NIST/OLES VoIP Roundtable
Request For Comments: *nnnn*
Category: Informational
Version 1.0

R. Mitchell
Twisted Pair Solutions
C. Eckel
Cisco

April 2008

Implementation Profile for Interoperable Bridging Systems Interfaces (Phase 1)

Status of this Memo

This document specifies an implementation profile for the public safety community and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

Abstract

This document describes an implementation profile for interoperability among bridging systems. A bridging system is a device that enables disparate radio frequencies or technologies to communicate with each other. Other communications devices such as analog phones, mobile phones, IP telephones and personal computers may be included in a bridging system; however, this is not an exhaustive list of possible devices that connect to a bridging system. The interface through which bridging systems communicate with other bridging systems is the Bridging Systems Interface (BSI).

Session Initiation Protocol (SIP) is used as the basis for the implementation profile for the BSI. SIP is an industry accepted IP-based control protocol for creating, modifying and terminating sessions with one or more participants. These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences.

1 Introduction

The pervasiveness of radio systems in the market today provides for wide-ranging, broad-based communications between radio users or between radio users and their dispatchers. These radios come in many different technologies including analog, digital, trunked, P.25, TETRA, etc., and different frequencies such as UHF, VHF, HF, 700 MHz, 800 MHz, etc.

For the remainder of this document, the terms "radio" and "radio systems" shall refer to the collective group of disparate radio technologies and frequencies in use today in the community.

Radio systems have long been the backbone of communications for public safety personnel in the field responding to emergencies. They are also widely used in other markets such as defense, transportation and utilities. Radios are used by so many industries for a wide variety of reasons that advances in radio technology have been able to offer a multitude of services to subscribers. In addition to audio, some radio systems offer data, DTMF and other services. Because agencies have different requirements and different budget cycles, each agency may have different radio systems. Even within agencies there can be a variety of radio systems deployed. With different radio systems deployed, agencies must still be able to communicate among themselves as well as on an intra- and inter-region level, all the way up to the federal/state government.

Interoperability between radio systems, regardless of the agency level, is crucial to emergency responders. As alluded to previously, interoperability does not exist just on a radio system level but also at an agency level. Agencies need the ability to communicate to other agencies, regardless of level, when authorized to do so; however, incompatibilities among different radio systems often make this difficult. In order to give multiple radio systems the ability to communicate with each other, Bridging Systems Interfaces (BSIs) were developed by many vendors to enable interoperability.

Differences in technology and frequency are not the only limiting factors that prevent radio users from communicating with each other. The lack of operational policies often interferes with communications during an incident. For the purpose of this document, operational policies are out of scope.

Many different radio gateways exist in the market today. Some of these radio gateways support Radio over IP (RoIP) which, at a very high level, is the ability to pass audio and other control functions of a radio system across an IP network.

SIP, anchored by RFC 3261 [1] and extended by many other RFCs and drafts, provides a well defined infrastructure for establishing communication sessions. The goal of this document is to specify an implementation profile that narrows the field to that which is required to establish audio communication channels between BSIs within the public safety space.

2 Overview of Bridging Systems Interfaces

A Bridging Systems Interface (BSI) is a hardware or software platform that enables radio system or radio gateway interoperability. To see where BSIs fit into an overall system architecture, refer to the following architecture:

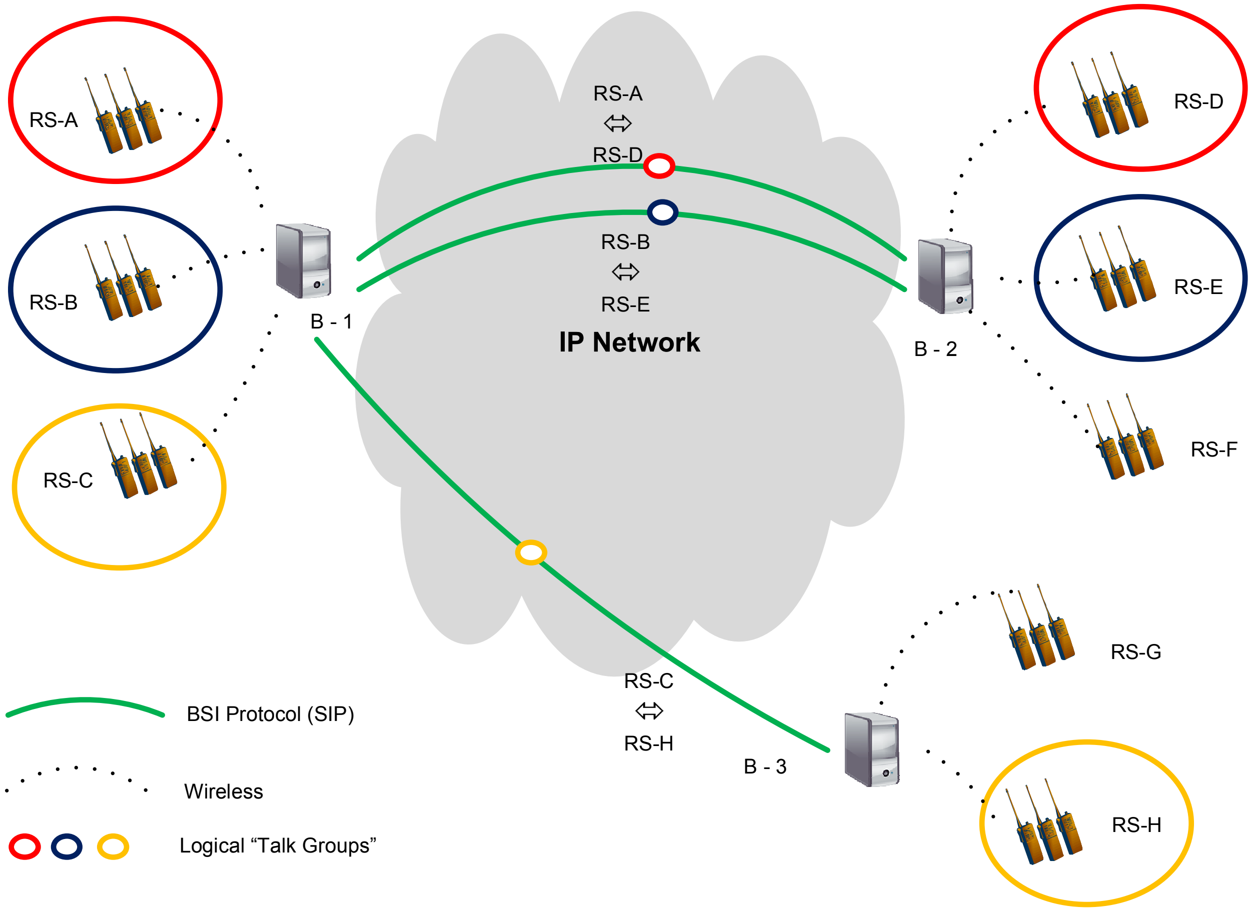
Radio System ←→ BSI ←→ **BSI Protocol** ←→ BSI ←→ Radio System

Note that this architecture is not indicative of every scenario for a BSI. Also, it is possible that the BSI and radio gateway are the same physical device.

