Configuring SBC for WRTC

In this section:

Related articles:

- Configuring ICE and DTLS Direct Media Call
- Configuring ICE and DTLS for Late Media Calls

Overview

Sonus WebRTC Gateway (WRTC) is a new technology that enables web browsers to participate in audio, video, and data communications, without any kind of additional plug-ins or application downloads. Using a WRTC enabled browser user can place a call, participate in multi-party video and audio conferencing, and engage in screen sharing collaboration. Sonus Web Service Solution bridges the web and SIP worlds to facilitate the integration of communications (voice, video, and data) in applications.

Sonus SBC is a component of Sonus Web Service Solution. Sonus SBC provides media service functionality when WRTC endpoints are behind a NAT.

Sonus SBC acts as a WRTC to SIP media gateway. It enables WRTC users to communicate to any back-end SIP system and PSTN. Sonus SBC also provides routing, security, transcoding, and interworking. It supports the following functionalities:

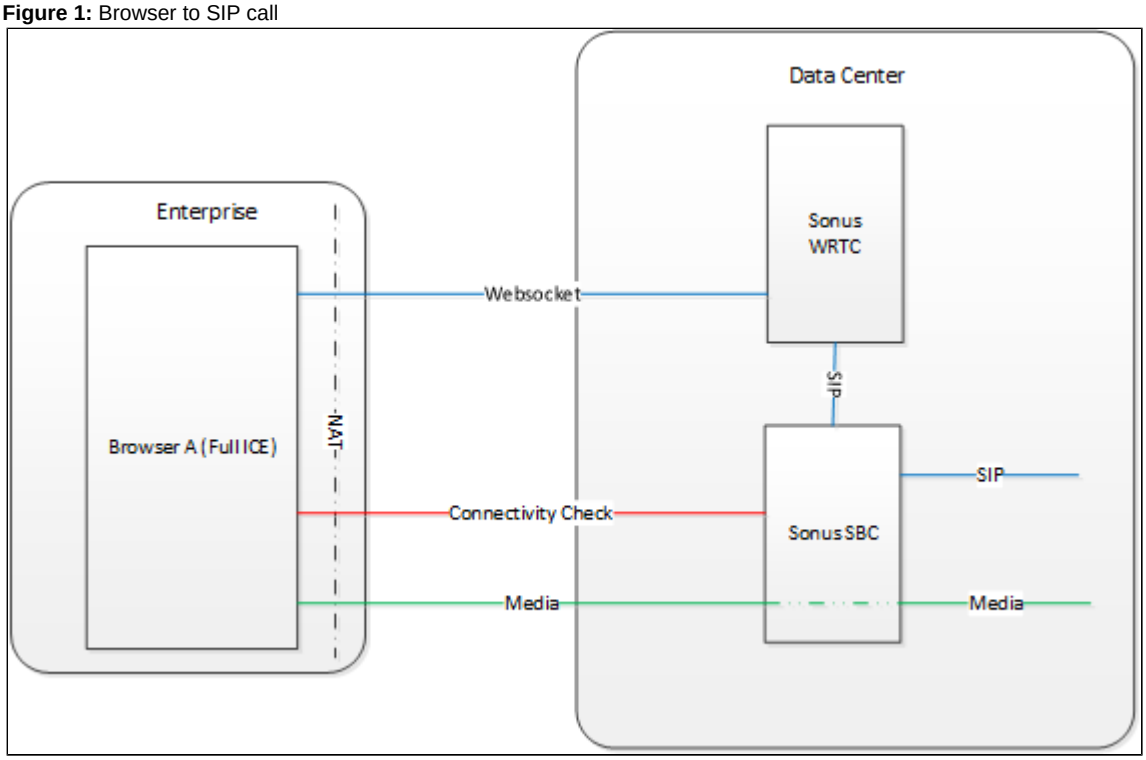
- Relays and monitors the media streams.
- Inter-works WRTC media DTLS/SRTP to traditional RTP/UDP.
- Relays or transcodes opus to G7xx voice codecs.
- Relays VP8/VP9, and H.264 video codecs.
- Supports ICE and STUN procedures for NAT traversal.

Deployment Scenarios

WRTC Enabled Device to SIP Call (SBC in Data Center)

The WRTC enabled device employs the ICE procedures and connects to the SBC on a public address. The SBC acts as an ICE agent to support the WRTC enabled device to punch the pinholes in the NAT for media exchange with the SBC. This can work with any Firewall in front of the WRTC enabled device that can support opening NAT Pinholes for the UDP traffic. The NAT can be Full-Cone, restricted, or symmetric NAT.

