Internet Engineering Task Force (IETF)
Request for Comments: 6796
Category: Standards Track
ISSN: 2070-1721

V. Hilt
Bell Labs/Alcatel-Lucent
G. Camarillo
Ericsson
J. Rosenberg
jdrosen.net
D. Worley
Ariadne

December 2012

A User Agent Profile Data Set for Media Policy

Abstract

This specification defines an XML document format to describe the media properties of Session Initiation Protocol (SIP) sessions. Examples for media properties are the codecs or media types used in the session. This document also defines an XML document format to describe policies that limit the media properties of SIP sessions.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6796.

Table of Contents

1. Introduction

Within the Session Initiation Protocol (SIP) [RFC3261], "A Framework for Session Initiation Protocol (SIP) User Agent Profile Delivery" [RFC6080] and "A Framework for SIP Session Policies" [RFC6794] define mechanisms to convey session policies and configuration information from a network server to a user agent. An important piece of the information conveyed to the user agent relates to the media properties of the SIP sessions set up by the user agent. Examples for these media properties are the codecs and media types used, the media-intermediaries to be traversed, or the maximum bandwidth available for media streams.

This specification defines a document format for media properties of SIP sessions: the Media Policy Data Set Format (MPDF). This format can be used in two ways. First, it can be used to describe the properties of a given SIP session (e.g., the media types and codecs used). These MPDF documents are called session info documents and they are usually created based on the session description of a session. Second, the MPDF format can be used to define policies for SIP sessions in a session policy document. A session policy document defines properties for a session (e.g., the media types allowed in a session), independent of a specific session description.

If used with "A Framework for SIP Session Policies" [RFC6794], session info documents are used in conjunction with session-specific policies. A session info document is created by a user agent (UA) based on the current session description and submitted to the policy server. The policy server examines the session info document, modifies it if necessary (e.g., by removing video streams if video is not permitted), and returns the possibly modified session info document to the UA. Session policy documents, on the other hand, are used to describe session-independent policies that can be submitted to the UA independent of a specific session.

The two types of MPDF documents, session information and session policy documents, share the same set of XML elements to describe session properties. Since these elements are used in different contexts for session info and session policy documents, two different root elements exist for the two document types: <session-info> is the root element for session information documents and <session-policy> is the root element for session policy documents.

A user agent can receive multiple session policy documents from different sources. This can lead to a situation in which the user agent needs to apply multiple session policy documents to the same session. This standard specifies merging rules for those XML elements that can be present in session policy documents. It should

be noted that these merging rules are part of the semantics of a session policy XML element. User agents implement the merging rules as part of implementing the element semantics. As a consequence, it is not possible to build an entity that can mechanically merge two session policy documents without understanding the semantics of all elements in the input documents.

Merging rules are not needed for elements of session information documents since they are created by one source and describe a specific session.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Media Policy Data Set Format

This section discusses fundamental properties of the Media Policy Data Set Format (MPDF).

3.1. Namespace and Media Type

The MPDF format is based on XML [W3C.REC-xml-20081126]. An MPDF document MUST be well-formed and MUST be valid according to schemas, including extension schemas, available to the validator and applicable to the XML document. MPDF documents MUST be based on XML 1.0 and MUST be encoded using UTF-8.

MPDF makes use of XML namespaces [W3C.REC-xml-names-19990114]. The namespace URIs for elements defined in this specification are URNs [RFC2141], using the namespace identifier 'ietf' defined by [RFC2648] and extended by [RFC3688]. The namespace URN for the MPDF schema is:

urn:ietf:params:xml:ns:mediadataset

The media type for the Media Policy Data Set Format is:

application/media-policy-dataset+xml

3.2. Extensibility

The MPDF format can be extended using XML extension mechanisms if additional media properties are needed. In particular, elements from different XML namespaces MAY be present within a MPDF document for the purposes of extensibility; elements or attributes from unknown namespaces MUST be ignored.

3.3. Attributes

The following attributes can be used with elements of the MPDF format. The specification of each MPDF element lists which of these attributes can be used. If an element bears an attribute that may not be used with it, the user agent MUST ignore the attribute.

3.3.1. The 'visibility' Attribute

The attribute 'visibility' specifies whether or not the user agent is advised to display the property value to the user. This is used to hide setting values that the administrator may not want the user to see or know. The 'visibility' attribute has two possible values:

- visible: specifies that display of the property value is not restricted. This is the default value of the attribute if it is not specified.
- hidden: Specifies that the user agent is advised not to display the property value. Display of the property value may be allowed using special administrative interfaces, but it is not appropriate for the ordinary user.

3.3.2. The 'direction' Attributes

Some properties are unidirectional and only apply to messages or data streams transmitted into one direction. For example, a property for media streams can be restricted to outgoing media streams only. Unidirectional properties can be expressed by adding a 'direction' attribute to the respective element.

The 'direction' attribute can have the following values:

- recvonly: the property only applies to incoming streams.
- sendonly: the property only applies to outgoing streams.
- sendrecv: the property applies to streams in both directions. This is the default value that is used if the 'direction' attribute is omitted.

3.3.3. The 'q' Attribute

It is possible to express a preference for a certain value relative to the other values within a set of multiple values that are allowed within a property. For example, it is possible to express that the codecs G.711 and G.729 are allowed, but G.711 is preferred. Preferences are to be expressed by adding a 'q' attribute to a

property element. The 'q' attribute is only allowed in elements that specify allowed values (as opposed to elements that specify forbidden values).

The value of the 'q' attribute is a decimal number within the range of 0 to 1, inclusive, with two or fewer decimal places. An element with a higher 'q' value is preferred over one with a lower 'q' value.

3.3.4. The 'media-type' Attribute

The media-type attribute is used to define that an element only applies to streams of a certain media type, as defined in Section 8.2.1 of [RFC4566]. For example, it may only apply to audio streams. The value of the 'media-type' attribute MUST be the media type, such as audio, video, text, or application.

3.3.5. The 'label' Attribute

The label attribute is used to identify a specific media stream. The value of the label attribute is a token, whose syntax is defined in [RFC4574]. The token can be chosen freely; however, it MUST be unique among all <stream> elements in a session-info document.

3.3.6. The 'enabled' Attribute

The 'enabled' attribute specifies whether or not the user agent is allowed to establish a media stream. This boolean attribute has two possible values:

- yes: specifies that the media stream can be established. This is the default value of the attribute if it is not specified.
- no: specifies that the user agent MUST NOT establish the media stream.

4. Session Info Documents

Session info documents describe key properties of a SIP session such as the media streams used in the session. Session info documents are typically created based on a session description expressed using Session Description Protocol (SDP) [RFC4566] or an SDP offer/answer pair [RFC3264].

Session info documents can be used for session-specific policies [RFC6794]. In this usage, a UA creates a session info document based on its session description(s) and sends this document to the policy server. The policy server modifies this document according to the policies that apply to the described session and returns a version of

the session info document that is compliant to the policies. For example, if video streams are not permissible under current policies and the UA submits a session info document that contains a video stream, the policy server will disable (i.e., enabled="no") the video stream in the session info document that it returns to the UA.

