The European Organisation for Civil Aviation Equipment
L'Organisation Européenne pour l'Equipement de l'Aviation Civile

Interoperability Standards for VoIP ATM Components

Part 3: Recording

ED-137 Part 3
"Month Year"

102 rue Etienne Dolet
92240 MALAKOFF, France
Web Site: www.eurocae.eu

Tel: 33 1 40 92 79 30
Fax: 33 1 46 55 62 65
Email: eurocae@eurocae.net

Interoperability Standards for VoIP ATM Components

Part 3: Recording

ED-137 Part 3
"Month Year"

FOREWORD

1 The document ED-137 "**Interoperability Standards for VoIP ATM Components**" was prepared by EUROCAE Working Group 67 and was accepted by the Council of EUROCAE on "Month Year".

2 EUROCAE is an international non-profit making organisation. Membership is open to manufacturers and users in Europe of equipment for aeronautics, trade associations, national civil aviation administrations and non-European organisations. Its work programme is principally directed to the preparation of performance specifications and guidance documents for civil aviation equipment, for adoption and use at European and world-wide levels.

3 The findings of EUROCAE are resolved after discussion among its members and, where appropriate, in collaboration with RTCA Inc, Washington D.C. USA and/or the Society of Automotive Engineers (SAE), Warrendale, PA, USA through their appropriate committee.

4 The document represents "the minimum specification required for Manufacturers and Users to assure Interoperability between VoIP ATM Components".

5 EUROCAE performance specifications are recommendations only. EUROCAE is not an official body of the European governments; its recommendations are valid statements of official policy only when adopted by a particular government or conference of governments.

6 Copies of this document may be obtained from:

EUROCAE
102 rue Etienne Dolet
92240 MALAKOFF
France

Tel: 33 1 40 92 79 30
Fax: 33 1 46 55 62 65
Email: eurocae@eurocae.net
Web Site: www.eurocae.org

TABLE OF CONTENTS

FIGURE INDEX

TABLE INDEX

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Ground-Ground (G-G) ATM voice systems have been based upon analogue and more recently, digital Time Division Multiplexing / Pulsed Code Modulation (TDM/PCM) technologies for many years.

Nowadays, however, convergence of voice and data into one multimedia network is a popular trend with a variety of technical solutions available on the market. Following in this direction ATM communication networks are adopting, by a process of gradual evolution, a common infrastructure for voice and data services.

As the technology has developed IP Technology has now the true potential to fulfil operational and technical ATM communication requirements - including those of voice / data convergence, Quality of Services (QoS), security and safety. There is also the possibility that IP may deliver solutions that will, over time, bring about true savings in investment and operating costs.

EUROCAE Working Group 67 (WG-67) undertook the mission to assess the feasibility of using Voice over Internet Protocol (VoIP) for providing ATM voice services. The group defined criteria, requirements and guidelines based upon the following operational needs and constraints:

- Operational and Technical Air-Ground (A-G) and Ground-Ground (G-G) ATM Voice system requirements
- Existing IP Voice protocols and signalling standards
- IP network capabilities for Voice services
- Security, Quality of Service (QoS), and Convergence (infrastructure, protocol, applications)
- Existing IP Voice ATM system capabilities and service interfaces.

The following tasks were identified to fulfil the WG-67 mission:-

- Define ATM Systems and identify their components (Voice Communication System / VCS, Ground-based Radio Station / GRS)
- Determine possible additional operational and technical ATM requirements for new ATM voice systems, also taking into consideration A-G communications.
- Make recommendations to upgrade current standardisation documents.
- Develop a Technical Specification for a VoIP Voice ATM System including:
    - Minimum performance and safety/security requirements for the system and, if appropriate, for components;
    - Interoperability requirements between IP components of the VoIP ATM system;
    - Minimum performance requirements of an IP Network to support ATM Voice services;
    - Guidelines for qualification tests of VoIP ATM systems and their components.

Consequently the following four documents were delivered:

ED-136 - VoIP ATM System Operational and Technical Requirements

ED-137 - Interoperability Standards for VoIP ATM Components

ED-138 - Network Requirements and Performances for VoIP ATM Systems

ED-139 - Qualification tests for VoIP ATM Components and Systems

The contents of all four documents are premised on the **"Vienna Agreement"** which defines the different components of a VoIP ATM system and their mutual interfaces as depicted in Fig. 1.

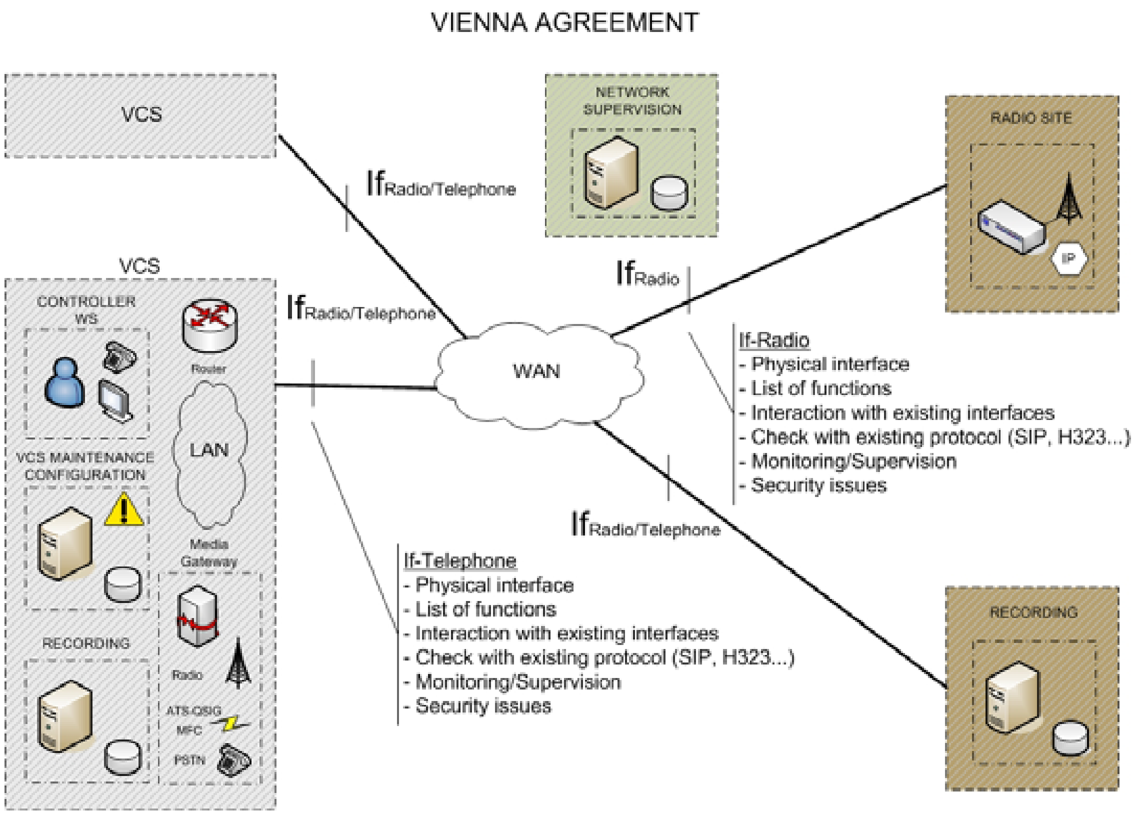

Fig. 1 – Vienna Agreement

VoIP components are interconnected through an IP network and suppliers are free to define their internal architecture (IP/Ethernet, TDM/PCM - Time Division Multiplexing / Pulsed Code Modulation, …). Between VoIP components, required interfaces are defined to guarantee their functional and technical interoperability.

