EBU

OPERATING EUROVISION AND EURORADIO

TECH 3368

AUDIO CONTRIBUTION OVER IP

PROFILES

SOURCE: FNS-ACIP

STATUS: VERSION 1.0

Geneva
November 2014

Scope

This document defines a set of requirements for the use of profiles between interoperable contribution-quality audio over IP equipment.

In this document the following bold, uppercase words have special meanings.

MUST and **SHALL** identify mandatory elements that need to be implemented or followed in order to achieve interoperability.

SHOULD and **RECOMMENDED** identify elements that are not mandatory, but whose implementation is advisable.

MAY and **OPTIONAL** identify facultative elements which, if present, SHOULD be implemented as specified for better interoperability with other equipment implementing the same elements.

Feedback on this document is invited; it should be sent to Mathias Coinchon (coinchon@ebu.ch), project manager of the EBU FNS-ACIP group.

Terminology

User agent	An Internet endpoint (as defined in RFC3261), which may be any hardware or software device capable of transmitting or receiving audio streams using SIP
IP audio codec	Any hardware or software device capable of transmitting or receiving audio streams using SIP
Caller	The user agent initiating the call
Callee	The user agent receiving the call

Contents

Audio Contribution over IP

Profiles

EBU Committee	*First Issued*	*Revised*	*Re-issued*
TC	2014		

Keywords: Audio Contribution, Internet protocol, User Requirements, Interoperability, Profile.

1. Introduction

Audio contribution over IP will inevitably occur over several different types of networks using several different types of IP audio codecs. Different networks have different characteristics concerning stability, latency and maximum throughput. When establishing a connection, codecs must negotiate a common set of parameters and audio engineers or reporters should not be required to specify these parameters in detail; instead, it would be more useful to be able to select a profile to use for this particular connection. In an ideal environment, the IP audio codec could automatically select the profile depending on network characteristics.

Currently, EBU Tech 3326 and SIP (RFC3261), specify mechanisms for negotiating parameters for setting up an audio connection. The most important parameters are already included in the relevant RFCs for VoIP communication. However, for broadcasters, some vital parameters are not. This document describes these additional parameters, why they are vital and proposes a method of including them in the SIP negotiation process.

2. Profiles

A profile is a set of parameters describing how to transmit and receive audio streams and for the decoder to successfully decode the audio, based on the parameters sent. The set includes only parameters required for a specific connection and is subject to negotiation when the connection is established. The profile does not contain global configuration of the end devices. The ability to store and recall global configurations is a feature determined by the manufacturer of the end device.

Profiles may be sent from any user agent. Any user agent may be a caller or a callee. It must be possible to store a number of pre-configured profiles in all user agents.

2.1 Asymmetric profiles

An asymmetric profile is a profile that specifies different parameter values for the sent and received streams. Asymmetric profiles also include one-way transmissions. The default behaviour of a SIP connection is to use the same settings for sending and receiving streams, but in some cases this may not be suitable. For example, if the network is asymmetric in itself, i.e. a DSL line with

low upload capacity and high download capacity, the bit-rate of the sent and the received streams will have to be different to maximise the usage of the connection without overflowing it.

A connection established with an asymmetric profile must make use of the a=sendonly and/or a=recvonly attributes inside the session description of the SIP INVITE message. Unidirectional connections simply use one media description with one of these attributes while bi-directional connections, requiring different formats in each direction, include two media descriptions: One with a=sendonly and the other with a=recvonly. Symmetric, bi-directional session descriptions contain a single media description and the bi-directional mode is signalled either implicitly by omitting a=sendonly and a=recvonly or explicitly with the SDP attribute a=sendrecv.

Profile management in user agents must support asymmetric profiles since this mode of operation will often be required.

Note that RFC4566 explicitly states that using the same port number in two media descriptions is "undefined", however, no provisions are made for this particular case. The mandated use of a=sendonly and a=recvonly removes the ambiguity and this document allows use of the same port, if and only if, the directions of the streams are different in the concerned media descriptions.

2.1.1 Asymmetric call example: Send standard aptX, receive AAC or g.722

```
v=0
o=mhandley 4856533 0 IN IP4 203.0.113.85
s=Asymmetric call
c=IN IP4 203.0.113.85
t=0 0
m=audio 5004 RTP/AVP 98
a=rtpmap:98 aptx/44100/2
a=fmtp:98 variant=standard; bitresolution=16;
a=ptime:4
a=sendonly
m=audio 5004 RTP/AVP 96 9
a=rtpmap:96 mpeg4-generic/48000/1
a=rtpmap:9 G722/8000
a=fmtp:96 streamtype=5; profile-level-id=16; config=1188;mode=AAC-hbr;
sizeLength=13; indexLength=3; indexDeltaLength=3; constantDuration=1024;
bitrate=48000
a=recvonly
```

2.2. Parameters of a profile

This table shows the parameters that are considered to be included in the profile, whether and where they are signalled and whether they are handled by existing RFC's.