A device that enables interoperability with or between radios or other devices (e.g. phones and computers) can be considered a BSI. Such a BSI is stand-alone in nature and functions on its own.

Disparate radio gateways that enable Radio over IP connections that need to be interoperable with each other need to communicate with each other using a BSI.

BSIs are similar to radio devices in that market demands, timing and budget cycles can have a large effect on which BSI is used by a particular group of people within an agency or between agencies. When you have multiple agencies or multiple groups within an agency using different BSIs, a method must exist for these BSIs to interoperate.



Sample Interconnect Scenario

The implementation profile described within this document serves as an initial implementation profile for the BSI protocol. SIP serves as the basis of this profile, but SIP is not a vertically integrated communications system. Rather, SIP is a protocol that can be used with other IETF protocols to build a complete multimedia architecture. These include Real-time Transport Protocol (RTP) (RFC 3550 [3]) for transporting real-time data and providing QoS feedback, and the Session Description Protocol (SDP) (RFC 4566 [4]) for describing multimedia sessions.

SIP does not provide services. Rather, SIP provides primitives that can be used to implement different services. For example, SIP can be used to identify an audio channel accessible via a BSI and express the intent of another BSI to join that audio channel. If this primitive is used to deliver a session description written in SDP, for instance, the BSIs can agree on the parameters of a session that exchanges audio streams between them.

SIP does not offer conference control services such as floor control or priority, nor does it describe how a conference is to be managed. SIP can be used to initiate a session that uses some other conference control protocol.

SIP provides a suite of security services, which include denial-of-service prevention, authentication (both user to user and proxy to user), integrity protection, and encryption and privacy services.

SIP works with both IPv4 and IPv6.

3 Scope

The implementation profile defined in this document is intended to define the minimum standards, parameters and values required to enable basic voice interoperability across BSIs as described in section 2 of this document. As such, it provides value to both manufacturers (for development purposes) and purchasers (for specification and conformance purposes) of devices with a BSI. In developing this implementation profile, the goals were:

- Make use of existing standards.
- Avoid any proprietary extensions to these standards.
- Define a minimal set of required functionality that is both broad enough to meet the immediate needs of the public safety community and narrow enough to facilitate rapid rollout of interoperable implementations of said functionality by manufacturers.
- Clearly define the semantics for how this functional subset is to be used between BSIs, including naming conventions adhered to across compliant BSIs, and identification of a priori knowledge required for proper configuration.

Implementations are free to use mechanisms not defined within this profile; however, they MUST NOT require or assume support for any mechanisms not explicitly listed as REQUIRED within this profile.

It is assumed that subsequent revisions and extensions of this initial implementation profile will provide additional and advanced functionality in a phased approach. The definition of future phases, including the timelines and functionality of each, are outside of the scope of this document. However, it is the intent of this initial phase to serve as a foundation to be extended by future phases, not a throw away prototype for demonstration purposes. For example, though floor control is not addressed in this first phase, session and media negotiation mechanisms put in place in this phase provide mechanisms that may be extended in future phases to address floor control. Other requirements slated to be addressed in future versions include exchange of call metadata (e.g., call priority, confirmed vs. unconfirmed calls) and arbitration of resources (e.g., push-to-talk management information).

When discussing BSIs, this document refers to audio only. Other services offered by the BSI or radio gateway are considered ancillary and out of scope.

4 Terminology

In this document, the key words "MUST, "MUST NOT, "REQUIRED", "SHALL, "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [2] and indicate requirement levels for compliant implementation.

5 Signaling Layer

The signaling layer deals with the protocol that will be used for call establishment, in-call modification (such as changing session parameters) and call release.

Session Initiation Protocol (SIP), as outlined in RFC 3261 [1], has been used in many applications and is the protocol used for BSI interoperability. SIP is a versatile protocol with several applications including Voice over IP (VoIP) telephony calls.

SIP is used to allow audio traffic to flow between different BSI systems.

5.1 Structure of the Protocol

SIP messaging, as defined in RFC 3261 [1], defines the structure of the protocol. This document does not attempt to explain SIP in detail. Further, the exact syntax of messages (e.g. call setup, in-call handling and call tear down, etc.), message processing (e.g. errors, unrecognized responses, etc.), and timer handling should be referenced in [1] and its related RFCs and drafts. This document focuses on the specification of the minimum set of functionality REQUIRED to comply with the implementation profile for BSI interworking and offers suggestions for RECOMMENDED functionality.

5.1.1 Requests

SIP is based on an HTTP-like request/response transaction model. Each transaction consists of a request that invokes a particular method, or function, on the server, resulting in zero or more provisional responses and a final response.

SIP requests are distinguished by having a Request-Line for a start-line. A Request-Line contains a method name, a Request-URI, and the protocol version separated by a single space (SP) character.

SIP is characterized by an ever increasing number of method names, with a base set defined in [1] and additional methods defined in other RFCs. The implementation profile limits the REQUIRED methods to INVITE, ACK, CANCEL, BYE, and OPTIONS. INVITE, ACK, and CANCEL are for setting up sessions, BYE is for terminating sessions, and OPTIONS is for querying servers about their capabilities. Sending OPTIONS requests is OPTIONAL, but being able to respond appropriately when receiving an OPTIONS requests is REQUIRED. Implementations are free to support other methods, but they MUST NOT assume support for any other methods.

An Allow header field SHOULD be present in the INVITE and in responses to INVITE. It indicates which methods can be invoked, within a dialog, on the system sending the

INVITE, for the duration of the dialog. For example, a system capable of receiving only the mandatory methods SHOULD include an Allow header field as follows:

Allow: INVITE, ACK, CANCEL, BYE, OPTIONS

The Request-URI is a SIP or SIPS URI as defined in [1]. For this implementation profile, SIP URIs MUST be supported. Support for other URI schemes, including SIPS, is OPTIONAL but MUST NOT be assumed to exist.

SIP-Version is specified in all requests and response. To be compliant with this profile, systems sending SIP messages MUST include a SIP-Version of "SIP/2.0". The SIP-Version string is case-insensitive.

5.1.2 Responses

SIP responses are distinguished from requests by having a Status-Line as their start-line. A Status-Line consists of the protocol version followed by a numeric Status-Code and its associated textual phrase, with each element separated by a single SP character.

