NIST/OLES VoIP Roundtable
Category: Informational
Version 1.1

R. Mitchell
Twisted Pair Solutions
C. Eckel
Cisco
J. Mathis
Motorola
September 2009

Implementation Profile for Interoperable Bridging Systems Interfaces (BSI-Core 1.1)

Status of this Memo

This document specifies an updated Voice over Internet Protocol (VoIP) implementation profile for the public safety community. Discussion and suggestions for improvements are invited. Distribution of this memo is unlimited.

Abstract

This document describes a minimum set of standards, parameters, and values (i.e., an implementation profile) that are required for interoperability among bridging systems. The document is written for manufacturers, developers, and integrators that intend to increase public safety communications interoperability through the use of bridging systems and VoIP.

A bridging system is a device that enables voice communication among disparate radio frequencies, systems, or technologies. The disparate devices connected via a bridging device may include land mobile radios, analog phones, mobile phones, IP telephones, and personal computers; however, this is not an exhaustive list of connective devices. The interface through which bridging systems communicate with each other is the Bridging Systems Interface (BSI).

When an interoperability setup uses more than two bridges, the BSI-Core bridges operate as a loosely coupled conference with end-system mixing, rather than as a tightly coupled conference (such as those described in Request for Comment [RFC] 4353). A loosely coupled conference brings simplicity at the expense of scale.

Session Initiation Protocol (SIP) provides the basis for the BSI implementation profile. 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. IP-based protocols commonly used with SIP are also used in the BSI implementation profile.

Version 1.1 describes a backwards-compatible update to the BSI-Core 1.0 profile. Implementations conforming to BSI-Core 1.0 and correctly implementing the required RFCs still conform to BSI-Core 1.1.

Table of Contents

1 Introduction/Background

The pervasiveness of land mobile radio systems[1] in the market today provides for wide-ranging, broad-based communications between radio users and between radio users and dispatchers. Land mobile radio systems employ multiple technologies and standards including analog, digital, trunked, Project 25 (P25), and TETRA, and operate in different frequency bands such as UHF, VHF, HF, 700 MHz, and 800 MHz. For the remainder of this document, the terms "radio" and "radio systems" shall refer to the collective group of disparate radio technologies and frequencies used today in the public safety community.

Radio systems are the backbone of mobile communications – not only for public safety, but also in other markets such as defense, transportation, and utilities. Many industries use radios for such a wide variety of reasons that advances in radio technology offer a multitude of services to subscribers. For example, some radio systems offer data, dual-tone multi-frequency (DTMF), and other services in addition to audio services. Due to different agency requirements and budget cycles, each agency may have a different radio system – this is true even for individual agencies, which can deploy multiple radio systems within themselves. Although different radio systems are deployed, agencies must still be able to communicate among themselves and on an intra- and inter-region level as well as all the way up to the state and Federal government.

Regardless of agency (city, county, state, tribal, or Federal; law enforcement; fire; EMS; etc.), interoperability between radio systems is crucial to emergency responders. Agencies need the ability to communicate with other agencies, regardless of level, when authorized to do so. However, incompatibilities between different radio systems often make this difficult. To enable multiple radio systems to communicate with each other, vendors developed Bridging Systems Interface (BSIs) 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 as well. However, for the purpose of this document, operational policies are considered out of scope, and are addressed in the BSI Best Practices Document.[2]

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.

Session Initiation Protocol (SIP), anchored by Request for Comment (RFC) 3261 [Reference 1] and extended by many other RFCs, provides a well-defined infrastructure for establishing communication sessions. The goal of this document is to specify an

[1] For the purposes of this document, the terms "radio" and "radio systems" refer to the collective group of disparate radio technologies and frequencies in use today by the public safety community.
[2] http://www.safecomprogram.gov/NR/rdonlyres/E831E013-B893-4CFA-93F8-47C36B615BFB/0/BSIBestPracticesFINAL_42010.pdf

implementation profile that narrows the field to the requirements to establish audio communication channels between BSIs within the public safety space. This is the second version of this document, and provides backward-compatible clarifications to the earlier document.

2 Overview of Bridging Systems Interfaces

A BSI is a hardware or software platform that enables radio system or radio gateway interoperability. The following diagram illustrates where BSIs fit into the overall system architecture:

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

Note that this architecture is not indicative of every scenario for a BSI. 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, computers) can be considered a BSI. These BSIs are often stand-alone in nature and function on their own. Radio gateways that enable interoperable RoIP connections between those gateways can communicate with each other using the BSI protocol.

BSIs and radio devices are similar in that market demands, timing, and budget cycles can affect which radio devices or BSIs are available, either within an agency or between agencies. When multiple agencies or groups within an agency use different BSIs, they need a method for these BSIs to interoperate.

The following figures show several examples of potential topologies of BSI interconnections. A variety of IP technologies (including private IP networks, virtual private network [VPN] over public/private IP networks, and IP satellite links) may achieve these interconnection links. The Best Practices document [References 30] contains additional information on how to engineer the networks that support the BSI interconnections to help ensure the necessary voice quality. The interconnection links between bridges may be either pre-configured or ad hoc.

