SIPPING Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 6, 2010

V. Hilt
Bell Labs/Alcatel-Lucent
D. Worley
Nortel Networks Corp.
G. Camarillo
Ericsson
J. Rosenberg
jdrosen.net
March 5, 2010

A User Agent Profile Data Set for Media Policy
draft-ietf-sipping-media-policy-dataset-09

Abstract

This specification defines a document format for the media properties of Session Initiation Protocol (SIP) sessions. Examples for media properties are the codecs or media types used in a session. This document format is based on XML and can be used to describe the properties of a specific SIP session or to define policies that are then applied to SIP sessions.

Table of Contents

1. Introduction

The Framework for Session Initiation Protocol (SIP) [RFC3261] User Agent Profile Delivery [I-D.ietf-sipping-config-framework] and the Framework for SIP Session Policies [I-D.ietf-sip-session-policy-framework] 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 Dataset 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 the Framework for SIP Session Policies [I-D.ietf-sip-session-policy-framework], session info documents are used in conjunction with session-specific policies. A session info document is created by a 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 policy documents to the same session. This document specifies rules for merging XML elements from multiple sources and applying them to the same session. It should be noted

that these merging rules are part of the semantics of the 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 is not needed for 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. Design Considerations

This section discusses design considerations for the Media Policy Dataset Format (MPDF).

3.1. Namespace and MIME Type

The MPDF format is based on XML [W3C.REC-xml-20040204]. A 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 schemas 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 MIME type for the Media Policy Dataset 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. For each MPDF element it is defined, which of these attributes can be used. Attributes that are not defined for an element MUST be ignored.

3.3.1. The 'visibility' Attribute

The attribute "visibility" specifies whether or not the user agent is permitted 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 SHOULD NOT display the property value. Display of the property value may be allowed using special administrative interfaces, but is not appropriate to the ordinary user.

3.3.2. The 'policy' Attributes

The 'policy' attribute attribute is used to define the constraining properties of an element. It defines how the element value is used by an endpoint (e.g. whether it can or can not be used in a session). The following values are defined for the 'policy' attribute:

- allow: the value contained in the element is allowed and SHOULD be used in sessions. This is the default value that is used if the 'policy' attribute is omitted.
- disallow: the value contained in the element is forbidden and MUST NOT be used in sessions.

The policy attribute can be omitted if the default policy 'allow' applies.

3.3.3. The 'excluded-policy' Attributes

The "setting_container" element has an optional 'excluded-policy' attribute. This attribute specifies the default policy for all values that are not in the container. Elements that are present in the container have their own 'policy' attribute, which defines the policy for that element. The following values are defined for the 'excludedPolicy' attribute:

- o allow: values not listed in the container are allowed and MAY be used in sessions. This is the default value that is used if the 'excludedPolicy' attribute is omitted.
- o disallow: values not listed in the container are forbidden and MUST NOT be used in sessions.

The excludedPolicy attribute can be omitted if the default policy 'allow' applies.

3.3.4. 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:

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

3.3.5. The 'q' Attribute

It should be possible to express a preference for a certain value, if multiple values are allowed within a property. For example, it should be possible to express that the codecs G.711 and G.729 are allowed, but G.711 is preferred. Preferences can be expressed by adding a 'q' attribute to a property element. Elements derived from the "setting" element for which multiple occurrences and values are allowed SHOULD have a "q" attribute if the order is significant. Typically these elements are contained in an element derived from the "setting_container" element. The 'q' attribute is only meaningful if the 'policy' attribute set to 'allowed'. It must be ignored in all other cases.

An element with a higher 'q' value is preferred over one with a lower 'q' value. 'q' attribute values range from 0 to 1. The default value is 0.5.

3.4. Merging Property Sets

A UA may receive property sets from multiple sources, which need to be merged into a single combined document the UA can work with.

Properties that have a single value (e.g. the maximum bandwidth allowed) require that a common value is 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. Properties that allow multiple values (i.e. property containers) need to be merged by combining the values from the different data sets. The following sections describe common merging algorithms. The definition of an MPDF element can refer to these algorithms.

OPEN ISSUE: Can we define an order for 'device', 'user', and 'application' profiles to simplify merging?