The Status-Code is a 3-digit integer result code that indicates the outcome of an attempt to understand and satisfy a request. The Reason-Phrase is intended to give a short textual description of the Status-Code. The Status-Code is intended for use by automata, whereas the Reason-Phrase is intended for a human user. A client is not required to examine or display the Reason-Phrase.

The SIP-Version MUST match the SIP-Version as outlined in Section 5.1.1.

The Status-Code MUST be one of the pre-defined status codes outlined in section 7.2 of RFC 3261 [1].

The Reason-Phrase is a logical, text-based phrase that expands on the status-code. This profile does not outline or dictate the reason phrases that may be used.

5.2 Initiation of a Call Session

Sessions within SIP are initiated using the INVITE method. Section 13 of RFC 3261 [1] discusses, in detail, how the UAC (User Agent Client) and UAS (User Agent Server) formulate and process initial INVITE requests. In terms of SIP, the successful establishment of an INVITE initiated call session is referred to as a dialog. For the purpose of this implementation profile document, a call session is equivalent to a SIP dialog.

It is RECOMMENDED that a BSI, acting as a UAC, include a SIP URI identifying the address of record (AoR) of the calling resource in the From header of the INVITE (e.g. sip:chn1@bsi1.example.com). Doing so facilitates the use of the From header by another BSI, acting as a UAS, to help determine whether or not to accept the request.

5.2.1 Rules of Engagement

Certain rules of engagement MUST be followed to facilitate call sessions between BSIs. In some situations, well known and previously configured BSI resources may be desired by an end-user customer. In other situations, on-the-fly, ad-hoc resources may be desired. Both of these scenarios MUST BE supported by implementations compliant with this profile. Consequently, all implementations MUST support the following:

- Pre-configuration to enable call sessions for a specified resource available at a pre-configured BSI.
- Ad-hoc configuration to enable call sessions for a specified resource available at a pre-configured or an ad-hoc BSI.
- Pre-configuration to accept call session requests for a specified resource from a pre-configured BSI.
- Ad-hoc configuration to accept call sessions requests from a preconfigured or an ad-hoc BSI.

The distinction between pre-configured and ad-hoc is that pre-configuration is done ahead of time to accept inbound calls from or initiate outbound calls to specified BSIs at any time. The resulting call sessions may be long duration sessions agreed to by the participating agencies ahead of time or activated only when necessary. Ad-hoc configuration facilitates sharing of resources that were not anticipated ahead of time, requiring configuration on-the-fly by both the origination and destination BSIs. These resulting sessions are potentially one time and/or short duration call sessions resulting from real time agreement by the participating agencies, but they may also result in long duration sessions.

In both cases, establishment of a call session between two BSIs is possible only after the corresponding agencies have enabled such sessions via pre-configuration or ad-hoc configuration. While phase 1 of the BSI implementation profile provides for both pre-configured and ad-hoc modes of operation, both cases are treated essentially the same. Agencies wishing to inter-connect their systems through BSIs complying with phase 1 of this profile need to exchange and configure the same amount of information, as described in detail in section 9, for either mode of operation. It is anticipated that future phases of this profile will define mechanisms that simplify ad-hoc operations for the benefits of the public safety community.

The naming conventions for inter-agency resources are discussed in section 7.4. The security aspects of the call sessions are discussed in section 8. The call sessions are understood to be half-duplex push-to-talk in nature. The detection of audio is described in section 10.

5.3 In-Call Session Control

SIP includes the ability to modify a session that is already established between two SIP endpoints. This modification can involve changing addresses or ports, adding a media stream, deleting a media stream, and so on. This is accomplished by sending a new

INVITE request within the same dialog that established the session. An INVITE request sent within an existing dialog is known as a re-INVITE.

A special case of a re-INVITE is the sending of a re-INVITE to confirm that the call session is still active at the signaling level. This may be done as part of recovering from a temporary lapse in connectivity or the detection of a loss of media. In this case, the re-INVITE does not actually modify the call session. This is discussed in more detail in section 10.1.

5.3.1 Re-INVITEs to Modify SIP Sessions

The use of re-INVITEs to modify a call session is not included within the implementation profile, and support for modifying call sessions MUST NOT be assumed by any implementation compliant with this profile.

5.3.2 Separate SIP Sessions Required

Each SIP session MUST have one device or audio participant on each end of the SIP call. Separate SIP sessions MUST BE established for each media stream between BSIs. If the BSI is able to “patch” multiple audio streams together into a single mixed stream, that single stream may then become part of the SIP session, but such functionality is implementation specific and outside the scope of this profile.

5.4 Call Session Release

All call sessions MUST be gracefully released by sending a BYE, except in situations in which doing so is not possible due to hardware/software failures, loss of connectivity, etc. BYE instructs the User Agent (UA) on the far side that the party wishes to disconnect. At that point, both UAs MUST stop listening for and sending media.

Call sessions may end non-gracefully due to hardware/software failure, loss of connectivity, loss of media, etc. Implementations of the profile MUST be able to detect and handle such cases. In such cases, a BYE may not be received from the other side despite the fact it is no longer able to participate in the session. Detection and recovery from such cases is addressed in section 10.1.

5.5 Example Call Flow

This section illustrates session establishment between two BSIs, BSI1 (bsi1.example.com) and BSI2 (bsi2.example.com). BSI1 and BSI2 are assumed to be BSIs compliant with phase 1 of the implementation profile. The successful negotiation between the agencies responsible for BSI1 and BSI2 is assumed to have resulted in the configuration of 5000 (sip:5000@bsi2.example.com) as an identifier for a channel existing on BSI2 that BSI1 should be able to access. On BSI1, chn1 (sip:chn1@bsi1.example.com) originates the connection; however, the identifier for chn1 need not necessarily be configured on BSI2 for this scenario. Exactly how BSI2 chooses to determine whether or not to allow requests from BSI1 is not in scope for this implementation profile. What is REQUIRED is for BSI2 to provide some means of configuring that such requests are to be allowed. One option is to use the SIP URI in the From header, as described in 5.2, in this determination.

The following example call flow shows the initial signaling, the exchange of media information in the form of SDP payloads, the establishment of the media session, then finally the termination of the call.

```
BSI1                     BSI2
 |                         |
 |       INVITE F1         |
 |------------------------>|
 |       200 OK F2         |
 |<------------------------|
 |         ACK F3          |
 |------------------------>|
 |   Both Way RTP Media    |
 |<=======================>|
 |                         |
 |         BYE F4          |
 |<------------------------|
 |       200 OK F5         |
 |------------------------>|
 |                         |
```

In this scenario, BSI1 completes a call to BSI2 directly. The use of DNS resolvable hostnames (i.e. bsi1.example.com and bsi1.example.com) is for illustrative purposes only. Support for DNS is OPTIONAL; therefore, implementations MUST NOT assume support for DNS when constructing SIP messages. Implementations MUST be able to restrict themselves to using IP addresses in the SIP headers they add that effect the routing of SIP messages. These headers include any Via, Contact, Record-Route, or Route headers that they add.