Session info documents use the <session-info> root element. They use elements described in this section and common elements described in Section 6.

Elements that are only present in session info documents do not require merging rules. If used in the context of session-specific policies, session info documents are sent to one policy server at a time only; therefore, a UA does not need to merge multiple session info documents into one. A policy server needs to modify a session info document it has received according to its policies. The modification of session info documents is determined by the local policies of the policy server and is, thus, outside the scope of this standard.

A policy server can completely reject a session by returning a session info document with an empty <session-info> element:

```
<session-info></session-info>
```

4.1. Mapping between SDP and Session Info Documents

This section specifies how to map information in a session description or an SDP offer/answer pair [RFC3264] to session info documents. It also specifies how to map a session info document into a session description. Note that these mapping rules do not include rules for all elements that need to be present in a session info document or in a session description. That is, some of those elements are generated following their associated general rules (e.g., the general rules to generate SDP "v=" and "t=" lines).

A UA with a session description that needs to create a session info document uses the data in the session description and maps it following the rules below. A UA with an SDP offer/answer pair that needs to create a session info document uses the data that has been agreed in the offer/answer exchange.

A UA MUST create a separate <stream> element for each "m=" line in a session description or SDP offer/answer pair; the order of the <stream> elements corresponds to the order of the "m=" lines. For a session description, the UA MUST insert the media type from the "m=" line into a <media-type> element and MUST create a <codec> element for each codec listed in the "m=" line. For an SDP offer/answer

pair, the UA MUST insert a <codec> element for each of the codecs that were agreed upon for the particular stream in the offer/answer exchange. The <codec> elements MUST have 'q' attributes with values that decrease with the order the codecs are given in the "m=" line. (Other than the ordering restriction, the particular values used are not specified by this document.)

The UA MUST create a <local-host-port> element for each stream using the port taken from the "m=" line and the address from the corresponding "c=" line of the local session description. The UA SHOULD create a <remote-host-port> element using the port and address from the "m=" and "c=" lines for the same stream taken from the remote session description if this session description is available. (The local SDP is the one sent by the UA; the remote SDP is the one received from the remote UA.)

> The <remote-host-port> contains information that may be considered sensitive from a privacy standpoint. A UA configured not to disclose that information would not include the <remote-host-port> element in its session info documents.

The numeric value in a "b=CT:..." attribute in a session description is used to set the content of a <max-bw> element with the direction attribute value corresponding to which SDP contains the "b=" attribute.

The numeric value in a "b=AS:..." attribute at the session level in a session description is used to set the content of a <max-session-bw> element with the direction attribute value corresponding to the SDP which contains the "b=" attribute.

The numeric value in a "b=AS:..." attribute at the media level in a media description is used to set the content of a <max-stream-bw> element child of the appropriate <stream> element, with the direction attribute value corresponding to the SDP which contains the "b=" attribute.

An "a=label:..." attribute [RFC4574] is used to set the 'label' attribute of the appropriate <stream> element.

The mapping from a session info document to a session description follows the same rules in the reverse direction.

For any particular "m=" line, the codecs MUST be listed in decreasing order of the values of the 'q' attributes of the corresponding <codec> elements.

4.2. The <session-info> Element

The <session-info> element describes the properties of a specific SIP session. The <session-info> element MAY contain the optional <context> and <streams> elements, and multiple (including zero) <max-bw>, <max-session-bw>, <max-stream-bw>, <media-intermediaries>, and <qos-dscp> elements, as well as elements from other namespaces.

4.3. The <streams> Element

The <streams> element is a container that is used to describe the media streams used in a session. A <streams> element contains zero or more <stream> elements. Each <stream> element describes the properties (e.g., media type, codecs, and IP addresses and ports) of a single media stream.

4.3.1. The <stream> Element

The <stream> element describes a specific media stream. It contains the media type, codecs, and the hostname(s) or IP address(es) and port(s) of this stream.

The hostname(s) or IP address(es) and port number(s) of a stream correspond to the ones listed in the session description(s). A UA that generates a <stream> element MUST insert the hostname/port found in the local session description for this media stream into the local-host-port element. The UA SHOULD insert the hostname/port of the remote session description into the <remote-host-port> element, if the remote session description is available to the UA. If not, the UA generates a stream element that only contains the <local-host-port> element.

This element MAY have the direction, label, and enabled attributes (see Section 3.3).

The 'label' attribute is used to identify a specific media stream. The value of the label attribute is a token that is unique among all <stream> elements in a session-info document and whose syntax is defined in [RFC4566].

The 'enabled' attribute specifies whether or not the user agent is allowed to establish a media stream.

The <stream> element MUST contain one <media-type> element, one or more <codec> elements and one <local-host-port> element. The <stream> element MUST contain zero or one <remote-host-port> elements.

4.3.1.1. The <local-host-port> Element

The <local-host-port> element contains the hostname or IP address and the receiving port number of the media stream in the local session description. The hostname or IP address is separated from the port by a ":". An example is: "host.example.com:49562".

The hostname or IP address of element is found in the "c=" element for the stream in the local session description. The port number is found in the "m=" element.

4.3.1.2. The <remote-host-port> Element

The <remote-host-port> element is structured exactly as the <local-host-port> element. However, it identifies the hostname or IP address and receiving port number of the media stream in the remote session description.

4.4. The <media-intermediaries> Element

The <media-intermediaries> element expresses a policy for routing media streams through media intermediaries. The purpose of the <media-intermediaries> element is to tell the UA to send media streams through a chain of media intermediaries. The manner in which the UA arranges for a media stream to pass through the intermediaries depends on the type of intermediary.

The <media-intermediaries> element is a container that lists all media intermediaries to be traversed. Media intermediaries should be traversed in the order in which they appear in this list. The topmost entry should be traversed first, the last entry should be traversed last.

Different types of intermediaries exist. These intermediaries are not necessarily interoperable and it may not be possible to chain them in an arbitrary order. A <media-intermediaries> element SHOULD therefore only contain intermediary elements of the same type.

This element MAY have the 'direction' attribute (see Section 3.3).

Multiple <media-intermediaries> elements MUST NOT be present in a container unless each applies to a different set of streams (e.g., one <media-intermediaries> element for incoming and one for outgoing streams). The <media-intermediaries> element MUST contain one or more elements defining a specific media intermediary, such as <fixed-intermediary> or <turn-intermediary>.

Note: it is not intended that the <media-intermediaries> element replace connectivity discovery mechanisms such as Interactive Connectivity Establishment (ICE). Instead of finding media relays that provide connectivity, this element defines a policy for media intermediaries that should be traversed. The set of intermediaries defined in the <media-intermediaries> element and the ones discovered through ICE may overlap but don't have to.

4.4.1. The <fixed-intermediary> Element

A fixed intermediary relies on pre-configured forwarding rules. The user agent simply sends media to the first media intermediary listed. It can assume that this media intermediary has been pre-configured with a forwarding rule for the media stream and knows where to forward the packets. The configuration of forwarding rules in the intermediary must be done through other means.