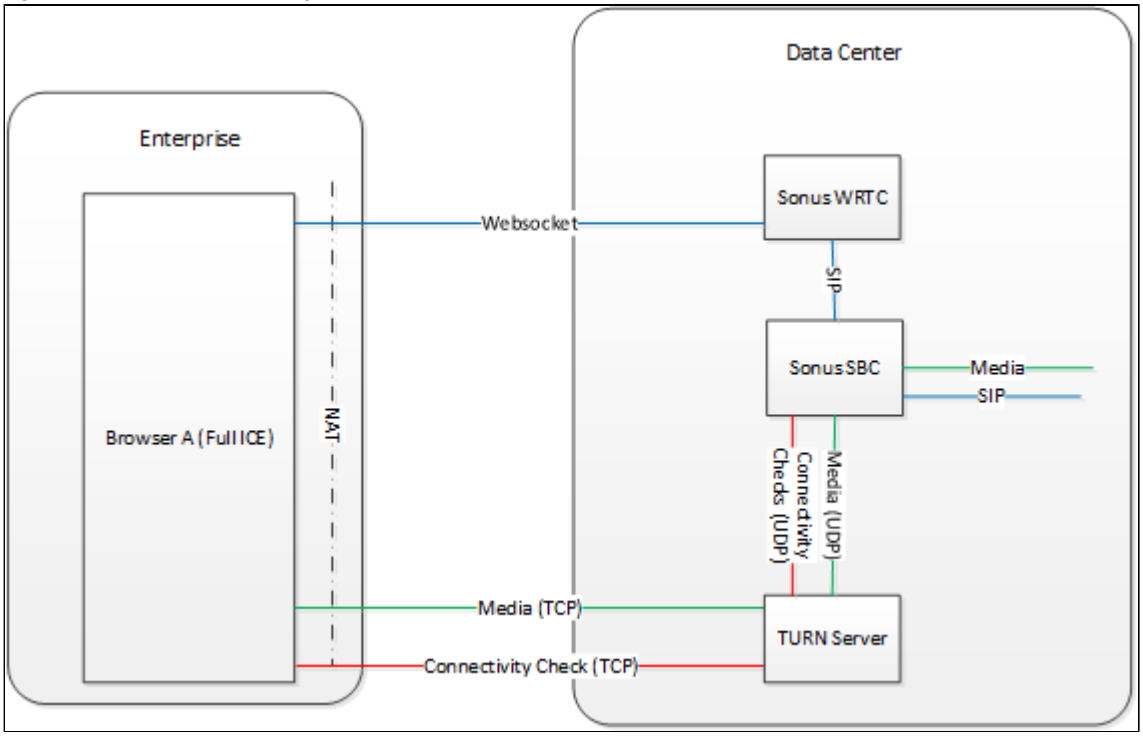
Figure 1: Browser to SIP call



WRTC Enabled Device to SBC Through TURN Server

In this case, media is exchanged between the WRTC enabled device and the SBC. The ICE mechanism is used to negotiate a relay address for the firewalls in front of the WRTC enabled device to use for media exchange over TCP or http ports. A TURN relay is used with media path to convert RTP/TCP to RTP/UDP towards SBC.

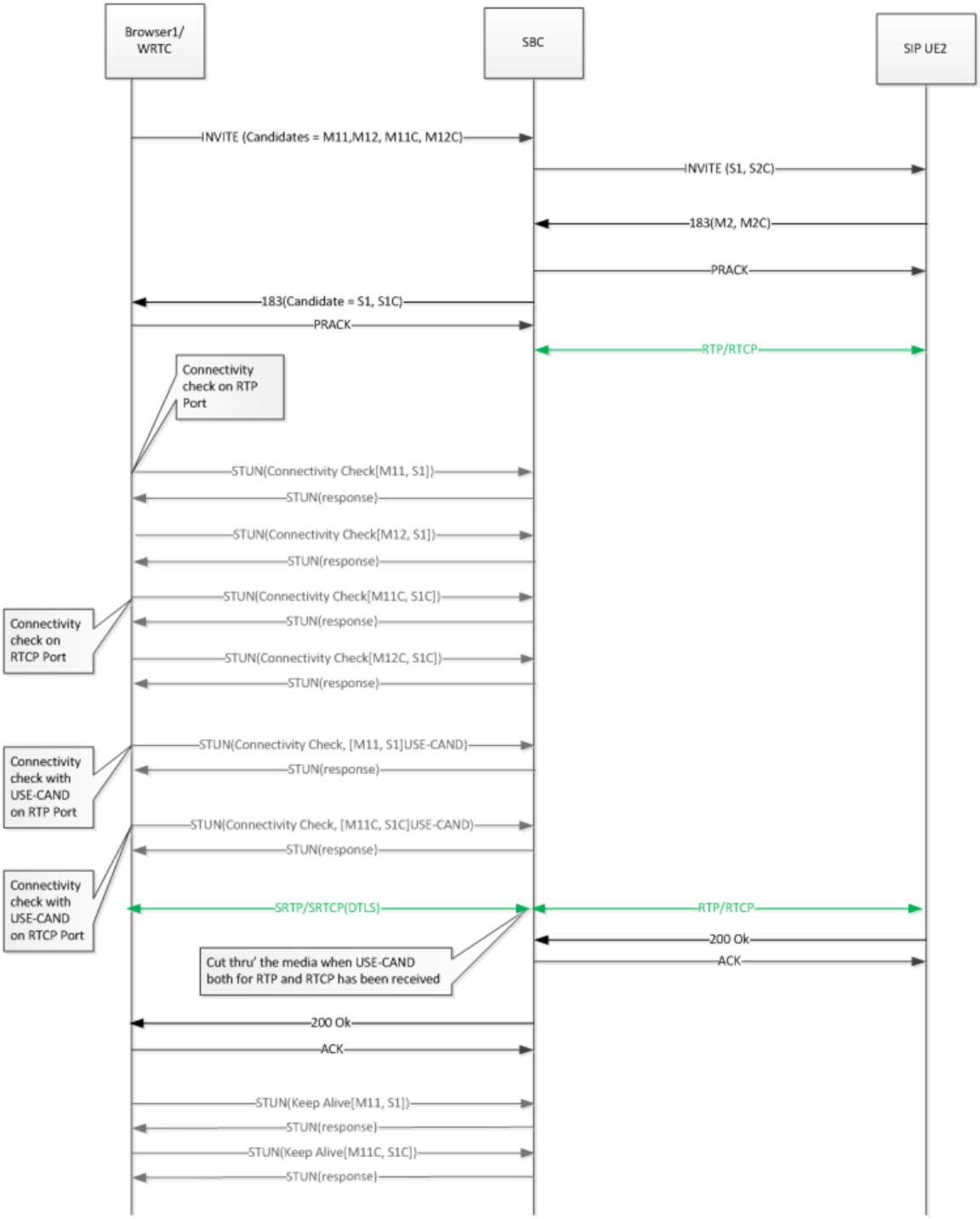
Figure 2: Browser to SBC through TURN server



Call Flows

Basic call (Full ICE to No ICE)

