[MS-AVEDGEA]:
Audio Video Edge Authentication Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial version
04/25/2008	0.2		Revised and edited the technical content
06/27/2008	1.0		Revised and edited the technical content
08/15/2008	1.01		Revised and edited the technical content
12/12/2008	2.0		Revised and edited the technical content
02/13/2009	2.01		Revised and edited the technical content
03/13/2009	2.02		Edited the technical content
07/13/2009	2.03	Major	Revised and edited the technical content
08/28/2009	2.04	Editorial	Revised and edited the technical content
11/06/2009	2.05	Editorial	Revised and edited the technical content
02/19/2010	2.06	Editorial	Revised and edited the technical content
03/31/2010	2.07	Major	Updated and revised the technical content
04/30/2010	2.08	Editorial	Revised and edited the technical content
06/07/2010	2.09	Editorial	Revised and edited the technical content
06/29/2010	2.10	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.10	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	3.0	Major	Significantly changed the technical content.
11/15/2010	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	3.0	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document specifies the Audio Video Edge Authentication Protocol. This is a proprietary protocol used by protocol clients to get security tokens needed to authenticate themselves with a server that implements the Traversal Using Relay NAT (TURN) Extensions protocol, as described in [MS-TURN], for use with the Interactive Connectivity Establishment (ICE) Extensions protocol, as described in [MS-ICE] and [MS-ICE2].

1.1 Glossary

The following terms are defined in [MS-GLOS]:

fully qualified domain name (FQDN)
Hash-based Message Authentication Code (HMAC)
network address translation (NAT)
SHA-1 hash
Transmission Control Protocol (TCP)
User Datagram Protocol (UDP)

The following terms are defined in [MS-OFCGLOS]:

Audio/Video Edge Server (A/V Edge Server)
Content-Type header
endpoint
SERVICE
Session Initiation Protocol (SIP)
Transport Layer Security (TLS)
TURN server
Uniform Resource Identifier (URI)

The following terms are specific to this document:

shared-secret: Data that is known only to the parties that are involved in a secure communication.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[IETFDRAFT-SIPSOAP-00] Deason, N., "SIP and SOAP", draft-deason-sip-soap-00, June 30 2000, http://www.softarmor.com/wgdb/docs/draft-deason-sip-soap-00.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

[XMLSCHEMA1/2] Thompson, H.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures Second Edition", W3C Recommendation, October 2004, http://www.w3.org/TR/xmlschema-1/

[XMLSCHEMA2/2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, http://www.w3.org/TR/xmlschema-2

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary".

[MS-ICE] Microsoft Corporation, "Interactive Connectivity Establishment (ICE) Extensions"

[MS-ICE2] Microsoft Corporation, "Interactive Connectivity Establishment (ICE) Extensions 2.0"

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary".

[MS-SIPRE] Microsoft Corporation, "Session Initiation Protocol (SIP) Routing Extensions"

[MS-TURN] Microsoft Corporation, "Traversal Using Relay NAT (TURN) Extensions"

[RFC2104] Krawczyk, H., Bellare, M., and Canetti, R., "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997, http://www.ietf.org/rfc/rfc2104.txt

[XML10] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Third Edition)", February 2004, http://www.w3.org/TR/REC-xml

1.3 Protocol Overview (Synopsis)

[MS-TURN] is used for **network address translation (NAT)** and firewall traversal. To help protocol clients traverse NATs and firewalls, a **TURN server** or servers need to be deployed at particular places in the network topology. With the security tokens supplied by this protocol, a protocol client can authenticate with these TURN servers.

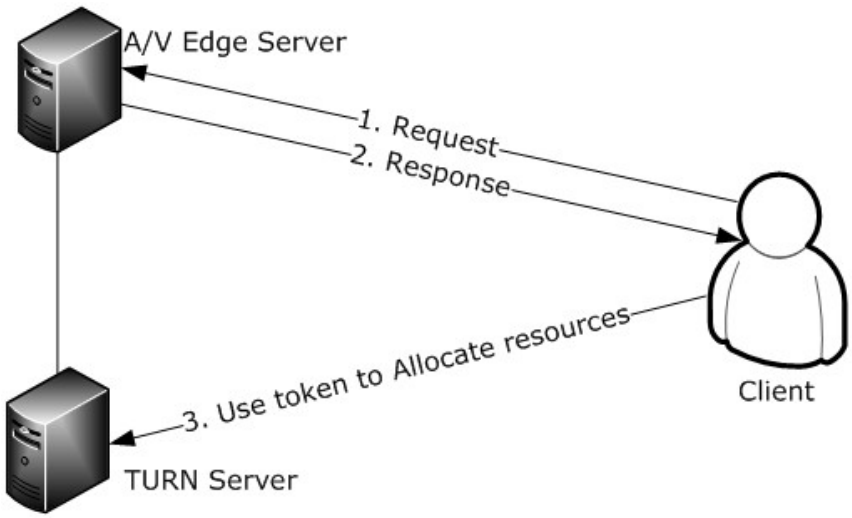

Figure 1: Protocol overview

The **Audio/Video Edge Server (A/V Edge Server)** is associated with a TURN server and is aware of the configuration details of the associated TURN server. When the protocol client needs tokens, it

sends a **Session Initiation Protocol (SIP) SERVICE** request, as described in [IETFDRAFT-SIPSOAP-00], to the A/V Edge Server with the body of the message encoded in an XML format, as described in [XML10]. The server responds with a SIP SERVICE response message and a response code which indicates whether the response was a success or failure. If it was a success, the response contains the security tokens along with location information of the associated TURN server. If it was a failure, the response code indicates the type of failure. If there was an error with the XML body of the request, the response also contains an XML body that describes the exact cause of the problem.

The A/V Edge Server shares **shared-secret** keys with the associated TURN server and uses these keys to generate tokens. A security token consists of a user name and password. The token is valid only for a certain amount of time. If the protocol client requires the token to be valid for a shorter time interval, it can specify the length of the interval in the XML request. The server honors this value if it is less than the default duration it uses. Because the expiration time in the token is not easily decipherable, the response also includes a duration element that specifies how long the token is valid. The token also includes a hash of the identity specified by the protocol client, which the TURN server can use to implement resource management.

1.4 Relationship to Other Protocols

This protocol uses [MS-SIPRE] for receiving requests and sending out responses. This protocol uses the SIP SERVICE method, as described in [IETFDRAFT-SIPSOAP-00], which is an extension to the standard SIP, to receive and send responses. The security tokens received from the A/V Edge Server are used to obtain access to the TURN server for use with the [MS-ICE] protocol.