3.4.1. Multiple Enumerated Value Merging Algorithm

Multiple values in property containers are merged by combining the values from each of the competing data sets. This is accomplished by copying the elements from each property container into the merged container. Elements with identical values are only copied once. The 'policy' attribute of two elements with the same value is adjusted during the merging process according to Table 1. If an element exists only in one property container, then the default policy of the other container (i.e. the excludedPolicy) is used when accessing Table 1. For example, if an element is disallowed in one data set and the element is not contained in the other data set but the default policy is allowed for that data set, then the values disallowed and allowed are used to access Table 1. Consequently, the element will be disallowed in the merged data set. Finally, the excludedPolicy attributes of the containers are also merged using Table 1. In addition to these merging rules, each schema may define specific merging rules for each property container.

set 1 \ set 2	mandatory	allow	disallow
mandatory	mandatory	mandatory	conflict!
allow	mandatory	allow	disallow
disallow	conflict!	disallow	disallow

Table 1: merging policies.

The following example illustrates the merging process for two data sets. All elements are merged into one container and the policy attributes are adjusted according to Table 1. The merged container has the default policy disallow, which is determined using Table 1. The entry for PCMA in the merged data set is redundant since it has the default policy.

```
Data set 1:
<codecs excluded-policy="allow">
  <codec policy="disallow">
    <mime-type>audio/PCMA</mime-type>
  </codec>
</codecs>

Data set 2:
<codecs excluded-policy="disallow">
  <codec policy="allow">
    <mime-type>audio/PCMA</mime-type>
  </codec>
  <codec policy="allow">
    <mime-type>audio/G729</mime-type>
  </codec>
</codecs>

Merged data set:
<codecs excluded-policy="disallow">
  <codec policy="disallow">
    <mime-type>audio/PCMA</mime-type>
  </codec>
  <codec policy="allow">
    <mime-type>audio/G729</mime-type>
  </codec>
</codecs>
```

Some constellations of policy attributes result in an illegal merged data set. They constitute a conflict that can not be resolved automatically. For example, two data sets may define two non-overlapping sets of allowed audio codecs and both disallow all other codes. The resulting merged set of codecs would be empty, which is illegal according to the schema definition of the codecs element. If the use of these properties is enforced by both networks, the UA may experience difficulties or may not be able to set up a session at all.

The combined property set 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.

3.4.2. Closest Value First Merging Algorithm

Some properties require that the values from different data sets are ordered based on the origin of the data set during the merging process. Property values that come from a domain close to the user agent take precedence over values that were in a data set delivered

by a remote domain. This order can be used, for example, to select the property value from the closest domain. In many cases, this is the local domain of the user agent. For example, the URI of an outbound proxy could be merged this way. This order can also be used to generate an ordered list of property values during the merging process. For example, multiple values for media intermediaries can be ordered so that the closest media intermediary is traversed before the second closest intermediary and so on.

This merging algorithm requires that the source of a data set is considered.

If property sets are delivered through the configuration framework [I-D.ietf-sipping-config-framework], the value received through a subscription using the "local-network" profile-type takes precedence over values received through other profile-type subscriptions.

The session-specific policy mechanism [I-D.ietf-sip-session-policy-framework] provides an order among policy servers. This order is based on the order, in which a SIP message traverses the network, starting with the closest domain. This order can directly be used to order property values as described above.

3.5. The <property-set> Element

The root element of a property set is <property-set> it is the container that is provided to the user agent. The elements contained within a <property-set> contain the specific properties which are to be applied to the user agent.

The <property-set> element is the root element for Session Info and Session Policy documents.

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 an SDP [RFC4566] session description or an SDP offer/answer pair [RFC3264].

Session info documents can be used for session-specific policies [I-D.ietf-sip-session-policy-framework]. In this usage, a UA creates a session info document based on its SDP 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 all 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 remove the video stream from the XML markup and return the modified session info document to the UA.

Session info documents use the <session-info> element.

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

```
<session-info><\session-info>
```

4.1. The <session-info> Element

The <session-info> element describes the properties of a specific SIP session. The <session-info> element MAY occur multiple times inside a <property_set> element.

The <session-info> element MAY contain one optional <streams>, <context> 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. The MPDF elements are defined in Section 6.

4.2. Mapping SDP to Session Info Documents

If a UA has an SDP offer as well as an answer [RFC3264] and wants to create a session info document, the UA MUST use the answer to fill in the elements of the session info document except for the remote-host-port and local-host-port elements, which are taken from the remote and local session description respectively. The local session description is the one created locally by the UA (i.e., the offer if the UA has initiated the offer/answer exchange). The remote session description is the one received from the remote UA.

