Session Initiation Protocol (SIP)

bintec-Dm 766-I

Legal Notice

Warranty

This publication is subject to change.

bintec offers no warranty whatsoever for information contained in this manual.

bintec is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

Chapter 1 Introduction

1.1 Introduction

1.1.1 SIP Signaling Protocol

The Session Initiation Protocol (SIP) is a signaling protocol whose main function is to create, modify and terminate sessions over IP networks. To that end, it allows you to locate users and share information on the media involved in the session. It is totally independent of the type of session being established and, consequently, can be used to initiate voice conversations, videoconferences, shared applications, etc. It is also independent from the transport protocol (UDP, TCP, and TLS/TCP) and from the protocol used to negotiate session parameters, which is Session Description Protocol (SDP) by default. Since it is very generic and versatile, numerous applications and features can be implemented.

The Internet Engineering Task Force (IETF) standardized this protocol through the RFC 2543 release 1.0, followed by the current release 2.0 described in RFC 3261. In this latter RFC, the basic function of the protocol is described, citing other accompanying RFCs (such as RFCs 3262-3265) for other specific aspects. Additionally, the development of the protocol was predicted through new RFCs. Consequently, this has allowed for numerous extensions to appear (increasing the functionalities and possibilities).

SIP is a textual protocol, whose syntax is described through an ABNF notation. It’s very similar to the HTTP protocol. SIP messages come in two types; requests or methods and responses. A SIP transaction consists of a request and one or various responses. SIP entities incorporate a state machine for each transaction so retransmissions are produced when the transport protocol used is unreliable (UDP) and the protocol resend timers time out.

The methods defined in the basic norm are: INVITE, ACK, CANCEL, BYE, REGISTER and OPTIONS. INVITE and ACK are special cases, which now form part of a special transaction used to initiate a session. This consists of the INVITE petition, one or various responses to this and the ACK method, which does not receive a response, and terminates the transaction. This way, a more reliable transaction is achieved, as 3 messages need to be exchanged. Various RFCs extend the basic norm defining new applications for presence, instant messaging, session transfer, indication of new messages in the voice mail, etc. Some of these methods are: REFER, NOTIFY, SUBSCRIBE, MESSAGE and UPDATE.

A session is initiated through INVITE/ACK, while BYE is used to terminate sessions. CANCEL allows you to cancel a current transaction (although, in practice, this is only applied to the INVITE initial transaction). REGISTER is a message indicating the user location registering this in a server and OPTIONS is a transaction used to find out what methods and extensions are supported by a specified SIP entity, which appear in its response.

The responses have a 3-digit code, coinciding with those used in HTTP, which, depending on the first digit of the code, mean something different. These can be divided into provisional responses (1xx) and end responses (2xx-6xx). A transaction is made up of various or no provisional responses and a single end response. The types of responses, depending on the first digit, are as follows:

> 1xx: provisional responses. These provide information but do not end a transaction.
>
> 2xx: successful responses. These indicate that the request has been successful.
>
> 3xx: re-addressing responses. The request has not been successful but is being redirected to another address.
>
> 4xx: error responses from the client making the request.
>
> 5xx: error responses from the server attending the request.
>
> 6xx: global error responses to the entire network.

The following entities are defined in a SIP scenario:

- **User Agent (UA):** These are user agents between those establishing the sessions. They can act as User Agent Client (UAC) when sending requests, or as User Agent Server (UAS) when sending responses to the requests received.

Registrar: This is the entity receiving REGISTER requests. The UAs send these requests to associate the register’s known public SIP address (user@domain.es) with the current SIP address (*user@X.X.X.X:XXXX*). The Registrar updates a registered user database known as Location Server.

- **Location Server:** This entity is a database containing information on the registered users. It is a logical entity that does not communicate through SIP. It can be implemented as a database in the memory, an external database accessible through SQL, LDAP languages, etc. It is updated by the Registrar and used by a SIP Server to route messages to the users' current location.

- **Proxy Server**: This is the most common SIP Server. It receives a SIP request and forwards it to the destination entity. To do this, it consults the Location Server or sends the request in accordance with a programmed method, which can depend on the called user, the time of day, etc. It can maintain various context levels depending on whether it is *stateless* (does not maintain information between messages), *stateful* (implements a state machine for each transaction) or *call stateful* (maintains the signaling route during the whole session).
- **Redirect Server:** This is the simplest SIP Server. Instead of redirecting messages to the appropriate entity, it always responds with a special redirect message indicating the address the request must be sent to. The UA receiving this redirect message must repeat the request sending it to the URL it has been redirected to.
- **Back-to-Back User Agent (B2BUA):** This is also a SIP Server, made up of two UAs. One receives the call and launches another session with the destination through another UA request. From the point of view of the signaling, both ends of the call see the B2BUA as the interlocutor. While the signaling generates in one session, it is received in the other. This functionality could be considered as a SIP-SIP gateway. This entity grants greater control over the sessions as, unlike the other servers mentioned above, the B2B can spontaneously generate SIP requests to the two permitted ends (e.g., end a session).

A device frequently groups some of the previously described entities. The Registrar function is usually housed in the same device as the one behaving as a SIP Server.

There can be two types of SIP URLs:

- *sip:usuario@host:port;tag1=value1;tag2=value2;...* Normal SIP address. The host can be both a generic domain name and the domain name of a specific device or even an IP address. The additional parameters allow you to specify further information, such as the transport protocol.
- *sips:usuario@host:port;tag1=value1;tag2=value2;...* Secure SIP address. This ensures that TLS is used (Transport Layer Secure) between the source and destination domain.

The following schema shows how a session is started between two ends. In this example, Alice has telephone software in her PC and wants to begin a conversation with Bob. The telephone sends an INVITE request to its SIP server (in this case, a proxy). The latter resends this to the destination domain proxy server (biloxi.com), which is different from Alice's (atlanta.com). The proxy consults its Location Server and discovers Bob's whereabouts, redirecting the INVITE to him. Each intermediate server generates a 100 Trying request to stop INVITE retransmissions. Bob's telephone sends a provisional response by ringing, which is resent to Alice's telephone for an appropriate indication. Finally, Bob picks up his phone and sends the final 200 OK response. When the response reaches Alice's telephone, ACK is sent to terminate the transaction. In this example, the signaling is directly sent from one end to the other without passing through the servers as, in the response, Bob will have given Alice his contact address. Servers, however, may require that all messages pass through them.

```
               atlanta.com  . . . biloxi.com
           .      proxy              proxy     .
         .                                       .
 Alice's  . . . . . . . . . . . . . . . . . . . .  Bob's
softphone                                        SIP Phone
   |                |                |                |
   |    INVITE F1   |                |                |
   |--------------->|    INVITE F2   |                |
   |  100 Trying F3 |--------------->|    INVITE F4   |
   |<---------------|  100 Trying F5 |--------------->|
   |                |<-------------- | 180 Ringing F6 |
   |                | 180 Ringing F7 |<---------------|
   | 180 Ringing F8 |<---------------|     200 OK F9  |
   |<---------------|    200 OK F10  |<---------------|
   |    200 OK F11  |<---------------|                |
   |<---------------|                |                |
   |                       ACK F12                    |
   |------------------------------------------------->|
   |                   Media Session                  |
   |<================================================>|
   |                       BYE F13                    |
   |<-------------------------------------------------|
   |                     200 OK F14                   |
   |------------------------------------------------->|
   |                                                  |
```