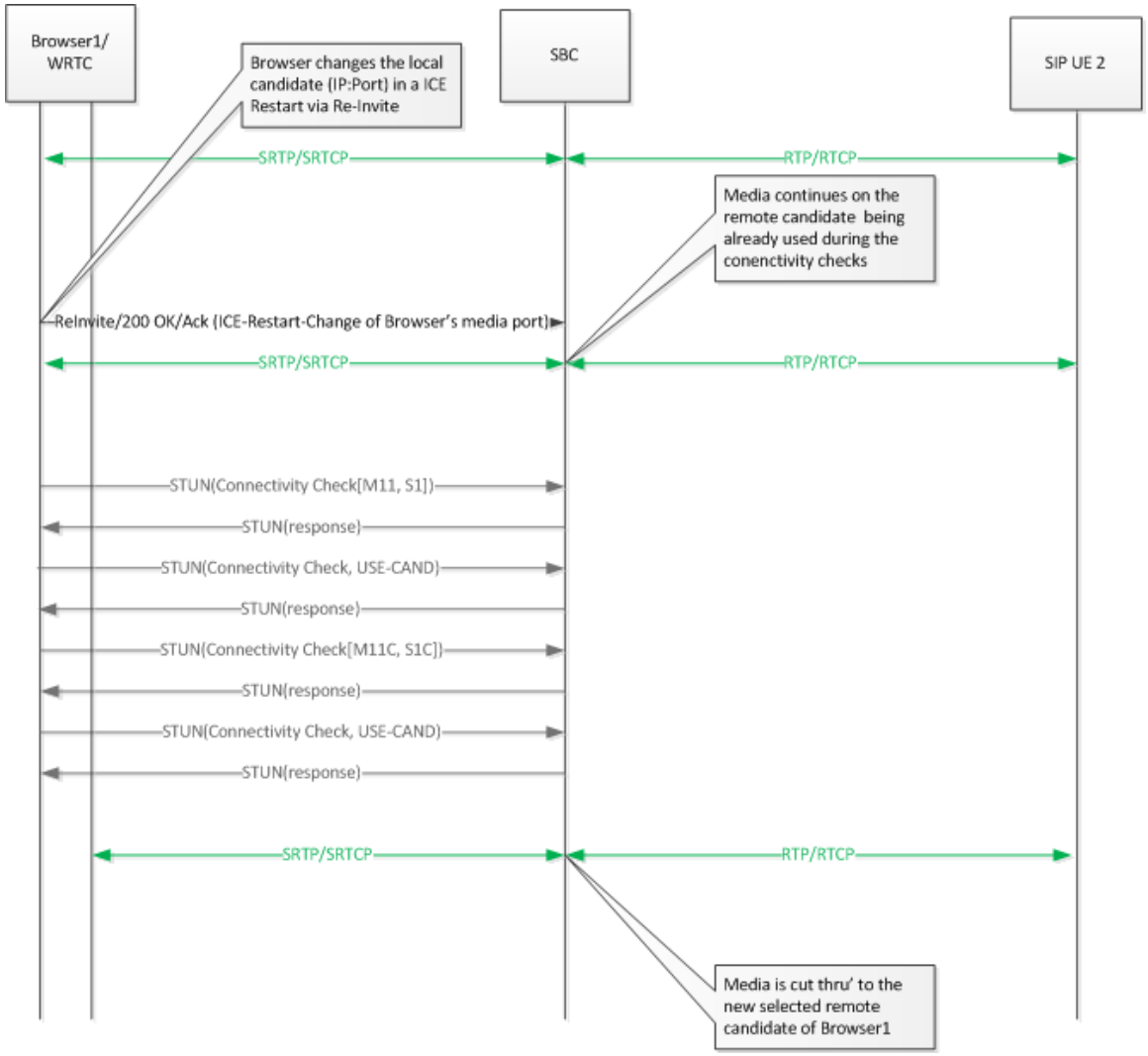
Figure 3: Basic Call between UE supporting ICE and no ICE



- M11 - RTP Sever Reflexive candidate
- M12 - RTP Host candidate
- M11C - RTCP Sever Reflexive candidate
- M12C - RTCP Host candidate

Mid Call ICE Restart

Figure 4: Mid call ICE restart



Configuring WRTC includes:

- Configuring ICE
- Configuring DTLS-SRTP
- Supporting Opus Codec

Configuring ICE

When `natTraversal` is set for `iceSupport`, it is recommended that both `mediaNat` or `secureMediaNatPrefix` are not configured.

To configure ICE for a WRTC call:

- SIP Trunk Group Configuration
- Configuring Relay SDP Parameters
- SDP Method for Multiple IP Version
- Policing Logic for STUN Packets

SIP Trunk Group Configuration

The ICE capability is enabled on the trunk group towards the WRTC endpoints:

```
set addressContext default zone ZONE_WRTC sipTrunkGroup TG_SIPART_WRTC services natTraversal
iceSupport iceFull
```

- SBC uses `iceFull` to support faster completion on the ICE exchange as the two end points locks down on the first accessible connection path attempted.
- SBC uses `iceWebrtc` to allow selection of the optimum connection path, for example, Host vs TURN address.

Configuring Relay SDP Parameters

The `sdpAttributesSelectiveRelay` control must be enabled to support WSX-SBC-WSX call scenarios.

```
set addressContext default zone ZONE_WRTC sipTrunkGroup TG_SIPART_WRTC media
sdpAttributesSelectiveRelay enabled
commit
```

SDP Method for Multiple IP Version

To configure the SDP method, ICE support must be enabled first.

```
set addressContext default zone ZONE_WRTC sipTrunkGroup TG_SIPART_WRTC media mediaAddrType
iPv4andiPv6 ice <offerPreference | answerPreference>
set addressContext default zone ZONE_WRTC sipTrunkGroup TG_SIPART_WRTC media mediaAddrType
iPv4andiPv6 ice offerPreference <ipv4 | ipv6 | matchSigAddrType>
set addressContext default zone ZONE_WRTC sipTrunkGroup TG_SIPART_WRTC media mediaAddrType
iPv4andiPv6 ice answerPreference <honorRecvPrec | ipv4 | ipv6 | matchSigAddrType>
```

For detailed information on iPv4 and iPv6 CLI changes, refer to SIP Trunk Group - Media - CLI.

Policing Logic for STUN Packets

When policing is enabled, SBC uses the following prefix lengths to screen the packets that are received from the network. IP addresses that match are allowed to be processed at a higher frequency than IP addresses that do not match.

- RTP IPV6 Host Address - Hard-coded 128 bit prefix
- RTP IPV4 TURN Address - Hard-coded 32 bit prefix
- RTP IPV6 TURN address - Hard-coded 128 bit prefix
- RTP IPV4 Server Reflexive address - Prefix based on the provisioned length

If policing is disabled, all the packets are treated at the lower frequency of processing and can be dropped if there is an excessive amount of traffic received.

```
set addressContext default zone ZONE_WRTC sipTrunkGroup TG_SIPART_WRTC services natTraversal
iceSourceAddressFilterPriority <serverReflexivePrefixLength  | state>
set addressContext default zone ZONE_WRTC sipTrunkGroup TG_SIPART_WRTC services natTraversal
iceSourceAddressFilterPriority serverReflexivePrefixLength  <0..32>
set addressContext default zone ZONE_WRTC sipTrunkGroup TG_SIPART_WRTC services natTraversal
iceSourceAddressFilterPriority state <enabled | disabled>
```

The aggregate policer screen shows information about the number of STUN packet accepts and discards that have occurred for a given address context. The command for aggregate policer is :

```
show table addressContext default ipAccessControlList getAggrPolicers

POL  POLICING  ZONE  POLICING                                  PACKET  PACKET   AGG POL
ID   TYPE      ID    MODE      BUCKET SIZE  CREDIT RATE       ACCEPT  DISCARD  NAME
-------------------------------------------------------------------------------------
0    Link      -     DataRate  300000 byte  62500000 byte/s   0       0        LINK_pkt0
1    Link      -     DataRate  300000 byte  62500000 byte/s   0       0        LINK_pkt1
4    StunDtls  -     PktRate   100 pkt      10000 pkt/s       0       0        STUN
5    StunDtls  -     PktRate   100 pkt      10000 pkt/s       0       0        DTLS
```