The contents of a <fixed-intermediary> element MUST be echoed to all policy servers that provide policies for a session. That is, if multiple policy servers provide policies for the same session, this element needs to be forwarded to all of them, possibly in a second round of session-specific policy subscriptions as described in [RFC6794] in the "Contacting the Policy Server" section.

The <fixed-intermediary> element MUST contain one <int-host-port> element and MAY contain multiple optional <int-addl-port> elements.

4.4.1.1. The <int-host-port> Element

The <int-host-port> element contains the hostname or IP address and port number of a media intermediary. The UA uses this hostname/IP address and port to send its media streams to the intermediary. The hostname or IP address is separated from the port by a ":".

If a protocol uses multiple subsequent ports (e.g., RTP), the lowest port number SHOULD be included in the <int-host-port> element. All additional port numbers SHOULD be identified in <int-addl-port> elements.

4.4.1.2. The <int-addl-port> Element

If a protocol uses multiple subsequent ports (e.g., RTP), the lowest port number SHOULD be included in the <int-host-port> element. All additional port numbers SHOULD be identified in <int-addl-port> elements.

4.4.2. The <turn-intermediary> Element

The Traversal Using Relays around NAT (TURN) [RFC5766] protocol provides a mechanism for inserting a relay into the media path. Although the main purpose of TURN is NAT traversal, it is possible for a TURN relay to perform other media intermediary functionalities. The user agent establishes a binding on the TURN server and uses this binding to transmit and receive media.

The <turn-intermediary> element MUST contain one <int-host-port> element and MAY contain multiple optional <int-addl-port> elements and zero or one each of the <shared-secret>, <user>, and <transport> elements. If no <transport> element is present, UDP is assumed.

4.4.2.1. The <shared-secret> Element

The <shared-secret> element contains the shared secret needed to authenticate at the media intermediary.

4.4.2.2. The <user> Element

The <user> element contains the user ID needed to authenticate to the media intermediary.

4.4.2.3. The <transport> Element

The <transport> element contains the name of the transport to be used for communicating with the TURN server. This document defines the values "tcp" and "udp" for use in the <transport> element. Other specifications may define additional values.

4.4.3. The <msrp-intermediary> Element

The Message Session Relay Protocol (MSRP) Relay Extensions [RFC4976] define a means for incorporating relays into the media path of an MSRP [RFC4975] session. MSRP is explicitly designed for a variety of purposes, including policy enforcement.

The <msrp-intermediary> element MUST contain one <msrp-uri> element, and may contain zero or one of each of the <shared-secret> and <user> elements.

4.4.3.1. The <msrp-uri> Element

The <msrp-uri> element contains a URI that indicates the MSRP server to use for an intermediary. The UA uses this URI to authenticate with the MSRP relay, and then uses the URI it learns through that authentication process for any MSRP media it sends or receives. The URIs in the <msrp-uri> element MUST have a scheme of "msrps:".

5. Session Policy Documents

Session policy documents describe policies for SIP sessions. Session policy documents are independent of any specific session description and express general policies for SIP sessions. A session policy document is used to determine if a SIP session is policy-conformant and can be used to modify the session, if needed, to conform to the described policies.

Session policy documents can be used to encode session-independent policies [RFC6794]. In this usage, a policy server creates a session policy document and passes this document to a UA. The UA applies the policies defined to the SIP sessions it is establishing. For example, a session policy document can contain an element that prohibits the use of video. To set up a session that is compliant to this policy, a UA does not include the video media type in its SDP offer or answer.

Session policy documents use the <session-policy> root element. They use elements described in this section and common elements described in Section 6.

5.1. Merging Session Policies

A UA may receive session policy documents from multiple sources; multiple session policy documents can be merged into a single session policy document that expresses the logical AND of the policies.

5.1.1. Single Value Selection

Properties that have a single value (e.g., the maximum bandwidth allowed) require that a common value be determined for this property during the merging process. The merging rules for determining this value need to be defined individually for each element in the schema definition (e.g., select the lowest maximum bandwidth).

5.1.2. Merging Sets

The <media-types-allowed>, <media-types-excluded>, <codecs-allowed> and <codecs-excluded> elements are containers that hold a set of media-type/codec elements. The values defined in these containers MUST be merged to determine the set of media types/codecs that are permissible in a session. Note that for a particular codec, the <mime-parameter> element (see Section 6.2.2) allows identifying a particular encoding or profile of the codec. Therefore, when the <mime-parameter> element is present, what is allowed or excluded is the particular encoding or profile. Other encodings or profiles of the same codec are unaffected.

To merge the media-types-* and codecs-* containers, a UA MUST apply all containers it has received one after the other to the set of media types/codecs it supports. After applying media-types-*/ codecs-* elements, the UA has the list of media types/codecs that are allowed in a session. The containers MAY be applied in any order. However, each time a container is applied to the set of media types/ codecs allowed, this set MUST stay the same or be reduced. Media types/codecs cannot be added during this process.

The following example illustrates the merging process for two data sets. In this example, the UA supports the following set of audio codecs: PCMA, PCMU, and G729. After applying session policy document 1, the UA removes PCMA as it is disallowed by this policy. The remaining set of codecs is PCMU and G729. Session policy document 2 disallows all codecs that are not listed. After applying this policy, the set of codecs allowed is G729.

```
Session Policy Document 1:
<codecs-excluded>
  <codec><media-type-subtype>audio/PCMA</media-type-subtype></codec>
</codecs-excluded>

Session Policy Document 2:
<codecs-allowed>
  <codec><media-type-subtype>audio/PCMA</media-type-subtype></codec>
  <codec><media-type-subtype>audio/G729</media-type-subtype></codec>
</codecs-allowed>
```

It is possible that two session policy documents define non-overlapping sets of allowed media types or codecs. The resulting merged set would be empty, which is illegal according to the schema definition of the media-type/codec elements. This constitutes a conflict that cannot be resolved automatically. If these properties are enforced by both networks, the UA will not be able to set up a session.

The combined set of media types/codecs MUST again be valid and well-formed according to the schema definitions. A conflict occurs if the combined property set is not a well-formed document after the merging process is completed.

5.1.3. Local Policy Server Selection

Some properties require that only values from the local policy server are used. The local policy server is the policy server that is in the local domain of the user agent.

If policy documents are delivered through the configuration framework [RFC6080], the value received through a subscription using the "local-network" profile-type SHOULD used. Values received through other profile-type subscriptions SHOULD be discarded.

If policy documents are delivered through the session-specific policy mechanism [RFC6794] the value received from the policy server identified by the Local Policy Server URI SHOULD used. Values received from other policy servers SHOULD be discarded.

5.2. The <session-policy> Element

The <session-policy> element describes a policy that applies to SIP sessions. The <session-policy> element MAY contain the optional <context> and <local-ports> elements and multiple (including zero) <media-types-allowed>, <media-types-excluded>, <codecs-allowed>, <codecs-excluded>, <max-bw>, <max-session-bw>, <max-stream-bw>, and <qos-dscp> elements as well as elements from other namespaces.

5.3. The <media-types-allowed> Element

The <media-types-allowed> element is a container that is used to define the set of media types (e.g., audio, video) that are allowed in a session. All media types that are not listed in this container are not permitted in a session. A specific media type is allowed by adding the corresponding <media-type> element to this container.