Therefore, VoIP ATM Systems are composed of:

- **VoIP VCS Components** performing Radio and / or Telephone functions, including:
    1. Controller Working Positions, assuring HMI including voice devices (microphone and loudspeaker);

2. Possible local VCS Maintenance and Configuration stations;
3. Possible local Recording devices;
4. Possible LAN for local interconnection;
5. Possible Media Gateways to legacy systems (ATS-QSIG, ATS-R2, ATS-No.5, PSTN, Radio analogue lines, …).

- **VoIP Ground Radio Station Components** performing AM VHF and UHF Radio functions.

- **VoIP Supervision System Components** performing monitoring and control functions.

- **VoIP Recording Components** performing recording functions.

- **IP WAN Components** performing interconnection services between two or more different physical components.

1.2 ED-137 PRESENTATION

The scope of the WG67 ED-137 Document is to define the rules for VoIP implementations to support ATM communications. This includes the performances requested for radio (Part 1 of ED-137), the existing signalling in use for telephone (Part 2 of ED-137), for recording (Part 3 of ED-137) and for supervision (Part 4 of ED-137).

The present document, that is the Part 3 of the ED-137, proposes a profile standard for the use of RTSP to establish, terminate and modify recording sessions of the Ground Telephone Service and the Radio Service in an Air Traffic Services Ground Voice Network (AGVN).

RTSP is an application layer protocol for establishing, terminating and modifying multimedia sessions. It is typically carried over the Internet Protocol (IP) (IETF RFC 791 [2] and IETF RFC 2460 [6]). RTSP is defined in IETF RFC 2326 [5].

This document proposes a specification for the signalling profile both for basic services that provide a bi-directional transfer capability for speech media between user terminals, radios and a recorder in an IP AGVN employing SIP and RTSP in support of ATS recording services.

Interworking between an IP AGVN and a public IP network is out of scope of this document.

1.3 TERMINOLOGY FOR REQUIREMENTS, RECOMMENDATIONS AND OPTIONS

The terminology for requirements, recommendations and options in this document is based on RFC 2119 [4], which specifies Best Current Practice regarding the use of Key Words for the Internet Community. As such, the following terminology is applied:

- The word **SHALL** denotes a mandatory requirement;
- The word **SHOULD** denotes a recommendation;
- The word **MAY** denotes an option.

To avoid confusion with their natural meanings in the English language, the words **SHALL**, **SHOULD**, and **MAY** take on the meaning stated above only where printed in boldface. When printed in normal (Arial) typeface, the natural English meaning is meant.

Detailed description of terminology:

1. **SHALL** This word has the same meaning as the phrase "REQUIRED" and means that the definition is an absolute requirement of the specification.

2. **SHALL NOT** This phrase means that the definition is an absolute prohibition of the specification.

3. **SHOULD** This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

4. **SHOULD NOT** This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.

5. **MAY** This word, or the adjective "OPTIONAL", mean that an item is truly optional.

CHAPTER 2

RECORDING MODEL

2.1 ACTIVE RECORDING

1 [RECORDING] Active Recording

Recording **SHALL** be based on active sessions opened from clients (User Terminal, Radio Transmitter/Receiver or specific 3rd party devices) to one recording device (or two devices required for redundancy). Active means that any client that sends or receives media streams (i.e. audio) takes the responsibility to send a copy of either stream to the recorders. The used time source **SHALL** be synchronized to the ATSU time source. This is assumed to be Universal Time Coordinated (UTC) to the accuracy specified by ICAO.

Note: In order to simplify drawings, the following just mentions a single recording device. All described mechanisms are valid for two or a defined number of recorders.

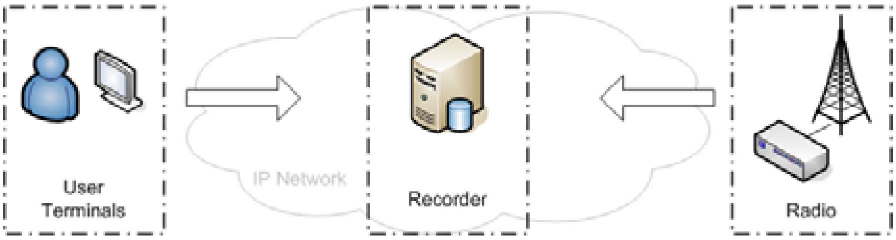

Fig. 1 – Recording Sessions

2.2 RECORDING PHONE COMMUNICATION

2 [RECORDING] Phone Communication

User Terminals participating a G/G communication session **SHALL** provide a single audio stream that summarizes all incoming (IN) and outgoing (OUT) audio streams.

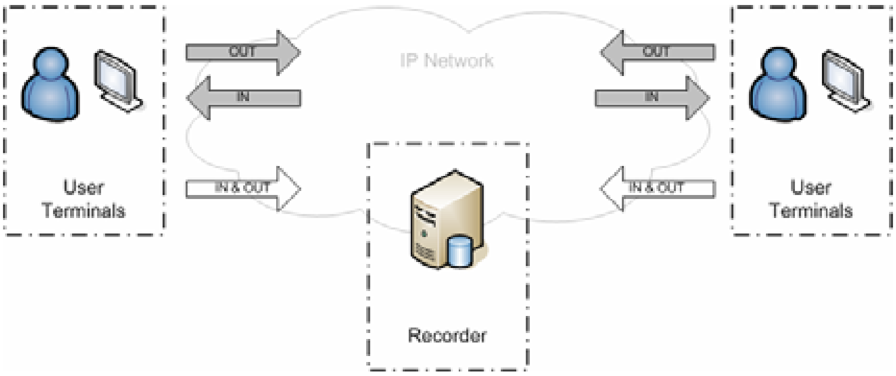

Fig. 2– Recording Phone Communication

2.3 RECORDING RADIO COMMUNICATION

3 [RECORDING] Radio Communication

User Terminals participating a A/G communication session **SHALL** provide a single audio stream that summarizes all received (RX) and transmitted (TX).

Radios (or Gateways connecting legacy radios to an IP network) **SHALL** provide a single audio stream that contains the received (RX) audio stream related to a single radio channel.

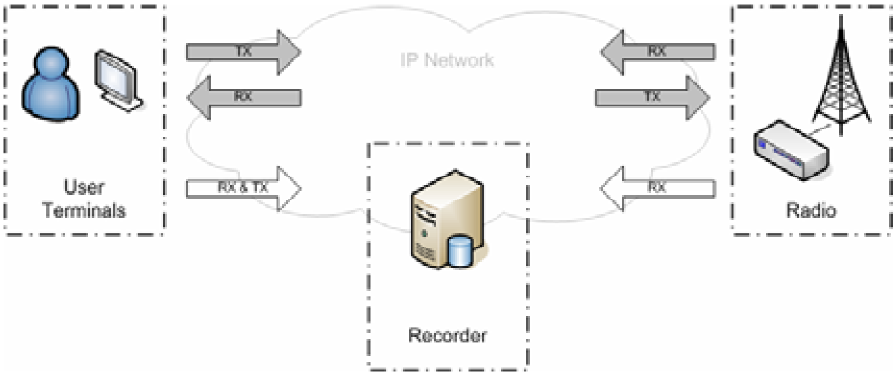

Fig. 3– Recording Radio Communication

2.4 SESSION SETUP

Active recording requires an established session (i.e. a certain number of parameters that are exchanged between entities prior to any recording). User Terminal, Radio and Recorder **SHALL** use RTSP for such sessions. As RTSP relies on a transport layer protocol (TCP or UDP), these entities **MAY** use SIP to exchange capabilities and connection information (i.e. IP address, port number, and transport protocol). The following section describes the session setup using SIP and RTSP.