Table 1: Profile Parameters

Parameter	Signalled in SDP	Handled by existing RFCs	Related to
Profile number	No	No	Profile
Profile name	No	No	Profile
Coding/decoding Algorithm	Yes	Yes	Media
Bitrate	Yes	Yes	Media
Sampling Frequency	Yes	Yes	Media
Channel Mode	Yes	Yes	Media

Profile version	Yes	No	Session
RX jitter buffer	Yes	No	Media & Session
Packet length	Yes	Yes*	Media
QoS-Recommendation	Yes+	No	Media & Session
Protection	Yes	Yes	Media

* Packet time, or ptime is specified in RFC4566, but this document discusses a required extension

+ For clarification, the QoS-Recommendation parameter is NOT setting the DS field of the IP packet carrying the SIP INVITE packet. See description of QoS-Recommendation below.

2.3 Profiles Specification

As profiles use SDP for their transportation, the same rules and conditions as specified in clause 5 of RFC4566 apply.

2.3.1 Signalling parameters

Many of the parameters listed as part of a profile are already signalled in a satisfactory way, using existing SDP attributes. However, some are not and this specification defines a principle for signalling these with a single new attribute. A profile description consists of a number of lines of text of the form:

```
a=ebuacip:<parameter> <options>
```

and must comply with the sub-rules as defined in RFC4566 SDP Session Description Protocol:

```
; sub-rules of 'a='
   attribute =          (att-field ":" att-value) / att-field
   att-field =          token
   att-value =          byte-string
```

This attribute will cater for signalling both format-specific parameters and general parameters.

Whitespace MUST NOT be used on either side of the "=" sign.

The profile parameters are all used with the a= type attribute line(s) within a SDP description. No other type is used.

The use of any profile parameters is OPTIONAL, however, if any are used, then the ebuacip protocol version (`version`) MUST also be included. Additionally, those profile parameters used MUST appear in exactly the order given here; the fixed order greatly enhancing error detection and allowing for a simple parser.

`version` (ebuacip protocol version)

`jb` (jitter buffer)

`jbdef` (jitter buffer definition)

`plength` (packet length)

`qosrec` (QoS-recommendation)

`protp` (protection)

An SDP parser MUST ignore any attribute it doesn't understand.

2.4 Offer/Answer consideration

For all parameters signalled, in accordance with the offer/answer model, RFC3264, the answer message sent back to the caller must contain the selected option and its accompanying definition.

2.5 Rejected calls

If a call is rejected for any reason when sending any of these parameters, any new call attempted should be made with a new session and not a RE-INVITE, using new parameters and/or a new (different) profile to retry.

3. The Parameters

3.1 Profile number

This is a number assigned to the profile for identification purposes, along with the Profile Name below. More than one profile will be stored within a user agent and the profile number assists with allowing users to select the appropriate profile from a list. This parameter is never signalled during SIP negotiation.

3.2 Profile name

This is an alphanumeric string assigned to a profile for identification purposes. More than one profile will be stored within a user agent and the profile name assists with allowing users to select the appropriate profile from a list. This parameter is never signalled during SIP negotiation.

3.3 EBU ACIP protocol version number

This attribute gives the version of the EBU ACIP protocol being used.

This document defines version 0. There is no minor version number.

```
a=ebuacip:version 0
```

3.3.1 Grammar

```
version = "a=ebuacip:" "version" 1*(DIGIT)
```

3.4 RX jitter buffer

The jitter buffer serves to ensure a steady stream of media packets can be played out in a regular manner, at the same rate as they were sent. If the jitter buffer is too small, packets can fill up the buffer quicker than they can be played out, causing packets to be lost and hence adversely affecting the media quality. If the jitter buffer is too large, all packets can expect to be safe, but then the total delay in the system may be unacceptably high, particularly for two way conversations.

It is likely that the caller user agent, or the person or system using the caller user agent, may well be aware of the network properties of the network that will be used for the call. This makes it reasonable for the caller to suggest settings for the receiving buffer (at the callee end).

There are devices on the market today that automatically measure the network performance and adjust settings accordingly. Even if the devices do not do this themselves, the SIP registrar/location service does know the IP address of both user agents. This is true, at least in the most common

case where all user agents are registered in the same domain. These network locations can provide a reasonable basis for making assumptions about the types of networks the stream will have to be transported through. This means that a user agent that also communicates with a presence server, which in turn communicates with the SIP register/location service, can get recommendations about what profile to use for this particular call.