This element MAY have the 'direction' and 'visibility' attributes (see Section 3.3).

Multiple <media-types-allowed> elements MUST NOT be present in a container element unless each applies to a different set of streams (e.g., one <media-types-allowed> element for incoming and one for outgoing streams). The <media-types-allowed> element MUST contain zero or more <media-type> elements.

A <media-types-allowed> element MUST NOT be used in a container that contains a <media-types-excluded> element. The absence of both elements in a container indicates no restrictions regarding media types.

Merging of session-policy documents: <media-types-allowed> containers are merged as described in "Merging Sets" Section 5.1.2.

5.4. The <media-types-excluded> Element

The <media-types-excluded> element is a container that is used to define the set of media types (e.g., audio, video) that are not permitted in a session. All media types that are not listed in this container are allowed and can be used in a session. A specific media type is excluded from a session by adding the corresponding <media-type> element to this container.

This element MAY have the 'direction' and 'visibility' attributes (see Section 3.3).

Multiple <media-types-excluded> elements MUST NOT be present in a container element unless each applies to a different set of streams (e.g., one <media-types-excluded> element for incoming and one for outgoing streams). The <media-types-excluded> element MUST contain zero or more <media-type> elements.

A <media-types-excluded> element MUST NOT be used in a container that contains a <media-types-allowed> element. The absence of both elements in a container indicates no restrictions regarding media types.

Merging of session-policy documents: <media-types-excluded> containers are merged as described in "Merging Sets" Section 5.1.2.

5.5. The <codecs-allowed> Element

The <codecs-allowed> element is a container that is used to define the set of codecs that may be used in a session. All codecs not listed in the <codecs-allowed> element are disallowed and MUST NOT be used in a session. A policy MUST allow the use of at least one codec per media type. A specific codec is allowed by adding the corresponding <codec> element to this container.

The <codecs-allowed> element MAY have the 'direction' and 'visibility' attributes (see Section 3.3).

Multiple <codecs-allowed> elements MUST NOT be present in a container element unless each applies to a different set of streams (e.g., one <codecs-allowed> element for incoming and one for outgoing streams). The <codecs-allowed> element MUST contain zero or more <codec> elements.

A <codecs-allowed> element MUST NOT be used in a container that contains a <codecs-excluded> element. The absence of both elements in a container indicates no restrictions regarding codecs.

Merging of session-policy documents: <codecs-allowed> containers are merged as described in "Merging Sets" Section 5.1.2.

5.6. The <codecs-excluded> Element

The <codecs-excluded> element is a container that is used to define the set of codecs that are disallowed in a session. All codecs not listed in the <codecs-excluded> element are permitted and MAY be used in a session. A specific codec is disallowed by adding the corresponding <codec> element to this container.

The <codecs-excluded> element MAY have the 'direction' and 'visibility' attributes (see Section 3.3).

Multiple <codecs-excluded> elements MUST NOT be present in a container element unless each applies to a different set of streams (e.g., one <codecs-excluded> element for incoming and one for outgoing streams). The <codecs-excluded> element MUST contain zero or more <codec> elements.

A <codecs-excluded> element MUST NOT be used in a container that contains a <codecs-allowed> element. The absence of both elements in a container indicates no restrictions regarding codecs.

Merging of session-policy documents: <codecs-excluded> containers are merged as described in "Merging Sets" Section 5.1.2.

5.7. The <local-ports> Element

Domains often require that a user agent only uses ports in a certain range for media streams. The <local-ports> element defines a policy for the ports a user agent can use for media. The value of this element consists of the decimal representation of a start port number and an end port number, separated by a hyphen ("-"). The start/end port numbers are the first/last port numbers that can be used, that is, the range is inclusive. The start/end port numbers must be in the range 1 to 65535 (inclusive).

As with other policy elements, there are values of the <local-ports> element that allow no sessions. This happens if the start port number is greater than the end port number.

The default value for <local-ports> is "1-65535".

This element MAY have the 'visibility' attribute (see Section 3.3).

Merging of session-policy documents: the permitted ranges specified by the two policies are set-intersected. If the resulting set is empty, the resulting <local-ports> element value MUST be any allowed value with a start port number greater than the end port number.

6. Common Media Policy Data Set Elements

This section describes common XML elements that are used in session info and session policy documents to encode the media properties of SIP sessions.

6.1. The <media-type> Element

The <media-type> element identifies a specific media type. The value of this element MUST be the name of a media type, as defined in Section 8.2.1 of [RFC4566], such as audio, video, text, or application.

This element MAY have the 'q' attribute (see Section 3.3).

If used in a session policy document inside a <media-types-allowed> element, the media types defined MAY be used in a session. If used in a session policy document inside a <media-types-excluded> element, the media types defined MUST NOT be used in a session.

6.2. The <codec> Element

The <codec> element identifies a specific codec. The content of this element MUST be a media type and subtype (e.g., audio/PCMA [RFC4856] or video/H263 [RFC4629]), possibly with parameters.

The <codec> element MAY have the 'q' attribute (see Section 3.3).

If used in a session policy document inside a <codecs-allowed> element, the codec defined MAY be used in a session. If used in a session policy document inside a <codecs-excluded> element, the codec defined MUST NOT be used in a session.

The <codec> element MUST contain one <media-type-subtype> element and MAY contain multiple optional <mime-parameter> elements.

6.2.1. The <media-type-subtype> Element

The <media-type-subtype> element contains a media type and subtype that identifies a media format [RFC4566] (e.g., a codec). For audio and video streams, the value of this element MUST be a media type and subtype that is registered as an RTP Payload Type [RFC4855] separated by a forward slash ("/"), e.g., audio/PCMA, audio/G726-16 [RFC4856], or video/H263 [RFC4629]. For other media types, SDP sometimes encodes the actual media format as part of the transport protocol field (e.g., TCP/MSRP [RFC4975] and TCP/TLS/BFCP [RFC4583]). In these cases, this element MUST contain the media type and the media format part (e.g., message/msrp and application/bfcp).

6.2.2. The <mime-parameter> Element

The <mime-parameter> element may be needed for some codecs to identify a particular encoding or profile. The value of this element MUST be a name-value pair containing the name and the value of a media type parameter for the codec [RFC4855]. The name and value are separated by an equals sign ("="). For example, the parameter "profile=0" can be used to specify a specific profile for the codec video/H263-2000 [RFC4629].

6.3. The <max-bw> Element

The <max-bw> element defines the overall maximum bandwidth in kilobits per second (i.e., 1024 bits per second) an entity can/will use for media streams at any point in time. It defines an upper limit for the total bandwidth an entity can/will use for the transmission of media streams. The limit corresponds to the sum of the maximum session bandwidth of all sessions a UA may set up in parallel.

The bandwidth limit given in the <max-bw> element includes the bandwidth needed for lower-layer transport and network protocols (e.g., UDP and IP).

The <max-bw> element MAY have the 'direction' attribute (see Section 3.3).