2.4.1 SIP

Note that this section assumes that SIP is used for session setup hence the terminology for requirements, recommendations and options is only valid for this case.

Any entity involved in a recording session (User Terminal and Recorder) **SHOULD** register with a SIP Registrar using the REGISTER method according RFC3261. It **SHOULD** be possible to register multiple contacts for a single Address of Record (AOR).

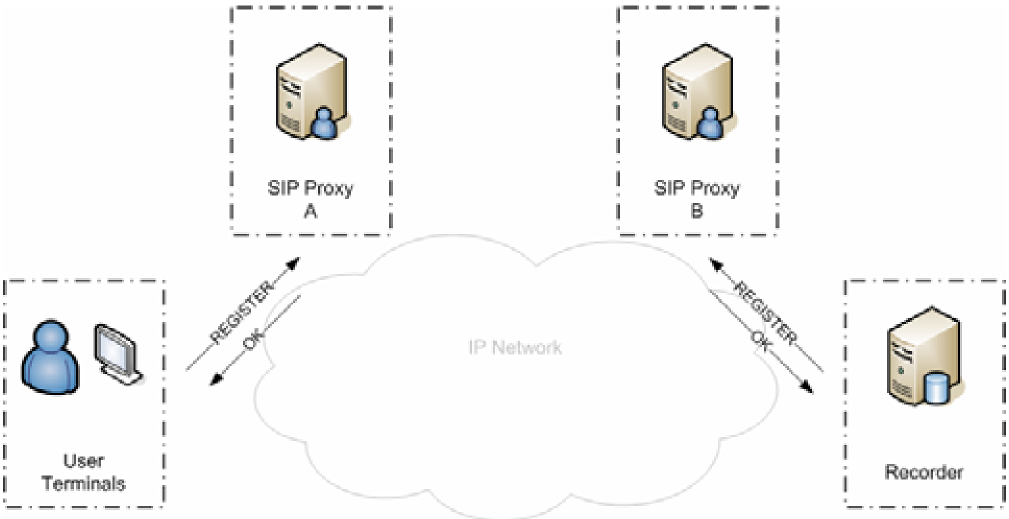

Fig. 4– SIP Registration

Participants (User Terminal, Radio) **SHALL** use INVITE to establish a session. This session setup provides the session description (connection information) and media description (media name and transport address) of each participant.

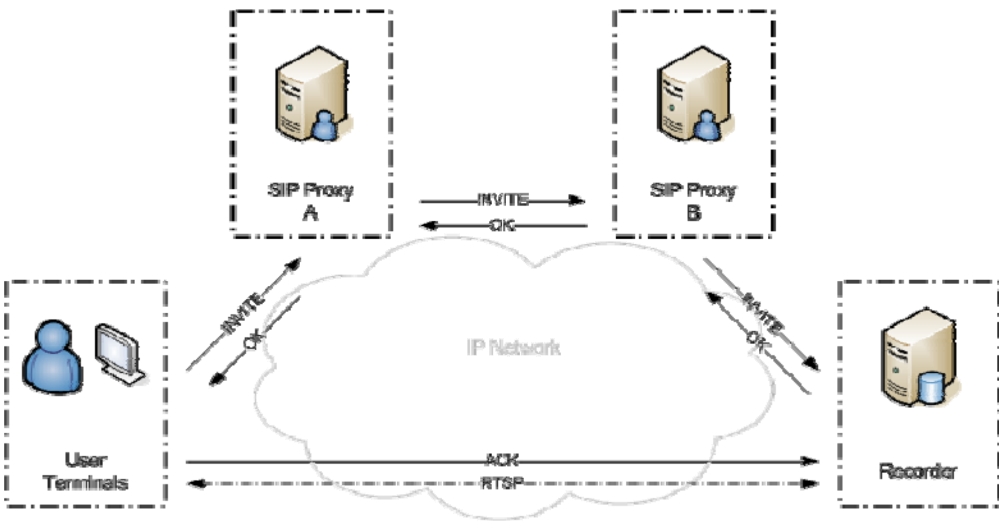

Fig. 5– SIP Session Setup

```
 Recorder          Terminal          Radio          Recorder
          |   INVITE       |             |                |
          | <------------- |             |                |
          |    200 OK      |             |     INVITE     |
          | -------------> |             | -------------> |
          |     ACK        |             |     200 OK     |
          | <------------- |             | <------------- |
          |                |             |      ACK       |
          |     TCP        |             | -------------> |
          | <============> |             |                |
          |                |             |      TCP       |
          |                |             | <============> |
          |                |             |                |
```

Fig. 6– SIP Session Setup: Message Sequence

Example for a SIP session setup from a User Terminal or Radio to the Recorder:

Request:

```
INVITE sip:recoder@atc.org SIP/2.0
...
Content-Type: application/sdp
Content-Length: 87

v=0
o=0 0 IN IP4 192.0.2.94
s=Recording
t=0 0
c=IN IP4 192.0.2.94
m=application 10554 rtsp rec
```

Response:

```
SIP/2.0 200 OK
...
Content-Type: application/sdp
Content-Length: 87

v=0
o=0 0 IN IP4 192.0.2.25
s=Recording
t=0 0
c=IN IP4 192.0.2.25
m=application 20554 rtsp rec
```

The session description **SHALL NOT** specify the used transport protocol, as this is part of the RTSP session description.

2.4.2 RTSP

User Terminals **SHALL** use RTSP to enable controlled, on-demand delivery of real-time data. Systems implementing RTSP **SHOULD** support carrying RTSP over TCP and **MAY** support UDP. The default port for the RTSP server **SHALL** be 554 for both UDP and TCP.

The following assumes that the IP address of the Recorder is known and a TCP session has been established. Participants (User Terminals, Radios) **SHALL** use ANNOUNCE and SETUP to establish a recording session. Participants (User Terminals) **MAY** use DESCRIBE and SETUP to establish a replay session. This session setup provides the session description (connection information) and media description (media name and transport address) of each participant. Participants (User Terminals, Radios) **SHALL** use TEARDOWN to close a session.

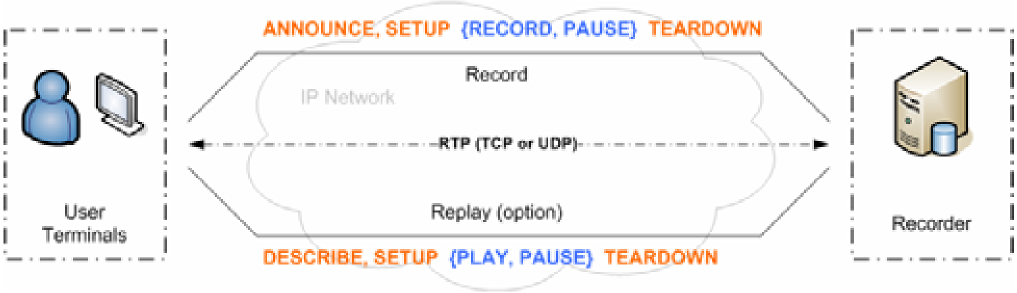

Fig. 7– RTSP Record and Replay Session

2.5 TRANSPORT

4 [RECORDING] Transport

Transport of media **SHOULD** be based on Embedded (interleaved) Binary Data and **MAY** be based on RTP over independent TCP or RTP over UDP, as described later in this section. The Transport request and response header field indicates which transport protocol is to be used and configures its parameters such as destination address, compression, multicast time-to-live and destination port for a single stream. It sets those values not already determined by a presentation description.