3.4.1 RX Adaptive jitter buffer

When an adaptive jitter buffer is recommended, two parameters are required. Recommended minimum and maximum size of the receive jitter buffer of the callee in milliseconds. The reasoning behind these two parameters is that the fluctuations in audio delay must be controlled as much as possible to minimise conversation difficulties. The upper limit will reduce the amount of cross-talk (of speakers) and the lower limit will allow speakers to adjust to the connection without being forced to constantly re-adjust.

The actual jitter buffer size at any time during the session will vary within the range of the minimum and maximum parameter values specified according to the network conditions measured by the callee user agent.

3.4.2 RX Static jitter buffer

It is not always ideal to use an adaptive jitter buffer. Sometimes it is more important to have an acceptable fixed delay suitable for the intended scenario, rather than a short delay. When a static jitter buffer is recommended, this parameter specifies a recommended fixed size of the receiving jitter buffer of the callee in milliseconds.

Alternatively a range can be specified indicating that any fixed jitter buffer size within the range will be acceptable. This version of the static jitter buffer differs subtly from the adaptive jitter buffer in that the actual jitter buffer size at any time during the session will be fixed, once set for the session, within the range of the minimum and maximum parameter values specified.

3.4.3 Use-case

Consider two user agents, A and B. User agent A is connected to a corporate Wide Area Network (WAN) while user agent B is connected to a public wifi network in a hotel room. User agent A is currently set to use a fixed jitter buffer size of six milliseconds as this is what is most commonly used for calls between user agents within the corporate WAN. User agent B now invites user agent A for a bi-directional call. Without changing the jitter buffer size in user agent A, this call will fail or at the very best be full of audio glitches, since the stream is transported over the public Internet. On the Internet the jitter is likely to exceed six milliseconds. If user agent B could inform user agent A of a jitter buffer size that is appropriate for the network that will be used for this particular call, the call is more likely to have glitch free audio.

3.4.4 Signalling

Jitter buffer signalling is sent in the SDP part of a SIP INVITE message and consists of two parts. The first part is a list of available options and the order of preference for these. The second part specifies each option in detail.

```
a=ebuacip:jb <option list>
a=ebuacip:jbdef <option> <jb-option>
```

A <jb-option> can specify either a fixed or automatic jitter buffer. The fixed jitter buffer option specifies a time in milliseconds indicating requested buffer size. Additionally, a minimum-maximum range in milliseconds may be given, indicating that any buffer size in that interval is supported within the context of the offer. The automatic jitter buffer option specifies a minimum and

maximum time, in milliseconds, indicating the range the jitter buffer size is allowed to fluctuate in real-time operation.

3.4.4.1 Examples

Three options for jitter buffer size, including a range

```
a=ebuacip:jb 0 1 2
a=ebuacip:jbdef 0 fixed 20
a=ebuacip:jbdef 1 auto 20-50
a=ebuacip:jbdef 2 fixed 20-100
```

This example shows part of an INVITE message allowing three different settings for the jitter buffer. The first choice is a fixed jitter buffer of 20 ms. If the callee does not accept this, the second option is to use an automatic jitter buffer with a minimum size of 20 ms and a maximum size of 50 ms. If the callee also refuses this setting, the third option is to use any fixed jitter buffer size in the interval between 20 and 100 ms. If none of these options can be accommodated, the invitation will be rejected with a "`488 Not Acceptable Here`" response, as defined in RFC3261.

Note: If the invitation is rejected, as described in the above example, the caller may always attempt to call again with new parameters and/or a new (different) profile to retry.

3.4.4.2 Grammar

```
jb        = "a=ebuacip:" "jb" 1*(SP jb-index)
jbdef     = "a=ebuacip:" "jbdef" SP (jb-index) SP (jb-option)
jb-option = "fixed" SP 1*(DIGIT) ["-" 1*(DIGIT)] /
            "auto" SP 1*(DIGIT) "-" 1*(DIGIT)
jb-index  = 1*(DIGIT)
```

3.5 Packet length

Packet length, or the amount of audio contained in a single packet, may be signalled using `a=ptime` (packet time) as defined in RFC4566 This is a "media-level" attribute that adds information about the media stream, but not the media format, in the session description protocol (SDP) section of the SIP INVITE. The media format is conveyed via the media descriptions, `m=` and the subsequent `a=rtpmap` lines of the SDP as defined in RFC4566.

Different media formats impose different requirements and ranges for packet time values and because of this, a parameter that can be different for each media format included in the INVITE message is needed.

Also, different networks, in particular last mile accesses, are optimised for different packet lengths. In order to maximise efficiency we need to be able to specify packet length as close to the optimum as possible.