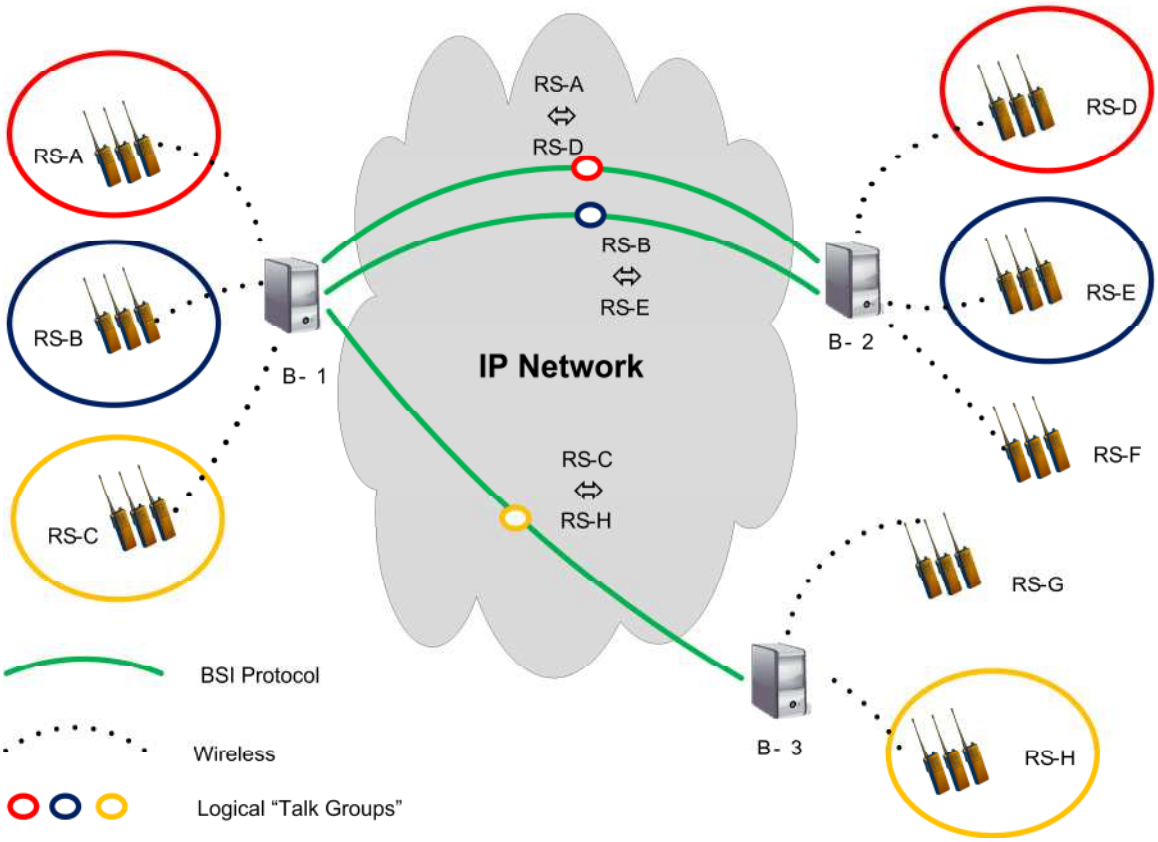


Figure 1. Basic Example Interconnect Scenario for BSI

Figure 1 illustrates the simplest topology, where pair-wise interconnection is defined for several radio systems. The diagram contains eight radio systems (RS-A through RS-H) and three bridging systems (B-1 through B-3). Each bridging system includes a radio (referred to as a "donor radio") from the radio system (RS) to which it is connected. The green lines show the bridging interconnections that allow users on one radio system to communicate with users on another system, despite radio technology incompatibilities and differing radio bands (e.g., RS-A and RS-D, RS-B and RS-E, and RS-C and RS-H).

The interconnected agencies are responsible for agreeing on mutually acceptable security means (e.g., VPNs) to protect traffic over the BSI links. Each agency is also responsible for issues relating to traversal of its firewalls.

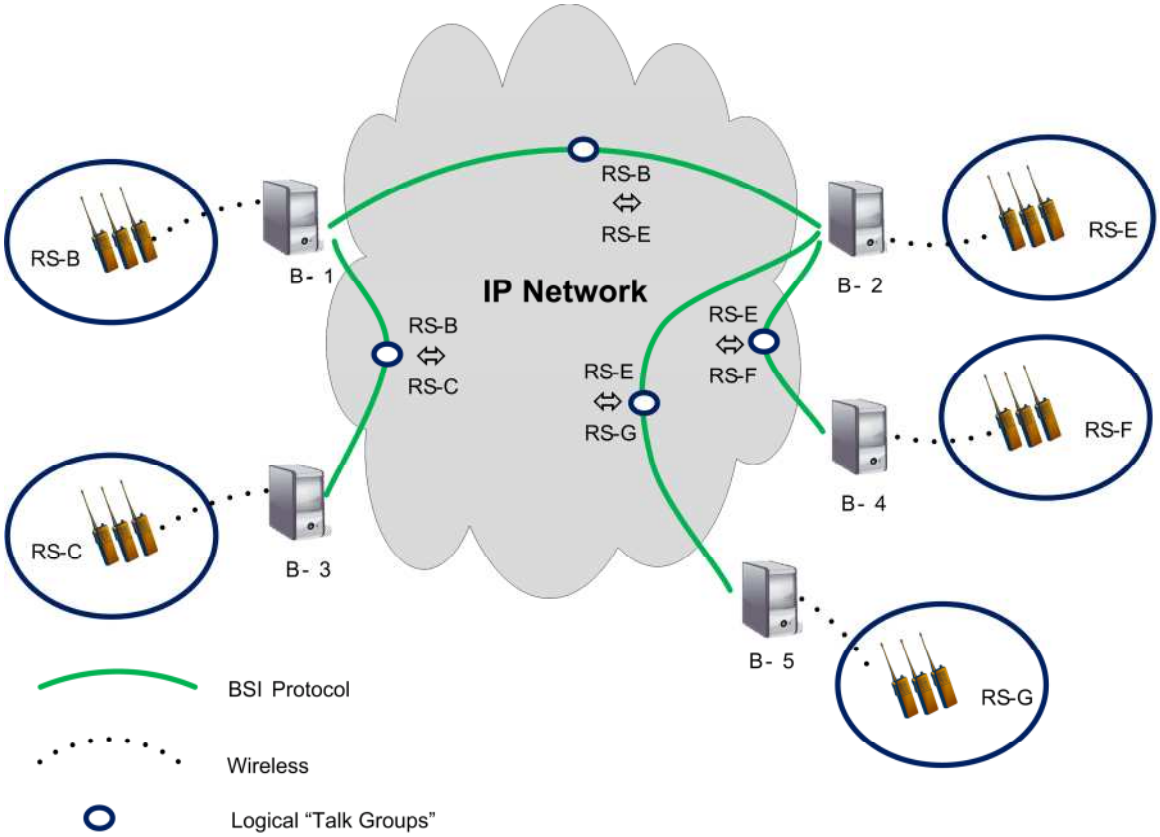


Figure 2. Five Bridge Example Interconnect Scenario for BSI

Figure 2 illustrates a more complex topology where multiple radio systems are interconnected. This topology of five bridges and donor radios could satisfy the U.S. Department of Homeland Security "Tanker Truck Rollover Scenario" [31].