If used in a <session-policy> element, the <max-bw> element MAY also have the 'visibility' attribute (see Section 3.3).

If the <max-bw> element occurs multiple times in a container element, each instance MUST apply to a different set of media streams (i.e., one <max-bw> element for outgoing and one for incoming streams).

Merging of session-policy documents: the lowest <max-bw> value MUST be used.

6.4. The <max-session-bw> Element

The <max-session-bw> element defines the maximum bandwidth in kilobits per second (i.e., 1024 bits per second) an entity can/will use for media streams in the described session. It defines an upper limit for the total bandwidth of a single session. This limit corresponds to the sum of the maximum stream bandwidth of all media streams in a session.

The bandwidth limit given in the <max-session-bw> element includes the bandwidth needed for lower-layer transport and network protocols (e.g., UDP and IP).

The <max-session-bw> element MAY have the 'direction' attribute (see Section 3.3).

If used in a <session-policy> element, the <max-session-bw> element MAY also have the 'visibility' attribute (see Section 3.3).

If the <max-session-bw> element occurs multiple times in a container element, each instance MUST apply to a different set of media streams (i.e., one <max-session-bw> element for outgoing and one for incoming streams).

Merging of session-policy documents: the lowest <max-session-bw> value MUST be used.

6.5. The <max-stream-bw> Element

The <max-stream-bw> element defines the maximum bandwidth in kilobits per second (i.e., 1024 bits per second) an entity can/will use for each media stream in the described session.

The bandwidth limit given in the <max-stream-bw> element includes the bandwidth needed as encapsulated in IP (i.e., the RTP, UDP, and IP overheads are included).

The <max-stream-bw> element MAY have the 'direction' and 'media-type' attributes (see Section 3.3).

If used in a <session-policy> element, the <max-stream-bw> element MAY also have the visibility attribute (see Section 3.3).

If used in a <session-info> element, the <max-stream-bw> element MAY also have the label attribute.

The media-type attribute is used to define that the <max-stream-bw> element only applies to streams of a certain media type (e.g., audio streams).

The <max-stream-bw> element is used to define a bandwidth limit for a specific media stream. The use of this attribute requires that the <stream> element that represents the media stream to which this bandwidth limit applies also has a 'label' attribute. A <max-stream-bw> element with a 'label' attribute applies only to the stream element that has a 'label' attribute with the same value. If no matching <stream> element exists, then the <max-stream-bw> element MUST be ignored.

If the <max-stream-bw> element occurs multiple times in a container element, each instance MUST apply to a different set of media streams (i.e., one <max-stream-bw> element for outgoing and one for incoming streams).

Merging of session-policy documents: the lowest <max-stream-bw> value MUST be used.

6.6. The <qos-dscp> Element

The <qos-dscp> element contains a Differentiated Services Codepoint (DSCP) [RFC2474] value that should be used to populate the IP DS field of media packets. The <qos-dscp> contains a decimal integer value that represents a 6-bit field and therefore ranges from 0 to 63.

This element MAY have the 'direction' and 'media-type' attributes (see Section 3.3)).

If used in a <session-policy> element, the <qos-dscp> element MAY also have the 'visibility' attribute (see Section 3.3).

The 'media-type' attribute is used to specify that the <qos-dscp> element only applies to streams of a certain media type (e.g., audio streams).

The <qos-dscp> element is optional and MAY occur multiple times inside a container. If the <qos-dscp> element occurs multiple times, each instance MUST apply to a different media stream (i.e., one <qos-dscp> element for audio and one for video streams).

Merging of session-policy documents: the local domain of the user agent has precedence over other domains and its DSCP value MUST be used. During the merging process, <qos-dscp> element values from local policy server selected as described in "Local Policy Server Selection" Section 5.1.3 are used.

6.7. The <context> Element

The <context> element provides context information about a session policy or session information document.

The <context> element MAY contain multiple <contact> elements and one <info> element. It can also contain optional <policy-server-URI> and <token> elements.

If used in a <session-info> element, the <context> element MAY also contain a <request-URI> element.

Merging of session-policy documents: the resulting <context> element MUST be determined by local policy.

6.7.1. The <policy-server-URI> Element

The <policy-server-URI> element contains the URI (including the URI scheme) of the policy server that has issued this policy.

6.7.2. The <contact> Element

The <contact> element contains a URI that is a contact address (e.g., a SIP URI or mailto URI) by which a human representative of the issuer of this document can be reached.

6.7.3. The <info> Element

The <info> element provides a short textual description of the policy or session that should be intelligible to the human user.

6.7.4. The <request-URI> Element

The <request-URI> element contains the request-URI (including the URI scheme) of the dialog-initiating request of the session.

The <request-URI> element is only permitted inside <session-info> documents and, thus, MUST NOT be included in session policy documents.

6.7.5. The <token> Element

The <token> element provides a mechanism for a policy server to return an opaque string to a UA. Such a string is sometimes needed to construct a Policy-ID header that ensures that all policy requests concerning a single session are routed to the same policy server. The use of this token is described in "A Framework for Session Initiation Protocol (SIP) Session Policies" [RFC6794]. The syntax for the token value is defined in Section 4.4.5.1 of RFC 6794 [RFC6794], which builds on the syntax defined in Section 25.1 of RFC 3261 [RFC3261]. (Note that the token value is encodable as a SIP URI parameter value, although some characters may require escaping).

6.8. Other Session Properties

A number of additional elements have been proposed for a media property language. These elements are deemed to be outside the scope of this format. However, they may be defined in extensions of MPDF or other profile data sets.

- maximum number of streams
- maximum number of sessions
- maximum number of streams per session
- external address and port
- media transport protocol
- outbound proxy
- SIP methods
- SIP option tags
- SIP transport protocol
- body disposition
- body format
- body encryption

7. Examples

7.1. Session Policy Documents

The following example is a session policy document that allows the use of audio and video and prohibits the use of other media types. It allows the use of any codec except G.723 and G.729.

```
<session-policy xmlns="urn:ietf:params:xml:ns:mediadataset">
  <context>
 <policy-server-URI>sips:policy@biloxi.example.com</policy-server-URI>
    <contact>sip:policy_manager@example.com</contact>
    <info>Access network policies</info>
  </context>
  <media-types-allowed>
    <media-type>audio</media-type>
    <media-type>video</media-type>
  </media-types-allowed>
  <codecs-excluded>
    <codec>
      <media-type-subtype>audio/G729</media-type-subtype>
    </codec>
    <codec>
      <media-type-subtype>audio/G723</media-type-subtype>
    </codec>
  </codecs-excluded>
</session-policy>
```

7.2. Session Information Documents

The following examples contain session descriptions and the session information documents that represent these sessions.

7.2.1. Example 1

In this example, a session info document is created based on one session description. This session info document would be created, for example, by a UA that has composed an offer and is now contacting a policy server.