3.5.1 Use-case

Consider two user agents, one located on a corporate high performance network and the other on a cheap, low performance network, for example an ADSL line. In the corporate network, small packets are typically used to keep the delay low, but if the unit on the other end tries to send a similar number of small packets, the cheap ADSL modem may not be able to cope with the traffic. Since the caller has a chance of knowing the network conditions, via a presence service or measurement, it is logical that it suggests a packet length for the callee to use on its return

stream.

3.5.2 Signalling

The packet length is signalled in the SDP section of a SIP INVITE.

```
a=ebuacip:plength <format> <milliseconds>
```

The format is the format number defined in the corresponding m= line and milliseconds is an integral number of milliseconds of audio contained in each packet.

If a plength attribute is not specified for a certain format, user agents must use a packet length that is allowed by the format and is closest to the specified ptime. If neither plength nor ptime are specified, the default packet length of the format must be used.

If an incoming SDP contains both ptime and plength attributes, the plength attribute must be used in preference to ptime. If the value specified in plength cannot be accommodated, the call must be rejected.

When sending an SDP, the plength attribute should be signalled for each format, but a ptime attribute should also be included. The value to select for the ptime attribute shall be any value between and including, the minimum or maximum values of the plength attributes:

```
minimum plength value <= ptime <= maximum plength value
```

A smaller ptime prioritises low latency while larger values reduce the network load and overhead. The choice is entirely up to the manufacturer, unless it is exposed for configuration in the user interface.

The maxptime attribute may still be used with or without plength attributes. When signalled, the largest plength must be smaller than or equal to the maxptime value.

3.5.2.1 Example for packet length

```
a=ebuacip:plength 98 4
```

This example shows part of an INVITE message specifying a four millisecond packet length for format 98.

3.5.2.2 Grammar

```
plength    = "a=ebuacip:" "plength" SP (format-index) SP 1*(DIGIT)
format-index = 1*(DIGIT)
```

3.6 QoS-Recommendation

A recommended quality of service in the network may be requested by using the DiffServ (DS) field of the IP headers of the RTP packets. The DiffServ method as described in RFC2474 shall be implemented.

The DS field is the 6 most-significant bits of the deprecated TOS field in the IP header. The six bits of the DS field hold traffic class information in the form of a Differentiated Services Code Point (DSCP). It uses this field in each IP packet header to mark packets according to their traffic class so that the network can easily recognize packets that need to be treated preferentially. Since the value is entirely handled by the network and is typically ignored by a user agent receiving an audio stream, signalling the value is only relevant for a=sendrecv and a=recvonly sessions. For these, signalling the value for the DS field is a means for recommending how the callee should tag its

outbound RTP stream. Recommending a DSCP value is only relevant if the callee is located in a known network, where DSCP values are not ignored or stripped from the packets in at least part of the transmission path.

The QoS-Recommendation may also include a DSCP value for the SIP packets.

3.6.1 Use-case

Consider a broadcaster using two separate wide area networks, A and B, managed by two different network operators. The operator managing network A specifies the use of a DS field of 46 (Expedited Forwarding) for prioritised audio streams, while the operator managing network B, specifies the use of a DS field of 34 (AF41). Also consider a user agent that will be moved back and forth between network A and B. A static configuration of the DS field will not be enough to ensure that the correct value is set in the DS field - it depends on which network the device is currently connected to.

3.6.2 Signalling

Signalling of QoS is straight-forward:

```
a=ebuacip:qosrec <dscp-value-rtp> [<dscp-value-sip>]
```

Where <dscp-value-rtp> and <dscp-value-sip> are decimal integers representing the 6 bits of the DS-field of the IP-header for rtp or sip.

3.6.2.1 QoS Recommendation offer/answer consideration

Although the offer/answer consideration in clause 2.4 above applies, if the QoS recommendation is signalled, the call must always be accepted, even if the QoS recommendation is not used.

3.6.2.2 Examples

DSCP 46 - PHB EF

```
a=ebuacip:qosrec 46
```

DSCP 34 - PHB AF41

```
a=ebuacip:qosrec 34
```

3.6.2.3 Grammar

```
qosrec         = "a=ebuacip:" "qosrec" SP dscp-value-rtp [SP dscp-value-sip]
dscp-value-rtp  = 1*(DIGIT)
dscp-value-sip  = 1*(DIGIT)
```

3.7 Protection

Protection is a generic term to describe the type of protection that is being used (if any). This might be Forward Error Protection (FEC), another type of FEC scheme or some proprietary system.

There are many ways to protect media streams. The various approaches perform best under different conditions and for that reason, user agents must be able to negotiate the protection mechanism to use from connection to connection.