F1 INVITE BSI1 -> BSI2

```
INVITE sip:5000@bsi2.example.com SIP/2.0
Via: SIP/2.0/TCP 192.11.11.111:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: <sip:chn1@bsi1.example.com>;tag=9fxced76sl
To: <sip:5000@bsi2.example.com>
Call-ID: 3848276298220188511@bsi1.example.com
CSeq: 1 INVITE
Contact: <sip:chn1@192.11.11.111;transport=tcp>
Content-Type: application/sdp
Content-Length: 189

v=0
o=chn1 2890844526 2890844526 IN IP4 192.11.11.111
s=-
c=IN IP4 192.11.11.111
t=0 0
```

```
m=audio 49172 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
```

F2 200 OK BSI2 -> BSI1

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP 192.11.11.111:5060;branch=z9hG4bK74bf9
 ;received=192.11.11.111
From: <sip:chn1@bsi1.example.com>;tag=9fxced76sl
To: <sip:5000@bsi2.example.com>;tag=8321234356
Call-ID: 3848276298220188511@bsi1.example.com
CSeq: 1 INVITE
Contact: <sip:5000@192.22.22.222;transport=tcp>
Content-Type: application/sdp
Content-Length: 185

v=0
o=5000 2890844527 2890844527 IN IP4 192.22.22.222
s=-
c=IN IP4 192.22.22.222
t=0 0
m=audio 3456 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
```

F3 ACK BSI1 -> BSI2

```
ACK sip:5000@192.22.22.222 SIP/2.0
Via: SIP/2.0/TCP 192.11.11.111:5060;branch=z9hG4bK74bd5
Max-Forwards: 70
From: <sip:chn1@bsi1.example.com>;tag=9fxced76sl
To: <sip:5000@bsi2.example.com>;tag=8321234356
Call-ID: 3848276298220188511@bsi1.example.com
CSeq: 1 ACK
Content-Length: 0
```

```
/* RTP streams are established between BSI1 and BSI2 */

/* BSI2 Hangs Up with BSI1. Note that the CSeq is NOT 2, since
   BSI1 and BSI2 maintain their own independent CSeq counts.
   (The INVITE was request 1 generated by BSI1, and the BYE is
   request 1 generated by BSI2). CSeq need not start at 1, but they
   MUST be incremented by 1 for each new request */
```

F4 BYE BSI2 -> BSI1

```
BYE sip:chn1@192.11.11.111 SIP/2.0
Via: SIP/2.0/TCP 192.22.22.222:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
From: <sip:5000@bsi2.example.com>;tag=8321234356
To: <sip:chn1@bsi1.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@bsi1.example.com
CSeq: 1 BYE
Content-Length: 0
```

F5 200 OK BSI1 -> BSI2

```
SIP/2.0 200 OK
Via: SIP/2.0/TCP 192.22.22.222:5060;branch=z9hG4bKnashds7
 ;received=192.22.22.222
From: <sip:5000@bsi2.example.com>;tag=8321234356
To: <sip:chn1@bsi1.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@bsi1.example.com
CSeq: 1 BYE
Content-Length: 0
```

RFC 3665 [5] provides numerous other example flows that may be of interest to implementers of this profile despite the fact that support for many of the sample flows in RFC 3665 are beyond what is required for this profile.

5.6 Proxy Servers

Proxy Servers route SIP requests from User Agent Clients (UACs) to User Agent Servers (UASs). In general, one or more proxy servers may exist between the UAC and UAS. Support for proxy servers between two BSIs is OPTIONAL for phase 1 of the implementation profile. Implementations compliant with the implementation profile MUST support direct communication with another BSI. Support for routing SIP messages through proxy servers MUST NOT be assumed by implementations compliant with this profile.

5.7 Registrars

SIP Registrars allow User Agents (UAs) to register their location and other information to a centrally located and known server. This makes call setup more dynamic when the network knows where to locate the called party.

A BSI MAY be configured with a SIP registrar’s address and be able to send and receive the appropriate REGISTER messages and responses with the registrar server. Likewise, the BSI may also be configured with the location of other BSIs to which it may establish SIP-based communications. Implementations compliant with the profile MUST support the latter. Interactions with SIP Registrars is OPTIONAL for phase 1 of the profile.

6 Media Layer

The media layer of the implementation profile deals with the media streams exchanged between BSIs. The details of the call session established at the signaling layer, such as the type of media, codec, and sampling rate, are not described using SIP. Rather, the body of a SIP message contains a description of the session, encoded in some other protocol format. One such format is the Session Description Protocol (SDP) (RFC 4566 [4]). This SDP message (shown in the example in section 5.5) is carried by the SIP message in a way that is analogous to a document attachment being carried by an email message, or a web page being carried in an HTTP message. In order to comply with the implementation profile, a BSI MUST support SDP as a means to describe media sessions, and its usage for constructing offers and answers MUST follow the procedures defined in "An Offer/Answer Model with SDP", RFC 3264 [6].

6.1 Offer/Answer Model

RFC 3264 [6] describes the complete offer/answer model. It provides a variety of ways in which media negotiation may occur between two endpoints. For the purpose of the implementation profile, a limited set of mechanisms is specified as REQUIRED for a BSI to be compliant to be compliant with the profile.

The example flow in section 5.5 shows the classic offer/answer exchange in which the offer is included by BSI in the INVITE request and the answer is included by BSI2 in the 200 response. At a minimum, this exchange MUST be supported by a BSI in order to comply with the implementation profile. Other exchanges, such as sending an offerless INVITE or modifying the media with subsequent offer/answer exchanges via re-INVITEs MAY be supported, but an implementation MUST NOT assume support for such exchanges. It is perfectly valid for a BSI in compliance with this implementation profile to reject an offerless INVITE. Likewise, if receiving a re-INVITE with a new offer that attempts to modify the existing media session, it is valid for the BSI to reject the re-INVITE. If the receiving BSI happens to support modification of the existing media session via re-INVITE, it may accept and modify the media session accordingly. In either case, both BSIs MUST act in a way that complies with section 4 of RFC 3264 [6].

The use of re-INVITEs that do not modify the media session for the purpose of recovering from media loss is described in section 10.1.

6.2 Media Streams

A BSI complying with this implementation profile MUST support the use of an offer/answer exchange to negotiate a single audio stream. The media negotiated by the offer/answer exchange is limited by this implementation profile to a single audio stream. Attempts to negotiate multiple audio streams or a non-audio stream (i.e. video) may result in unexpected results. Any BSI complying with this implementation profile MUST be able to handle unexpected results if it tries to negotiate anything beyond a single audio stream. It is RECOMMENDED that a BSI wishing to negotiate more than a single audio stream sets the first stream within the offer to be an audio stream as defined in 6.5.6.

6.3 Voice Encoders

Each BSI MAY support whichever voice encoders are necessary for proper functioning. However, in order to be compliant with this specification, each BSI MUST support, at a minimum, the following codec for SIP sessions: G.711 u-Law as defined in RFC 3551 [12].

6.3.1 Optional Voice Encoders

Other voice encoders are considered OPTIONAL for this implementation profile. In order to minimize the need for transcoding, the following codecs, though not required, are RECOMMENDED: G.711 A-Law, GSM Full Rate, and G.729, as defined in RFC 3551 [12], and IMBE as defined in TIA/EIA/IS-102.BABA [20]. These, and other codecs, MAY be included in the media offered when establishing a call session; however, G.711 u-law MUST be included in the media offered regardless of how many OPTIONAL codecs are offered.

6.3.2 Voice Encoder Fees

Several of the codecs mentioned in this specification are not free of charge. Some come with licensing and royalty fees that may be cost prohibitive to market entrants. While this specification aims to keep the entry cost low, it is not feasible to come up with a list of only free codecs that are to be required for the BSI protocol. For example, G.729 is a voice encoder that provides beneficial tradeoffs: low bandwidth, high quality and widely accepted with (relatively) minor fees for royalties.

6.3.3 Voice Encoder Tandeming

Connecting multiple voice encoders back-to-back is referred to as tandeming. Tandeming more than one low-bit-rate voice encoder (e.g., IMBE <-> GSM) may impact the quality of the voice signal passing through the network/system, so it is recommended that this be avoided whenever possible.

6.4 DTMF

A BSI complying with this implementation profile MUST support "RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals", RFC 4733 [7] for the sending and receiving of DTMF digits. The payload type used is dynamic, meaning it MAY be anything within the range of 96-127. In order to comply with industry convention, it is RECOMMENDED to use 101 as the payload format; however, support for other dynamic payload formats MUST be supported.

A BSI MUST support at least events 0-15 (the DTMF events) in order to comply with this implementation profile. Support for additional events is OPTIONAL.

The actual encoding and decoding of DTMF by the bridging system at its radio interfaces is OPTIONAL. How DTMF is detected by and transmitted to the various radio interfaces of the bridging system is outside of the scope of this implementation profile. This implementation profile states how DTMF digits are to be transmitted and received between BSIs; however, it does not specify or guarantee that the DTMF digits will be faithfully detected by and transmitted to all devices accessible through other interfaces of each bridging system.

6.5 Additional Requirements and Recommendations

When formulating an SDP offer or answer, a BSI complying with this implementation profile MUST comply with RFC 4566 [4] and RFC 3264 [6]. Some requirements are restated here, and additional recommendations are included to limit the effort involved in developing interoperable implementations of the profile.

6.5.1 Protocol Version (“v=” line)

The "v=" line gives the version of the Session Description Protocol. RFC 4566 [4] defines version 0. There is no minor version number. BSIs complying with the implementation profile MUST include the version line with a value of 0.