The following rules describe the creation of session info documents based on SDP description(s) for a few exemplary elements. Other elements are created following the same principles.

A UA MUST create a separate <stream> element for each m= line in an SDP 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.

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 MUST

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

5. Session Policy Documents

Session policy documents describe a policy for SIP sessions. Session policy documents are independent of a 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 to modify this session, if needed, according to the described policies.

Session policy documents can be used to encode session-independent policies [I-D.ietf-sip-session-policy-framework]. 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 media type video in its SDP offer or answer.

Session policy documents use the <session-policy> element.

5.1. The <session-policy> Element

The <session-policy> element describes a policy that applies to SIP sessions. The <session-policy> element MAY occur multiple times inside a <property_set> element.

The <session-policy> element MAY contain one optional <context> and <local-ports> element and multiple (including zero) <media-types>, <codecs>, <max-bw>, <max-session-bw>, <max-stream-bw> and <qos-dscp> elements as well as elements from other namespaces. The MPDF elements are defined in Section 6.

6. Media Property Elements

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

6.1. The <media-types> Element

The <media-types> element is a container that is used to define the set of media types (e.g., audio, video) that can or cannot be used in a session. A specific media type is included in the set by adding the corresponding <media-type> element to this container.

The <media-types> element can only be used in session policy document (i.e., inside the <session-policy> container).

This element MAY have the following attributes (see Section 3.3): direction, visibility, excluded-policy.

Multiple <media-types> elements MAY only be present in a container element if each applies to a different set of streams (e.g., one <media-types> element for incoming and one for outgoing streams). The <media-types> element MUST contain one or more <media-type> elements.

> Merging of session-policy documents: <media-types> containers are merged using the "Multiple Enumerated Value Merging Algorithm" Section 3.4.

6.1.1. The <media-type> Element

The <media-type> element identifies a specific media type. The value of this element MUST be the name of a IANA registered media type (see RFC4566 [RFC4566]), such as 'audio', 'video', 'text', or 'application'.

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

If used inside a <session-policy> element, this element MAY have the following additional attribute (see Section 3.3): policy. Media types that have the policy 'allowed' MAY be used and media types with the policy 'disallowed' MUST NOT be used.

6.2. The <codecs> Element

The <codecs> element is a container that is used to define the set of codecs that may or may not be used in a session. A policy MUST allow the use of at least one codec per media type. A specific codec is included in the set by adding the corresponding <codec> element to this container.

The <codecs> element can only be used in a session policy document (i.e., inside the <session-policy> container).

The <codecs> element MAY have the following attributes (see Section 3.3): direction, visibility, excluded-policy.

Multiple <codecs> elements MAY only be present in a container element if each applies to a different set of streams (e.g., one <codecs> element for incoming and one for outgoing streams). The <codecs> element MUST contain one or more <codec> elements.

Merging of session-policy documents: <codecs> containers are merged using the "Multiple Enumerated Value Merging Algorithm" Section 3.4.

6.2.1. The <codec> Element

The <codec> element identifies a specific codec. The content of this element MUST be a registered MIME type [RFC4855] using media type and subtype (e.g., audio/PCMA [RFC4856] or video/H263 [RFC4629]) and possibly additional registered MIME type parameters.

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

If used inside a <session-policy> element, the <codec> element MAY have the following additional attribute (see Section 3.3): policy. Codecs that have the policy 'allowed' MAY be used and codecs with the policy 'disallowed' MUST NOT be used.

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

6.2.1.1. The <mime-type> Element

The <mime-type> element contains a MIME type that identifies a codec. The value of this element MUST be a combination of a registered MIME media type and subtype [RFC4855] separated by a "/" (e.g., audio/PCMA, audio/G726-16 [RFC4856] or video/H263 [RFC4629]).

6.2.1.2. 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 registered MIME type parameter for the codec [RFC4855]. The name and value are separated by a "=". For example, the parameter "profile=0" can be used to specify a specific profile for the codec "video/H263-2000" [RFC4629].

6.3. The <streams> Element

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

The <streams> element is only defined for session information documents (i.e., in a <session-info> container).

The <streams> element MUST contain one or more <stream> elements.

6.3.1. 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 <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 MUST 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 following attributes (see Section 3.3): direction, label.