The green lines show the bridging interconnections that allow, for example, users on RS-B to communicate with users on RS-C, RS-E, RS-F, and RS-G, despite possible incompatibilities of radio technology and differing radio bands. This diagram also illustrates the need for bridges to repeat interconnections to other local bridges to eliminate the need for a fully interconnected mesh of nodes. For example, the bridges on the left side of Figure 2 could belong to the agencies of one state, and the bridges on the right side could belong to the agencies of an adjacent state.

The addition of multiple bridges also brings up the issue of the interconnection configuration varying over time. For example, in Figure 2 the interconnection of RS-B and RS-C may be set up independently of the interconnection of RS-E, RS-F, and RS-G. After this initial deployment, the RS-B to RS-E link is established effectively interconnecting the five radio systems.

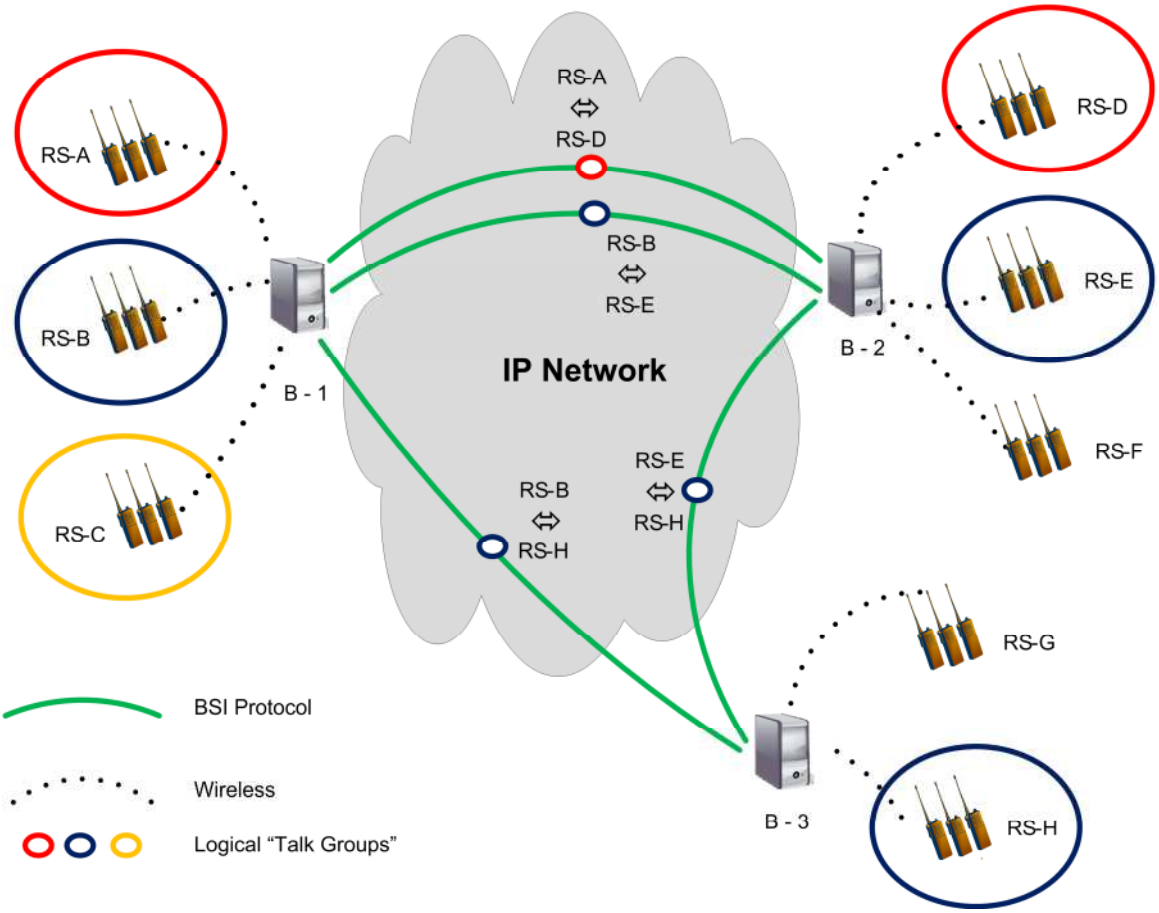


Figure 3. Illegal Looped Interconnect Scenario

Figure 3 illustrates how an incorrectly configured set of bridges introduces an audio feedback loop. While each individual BSI link looks correct, closer inspection shows that RS-B is interconnected to RS-E, which is interconnected to RS-H, which is interconnected back to RS-B. Because this loop is on the IP network, the Real-Time Transport Protocol (RTP) loop detection procedure may identify the loop and prevent audio feedback. The operator should take action to correctly configure the bridge interconnections.