1.5 Prerequisites/Preconditions

This protocol assumes that the TURN server associated with the A/V Edge Server has two network interfaces, one facing the internet and the other facing the intranet as shown in the following figure.

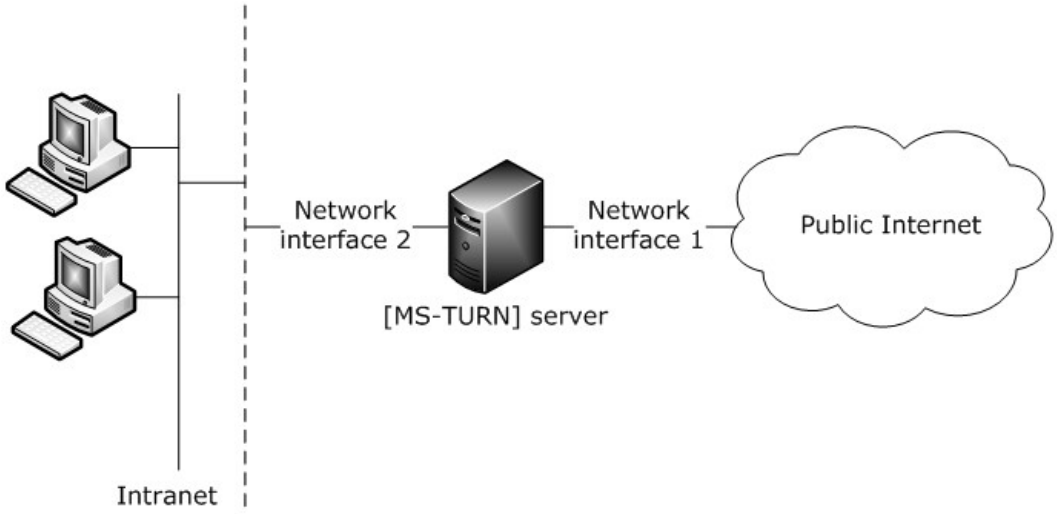

Figure 2: TURN server with two interfaces

The server implementing this protocol is assumed to be configured with the following:

- A certificate for establishing **Transport Layer Security (TLS)** connections. The certificate has a private key.

- Two shared-secret keys, known both to the A/V Edge Server and the associated TURN server.
- **User Datagram Protocol (UDP)** and **Transmission Control Protocol (TCP)** ports, on which the associated TURN server listens for protocol requests described in [MS-TURN]. The default for UDP is port 3478. The default for TCP is port 443.
- The IP address and **FQDN** of each network interface of the associated TURN server.
- The server version with the value "2.0".

1.6 Applicability Statement

This protocol is used to provide a protocol client with security tokens for accessing the TURN server.

1.7 Versioning and Capability Negotiation

This protocol negotiates versions in the following manner:

1. The protocol client specifies the version in the XML body of the request.
2. If the server does not support the version requested by the protocol client, the request is rejected with a "Version Mismatch" error.

For more information, see section 3.1.5.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Audio/video edge authentication protocol messages MUST be transported using SIP SERVICE messages through a connection secured with TLS.

2.2 Message Syntax

The request and response messages of this protocol MUST be SIP SERVICE messages, as specified in [IETFDRAFT-SIPSOAP-00]. Protocol requests MUST include a **Content-Type header**, "application/msrtc-media-relay-auth+xml". The schema definition specified in [XMLSCHEMA1/2] and [XMLSCHEMA2/2] for the request and response messages is documented in section 6.

2.2.1 Request by the Client

The XML request sent by the protocol client MUST include exactly one **request** element.

2.2.1.1 request Element

2.2.1.1.1 request Element Definition

The schema definition for the **request** element is as follows.

```
<!-- REQUEST ELEMENT-->
<xs:element name="request" type="tns:requestType" />
```

2.2.1.1.2 requestType Type Definition

The schema definition for the **requestType** type is as follows.<1>

```
<!-- REQUEST TYPE-->
  <xs:complexType name="requestType">
    <xs:sequence>
      <!-- number of credentials requests will be bound within MRAS-->
      <xs:element name="credentialsRequest" type="tns:credentialsRequestType"
                  minOccurs="1"  maxOccurs="100"/>
    </xs:sequence>
    <xs:attribute name="requestID" type="tns:max64CharStringType" use="required"/>
    <xs:attribute name="version" type="tns:versionType" use="required"/>
    <xs:attribute name="to" type="tns:mrasUriType" use="required"/>
    <xs:attribute name="from" type="tns:mrasUriType" use="required"/>
    <xs:attribute name="route" type="tns:routeType" use="optional"
                  default="loadbalanced"/>
  </xs:complexType>
```

Attribute	Description
requestID	An ID that is used to identify the request. This can be used by the protocol client to associate the response with the request, in case the protocol client sent multiple simultaneous requests to the server with a unique **requestID**.
version	This is the version requested by the protocol client. The version MUST be as specified in section 3.1.5.2.1.

Attribute	Description
to	A restricted length URI (Uniform Resource Identifier) type that identifies the entity to which the request needs to be sent. It MUST be a SIP URI.
from	A restricted length URI type that identifies the entity that originated the request. It MUST be a SIP URI.
route	An optional attribute with a default value of "loadbalanced". If the value of the attribute is "directip", the server MUST return the IP address of the TURN server **endpoint**. If the attribute is not present or if the value is "loadbalanced", the server SHOULD return the fully qualified domain name (FQDN) (1) that resolves to the TURN server's IP address or the IP address of the TURN server.

2.2.1.1.3 Child Elements

This protocol implements the following **requestType** child elements.

2.2.1.1.3.1 credentialsRequest Element Definition

<xs:element name="**credentialsRequest**" type="tns:credentialsRequestType" minOccurs="1" maxOccurs="100" />The following table describes the **credentialsRequest** element.

Element	Description
credentialsRequest	One or more subelements of type **credentialsRequestType** MUST be present within the **requestType**. The **maxOccurs** attribute specifies the maximum number of subelements that MUST be present in the request. The schema allows 100 subelements. If the XML request by the protocol client adheres to the schema, except that the number of elements in the request exceeds **maxOccurs**, a "Request Too Large" error MUST be returned by the server. The server can implement policies that restrict the number of subelements that are allowed in the request.