In the body of the session start messages, all the media details going to be used in this session are negotiated: number of connections, type (audio, video, application, etc.), ips and ports, codecs, etc. This negotiation is independent of the SIP and can be carried out in any protocol, although in practice it's executed in SDP (RFC 2327), which is viewed by SIP as the default protocol and which all the UAs are constrained to understand. What is defined in SIP is a negotiation model known as offer/response. This method consists of a UA sending an offer, with the medias it wishes to use in the session, specifying the type of medias, codecs and the ips and ports where it expects to receive the data on each media. The other UA responds with a subset of media and codecs it can support for the session. This goes together with the corresponding ips and ports through which it expects to receive the data. The use of SDP to negotiate in SIP is described in RFC 3264.

In this example, the session finishes when either agent sends a BYE request.

1.1.2 Available Functionality in the router

The device is equipped with a Back-to-Back SIP Server, a User Agent (when behaving as a voice gateway), a Registrar and a Location Server. This means it can act as an autonomous SIP call switchboard or, together with other external SIP servers, as part of a more extensive SIP network. It can act as an emergency switchboard, capable of substituting an external server when faced with a drop in service (providing a basic functionality to maintain the IP telephony system running).

Two non-exclusive operating modes (server and gateway) can be configured for this:

You can configure an external SIP proxy in any operating mode and associate it with the supervision functionality offered by the network at service level. This is known as Network Service Level Advisor (bintec-Dm754-I NSLA) and uses the information obtained by the Network Service Monitor (bintec-Dm749-I NSM). Consequently, the server is only active if the corresponding NSLA advisor is enabled.

The supported transport modes are UDP, TCP and TLS (over TCP).

With incoming SIP calls, the dial plan configured in the Telephony menu is checked (bintec-Dm722-I Telephony over IP,). It is based on dial-peers as the basic configuration unit.

- **Server:**

In this mode, the device admits user registers and creates dynamic dial-peers, allowing calls to be established with the registered telephone numbers. The behavior regarding the registers changes depending on whether there is an active SIP proxy or not. With a SIP proxy, the register resends the message and the response is sent to the user. Without a proxy, the device behaves as a Registrar and responds to the register.

When the device resends registers to the configured SIP proxy, the contact address registered in the server is the telephone address. Consequently, the proxy establishes the calls to the telephones without going through the device. This is the desired and most common behavior. However, it can be changed through the **application server local-ip-registrations** command. This causes the registers to be resent by substituting the telephone IP/port for the device IP/port. Loops can take place if there is a dial-peer to send all the calls to the proxy, as the call returns to the device. This can be resolved by configuring a more advanced dial plan, using the **dial-plan** and **incoming acc-list** commands found in the dial-peers menu.

If the destination for an incoming SIP call is another SIP dial-peer, the device behaves as Back-to-Back between the two dial-peers. If the output dial-peer cannot be found and there is an active SIP proxy, the call is established through the proxy. This latter behavior can be avoided by configuring the **call application outgoing-match force** command found in the *tlphy* menu, which ensures a call that does not find an outgoing dial-peer fails.

When the external SIP proxy is inactive, all the dial-peers with *target sip-proxy* are also inactive.

- **Gateway:**

The device is capable of behaving as a SIP gateway, receiving SIP calls and rerouting them to any of its VoIP interfaces, or forwarding calls received via any VoIP interfaces towards a SIP dial-peer. In addition, if there is a SIP proxy or Registrar configured, the device registers the voice-port or group dial-peers here.

Chapter 2 Configuration

2.1 Accessing the Configuration Menu

The SIP protocol configuration commands must be entered in the configuration menu associated with the SIP (*SIP Config>*). To access said menu, enter the **protocol sip** command in the general configuration menu (*Config>*).

```
Config>protocol sip

SIP Config>
```

If you want the commands to activate immediately (i.e., without rebooting the router), access the configuration through the dynamic general configuration menu (*Config$*).

```
Config$protocol sip

SIP Config$
```

2.2 SIP Menu Configuration Commands

2.2.1 [NO] APPLICATION

2.2.1.1 APPLICATION ADDRESS

Configures the IP address the SIP uses to send and receive signaling messages. If no IP address is configured, the device's internal IP address is used.

Syntax:

```
SIP Config$[no] application address <a.b.c.d>    Ipv4 format
```

2.2.1.2 APPLICATION GATEWAY

Activates the SIP gateway so it can establish calls between the device's VoIP interfaces and the SIP dial-peers.

Syntax:

```
SIP Config$[no] application gateway
```

2.2.1.3 APPLICATION PORT

Configures both the TCP and the UDP port where the device SIP server listens. Default port is 5060. The TLS server listens at the next configured port, default is 5061.

Syntax:

```
SIP Config$[no] application port udp <1..65535>    Value in the specified range
```

2.2.1.4 APPLICATION SERVER DEFAULT

Activates the SIP server with its default behavior, the counter command, **no application server default**, terminates the SIP server and eliminates all the calls if this is executed from the dynamic configuration. Functionality in this case is described in chapter 1, section 1.2.

Syntax:

```
SIP Config$[no] application server default
```

2.2.1.5 APPLICATION SERVER MESSAGE-FILTERING

Modifies the device SIP server behavior so that when SIP messages are resent from the source to the destination, some unacknowledged signaling parameters are eliminated instead of directly copying the headers. This can help make two external devices from different manufacturers, which otherwise would not be able to understand each other, compatible. The **no application server message-filtering** command disables this behavior.

Syntax:

```
SIP Config$[no] application server message-filtering
```

2.2.1.6 APPLICATION SERVER LOCAL-IP-REGISTRATIONS

Modifies the device SIP server behavior so that, on resending the received registers to the configured proxy, the telephone contact address is replaced by the one belonging to the device SIP application. Consequently, the extensions for the proxy are located in the device that receives the incoming calls signaling. The **no application server local-ip-registrations** command disables this behavior, so that registers are able to reach the server with their original contact and the incoming calls from the external proxy are signaled directly towards them (without passing through the device). This is the default behavior.

Syntax:

```
SIP Config$[no] application server local-ip-registrations
```

2.2.1.7 APPLICATION SERVER TRACK NSLA-ADVISOR

Modifies the behavior of the SIP server depending on the state of the configured NSLA advisor. This only affects the router's reply to the telephone registers when this acts as the autonomous register server (i.e., when there isn't an active external proxy). If the poll result is true, telephone registers are admitted, a 200OK response is sent to the registers and the information is saved in a dynamic dial-peer. However, if the poll result is false, the registers are rejected through the 405 code.

