|  |  |
| --- | --- |
| **Rapporteur Meeting of Questions 2, 3, 4, 5, 12, 21, 22, and 25/16** | **AVD-4132(Q1-K09)** |
| **TELECOMMUNICATIONSTANDARDIZATION SECTOR**STUDY PERIOD 2009-2012 | **STUDY GROUP 16** |
| **Original: English** |
| **Question(s):** | 1, 5, 12 | Andover, MA, USA, 18 - 22 July 2011 |
| **RAPPORTEUR MEETING DOCUMENT** |
| **Source:** | Cisco Systems, Inc. |
| **Title:** | Capability Exchange within AMS |
| **Purpose:** | Proposal |

Summary

This document discusses capability exchange within AMS. This is a topic experts have discussed several times before, but still needs further discussion in order to reach closure.

# Introduction

We previously presented several contributions on the topic of capability exchange within AMS. Most recently, we prepared a contribution for the RTP meeting found in AVD-4036. This paper continues that discussion, with an aim of reaching some agreements. We believe we have reached some agreements already, but those agreements are not formalized in meeting reports.

# Discussion

## Application Registration

A basic behaviour of AMS is that applications will register with the Container, much as the Container might register with a service node. After several iterations in previous contributions, we concluded in AVD-4036 that applications should not register detailed capability information. For example, a video display device should not register with the Container indicating supported video codecs. Doing so complicates the signalling and the Container logic. After all, it is not the Container that negotiates use of particular applications and the Container should have no understanding of a codec. Applications are selected by the user and applications negotiate capabilities between themselves.

We also concluded that we should have syntax that allows a single physical device, executable program, etc. to register multiple AMS applications. For example, a Personal Computer might register six different applications that are addressable over a given Interface B. We do not propose requiring that every application running on a computer, for example, coordinate communication through a single process. However, we do propose enabling a single executable to expose several applications (e.g., a single executable might expose a voice and video application) by using a flexible syntax.

By allowing a single executable or device to register a plurality of applications at once, it will be necessary to have syntax that allows for the invocation of a particular application. We suggest making this possible by assigning a locally unique identifier to each application. So, for any given Container/Application interface, there may be messages sent between the Container and any number of applications that are differentiated by an identifier. The Container will need to know how to address a specific application and will need to be able to recognize messages from a given application.

The following syntax was suggested in AVD-4036:

<RegistrationRequest>
 ...
<id>ef0451a4-2fe2-11df-84c6-12313a006823</id>
<name>Personal Computer</name>
<applicationList>
<application id="1" type="http://www.itu.int/xml-namespace/itu-t/h.325/file/">
<applicationName>Video</applicationName>
</application>
<application id="2" type="http://www.itu.int/xml-namespace/itu-t/h.325/appshare/">
<applicationName>Application Sharing</applicationName>
</application>
<application id="3" type="http://www.itu.int/xml-namespace/itu-t/h.325/whiteboard/">
<applicationName>Electronic Whiteboard</applicationName>
</application>
<application id="4" type="http://www.itu.int/xml-namespace/itu-t/h.325/video/">
<applicationName>Video</applicationName>
</application>
<application id="5" type="http://www.itu.int/xml-namespace/itu-t/h.325/display/">
<applicationName>Display</applicationName>
</application>
<application id="6" type="http://www.itu.int/xml-namespace/itu-t/h.325/camera/">
<applicationName>Camera</applicationName>
</application>
</applicationList>
 ...
</RegistrationRequest>

We do not seek to propose the above syntax in this contribution, but do wish to get agreement on these points:

1. We will not carry detailed application capabilities within a registration message;
2. Within a registration message, there may be registrations for multiple distinct applications that are differentiated by some type of locally unique identifier, where “locally unique” means unique to a given Interface B (e.g., IP address and port);
3. All Interface B messages will allow the receiver to differentiate between applications based on this locally unique identifier; and
4. We *may* want to allow some type of “role” to be advertised to the Container, such as “primary video” or “presentation video” (similar to H.239)

## Session Establishment

We may want to allow for some type of “Fast Connect” exchange in AMS and the following is not intended to suggest otherwise.

When establishing a session with a remote AMS Container, we propose that the initiating Container include only a list of available applications. For each application, the Container should assign a unique number such that no two applications presented to the remote Container share the same number. (The same number may be used between the Container and various local applications since each would be addressed over a different Interface B.) What we proposed in AVD-4036 was the following:

<SessionEstablish>
 ...
<SessionID>de1abe38-2fe1-11df-9c13-12313a006823</SessionID>
<name lang="en-US">Paul E. Jones</name>
<name lang="zh-CN">保罗‧琼斯</name>

<!-- Display Screen -->
<applicationList>
<application id="1" type="http://www.itu.int/xml-namespace/itu-t/h.325/display/"/>