```
v=0
```

6.5.2 Origin (“o=” line)

The "o=" line gives the originator of the session (his/her username and the address of the user's host) plus a session identifier and version number. The network type MUST be “IN”, the address type MUST be “IP4”, and the IP address or hostname MUST resolve to a unicast address.

```
o=chn1 2890844526 2890844526 IN IP4 192.11.11.111
```

6.5.3 Session Name (“s=” line)

The "s=" line is the textual session name. There MUST be one and only one "s=" line per session description. In accordance with RFC 3264 [6], the session name is RECOMMENDED to be “-“.

```
s=-
```

6.5.4 Connection Data (“c=” line)

The "c=" line contains connection data. A session description MUST contain either at least one "c=" line in each media description or a single "c=" line at the session level. The network type MUST be "IN", the address type MUST be "IP4", and the IP address must be a unicast address. It is REQUIRED that BSIs support one "c=" line at the session level.

```
c=IN IP4 192.22.22.222
```

6.5.5 Timing ("t=" line)

The "t=" line specifies the start and stop time for a session. In accordance with RFC 3264 [6], it is RECOMMENDED that a BSI complying with the implementation profile includes a single "t=" line with a start and stop value of 0.

```
t=0 0
```

6.5.6 Media Description ("m=" line)

Each media description starts with an "m=" line, and is terminated by either the next "m=" line or by the end of the session description. Although an offer/answer MAY include multiple media descriptions, to comply with this implementation profile a BSI need only support one media descriptor and MUST be prepared for additional media descriptors to be rejected as described in RFC 4566 [4].

The only media type REQUIRED by the implementation profile is "audio". The only media transport protocol REQUIRED by the implementation profile is "RTP/AVP".

It is REQUIRED that a BSI uses media type "audio" and protocol "RTP/AVP" in the first media description included in the offer.

It is REQUIRED that the RTP port number specified be an even number, with the implicit assumption being that (port + 1) is used for RTCP.

```
m=audio 49172 RTP/AVP 0 3 18 8 101
```

This above example includes all the REQUIRED and RECOMMENDED payload types using the values refined in RFC 3551 [12]. The order of the payload formats indicates the order of preference, so in this example, G.711 u-law is preferred, followed by GSM Full Rate, G.729, and G.711 A-law. It indicates 101 as the dynamic payload type for DTMF events. Other values with the range 96-127 MAY be used instead.

Other payload formats are OPTIONAL. All that is REQUIRED is that the offer includes the REQUIRED payload format (0) and a dynamic payload format for DTMF. The order of the payload formats MAY be set as preferred by the BSI. For example, the following is perfectly valid media description.

```
m=audio 49172 RTP/AVP 0 100
```

This indicates the BSI wants to use G.711 u-law with 100 as the DTMF payload type. As long as an offer contains the REQUIRED payload format, a BSI MUST be able to respond with an answer accepting that payload format.

6.5.7 Attributes (“a=” line)

Attributes are the primary means for extending SDP. Attributes may be defined to be used as "session-level" attributes, "media-level" attributes, or both. There MAY be any number of attribute lines; however, the only attribute line REQUIRED by this implementation profile is one specifying the dynamic payload format for DTMF events.