The basic principles are based on RFC5109 but the mechanism there has been generalised to allow a wider range of protection mechanisms.

3.7.1 Use-case

Consider two user agents of different manufacture, A and B. User agent A supports two error correction methods, W and Z. User agent B only supports error correction method Z. User agent A invites user agent B to a call using PCM over a network known to drop a single packet every now and then, i.e. a typical corporate WAN. Every single dropped packet will cause an audible glitch in the audio. If user agent A could inform user agent B that an error correction method should be used, the audio in those lost packets will have a good chance of being repaired. If user agent A could also say that it is capable of any of the two methods W and Z, the standard negotiation of SIP increases compatibility by allowing user agent B to choose method Z for this call.

3.7.2 Protection Mechanisms

- Media duplication redundancy: When copies of the same RTP packets are transmitted on multiple streams
- Forward error correction: When specially prepared repair data is sent on a separate stream
- Multiplexing protection: When the media duplication redundancy or the forward error correction is multiplexed into the same RTP stream as the main audio

3.7.3 Additional parameters

RFC5109 specifies a mechanism to signal whether or not protection is to be used for a particular call, it also allows for selecting the type of protection. The RFC does not specify a mechanism to control details of the selected protection type. This profile specification provides a mechanism for specifying parameters for protection streams in a similar way as `a=fmtp` provides the same functionality for media streams.

One format specific parameter, ratio, is also defined for use with parity based FEC protection. This parameter specifies how many audio packets that are covered by each FEC packet. The ratio parameter essentially specifies the maximum number of bits that will be set to one in the MASK header field of the FEC RTP packets, as defined in clause 7.4 of RFC5109. This parameter enables negotiation of how much FEC data is to be sent and this allows for network usage optimisation.

Note: Ideally, the existing `a=fmtp` attribute would be used for this. However this would mean additional work on updating RFC5109. Until this feature is added, the `a=ebuacip:protp` attribute should be used.

3.7.4 Signalling

3.7.4.1 Media duplication redundancy

Media duplication redundancy streams shall be signalled using multiple media descriptions, the media address (port number) of the media descriptions must be different and the media descriptions may contain connection information ("c=") to indicate different interfaces.

3.7.4.2 Forward error correction (FEC)

RFC5109 shall be used. As with media duplication redundancy, the grouped media descriptions may contain connection information ("c=") to indicate different interfaces.

3.7.4.3 Multiplexing protection

When the protection data and the main media are transported in the same RTP stream, RFC2198 "RTP Payload for Redundant Audio Data" shall be used. RFC5109 defines how this shall be used with FEC.

3.7.4.4 Additional parameters

Signalling of the `protp` attribute is similar to the `a=fmtp` attribute:

```
a=ebuacip:protp <format> <format specific parameters>
```

This document specifies one parameter specific to types `application/ulpfec` and `application/parityfec` (see RFC3009):

```
ratio = "ratio=" 1*(DIGIT)
```

3.7.4.5 Grammar

```
protp    = "a=ebuacip:" "protp" SP (format-index) SP ratio
ratio = 1*(DIGIT)
```

3.7.4.6 ProtectionExamples

This section presents various protection examples related specifically to audio contribution use cases.

Table 2: Offering two types of forward error correction for a single audio stream

SDP	Annotation
v=0	Version 0 of SDP
o=adam 289083124 289083124 IN IP4 host.example.com	Origin
s=Test case	Session name
c=IN IP4 192.0.2.10	Default connection for all media streams
a=sendrecv	Bidirectional streams, for both audio and protection
a=group:FEC 1 2	Session level attribute specifying a group, RFC5888 FEC semantics specified in RFC5956
t=0 0	Timing (permanent)
m=audio 5004 RTP/AVP 9 8	Media 1 (Audio for the group, G722 or G711)
a=mid:1	Identifier used in the group attribute, RFC5888
m=application 5006 RTP/AVP 100 101	Media 2 (Protection for the group)
a=mid:2	Identifier used in the group attribute, RFC5888
a=rtpmap:100 ulpfec/8000	FEC according to RFC5109
a=rtpmap:101 parityfec/8000	FEC according to RFC2733
a=ebuacip:protp 100 ratio=2	50% FEC
a=ebuacip:protp 101 ratio=2	50% FEC

Table 3: Offering two types of forward error correction for a single audio stream, the protection stream and the audio stream use separate network interfaces