2.2.1.1.3.2 credentialsRequestType Type Definition

The schema definition for the **credentialsRequestType** type is as follows.

```
<!-- CREDENTIALS REQUEST TYPE-->
  <xs:complexType name="credentialsRequestType">
    <xs:sequence>
      <xs:element name="identity" type="tns:max64kCharStringType" />
      <xs:element name="location" type="tns:locationType" minOccurs="0" />
      <xs:element name="duration" type="xs:positiveInteger" minOccurs="0" />
    </xs:sequence>
    <xs:attribute name="credentialsRequestID" type="tns:max64CharStringType"
                  use="required" />
  </xs:complexType>
```

Element	Description
identity	A restricted length string type that identifies the entity to which the token SHOULD be issued. This is a required field.
location	This field identifies the interface for which the tokens and FQDN/IP address and port information of the associated TURN server MUST be returned. If the attribute is not present,

Element	Description
	details related to both interfaces MUST be returned.
duration	The number of minutes for which the token needs to be valid. The server MUST use this value if it is less than the default value. If it is not included, the server MUST use the default value. The default duration is 480 minutes.

Attribute	Description
credentialsRequestID	A restricted length string type that identifies the **credentialsRequest** element within a **requestType**.

2.2.2 Response by the Server

The XML response sent by the server MUST include exactly one **response** element.

2.2.2.1 response Element

2.2.2.1.1 response Element Definition

The schema definition for the **response** element is as follows.

```
<!-- RESPONSE ELEMENT-->
<xs:element name="response" type="tns:responseType" />
```

2.2.2.1.2 responseType Type Definition

The schema definition for the **responseType** type is as follows.<2>

```
<!-- RESPONSE TYPE-->
<xs:complexType name="responseType">
  <xs:sequence>
    <xs:element name="credentialsResponse" type="tns:credentialsResponseType"
                minOccurs="0" maxOccurs="100"/>
  </xs:sequence>
  <xs:attribute name="requestID" type="tns:max64CharStringType"/>
  <xs:attribute name="version" type="tns:versionType" use="required"/>
  <xs:attribute name="serverVersion" type="tns:versionType" use="optional"/>
  <xs:attribute name="to" type="tns:mrasUriType"/>
  <xs:attribute name="from" type="tns:mrasUriType"/>
  <xs:attribute name="reasonPhrase" type="tns:reasonPhraseType" use="required"/>
</xs:complexType>
```

Attribute	Description
requestID	If the server cannot read the request because it does not adhere to the schema, this attribute is not present. If present, this attribute MUST have the same value as the **requestID** attribute in the request.
version	The version MUST be set as specified in section 3.1.5.2.2.

Attribute	Description
serverVersion	This is an optional attribute. If present, this SHOULD be the server version, as specified in section 1.5.
to	If the server cannot read the request because it does not adhere to the schema, this attribute is not present. If present, this attribute MUST have the same value as the **to** attribute in the request.
from	If the server cannot read the request because it does not adhere to the schema, this attribute is not present. If present, this attribute MUST have the same value as the **from** attribute in the request.
reasonPhrase	Specifies the reason for the error. It MUST be one of the values specified in the following schema. A detailed description of when each error message is thrown is provided in section 3.1.5. <xs:simpleType name="reasonPhraseType"> <xs:restriction base="xs:string"> <xs:enumeration value="OK"/> <xs:enumeration value="Request Malformed"/> <xs:enumeration value="Request Too Large"/> <xs:enumeration value="Not Supported"/> <xs:enumeration value="Server Busy"/> <xs:enumeration value="Time Out"/> <xs:enumeration value="Forbidden"/> <xs:enumeration value="Internal Server Error"/> <xs:enumeration value="Other Failure"/> <xs:enumeration value="Version Mismatch"/> </xs:restriction> </xs:simpleType>

2.2.2.1.3 Child Elements

2.2.2.1.3.1 credentialsResponse Element Definition

<xs:element name="credentialsResponse" type="tns:credentialsResponseType" minOccurs="0" maxOccurs="100" />The following table describes the **credentialsResponse** element.

Element	Description
credentialsResponse	For each **credentialsRequest** in the request from the protocol client, if the request succeeded, a **credentialsResponse** element MUST be included with the TURN server information and the token for the identity in the **credentialsRequest**. If the processing of the request does not succeed, there MUST be no **credentialsResponse** in the response sent by the server.

2.2.2.1.3.2 credentialsResponseType Type Definition

The schema definition for **credentialsResponseType** type is as follows.

```
<!--CREDENTIALS RESPONSE TYPE-->
  <xs:complexType name="credentialsResponseType">
    <xs:sequence>
      <xs:element name="credentials" type="tns:credentialsType" />
      <xs:element name="mediaRelayList" type="tns:mediaRelayListType" />
    </xs:sequence>
    <xs:attribute name="credentialsRequestID" type="tns:max64CharStringType" use="required"
/>
  </xs:complexType>
```

Attribute	Description
credentialsRequestID	This MUST be the same as the **credentialsRequestID** attribute in the request.

2.2.2.1.3.3 credentialsResponseType Child Elements

The following table describes the **credentialsResponseType** element.

Element	Description
credentials	It MUST contain the following subelements: ▪ a **username** and **password**, which form the security token needed by the protocol client to contact the TURN server. ▪ **duration** specifies how long the token is valid. It SHOULD contain an optional **realm** element that specifies which network segment the server belongs to.
mediaRelayList	It MUST contain the following subelements: ▪ **location** indicates internet or intranet. ▪ If the **route** attribute is set to "loadbalanced" in the **request** element, the subelement **hostName** MUST be present and SHOULD contain the FQDN that resolves to the TURN server IP address or the IP address of the TURN server. ▪ Otherwise, if the route attribute is set to "directip" in the **request** element, the subelement **directIPAddress** MUST be present and MUST contain the IP address of the TURN server. ▪ **tcpPort** specifies the port the TURN server is using to listen for TCP. ▪ **udpPort** specifies the port the TURN server is using to listen for UDP.

2.2.3 Basic Datatypes

This section lists the basic datatypes used in the XML request from the client and the response from the server.

2.2.3.1 routeType

The schema definition for the **routeType** datatype is as follows.

```
  <xs:simpleType name="routeType">
```

```
    <xs:restriction base="xs:string">
      <xs:enumeration value="loadbalanced"/>
      <xs:enumeration value="directip"/>
    </xs:restriction>
  </xs:simpleType>
```

This is a string type that can take only 2 possible values, "loadbalanced" or "directip".