```
a=rtpmap:101 telephone-event/8000
```

It is RECOMMENDED, as stated in RFC 3264 [6], that attribute lines be included in the SDP for static payload type mappings. For example, G.711 u-law, GSM Full Rate, G.711 A-law, and G.729 all have a default clock rate of 8000 Hz and a default packet time of 20 milliseconds. The following attribute lines restate default values for these codecs.

```
a=rtpmap:0 PCMU/8000
a=rtpmap:3 GSM/8000
a=rtpmap:18 G729/8000
a=rtpmap:8 PCMA/8000
a=ptime:20
```

If different clock rates or packet times are desired, they MUST be specified explicitly in the SDP. To comply with the implementation profile, all BSIs MUST support the default values for clock rate and packet time. Support for other values is OPTIONAL, and such support MUST NOT be assumed.

The default media mode for audio sessions is “sendrecv”. This mode MUST be supported. Support for other values is OPTIONAL, and such support MUST NOT be assumed. The following attribute line is OPTIONAL because it restates the default value.

```
a=sendrecv
```

When using G.729, support for annex B is the default, as specified in RFC 3555 [23]. Annex B provides for voice activity detection (VAD) and comfort noise generation (CNG). BSI implementations are required to not send VAD or CNG packets; therefore BSI implementations are RECOMMENDED to state the lack of Annex B support whenever advertising support for G.729.

```
a=fmtp:18 annexb=no
```

When specifying the dynamic payload type for DTMF events, support for events 0-15 is REQUIRED. BSIs MUST support these events. Support for additional events is OPTIONAL. The following attribute line is REQUIRED to state the events the BSI is capable of receiving. For backward compatibility with pre-RFC 4733 implementations, if no "events" parameter is specified, support for the DTMF events 0-15 but for no other events should be assumed.

```
a=fmtp:101 0-15
```

One attribute not currently specified in the SDP is a preference for half-duplex versus full-duplex. For phase 1 of this profile, it is assumed that all SDP negotiations are implicitly half-duplex. It is anticipated that future phases of this profile will provide mechanisms for requesting half-duplex explicitly.

6.6 Example Offer/Answer Exchange

The following example illustrates an offer that both complies with all the requirements of the implementation profile and demonstrates how to indicate support for all the required and recommended codecs (IMBE is omitted because there is no defined RTP payload type for it at this time) with all the default values specified explicitly for illustrative purposes.

```
v=0
o=chn1 2890844526 2890844526 IN IP4 192.11.11.111
s=-
c=IN IP4 192.11.11.111
t=0 0
m=audio 49172 RTP/AVP 0 3 8 18 101
a=rtpmap:0 PCMU/8000
a=rtpmap:3 GSM/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:18 annexb=no
a=fmtp:101 0-15
a=sendrecv
a=ptime:20
```

The next example illustrates a corresponding answer indicating the selection of G.711 u-law as the audio codec and agreeing to use 101 as the DTMF payload type. While support for DTMF events 0-15 is not specified explicitly in the answer, it is implied because support for 0-15 is the default. Similarly, the absence of a specification of the mode as “sendrecv” and of the packet time as “20” is implied because they are the default values.

```
v=0
o=5000 2890844527 2890844527 IN IP4 192.22.22.222
s=-
c=IN IP4 192.22.22.222
t=0 0
m=audio 3456 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
```

```
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

7 Network

BSI interoperability requires an IP network that is adequate for bidirectional voice. The bandwidth should be appropriate for the expected number of simultaneous call sessions, and the jitter and packet loss should be considered as well. This document does not attempt to dictate all the requirements of the network layer in terms of bandwidth and supported services. However, basic network transport mechanisms are defined for the signaling and media layers, including a recommendation for IP layer packet marking of the media packets. Addressing schemes and naming conventions are defined as well, as are NAT/firewall traversal and high availability requirements.

7.1 Signaling Transport Layer

In accordance with RFC 3261 [1], all SIP implementations MUST support User Datagram Protocol (UDP) (RFC 768 [8]) and Transmission Control Protocol (TCP) (RFC 761 [9]). However, it is REQUIRED that BSIs compliant with this implementation profile use only TCP when interworking with other BSIs.

The reason for restricting use to only one of UDP or TCP is simply to minimize implementation, testing, and interoperability effort. The reasons for choosing TCP over UDP include the following:

- According to RFC 3261 [1], "If a request is within 200 bytes of the path [maximum transmission unit] MTU, or if it is larger than 1300 bytes and the path MTU is unknown, the request MUST be sent using an RFC 2914 [10] congestion controlled transport protocol, such as TCP". Using TCP from the start removes the need to operate in a dual UDP/TCP stack mode.
- Recently within the IETF, there have been talks of deprecating support for UDP. This may not happen, but it may be the case that some new mechanisms, such as the sip outbound draft, focus on TCP and drop UDP considerations.
- TCP lays the foundation for using TLS, which is a widely supported mechanism for securing communication and expected to be used in future phases of implementation profile to secure the signaling transport layer.

The use of TCP must be noted explicitly within the actual SIP messages. An example of this is the following SIP message (note the inclusion of "TCP" in the Via header and "transport=tcp" in the Contact header).

```
INVITE sip:5000@bsi2.example.com SIP/2.0
Via: SIP/2.0/TCP 192.11.11.111:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: <sip:chn1@bsi1.example.com>;tag=9fxced76sl
To: <sip:5000@bsi2.example.com>
```

```
Call-ID: 3848276298220188511@bsi1.example.com
CSeq: 1 INVITE
Contact: <sip:chn1@192.11.11.111;transport=tcp>
Content-Type: application/sdp
Content-Length: 189

v=0
o=chn1 2890844526 2890844526 IN IP4 192.11.11.111
s=-
c=IN IP4 192.11.11.111
t=0 0
m=audio 49172 RTP/AVP 0 101
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
```

The "TCP" in the Via indicates that responses as expected to be received over TCP, and the "transport=tcp" in the Contact indicates future requests related to this SIP dialog are expected to be received over TCP.

One potential concern with using TCP is its performance when operating in a high latency, low bandwidth (HLLB) environment. For such environments, it is RECOMMENDED to follow the strategies outlined in RFC 2488, "Enhancing TCP over Satellite Channels using Standard Mechanisms" [11].

7.1.1 Persistent Connections

It is highly RECOMMENDED by this implementation profile that the TCP connections between BSIs be maintained as persistent connections, not only for the duration of an individual SIP transactions but also across multiple transactions and multiple call sessions. The SIP community recommends that servers keep connections up unless they need to reclaim resources, and that clients keep connections up as long as they are needed. Connection reuse works best when the client and the server maintain their connections for long periods of time. SIP entities therefore SHOULD NOT automatically drop connections on completion of a transaction or termination of a dialog.