Syntax:

```
SIP Config$[no] application server track nsla-advisor <1..65535>
```

2.2.2 [NO] CALLED-NUMBER TO-HEADER

If this command is configured, the number called in incoming SIP calls is obtained from the TO header field of the INVITE request rather than from the Request URI.

Syntax:

```
SIP Config$[no] called-number to-header
```

Command history:

Version	Modification
10.9.20	This command was implemented in version 10.9
11.0.0.2.6	This command was implemented in version 11.0.0.2.6
11.0.3	This command was implemented in version 11.0.3

2.2.3 [NO] CONTACT-ADDRESS

Allows you to configure the host to be used in the contact address of SIP messages sent by the router. You can specify an IP or a domain name.

Syntax:

```
SIP Config$[no] contact-address <word>    Text
```

2.2.4 [NO] CONTACT-MATCHING

Helps configure the level of strictness of the *contact-address* field when comparing the value sent by the router in a REGISTER petition and the value received in the response. You then need to indicate an option to specify the type of comparison.

2.2.4.1 CONTACT-MATCHING STRICT

Indicates that all parts making up the *contact-address* field of the response to the REGISTER petition should be checked. If they do not not correspond to the expected values (those sent in the petition), then the response stops processing.

This is the default option.

Syntax:

```
SIP Config$contact-matching strict
```

2.2.4.2 CONTACT-MATCHING USER

Checks that the user name, found in the *contact-address* field in the response, is the same as the one sent in the REGISTER petition. If the user does not correspond to the expected value, then the response stops processing. This command is particularly indicated for cases where an intermediate node in the network modifies the *contact-address* (e.g., a device using NAT that alters the IP address for this field).

Syntax:

```
SIP Config$contact-matching user
```

2.2.5 [NO] CRYPTO

Through this command, you can configure the parameters relative to TLS crypto. These parameters are used when the transport mode for a call is TLS. In order to use TLS transport, the device must have a user certificate configured through the **crypto signaling default user** command.

2.2.5.1 CRYPTO CLIENT CERTIFICATE DONT-VERIFY

If this option is active, when the device is acting as a client it does not verify if the server's X509 certificate is signed by a reliable certification authority. By default this option is not active and the device checks that the server certificate has been signed by a reliable authority.

Syntax:

```
SIP Config$[no] crypto client certificate dont-verify
```

2.2.5.2 CRYPTO SERVER CERTIFICATE DONT-VERIFY

If this command is configured, when the device is acting as server it does not request the X509 certificate from the connected client. Consequently, clients are not authenticated. By default this option is not active and the certificate is requested from the clients and checked to ensure it is signed by a reliable certifier.

Syntax:

```
SIP Config$[no] crypto server certificate dont-verify
```

2.2.5.3 CRYPTO SIGNALING DEFAULT CA <ca-name>

Configures the reliable certifier authorities for TLS sessions established by the SIP protocol. You can configure various certificates from different certification authorities. The certificates must be previously loaded in the device.

The certification authority allows you to validate device certificates. These are communicated to the router through TLS, either in client or server mode.

To load a certificate in base64, you can execute **certificate <certname> base64** and introduce the certificate in the device through the ipsec certificate menu. This generates a configuration in ipsec as shown in the example. For further details, please see the section on Certificates (chapter 2) in manual bintec-Dm739-I IPSEC.

Syntax:

```
SIP Config$[no] crypto signaling default ca <ca-name>
```

Example:

Loading a bintec example root certificate through the **certificate <name> base64** command in the *protocol ip>ipsec>cert* menu.

```
CERTIFICATES config$certificate BINTECCA.CER base64

Introduce the Certificate (Base 64 format)
Enter <cr> to escape
-----BEGIN CERTIFICATE-----
MIIEmzCCA4OgAwIBAgIJAIjCeKBqciDFMA0GCSqGSIb3DQEBBAUAMIGPMQswCQYD
VQQGEwJTUDEPMA0GA1UECBMGTWFkcmlkMRQwEgYDVQQHEwtUcmVzIENhbnRvczEU
MBIGA1UEChMLVGVsZGF0IFMuQS4xGzAZBgNVBAsTEklQIFRlbGVwaG9ueSBHcm91
cDEmMCQGA1UEAxMdVGVkYXQgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkwHhcNMDcw
NjExMTUxNDE2WhcNMTcwNjA4MTUxNDE2WjCBjzELMAkGA1UEBhMCU1AxDzANBgNV
BAgTBk1hZHJpZDEUMBIGA1UEBxMLVHJlcyBDYW50b3MxFDASBgNVBAoTC1RlbGRh
dCBTLkEuMRswGQYDVQQLExJJUCBUZWxlcGhvbnkgR3JvdXAxJjAkBgNVBAMTHVRl
ZGF0IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MIIBIjANBgkqhkiG9w0BAQEFAAOC
```

```
AQ8AMIIBCgKCAQEAmSRtZ9wHCksPAzkdMvqYyUAnOecJWw/Aai67TObXhi/a4w5T
Onbf8LKjsGWamksMU6p7iv7n4rd6Kqyr1q/S1yP9XfENiVfsmu3dq9ehkipg5ixw
E16xAdpGXJpdob8zOkUwiKaJib8LsTE38upaA2iV++bQSIMKcma4rnlPW1wn9jAJ
mMwTMKCT7vT7OfcEIVzB7P1RW9phTMmQsSTTg7SMlRxTN0c2WW216aLOO5qRwvt4
xzcoXRVYbm2aBj7LucjsOrgoEdscmga8kK7PYdetxqti1n6RfjP2BXmAUrKh91c3
61fazv+pNxpKSL0hQ8Gb+hUxPyjZJTTW+Zih+wIDAQABo4H3MIH0MB0GA1UdDgQW
BBQsJNVrUzOnr7Rxj4FfdiBLKOSv9DCBxAYDVR0jBIG8MIG5gBQsJNVrUzOnr7Rx
j4FfdiBLKOSv9KGBlaSBkjCBjzELMAkGA1UEBhMCU1AxDzANBgNVBAgTBk1hZHJp
ZDEUMBIGA1UEBxMLVHJlcyBDYW50b3MxFDASBgNVBAoTC1RlbGRhdCBTLkEuMRsw
GQYDVQQLExJJUCBUZWxlcGhvbnkgR3JvdXAxJjAkBgNVBAMTHVRlZGF0IENlcnRp
ZmljYXRpb24gQXV0aG9yaXR5ggkAiMJ4oGpyIMUwDAYDVR0TBAUwAwEB/zANBgkq
hkiG9w0BAQQFAAOCAQEAR16Gjs16Sqz04v/RJeRb+fcbKvAgzO3sWpUyYwzU/j6L
7R5XVbgimX4FQ3qxnrNeYXCTtZAM8yMWKpnX1d9ZDgGqZsOV0NrjlSGAYk3yvdM5
cNEXQpLDkKhjN8ageD48yNWpBTzbTDk/jQXCfktF3L93qpB/W76taC54bb1LojHs
kcPXB4pzgN7QGct/wVyg2KNcMaQITmOesY+Qqt8T0QxZomsn8ldz6c7HAoRurmnB
x/SCdpqfwMMnS7ap/5y+uPNuROw3ib8GWWqq6I3/bUqxgkgEwWD8OdkYHKNJV5h8
0zJjXH5/jqf1hmwKV07QQ+WxENxdtc6FB3Idmgj33w==
-----END CERTIFICATE-----
```