Configuring DTLS-SRTP

- If the latest developer version of "Firefox" is used, additional configuration is required to correct the following error:
  091 09042015 115022.824913:1.01.00.21882.MAJOR .DTLS_SRTP: *DTLS Error no shared cipher
  Enter the following command to correct the error:

```
config
set profiles security dtlsProfile defaultDtlsProfile cipherSuite2
tls_ecdhe_rsa_with_aes_128_cbc_sha
commit
```

- The DTLS-SRTP and SCTP relay controls must be enabled on the Packet Service Profile for the end-to-end DTLS handshake for WSX-SBC-WSX call flows.

Using the Default DTLS Profile

The default DTLS profile is already present when the system is up and can be used to run WRTC calls.

```
show profiles security dtlsProfile defaultDtlsProfile
handshakeTimer 60;
sessionResumpTimer 300;
cipherSuite1      rsa-with-aes-128-cbc-sha;
dtlsRole          server;
hashType          sha1;
CertName          defaultDtlsSBCCert;
cookieExchange    enabled;
v1_0              enabled;
v1_1              disabled;
v1_2              disabled;

[ok]
```

- In this example, the call setup time to establish a SIP call from a mobile phone may be longer, so the DTLS `handshakeTimer` is set to 60 seconds.

- For special configuration requirement in the DTLS profile, the default DTLS profile can be modified or a a new DTLS profile can be created. For details, refer to the section Creating the DTLS Profile.

Creating the Default DTLS Certificate

In case of an upgrade, if the certificate `defaultDtlsSBCCert` is not present by default, it must be created and enabled before adding it to the DTLS profile.

To check the availability of the certificate `defaultDtlsSBCCert`, enter the following command:

```
show configuration system security pki certificate
certificate defaultSBCCert {
    state      enabled;
    fileName   sonuscert.p12;
    passPhrase $7$D9bBhC0fE+n89v5kimypN4dl1KCGAwRj;
    type       local;
}
certificate defaultDtlsSBCCert {
    state      enabled;
    fileName   defaultDtlsCert.p12;
    passPhrase $7$D9bBhC0fE+n89v5kimypN4dl1KCGAwRj;
    type       local;
}
[ok]
```

To create and enable the certificate `defaultDtlsSBCCert`, enter the following command:

```
set system security pki certificate defaultDtlsSBCCert fileName defaultDtlsCert.p12 type local
passPhrase gsx9000 state enabled
Commit
```

⚠ The file `defaultDtlsCert.p12` must be present while creating the certificate `defaultDtlsSBCCert.`

Creating the DTLS Profile

```
set profiles security dtlsProfile d1 CertName defaultDtlsSBCCert cipherSuite1
rsa-with-aes-128-cbc-sha cipherSuite2 nosuite cipherSuite3 nosuite cookieExchange enabled dtlsRole
server handshakeTimer 5 hashType sha1 sessionResumpTimer 300 v1_0 enabled v1_1 disabled v1_2
disabled
```

Attaching the DTLS Profile to Trunk Group

```
set addressContext default zone ZONE_WRTC sipTrunkGroup TG_SIPART_WRTC media dtlsProfileName d1
```

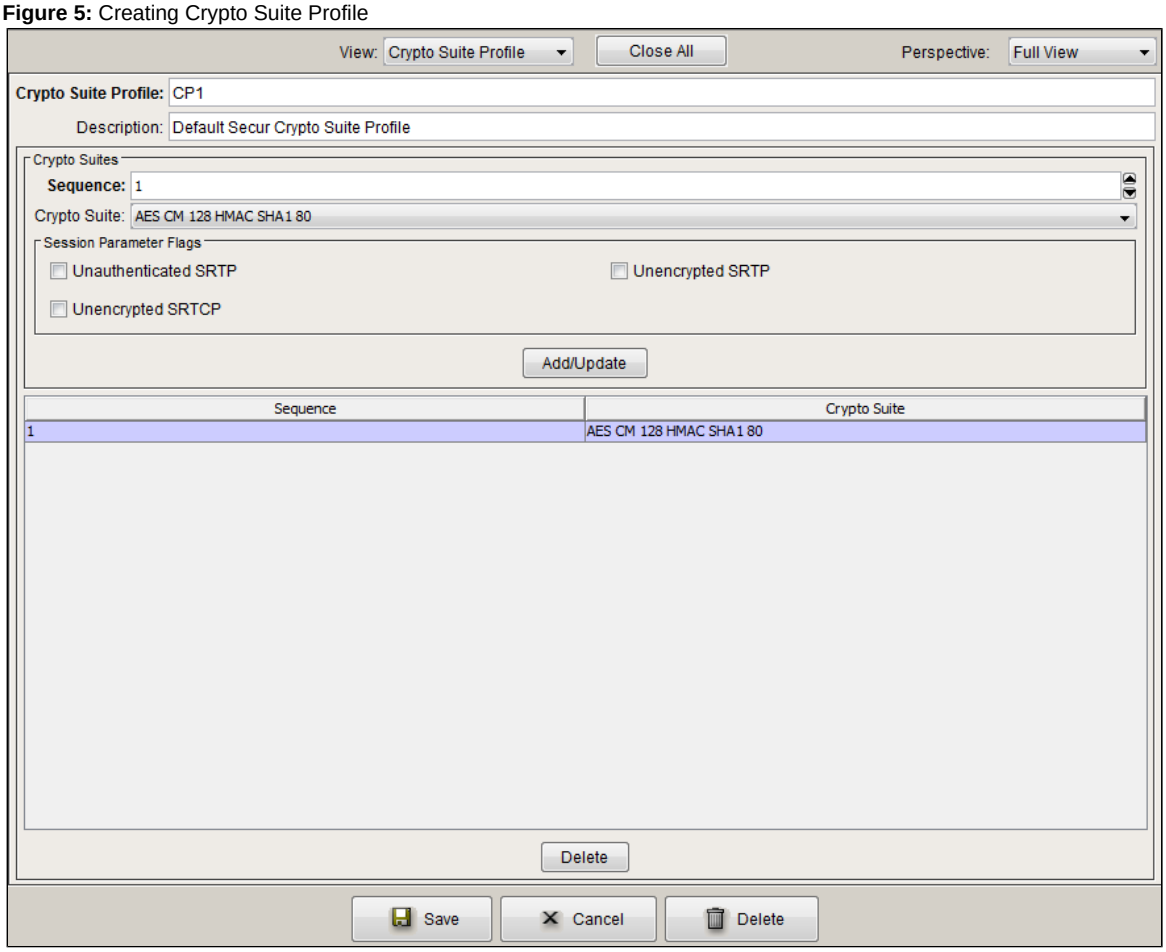
Creating Crypto Suite Profile

On SBC ERE

```
set profiles security cryptoSuiteProfile cp1 entry 1 cryptoSuite AES-CM-128-HMAC-SHA1-80
```