In some scenarios, it may be required for security or other reasons to open parallel TCP connections between two BSIs, one initiated by each BSI. BSIs in compliance with this implementation profile MAY initiate parallel TCP connections, and they MUST be able to accept parallel TCP connections initiated by other BSIs.

In order to maintain TCP connections, it is RECOMMENDED that implementations of this profile support the CRLF keep-alive technique specified in draft-ietf-sip-outbound-10 [22]. The client periodically sends a double-CRLF (the "ping") then waits to receive a single CRLF (the "pong"). If the client does not receive a "pong" within an appropriate amount of time, it considers the TCP connection failed. Support for the CRLF keep-alive technique MUST NOT be assumed. It is to be used only if support for it is advertised in the SIP signaling as described in [22].

7.2 Media Transport Layer

As specified in section 6.5.6, the REQUIRED media layer transport protocol is "RTP/AVP". Real-Time Transport Protocol (RTP), and Real-Time Control Protocol (RTCP), both specified in RFC 3550 [3], MUST be supported to be compliant with this profile. Other media transport protocols are OPTIONAL at this time.

Further, standard RTP MUST be used. Additionally, header extensions to the RTP/AVP profile MUST follow the guidance in draft-ietf-avt-rtp-hdrext-13 [19], and those extensions MUST be documented within this profile. Header extensions or proprietary headers MAY be used; however, support for such extensions or headers MUST NOT be assumed to be supported by other BSIs.

7.2.1 IP Layer Packet Marking

It is critical that the packets required to construct the media streams at either end of the BSIs are delivered in a timely fashion in order to avoid choppy or unintelligible speech. To aid in the effort, this first phase of the implementation profile RECOMMENDS support for service marking as specified in RFC 2475 [17]. More specifically, it is RECOMMENDED to set the Type of Service (TOS) field of all media packets to request the network to minimize delay as specified in RFC 1349 [18]. Note that setting the TOS field of the media packets is a recommendation for the BSI application, and supporting the TOS field setting is a recommendation for the network.

7.3 Addressing

IPv4 (Internet Protocol version 4) MUST be supported and will be the addressing scheme used by implementation profile. Although there are other schemes available, like IPv6, these are not as ubiquitous as IPv4.

Support for IPv6 is a subject for future study.

7.4 Naming Conventions

Within the SIP domain, resources are identified by SIP Uniform Resource Indicators (URIs). The implementation profile for BSI interoperability RECOMMENDS that BSIs conform to a hierarchical naming convention of SIP URIs for the resources they intend to share with other BSIs. These SIP URIs SHOULD be of the form:

sip:<Resource Name>@<Jurisdiction Domain Name>

where:

<Resource Name> is a unique name within the given jurisdiction domain

and

<Jurisdiction Domain Name> is the jurisdiction in which the bridging system is operating.

This implementation profile does not place any requirements on the format of the <Resource Name> in this phase; however, it is anticipated that future phases of this profile will recommend the use of descriptive, alpha-numeric resource names, possibly requiring specific formatting of these names to aid in things such as loop prevention.

It is REQUIRED that implementations of this profile support at least 128 byte SIP URIs, and RECOMMENDED that they support at least 1024 byte SIP URIs.

7.5 NAT and Firewall Traversal

NAT and firewall traversal is one of the most complex and debated topics within the SIP community. Numerous internet drafts and RFCs related to this topic are published or in progress. The implementation profile for BSI interoperability postpones detailed recommendations within this area for phase 1. Rather, a BSI MUST use routable IP addresses, and any firewalls between BSIs MUST open ports for SIP signaling and RTP/RTCP media between the BSIs. Support for symmetric responses for signaling and symmetric RTP and RTCP for media, as described in RFC 4961 [21], is RECOMMENDED. It is also RECOMMENDED that the range of RTP/RTCP ports used be configurable.

7.6 High Availability

Due to the nature of the public safety market, it is critical that any solution for BSI interoperability takes high availability into account. However, recommendations for network and system design to achieve this are outside of the scope of this document. It is anticipated that a Best Common Practices document regarding this subject will be published in the future.

8 Security

Security, especially in the public safety market, is paramount. SIP includes various security mechanisms, such as digest authentication for SIP requests [1], TLS to secure the SIP signaling [13], and SRTP to secure the media [14]. Numerous other security mechanisms are defined for SIP or are in the process of being defined.

However, for this phase of the implementation profile, it is REQUIRED that BSIs be able to interoperate in the absence of such security mechanisms. Instead, the assumption is made that the IP network connection between BSIs is secured through mechanisms such as IPSec [15], VPNs [16], etc.

9 Management

This specification recognizes the need for management for BSI interoperability; however, for phase 1, management is considered to be out of scope. It is assumed that agencies wishing to have their BSIs interwork exchange corresponding IP addresses and resource names, and that they enable SIP signaling and media traffic between their BSIs at their own discretion. Mechanisms such as the exchange of certificates for authentication and the use of DNS and SIP registrars for registering and locating resources is left for future phases of the implementation profile.

The following table provides a summary of the BSI related information agencies are expected to exchange for phase 1.

	PARAMETER TO BE PRE-EXCHANGED	**DESCRIPTION**	**COMMENT/STATUS**
1	SIP signaling IP address	The IP address of a host that runs the BSI SIP signaling entity (UAC/UAS)	REQUIRED
2	SIP signaling port	The TCP port number used for BSI SIP signaling	REQUIRED; 5060 is the default if not specifically exchanged
3	Media IP address(es)	The media IP address(es) used to send and receive RTP/RTCP audio packets during a BSI media session	RECOMMENDED for the benefit of firewall pre-configuration
4	RTP/RTCP media port range	The media UDP ports range used to send and receive RTP/RTCP audio packets during a BSI media session	RECOMMENDED for the benefit of firewalls pre-configuration
5	Resource identifier(s) (SIP URI(s))	SIP URI(s) representing radio resource(s) at a BS. One SIP URI is specified for each resource.	REQUIRED; the format sip:<ResourceName> @<JurisdictionDomain> is RECOMMENDED.

10 Push-to-Talk (PTT)

Several existing radio gateways and BSI systems rely on some function tone or signal to know when to key up the radio attached to the radio gateway/BSI. Because many of these signals or tones are proprietary, the "signal" to key up or key down in the first phase of the implementation profile is the presence or lack of audio packets within the media stream established by the SIP call session. Voice packet detection MUST be supported in the BSI. This REQUIRES that RTP packets NOT be sent representing silence.

The BSI or radio gateway MAY continue to use proprietary tones or signals internally to know when to key up the radio, but the implementation profile requires voice packet detection only, where voice packet detection is defined as the reception of RTP audio packets. Any non-audio based RTP packets, such as RTP keep-alive packets, do not result in voice packet detection.