The label attribute is used to identify a specific media stream in a session description. The value of the label attribute is a token. The token can be chosen freely, however, it MUST be unique among all <stream> element in a session-info document. If a label attribute [RFC4574] is present in the SDP description, its value MUST be carried over to the label attribute of the corresponding <stream> element.

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

6.3.1.1. The <local-host-port> Element

The <local-host-port> element contains the hostname or IP address and the 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 SDP description. The port number is found in the m= element.

6.3.1.2. 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 port number of the media stream in the remote session description.

6.4. The <max-bw> Element

The <max-bw> element defines the overall maximum bandwidth in kilobits 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 following attribute (see Section 3.3): direction.

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

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 is used.

6.5. The <max-session-bw> Element

The <max-session-bw> element defines the maximum bandwidth in kilobits 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 value of the <max-session-bw> element is equivalent to the CT bandwidth in the b= line of an SDP [RFC4566] announcement.

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

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

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 is used.

6.6. The <max-stream-bw> Element

The <max-stream-bw> element defines the maximum bandwidth in kilobits 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 for lower-layer transport and network protocols (e.g., UDP and IP).

The value of the <max-stream-bw> element is equivalent to the AS bandwidth in the b= line of an SDP [RFC4566] announcement.

The <max-stream-bw> element MAY have the following attribute (see Section 3.3): direction, media-type.

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

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

The media-type attribute is used to define that the <max-stream-bw> element only applies to streams of a certain media type. For example, it may only apply to audio streams. The value of the

'media-type' attribute MUST be the name of a IANA registered media type (see RFC4566 [RFC4566]), such as 'audio', 'video', 'text', or 'application'.

The label attribute 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 is used.

6.7. The <media-intermediaries> Element

The <media-intermediaries> element expresses a policy for routing a media stream through a media intermediary. The purpose of the <media-intermediaries> element is to tell the UA to send a media stream through one (or a chain of) media intermediaries. Instead of sending the media directly to its final destination, the UA specifies a source route, which touches each intermediary and then reaches the final recipient. If there are N hops, including the final recipient, there needs to be a way for the media stream to specify N destinations.

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 following attributes (see Section 3.3): direction.

Multiple <media-intermediaries> elements MAY only be present in a

container if 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>.

Merging of session-policy documents: the intermediaries defined in all policies are traversed. In general, local intermediaries should be traversed before remote intermediaries. During the merging process, <media-intermediaries> element values from different servers are ordered using the "Closest Value First Merging Algorithm" Section 3.4. The intermediaries should be traversed in this order.

Note: it is not intended that the <media-intermediaries> element replaces connectivity discovery mechanisms such as 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.

6.7.1. 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 to. 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. I.e., 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 [I-D.ietf-sip-session-policy-framework] in section Contacting the Policy Server.

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

6.7.1.1. 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.

6.7.1.2. 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.

6.7.2. The <turn-intermediary> Element

The TURN [I-D.ietf-behave-turn] 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.

6.7.2.1. The <shared-secret> Element

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

6.7.2.2. The <user> element

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

6.7.2.3. 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.

6.7.3. The <msrp-intermediary> Element

The 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 each of the <shared-secret> and <user> elements.

6.7.3.1. 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. Only URIs with a scheme of "msrps:" are valid in the <msrp-uri> element.

6.8. The <qos-dscp> Element

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

This element MAY have the following attributes (see Section 3.3): direction, media-type.

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

The media-type attribute is used to define that <qos-dscp> element only applies to streams of a certain media type. For example, it may only apply to audio streams. The value of the 'media-type' attribute MUST be the name of a IANA registered media type (see RFC4566 [RFC4566]), such as 'audio', 'video', 'text', or 'application'.

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 domain that is first traversed by the media stream has precedence and its DSCP value is used. During the merging process, <qos-dscp> element values from different servers are ordered using the "Closest Value First Merging Algorithm" Section 3.4. The DSCP value from the closest server is used.

6.9. 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 a start port and an end port separated by a "-". The start/end port is the first/last port that can be used.

This element MAY have the following attributes (see Section 3.3): visibility.

The <local-ports> element is only defined for session policy documents (i.e., in a <session-policy> container).

Merging of session-policy documents: the domain that is first traversed by the media stream has precedence and its local ports value is used. During the merging process, <local-ports> element values from different servers are ordered using the "Closest Value First Merging Algorithm" Section 3.4. The value from the closest server is used.