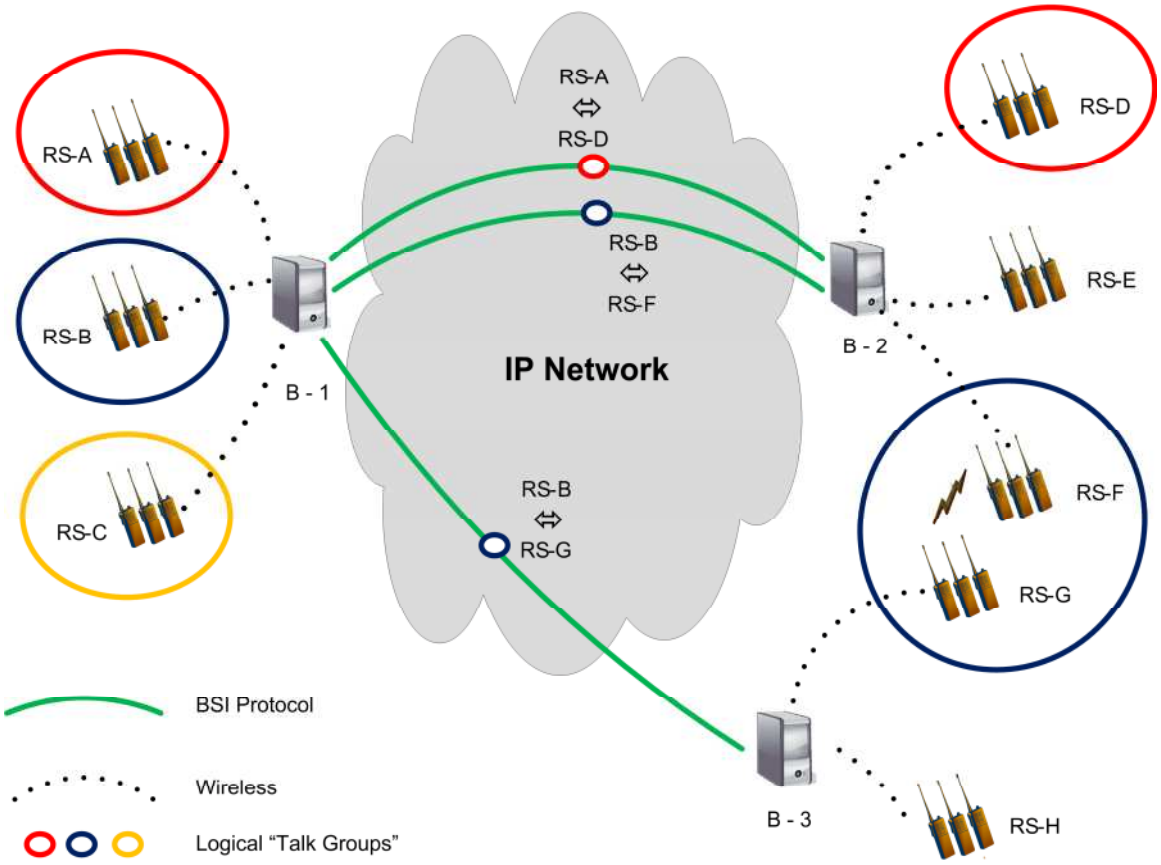


Figure 4. Illegal RF Looped Scenario

Incorrect configurations introduce audio feedback loops such as those enabled by multiple donor radios connecting to the same system, channel, or talkgroup. Figure 4 illustrates a case where RS-G and RS-F are on the same radio channel in the same coverage area (such as a mutual aid channel). Because this loop is created outside of the BSI interconnect network, the bridges cannot detect this loop using the RTP mechanisms defined for the BSI profile. Identifying and resolving audio feedback loops created by incorrectly configured donor radios or radio systems is beyond the scope of this profile.

This document describes an implementation profile that serves as the initial implementation profile for the BSI protocol. In this case, that means that this document specifies the minimum set of standards, parameters, and values required to successfully implement an interoperable BSI protocol.

SIP serves as the basis of the BSI 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 RTP (RFC 3550 [3]) for transporting real-time data and providing Quality of Service (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, use SIP to identify an audio channel accessible via a BSI and to 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 to manage a conference. SIP can be used to initiate a session that uses some other conference control protocol.

SIP provides a suite of security services including denial-of-service prevention, authentication (both user to user and proxy to user), integrity protection, and encryption and privacy services. However, to promote basic interoperability, the BSI profile does not specify specific security services; instead, it assumes that the deployment environment will include the necessary security mechanisms achieved though, for example, secure tunnels, VPN, or Security Gateways (SEG) [24]. The many different mechanisms, algorithms, and customer approaches for securing traffic makes picking one for use in BSI impractical. When bridges are deployed, the interconnected agencies must reach a mutually acceptable agreement on how security is provided.

SIP and the related protocols all work with both the IPv4 and IPv6 networks. However, all the examples in this document assume IPv4 and only support for IPv4 is required.

3 Scope

This implementation profile provides value to manufacturers (for development purposes) and purchasers (for specification and conformance purposes) of devices with a BSI. The goals in developing it include:

- 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 to use this functional subset between BSIs, including naming conventions adhered to across compliant BSIs, and identification of prior 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.

This document assumes that subsequent revisions and extensions of the 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. For example, although this document does not address floor control, session and media negotiation mechanisms put in place during this phase provide mechanisms that may extend to future phases to address floor control. Other requirements slated to be addressed in future versions include the 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, interpret the key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "NOT RECOMMENDED," "MAY," and "OPTIONAL" as they are described in BCP 14, RFC 2119 [2] and indicate requirement levels for compliant implementation.

5 Signaling Layer

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

SIP, as outlined in RFC 3261 [1], is 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 allows 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, reference the exact syntax of messages (e.g., call setup, in-call handling, call tear down), message processing (e.g., errors, unrecognized responses), and timer handling in [1] and its related RFCs. This document identifies the minimum set of functionality REQUIRED to comply with the implementation profile for BSI and offers suggestions for RECOMMENDED functionality. The following subsections describe the SIP functionality employed by the BSI Profile [I'm assuming this is true].

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-Uniform Resource Indicator (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 RFC 3261 [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 set up sessions, BYE terminates sessions, and OPTIONS queries 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]. This implementation profile MUST support SIP URIs. 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 three-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

The INVITE method initiates sessions within SIP. Section 13 of RFC 3261 [1] discusses, in detail, how the User Agent Client (UAC) and User Agent Server (UAS) 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. Refer to Section 7.4 for additional recommendations related to the SIP URI.

If the requested resource is unknown at the receiving BSI, the BSI SHOULD return a response of "404 Not Found". For security, it is RECOMMENDED that in the case of a non-matching URI, the BSI device SHOULD return a "400 Bad Request" or "404 Not Found" and SHOULD NOT use "484 Address Incomplete" response.

If the resource is known at the receiving BSI, but is currently unavailable for some reason, the BSI SHOULD respond with "480 Temporarily Unavailable" and SHOULD provide a reason-phrase in the error message. See also Section 5.2.3 for busy resources.

While exactly how a BSI determines whether or not to allow requests from another BSI is not in scope for this implementation profile, if a BSI chooses to not allow the connection to a known and available resource it is SHOULD return a "403 Forbidden" response.