Transports are comma separated, listed in order of preference. Parameters may be added to each transport, separated by a semicolon. The server **SHOULD** return a Transport response-header field in the response to indicate the values actually chosen. The Transport header field **MAY** also be used to change certain transport parameters. A server **MAY** refuse to change parameters of an existing stream.

The general syntax for the transport specifier is a list of slash separated tokens:

```
Value1/Value2/Value3...
```

Which for RTP transports take the form:

```
RTP/profile/lower-transport
```

The default value for the "`lower-transport`" parameters is specific to the profile. For RTP/AVP, the default is UDP. The next section describes alternative transport methods.

2.5.1 EMBEDDED BINARY DATA

RTSP contains a syntax for interleaving the RTSP control stream with the data stream. This is called embedded (interleaved) binary data. Interleaved binary data **SHOULD** be used when RTSP is carried over TCP.

The channel identifier (CID) is defined in the transport header with the interleaved parameter. The following illustrates a client server session example using interleaved binary data with `0` as channel identifier.

Request:

```
SETUP rtsp://recorder:554/iprecorder/ RTSP/1.0
CSeq: 1
Transport: RTP/AVP/TCP;interleaved=0
```

Response:

```
RTSP/1.0 200 OK
CSeq: 1
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
Transport: RTP/AVP/TCP;interleaved=0
```

2.5.1.1 FRAMING METHOD

Stream data such as RTP packets is encapsulated by an ASCII dollar sign (24 hexadecimal), followed by a one-byte channel identifier (CID), followed by the length of the encapsulated binary data as a binary, two-byte integer in network byte order. The stream data follows immediately afterwards, without a CRLF, but including the upper-layer protocol headers. Each $ block contains exactly one upper-layer protocol data unit, e.g., one RTP packet, see [5].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 ---------------------------------------------------------------
|   ASCII ´$´    |   CID          |          LENGTH              |
 ---------------------------------------------------------------
|          RTP packet with header ...                            |
 ---------------------------------------------------------------
```

Fig. 8– Embedded Binary Data Format

2.5.2 RTP OVER INDEPENDENT TCP

This section adapts the guidelines for using RTP over TCP within SIP/SDP to work with RTSP as in [21].

There are two different methods for how to specify where the media should be delivered:

- `dest_addr`: The presence of this parameter and its values indicates the destination address or addresses (host address and port pairs for IP flows) necessary for the media transport.

- No `dest_addr`: The lack of the dest_addr parameter indicates that the server SHALL send media to the same address for which the RTSP messages originates. This does not work for transports requiring explicitly given destination ports.

A client codes the support of RTP over independent TCP by specifying an RTP/AVP/TCP transport option without an interleaved parameter. This transport option **MUST** include the "`unicast`" parameter. If the client wishes to use RTP with RTCP, two ports (or two address/port pairs) are specified by the `dest_addr` parameter. If the client wishes to use RTP without RTCP, one port (or one address/port pair) is specified by the `dest_addr` parameter.

If the client wishes to play the active role in initiating the TCP connection, it **MAY** set the "`setup`" parameter on the Transport line to be "`active`", or it **MAY** omit the setup parameter, as active is the default. If the client signals the active role, the ports for all `dest_addr` values **MUST** be set to 9 (the discard port).

If the client wishes to play the passive role in TCP connection initiation, it **MUST** set the "`setup`" parameter on the Transport line to be "`passive`". If the client is able to assume the active or the passive role, it **MUST** set the "`setup`" parameter on the Transport line to be "`actpass`". In either case, the `dest_addr` port value for RTP **MUST** be set to the TCP port number on which the client is expecting to receive the RTP stream connection, and the `dest_addr` port value for RTCP **MUST** be set to the TCP port number on which the client is expecting to receive the RTCP stream connection.

If upon receipt of a non-interleaved RTP/AVP/TCP request, a server decides to accept this requested option, the 2xx reply **MUST** contain a Transport option that specifies RTP/AVP/TCP (without using the interleaved parameter, and with using the unicast parameter).
The `dest_addr` parameter value **MUST** be echoed from the parameter value in the client request unless the destination address (only port) was not provided in which can the server **MAY** include the source address of the RTSP TCP connection with the port number unchanged.

In addition, the server reply **MUST** set the setup parameter on the Transport line, to indicate the role the server will play in the connection setup. Permissible values are "`active`" (if a client set "`setup`" to "`passive`" or "`actpass`") and "`passive`" (if a client set "`setup`" to "`active`" or "`actpass`").

If a server sets "`setup`" to "`passive`", the "`src_addr`" in the reply **MUS**T indicate the ports the server is willing to receive an RTP connection and (if the client requested an RTCP connection by specifying two dest_addr ports or address/port pairs) and RTCP connection. If a server sets "`setup`" to "`active`", the ports specified in "`src_addr`" MUST be set to 9.

The following illustrates a client server session example using RTP over independent TCP.

Request:

```
SETUP rtsp://recorder:554/iprecorder/ RTSP/1.0
CSeq: 1
Transport: RTP/AVP/TCP;unicast;mode="RECORD";dest_addr=":9";setup=active;connection=new
```

Response:

```
RTSP/1.0 200 OK
CSeq: 1
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
Transport: RTP/AVP/TCP;unicast;dest_addr=":9";src_addr="192.0.2.5:9000";setup=passive
           connection=new;ssrc=93CB001E
```

2.5.2.1 FRAMING METHOD

A 16-bit unsigned integer LENGTH field, coded in network byte order (big-endian), begins the frame. If LENGTH is non-zero, an RTP or RTCP packet follows the LENGTH field. The value coded in the LENGTH field **MUST** equal the number of octets in the RTP or RTCP packet. Zero is a valid value for LENGTH, and it codes the null packet, as in [25].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 ---------------------------------------------------------------
|             LENGTH              |  RTP or RTCP packet ...     |
 ---------------------------------------------------------------
```

Fig. 9– TCP Frame Format

2.5.3 RTP OVER UDP

The implementation of RTP over UDP **SHALL** be implemented according the guidelines of RFC2326, see [5].

2.6 RTSP CONTROL MESSAGES

2.6.1 ANNOUNCE AND SETUP

These messages **SHALL** be used to establish a recording session. The message body of ANNOUNCE **SHALL** contain a description of the media referenced by the requested URL, (e.g. `rtsp://recorder:554/iprecorder/`) using SDP, as in [8].