The certificate data can be seen through the **list loaded-certificates** command found in this menu. If you have loaded from the static configuration, you’ll need to save it and restart the device in order to save the certificate in the memory.

```
CERTIFICATES config$list loaded-certificates
----------------------- BINTECCA.CER (from config)
 Subject:
  CN (Common Name        ): bintec Certification Authority
  OU (Organizational Unit): IP Telephony Group
  O  (Organization Name  ): bintec
  L  (Locality           ): Tres Cantos
  S  (State or Province  ): Madrid
  C  (Country Name       ): SP

 Issuer: A:BINTECCA.CER

CERTIFICATES config$
```

This has generated the following configuration in the certificates menu. This configuration can be directly used in other devices without having to reload the certificate in base64.

```
protocol ip
; -- Internet protocol user configuration –
   Ipsec
; -- IPSec user configuration --
      Cert
; -- Cert user configuration --
         file new BINTECCA.CER
         file add 0x3082049B30820383A00302010202090088C278A06A7220C5300D06092A864886
         file add 0xF70D01010405003081 8F310B3009060355040613025350310F300D0603550408
         file add 0x13064D61647269643114301206035504071 30B547265732043616E746F733114
         file add 0x3012060355040A130B54656C64617420532E412E311B3019060355040B131249
         file add 0x502054656C6570686F6E792047726F7570312630240603550403131D54656461
         file add 0x7420436572746966696361746 96F6E20417574686F72697479301E170D303730
         file add 0x3631313135313431365A170D3137303630383135313431365A30818F310B3009
         file add 0x0603550406130253503 10F300D060355040813064D616472696431143012060 3
         file add 0x550407130B547265732043616E746F7331143012060355040A130B54656C6461
         file add 0x7420532E412E311B3019060355040B131249502054656C6570686F6E79204772
         file add 0x6F7570312630240603550403131D5465646174204365727469666963617469 6F
         file add 0x6E20417574686F7269747930820122300D06092A864886F70D01010105000382
         file add 0x010F003082010A0282010100 99246D67DC070A4B0F03391D32FA98C9402739E7
         file add 0x095B0FC06A2EBB4CE6D7862FDAE30E533A76DFF0B2A3B0659A9A4B0C53AA7B8A
         file add 0xFEE7E2B77A2AACABD6AFD2D723FD5DF10D8957EC9AEDDDABD7A1922A60E62C70
         file add 0x135EB101DA465C9A5DA1BF333A453088A68989BF0BB13137F2EA5A036895FBE6
         file add 0xD048830A7266B8AE794F5B5C27F6300998CC1330A093EEF4FB39F704215CC1EC
         file add 0xFD515BDA614CC990B124D383B48C951C53374736596DB5E9A2CE3B9A91C2FB78
         file add 0xC737285D15586E6D9A063ECBB9C8EC3AB82811DB1C9A06BC90AECF61D7ADC6AB
         file add 0x62D67E917E33F605798052B2A1F75737EB57DACEFFA9371A4A48BD2143C19BFA
         file add 0x15313F28D92534D6F998A1FB0203010001A381F73081F4301D0603551D0E0416
         file add 0x04142C24D56B5333A7AFB4718F815F76204B28E4AFF43081C40603551D230481
         file add 0xBC3081B980142C24D56B5333A7AFB4718F815F76204B28E4AFF4A18195A48192
         file add 0x30818F310B3009060355040613025350310F300D060355040813064D61647269
         file add 0x643114301206035504071300B547265732043616E746F7331143012060355040A
         file add 0x130B54656C64617420532E412E311B3019060355040B131249502054656C6570
         file add 0x686F6E792047726F75703126302406035504031310D5465646174204365727469
         file add 0x6669636174696F6E20417574686F7269747982090088C278A06A7220C5300C06
         file add 0x03551D13040530030101FF300D06092A864886F70D01010405000382010100047
         file add 0x5E868ECD7A4AACF4E2FFD125E45BF9F71B2AF020CCEDEC5A9532630CD4FE3E8B
         file add 0xED1E5755B822997E05437AB19EB35E617093B5900CF323162A99D7D5DF590E01
         file add 0xAA66C395D0DAE3952180624DF2BDD33970D1174292C390A86337C6A0783E3CC8
         file add 0xD5A9053CDB4C393F8D05C27E4B45DCBF77AA907F5BBEAD682E786DBD4BA231EC
         file add 0x91C3D7078A7380DED019CB7FC15CA0D8A35C31A4084E639EB18F90AADF13D10C
         file add 0x59A26B27F25773E9CEC702846EAE69C1C7F482769A9FC0C3274BB6A9FF9CBEB8
         file add 0xF36E44EC3789BF06596AAAE88DFF6D4AB1824804C160FC39D9181CA34957987C
         file end 0xD332635C7E7F8EA7F5866C0A574ED043E5B110DC5DB5CE8507721D9A08F7DF
;
         certificate BINTECCA.CER load
      exit
;
   exit
;
exit
```

Now you only need to configure this certificate as a trusted certification authority for the SIP protocol TLS connections.

```
protocol sip
   crypto signaling default ca BINTECCA.CER
exit
```

2.2.5.4 CRYPTO SIGNALING DEFAULT CIPHERS

Configures the encryption and authentication algorithms negotiated in TLS connections.

The string format is the same as that used by openssl. To see the format and the examples, please visit the following site: *http://www.openssl.org/docs/apps/ciphers.html* .

Syntax:

```
SIP Config$[no] crypto signaling default ciphers <string>
```

2.2.5.5 CRYPTO SIGNALING DEFAULT USER <cert-name>

Configures the certificate used by the device when behaving as a TLS connections server. This is also used if the device is behaving as a client and the server it connects to requests a certificate. The certificate must have been previously loaded in the device and have a private key assigned.

To generate a private key and load the signed certificate in the device, please see bintec-Dm739-I IPSEC.

