The SIPREC protocol defines the interaction between a Session Recording Client (SRC) and a Session Recording Server (SRS), and controls the recording of media transmitted in the context of a communications session (CS) between multiple user agents.

SBC 5000 series supports the following SIP recording (SIPRec) interfaces:

- SIPRec SIP-based session recording
- Call monitoring MCT
- NICE session recording

A basic call, or communication session (CS), is established between SIP phone 1 and SIP phone 2 through SBC. The SBC 5000 Series establishes a Recording session (RS) for the CS based on configuration towards SRS.

The SRC and SRS act as SIP User Agents (UAs) whereby the SRC provides additional information (called metadata) to the SRS recording session to describe the communication sessions, participants and media streams to facilitate archival and retrieval of the recorded information.

The SBC 5000 Series determines if a call is configured to be recorded, and if SIPREC is to be used.

**SIPRec Functionality**

For every call, SBC 5000 Series performs a PSX/ePSX query to determine whether the call needs to be recorded based on pre-configured criteria. If the call needs to be recorded, the query result will indicate the type of recording along with the Recorder (SRS) information.

If a call is classified as to be recorded using SIPREC, SBC 5000 Series initiates a Recording Session towards the SRS with an INVITE request to the SRS IP address/port pair along with the metadata giving the details of the Communication Session. SBC uses the same Codecs that are used in the CS towards the SRS. There is no transcoding supported for recording streams.

The SRS can choose to record or not record or terminate the incoming Recording Session. If the SRS determines that recording is not needed but still wants to retain the RS session, it sends a SDP answer with the mode set to “inactive”.

If RS Session needs to be terminated for any reason by the SRS, it can send a BYE and terminate RS. CS will continue uninterrupted.

Any updates to the call/media attributes during midcall are updated to the SRS using a RE-INVITE. The SDP indicates the new media attributes, and the metadata indicates the call attributes. SBC always sends complete metadata in the RE-INVITEs. Partial updates are not supported.

SBC can virtually support recording all CS sessions in the system. But each RS session is also counted as a call, thus the total number of calls in the system are adjusted accordingly.

**Supported Features**

1. SBC supports recordings initiated from the SRC based on the policy (PSX/ePSX)
   a. Recording can also be initiated by CLI providing GCID and other details.
   b. Recording can be stopped from CLI.
2. SIPREC recording is supported for SIP and SIP-I calls.
3. For calls recorded using SIPREC, the following is supported by SBC:
   a. RTCP and RTP are transparently muxed towards the SRS
   b. Adding feature tag ‘+sip.src’ in Contact URI of Recording session (RFC 3840)
   c. Addition of ‘siprec’ option tag in Recording session
   d. Addition of SDP label attribute (RFC 4574) in the SDP towards Recording session
   e. A=record attributes support
   f. Forming MIME body with application/rs-metadata
   g. Basic elements (mandatory XML elements like recording, datamode, session, participant, stream) in the application/rs-metadata

The SIPRec feature is controlled by a system-wide SBC license (SBC-SIPREC). If the license is not available, any SIPREC recording returned by a PSX is ignored.
4. SBC supports following redundancy:
   a. SRS redundancy – active/standby SRS with the recording call reestablished towards standby after active is down
   b. SRC redundancy as with any normal calls.
   c. If Options ping mechanism is configured on the IP peer, it is used as a keep-alive mechanism for all the recording sessions.
      Otherwise, normal SIP keep alive mechanisms are used.
5. SIPRec feature supports configuring up to 128 Recording criteria and 128 Recorder profiles.
6. Codec negotiation towards SRS is not supported. If the codec is not supported by SRS, it is assumed that it will terminate the RS.
   a. The RS streams use the same codec as the CS.
   b. No support for separate transcoding of voice streams towards SRS
   c. No support for mixing the streams towards the SRS. The user streams will be transmitted as-is towards the SRS separately.
7. RTCP feedback mechanisms/bandwidth negotiation and packet-loss handling towards the SRS is not supported.
   a. RTCP reports from the SRS are not sent to the UE.
8. No support for “recording-aware” UEs.
9. Persistent recording is not supported.
10. Registration of SRC to SRS is not supported.
11. Video calls recording not supported.
12. SRTP streams will always be decrypted and sent towards the SRS.
13. ERE support is not available for SIPREC.
14. “Call-Recorded” indication is marked in the STOP CDRs along with the SRS information.
15. SRS-initiated calls are not supported.

**Feature Interactions**

When considering which combinations of NICE recording, MCT, Lawful Intercept, Call Trace and SIPRec are supported, the Priority order is the following:

1. Lawful Intercept
2. Other recording (NICE, Veriant or SIPREC)
3. MCT
4. Call-media-trace

The following conditions apply to the above features:

- LI and SIPREC can be used simultaneously on a call if LI is using MCAST. However, if LI uses Splitter, SIPREC config is ignored by the SBC.
- Old NICE and SIPREC recordings cannot co-exist because both use Splitter. If SIPREC recording is in progress, NICE INVITE generally comes late (after the call is answered), thus will get rejected by SBC if other is present.
- MCT and SIPREC use the same trigger logic from PSX. PSX returns the recorder profile of the best matched criteria. When matching criteria, SIPREC is given the highest score. So if both SIPREC and MCT are configured, PSX returns the recorder profile of SIPREC. So, if a SIPREC call has to be debugged, MCT has to be triggered by CLI using GCID and in this case MCT uses MCAST.
- Call trace with Packet capture feature will stop when any of the other features are being used.

**Figure 1:** SipRec - Start Recording
Figure 2: Siprec - Stop Recording

Table 1: Siprec - Start Recording Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gcid</td>
<td>Specifies the GCID of call to start recording.</td>
</tr>
<tr>
<td>Call Leg</td>
<td>Call leg associated with this recording session.</td>
</tr>
<tr>
<td></td>
<td>• egress</td>
</tr>
<tr>
<td></td>
<td>• ingress (default)</td>
</tr>
<tr>
<td>Srs Ip Address</td>
<td>Specifies the IP address of the Recording Server.</td>
</tr>
<tr>
<td>Srs Port</td>
<td>Specifies the UDP Port of the Recording Server.</td>
</tr>
<tr>
<td>Trunk Group</td>
<td>Specifies the name of the siprec trunk group associated with this recording session.</td>
</tr>
</tbody>
</table>
Table 2: Siprec - Stop Recording Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gcid</td>
<td>Specifies the GCID of call to stop recording.</td>
</tr>
</tbody>
</table>