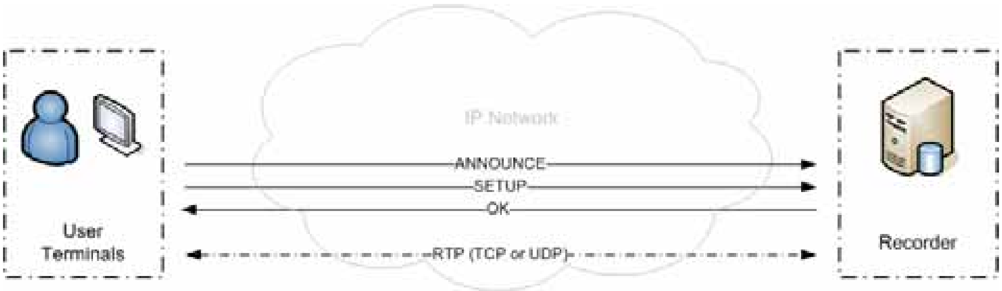

Fig. 10– RTSP Record Session Setup

The following gives an example for a RTSP session setup using embedded (interleaved) binary data (request and response):

```
ANNOUNCE rtsp://recorder:554/iprecorder/ RTSP/1.0
CSeq: 1
Content-Type: application/sdp
…
v=0
o=first 2520644554 2838152170 IN IP4 first.example.net
s=Example
t=0 0
c=IN IP4 192.0.2.105
m=audio 0 RTP/AVP 8
a=rtpmap:8 PCMA/8000
…
```

```
SETUP rtsp://recorder:554/iprecorder/ RTSP/1.0
CSeq: 1
Transport: RTP/AVP/TCP;interleaved=0
…
```

```
RTSP/1.0 200 OK
CSeq: 1
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
Transport: RTP/AVP/TCP;interleaved=0
…
```

```
   Terminal         Recorder
        |  ANNOUNCE      |
        |------------->  |
        |    SETUP       |
        |------------->  |
        |    200 OK      |
        |<-------------  |
        |                |
        |     RTSP       |
        |<============>  |
        |                |
```

Fig. 11– RTSP Record Session Setup: Messages and Sequence

2.6.2 RECORD

This message **SHALL** be used to start data transmission on the stream allocated via SETUP. Clients (Terminals) **MAY** offer a connection reference to the recorder using an XML encoded message body (see section 2.7 for details). If clients are not able to provide a connection reference in their initial request, the answer or server response **SHALL** contain a server generated connection reference.

However, clients **MAY** already submit call record data using the defined XML structure (see section 2.7 for details) within the RECORD message and **SHALL** leave the `connref` parameter blank if they are not able to provide a connection reference value.

If the connection reference is provided by the client (request), the server (recorder) **SHALL** use the same `connref` value in the response. The following gives an example to start recording including a client generated connection reference value (request and response):

```
RECORD rtsp://recorder:554/iprecorder/ RTSP/1.0
CSeq: 2
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
Content-Type: application/x-crd+xml

<call-record-data connref="403C232A-C510-45C7-973E-D55F5CF996AF" />
(see section 2.7. for content details)
```

```
RTSP/1.0 200 OK
CSeq: 2
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
Content-Type: application/x-crd+xml

<call-record-data connref="403C232A-C510-45C7-973E-D55F5CF996AF" />
(see section 2.7. for content details)
```

```
       T/R            Recorder
        |    RECORD      |
        | -------------> |
        |    200 OK      |
        | <------------- |
        |   RTP (media)  |
        | =============> |
```

Fig. 12– RECORD: Messages and Sequence

2.6.3 PAUSE

This message **SHALL** be used to interrupt (halt) stream delivery on the stream allocated via ANNOUNCE/SETUP (request and response):

```
PAUSE rtsp://recorder:554/iprecorder/ RTSP/1.0
CSeq: 2
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
```

```
RTSP/1.0 200 OK
CSeq: 2
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
```

```
       T/R            Recorder
        |     PAUSE      |
        | -------------> |
        |    200 OK      |
        | <------------- |
        |                |
```

Fig. 13– PAUSE: Messages and Sequence

2.6.4 SET_PARAMETER

This message **SHALL** be used to set the value of a parameter (call record data) for a presentation or stream specified by the URI (request and response):

```
SET_PARAMETER
rtsp://recorder:554/iprecorder/ RTSP/1.0
CSeq: 3
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
Content-Type: application/x-crd+xml
```
(see section 2.7. for content details)

```
RTSP/1.0 200 OK
CSeq: 3
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
```

```
       T/R            Recorder
        | SET_PARAMETER    |
        | -------------->  |
        |    200 OK        |
        | <--------------  |
```

Fig. 14– SET_PARAMETER: Messages and Sequence

2.6.5 TEARDOWN

This message **SHALL** be used to free resources associated with the stream specified by the URI (request and response):

```
TEARDOWN rtsp://recorder:554/iprecorder/ RTSP/1.0
CSeq: 4
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
```

```
RTSP/1.0 200 OK
CSeq: 4
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
```

```
       T/R            Recorder
        |    TEARDOWN      |
        | -------------->  |
        |     200 OK       |
        | <--------------  |
```

Fig. 15– TEARDOWN: Messages and Sequence

2.6.6 REPLAY (optional)

This message sequence **MAY** be used to replay stored information at a replay client. Note: This is seen as optional feature.

```
DESCRIBE rtsp://recorder:554/replay/?connref=403C232A-C510-45C7-973E-D55F5CF996AF RTSP/1.0
CSeq: 2
Accept: application/sdp
```

```
RTSP/1.0 200 OK
CSeq: 2
Server: Example Recorder
Content-Type: application/sdp
Content-Length: 157

v=0
o=unnamed 0 0 IN IP4 playback.example.net
s=Example Stream
t=0 0
a=range:npt=0.0-9.420000000
a=length:npt=9.420000000
m=audio 0 RTP/AVP 8
a=rtpmap:8 PCMA/8000
```

```
SETUP rtsp://recorder:554/replay/?connref=403C232A-C510-45C7-973E-D55F5CF996AF RTSP/1.0
CSeq: 3
Transport: RTP/AVP/TCP;unicast;mode="PLAY";dest_addr=":9";setup=active;connection=new
```

```
RTSP/1.0 200 OK
CSeq: 3
Session: 2da07059-961e-4998-81f8-0f6345e0b15f
Server: Example Recorder
Transport: RTP/AVP/TCP;unicast;dest_addr=":9"/":9";src_addr="192.0.2.5:9000";setup=passive
    connection=new;ssrc=93CB001E
```

```
PLAY rtsp://recorder:554/replay/?connref=403C232A-C510-45C7-973E-D55F5CF996AF RTSP/1.0
CSeq: 4
Session: 2da07059-961e-4998-81f8-0f6345e0b15f
Range: npt=0-9.419000
```

```
RTSP/1.0 200 Success
CSeq: 4
Server: Example Recorder
Session: 2da07059-961e-4998-81f8-0f6345e0b15f
Range: npt=0-9.419
RTP-Info:url=rtsp://recorder:554/replay/?connref=403C232A-C510-45C7-973E-
D55F5CF996AF;rtptime=3188274789;seq=4082
```

```
PAUSE rtsp://recorder:554/replay/?connref=403C232A-C510-45C7-973E-D55F5CF996AF RTSP/1.0
CSeq: 5
Session: 2da07059-961e-4998-81f8-0f6345e0b15f
```

```
RTSP/1.0 200 Success
CSeq: 5
Session: 2da07059-961e-4998-81f8-0f6345e0b15f
Server: Example Recorder
```

```
TEARDOWN rtsp://recorder:554/replay/?connref=403C232A-C510-45C7-973E-D55F5CF996AF RTSP/1.0
CSeq: 6
Session: 2da07059-961e-4998-81f8-0f6345e0b15f
```

```
RTSP/1.0 200 OK
CSeq: 4
Session: c408358f-a233-4dd2-9fb6-a338953cc8b2
```

```
   Terminal         Recorder
        |   DESCRIBE     |
        | -------------> |
        |    200 OK      |
        | <------------- |
        |    SETUP       |
        | -------------> |
        |    200 OK      |
        | <------------- |
        |    PLAY        |
        | -------------> |
        |  200 Success   |
        | <------------- |
        |    Media       |
        | <============= |
        |    PAUSE       |
        | -------------> |
        |  200 Success   |
        | <------------- |
        |   TEARDOWN     |
        | -------------> |
        |    200 OK      |
        | <------------- |
```