On PSX/external PSX

Figure 5: Creating Crypto Suite Profile



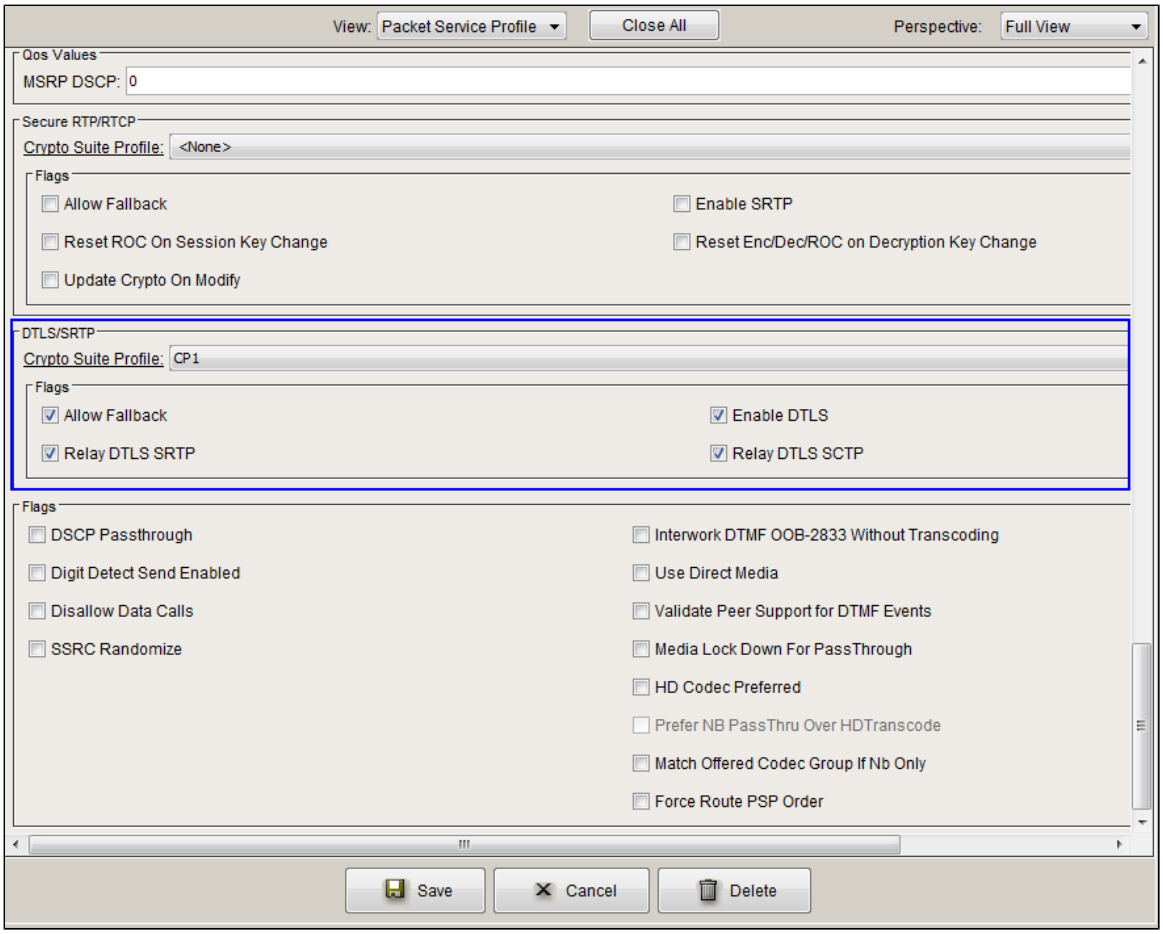
Attaching the Crypto Suite Profile to the Packet Service Profile

On SBC ERE

```
set profiles media packetServiceProfile PSP_WRTC dtls dtlsCryptoSuiteProfile cp1
```

On PSX/external PSX

Figure 6: Attaching Crypto Suite Profile to Packet Service Profile



Enabling the DTLS Crypto Suite Profile Parameters

```
set profiles media packetServiceProfile PSP_WRTC dtls dtlsCryptoSuiteProfile cp1 dtlsFlags
allowDtlsFallback enable enableDtlsSrtp enable
```

The `allowDtlsFallback` parameter enables a fall back to standard RTP when corresponding leg does not have DTLS-SRTP support. If this parameter is disabled, SBC does not allow any other call other than DTLS-SRTP on that leg.

Enabling the DTLS SRTP and DTLS SCTP Relay Flags in Packet Service Profile

```
set profiles media packetServiceProfile PSP_WRTC dtls dtlsFlags dtlsSrtpRelay enable dtlsSctpRelay
enable
```

Attaching the Packet Service Profile to the Sip Trunk Group

```
set addressContext default zone ZONE_WRTC sipTrunkGroup TG_SIPART_WRTC policy media
packetServiceProfile PSP_WRTC
```

The Packet Service Profile can be attached either to the ingress or the egress Sip Trunk Group between WRTC and SBC.

Licensing

The SRTP license must be enabled for DTLS support.