If the BSI is rebooting, or in some other mode (e.g., maintenance) and unable to accept connections, it is MAY respond with "503 Service Unavailable", or it MAY refuse the connection or drop the request instead of returning a "503 Service Unavailable" response.

5.2.1 Rules of Engagement

Certain rules of engagement MUST be followed to facilitate call sessions between BSIs. In some situations, an end-user customer desires well-known and previously configured BSI resources. Other situations might desire on-the-fly, ad hoc resources. 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 the former is pre-configured to accept inbound calls from or initiate outbound calls to specified BSIs at any time. The resulting call sessions may be long-duration sessions that the participating agencies agreed upon or activated only when necessary. Ad hoc configuration facilitates unanticipated resource sharing, which would require on-the-fly configuration 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, establishing a call session between two BSIs is possible only after the corresponding agencies enable such sessions via pre-configuration or ad hoc configuration. While the BSI implementation profile provides for both pre-configured and ad hoc modes of operation, it treats both cases operationally as though they were essentially the same. Agencies wishing to interconnect their systems through bridges that implement the BSI-Core profile need to exchange and configure the same amount of information (as described in detail in Section 9) for either mode of operation. This document anticipates that future phases of this profile will define mechanisms that simplify ad hoc operations for the benefit of the public safety community.

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

5.2.2 Configuration Information

To assist in setup, configuration, and debugging of BSI configurations, the SIP call control signaling can communicate various pieces of static configuration information. This information SHOULD be logged or be displayed or be readily displayable to the BSI operator.

A BSI MAY include a display name per Section 8.1.1.3 of RFC 3261 [1] in the "From:" and "Contact:" headers of the SIP INVITE; the BSI MAY use display names in other headers as appropriate. The contents of the display name are not intended for automated processing and SHOULD be constructed by each bridge in a way to maximize user understanding. Since this is likely to involve the use of white space, this display name MUST be a single quoted string that conveys the static configuration information in a meaningful way.

A BSI SHOULD prepare to receive and handle a display name of at least 32 characters, although the sending BSI sets the exact length and format of the display name. The contents of the display name SHOULD be configurable in each BSI but the contents are otherwise not specified in the BSI profile. The Best Practices [30] document has suggestions for how the display name may be used.

5.2.3 Multiple Call Sessions

If a BSI accepts multiple call sessions to a given resource, it SHOULD implement the Media Forwarding requirements for multiple call sessions as specified in Section 6.4; otherwise the BSI SHOULD reject SIP INVITE call attempts from additional BSIs. Using a "486 Busy Now" is the RECOMMENDED SIP response; but the BSI MAY use other 4xx response codes, such as "480 Temporarily Unavailable"; use of 5xx or 6xx response codes is NOT RECOMMENDED.

5.3 In-Call Session Control

SIP includes the ability to modify an established session 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. Section 10.1 discusses this in more detail.

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. However, a BSI implementation MUST correctly handle a re-INVITE that does not change the parameters of the call session since some implementations may use this as a keep-alive technique.

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

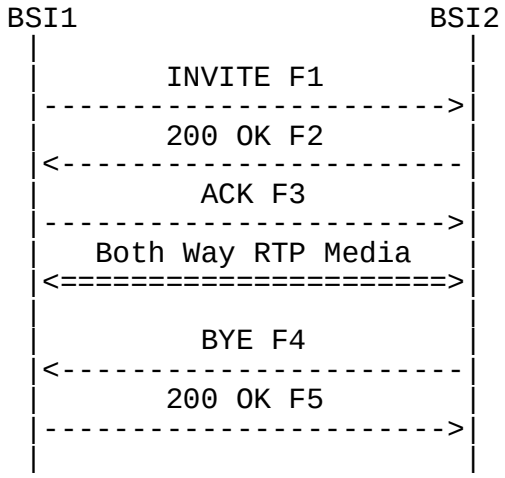
All call sessions MUST be gracefully released by sending a BYE, except in situations in which doing so is not possible due to hardware/software failures, loss of connectivity, etc. BYE instructs the User Agent (UA) on the far side that the party wishes to disconnect. At that point, both UAs MUST stop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

Call sessions may end non-gracefully due to hardware or software failure, loss of connectivity, and loss of media. 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 compliant with the implementation profile. The successful negotiation between the agencies responsible for BSI1 and BSI2 is assumed to have resulted in the configuration of LE 12 (sip:LE12@bsi2.example.com) as an identifier for a channel existing on BSI2 that BSI1 should be able to access. On BSI1, LE 1 (sip:LE1@bsi1.example.com) originates the connection; however, the identifier for LE 1 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. The implementation profiles REQUIRES BSI2 to provide some means of configuration to allow such requests. One option is to use the SIP URI in the "From:" header, as described in Section 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.



In this scenario, BSI1 completes a call to BSI2 directly. The use of Domain Name System (DNS) resolvable hostnames (e.g., bsi1.example.com, bsi2.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. Note that Call-ID is not a SIP routing header and including DNS names as part of the text of the Call-ID string is allowed.

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.

5.5.1 Flows for BSI Using Only G.711 Codec

Per Sections 6.3 and 6.5, the required audio codec for BSI is G.711 and the DMTF events MUST be supported. This results in the following example message exchange.