2.2.3.2 locationType

The schema definition for the **locationType** datatype is as follows.

```
<xs:simpleType name="locationType">
  <xs:restriction base="xs:string">
     <xs:enumeration value="intranet"/>
     <xs:enumeration value="internet"/>
  </xs:restriction>
</xs:simpleType>
```

This is a string type that can take only 2 possible values, "intranet" or "internet".

2.2.3.3 credentialsType

The schema definition for the **credentialsType** datatype is as follows.

```
<xs:complexType name="credentialsType">
    <xs:sequence>
        <xs:element name="username" type="tns:max64kCharStringType" />
        <xs:element name="password" type="tns:max64kCharStringType" />
        <xs:element name="duration" type="xs:positiveInteger" />
        <xs:element name="realm" type="tns:max64kCharStringType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

The **credentialsType** datatype is a complex datatype that contains the following elements.

Element	Description
username	Type max64kCharStringType, as defined in section 2.2.3.9. Part of the security token sent by the server to the protocol client.
password	Type max64kCharStringType, as defined in section 2.2.3.9. Part of the security token sent by the server to the protocol client.
duration	A positive integer that specifies how long the token is valid.
realm	Optional. Specifies which network segment the server belongs to.

2.2.3.4 mediarelayType

The schema definition for the **mediarelayType** datatype is as follows.

```
<xs:complexType name="mediaRelayType">
   <xs:sequence>
      <xs:element name="location" type="tns:locationType"/>
      <xs:choice>
        <xs:element name="hostName" type="tns:hostNameType"/>
        <xs:element name="directIPAddress" type="tns:max64CharStringType"/>
       </xs:choice>
      <xs:element name="udpPort" type="xs:unsignedShort" minOccurs="0"/>
      <xs:element name="tcpPort" type="xs:unsignedShort" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>
```

The **mediarelayType** datatype is a complex datatype that specifies the details of the TURN server.

Element	Description
location	Type **locationType**, as defined in section 2.2.3.2. The location of the TURN server whose **IPAddress** and **Port** information are returned to the client.
Hostname	Type **hostNameType**, as defined in section 2.2.3.6. The FQDN or the IP address of the TURN server. Either this element or the **directIPAddress** element in the **mediarelayType** is present.
directIPAddress	Type **max64CharStringType**, as defined in section 2.2.3.8. The **IPAddress** of the TURN server.
udpPort	Type unsignedShort. The UDP port in which the TURN server is listening.
tcpPort	Type unsignedShort. The TCP port in which the TURN server is listening.

2.2.3.5 mediarelayListType

The schema definition for the **mediaRelayListType** datatype is as follows.

```
<xs:complexType name="mediaRelayListType">
    <xs:sequence>
      <xs:element name="mediaRelay" type="tns:mediaRelayType" minOccurs="1"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

The **mediarelayListType** datatype contains a list of elements of type **mediaRelayType**, as defined in section 2.2.3.4.

2.2.3.6 hostNameType

The schema definition for the **hostNameType** datatype is as follows.

```
<xs:simpleType name="hostNameType">
   <xs:restriction base="xs:string">
```

```
      <xs:pattern value="[a-zA-Z0-9_\-\.]*" />
      <xs:maxLength value="255"/>
   </xs:restriction>
</xs:simpleType>
```

The **hostNameType** datatype is a string type with a length restriction of 255 characters and the pattern of characters adhering to the regular expression defined previously in the schema.

2.2.3.7 reasonPhraseType

The schema definition for the **reasonPhraseType** datatype is as follows.

```
<xs:simpleType name="reasonPhraseType">
   <xs:restriction base="xs:string">
     <xs:enumeration value="OK"/>
     <xs:enumeration value="Request Malformed"/>
     <xs:enumeration value="Request Too Large"/>
     <xs:enumeration value="Not Supported"/>
     <xs:enumeration value="Server Busy"/>
     <xs:enumeration value="Time Out"/>
     <xs:enumeration value="Forbidden"/>
          <xs:enumeration value="Internal Server Error"/>
          <xs:enumeration value="Other Failure"/>
     <xs:enumeration value="Version Mismatch"/>
   </xs:restriction>
</xs:simpleType>
```

The **reasonPhraseType** datatype is a string type that can only take one of the values specified in the previous enumeration.

2.2.3.8 max64CharStringType

The schema definition for the **max64CharStringType** datatype is as follows.

```
 <xs:simpleType name='max64CharStringType'>
 <xs:restriction base='xs:string'>
    <xs:maxLength value='64'></xs:maxLength>
 </xs:restriction>
</xs:simpleType>
```

The **max64CharStringType** datatype is a string type with a length restriction of 64 characters.

2.2.3.9 max64kCharStringType

The schema definition for the **max64kCharStringType** datatype is as follows.

```
<xs:simpleType name='max64kCharStringType'>
<xs:restriction base='xs:string'>
   <xs:maxLength value='64000'></xs:maxLength>
</xs:restriction>
  </xs:simpleType>
```

The **max64kCharStringType** datatype is a string type with a length restriction of 64000 characters.

2.2.3.10 max8CharStringType

The schema definition for the **max8CharStringType** datatype is as follows.

```
<xs:simpleType name='max8CharStringType'>
    <xs:restriction base='xs:string'>
      <xs:maxLength value='8'></xs:maxLength>
    </xs:restriction>
</xs:simpleType>
```

The **max8CharStringType** datatype is a string type with a length restriction of 8 characters.

2.2.3.11 mrasUriType

The schema definition for the **mrasUriType** datatype is as follows.

```
<xs:simpleType name='mrasUriType'>
    <xs:restriction base='xs:anyURI'>
      <xs:maxLength value='10000'></xs:maxLength>
    </xs:restriction>
</xs:simpleType>
```

The **mrasUriType** datatype is defined as an **anyURI** type with a length restriction of 10000 characters.

2.2.3.12 versionType

The schema definition for the **versionType** datatype is as follows.

```
<xs:simpleType name='versionType'>
  <xs:restriction base='xs:string'>
    <xs:pattern value="[0-9]+\.[0-9]+"></xs:pattern>
    <xs:maxLength value="5"/>
  </xs:restriction>
</xs:simpleType>
```

The **versionType** datatype is a string type with a length restriction of 5 characters and a pattern adhering to the regular expression as defined previously in the schema.