Local session description:

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.somewhere.example
s=
c=IN IP4 host.somewhere.example
t=0 0
m=audio 49562 RTP/AVP 0 1 3
```

```
a=rtpmap:0 PCMU/8000
a=rtpmap:1 1016/8000
a=rtpmap:3 GSM/8000
m=video 51234 RTP/AVP 31 34
a=rtpmap:31 H261/90000
a=rtpmap:34 H263/90000
```

MPDF document:

```
<session-info xmlns="urn:ietf:params:xml:ns:mediadataset">
  <context>
    <contact>sip:alice@somewhere.example</contact>
    <info>session information</info>
  </context>
  <streams>
    <stream>
      <media-type>audio</media-type>
      <codec q="1.0">
        <media-type-subtype>audio/PCMU</media-type-subtype>
      </codec>
      <codec q="0.9">
        <media-type-subtype>audio/1016</media-type-subtype>
      </codec>
      <codec q="0.8">
        <media-type-subtype>audio/GSM</media-type-subtype>
      </codec>
      <local-host-port>host.somewhere.example:49562</local-host-port>
    </stream>
    <stream>
      <media-type>video</media-type>
      <codec q="1.0">
        <media-type-subtype>video/H261</media-type-subtype>
      </codec>
      <codec q="0.9">
        <media-type-subtype>video/H263</media-type-subtype>
      </codec>
      <local-host-port>host.somewhere.example:51234</local-host-port>
    </stream>
  </streams>
</session-info>
```

7.2.2. Example 2

In this example, a session info document is created that represents two session descriptions (i.e., an offer and answer). This session info document would be created, for example, by a UA that has received an answer from another UA and is now contacting a policy server.

```
   Local session description:

   v=0
   o=alice 2890844526 2890844526 IN IP4 host.somewhere.example
   s=
   c=IN IP4 host.somewhere.example
   t=0 0
   m=audio 49562 RTP/AVP 0 1 3
   a=rtpmap:0 PCMU/8000
   a=rtpmap:1 1016/8000
   a=rtpmap:3 GSM/8000
   m=video 51234 RTP/AVP 31 34
   a=rtpmap:31 H261/90000
   a=rtpmap:34 H263/90000

   Remote session description:

   v=0
   o=bob 2890844730 2890844730 IN IP4 host.anywhere.example
   s=
   c=IN IP4 host.anywhere.example
   t=0 0
   m=audio 52124 RTP/AVP 0 3
   a=rtpmap:0 PCMU/8000
   a=rtpmap:3 GSM/8000
   m=video 50286 RTP/AVP 31
   a=rtpmap:31 H261/90000

   MPDF document that represents the local and the remote session
   description:

  <session-info xmlns="urn:ietf:params:xml:ns:mediadataset">
    <context>
      <contact>sip:alice@somewhere.example</contact>
      <info>session information</info>
    </context>
    <streams>
      <stream>
        <media-type>audio</media-type>
        <codec q="1.0">
          <media-type-subtype>audio/PCMU</media-type-subtype>
        </codec>
        <codec q="0.9">
          <media-type-subtype>audio/GSM</media-type-subtype>
        </codec>
        <local-host-port>host.somewhere.example:49562</local-host-port>
        <remote-host-port>host.anywhere.example:52124</remote-host-port>
      </stream>
```

```
      <stream>
        <media-type>video</media-type>
        <codec q="1.0">
          <media-type-subtype>video/H261</media-type-subtype>
        </codec>
        <local-host-port>host.somewhere.example:51234</local-host-port>
        <remote-host-port>host.anywhere.example:50286</remote-host-port>
      </stream>
    </streams>
  </session-info>
```

The following MPDF document is a modified version of the above document, which can be returned by a policy server. This document reflects a policy that defines a maximum session bandwidth of 192 kbit and a maximum bandwidth for the H261 video stream of 128 kbit.

```
  <session-info xmlns="urn:ietf:params:xml:ns:mediadataset">
    <context>
      <contact>sip:alice@somewhere.example</contact>
      <info>modified session information</info>
    </context>
    <streams>
      <stream label='1'>
        <media-type>audio</media-type>
        <codec q="1.0">
          <media-type-subtype>audio/PCMU</media-type-subtype>
        </codec>
        <codec q="0.9">
          <media-type-subtype>audio/GSM</media-type-subtype>
        </codec>
        <local-host-port>host.somewhere.example:49562</local-host-port>
        <remote-host-port>host.anywhere.example:52124</remote-host-port>
      </stream>
      <stream label='2'>
        <media-type>video</media-type>
        <codec q="1.0">
          <media-type-subtype>video/H261</media-type-subtype>
        </codec>
        <local-host-port>host.somewhere.example:51234</local-host-port>
        <remote-host-port>host.anywhere.example:50286</remote-host-port>
      </stream>
    </streams>
    <max-stream-bw label='2'>128</max-stream-bw>
    <max-session-bw>192</max-session-bw>
  </session-info>
```

8. RELAX NG Definition

```
<?xml version="1.0"?>
    <grammar xmlns="http://relaxng.org/ns/structure/1.0"
     ns="urn:ietf:params:xml:ns:mediadataset"
     datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">

       <start>
          <choice>
                <element name="session-info">
                    <interleave>
                    <optional>
                        <ref name="ElementStreams"/>
                    </optional>
                    <zeroOrMore>
                        <ref name="ElementMaxBandwidth"/>
                    </zeroOrMore>
                    <zeroOrMore>
                        <ref name="ElementMaxSessionBandwidth"/>
                    </zeroOrMore>
                    <zeroOrMore>
                        <ref name="ElementMaxStreamBandwidth"/>
                    </zeroOrMore>
                    <zeroOrMore>
                        <ref name="ElementMediaIntermediaries"/>
                    </zeroOrMore>
                    <zeroOrMore>
                        <ref name="ElementQoSDSCP"/>
                    </zeroOrMore>
                    <zeroOrMore>
                        <ref name="ElementAny"/>
                    </zeroOrMore>
                    </interleave>
                </element>

                <element name="session-policy">
                    <interleave>
                    <optional>
                        <ref name="ElementContext"/>
                    </optional>
                    <optional>
                        <ref name="ElementLocalPorts"/>
                    </optional>
                    <zeroOrMore>
                        <ref name="ElementMediaTypesAllowed"/>
                    </zeroOrMore>
                    <zeroOrMore>
                        <ref name="ElementMediaTypesExcluded"/>
```

```
                    </zeroOrMore>
                    <zeroOrMore>
                        <ref name="ElementCodecsAllowed"/>
                    </zeroOrMore>
                    <zeroOrMore>
                        <ref name="ElementCodecsExcluded"/>
                    </zeroOrMore>
                    <zeroOrMore>
                        <ref name="ElementMaxBandwidth"/>
                    </zeroOrMore>
                    <zeroOrMore>
                        <ref name="ElementMaxSessionBandwidth"/>
                    </zeroOrMore>
                    <zeroOrMore>
                        <ref name="ElementMaxStreamBandwidth"/>
                    </zeroOrMore>
                    <zeroOrMore>
                        <ref name="ElementQoSDSCP"/>
                    </zeroOrMore>
                    <zeroOrMore>
                        <ref name="ElementAny"/>
                    </zeroOrMore>
                    </interleave>
               </element>
        </choice>
    </start>