Syntax:

```
SIP Config$[no] crypto signaling default user <cert-name>
```

Example:

Generating a private key and a Certificate Signing Request (CSR):

```
IPSec config>key rsa generate user.csr 512
RSA Key Generation.
Please, wait for a few seconds.
 RSA Key Generation done.
Checking..OK
Key Generation Process Finished.
Generate CSR?
(Yes/No)? y
Common Name        :  []? Sample User Certificate
Country            :  []? SP
Locality           :  []? Tres Cantos
State or Province  :  []? Madrid
Organization       :  []? bintec
Organizational Unit:  []? IP Telephony Group
```

```
E-mail             :  []?
RSA Signature(MD5/SHA1/MD2): [md5]?
Save in file(Yes/No)? y
File Name: []? sample.csr
File Name: [A:USER.CSR]? y
CSR saved.
Do not forget to save RSA keys.
IPSec config>
```

On displaying the configuration, you can see the private key generated and encrypted. By listing the user.csr file, you can obtain the CSR that the certifier authority must sign.

```
IPSec config>sho conf
; Showing Menu and Submenus Configuration for access-level 15 ...

         key rsa file add 0x341DC332F23BD75492E8583A82F10A8CFCA4349F531563EF
         key rsa file add 0xCA002248A79CFCFE24FBBCE0FA9C3BF99B02C316C9C5EB44
         key rsa file add 0xA2630B28F04183766A79C93BD967380425955159D32B7035
         key rsa file add 0x448A8DB2954FA8E53132CDAFE7365FED6BAE5CA55ED8809E
         key rsa file add 0x89191F4762B0850603BE8A61AFD3CC786EC5ED1EB08F191C
         key rsa file add 0xCF7B8FD6AF0C37BDF33BEC201C6B58B1FAC419DF8F68F525
         key rsa file add 0x31F285C22AD9896587FE9095A8355C6F35075CDFAE7E2485
         key rsa file add 0x6E75FC669194A00DED45B8AFDF3B99B6A162F7FE14B0F3B8
         key rsa file add 0xDC0E4D442F9EC916D05F161BABA2D3D803AABADF64A8E6EC
         key rsa file add 0x1CBD2973DC158D77A872B75FE4E99277E877E49C117FC6D9
         key rsa file add 0xC8E29F6D1D84030B955E1A6A15E2F15386CA5F6598148876
         key rsa file add 0x0C27B15CF5D812C2922706CF25C7D42DE09DCB4330125C2B
         key rsa file add 0x911BC4C084B9ADE1D6D5B7DDDDD030692F2EC9E66E0E7D74
         key rsa file add 0x782F99AC347A1BCAC42CEBCEF011F1D25646465BED83ABE9
         key rsa file add 0xBCC82DFBE5BA79E4D024AA7F6AB05D03101335AD37882784
         key rsa file add 0xF5F0142911803717889 4D1823871D8F498F9B2B5C1EB488D
         key rsa file add 0x9D6349C59E11A617F5622FFC33D8D6272D7C13C6F9B494CE
         key rsa file add 0x3148B09CB12A6B438EC87E272FBC4A0C629CD9DDCAC1B9A3
         key rsa file add 0x8DFEC5C76588A09F6417A94ACF76313C89198FE0D7B5E1B2
         key rsa file add 0x02249303A5A3731A0162B207950C41E18A3952DC84415084
         key rsa file add 0x41E09EF3908BD169243C9A611D1EA318FCEF7A2BD0378108
         key rsa file add 0xFD2E886FE114EAFBB1F18892F67FEA2173D8F05B5ED54B67
         key rsa file add 0x0D649530B2392230C9AA2D9974777147DFBCCD2067222A11
         key rsa file add 0x8C49A3D60E22901AC5103313CE5CC0B9FA1A2F1607BC55EE
         key rsa file add 0xA6EB05FB527E786CD4529F1388F6E66AFBFA41234902488E
         key rsa file end 0xB4303ABF65069D25D17145D8695CBD88EC0C92EECC210B36
IPSec config>
IPSec config>ex
IP config>ex
Config>file type a:user.csr
-----BEGIN CERTIFICATE REQUEST-----
MIIBRDCB7wIBADCBiTEgMB4GA1UEAxMXU2FtcGxlIFVzZXIgQ2VydGlmaWNhdGUx
CzAJBgNVBAYTAlNQMRQwEgYDVQQHEwtUcmVzIENhbnRvczEPMA0GA1UECBMGTWFk
cmlkMRQwEgYDVQQKEwtUZWxkYXQgUy5BLjEbMBkGA1UECxMSSVAgVGVsZXBob255
IEdyb3VwMFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBANflczMh4//+vDYBHSrzou5N
LamVxi4GStHWJatFHdsKfiYU7S6DKjgmB9Acyi2haH+bydDqD2pMDCPMvfyURjEC
AwEAAaAAMA0GCSqGSIb3DQEBBAUAA0EAEzLTh/ZwLKM2J9IyEeYqCevSFm/zye53
bz1R58go44cpLWgT9YL3kZoKrKAqevWdNRyjHKuwZmt+XoHaRVkSog==
-----END CERTIFICATE REQUEST-----
```