3 Protocol Details

3.1 Server Details

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

When a SIP SERVICE request, as described in section 2.2.1, is received by the server, the request MUST be processed based on the rules given in section 3.1.5 and a SIP SERVICE response message, as described in section 2.2.2, MUST be sent back to the user. The message contains the token information if the processing of the request succeeded, or a detailed error description if the processing failed.

3.1.5 Message Processing Events and Sequencing Rules

The SIP SERVICE request described in section 2.2.1 MUST be the only message type that is accepted. The server MUST always respond with a SIP SERVICE response message, as described in section 2.2.2. The error codes indicate the type of the error in the request.

3.1.5.1 General Rules

When a request is received from the protocol client, it is processed based on the following rules:

If the request message type is not SIP SERVICE, the request MUST be rejected and an **UnsupportedMessageType** error response MUST be sent.

If the **contenttype** of the request is not equal to "application/msrtc-media-relay-auth+xml", an **UnsupportedContentType** error response MUST be sent. The SIP header of the response MUST include an **Accept** header with the value "application/msrtc-media-relay-auth+xml".

For the previous two rules, the server does not send a XML response in its body. The error codes described in this section indicate the nature of the problem. In the checks that follow, if an error condition occurs, an XML body adhering to conditions described in section 2.2.2 MUST be sent. The **reasonPhrase** of the error message MUST be as described in the following sections.

If the request does not adhere to schema rules, the request MUST be rejected with a **reasonPhrase** set to "RequestMalformed". If the request adheres to schema rules, except that the number of **credentialsRequest** in the XML request is greater than the **maxOccurs** attribute in the schema, the **reasonPhrase** in the error response MUST be "RequestTooLarge".

The server can implement policies that restrict the request that can be sent by the protocol client. If these policies are violated, the server MAY send an error response with a "Forbidden" **reasonPhrase**.

3.1.5.2 Version Validation

The **version** attribute in the request and response is of the type **versionType**, as defined in section 2.2.3.12<3>.

3.1.5.2.1 Validating the Version in the Request

The version in the request SHOULD<4> be either "2.0" or "1.0". Otherwise, the server SHOULD return an error response with reasonPhrase "Version Mismatch."

3.1.5.2.2 Setting the Version in the Response

If the server cannot read the protocol client's request because it does not adhere to the schema, the **version** in the response SHOULD<5> be the server version.

If the protocol client's request adheres to the schema:

- If a "VersionMismatch" error response is returned to the protocol client, the version supported by the server that is closer to the protocol client's version SHOULD be returned in the response.
- Otherwise, the **version** MUST be the version in the protocol client's request.

3.1.5.3 Checking the Attributes of the Request

The version validation is done as specified in section 3.1.5.2.1.

The server can include an optional **serverVersion** attribute in the response which, if present, SHOULD<6> be the same as the server version. If the client's request is valid and the version is "1.0" in the protocol client's request, the server MUST NOT include the **serverVersion** attribute.

The **from** and **to** attributes SHOULD be SIP **URIs**. If the server is not able to parse the URI, the server MUST send an error message with **reasonPhrase** with the value "RequestMalformed". If these attributes are valid, the values of these attributes are copied to the response.

3.1.5.4 Generating the credentialResponse

For each **credentialRequest** in the XML request:

- If the **duration** attribute is present in the request, the lifetime of the token MUST be calculated as the minimum of the duration specified by the protocol client and the preconfigured default lifetime value.
- The tokens, **username** and **password,** are generated using the lifetime calculated as stated previously and the **identity** specified by the protocol client in the XML request. The **username** and **password** generated MUST use Base 64 encoding and be included in the XML response.
- A **credentialResponse** MUST be created with the same **credentialsRequestID** as the **credentialsRequest** element in the protocol client's request. The token information MUST be in the **credentials** element and the information regarding the TURN server MUST be in the **mediaRelayList** element. If the **location** element was specified by the protocol client, the TURN server information related to that location only MUST be included in the **mediarelayList** element. Otherwise, both the intranet and internet information of the TURN server, as shown in the figure titled TURN server with two interfaces earlier in this document, MUST be included in the **mediaRelayList** element.

3.1.5.5 Populating Attributes of the Response

If the request was processed successfully without an error, the **reasonPhrase** MUST be set to "Ok", and the following apply:

- The **version** attribute in the response MUST be set as specified in section 3.1.5.2.2.
- The **from**, **to**, and **requestID** attributes MUST be included and MUST be equal to the appropriate values in the XML request.

If the **reasonPhrase** is "RequestMalformed", the **from**, **to**, and **requestID** attributes might not be included in the response. Otherwise, the **from**, **to**, and **requestID** attributes MUST be included and MUST be equal to the appropriate values in the XML request.

If an unexpected server error occurs during the processing of the request, the **reasonPhrase** MUST be "InternalServerError."

The SIP error codes that MUST be sent for the different response messages are specified in section 3.1.5.6.

3.1.5.6 Error Codes

The following table shows SIP error codes corresponding to the different **reasonPhrases** in the SIP responses. Some of these **reasonPhrases** are not currently used by the server. The unused values are indicated in the table with a "No" value in the column titled "Used now".

reasonPhrase in the response	SIP error codes	Used now
Ok	200	Yes
RequestMalformed	400	Yes
Forbidden	403	Yes
TimeOut	408	No
RequestTooLarge	413	Yes
ServerBusy	486	No
InternalServerError	500	Yes
OtherFailure	500	No
NotSupported	501	No
VersionMismatch	501	Yes

The following two error responses are used when the XML body of the response is not sent.

Response	SIP error codes	Used now
UnsupportedMessageType	501	Yes
UnsupportedContentType	415	Yes

3.1.5.7 Token Generation

The A/V Edge Server and the associated TURN server share two secret keys. These keys are used to create the security tokens **username** and **password**, as defined in section 2.2.3.3, which the protocol clients use to authenticate themselves with the TURN server. The server MUST include the lifetime of the token and the **identity** specified by the protocol client in the XML request while generating **username**. If the **duration** attribute is present in the request, the lifetime of the token MUST be calculated as the minimum of the duration specified by the protocol client and the configured lifetime value in the server. The server MUST base64 encode the **username** and **password** before including it in the XML response.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

The following examples illustrate the protocol request-response sequence.

