Entities enumeration with lots of devices (and how to reduce required traffic / reduce enumeration time)

O owner-avbtp-decc@LISTSERV.IEEE.ORG on behalf of Christophe CALMEJANE <cl

Reply all |

Thu 01-17, 04:10

AVBTP-DECC@LISTSERV.IEEE.ORG

Hi all,

As explained during yesterday's 1722.1 call, I'd like to start discussions (and proposals) on how to improve the standard regarding entities enumeration, especially when the network has a lot of them. The objectives being how to reduce the required traffic between a controller and the entities, and the overall enumeration time.

Background

Some controllers are required to enumerate all entities on a network, and keep an accurate dynamic state for each of them (for all or just a part of the AEM).
To do that, said controllers must retrieve all the descriptors it's interested in, and subscribe to unsolicited notifications.
Descriptors are split into 2 parts: static and dynamic information (some descriptors only having a static part).
The READ_DESCRIPTOR command returns both parts in a single command, but at the cost of large responses (and one command-response cycle for each descriptor).

On a large network (hundreds of entities) this full enumeration starts to add up, to the point where entities are overflowed and responses starts to timeout (observed in real case scenario).

Currently possible improvement

One current possible optimization is to use AEM caching (either by pre-caching AEM from aemxml files, or by dumping already enumerated devices), but this is only possible for the static part of descriptors.

Reply all | Delete Junk |

This light enumeration helps reducing the load on the network, but only by a small factor (see attached spreadsheet). It mainly reduces the required bandwidth, not the number of command-response cycles (it actually increase it).
Overall, the measured enumeration time is a little bit faster, but it's not satisfying.

Proposal 1

A first proposal would be to add new AEM commands to the standard, that would aggregate the dynamic information of each descriptor into a unique command.
GET_AUDIO_UNIT_DYNAMIC_INFO_COMMAND: Returns AUDIO_UNIT object_name and current_sampling_rate (instead of having to send GET_NAME and GET_SAMPLING_RATE)
GET_STREAM_DYNAMIC_INFO COMMAND: Returns STREAM object_name and current_format (instead of having to send GET_NAME and GET_STREAM_FORMAT)
GET_MEMORY_OBJECT_DYNAMIC_INFO COMMAND: Returns MEMORY_OBJECT object_name and length (instead of having to send GET_NAME and GET_MEMORY_OBJECT_LENGTH)
GET_CLOCK_DOMAIN_DYNAMIC_INFO COMMAND: Returns CLOCK_DOMAIN object_name and clock_source_index (instead of having to send GET_NAME and GET_CLOCK_SOURCE)

For the other descriptors that only have one dynamic information, the current commands should be used (GET_NAME for AVB_INTERFACE.object_name, CLOCK_SOURCE.object_name, AUDIO_CLUSTER.object_name)

This proposal improves a little bit the number of command-response cycles (see attached spreadsheet for simulations), as well as the bandwidth.
Overall, the estimated enumeration time is greatly improved, down to only 37% of full enumeration time.

Proposal 2

A second proposal would be to add a single new AEM command to the standard, that would aggregate all the descriptor dynamic parts a controller is interested in (for all descriptors).
For example, the command could be a list of descriptor dynamic information the controller wants:
- 16 bits: configuration_index
- 16 bits: number_of_dynamic_info
- 16 + 16 + 16 bits: dynamic_info_1 (command_type / descriptor_type / descriptor_index)
- 16 + 16 + 16 bits: dynamic_info_2 (command_type / descriptor_type / descriptor_index)
...
- 16 + 16 + 16 bits: dynamic_info_n (command_type / descriptor_type / descriptor_index)

Since all the requested information will not fit in a single response, we'll have to find a clever way to split the responses.
The easiest solution I can think of right now, is for the controller to compute the required response size and only send a query for a number dynamic_info that will not exceed the maximum allowed size. If the response would exceed the entity responds with an error code.

Reply all | Delete Junk |

internal state. It means that it can receive an unsolicited in-between responses, as long as the entity guarantees that when a message is sent is always contains the most accurate information.

I made a rough estimation (see attached spreadsheet) for this proposal, and it decreases the enumeration time down to only 5% of full enumeration and reduces the bandwidth down to 25%. The number of command-response cycles is also reduced to only 10% of full enumeration.

Other possibilities

I'm sure there are a lot of other possibilities to improved enumeration efficiency, so please share your ideas!
Just note that I'm not in favor of using a memory object, since it requires devices to implement AddressAccess and the state might change before the controller have a chance to read it. Might be hard to synchronize everything. Also, the entity might have to hold on the allocated memory buffer for a long period of time (and maybe have one memory object per controller?)

Thanks,

Christophe Calmejane
Software Department

L-Acoustics
13 rue Levacher Cintrat - 91460 Marcoussis - France
Office : +33 1 70 84 08 98