F1 INVITE BSI1 -> BSI2

```
INVITE sip:LE12@bsi2.example.com SIP/2.0
Via: SIP/2.0/TCP 192.11.11.111:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: <sip:LE1@bsi1.example.com>;tag=9fxced76sl
To: <sip:LE12@bsi2.example.com>
Call-ID: 3848276298220188511@bsi1.example.com
CSeq: 1 INVITE
Contact: <sip:LE1@192.11.11.111;transport=tcp>
Content-Type: application/sdp
Content-Length: 194

v=0
o=LE1 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
a=fmtp:101 0-15
```

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
f: <sip:LE1@bsi1.example.com>;tag=9fxced76sl
t: <sip:LE12@bsi2.example.com>;tag=8321234356
i: 3848276298220188511@bsi1.example.com
CSeq: 1 INVITE
m: <sip:LE12@192.22.22.222;transport=tcp>
c: application/sdp
l: 194

v=0
o=LE12 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
```

F3 ACK BSI1 -> BSI2

```
ACK sip:LE12@192.22.22.222 SIP/2.0
Via: SIP/2.0/TCP 192.11.11.111:5060;branch=z9hG4bK74bd5
Max-Forwards: 70
From: <sip:LE1@bsi1.example.com>;tag=9fxced76sl
To: <sip:LE12@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:LE1@192.11.11.111 SIP/2.0
Via: SIP/2.0/TCP 192.22.22.222:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
f: <sip:LE12@bsi2.example.com>;tag=8321234356
t: <sip:LE1@bsi1.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@bsi1.example.com
CSeq: 1 BYE
l: 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:LE12@bsi2.example.com>;tag=8321234356
To: <sip:LE1@bsi1.example.com>;tag=9fxced76sl
```

```
Call-ID: 3848276298220188511@bsi1.example.com
CSeq: 1 BYE
Content-Length: 0
```

Also note this example includes the use of the compact form of SIP header fields, support for which is REQUIRED in RFC 3261.

5.5.2 Flows for BSI Using Optional Codec

BSI also allows the use of optional codecs. Using the SDP offer/answer negotiation, any codec can be offered as receivers MUST ignore codecs they do not understand or support. Section 6.3.1 lists suggested optional voice encoders that may be useful in bridging situations.

By some mechanism not defined in the profile, BSI1 decides to prefer to use the lower data rate GSM 06.10 13.2 kbps optional codec over the G.711 PCMU 64 kbps required codec and lists the GSM 06.10 codec first in the media offer. Because the G.711 is required, it must be listed in every initial offer but may be listed last.

F1 INVITE BSI1 -> BSI2

```
INVITE sip:LE12@bsi2.example.com SIP/2.0
Via: SIP/2.0/TCP 192.11.11.111:5060;branch=z9hG4bK74bf9
Max-Forwards: 70
From: "LE 1" <sip:LE1@bsi1.example.com>;tag=9fxced76sl
To: <sip:LE12@bsi2.example.com>
Call-ID: 3848276298220188511@bsi1.example.com
CSeq: 1 INVITE
Contact: "LE 1" <sip:LE1@192.11.11.111;transport=tcp>
Content-Type: application/sdp
Content-Length: 219

v=0
o=LE1 2890844526 2890844526 IN IP4 192.11.11.111
s=-
c=IN IP4 192.11.11.111
t=0 0
m=audio 49172 RTP/AVP 3 0 101
a=rtpmap:3 GSM/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

Since the GSM 06.10 codec was listed first in the "m=" offer and is supported by BSI2, the GSM 06.10 codec would typically be selected by BSI2 (see Section 6.1 of RFC 3264 [6]). If the offer had the G.711 codec listed first, then by some mechanism not defined

in the profile, BSI2 could decide to respond to BSI1 accepting the lower data rate GSM 06.10 codec rather than the G.711 codec also proposed.

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: "LE 1" <sip:LE1@bsi1.example.com>;tag=9fxced76sl
To: <sip:LE12@bsi2.example.com>
Call-ID: 3848276298220188511@bsi1.example.com
CSeq: 1 INVITE
Contact: "LE 12" <sip:LE12@192.22.22.222;transport=tcp>
Content-Type: application/sdp
Content-Length: 195

v=0
o=LE12 2890844527 2890844527 IN IP4 192.22.22.222
s=-
c=IN IP4 192.22.22.222
t=0 0
m=audio 3456 RTP/AVP 3 101
a=rtpmap:3 GSM/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
```

F3 ACK BSI1 -> BSI2