Fig. 16– RTSP Replay Session: Messages and Sequence

2.7 CALL RECORD DATA FORMAT

The following XML structure **SHALL** be used to transmit call record data within a SET_PARAMETER message:

```
<call-record-data connref="403C232A-C510-45C7-973E-D55F5CF996AF">
        <properties>
                <property name="">value</property>
        </properties>
        <operations>
                <operation name="" time="[utc-datetime]">value</operation>
        </operations>
</call-record-data>
```

Call Record data **SHALL** be composed of properties and operations. Any timestamp **SHOULD** be set by the client since it has the exact time reference for any local event. If a timestamp value is omitted, the server **SHALL** use the timestamp of arrival of the message.

2.7.1 PROPERTIES

Properties are single values that will not change during the lifetime of a connection and usually do not require a time reference, except for properties that are representing timestamp information.
A client **MAY** send an updated value of a property that he already set if the new value is a more accurate one. In that case the recorder **MAY** overwrite the previous value if present. The recorder does not need to hold any previous values since properties are only those values that can have only one instance for a connection. For instance, the direction of the connection never changes during its lifetime. Examples:

- Direction (of the connection): 0 = unknown (default), 1 = incoming, 2 = outgoing
  <property name="Direction">1</property>
- Priority: 0 = highest ... 4 = lowest (normal, default)
  <property name="Priority">3</property>
- CallingNr, CalledNr, AlertingNr, ConnectedNr: preferred in "tel:" URI format
  <property name="CallingNr">tel:+4311503</property>
- SetupTime, AlertTime, ConnectTime, DisconnectTime, ReleaseTime: utc-datetime
  <property name="SetupTime">20070801T054030.123Z</property>
- DisconnectCause: Cause values according to ITU-T Rec. Q.931
  <property name="DisconnectCause">19</property>
- DisconnectSource: 0 = unknown (default), 1 = endpoint, 2 = other
  <property name="DisconnectSource">1</property>
- Type: Classification of transported data. Values according to BasicService enumeration of ECMA 242 (default = 1, speech)
  <property name="Type">1</property>

2.7.2 OPERATIONS

Operations are events during the lifetime of a connection that may happen at any time and SHOULD be preserved at the recorder. Examples:

- RedirectedNr: Representing a "tel:" URI format to notify a redirection with the new target.
  <operation name="RedirectedNr" time="20070801T054035Z456">+431156</operation>
- CallRef, ThreadRef (including e.g. a UUID): Values are typically changing during call transfers.
  <operation name="CallRef" time="20070801T054059.001Z">FD306648-4EBA-48D5-B41E-00002EA20B76</operation>
  <operation name="ThreadRef" time="20070801T054059Z001">ACB734C8-2843-4FE4-AFBD-00058FA9BD0F</operation>
- PTT-State: Change of PTT state; 0 = off, 1 = on
  <operation name="PTT-State" time="20070801T055000.789Z">0</operation>

2.8 REFERENCING CALL SCENARIOS

Please note that this section is seen as recommendation for referencing call scenarios and not as mandatory requirement. Generally, a call establishing endpoint has to tell its partner a reference with which both can assign their recordings. With this reference, later statistical evaluations about the call scenario can be done. If recorded connections are not referenced, just limited evaluation is possible.

The recorder **SHOULD** know three reference values:

- ConnRef: Identifying a connection that describes the details from the viewpoint of an endpoint.
- CallRef: Identifying a call that has one or typically two connections assigned.
- ThreadRef: Identifying a thread (a call scenario in general) that has one or more calls assigned.

For instance, endpoint A starts a new call scenario by creating an outgoing connection. In this case, it also creates new call and thread references which will be sent along the setup messages. Endpoint B receives an incoming request, creates an incoming connection and associates it with the call and thread references that were sent along with the setup.

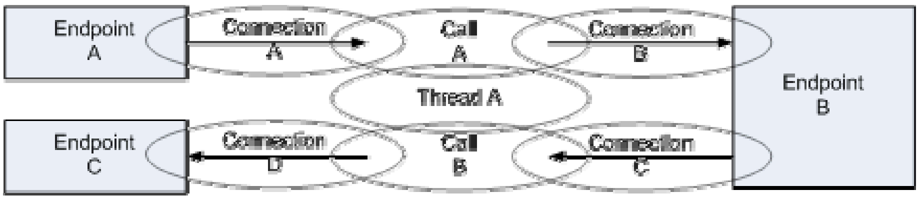

Fig. 17– Call Scenario

If such references are missing, B creates new ones. After some time, B wants to make a call to C. B puts A on hold and creates an outgoing connection together with a new call reference, but assigns the existing thread reference. Endpoint C receives an incoming request and behaves like B before. If now B wanted to transfer A to C it would, as the initiator, create a new call reference and send it along with the transfer notification message. B then would release its connections. Otherwise A and C assign their connections to the newly created call reference but would still remain under the same thread reference. This way all operations are referenced via the thread. Such reference values, defining a call or thread, MAY be transported to the other endpoint using a SIP method (like INFO).

CHAPTER 3

PHONE

3.1 AUDIO SOURCE AND CODING

The User Terminal **SHALL** provide a summarized audio signal (IN & OUT) as a single coded PCM (G.711a) stream that is sent to the Recorder using RTP.

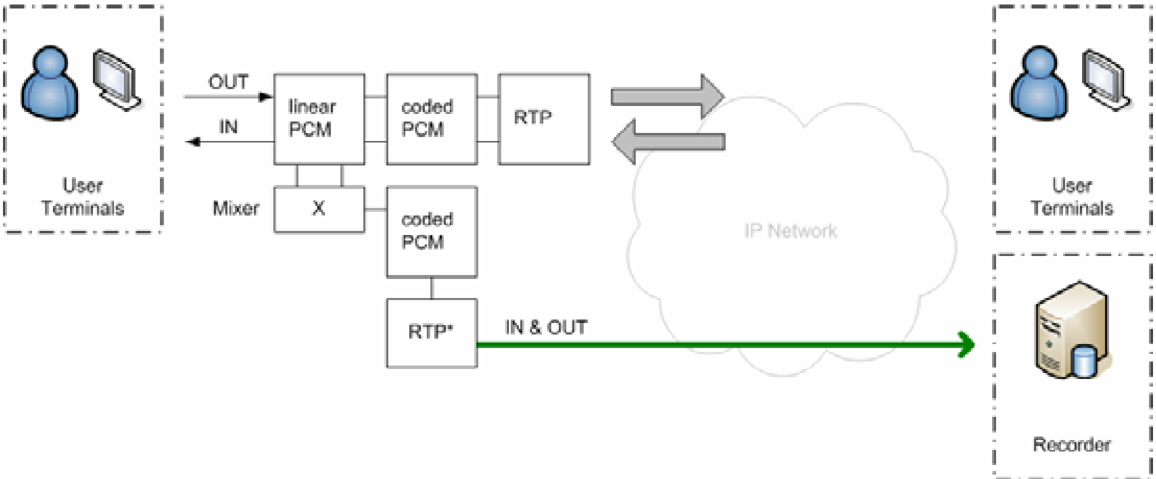

Fig. 18– Audio Source at User Terminal (G/G)

3.2 CALL RECORD DATA

User Terminals (T) **SHALL** transmit the following properties to the Recorder using SET_PARAMETER.