6.10. The <context> Element

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

The <context> element MAY contain multiple <contact> and one <info> element.

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

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

Merging of session-policy documents: the <context> element is not subject to merging.

6.10.1. The <policy-server-URI> Element

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

The <policy-server-URI> element is only defined inside a <session-policy> element.

6.10.2. The <contact> Element

The <contact> element contains a contact address (e.g., a SIP URI or email address) under which the issuer of this document can be reached.

6.10.3. The <info> Element

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

6.10.4. The <request-URI> Element

The <request-URI> element identifies the request-URI the dialog initiating request of a session is sent to.

The <request-URI> element is only defined inside a <session-info> element.

6.10.5. The <token> Element

The <token> element provides a mechanism for a policy server to return an opaque token to a UA. This is sometimes needed to ensure that all requests for a session are routed to the same policy server. The use of this token is described in the Framework for SIP Session Policies [I-D.ietf-sip-session-policy-framework].

The <token> element is only defined inside a <session-info> element.

6.11. 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 describes 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.

```
<property-set>
  <session-policy>
    <context>
      <policy-server-URI>policy@biloxi.example.com</policy-server-URI>
      <contact>sip:policy_manager@example.com</contact>
      <info>Access network policies</info>
    </context>
    <media-types excluded-policy="disallow">
      <media-type policy="allow">audio</media-type>
      <media-type policy="allow">video</media-type>
    </media-types>
    <codecs excluded-policy="allow">
      <codec policy="disallow">
        <mime-type>audio/G729</mime-type>
      </codec>
      <codec policy="disallow">
        <mime-type>audio/G723</mime-type>
      </codec>
    </codecs>
  </session-policy>
</property-set>
```

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 SDP 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:

```
 <property-set>
   <session-info>
     <context>
       <contact>sip:alice@somewhere.example</contact>
       <info>session information</info>
     </context>
     <streams>
       <stream>
         <media-type>audio</media-type>
         <codec><mime-type>audio/PCMU</mime-type></codec>
         <codec><mime-type>audio/1016</mime-type></codec>
         <codec><mime-type>audio/GSM</mime-type></codec>
         <local-host-port>host.somewhere.example:49562</local-host-port>
       </stream>
       <stream>
         <media-type>video</media-type>
         <codec><mime-type>video/H261</mime-type></codec>
         <codec><mime-type>video/H263</mime-type></codec>
         <local-host-port>host.somewhere.example:51234</local-host-port>
       </stream>
     </streams>
   </session-info>
 </property-set>
```

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 SDP 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 SDP 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:

```
<property-set>
  <session-info>
    <context>
      <contact>sip:alice@somewhere.example</contact>
      <info>session information</info>
    </context>
    <streams>
      <stream>
        <media-type>audio</media-type>
        <codec><mime-type>audio/PCMU</mime-type></codec>
        <codec><mime-type>audio/GSM</mime-type></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><mime-type>video/H261</mime-type></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>
</property-set>
```

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.

```
<property-set>
  <session-info>
    <context>
      <contact>sip:alice@somewhere.example</contact>
      <info>modified session information</info>
    </context>
    <streams>
      <stream label='1'>
        <media-type>audio</media-type>
        <codec><mime-type>audio/PCMU</mime-type></codec>
        <codec><mime-type>audio/GSM</mime-type></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><mime-type>video/H261</mime-type></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>
</property-set>
```

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">

     <include href="uaprofile.rng"/>

     <define name="PropertySetExtension" combine="interleave">
        <choice>
              <element name="session-info">
                  <ref name="SettingContainerAttributes"/>
                  <optional>
                      <ref name="ElementContext"/>
                  </optional>
                  <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>
            </element>

            <element name="session-policy">
                <ref name="SettingContainerAttributes"/>
                <optional>
                    <ref name="ElementContext"/>
                </optional>
                <optional>
                    <ref name="ElementLocalPorts"/>
                </optional>
                <zeroOrMore>
                    <ref name="ElementMediaTypes"/>
                </zeroOrMore>
                <zeroOrMore>
                    <ref name="ElementCodecs"/>
                </zeroOrMore>
                <zeroOrMore>
                    <ref name="ElementMaxBandwidth"/>
                </zeroOrMore>
                <zeroOrMore>
                    <ref name="ElementMaxSessionBandwidth"/>
                </zeroOrMore>
                <zeroOrMore>
                    <ref name="ElementMaxStreamBandwidth"/>
                </zeroOrMore>
                <zeroOrMore>
                    <ref name="ElementQoSDSCP"/>
                </zeroOrMore>
            </element>
        </choice>
    </define>