```
ACK sip:LE12@192.22.22.222 SIP/2.0
Via: SIP/2.0/TCP 192.11.11.111:5060;branch=z9hG4bK74bd5
Max-Forwards: 70
From: "LE 1" <sip:LE1@bsi1.example.com>;tag=9fxced76sl
To: <sip:LE12@bsi2.example.com>
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:LE1@192.11.11.111 SIP/2.0
Via: SIP/2.0/TCP 192.22.22.222:5060;branch=z9hG4bKnashds7
Max-Forwards: 70
From: "LE 12" <sip:LE12@bsi2.example.com>;tag=8321234356
To: <sip:LE1@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: "LE 12" <sip:LE12@bsi2.example.com>;tag=8321234356
To: <sip:LE1@bsi1.example.com>;tag=9fxced76sl
Call-ID: 3848276298220188511@bsi1.example.com
CSeq: 1 BYE
Content-Length: 0
```

5.6 Proxy Servers

Proxy Servers route SIP requests from UACs to 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 the BSI-Core implementation profile. Implementations compliant with the implementation profile MUST support direct communication with another BSI. Implementations compliant with this profile MUST NOT assume support for routing SIP messages through proxy servers.

5.7 Registrars

SIP Registrars allow 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 are OPTIONAL for the BSI-Core 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 SDP (RFC 4566 [4]). The SIP message carries this SDP message (shown in the examples in Section 5.5) in a way that is analogous to an e-mail message carrying a document attachment, or a HTTP message carrying a Web page. 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 and provides a variety of ways in which media negotiation may occur between two endpoints. For the purpose of the implementation profile, specify a limited set of mechanisms as REQUIRED for a BSI to be compliant with the profile.

The example flows in Section 5.5 show the classic offer/answer exchange in which the offer is included by BSI1 in the INVITE request and the answer is included by BSI2 in the 200 OK response. At a minimum, a BSI MUST support this exchange 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].

Section 10.1 describes the use of re-INVITEs that do not modify the media session for the purpose of recovering from media loss.

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. This implementation profile limits the media negotiated by the offer/answer exchange to a single audio stream. Attempts to negotiate multiple audio streams or a non-audio stream (e.g., 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 REQUIRED 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 Section 6.6.6.

To support loop detection, a BSI MUST include a synchronization source (SSRC) in the RTP stream allocated as described in Section 8 of RFC 3550 [3] and MUST NOT use a constant value for an SSRC or simply its IP address. A BSI MUST identify and resolve SSRC value collisions as described in Section 8 of RFC 3550. These are standard requirements on RTP implementations compliant with RCF 3550 but are called out since audio loops may be a problem with complex ad hoc BSI topologies.

6.3 Voice Encoders

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

6.3.1 Optional Voice Encoders

This implementation profile considers other voice encoders. To minimize the need for transcoding, the following codecs, though not required, are RECOMMENDED: G.711 A-Law, GSM 06.10 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 (PCMU) 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 costs low, it is not feasible to come up with a list of only free codecs allowed 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

Tandeming refers to connecting multiple voice encoders back-to-back. 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 Media Forwarding for Multiple Call Sessions

To reduce the need for a large mesh of interconnection links in more complex bridging topologies, multiple BSIs can connect to the same resource on a given BSI, as shown in

Figure 2 of Section 2. Support for multiple BSIs connecting to the same resource is OPTIONAL in BSI-Core. But if a BSI accepts multiple call sessions to the same resource, it MUST implement the Media Forwarding requirements in this section.

With multiple call sessions to the same resource, a BSI MUST forward media received from one bridge to the other bridges connected to it. For example, B-2 in Figure 2 forwards media received from B-4 and B-5 as well as media received from its own donor radio on to B-1. BSIs that forward incorrectly configured media may generate media loops, as shown in Figure 3.

BSIs that support call sessions to more than one other BSI (e.g., B-1 and B-2 in Figure 2) MUST implement media loop detection as described in Section 8.2 of RFC 3550 [3]. Such a BSI SHOULD implement the loop detection algorithm described in Section 8.2 of RFC 3550 [3] or MAY implement an alternate algorithm that provides loop detection. Loop detection is a standard requirement on RTP implementations per RFC 3550, but it is worth mentioning specifically since loops may be a problem with complex BSI topologies.

A BSI that receives it own transmissions MUST break the loop and MUST NOT forward that media to other bridges and MUST NOT send that media to its donor radio. This will prevent audio loops from disrupting operations. A BSI that detects a loop SHOULD alert its operator to correct the topology.

It is possible to receive media from multiple bridges simultaneously when a BSI connects to more than one other BSI. When a BSI is handling media from multiple source bridges, it MAY select to forward only the media from a single source or MAY sum or mix audio from multiple other BSIs and allow for user resolution of transmission conflicts. Passing of multiple distinct media streams to the attached radio is outside the scope of this profile. A BSI SHOULD only mix audio when using a codec that works well with multiple speakers, such as G.711. A BSI that sums or mixes audio forwarded to other BSIs MUST follow the requirements and guidelines of mixers in Section 7 of RFC 3550 [3]. BSI-Core does not specify how a bridge should handle audio collisions.

6.5 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. To comply with industry convention, it is RECOMMENDED to use 101 as the payload format; however, other dynamic payload formats MUST be supported.

A BSI MUST support at least events 0-15 (the DTMF events) 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 the various radio interfaces of the bridging system will detect and transmit the DTMF 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.

6.6 Additional Requirements and Recommendations

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

6.6.1 Protocol Version ("v=" line)

The "v=" line gives the version of the SDP. 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
```

6.6.2 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. For example:

```
o=fire1 2890844526 2890844526 IN IP4 192.11.11.111
```

6.6.3 Session Name ("s=" line)

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

```
s=-
```

6.6.4 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. For example:

```
c=IN IP4 192.22.22.222
```

6.6.5 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
```

6.6.6 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 use 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 defined 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 06.10 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. The actual codec that a dynamic payload type represents is defined using an a=rtpmap: line (Section 6.6.7).

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 a 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 formats, a BSI MUST be able to respond with an answer accepting that payload format.

6.6.7 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 06.10 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
```

Different clock rates or packet times 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 (CN) generation (CNG). BSI implementations are required to not send VAD or excessive CNG packets (see Section 10); therefore BSI implementations MAY 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 the BSI-Core profile, it is assumed that all SDP negotiations are implicitly half-duplex. This document anticipates that future phases of this profile may provide mechanisms for requesting half-duplex explicitly.