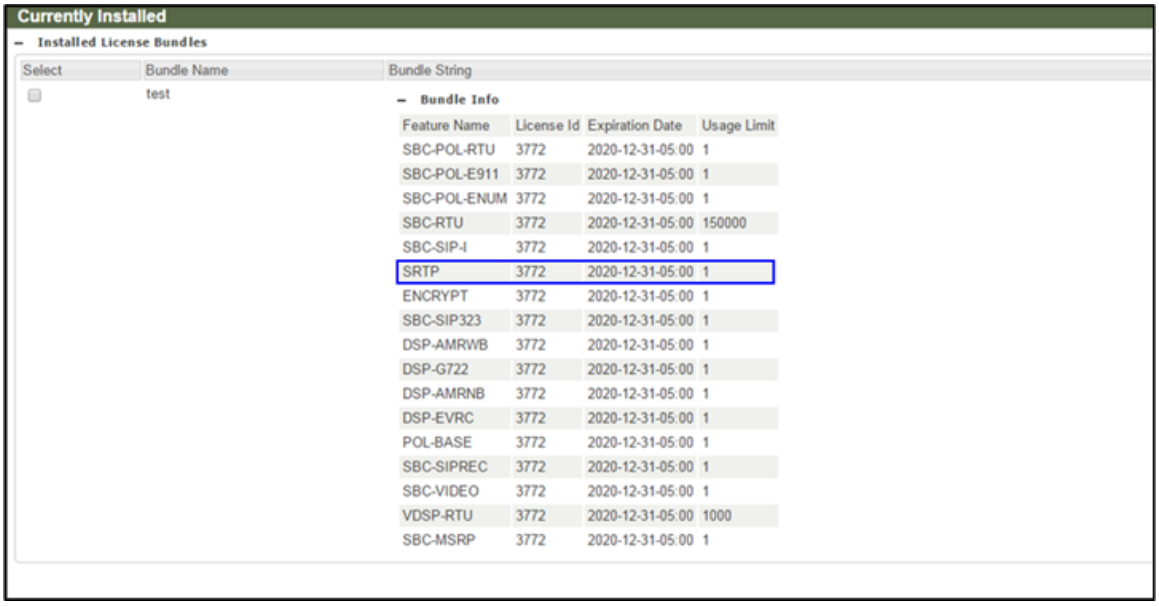
The license can be seen by executing the following command:

```
show table system licenseInfo

LICENSE USAGE
FEATURE NAME ID EXPIRATION DATE LIMIT
```

Navigate to **All > License > Bundle**

Figure 7: SRTP License



Supporting Opus Codec

The newer versions of Chrome browser always offer support for Opus codec when creating WRTC calls. This behavior is not supported by default on the SBC and therefore, SBC removes the codec lines that it understands. However, there are some codec lines, which SBC relays as unrecognized and causes a mismatch of codec information in the SDP and the chrome browser being used for WRTC calls rejects the SDP.

There are two options to resolve this issue:

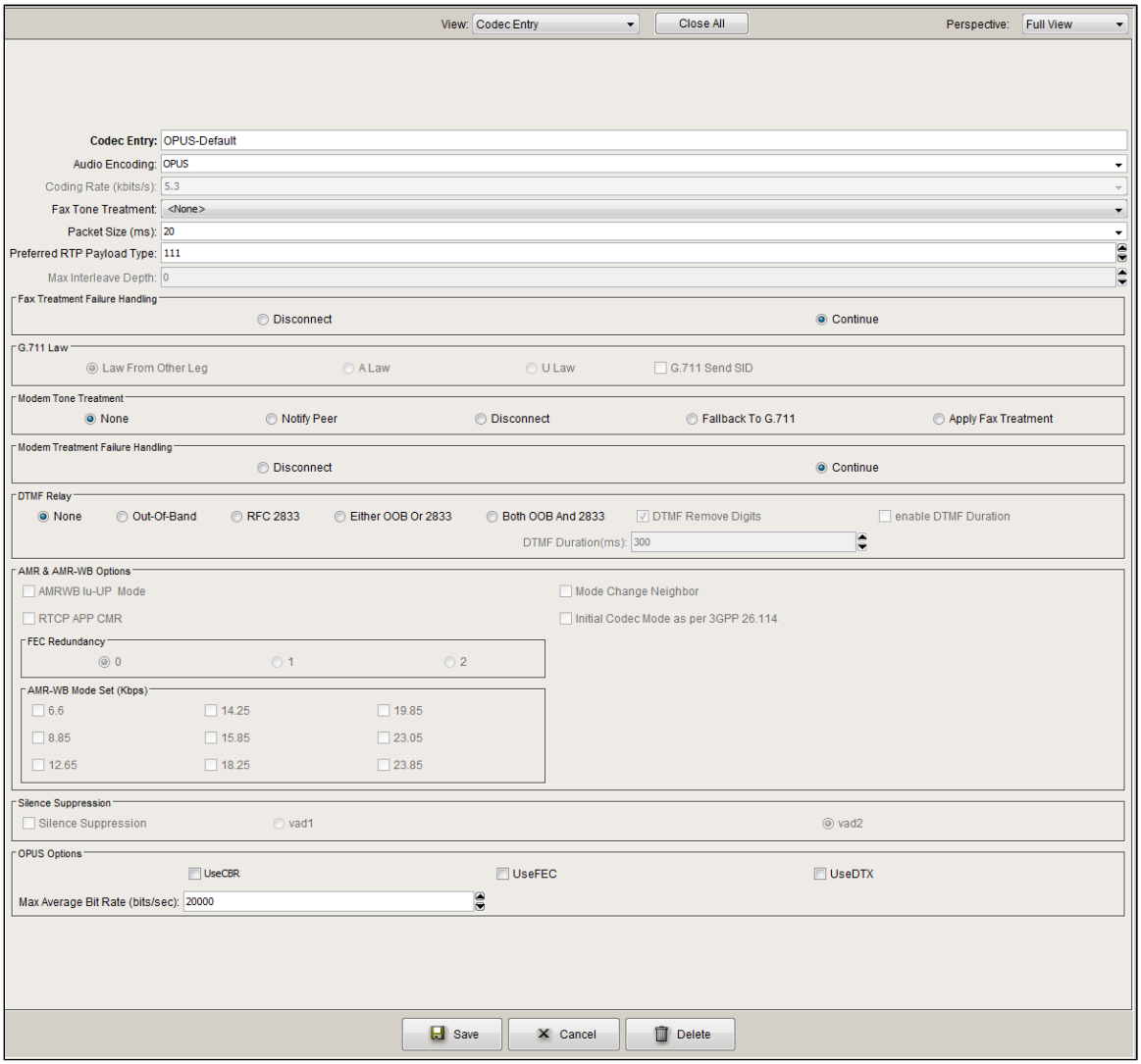
1. Enabling the opus codec in External PSX or ERE
2. Applying SMM Rules to Remove the Unrecognized Codec Lines

Enabling the Opus Codec in External PSX or ERE

- Creating the Codec Entry in External PSX
- Attaching the Codec Entry to the Packet Service Profile in External PSX
  or
- Creating Codec Entry in ERE
- Attaching Codec Entry to PSP in ERE