4.1 Request from Client to Server

This section follows the product behavior as described in footnote <7>.

```
SERVICE sip: relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA
SIP/2.0

Via: SIP/2.0/TLS 10.56.65.225:7012

Max-Forwards: 70

From: <sip:client@contoso.com>;tag=09f804a3b1;epid=4906ed5712

To: <sip: relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA>

Call-ID: 7b25d8f0304c4655814760e624d7c3aa

CSeq: 1 SERVICE

Contact: <sip: client@contoso.com;opaque=user:epid:4WjlENtBIVSXgZyN1UZ6VgAA;gruu>

User-Agent: UCCP/2.0.6545.0 OC/2.0.6545.0 (Microsoft Office Communicator)

Proxy-Authorization: NTLM qop="auth", realm="SIP Communications Service", opaque="9574F9DA",
crand="5999c389", cnum="580", targetname="server1.contoso.com",
response="0100000064386630f99f6cb864399660"

Content-Type: application/msrtc-media-relay-auth+xml
Content-Length: 471

<request requestID="990512"
from="sip:client@contoso.com"
version="2.0"
to="sip: relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA"
xmlns="http://schemas.microsoft.com/2006/09/sip/mrasp"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <credentialsRequest credentialsRequestID="990512">
    <identity>sip:client@contoso.com</identity>
    <location>intranet</location>
    <duration>480</duration>
  </credentialsRequest>
</request>
```

4.2 Server Response to Client

This section follows the product behavior as described in footnote <8>.

```
SIP/2.0 200 OK

Authentication-Info: NTLM rspauth="0100000000000000C614C5BD64399660", srand="B8926199",
snum="862", opaque="9574F9DA", qop="auth", targetname="server1.contoso.com", realm="SIP
Communications Service"
```

```
Via: SIP/2.0/TLS 10.56.65.225:7012;ms-received-port=7012;ms-received-cid=19E00

FROM: "Client"<sip:client@contoso.com>;tag=09f804a3b1;epid=4906ed5712

TO: <sip:
relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA>;tag=554ef3a784

CSEQ: 1 SERVICE

CALL-ID: 7b25d8f0304c4655814760e624d7c3aa

CONTENT-LENGTH: 960

CONTENT-TYPE: application/msrtc-media-relay-auth+xml

SERVER: RTCC/3.0.0.0 Media Relay Authentication Service

ms-edge-proxy-message-trust: ms-source-type=EdgeProxyGenerated;ms-ep-fqdn=
relay.contoso.com@contoso.com;ms-source-verified-user=verified

<?xml version="1.0"?>
<response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
requestID="990512"
version="2.0"
serverVersion="2.0"
to="sip: relay.contoso.com@contoso.com;gruu;opaque=srvr:MRAS:OKPDbAVxIEKtPh2g624vPAAA"
from="sip:client@contoso.com"
reasonPhrase="OK"
xmlns="http://schemas.microsoft.com/2006/09/sip/mrasp">

  <credentialsResponse credentialsRequestID="990512">
<credentials>
<username>AQAgAIaoZr4EM8gBrxTJGY83uqdEgRXUunam2c+RID/vAJeJSL4YINbAYMvRAHeANv+Zew==</username>
      <password>35yqSF/p3A8gWXFHOC9YJA2kdvY=</password>
      <duration>480</duration>
    </credentials>
    <mediaRelayList>
      <mediaRelay>
        <location>intranet</location>
        <hostName>relay.contoso.com</hostName>
        <udpPort>3478</udpPort>
        <tcpPort>443</tcpPort>
      </mediaRelay>
    </mediaRelayList>
  </credentialsResponse>
</response>
```

5 Security

5.1 Security Considerations for Implementers

5.1.1 Keyed Hash Function

This protocol uses the **HMAC-SHA-1 hash** keyed hash function for generating tokens.

5.1.2 Underlying Transport

Because the security tokens sent in this protocol response are in plain text, all the protocol clients MUST communicate with the A/V Edge Server through a channel secured by TLS, as specified in section 2.1.

5.1.3 Authentication

Using this protocol, it is possible for unauthorized protocol clients to request tokens and obtain them. Also, a protocol client without proper authorization can send audio/video edge authentication protocol requests with different identities and obtain security tokens. This type of unauthorized activity precludes attempts by the TURN server to perform resource management based on protocol client identity that is present as the hash in the token. Consequently, the A/V Edge Server MUST authenticate the protocol clients and verify the request before distributing tokens.

5.2 Index of Security Parameters

Security Parameter	Section
shared-secret keys between the A/V Edge Server and the TURN server	3.1.5.2
HMAC-SHA-1 hash, as specified in [RFC2104], keyed hash algorithm	3.1.5.2
Token generation algorithm	3.1.5.6
TLS certificate, if TLS is used	2.1

6 Appendix A: MS-AVEDGEA Schema

This section follows the product behavior as described in footnote <9>.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://schemas.microsoft.com/2006/09/sip/mrasp"
xmlns:tns="http://schemas.microsoft.com/2006/09/sip/mrasp"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">

  <xs:annotation>
    <xs:documentation xml:lang="en">
      XML Schema for the MS-AVEDGEA
    </xs:documentation>
  </xs:annotation>


  <!-- REQUEST ELEMENT-->
  <xs:element name="request" type="tns:requestType" />


  <!-- RESPONSE ELEMENT-->
  <xs:element name="response" type="tns:responseType" />


  <!-- REQUEST TYPE-->
  <xs:complexType name="requestType">
    <xs:sequence>

      <xs:element name="credentialsRequest" type="tns:credentialsRequestType" minOccurs="1"
maxOccurs="100" />
    </xs:sequence>

    <xs:attribute name="requestID" type="tns:max64CharStringType" use="required" />
   <xs:attribute name="version" type="tns:max8CharStringType" use="required" />
    <xs:attribute name="to" type="tns:mrasUriType" use="required" />
    <xs:attribute name="from" type="tns:mrasUriType" use="required" />
    <xs:attribute name="route" type="tns:routeType" use="optional" default="loadbalanced" />
  </xs:complexType>


  <!-- RESPONSE TYPE-->
  <xs:complexType name="responseType">
    <xs:sequence>
      <xs:element name="credentialsResponse" type="tns:credentialsResponseType" minOccurs="0"
maxOccurs="100" />
    </xs:sequence>