10.1 Detecting Loss of Media

One reason support for RTCP is REQUIRED by this implementation profile is to detect the loss of media within a SIP call session. Given the PTT nature of the media streams, it is possible that no RTP packets are exchanged for long periods of time. Therefore, periodic RTCP packets MUST be sent to, among other things, indicate that the media stream is still active. RTCP packets SHOULD be sent as specified in RFC 3550 [3]; and it is REQUIRED that the rate be at least one RTCP packet every 5 seconds.

It is RECOMMENDED that BSIs monitor the RTP and RTCP traffic for each media stream. If a loss of RTP/RTCP packets is detected (i.e. no RTP and no RTCP packets for some configurable amount of time), the media stream is considered lost. At this time, the BSI detecting the loss of media SHOULD send a re-INVITE with the same media description as negotiated in the previous offer/answer exchange. This is done by sending a re-INVITE with SDP with the same session version as the original SDP it sent for that call session. If the session still exists on the remote BSI, it SHOULD respond with a 200 to the re-INVITE, and include SDP with the same session version as the original SDP it sent for that call session.

Hopefully the success of the re-INVITE results in the re-establishment of RTP and/or RTCP for the session. If the re-INVITE fails, or if the loss of media persists despite the success of the re-INVITE, the BSI detecting the loss should tear down the session by sending a BYE.

The BSI initially configured to establish the session MAY attempt to re-establish the session at a later time, including immediately. The RECOMMENDED algorithm for session re-establishment is to retry once immediately. If that retry fails, retry periodically with the period between retries being pseudo random up to every 300 seconds. The randomness is to avoid periodic floods of reestablishment attempts.

11 Open Standards

Everything mentioned in this specification in terms of protocols is based on open standards. That is to say that SIP, SDP, RTP, and RTCP are open standards based on years of use and availability. By not relying on any proprietary mechanisms, this implementation profile facilitates the rapid development of low cost solutions for BSI interoperability.

12 Changes from Previous Versions

12.1 Changes from version 0.7 to 1.0

- Modified DTMF event support to conform to changes made to RFC 2833 by RFC 4733. The specification of the supported DTMF events in the SDP was changed from optional to required.

12.2 Changes from version 0.6 to 0.7

- Added reference to RFC 3555 for G.729 MIME/SDP encoding.

- Corrected specification of G.729 Annex B.
- Reworded handling of re-INVITEs in section 6.1.

12.3 Changes from version 0.5 to 0.6

- Specified sending of OPTIONS requests as optional in section 5.1.1.
- Clarified call release handling in section 5.4.
- Clarified sending and receiving of re-INVITEs that attempt to modify the media session in section 6.1.
- Removed lower bound on retry interval in section 10.1.

12.4 Changes from version 0.4 to 0.5

- Added recommendation that the SIP URI included in the From header be the address of record (AoR) of the calling resource.
- Clarified that DNS support is optional, and changed the sample message flows to use IP addresses instead of hostnames wherever resolvable IP addresses are required.
- Stated that half-duplex is implied within offer/answer negotiation for phase 1.
- Replaced requirement for support for 1024 byte URIs with requirement for 128 byte URIs and recommendation for 1024 byte URIs.
- Added requirement to accept parallel TCP connections.
- Added recommendation to support CRLF keep-alive technique for TCP connections.
- Added table summarizing the BSI related information agencies are expected to exchange for phase 1.

12.5 Changes from version 0.3 to 0.4

- Changed the title and the corresponding text in the rest of the documents to refer to the profile as an implementation profile rather than a implementation profile.
- Updated the abstract to differentiate between a bridging system and the Bridging System Interface (BSI).
- Clarified that support for DTMF by the bridging system via its non-BSI interfaces is out of scope for this profile.

12.6 Changes from version 0.2 to 0.3

- Moved scope information from Introduction section to its own Scope section.
- Added revisions section as section 12.
- Added reference to draft-ietf-avt-rtp-hdrext-13.txt for RTP extensions.
- Moved GSM Full Rate from REQUIRED to RECOMMENDED, and added IMBE as a RECOMMENDED codec.
- Added caveats regarding voice encoder tandeming as section 6.3.3.
- Added requirement that RTP port numbers in media line be even, with (port + 1) being for RTCP.
- Removed the recommendation to omit the specification of SDP attribute lines restating default values. This was done to comply with RFC 3264 [6].
- Added that network must be adequate for voice to Network section.
- Changed use of TCP for signaling from highly RECOMMENDED to REQUIRED.

- Removed recommendation for limiting <Resource Name> to numeric values.
- Added recommendation to support 1024 bit SIP URIs at a minimum.
- Added recommendation for symmetric responses for signaling and symmetric RTP/RTCP to NAT and Firewall Traversal section.
- Modified wording of pre-configured and ad-hoc call sessions

13 References

[1] 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.

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

[3] Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.

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

[5] Johnston, A., Donovan, S., Sparks, R., Cunningham, C., and K. Summers, "Session Initiation Protocol (SIP) Basic Call Flow Examples", RFC 3665, December 2003.

[6] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with SDP", RFC 3264, June 2002.

[7] Schulzrinne, H. and T. Taylor, "RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals", RFC 4733, December 2006.

[8] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[9] Postel, J., "DoD Standard Transmission Control Protocol", RFC 761, January 1980.

[10] Floyd, S., "Congestion Control Principles", RFC 2914, September 2000.

[11] Allman, M., Glover, D., Sanchez, L., "Enhancing TCP over Satellite Channels using Standard Mechanisms", RFC 2488, January 1999.

[12] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[13] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[14] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[15] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[16] Rosen, E. and Y. Rekhter, "BGP/MPLS VPNs", RFC 2547, March 1999.

[17] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, "An Architecture for Differentiated Service", RFC 2475, December 1998.

[18] Almquist, P., "Type of Service in the Internet Protocol Suite", RFC 1349, July 1992.

[19] Singer, D., H. Desineni, "A general mechanism for RTP Header Extensions", draft-ietf-avt-rtp-hdrext-13.txt, August 2007.

[20] "APCO Project 25 Vocoder Description", TIA/EIA/IS-102.BABA, January 1993.

[21] D. Wing, "Symmetric RTP / RTP Control Protocol (RTCP)", RFC 4961, July 2007.

[22] Jennings, C. and R. Mahy, "Managing Client Initiated Connections in the Session Initiation Protocol (SIP)", draft-ietf-sip-outbound-10, July 2007.

[23] Casner, S. and P. Hoschka, "MIME Type Registration of RTP Payload Types", RFC 3555, July 2003.