Once the certification authority has signed the CSR, the user certificate can be loaded into the router in the same way as the previous example. Now the certificate needs to be configured so that the SIP uses it as the user certificate. The final configuration looks like the one shown below and can be used in another device directly, without going through the whole process again.

```
protocol ip
; -- Internet protocol user configuration --
   ipsec
; -- IPSec user configuration --
      key rsa file add 0x341DC332F23BD75492E8583A82F10A8CFCA4349F531563EF
      key rsa file add 0xCA002248A79CFCFE24FBBCE0FA9C3BF99B02C316C9C5EB44
      key rsa file add 0xA2630B28F04183766A79C93BD967380425955159D32B7035
      key rsa file add 0x448A8DB2954FA8E53132CDAFE7365FED6BAE5CA55ED8809E
      key rsa file add 0x89191F4762B0850603BE8A61AFD3CC786EC5ED1EB08F191C
      key rsa file add 0xCF7B8FD6AF0C37BDF33BEC201C6B58B1FAC419DF8F68F525
      key rsa file add 0x31F285C22AD9896587FE9095A8355C6F35075CDFAE7E2485
      key rsa file add 0x6E75FC669194A00DED45B8AFDF3B99B6A162F7FE14B0F3B8
      key rsa file add 0xDC0E4D442F9EC916D05F161BABA2D3D803AABADF64A8E6EC
      key rsa file add 0x1CBD2973DC158D77A872B75FE4E99277E877E49C117FC6D9
      key rsa file add 0xC8E29F6D1D84030B955E1A6A15E2F15386CA5F6598148876
      key rsa file add 0x0C27B15CF5D812C2922706CF25C7D42DE09DCB4330125C2B
      key rsa file add 0x911BC4C084B9ADE1D6D5B7DDDDD030692F2EC9E66E0E7D74
      key rsa file add 0x782F99AC347A1BCAC42CEBCEF011F1D25646465BED83ABE9
      key rsa file add 0xBCC82DFBE5BA79E4D024AA7F6AB05D03101335AD37882784
      key rsa file add 0xF5F01429118037178894D1823871D8F498F9B2B5C1EB488D
      key rsa file add 0x9D6349C59E11A617F5622FFC33D8D6272D7C13C6F9B494CE
      key rsa file add 0x3148B09CB12A6B438EC87E272FBC4A0C629CD9DDCAC1B9A3
      key rsa file add 0x8DFEC5C76588A09F6417A94ACF76313C89198FE0D7B5E1B2
      key rsa file add 0x02249303A5A3731A0162B207950C41E18A3952DC84415084
      key rsa file add 0x41E09EF3908BD169243C9A611D1EA318FCEF7A2BD0378108
      key rsa file add 0xFD2E886FE114EAFBB1F18892F67FEA2173D8F05B5ED54B67
      key rsa file add 0x0D649530B2392230C9AA2D9974777147DFBCCD2067222A11
      key rsa file add 0x8C49A3D60E22901AC5103313CE5CC0B9FA1A2F1607BC55EE
      key rsa file add 0xA6EB05FB527E786CD4529F1388F6E66AFBFA41234902488E
      key rsa file end 0xB4303ABF65069D25D17145D8695CBD88EC0C92EECC210B36
      cert
; -- Cert user configuration --
         file new USER.CER
         file add 0x308203DB308202C3A003020102020101300D06092A864886F70D010104050030
         file add 0x818F310B3009060355040613025350310F300D060355040813064D6164726964
         file add 0x311430120603550407130B547265732043616E746F7331143012060355040A13
         file add 0x0B54656C64617420532E412E311B3019060355040B131249502054656C657068
         file add 0x6F6E792047726F7570312630240603550403131D546564617420436572746966
         file add 0x69636174696F6E20417574686F72697479301E170D3037303631313135323731
         file add 0x335A170D3137303630383135323731335A3073310B3009060355040613025350
         file add 0x310F300D060355040813064D616472696431143012060355040A130B54656C64
         file add 0x617420532E412E311B3019060355040B131249502054656C6570686F6E792047
         file add 0x726F75703120301E0603550403131753616D706C6520557365722043657274469
         file add 0x666963617465305C300D06092A864886F70D0101010500034B003048024100D7
         file add 0xE5733321E3FFFEBC36011D2AF3A2EE4D2DA995C62E064AD1D625AB451DDB0A7E
         file add 0x2614ED2E832A382607D01CCA2DA1687F9BC9D0EA0F6A4C0C23CCBDFC94463102
         file add 0x03010001A38201233082011F30090603551D1304023000302C06096086480186
         file add 0xF842010D041F161D4F70656E53534C2047656E657261746564204365727466966
         file add 0x6963617465301D0603551D0E0416041423F7EC38E8281BE6E95D1C4D09DFB04D
         file add 0x53909A1F3081C40603551D230481BC3081B980142C24D56B5333A7AFB4718F81
         file add 0x5F76204B28E4AFF4A18195A4819230818F310B3009060355040613025350310F
         file add 0x300D060355040813064D616472696431143012060355040407130B547265732043
         file add 0x616E746F7331143012060355040A130B54656C64617420532E412E311B301906
         file add 0x0355040B131249502054656C6570686F6E792047726F75703126302406035504
         file add 0x03131D54656461742043657274696669636174696F6E20417574686F72697479
         file add 0x82090088C278A06A7220C5300D06092A864886F70D01010405000382010100 7A
         file add 0xABF460D76CEECA2C4B6AE2203F9A1546B6DC646DC54F3D92D8EE57371496AA89
         file add 0x170A6BF836D7A40BD608480B73D53F8C38B27C110532B1B2B8E0967F3BB67DA3
         file add 0x7503EB26B08417D17246190E1D0584F325C1CA691748730AF1B1B9EB6E8F06B6
         file add 0x85F8A4600927F5F68E08D042DEBE97E62500C3D4AE19A3302B085FF572D0E5C8
         file add 0x68C56B6D1923EE9E33A50222A9D48A0515B44C2D324C6CDFB8E1B0F3D5E56FC1
         file add 0xEF618B5898E613E4EFC194D782B8241C1267523A6D8A02449D6AA07A609D4279
         file add 0x1739E27DA61804160075C9617E8D5C585A8F0E0400DD2650FC372EE8F7B22F3D
         file end 0xEEA5B7701DD27E44870E49F1486930B28E6D45874AA62D3F4F1BEFA4EAEF84
;
         certificate USER.CER load
      exit
;
   exit
;
Exit
;
protocol sip
   crypto signaling default user USER.CER
exit
```

2.2.5.6 CRYPTO SIPURI-SCHEME

If this command is configured, the type of URL used in TLS is SIP instead of SIPS. By default, SIPS is used with TLS transport.

Syntax:

```
SIP Config$[no] crypto sipuri-scheme
```

2.2.5.7 CRYPTO SSLV2

Allows you to accept version 2 SSL connections. If this is not enabled, only TLSv1 connections are permitted.

Syntax:

```
SIP Config$[no] crypto sslv2
```

2.2.6 [NO] HANDLE TRANSFER

If this command is configured, the device transfers a call without sending a REFER message via SIP. An INVITE request is sent to the new destination and the transferred end is linked to the new destination. This behavior is only applied to blind transfers. This is disabled by default, meaning the device sends a REFER message to the transferred end to perform blind transfers.

Syntax:

```
SIP Config$[no] handle transfer
```

2.2.7 [NO] HEADERS

Enables the use of some optional headers in the SIP messages being sent.

2.2.7.1 HEADER COPY-ID-NONE

Through this command, you can prevent the *P-Asserted-Identity* and *Remote-Party-ID* headers from being added to the INVITE transaction messages that are sent. This command is disabled by default.

Syntax:

```
SIP Config>headers copy-id-none
```

Command history:

Version	Modification
11.1.3	This command was implemented in version 11.1.3

2.2.7.2 HEADERS COPY-UNKNOWN

This copies the unknown headers in the back-to-back SIP calls when forwarding the SIP messages between both ends of the call. This command is disabled by default.

Syntax:

```
SIP Config$[no] headers copy-unknown
```

2.2.7.3 HEADERS FROM-TO TOLERANT-MATCH

On configuring this command, the criteria used to assign SIP packets to the established dialogs are less strict. Instead of checking that the URL's SIP and From and To header tags coincide, it only checks that the tags match. This command is disabled by default.

Syntax:

```
SIP Config$[no] headers from-to tolerant-match
```

2.2.7.4 HEADERS P-ASSERTED-ID

If this option is configured, it adds the P-Asserted-Identity header to the INVITE transaction messages that are sent. The User Agent identity sending the messages is specified in this field. This command is disabled by default.

Syntax:

```
SIP Config$[no] headers p-asserted-id
```

2.2.7.5 HEADERS P-PREFERRED-ID

If this option is configured, it adds the P-Preferred-Identity header to the INVITE transaction messages that are sent. The aim of this field is to report the preferred identity of the User Agent that sends the messages. This command is disabled by default.