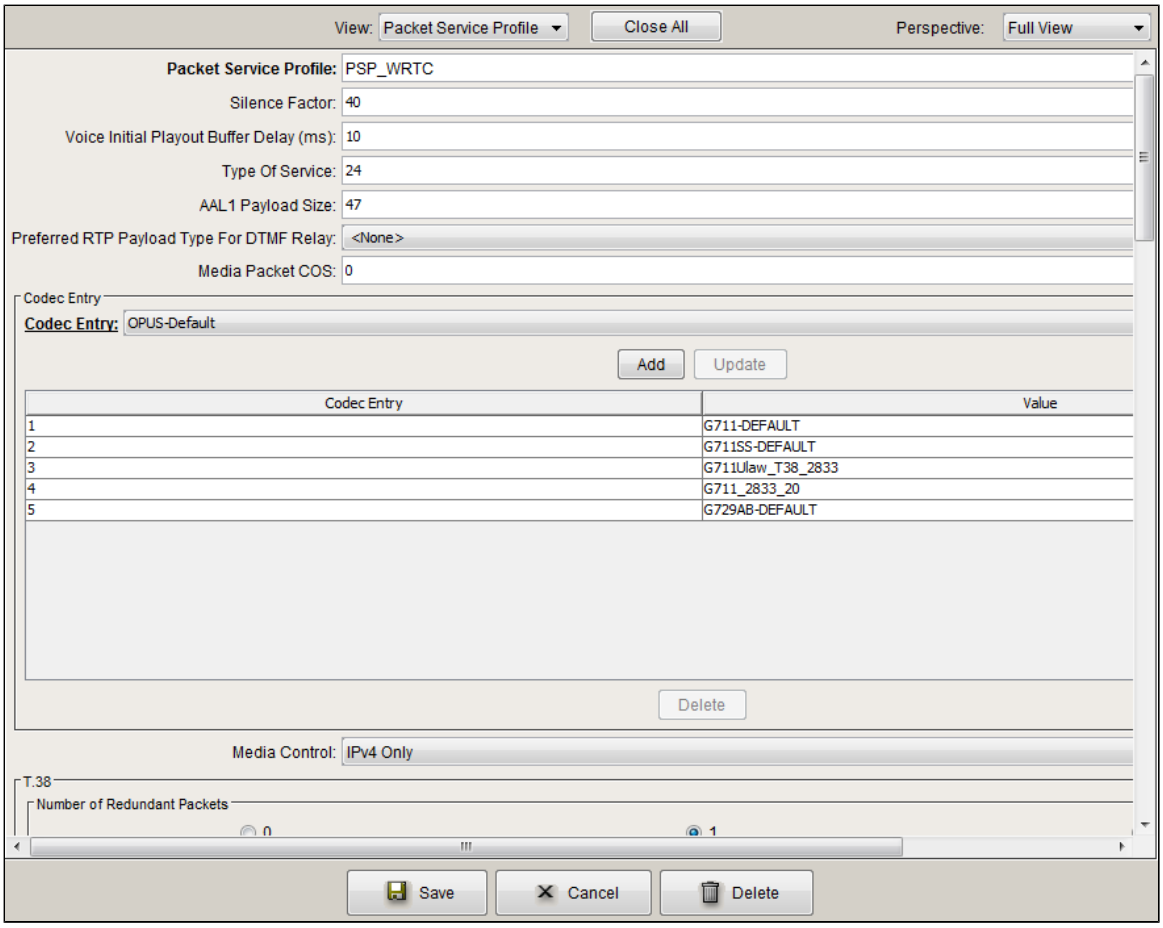
Creating the Codec Entry in External PSX

Figure 8: Codec Entry



Attaching the Codec Entry to the Packet Service Profile in External PSX

Figure 9: Attaching the Codec Entry to PSP



Creating Codec Entry in ERE

```
set profiles media codecEntry OPUS-Default codec opus packetSize 20 preferredRtpPayloadType 111 fax
failureHandling continue toneTreatment none
```

Attaching Codec Entry to PSP in ERE

```
set profiles media packetServiceProfile PSP_WRTC codec codecEntry12 OPUS-Default
```

Applying SMM Rules to Remove the Unrecognized Codec Lines

To remove the the unrecognized codec lines, refer to the section Defining SMM Rules.

Defining SMM Rules

As SBC does not support SAVPF, the following SMM rule is applied for inter-working with WRTC endpoints:

```
#### To replace RTP/SAVP to RTP/SAVPF ####
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 1
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 1 criterion 1 type message
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 1 criterion 1 type message message
messageTypes all condition exist
```

```
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 1 action 2 type messageBody
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 1 action 2 operation regsub
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 1 action 2 regexp string "RTP/SAVP"
matchInstance all
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 1 action 2 from type value value
"RTP/SAVPF"
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 1 action 2 to type messageBody
messageBodyValue all
set profiles signaling sipAdaptorProfile OUT_SMM_RULE state enable
commit
#### To replace actpass to active ####
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 applyMatchHeader one
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 criterion 1 type message
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 criterion 1 message
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 criterion 1 message messageTypes
request
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 criterion 1 message statusCode 200
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 criterion 2 type messageBody
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 criterion 2 messageBody
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 criterion 2 messageBody condition exist
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 action 1 type messageBody
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 action 1 operation regsub
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 action 1 from
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 action 1 from type value
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 action 1 from value
"a=setup:actpass\r\n"
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 action 1 to
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 action 1 to type messageBody
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 action 1 to messageBodyValue all
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 action 1 regexp
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 action 1 regexp string
"a=setup:active\r\n"
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 2 action 1 regexp matchInstance all
commit
#### To remove the unrecognized codec lines ####
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 3
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 3 criterion 1 type message
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 3 criterion 1 type message message
messageTypes all condition exist
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 3 action 1 type messageBody
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 3 action 1 operation regdel
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 3 action 1 regexp string
"a=rtcp-fb.*?\r\n" matchInstance all
set profiles signaling sipAdaptorProfile OUT_SMM_RULE rule 3 action 1 to type messageBody
messageBodyValue all
set profiles signaling sipAdaptorProfile OUT_SMM_RULE state enable
commit
#### To delete ssrc attribute from the incoming message ####
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 applyMatchHeader one
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 criterion 1 type message
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 criterion 1 message
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 criterion 1 message messageTypes all
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 criterion 1 message statusCode 200
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 criterion 2 type messageBody
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 criterion 2 messageBody
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 criterion 2 messageBody condition exist
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 action 1 type messageBody
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 action 1 operation regdel
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 action 1 to
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 action 1 to type messageBody
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 action 1 to messageBodyValue all
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 action 1 regexp
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 action 1 regexp string "a=ssrc:.*?\r\n"
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 1 action 1 regexp matchInstance all
commit
#### To delete extmap attribute from the incoming message ####
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 applyMatchHeader one
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 criterion 1 type message
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 criterion 1 message
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 criterion 1 message messageTypes all
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 criterion 1 message statusCode 200
```

```
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 criterion 2 type messageBody
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 criterion 2 messageBody
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 criterion 2 messageBody condition exist
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 action 1 type messageBody
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 action 1 operation regdel
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 action 1 to
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 action 1 to type messageBody
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 action 1 to messageBodyValue all
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 action 1 regexp
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 action 1 regexp string
"a=extmap:.*?\r\n"
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 2 action 1 regexp matchInstance all
commit
#### To delete msid-semantic attribute from the incoming message ####
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 applyMatchHeader one
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 criterion 1 type message
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 criterion 1 message
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 criterion 1 message messageTypes all
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 criterion 1 message statusCode 200
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 criterion 2 type messageBody
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 criterion 2 messageBody
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 criterion 2 messageBody condition exist
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 action 1 type messageBody
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 action 1 operation regdel
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 action 1 to
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 action 1 to type messageBody
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 action 1 to messageBodyValue all
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 action 1 regexp
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 action 1 regexp string
"a=msid-semantic:.*?\r\n"
```

```
set profiles signaling sipAdaptorProfile IN_SMM_RULE rule 3 action 1 regexp matchInstance all
set profiles signaling sipAdaptorProfile IN_SMM_RULE state enable
commit
```