    <define name="ElementMediaTypes">
        <element name="media-types">
```

```
            <ref name="PolicyGeneralAttributes"/>
            <optional>
               <ref name="SettingContainerAttributes"/>
            </optional>
            <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="AttributePolicy"/>
            </optional>
            <optional>
              <ref name="AttributeGeneric"/>
            </optional>
        </element>
    </define>

    <define name="ElementCodecs">
        <element name="codecs">
          <ref name="PolicyGeneralAttributes"/>
            <optional>
               <ref name="SettingContainerAttributes"/>
            </optional>
            <zeroOrMore>
               <ref name="ElementCodec"/>
            </zeroOrMore>
        </element>
    </define>

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

```
            <element name="mime-type">
              <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>
            <oneOrMore>
              <ref name="ElementStream"/>
            </oneOrMore>
        </element>
    </define>

    <define name="ElementStream">
        <element name="stream">
            <optional>
              <ref name="AttributeDirection"/>
            </optional>
            <optional>
              <ref name="AttributeLabel"/>
            </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="domain">
                  <data type="string" />
                </element>
              </optional>
              <optional>
               <element name="request-URI">
                  <data type="string" />
                </element>
              </optional>
              <optional>
               <element name="token">
                  <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="AttributeMediaType">
          <attribute name="media-type">
            <data type="string" />
          </attribute>
      </define>

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

   </grammar>
```

9. Security Considerations

Session policy information can be sensitive information. The protocol used to distribute session policy information SHOULD ensure privacy, message integrity and authentication. Furthermore, the protocol SHOULD provide access controls which restrict who can see who else's session policy information.

10. IANA Considerations

This document registers a new MIME type, application/ media-policy-dataset+xml, and a new XML namespace.

10.1. MIME Registration

MIME media type name: application

MIME subtype name: media-policy-dataset+xml

Mandatory parameters: none

Optional parameters: Same as charset parameter 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 which use this media type: This document type has been used to convey 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,

<volkerh@bell-labs.com>

Intended usage: COMMON

Author/Change controller: The IETF.

10.2. 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, SIPPING working group, <sipping@ietf.org>, Volker Hilt, <volkerh@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 Dataset Namespace</title>
</head>
<body>
  <h1>Namespace for Media Policy Datasets</h1>
  <h2>urn:ietf:params:xml:ns:mediadataset</h2>
  <p>See <a href="[[[URL of published RFC]]]">RFCXXXX</a>.</p>
</body>
</html>
END
```

11. References

11.1. Normative References

[I-D.ietf-behave-turn]
Rosenberg, J., Mahy, R., and P. Matthews, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", draft-ietf-behave-turn-16 (work in progress), July 2009.

[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.

[W3C.REC-xml-20040204]
Maler, E., Sperberg-McQueen, C., Paoli, J., Yergeau, F., and T. Bray, "Extensible Markup Language (XML) 1.0 (Third Edition)", World Wide Web Consortium FirstEdition REC-xml-20040204, February 2004, <http://www.w3.org/TR/2004/REC-xml-20040204>.

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

11.2. Informative References

[I-D.ietf-sip-session-policy-framework]
Hilt, V., Camarillo, G., and J. Rosenberg, "A Framework for Session Initiation Protocol (SIP) Session Policies", draft-ietf-sip-session-policy-framework-07 (work in progress), February 2010.

[I-D.ietf-sipping-config-framework]
Channabasappa, S., "A Framework for Session Initiation Protocol User Agent Profile Delivery", draft-ietf-sipping-config-framework-17 (work in progress), February 2010.

[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.

[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.

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 and Mary Barnes for reviewing the draft and to Jari Urpalainen for helping with the Relax NG schema.

Authors' Addresses

Volker Hilt
Bell Labs/Alcatel-Lucent
791 Holmdel-Keyport Rd
Holmdel, NJ 07733
USA

Email: volkerh@bell-labs.com

Dale R. Worley
Nortel Networks Corp.
600 Technology Park Dr.
Billerica, MA 01821
US

Email: dworley@nortel.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