    <define name="ElementMediaTypesAllowed">
        <element name="media-types-allowed">
            <ref name="PolicyGeneralAttributes"/>
            <zeroOrMore>
               <ref name="ElementMediaType"/>
            </zeroOrMore>
        </element>
    </define>

    <define name="ElementMediaTypesExcluded">
        <element name="media-types-excluded">
            <ref name="PolicyGeneralAttributes"/>
             <zeroOrMore>
               <ref name="ElementMediaType"/>
            </zeroOrMore>
        </element>
    </define>

    <define name="ElementMediaType">
        <element name="media-type">
            <data type="string" />
```

```
            <optional>
              <ref name="AttributeQ"/>
            </optional>
            <optional>
              <ref name="AttributeGeneric"/>
            </optional>
        </element>
    </define>

    <define name="ElementCodecsAllowed">
        <element name="codecs-allowed">
          <ref name="PolicyGeneralAttributes"/>
            <zeroOrMore>
               <ref name="ElementCodec"/>
            </zeroOrMore>
        </element>
    </define>

    <define name="ElementCodecsExcluded">
        <element name="codecs-excluded">
          <ref name="PolicyGeneralAttributes"/>
            <zeroOrMore>
               <ref name="ElementCodec"/>
            </zeroOrMore>
        </element>
    </define>

    <define name="ElementCodec">
        <element name="codec">
            <optional>
              <ref name="AttributeQ"/>
            </optional>
            <optional>
              <ref name="AttributeGeneric"/>
            </optional>
            <element name="media-type-subtype">
              <data type="string" />
            </element>
            <zeroOrMore>
              <element name="mime-parameter">
                <data type="string" />
              </element>
            </zeroOrMore>
        </element>
    </define>
```

```
<define name="ElementStreams">
    <element name="streams">
        <optional>
          <ref name="AttributeGeneric"/>
        </optional>
        <zeroOrMore>
          <ref name="ElementStream"/>
        </zeroOrMore>
    </element>
</define>

<define name="ElementStream">
    <element name="stream">
        <optional>
          <ref name="AttributeDirection"/>
        </optional>
        <optional>
          <ref name="AttributeLabel"/>
        </optional>
        <optional>
          <ref name="AttributeEnabled"/>
        </optional>
        <optional>
          <ref name="AttributeGeneric"/>
        </optional>
        <ref name="ElementMediaType"/>
        <oneOrMore>
          <ref name="ElementCodec"/>
        </oneOrMore>
        <element name="local-host-port">
          <data type="string" />
        </element>
        <optional>
          <element name="remote-host-port">
            <data type="string" />
          </element>
        </optional>
    </element>
</define>

<define name="ElementMaxBandwidth">
   <element name="max-bw">
        <data type="integer" />
        <ref name="PolicyGeneralAttributes"/>
    </element>
</define>
```

```
<define name="ElementMaxSessionBandwidth">
    <element name="max-session-bw">
        <data type="integer" />
        <ref name="PolicyGeneralAttributes"/>
    </element>
</define>

<define name="ElementMaxStreamBandwidth">
    <element name="max-stream-bw">
        <data type="integer" />
        <ref name="PolicyGeneralAttributes"/>
        <optional>
          <ref name="AttributeMediaType"/>
        </optional>
        <optional>
          <ref name="AttributeLabel"/>
        </optional>
    </element>
</define>

<define name="ElementMediaIntermediaries">
    <element name="media-intermediaries">
       <ref name="PolicyGeneralAttributes"/>
        <oneOrMore>
          <choice>
            <element name="fixed-intermediary">
              <element name="int-host-port">
                <data type="string" />
              </element>
              <zeroOrMore>
                <element name="int-addl-port">
                  <data type="integer" />
                </element>
              </zeroOrMore>
            </element>

            <element name="turn-intermediary">
              <element name="int-host-port">
                <data type="string" />
              </element>
              <zeroOrMore>
                <element name="int-addl-port">
                  <data type="integer" />
                </element>
              </zeroOrMore>
              <zeroOrMore>
                <element name="shared-secret">
                  <data type="string" />
```

```
                    </element>
                  </zeroOrMore>
                </element>
              </choice>
            </oneOrMore>
        </element>
    </define>

    <define name="ElementQoSDSCP">
        <element name="qos-dscp">
            <data type="integer" />
            <ref name="PolicyGeneralAttributes"/>
            <optional>
              <ref name="AttributeMediaType"/>
            </optional>
        </element>
    </define>

    <define name="ElementLocalPorts">
        <element name="local-ports">
            <data type="string" />
            <interleave>
              <optional>
                <ref name="AttributeVisibility"/>
              </optional>
              <optional>
                <ref name="AttributeGeneric"/>
              </optional>
           </interleave>
        </element>
    </define>

    <define name="ElementContext">
        <element name="context">
            <interleave>
            <optional>
              <element name="info">
                <data type="string" />
              </element>
            </optional>
             <optional>
             <element name="policy-server-URI">
                <data type="string" />
              </element>
            </optional>
             <optional>
             <element name="token">
                <data type="token" />
```

```
              </element>
            </optional>
            <optional>
             <element name="request-URI">
                <data type="string" />
              </element>
            </optional>
             <zeroOrMore>
              <element name="contact">
                 <data type="string" />
              </element>
            </zeroOrMore>
            </interleave>
        </element>
     </define>

     <define name="PolicyGeneralAttributes">
              <optional>
                <ref name="AttributeVisibility"/>
              </optional>
              <optional>
                <ref name="AttributeDirection"/>
              </optional>
              <optional>
                <ref name="AttributeGeneric"/>
              </optional>
     </define>


    <define name="AttributeVisibility">
        <attribute name="visibility">
          <choice>
            <value>hidden</value>
            <value>visible</value>
          </choice>
        </attribute>
    </define>

    <define name="AttributeDirection">
        <attribute name="direction">
          <choice>
            <value>sendonly</value>
            <value>recvonly</value>
            <value>sendrecv</value>
          </choice>
        </attribute>
    </define>
```

```
<define name="AttributeQ">
    <attribute name="q">
      <data type="decimal" />
    </attribute>
</define>

<define name="AttributeMediaType">
    <attribute name="media-type">
      <data type="string" />
    </attribute>
</define>

<define name="AttributeLabel">
    <attribute name="label">
      <data type="string" />
    </attribute>
</define>

<define name="AttributeEnabled">
    <attribute name="enabled">
      <data type="boolean" />
    </attribute>
</define>

 <define name="AttributeGeneric">
     <zeroOrMore>
      <attribute>
       <anyName>
        <except>
         <name ns="">visibility</name>
         <name ns="">direction</name>
         <name ns="">q</name>
         <name ns="">media-type</name>
         <name ns="">label</name>
         <name ns="">enabled</name>
        </except>
       </anyName>
      </attribute>
     </zeroOrMore>
 </define>

 <define name="ElementAny">
   <element>
     <anyName>
       <except>
         <name>context</name>
         <name>streams</name>
         <name>max-bw</name>
```