These SMM profile is assigned to the Trunk Group towards the WRTC.

Assigning SMM Profiles to Trunk Group

The SMM profile is applied to the Trunk Group as shown below:

```
set addressContext default zone ZONE_WRTC sipTrunkGroup TG_SIPART_WRTC signaling messageManipulation
inputAdapterProfile IN_SMM_RULE outputAdapterProfile OUT_SMM_RULE
commit
```

Other Configuration

```
set addressContext default zone ZONE_WRTC sipTrunkGroup TG_SIPART_WRTC services natTraversal
mediaNat disabled
set profiles media packetServiceProfile PSP_WRTC rtcpOptions rtcp enable
```

The STUN handling for media NAT and ICE are mutually exclusive. Therefore, `mediaNAT` is disabled when ICE is used.

For DTLS, an association is created for both RTP and RTCP. The RTCP control must be enabled for RTCP packets to flow.

- For routing related information, refer to the section Category - Call Routing.
- For configuring video call, refer to the section Configuring SIP-SIP Video.
- For more information on WRTC configuration, refer to the *WRTC Documentation*.

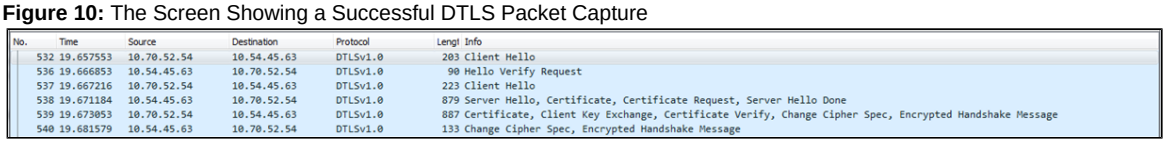
Viewing the Call Detail Status

To view the call detail status for an ICE enabled WRTC call:

```
show status global callDetailStatus
callDetailStatus 44 {
    mediaStreams                        audio;
    state                               Stable;
    callingNumber                       33002;
    calledNumber                        8095300530;
    addressTransPerformed               none;
    origCalledNum                       "";
    scenarioType                        SIP_TO_SIP;
    callDuration                        4;
    mediaType                           passthru;
    associatedGcid1                     44;
    associatedGcid2                     44;
    associatedGcidLegId1                1;
    associatedGcidLegId2                0;
    ingressSessionBandwidthkbps         76;
    egressSessionBandwidthkbps          72;
    ingressRemoteIpSockAddr             10.54.48.41;
    ingressRemotePort                   5080;
    egressRemoteIpSockAddr              10.70.52.68;
    egressRemotePort                    5060;
    ingressMediaStream1LocalIpSockAddr  "10.54.45.56/ 1538 (rtcp: 1539)";
    ingressMediaStream1RemoteIpSockAddr "10.70.52.68/ 63185 (rtcp: 63186)";
    egressMediaStream1LocalIpSockAddr   "10.54.47.56/ 1528 (rtcp: 1529)";
    egressMediaStream1RemoteIpSockAddr  "10.70.52.68/ 60526 (rtcp: 60527)";
    ingressMediaStream1Security
rtp-Encrypted,rtp-auth,srtp-terminated,rtcp-encrypted,rtcp-auth,crypto-aescm,hmacsha180;
    egressMediaStream1Security          rtp-disabled,rtcp-disabled;
    ingressMediaStream1Bandwidth        76;
    egressMediaStream1Bandwidth         72;
    ingressMediaStream1IceState         ST_ICE_COMPLETE;
    egressMediaStream1IceState          NONE;
    ingressDtlsStream1                  TERMINATED;
    egressDtlsStream1                   DISABLED;
    iceCallTypes                        ing-lcl-FULL-ICE,ing-rmt-FULL-ICE,eg-lcl-NONE,eg-rmt-NONE;
    ingressACName                       a1;
    ingressZoneName                     INTERNAL;
    ingressTrunkName                    ING_Coper_MaleSwe1;
    egressACName                        a1;
    egressZoneName                      EXTERNAL;
    egressTrunkName                     MALESWE1_EGR;
}
```

The following screen shows a successful DTLS handshake packet capture:

Figure 10: The Screen Showing a Successful DTLS Packet Capture



No.	Time	Source	Destination	Protocol	Lengt	Info
532	19.657553	10.70.52.54	10.54.45.63	DTLSv1.0	203	Client Hello
536	19.666853	10.54.45.63	10.70.52.54	DTLSv1.0	90	Hello Verify Request
537	19.667216	10.70.52.54	10.54.45.63	DTLSv1.0	223	Client Hello
538	19.671184	10.54.45.63	10.70.52.54	DTLSv1.0	879	Server Hello, Certificate, Certificate Request, Server Hello Done
539	19.673053	10.70.52.54	10.54.45.63	DTLSv1.0	887	Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
540	19.681579	10.54.45.63	10.70.52.54	DTLSv1.0	133	Change Cipher Spec, Encrypted Handshake Message