    <xs:attribute name="requestID" type="tns:max64CharStringType" />
    <xs:attribute name="version" type="tns:max8CharStringType" use="required" />
     <xs:attribute name="serverVersion" type="tns:versionType" use="optional"/>
    <xs:attribute name="to" type="tns:mrasUriType" />
    <xs:attribute name="from" type="tns:mrasUriType" />
    <xs:attribute name="reasonPhrase" type="tns:reasonPhraseType" use="required" />
  </xs:complexType>


  <!-- CREDENTIALS REQUEST TYPE-->
```

```
  <xs:complexType name="credentialsRequestType">
    <xs:sequence>
      <xs:element name="identity" type="tns:max64kCharStringType" />
      <xs:element name="location" type="tns:locationType" minOccurs="0" />
      <xs:element name="duration" type="xs:positiveInteger" minOccurs="0" />
    </xs:sequence>
    <xs:attribute name="credentialsRequestID" type="tns:max64CharStringType" use="required"
/>
  </xs:complexType>

  <!--CREDENTIALS RESPONSE TYPE-->
  <xs:complexType name="credentialsResponseType">
    <xs:sequence>
      <xs:element name="credentials" type="tns:credentialsType" />
      <xs:element name="mediaRelayList" type="tns:mediaRelayListType" />
    </xs:sequence>
    <xs:attribute name="credentialsRequestID" type=" tns:max64CharStringType" use="required"
/>
  </xs:complexType>


  <!--ROUTE TYPE-->
  <xs:simpleType name="routeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="loadbalanced" />
      <xs:enumeration value="directip" />
    </xs:restriction>
  </xs:simpleType>


  <!--LOCATION TYPE-->
  <xs:simpleType name="locationType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="intranet" />
      <xs:enumeration value="internet" />
    </xs:restriction>
  </xs:simpleType>
  <!--CREDENTIALS TYPE-->
  <xs:complexType name="credentialsType">
    <xs:sequence>
        <xs:element name="username" type="tns:max64kCharStringType" />
        <xs:element name="password" type="tns:max64kCharStringType" />
        <xs:element name="duration" type="xs:positiveInteger" />
        <xs:element name="realm" type="tns:max64kCharStringType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <!--MEDIA RELAY LIST TYPE-->
  <xs:complexType name="mediaRelayListType">
    <xs:sequence>
      <xs:element name="mediaRelay" type="tns:mediaRelayType" minOccurs="1"
maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>


  <!--MEDIA RELAY TYPE-->
  <xs:complexType name="mediaRelayType">
    <xs:sequence>
      <xs:element name="location" type="tns:locationType" />
```

```
    <xs:choice>
      <xs:element name="hostName" type="tns:hostNameType" />
      <xs:element name="directIPAddress"     type="tns:max64CharStringType" />
    </xs:choice>
    <xs:element name="udpPort" type="xs:unsignedShort" minOccurs="0" />
    <xs:element name="tcpPort" type="xs:unsignedShort" minOccurs="0" />
  </xs:sequence>
</xs:complexType>


<!--DOMAIN NAME TYPE-->
<xs:simpleType name="hostNameType">
  <xs:restriction base="xs:string">
    <xs:pattern value="[a-zA-Z0-9_\-\.]*" />
    <xs:maxLength value="255" />
  </xs:restriction>
</xs:simpleType>


<!--RESPONSE REASON PHRASE-->
<xs:simpleType name="reasonPhraseType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="OK" />
    <xs:enumeration value="Request Malformed" />
    <xs:enumeration value="Request Too Large" />
    <xs:enumeration value="Not Supported" />
    <xs:enumeration value="Server Busy" />
    <xs:enumeration value="Time Out" />
    <xs:enumeration value="Forbidden" />
    <xs:enumeration value="Internal Server Error" />
    <xs:enumeration value="Other Failure" />
    <xs:enumeration value="Version Mismatch" />
  </xs:restriction>
</xs:simpleType>


<!--MAX 64 CHAR STRING TYPE-->
<xs:simpleType name="max64CharStringType">
  <xs:restriction base="xs:string">
    <xs:maxLength value="64">
    </xs:maxLength>
  </xs:restriction>
</xs:simpleType>


<!--MAX 64k CHAR STRING TYPE-->
<xs:simpleType name="max64kCharStringType">
  <xs:restriction base="xs:string">
    <xs:maxLength value="64000">
    </xs:maxLength>
  </xs:restriction>
</xs:simpleType>

<!--MAX 8 CHAR STRING TYPE-->
<xs:simpleType name="max8CharStringType">
  <xs:restriction base="xs:string">
    <xs:maxLength value="8">
    </xs:maxLength>
  </xs:restriction>
```

```
  </xs:simpleType>


  <!--mrasUri-->
  <xs:simpleType name="mrasUriType">
    <xs:restriction base="xs:anyURI">
      <xs:maxLength value="10000">
      </xs:maxLength>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

6.1 Office Communications Server 2007 Schema

This section follows the product behavior as described in footnote <10>.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
  targetNamespace="http://schemas.microsoft.com/2006/09/sip/mrasp"
  xmlns:tns="http://schemas.microsoft.com/2006/09/sip/mrasp"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      XML Schema for the MS-AVEDGEA
    </xs:documentation>
  </xs:annotation>

  <!-- REQUEST ELEMENT-->
  <xs:element name="request" type="tns:requestType" />

  <!-- RESPONSE ELEMENT-->
  <xs:element name="response" type="tns:responseType" />

  <!-- REQUEST TYPE-->
  <xs:complexType name="requestType">
    <xs:sequence>
      <!-- number of credentials requests will be bound within MRAS-->
      <xs:element name="credentialsRequest" type="tns:credentialsRequestType" minOccurs="1"
maxOccurs="100"/>
    </xs:sequence>
    <xs:attribute name="requestID" type="tns:max64CharStringType" use="required"/>
    <xs:attribute name="version" type="tns:versionType" use="required"/>
    <xs:attribute name="to" type="tns:mrasUriType" use="required"/>
    <xs:attribute name="from" type="tns:mrasUriType" use="required"/>
    <xs:attribute name="route" type="tns:routeType" use="optional" default="loadbalanced"/>
  </xs:complexType>

  <!-- RESPONSE TYPE-->
  <xs:complexType name="responseType">
    <xs:sequence>
      <xs:element name="credentialsResponse" type="tns:credentialsResponseType" minOccurs="0"
maxOccurs="100"/>
    </xs:sequence>
    <xs:attribute name="requestID" type="tns:max64CharStringType"/>
    <xs:attribute name="version" type="tns:versionType" use="required"/>
```