<!-- Video Terminal -->
<application id="2" type="http://www.itu.int/xml-namespace/itu-t/h.325/video/"/>
<application id="3" type="http://www.itu.int/xml-namespace/itu-t/h.325/voice/"/>
<application id="4" type="http://www.itu.int/xml-namespace/itu-t/h.325/audio/"/>
<application id="5" type="http://www.itu.int/xml-namespace/itu-t/h.325/microphone/"/>
<application id="6" type="http://www.itu.int/xml-namespace/itu-t/h.325/display/"/>
<application id="7" type="http://www.itu.int/xml-namespace/itu-t/h.325/camera/"/>
<!-- PC -->
<application id="8" type="http://www.itu.int/xml-namespace/itu-t/h.325/file/"/>
<application id="9" type="http://www.itu.int/xml-namespace/itu-t/h.325/appshare/"/>
<application id="10" type="http://www.itu.int/xml-namespace/itu-t/h.325/whiteboard/"/>
<application id="11" type="http://www.itu.int/xml-namespace/itu-t/h.325/video/"/>
<application id="12" type="http://www.itu.int/xml-namespace/itu-t/h.325/display/"/>
<application id="13" type="http://www.itu.int/xml-namespace/itu-t/h.325/camera/"/>
</applicationList>

<invoke id="1" type="http://www.itu.int/xml-namespace/itu-t/h.325/voice/"/>
<invoke id="2" type="http://www.itu.int/xml-namespace/itu-t/h.325/video/"/>
 ...
</SessionEstablish>

We do not wish to propose the specific syntax, but we would like to get agreement that the Container will advertise each application and assign an identifier. Further, we want to seek agreement that the same minimal capabilities advertised in the application registration will be exchanged in the session establishment message.

A remote Container, upon receiving this message, could see that the Container supports 13 different applications and that some application types are the same (e.g., “video”). A remote Container *could* invoke an application by specifying a specific application identifier. However, it is unlikely that the remote application would know the difference between the video applications, for example. Even if a role is advertised, there may be multiple applications that serve in the same role.

As illustrated above with the <invoke> commands, the initiating Container requests to invoke applications based on type. (The “id” in those commands is intended to refer to the invocation identifier, not an application identifier.) We *may* wish to allow the invocation to include a hint as to the type of role the application is intended to play. If a primary video session is desired, then perhaps the Container includes a “role” parameter as a part of the <invoke>.

Having said that, there may be instances where a remote Container does know the application identifier that it wishes to invoke. For example, if an application is invoked and then terminated and then re-invoked, the application may wish to retain an affinity with the same peer application. Therefore, we propose that application invocations include both a “type” parameter and an “applicationId” parameter. The “applicationId” parameter should be optional and, if absent, allows the user to select the specific application to invoke. If present, the Container will attempt to invoke the specific application assigned that identifier. Thus, we wish to propose functionality similar to the following:

<invoke id="2" type="http://www.itu.int/xml-namespace/itu-t/h.325/video/"
applicationId="5"/>

Should the “type” parameter be absent if an “applicationId” is also present? It would be redundant, but keeping it there allows for more consistency.

## Advertising Roles

In the previous section, we discussed use of roles. It could be possible for an application to register supported roles like this:

<application id="3" type="http://www.itu.int/xml-namespace/itu-t/h.325/video/">
<role type="primary\_video"/>
<role type="presentation\_video"/>
</application>

An invocation request seeking presentation video might look like this:

<invoke id="2" type="http://www.itu.int/xml-namespace/itu-t/h.325/video/"
role="presentation\_video"/>

This would instruct the Container to locate a “video” application that has a “presentation\_video” role.

We have given little consideration to this topic in Q12 thus far, but this is likely an important topic for consideration for Questions 1 and 5. We do not wish to suggest the method used or to introduce a specific proposal at this time. Rather, we just wish to encourage dialog related to the use of roles within AMS applications.

## Application Capability Exchange

This has been a topic of discussion during the electronic meetings of Q12/16. We have two possible interfaces over which capabilities might be exchanged:

* Interface B 🡪Interface A – Container to Container signalling interface
* Interface K – Application to Application signalling interface

We would like to propose establishment of guidelines as to what type of communication will flow over Interface A and what type of traffic will flow over Interface K. Should we advertise the list of supported codecs, for example, over Interface A or K? Should we allow application to pick and choose what interface to utilize to exchange certain capabilities?

We propose that Interface A be utilized to invoke applications and to allow applications to exchange capability information, such as screen resolution, codec types, media address information, etc. We further propose that Interface K be utilized for exchanging state information (e.g., for exchanging whiteboard data as a part of an application handover process), data structures, “video fast update” type messages, etc. In other words, we propose that Interface A be used to exchange any information necessary to successfully reach agreement that a particular application is compatible and to start the application, while we use Interface K for the exchange of any data that might be useful or necessary once the application is invoked.

## Invoking an Alternative Application

Above, we discussed invoking applications and negotiating capabilities. What happens if the invocation fails due to lack of compatibility, yet there are alternative applications that might work? Should we provide a mechanism whereby an application can request an alternative application should an invocation request fail or should capability negotiation fail?

# Conclusion

The authors would like to reach consensus on approaches to be taken to capability exchange within AMS as outlined in this contribution.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_