Syntax:

```
SIP Config$[no] headers p-preferred-id
```

2.2.7.6 HEADERS REASON Q850

If this option is configured, the Reason header is included in the BYE and CANCEL messages sent, as well as in the SIP error responses. The cause of the call release is also included in this header in Q.850 format. This command is disabled by default.

Syntax:

```
SIP Config$[no] headers reason q850
```

2.2.7.7 HEADERS REMOTE-PARTY-ID

When this option is configured, the Remote-Party-ID header is inserted within the INVITE messages sent. This field aims to report the preferred identity of the User Agent sending the messages. This command is disabled by default.

Syntax:

```
SIP Config$[no] headers remote-party-id
```

Command history:

Version	Modification
11.0.3	This command was implemented in version 11.0.3

2.2.8 [NO] HOLD

Configures the way in which SIP calls are held. The possibilities are *inactive, RFC2543* and *RFC3261*. The configuration is global and applies to all SIP calls that do not have a specific configuration for the **sip-hold** command in the dial-peer used in the call (please see manual bintec-Dm722-I Telephony Over IP). Default is *RFC3261* **.**

2.2.8.1 HOLD INACTIVE

The device uses the *inactive* SDP attribute when sending a Re-INVITE to notify the other end that you want to hold the call.

Syntax:

```
SIP Config$hold inactive
```

2.2.8.2 HOLD RFC2543

When sending a Re-INVITE to notify the other end that you want to hold the call, the device places the audio 0.0.0.0 IP address in the SDP. This is the method described in the SIP specification under RFC2543.

Syntax:

```
SIP Config$hold rfc2543
```

2.2.8.3 HOLD RFC3261

The device uses the *sendonly* SDP attribute when sending a Re-INVITE to notify the other end that you want to hold the call. This is the method recommended in the latest SIP specification (RFC3261 and RFC3264). It is the default mode used by the device.

Syntax:

```
SIP Config$hold rfc3261
```

2.2.9 [NO] IP-TOS

Allows you to configure the TOS in the SIP packets sent by the device. The complete TOS byte is configured through its hexadecimal value. Default is 0.

Syntax:

```
SIP Config$ip-tos ?
  <hex 0x0..0xff>    Hexadecimal value in the specified range
```

2.2.10 [NO] KEEP-ALIVE

When the device has a call established, it periodically sends a petition to check if the other end also holds said call. If the configured consecutive polls fail, the call is canceled. Through this command, you can configure the type of request to be used in this process.

You can send two types of packets to check if the call is active: INFO or UPDATE. INFO is the default value. You can also alternate both packets through the **both** command. You can disable the keep-alive mechanism through the **keep-alive none command**. The **max-errors** subcommand allows you to configure the number of consecutive polls that must fail in order to cancel the call.

Syntax:

```
SIP Config$[no] keep-alive ?
  none       Disable keep-alive
  both        Use info and update SIP packets
  info       Use info SIP packets
  max-errors  Maximum number of keepalive errors before hangup
  update     Use update SIP packets
```

2.2.11 [NO] LOCAL-REGISTRAR

Allows you to configure various options related to the device’s Registrar server operations.

2.2.11.1 LOCAL REGISTER USER CHECK

When you configure this command, the router’s Registrar server checks the dial-peers configuration from the *telephony* menu to decide whether it admits the SIP telephone registration or not. This check only occurs when there is no active external proxy and the router itself responds to the registers.

In order to admit a SIP register, there must be a sip dial-peer with *target dynamic*, whose *destination-pattern* or *destination-alias*, fully coincides with the user trying to register. If, additionally, the **password** command is configured, authentication is requested from the SIP terminal and the credentials are checked.

This command is disabled by default, thereby allowing any SIP to register without a user or password check.

Example:

This is the configuration required so only SIP terminal registrations, whose user is 100 with password mypassword, are admitted.

```
   telephony
; -- Telephony configuration –
      dial-peer 1 sip
         destination-pattern 100
         password mypassword
         target dynamic
      exit
   exit
   protocol sip
; -- SIP protocol configuration --
      application address 192.168.212.174
      application server default
      local-registrar user-check
   exit
```

2.2.11.2 LOCAL REGISTRAR BIND TO HEADER

On configuring this command, the device's Registrar server uses the user part received in the register messages *To* header to save it in the registered user database. By default, the one received in the *Contact* header is saved instead.

Syntax:

```
SIP Config$local-registrar bind-to-header
```

Command history:

Version	Modification
11.0.4	The **bind-to-header** option has been implemented in version 11.0.4
11.1.0	The **bind-to-header** option has been implemented in version 11.1.0

2.2.12 [NO] MAP-CAUSE

Allows you to configure the conversion between the release causes used in the device's conventional telephony ports (Q.850 causes) and the SIP response codes. By default, the most common causes are converted following the *RFC 3398 Integrated Services Digital Network (ISDN) User Part (ISUP) to Session Initiation Protocol (SIP) Mapping* recommendations. The default conversion tables in both directions are given below.

Q.850 to SIP Conversion:

	Q.850 Cause		SIP Response Code
1	Unallocated (unassigned) number	404	Not found
17	User busy	486	Busy here
18	No user responding	408	Request timeout
19	No answer from user	480	Temporarily unavailable
21	Call rejected	403	Forbidden
28	Address incomplete	484	Address incomplete
31	Normal, unspecified	480	Temporarily unavailable
34	No circuit/channel available	503	Service unavailable
38	Network out of order	503	Service unavailable
88	Incompatible destination	503	Service unavailable
102	Recovery on timer expiry	408	Request timeout
	The rest of the causes are not listed.	500	Internal Server Error

SIP to Q.850 Conversion:

	SIP Response Code		Q.850 Cause
403	Forbidden	21	Call rejected
401	Unauthorized	21	Call rejected
404	Not found	1	Unallocated (unassigned) number
407	Proxy authentication required	2l	Call rejected
408	Request timeout	102	Recovery on timer expiry
480	Temporarily unavailable	18	No user responding
484	Address incomplete	28	Address incomplete
486	Busy here	17	User busy
600	Busy Everywhere	17	User busy
603	Decline	21	Call rejected
604	Does Not Exist Anywhere	1	Unallocated (unassigned) number
	The rest of the causes are not listed.	31	Normal, unspecified

2.2.12.1 MAP-CAUSE SIP <Q.850 cause> sip <sip cause>

Configures the SIP code that the indicated Q.850 release cause should be converted to.

Syntax:

```
SIP Config$[no] map-cause pstn <1..127> sip <400..699>
```

2.2.12.2 MAP-CAUSE SIP <sip cause> PSTN <Q.850 cause>

Configures the Q.850 cause to convert a given SIP error code.