Table 1– List of Phone Properties

Property	Format	Description/Example	Source	Requirement
Direction	INTEGER	0...unknown, 1...incoming, 2...outgoing	T	mandatory
Priority	INTEGER	1...highest ... 5...lowest	T	mandatory
CallingNr	TEL URI	`tel:+4311503`	T	mandatory
CalledNr	TEL URI	`tel:+4311503`	T	optional
AlertingNr	TEL URI	`tel:+4311503`	T	optional
ConnectedNr	TEL URI	`tel:+4311503`	T	optional
SetupTime	UTC DATETIME	`20070801T054030.123Z`	T	mandatory
AlertTime	UTC DATETIME	`20070801T054030.123Z`	T	optional
ConnectTime	UTC DATETIME	`20070801T054030.123Z`	T	optional
DisconnectTime	UTC DATETIME	`20070801T054030.123Z`	T	optional
ReleaseTime	UTC DATETIME	`20070801T054030.123Z`	T	optional
DisconnectCause	INTEGER	ITU-T Rec. Q.931	T	optional
DisconnectSource	INTEGER	1...endpoint, 2...other	T	optional
Type	INTEGER	1...speech	T	optional

User Terminals (T) **SHALL** transmit the following operations to the Recorder using SET_PARAMETER. Note: Operations include per definition a UTC date-time reference as unique timestamp.

Table 2– List of Phone Operations

Property	Format	Description/Example	Source	Requirement
RedirectedNr	TEL URI	`tel:+4311503`	T	mandatory
CallRef	UUID	`<uuid>`	T	optional
ThreadRef	UUID	`<uuid>`	T	optional

CHAPTER 4

RADIO

4.1 AUDIO SOURCE AND CODING

The Radio (or a gateway to the Radio) **SHALL** provide a single audio signal (RX) as a single coded PCM (G.711a) stream that is sent to the Recorder using RTP without header extension (HE).

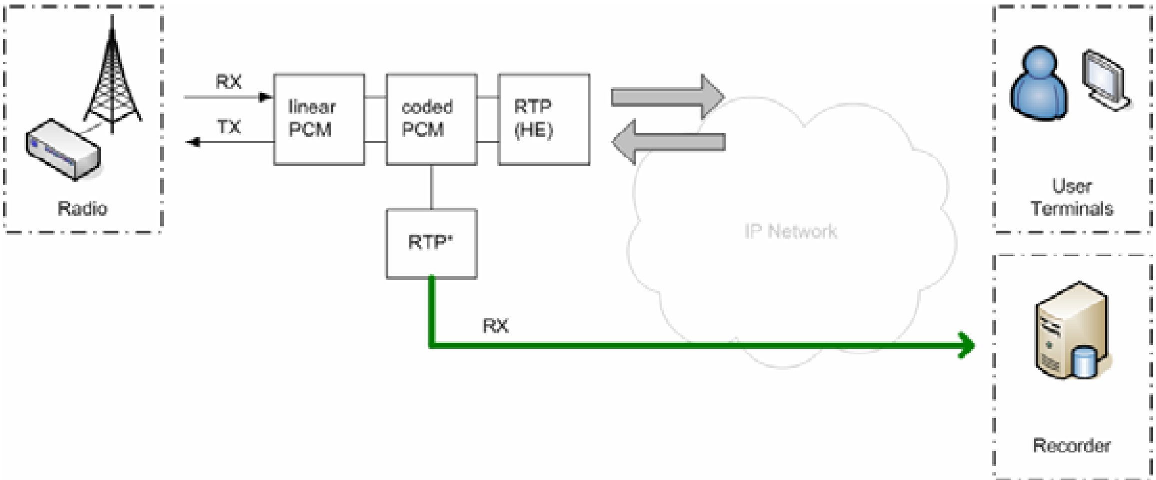

Fig. 19– Audio Source at Radio (A/G)

The User Terminal **SHALL** provide a summarized audio signal (RX & TX) as a single coded PCM (G.711a) stream that is sent to the Recorder using RTP without header extension (HE).

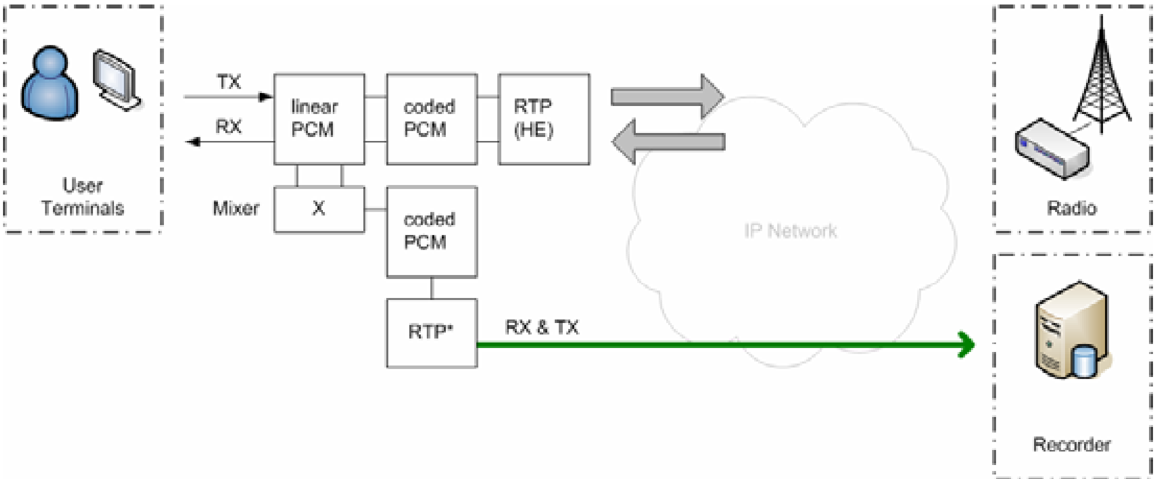

Fig. 20– Audio Source at User Terminal (A/G)

4.2 CALL RECORD DATA

User Terminals (T) **SHALL** and Radios (R) **MAY** transmit the following properties to the Recorder using SET_PARAMETER.

Table 3– List of Radio Properties

Property	Format	Description/Example	Source	Requirement
FrequencyID	STRING	118.005	T, R	mandatory
BSS Quality Index	INTEGER	-100...-70 (RSSI)	R	optional
BSS Method	INTEGER	0...7	R	optional

User Terminals (T) **SHALL** and Radios (R) **MAY** transmit the following operations to the Recorder using SET_PARAMETER. Note: Operations include per definition a UTC date-time reference as unique timestamp.

