel	Upper 16 bits defines media ID. Lower 16 bits defines a channel within that destination.	/avb/sink/*/map

Media

From an enumeration and connection management perspective, the primary purpose of the media interface is to allow a mapping of audio input and output channels to stream sources and stream sinks respectively through the /avb/source/*/map and /avb/sink/*/map facilities.

The AVBC proposal features several optional parameters related to audio processing and playback control. These optional capabilities are not currently addressed by this proposal.

Media source control (5.4)

A table of media sources (audio inputs) with the following columns is proposed.

[6] Does format need to be indicated on a per-channel basis?

[7] Integer is suggested by AVBC but this should probably look the 802.1AS MIB for a real data type

[8] *presentation* variable is available at both source and sink. Semantics are not well documented in the AVBC proposal.

Name	Access	Data type	Description	AVBC mapping
type	RO	string	Textural description of the type of audio input[9]	/media/source/*/type
description	RO	string	Textual description of the audio input	/media/source/*/description
channels	RO	Integer	Number of channels associated with the input	/media/source/*/channels

Media sink control (5.5)

A table of media sinks (audio outputs) with the following columns is proposed.

Name	Access	Data type	Description	AVBC mapping
type	RO	string	Textural description of the type of audio output[10]	/media/sink/*/type
description	RO	string	Textual description of the audio output	/media/sink/*/description
channels	RO	Integer	Number of channels associated with the output	/media/sink/*/channels

[9] There appears to be overlap between /media/source/*/type and /media/source/*/description

[10] There appears to be overlap between /media/sink/*/type and /media/sink/*/description