SDP	Annotation
v=0	Version 0 of SDP
o=adam 289083124 289083124 IN IP4 host.example.com	Origin
s=Test case	Session name
a=sendrecv	Bidirectional streams, for both audio and protection
a=group:FEC 1 2	Session level attribute specifying a group, RFC3388 FEC semantics specified in RFC5956
t=0 0	Timing (permanent)
m=audio 5004 RTP/AVP 9 8	Media 1 (Audio for the group, G722 or G711)
c=IN IP4 192.0.2.10	Connection for the audio stream
a=mid:1	Identifier used in the group attribute, RFC5888
m=application 5006 RTP/AVP 100 101	Media 2 (Protection for the group)
c=IN IP4 192.0.2.40	Connection for the protection stream
a=mid:2	Identifier used in the group attribute, RFC5888
a=rtpmap:100 ulpfec/8000	FEC according to RFC5109
a=rtpmap:101 parityfec/8000	FEC according to RFC2733
a=ebuacip:protp 100 ratio=1	100% FEC
a=ebuacip:protp 101 ratio=1	100% FEC

Table 4: Offering stream replication on a separate network interface

SDP	Annotation
v=0	Version 0 of SDP
o=adam 289083124 289083124 IN IP4 host.example.com	Origin
s=Test case	Session name
a=sendrecv	Bidirectional streams
a=group:FID 1 2	Session level attribute specifying a group, RFC5888
t=0 0	Timing (permanent)
m=audio 5004 RTP/AVP 9 8	Media 1 (Audio for the group, G722 or G711)
c=IN IP4 192.0.2.10	Connection for audio stream 1
a=mid:1	Identifier used in the group attribute, RFC5888
m=audio 5004 RTP/AVP 9 8	Media 2 (Audio for the group, G722 or G711)
c=IN IP4 192.0.2.40	Connection for audio stream 2
a=mid:2	Identifier used in the group attribute, RFC5888

Table 5: Offering FEC or stream duplication for a single audio stream, the protection stream and the audio stream are multiplexed in the same connection (on the same port)

SDP	Annotation
v=0	Version 0 of SDP
o=adam 289083124 289083124 IN IP4 host.example.com	Origin
s=Test case	Session name
c=IN IP4 192.0.2.10	Connection for the audio stream
a=sendrecv	Bidirectional streams, for both audio and protection

t=0 0	Timing (permanent)
m=audio 5004 RTP/AVP 121 9 8 100	Media (Redundancy, g722, g711, ulpfec). As redundancy (121) is first in list, this is the default format for this session
a=rtpmap:121 red/8000/1	Redundancy according to RFC2198
a=rtpmap:100 ulpfec/8000	FEC according to RFC5109
a=fmtp:121 100/9/8	Order of preference for the format of the redundancy packets is ulpfec, duplicated g722 or duplicated g711 (doesn't have to be the same order as in "m=" line)
a=ebuacip:protp 100 ratio=1	100% FEC

This means that there will be two payload types in the actual RTP stream, 121 and either 9 or 8. The packets with payload type 121 will contain one of ulpfec, g722 or g711.

4. Example profile descriptions

The following tables show example sets using all the parameters that have been defined within this document and how they could be used in a profile.

Table 6: Example profile 1

SDP	Annotation
v=0	Version 0 of SDP
o=adam 289083124 289083124 IN IP4 host.example.com	Origin
s=Test case	Session name
c=IN IP4 192.0.2.10	Default connection for all media streams
b=AS: 200	Bandwidth information (AS = application specific)
a=sendrecv	Bidirectional streams, for both audio and protection
a=ebuacip:jb 0 1 2	Three jitter buffer options including a range
a=ebuacip:jbdef 0 fixed 20	Fixed jitter buffer of 20 ms
a=ebuacip:jbdef 1 auto 20-50	Automatic jitter buffer range min 20 ms max 50 ms
a=ebuacip:jbdef 2 fixed 20-100	Fixed jitter buffer range min 20 ms max 100 ms
a=ebuacip:qosrec 46	Recommended QoS of Expedited Forwarding for return RTP media
	Section for protection - offering two types of forward error correction for a single audio stream
a=group:FEC 1 2	Session level attribute specifying a group, RFC5888 FEC sematics specified in RFC5956
t=0 0	Timing (permanent)
m=audio 5004 RTP/AVP 9 8	Media 1 (Audio for the group, G722 or G711)
a=mid:1	Identifier used in the group attribute, RFC5888
a=ebuacip:plength 9 4	4 ms packet length for format 9 (G.722)
a=ebuacip:plength 8 4	4 ms packet length for format 8 (G.711a – PCM A-law)
m=application 5006 RTP/AVP 100 101	Media 2 (Protection for the group)
a=mid:2	Identifier used in the group attribute, RFC5888
a=rtpmap:100 ulpfec/8000	FEC according to RFC5109
a=rtpmap:101 parityfec/8000	FEC according to RFC2733