```
    <xs:attribute name="serverVersion" type="tns:versionType" use="optional"/>
    <xs:attribute name="to" type="tns:mrasUriType"/>
    <xs:attribute name="from" type="tns:mrasUriType"/>
    <xs:attribute name="reasonPhrase" type="tns:reasonPhraseType" use="required"/>
  </xs:complexType>

  <!-- CREDENTIALS REQUEST TYPE-->
  <xs:complexType name="credentialsRequestType">
    <xs:sequence>
      <xs:element name="identity" type="tns:max64kCharStringType" />
      <xs:element name="location" type="tns:locationType" minOccurs="0"/>
      <xs:element name="duration" type="xs:positiveInteger" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="credentialsRequestID" type="tns:max64CharStringType" use="required"/>
  </xs:complexType>

  <!-- RESPONSE TYPE-->
    <xs:complexType name="credentialsResponseType">
      <xs:sequence>
        <xs:element name="credentials" type="tns:credentialsType" />
        <xs:element name="mediaRelayList" type="tns:mediaRelayListType" />
      </xs:sequence>
      <xs:attribute name="credentialsRequestID" type="xs:string" use="required"/>
    </xs:complexType>

  <!--ROUTE TYPE-->
  <xs:simpleType name="routeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="loadbalanced"/>
      <xs:enumeration value="directip"/>
    </xs:restriction>
  </xs:simpleType>

    <!--LOCATION TYPE-->
    <xs:simpleType name="locationType">
      <xs:restriction base="xs:string">
        <xs:enumeration value="intranet"/>
        <xs:enumeration value="internet"/>
      </xs:restriction>
    </xs:simpleType>

    <!--CREDENTIALS TYPE-->
    <xs:complexType name="credentialsType">
      <xs:sequence>
        <xs:element name="username" type="tns:max64kCharStringType" />
        <xs:element name="password" type="tns:max64kCharStringType" />
        <xs:element name="duration" type="xs:positiveInteger" />
        <xs:element name="realm" type="tns:max64kCharStringType" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>

  <!--MEDIA RELAY LIST TYPE-->
  <xs:complexType name="mediaRelayListType">
    <xs:sequence>
      <xs:element name="mediaRelay" type="tns:mediaRelayType" minOccurs="1"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
```

```
<!--MEDIA RELAY TYPE-->
  <xs:complexType name="mediaRelayType">
    <xs:sequence>
      <xs:element name="location" type="tns:locationType"/>
      <xs:choice>
        <xs:element name="hostName" type="tns:hostNameType"/>
        <xs:element name="directIPAddress" type="tns:max64CharStringType"/>
       </xs:choice>
      <xs:element name="udpPort" type="xs:unsignedShort" minOccurs="0"/>
      <xs:element name="tcpPort" type="xs:unsignedShort" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <!--DOMAIN NAME TYPE-->
  <xs:simpleType name="hostNameType">
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z0-9_\-\.]*" />
      <xs:maxLength value="255"/>
    </xs:restriction>
  </xs:simpleType>

  <!--RESPONSE REASON PHRASE-->
  <xs:simpleType name="reasonPhraseType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="OK"/>
      <xs:enumeration value="Request Malformed"/>
      <xs:enumeration value="Request Too Large"/>
      <xs:enumeration value="Not Supported"/>
      <xs:enumeration value="Server Busy"/>
      <xs:enumeration value="Time Out"/>
      <xs:enumeration value="Forbidden"/>
      <xs:enumeration value="Internal Server Error"/>
      <xs:enumeration value="Other Failure"/>
      <xs:enumeration value="Version Mismatch"/>
    </xs:restriction>
  </xs:simpleType>

  <!--MAX 64 CHAR STRING TYPE-->
  <xs:simpleType name='max64CharStringType'>
    <xs:restriction base='xs:string'>
      <xs:maxLength value='64'></xs:maxLength>
    </xs:restriction>
  </xs:simpleType>

  <!--MAX 1024 CHAR STRING TYPE-->
  <xs:simpleType name='max64kCharStringType'>
    <xs:restriction base='xs:string'>
      <xs:maxLength value='64000'></xs:maxLength>
    </xs:restriction>
  </xs:simpleType>

  <!--MAX 8 CHAR STRING TYPE-->
  <xs:simpleType name='max8CharStringType'>
    <xs:restriction base='xs:string'>
      <xs:maxLength value='8'></xs:maxLength>
    </xs:restriction>
  </xs:simpleType>

  <!--mrasUri-->
```

```
    <xs:simpleType name='mrasUriType'>
      <xs:restriction base='xs:anyURI'>
        <xs:maxLength value='10000'></xs:maxLength>
      </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name='versionType'>
      <xs:restriction base='xs:string'>
        <xs:pattern value="[0-9]+\.[0-9]+"></xs:pattern>
        <xs:maxLength value="5"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:schema>
```

7 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® Office Communications Server 2007
- Microsoft® Office Communications Server 2007 R2
- Microsoft® Office Communicator 2007
- Microsoft® Office Communicator 2007 R2
- Microsoft® Lync™ Server 2010
- Microsoft® Lync™ 2010

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 2.2.1.1.2: Office Communications Server 2007, Office Communicator 2007: The requestType is the same as in the document except for the type of the version attribute.

<2> Section 2.2.2.1.2: Office Communications Server 2007, Office Communicator 2007: The **responseType** is the same as in the document except that the **serverVersion** attribute is not present and the type of the version attribute is "tns:max8CharStringType".

<xs:attribute name="version" type="tns:max8CharStringType" use="required"/>

<3> Section 3.1.5.2: Office Communications Server 2007, Office Communicator 2007: The **version** attribute is defined as a string with 8 characters.

<4> Section 3.1.5.2.1: Office Communications Server 2007, Office Communicator 2007: The **version** attribute in the client's request MUST be equal to the server version. Otherwise an error response with reasonPhrase "VersionMismatch" MUST be returned.

<5> Section 3.1.5.2.2: Office Communications Server 2007, Office Communicator 2007: The version in the response MUST be the server version.

<6> Section 3.1.5.3: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<7> Section 4.1: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<8> Section 4.2: Office Communications Server 2007, Office Communicator 2007: The value of the **serverVersion** attribute is "1.0".

<9> Section 6: Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<10> Section 6.1: Supported in Office Communications Server 2007, Office Communicator 2007 only.

8 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

9 Index

T

V