6.7 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 the required and many of the recommended codecs with all the default values specified explicitly for illustrative purposes.

```
v=0
o=fire1 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=fire1 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 consider the jitter and packet loss as well. This document does not attempt to dictate all the requirements of the network layer in terms of bandwidth and supported services. However, it defines basic network transport mechanisms for the signaling and media layers, including a recommendation for IP layer packet marking of the media packets. The document addresses schemes and naming conventions, as well as Network Address Translation (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 focus on TCP and drop UDP considerations.
- TCP lays the foundation for using transport layer security (TLS), which is a widely supported mechanism for securing communication and may be used in future phases of implementation profile to secure the signaling transport layer.

Explicitly note the use of TCP 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: 178
```

```
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 to maintain the TCP connections between BSIs 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 necessary. 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.

To maintain TCP connections, implementations should support industry common practice NAT and firewall traversal mechanisms. Since Best Practices in this area is under flux, no specific mechanism is required for this profile.

7.2 Media Transport Layer

As specified in Section 6.6.6, the REQUIRED media layer transport protocol is RTP/AVP. 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 RFC 5285 [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 to deliver the packets required to construct the media streams at either end of the BSIs in a timely fashion to avoid choppy or unintelligible speech. To aid in the effort, the BSI-Core implementation profile RECOMMENDS support for service marking as described in RFC 2475 [17] as amended by RFC 3260 [26].

The network(s) that interconnect the BSIs SHOULD provide support for VoIP traffic as a per-domain behavior as described in RFC 3086 [27].

For networks that support Differentiated Services, for all media packets it is RECOMMENDED the BSI set the Differentiated Services Field (DSField) as defined in RFC 2474 [32] (previously the six most significant bits of the former IPv4 TOS octet) to a Diffserv Codepoint (DSCP) value that requests Expedited Forwarding (EF) Per-Hop Behavior (PHB) as defined in RFC 3246 [27]. Note that requesting EF PHB for the media packets is a recommendation for the BSI application, and supporting the EF PHB is a recommendation for the network. Since the EF DSCP value MAY be different for different DS domains, the EF DSCP value SHOULD be configurable.

For networks that do not support Differentiated Services, it is RECOMMENDED the BSI set the former IPv4 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 the implementation profile. Although there are other schemes available, such as IPv6, these are not as ubiquitous as IPv4.

Support for IPv6 is a subject for future study.

7.4 Naming Conventions

SIP URIs identify resources within the SIP domain. 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>, but this name should be unique within a jurisdiction and, if possible, descriptive of the resource being connected. While resources can be more than simply radio channels (e.g., telephones, talkgroups), the NCC/NPSTC Standard Channel Nomenclature for the Public Safety Interoperability Channels [33] can be used as a model for creating concise yet descriptive resource names.

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. Note that as required in RFC3261 [1], implementations MUST support escaping of special characters in the URI.

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 postpones detailed recommendations within this area. 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, any solution for BSI interoperability must take high availability into account. However, recommendations for network and system design to achieve this are outside of the scope of this document. The Department of Homeland Security, Office for Interoperability and Compatibility (DHS/OIC) will publish a Best Common Practices document regarding this subject 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 Secure Real-Time Transport Protocol to secure the media [14]. Standard Development Organizations (SDOs) have defined numerous other security mechanisms for SIP, or are in the process of defining them. However, for the BSI-Core implementation profile, it is REQUIRED that BSIs be able to interoperate in the absence of such security mechanisms. The IP network connection between BSIs is assumed secure through mechanisms such as IPSec [15], VPNs [16], SEGs [24], etc.

9 Management

This specification recognizes the need for management for BSI interoperability; however, the document considers management out of scope for BSI-Core. This document assumes that agencies wish to have their BSIs interwork exchange corresponding IP addresses and resource names, and they enable SIP signaling and media traffic between their BSIs at their own discretion. The profile leaves mechanisms, such as the exchange of certificates for authentication and the use of DNS and SIP registrars for registering and locating resources, for future phases of the implementation profile.

The following table provides a summary of the BSI-related information that agencies exchange for BSI-Core.

	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 a 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 BSI-Core implementation profile is the presence or lack of audio packets within the media stream established by the SIP call session. The BSI MUST support voice packet detection. This REQUIRES that RTP packets NOT be sent representing silence except as defined below.

If the receiving BSI has negotiated willingness to receive comfort noise packets, either explicitly or implicitly by using a codec that includes voice activity detection, the BSI MAY send at most three comfort noise packet at the end of a PTT transmission when it detects the lack of audio from its donor radio. The receiving BSI MUST deal with possible out-of-order arrival of the comfort noise and delayed media packets representing silence.

If sending comfort noise when using a codec that does not support comfort noise encoding, such as G.711, the BSI MUST use the comfort noise codec defined in RFC 3389 [25]. If the BSI is sending a comfort noise packet per RFC 3389, the offer/answer mechanism MUST have negotiated the CN payload using during call session establishment; as with any optional codec, the receiving BSI may not support the CN payload and may reject it during offer/answer negotiation. When using a codec that supports comfort noise encoding, such as AMR or G.729 Annex B, the BSI SHOULD use the codec's CN signaling.

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 this implementation profile REQUIRES support for RTCP is to detect the loss of media within a SIP call session. Given the PTT nature of the media streams, it is possible that there is no exchange of RTP packets for long periods of time. Therefore, (--) MUST send periodic RTCP packets 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 the BSIs detect a loss of RTP/RTCP packets (e.g., 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. To achieve this, a re-INVITE with SDP is sent with the same session version as the original SDP sent for that call session. If the session still exists on the remote BSI, it SHOULD respond with a 200

OK 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. 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 1.0.5 to 1.1

- Removed %20 from the examples
- Added recommendations on some SIP error codes

12.2 Changes from version 1.0.4 to 1.0.5

- Fixed problems with Figures printing correctly
- Added Figure 4 showing RF loop problem
- Changed examples to use NIMS recommended channel IDs per the Best Practices document
- Corrected content-length values

12.3 Changes from version 1.0.3 to 1.0.4

- Explicitly mention support for character escaping
- Example shows compact form of SIP header fields
- Consistently used "call session" terminology
- Added requirements related to multiple connections and media forwarding
- Added text talking about loosely-coupled conferences

12.4 Changes from version 1.0.2 to 1.0.3

- Updated example in 5.5.2 to use GSM 06.10 codec

12.5 Changes from version 1.0.1 to 1.0.2

- Corrected some RFC references
- Allow up to three comfort noise packets to be in line with RFC 4733 practices

12.6 Changes from version 1.0 to 1.0.1

- Added section on media forwarding
- Clarified use of comfort noise packets
- Added use of DSCP in lieu of TOS
- Added recommended Annex B be implemented if doing G.729
- Added use of display name in "From:" header
- Added Figs 2 & 3 and describing text

12.7 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.

12.8 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

12.9 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

12.10 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

12.11 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

12.12 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", BCP 75, 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", RFC 5285, July 2008.

[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.

[24] "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Network Domain Security; IP network layer security (Release 8)", 3GPP TS 33.210 V8.0.0, March 2008.

[25] R. Zopf, "Real-time Transport Protocol (RTP) Payload for Comfort Noise (CN)", RFC 3389, September 2002.

[26] Grossman, D., "New Terminology and Clarifications for Diffserv", RFC 3260 April 2002.

[27] Davie, B., A. Charny, J.C.R. Bennett, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", RFC 3246, March 2002.

[28] Nichols, K. and B. Carpenter, "Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification", RFC 3086, April 2001.

[29] Sjoberg, J., Westerlund, M., Lakaniemi, A., Xie, Q., "RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs", RFC 4867, April 2007.

[30] Department of Homeland Security, Office for Interoperability and Compatibility (DHS/OIC), "Bridging Systems Interface (BSI) Best Practices", April 2010.

[31] Department of Homeland Security, Office for Interoperability and Compatibility (DHS/OIC), "Bridging Systems Interface (BSI) Best Practices, Appendix C: Tanker Truck Rollover Scenario", April 2010.

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

[33] National Public Safety Telecommunications Council, "NCC/NPSTC Standard Channel Nomenclature for the Public Safety Interoperability Channels", June 2009.