Table 4– List of Radio Operations

Operation	Format	Description/Example	Source	Requirement
PTT	INTEGER	1...on 2...off	T	mandatory
SQU	INTEGER	1...on 2...off	R	optional
Simultaneous Transmission	INTEGER	0-MAX_NB_TRANS	R	optional

ANNEX A

REFERENCES

[1] IETF RFC 768: “User Datagram Protocol”, August 1980.

[2] IETF RFC 791: “Internet Protocol”, September 1981.

[3] IETF RFC 793: “Transmission Control Protocol”, September 1981.

[4] IETF RFC 2119: "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, March 1997.

[5] IETF RFC 2326: “Real Time Streaming Protocol (RTSP)”, April 1998.

[6] IETF RFC 2460: “Internet Protocol, Version 6 (IPv6) Specification”, December 1998.

[7] IETF RFC 3261: “SIP: Session Initiation Protocol”, June 2002.

[8] IETF RFC 3264: “An Offer/Answer Model with the Session Description Protocol (SDP)”, June 2002.

[9] IETF RFC 3265: “Session Initiation Protocol (SIP)-Specific Event Notification”, June 2002.

[10] IETF RFC 3311: “The Session Initiation Protocol (SIP) UPDATE Method”, September 2002.

[11] IETF RFC 3325: “Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks”, November 2002.

[12] IETF RFC 3398: “Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping”, December 2002.

[13] IETF RFC 3428: “Session Initiation Protocol (SIP) Extension for Instant Messaging”, December 2002.

[14] IETF RFC 3515: “The Session Initiation Protocol (SIP) Refer Method”, April 2003.

[15] IETF RFC 3550: “RTP: A Transport Protocol for Real-Time Applications”, July 2003.

[16] IETF RFC 3665: Session Initiation Protocol (SIP) Basic Call Flow Examples, December 2003.

[17] IETF RFC 3666: “Session Initiation Protocol (SIP). Public Switched Telephone Network (PSTN) Call Flows”, December 2003.

[18] IETF RFC 3840: “Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)”, August 2004.

[19] IETF RFC 3891: “The Session Initiation Protocol (SIP) ‘Replaces’ Header”, September 2004.

[20] IETF RFC 3911: “The Session Initiation Protocol (SIP) ‘Join’ Header”, October 2004.

[21] IETF RFC 4145: “TCP-Based Media Transport in the Session Description Protocol (SDP)”, September 2005.

[22] IETF RFC 4235: “An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)”, November 2005.

[23] IETF RFC 4346: “The Transport Layer Security (TLS) Protocol - Version 1.1”, April 2006.

[24] IETF RFC 4566: “SDP: Session Description Protocol”, July 2006.

[25] IETF RFC 4571: “Framing Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP) Packets over Connection-Oriented Transport”, July 2006.

[26] IETF RFC 4579: “Session Initiation Protocol (SIP) Call Control – Conferencing for User Agents”, August 2006.

[27] IETF SIPPING Working Group; draft-ietf-sipping-service-examples-13: “Session Initiation Protocol Service Examples”, July 2007.

[28] Eurocontrol. “Voice Communication System (VCS) Procurement Guidelines”, Ed. 2.0, February 2005.

[29] Eurocontrol: “Call Priority, Instantaneous Access and Intrusion Telephone Facilities: Operational Descriptions”, July 2006.

[30] Eurocontrol. “ATS R2 and ATS No.5 Signalling Protocol Specifications”, Ed. 1.0 (February 2005).

[31] Eurocontrol. “Voice Communication System (VCS) Procurement Guidelines”, Ed. 1.0, March 2003.

[32] Eurocontrol. “ATS Ground Voice Network Implementation and Planning Guidelines”, Ed. 1.0, February 2005.

[33] ITU-T. Rec. G.711. “Pulse Code Modulation (PCM) of Voice Frequencies”, November 1988.

[34] ITU-T. Rec. G.728. “Coding of Speech at 16 Kbit/s Using Low-delay Code Excited Linear Prediction”, September 1992.

[35] ECMA-143: “Private Integrated Services Network (PISN); Circuit Mode Bearer Services; Inter-Exchange Signalling Procedures and Protocol (ISO/IEC 11572)”.

[36] ECMA-312: “Private Integrated Services Network (PISN); Profile Standard for the Use of PSS1 (QSIG) in Air Traffic Services Networks”.

[37] ECMA-339: “Corporate Telecommunication Networks; Signalling Interworking between QSIG and SIP; Basic Services”.

ANNEX B

ACRONYMS

Ack	Acknowledge
AGVN	Air Traffic Services Ground Voice communications Network
A/G	Air/Ground
AM	Amplitude Modulation
ANSP	Air Navigation Service Provider
ATA	Analogue Telephone Adapter
ATC	Air Traffic Control
ATM	Air Traffic Management
ATS	Air Traffic Services
ATS-No.5	Air Traffic Services – No.5 signalling system
ATS-QSIG	Air Traffic Services – Q reference point SIGnalling system
ATS-R2	Air Traffic Services – R2 signalling system
AVP	Audio/Video Profile
CICL	Call Intrusion Capability Level
CIPL	Call Intrusion Protection Level
CPICL	Call Priority Interruption Capability Level
CPIPL	Call Priority Interruption Protection Level
CWP	Controller Working Position
DA	Direct Access
DNS	Domain Name Service
ECMA	European Computer Manufacturers Association
G/G	Ground/Ground
HMI	Human Machine Interface
HTTP	HyperText Transfer Protocol
IA	Instantaneous Access
ICCVC	Instantaneous Controller-Controller Voice Communication
IDA	InDirect Access
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ITU-T	International Telecommunication Union – Telecommunication standardization sector
LAN	Local Area Network
LD-CELP	Low Delay - Code Excited Linear Prediction
MF	Multi-Frequency
MFC	Multi-Frequency Code
MSC	Message Sequence Chart
PABX	Private Automatic Branch eXchange
PCM	Pulse Code Modulation
PINX	Private Integrated services Network eXchange
PISN	Private Integrated Services Network
PSS1	Private Signalling System no. 1
PSTN	Public Switched Telephone Network
QoS	Quality of Service
Rec.	Recommendation
RFC	Request For Comments
RTCP	Real-time Control Protocol
RTP	Real-time Transport Protocol
Rx	Reception
S/MIME	Secure / Multipurpose Internet Mail Extensions
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SS-IA	Instantaneous Access Supplementary Service
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TLS	Transport Layer Secure protocol

TU	Transaction User
Tx	Transmission
UA	User Agent
UAC	User Agent Client
UAS	User Agent Server
UDP	User Datagram Protocol
URI	Universal Resource Identifier
UHF	Ultra-High Frequency
VCS	Voice Communications System
VHF	Very High Frequency
VoIP	Voice over the Internet Protocol
WAN	Wide Area Network

ANNEX C

LIST OF EUROCAE WG-67 CONTRIBUTORS

SURNAME	NAME	COMPANY
ADRIAN	Andre	DFS
GELADA	Mario	ATIS
HAINDL	Bernhard	FREQUENTIS
KAMPICHLER	Wolfgang	FREQUENTIS
MARTÍN	Miguel A.	AENA
PALMER	John S.	JSP-TELECONSULTANCY
SMITH	Barry	FAA
STANDEREN	Egil	THALES-NO
WEGER	Roberto	SITTI
ZOMIGNANI	Marcelo	INDRA SISTEMAS