a=ebuacip:protp 100 ratio=2	50% FEC
a=ebuacip:protp 101 ratio=2	50% FEC

Table 7: Example profile 2

SDP	Annotation
v=0	Version 0 of SDP
o=alice 1818 114 IN IP4 172.129.171.79	Origin
s=Talk	Session name
c=IN IP4 172.129.171.79	Default connection for all media streams
a=sendrecv	Bidirectional streams, for both audio and protection
a=ebuacip:version 0	Profile version
a=ebuacip:jb 0	A single jitter buffer option
a=ebuacip:jbdef 1 auto 10-40	Automatic jitter buffer range min 10 ms max 40 ms
a=ebuacip:qosrec 34	Recommended QoS of Assured Forwarding for return RTP media
	Section for protection - Offering FEC or stream duplication for a single audio stream, the protection stream and the audio stream are multiplexed in the same connection (on the same port)
t=0 0	Timing (permanent)
m=audio 5004 RTP/AVP 121 9 8 100	Media (Redundancy, g722, g711, ulpfec). As redundancy (dynamic payload type 121) is first in list, this is the default format for this session
a=rtpmap:121 red/8000/1	Redundancy according to RFC2198
a=rtpmap:100 ulpfec/8000	FEC according to RFC5109
a=ebuacip:plength 9 4	4 ms packet length for format 9 (G.722)
a=ebuacip:plength 8 4	4 ms packet length for format 8 (G.711a – PCM A-law)
a=fmtp:121 100/9/8	Order of preference for the format of the redundancy packets is ulpfec, duplicated g722 or duplicated g711 (doesn't have to be the same order as in "m=" line)
a=ebuacip:protp 100 ratio=1	100% FEC

This means that there will be two payload types in the actual RTP stream, 121 and either 9 or 8. The packets with payload type 121 will contain one of ulpfec, g722 or g711.

5. Bibliography

References

The following RFCs may be found on the IETF website at http://www.ietf.org/rfc.html unless otherwise stated.

1. RFC2198: RTP Payload for Redundant Audio Data
2. RFC2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers
3. RFC2733: An RTP Payload Format for Generic Forward Error Correction
4. RFC3009: Registration of parityfec MIME types

5. RFC3168: The Addition of ECN to IP (This updates RFC2474)
6. RFC3260: New Terminology and Clarifications for Diffserv (This updates RFCs 2474, 2475, 2597 – Informational)
7. RFC3261: SIP: Session Initiation Protocol (SIPv2)
8. RFC3264: An Offer/Answer Model Session Description Protocol
9. EBU Tech 3326 (http://tech.ebu.ch/docs/tech/tech3326.pdf)
10. RFC3550: RTP: A Transport Protocol for Real-Time Applications
11. RFC3551: RTP: Profile for Audio and Video Conferences with Minimal Control (registers the name RTP/AVP)
12. RFC4566: SDP: Session Description Protocol
13. RFC5109: RTP Payload Format for Generic Forward Error Correction
14. RFC5234: Augmented BNF for Syntax Specifications-ABNF
15. RFC5888: SDP Grouping Framework
16. RFC5956: FEC Grouping Semantics in SDP
17. RFC6354: Forward-Shifted RTP Redundancy Payload Support (This updates: RFCs 2198, 4102)

6. Glossary

FEC	Forward Error Correction
DS	DiffServ
DSCP	Differentiated Services Code Point
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IP	Internet Protocol
RFC	Request For Comments (IETF standard)
RTP	Realtime Transport Protocol.
SDP	Session Description Protocol
SIP	Session Initiation Protocol
WAN	Wide Area Network

Annex A (Normative)

Session description types defined within the present document

This annex contains definitions for SDP parameters that are specific to SDP usage in EBU ACIP Profiles and therefore are not described in an RFC.

Profile Attribute

Editor's note: This annex forms the basis for IANA registration of the new SDP attribute. The registration should be performed by the EBU when the profiles work is declared 100% complete.

General

The profile attribute is used to send information, required by broadcasters, to the callee in addition to the negotiation parameters defined in EBU Tech 3326 and SIP (RFC3261).

ABNF Syntax

The EBUACIP profile description is a value attribute which is encoded as both a media level and session level SDP attribute with the ABNF syntax defined in Table 8. ABNF is defined in RFC5243.

Table A1: ABNF Syntax of `ebuacip` attribute

```
EBU-ACIP-Profile = "a=ebuacip: " ebuacip-parameter
ebuacip-parameter = version / jb / jbdef / plength / qosrec / protp / token
```

"version" defines the EBU ACIP protocol version number.

"jb" is the received jitter buffer to be applied.

"jbdef" defines each "jb" option in detail.