```
                <name>max-session-bw</name>
                <name>max-stream-bw</name>
                <name>media-intermediaries</name>
                <name>qos-dscp</name>
                <name>local-ports</name>
                <name>media-types-allowed</name>
                <name>media-types-excluded</name>
                <name>media-type</name>
                <name>codecs-allowed</name>
                <name>codecs-excluded</name>
              </except>
            </anyName>
            <ref name="anyExtension"/>
          </element>
        </define>

        <define name="anyExtension">
          <zeroOrMore>
            <choice>
              <element>
                <anyName/>
                <ref name="anyExtension"/>
              </element>
              <attribute>
                <anyName/>
              </attribute>
              <text/>
            </choice>
          </zeroOrMore>
        </define>

      </grammar>
```

9. Security Considerations

Section 5 of [RFC6794] discusses security aspects related to the transfer of session policy information between user agents and policy servers, including their authentication and the use of TLS between them. In particular, a UA needs to check the server's certificate and only accept policies from severs from which the UA is configured to accept policies. Section 7 of RFC 3470 [RFC3470] provides general security considerations regarding the transport of XML documents in network protocols. Session info and session policy information can be sensitive information. The protocol used to distribute session info and session policy documents SHOULD ensure authentication, confidentiality, and message integrity. The use of [RFC6795] to distribute session info and session policy document meets these requirements.

An attacker could attempt to modify session policy documents that were sent to a client so that their processing by the client would be more costly (e.g., in terms of merging policies). The attacker could also attempt to create its own fake policy documents and send them to the client with the same purpose or in order to get the client to comply with those fake policies as part of a Denial-of-Service (DoS) attack. The protocol used to distribute session policy documents SHOULD ensure authentication, privacy, and message integrity. The use of [RFC6795] to distribute session policy document meets these requirements.

The <shared-secret> element can contain a shared secret needed to authenticate at a media intermediary. The privacy of documents containing this element MUST be preserved when they are sent between a server and a UA. When [RFC6795] is used to distribute these documents, encryption as defined in [RFC3261] (i.e., TLS or S/MIME) MUST be used.

10. IANA Considerations

This document registers a new media type (application/ media-policy-dataset+xml), a new RELAX NG schema, and a new XML namespace.

10.1. Media Type Registration

Media type name: application

Media subtype name: media-policy-dataset+xml

Mandatory parameters: none

Optional parameters: Same as charset parameter of application/xml as specified in RFC 3023 [RFC3023].

Encoding considerations: Same as encoding considerations of application/xml as specified in RFC 3023 [RFC3023].

Security considerations: See Section 10 of RFC 3023 [RFC3023] and Section 9 of this specification.

Interoperability considerations: none.

Published specification: This document.

Applications that use this media type: This document type is used to convey session description and media policy information between SIP user agents and a domain.

Additional Information:

Magic Number: None

File Extension: .mpf or .xml

Macintosh file type code: "TEXT"

Personal and email address for further information: Volker Hilt <volker.hilt@bell-labs.com>

Intended usage: COMMON

Author/Change controller: The IETF.

10.2. RELAX NG Schema Registration

This specification registers a schema. The schema can be found as the sole content of Section 8.

URI: urn:ietf:params:xml:schema:mediadataset

Registrant Contact: IETF RAI area <rai@ietf.org>, Volker Hilt <volker.hilt@bell-labs.com>

RELAX NG Schema: The RELAX NG schema to be registered is contained in Section 8.

10.3. URN Sub-Namespace Registration

This section registers a new XML namespace, as per the guidelines in [RFC3688].

URI: The URI for this namespace is urn:ietf:params:xml:ns:mediadataset.

Registrant Contact: IETF RAI area <rai@ietf.org>, Volker Hilt <volker.hilt@bell-labs.com>

XML:

```
BEGIN
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
          "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
        content="text/html;charset=iso-8859-1"/>
  <title>Media Policy Data Set Namespace</title>
</head>
<body>
  <h1>Namespace for Media Policy Data Sets</h1>
  <h2>urn:ietf:params:xml:ns:mediadataset</h2>
  <p>See <a href="http://www.rfc-editor.org/rfc/rfc6796.txt">
     RFC 6796</a>.</p>
</body>
</html>
END
```

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2141] Moats, R., "URN Syntax", RFC 2141, May 1997.

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC4574] Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute", RFC 4574, August 2006.

[RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, February 2007.

[RFC4975] Campbell, B., Mahy, R., and C. Jennings, "The Message Session Relay Protocol (MSRP)", RFC 4975, September 2007.

[RFC4976] Jennings, C., Mahy, R., and A. Roach, "Relay Extensions for the Message Sessions Relay Protocol (MSRP)", RFC 4976, September 2007.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.

[RFC6795] Hilt, V. and G. Camarillo, "A Session Initiation Protocol (SIP) Event Package for Session-Specific Policies", RFC 6795, December 2012.

[W3C.REC-xml-20081126] Sperberg-McQueen, C., Yergeau, F., Maler, E., Bray, T., and J. Paoli, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126>.

[W3C.REC-xml-names-19990114] Hollander, D., Bray, T., and A. Layman, "Namespaces in XML", World Wide Web Consortium First Edition REC-xml-names-19990114, January 1999, <http://www.w3.org/TR/1999/REC-xml-names-19990114>.

11.2. Informative References

[RFC2648] Moats, R., "A URN Namespace for IETF Documents", RFC 2648, August 1999.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3470] Hollenbeck, S., Rose, M., and L. Masinter, "Guidelines for the Use of Extensible Markup Language (XML) within IETF Protocols", BCP 70, RFC 3470, January 2003.

[RFC4583] Camarillo, G., "Session Description Protocol (SDP) Format for Binary Floor Control Protocol (BFCP) Streams", RFC 4583, November 2006.

[RFC4629] Ott, H., Bormann, C., Sullivan, G., Wenger, S., and R. Even, "RTP Payload Format for ITU-T Rec", RFC 4629, January 2007.

[RFC4856] Casner, S., "Media Type Registration of Payload Formats in the RTP Profile for Audio and Video Conferences", RFC 4856, February 2007.

[RFC6080] Petrie, D. and S. Channabasappa, "A Framework for Session Initiation Protocol User Agent Profile Delivery", RFC 6080, March 2011.

[RFC6794] Hilt, V., Camarillo, G., and J. Rosenberg, "A Framework for Session Initiation Protocol (SIP) Session Policies", RFC 6794, December 2012.

Appendix A. Acknowledgements

Many thanks to Allison Mankin, Dan Petrie, Martin Dolly, Adam Roach, and Ben Campbell for the discussions and suggestions. Many thanks to Roni Even, Mary Barnes, Yaron Sheffer, Pete McCann, and Henry S. Thompson for reviewing the document and to Jari Urpalainen for helping with the RELAX NG schema.

Authors' Addresses

Volker Hilt
Bell Labs/Alcatel-Lucent
Lorenzstrasse 10
70435 Stuttgart
Germany

EMail: volker.hilt@bell-labs.com

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com

Jonathan Rosenberg
jdrosen.net
Monmouth, NJ
USA

EMail: jdrosen@jdrosen.net

Dale R. Worley
Ariadne Internet Services, Inc.
738 Main St.
Waltham, MA 02451
US

EMail: worley@ariadne.com