"plength" is the quantity of audio in each packet.

"qosrec" is the recommended QoS of service to be applied to the return stream.

"protp" defines the percentage of FEC to be applied.

This version of the specification only defines usage of the "version", "`jb`", "`jbdef`", "`plength`", "`qosrec`" and "`protp`" attribute values. Other values shall be ignored.

The "`ebuacip`" attribute is characterset-independent.

IANA Registration

NOTE: This subclause contains information to be provided to IANA for the registration of the `ebuacip` profile SDP attribute.

Contact name, email address, and telephone number:

EBU-FNS ACIP II Coordinator

coinchon@ebu.ch

+41(0)22 717 27 15

Attribute Name (as it will appear in SDP):

ebuacip

Long-form Attribute Name in English:

EBU ACIP (Audio Contribution over IP) Profile

Type of Attribute:

Media level

Is Attribute Value subject to the Charset Attribute?

This Attribute is not dependent on charset.

Purpose of the attribute:

This attribute specifies a set of parameters, called a profile, to be negotiated with an end user agent.

Appropriate Attribute Values for this Attribute:

The attribute is a value attribute. The values "`version`", "`jb`", "`jbdef`", "`plength`", "`qosrec`" and "`protp`" are defined.

Annex B (Normative)

Profiles Grammar

This annex only provides an Augmented BNF grammar for the profiles defined in this document. It does not include the additional grammar required for a full SDP description. This may be found in clause 9, SDP Grammar of RFC4566. ABNF is defined in RFC5234. Within the profiles grammar, there are no other types apart from attribute fields.

Table B1: ABNF Grammer SDP Profile Syntax

```
;SDP Profiles Syntax
EBU-ACIP-Profile = "a=ebuacip:" ebuacip-parameter
ebuacip-parameter = ebuacip-version
                    ebuacip-jb
                    ebuacip-jbdef
                    ebuacip-plength
                    ebuacip-qosrec
                    ebuacip-protp
                    token

ebuacip-version = "a=ebuacip:" "version" SP 1*(DIGIT)

;jb definition
ebuacip-jb = "a=ebuacip:" "jb" 1*(SP ebuacip-jb-index)
;jbdef definition
ebuacip-jbdef = "a=ebuacip:" "jbdef" SP (ebuacip-jb-index) SP (ebuacip-jb-option)
ebuacip-jb-option = "fixed" SP 1*(DIGIT)["-"1*(DIGIT)] / "auto" SP 1*(DIGIT)"-
"1*(DIGIT)
ebuacip-jb-index = 1*(DIGIT)

;plength definition
ebuacip-plength = "a=ebuacip:" "plength" SP (ebuacip-format-index) SP 1*(DIGIT)
ebuacip-format-index = 1*(DIGIT)

;qosrec definition
ebuacip-qosrec = "a=ebuacip:" "qosrec" SP ebuacip-dscp-value-rtp [SP ebuacip-dscp-
value-sip]
ebuacip-dscp-value-rtp = 1*(DIGIT)
ebuacip-dscp-value-sip = 1*(DIGIT)

;protp definition
ebuacip-protp = "a=ebuacip:" "protp" SP (ebuacip-format-index) SP ebuacip-ratio
ebuacip-ratio = 1*(DIGIT)

; generic sub-rules: datatypes

token-char = %x21 / %x23-27 / %x2A-2B /%x2D-2E / %x30-39
/ %x41-5A / %x5E-7E

token = 1*(token-char)

; external references:
; ALPHA, DIGIT, CRLF, SP, VCHAR: from RFC5234
```

The above ABNF will produce the following output from the suggested ABNF checker provided by the IETF: http://apps.mrochek.com/content/chris-newmans-abnf-validator

```
;unreferenced rule: EBU-ACIP-Profile
;ABNF validation (version 1.0) completed
```

Annex C (Informative)

Ancillary Data

Although no particular mechanism for signalling ancillary data is, as yet, mandated by this standard, it remains a desirable requirement for broadcasters to use. It is expected that this requirement will be included in a future edition of this specification.

Ancillary data is data sent in addition to the media stream and may be multiplexed in with the media stream or sent as a separate stream.

The data may be required to perform some action at the receiver, coinciding with particular points in time of the media stream content, i.e. the data action is synchronised to the media stream content. For example, the data sent will operate a relay to switch a lamp on to indicate that the content is now "On Air".

Alternatively, the data may be required to perform some action at the receiver, but not directly associated with any particular point in time in the media stream content. For example, the user may send some data asynchronously to indicate that they are ready to go "on air" within the next minute.

The signalling of ancillary data may be done either via SDP or proprietary means.

Devices are not required to implement signalling of ancillary data.