Syntax:

```
SIP Config$[no] map-cause sip <400..699> pstn <1..127>
```

2.2.13 [NO] MAP-PROGRESS PSTN

Configures mapping between progress messages received in calls going out through a traditional telephony interface, towards PSTN, and the SIP progress codes sent to the call source. Below, you can see the default conversion list in cases where this command isn't configured:

PSTN progress message	SIP response code
PROGRESS without IE Progress Indication (in-band audio)	Ignored, nothing is sent.
PROGRESS with IE Progress Indication (in-band audio)	183 with SDP for in-band audio.
ALERT without IE Progress Indication (in-band audio)	180 without SDP.
ALERT with IE Progress Indication (in-band audio)	180 with SDP for in-band audio.

Syntax:

```
SIP Config$[no] map-progress pstn
      progress                Progress without in-band audio
        sip                   SIP progress message
            <101..200>        Value in the specified range
               [add-sdp]      Add SDP in answer for Early Media
            Ignore            Ignore progress message
      alert                   Alerting without in-band audio
        sip                   SIP progress message
            <101..200>        Value in the specified range
               [add-sdp]      Add SDP in answer for Early Media
            Ignore            Ignore progress message
      pi-progress             Progress with in-band audio
        sip                   SIP progress message
            <101..200>        Value in the specified range
               [add-sdp]      Add SDP in answer for Early Media
            Ignore            Ignore progress message
      pi-alert                Alerting with in-band audio
        sip                   SIP progress message
            <101..200>        Value in the specified range
               [add-sdp]      Add SDP in answer for Early Media
            Ignore            Ignore progress message
```

For each type of progress message received through the PSTN interface, you can configure the SIP code to send between **101** and **200** (or not send anything using the **ignore** option). Additionally, you can add SDP to have in-band audio in the call (using the **add-sdp** option). If you configure the **200** code so that the call is established by SIP, SDP is always sent and subsequent progress and establishment messages are ignored.

2.2.14 [NO] MAP-PROGRESS SIP

Allows you to configure the conversion between the progress codes received in outgoing calls through SIP, and the progress messages sent that originated the call from a traditional telephony interface. The default conversion list is shown below; if this command isn't configured, the unlisted SIP causes are ignored.

SIP response code	PSTN progress message
183 with SDP for in-band audio	PROGRESS with IE Progress Indication (in-band audio)
183 without SDP	PROGRESS without IE Progress Indication (in-band audio)
180 with SDP for in-band audio	ALERT with IE Progress Indication (in-band audio)

180 without SDP	ALERT without IE Progress Indication (in-band audio)

Syntax:

```
SIP Config$[no] map-progress sip
      <101..199>              Value in the specified range
         pstn                 PSTN progress type
            progress          Progress without in-band audio
            alert             Alerting without in-band audio
            pi-progress       Progress with in-band audio
            pi-alert          Alerting with in-band audio
            ignore            Ignore progress message
```

For each SIP code, you can configure the type of progress message sent by the PSTN interface, or not send anything using the **ignore** command.

2.2.15 [NO] MAX-EXPIRES

Use this command to configure the maximum length of time, in seconds, a UA can be registered in the database for the users registered in the device. This time is only applicable when the router is behaving as Registrar, responding to the registers. This happens when it is a server and the configured proxy is not available. Default value is 3600 seconds.

Syntax:

```
SIP Config$[no] max-expires  <1..65535>    Value in the specified range
```

2.2.16 [NO] MAX-FORWARDS

Use this command to configure the max-forwards field value for the SIP messages sent by the router. This field defines the maximum number of hops a SIP packet can take before being dropped. Default value is 70.

Syntax:

```
SIP Config$[no] max-forwards <1..200>    Value in the specified range
```

2.2.17 [NO] OVERLAP-DIALING

This enables the support for overlap dialing from a SIP telephone. This system allows the caller to dial more digits when the called number does not match a dial-peer destination pattern, but could match if more digits are entered. This behavior is disabled by default, and the error message “404 Not Found” is returned when the called number does not match the dial plan.

There are two overlap dialing methods in SIP:

The first of these is triggered by the '484-response' option, to which the “484 Address Incomplete” status code is returned when the called number does not match a dial-peer destination pattern (but could match if more digits were entered).

The second method is triggered by the 'kpml' option, which uses a system based on RFC 4730 to perform overlap dialing. With this system, when the called number does not match a dial-peer destination pattern, but could match if more digits were added, a subscribe dialogue is initiated with the calling phone to receive the remaining digits. Once a number matches a dial-peer (or if there is no match), the subscription is terminated and the call is placed regularly where applicable.

Syntax:

```
SIP Config$[no] overlap-dialing ?
  484-response    Overlap dialing method using 484 responses
  kpml            KPML overlap dialing method
  <cr>            Overlap dialing method using 484 responses
```

Command history:

Version	Modification
11.0.3	'484-response' and 'kpml' options have been implemented in the command

2.2.18 [NO] PASSWORD

Password used by the SIP protocol when authentication is required. If there is a password configured in the dial-peer, this will be used. Otherwise, this global password is used instead.

Syntax:

```
SIP Config$[no] password <string>
```

2.2.19 [NO] PROXY

Use this command to configure an external SIP server. The device will use this server to establish calls when these have, as an outgoing dial-peer, one configured with target sip-proxy or, with the server mode activated, when an incoming SIP call cannot find an outgoing dial-peer. The proxy is also used to send its registers when a Registrar has not been configured. More than one proxy can be configured (in which case the device will use the first one that is active).

Syntax:

```
SIP Config$[no] proxy <proxy-id>
  default             Set this entry to its default values
  track               Track this entry
    nsla-advisor              Set the nsla advisor to track
    registrations             Track sip phone registrations
  port                Specify the port the host is listening to
  transport           Set the transport protocol of this proxy
    udp                  Use UDP to communicate with this proxy
    tcp                  Use TCP to communicate with this proxy
    tls                  Use TLS to communicate with this proxy
    system               Use global sip configuration to communicate with this proxy
```

proxy-id	Proxy IP or domain name.
default	Returns all the values associated with this entry to their default values.
track nsla-advisor	The proxy will only be active if the associated nsla entry is active.
track registrations	The proxy is considered down if the SIP registers sent by the device to the proxy do not receive any response. If two tracks are configured simultaneously, the proxy will be considered down should either of them fail. A logical AND is applied to the result of each track.
port	SIP messages destined for this proxy must be sent to this port.
transport udp	The proxy uses the udp transport protocol.
transport tcp	The proxy uses the tcp transport protocol.
transport tls	The proxy uses the tls transport protocol.
transport system	The proxy uses the transport protocol configured at the SIP protocol global level through the transport command.

2.2.20 [NO] REALM

Configures the SIP domain used by the router. If this is not configured, the **registrar** command or the active proxy are used.

Syntax:

```
SIP Config$[no] realm <word>    Text
```

2.2.21 [NO] REGISTER

When the device acts as server, it sends the register requests to the Registrar (or to the active proxy if there is no Registrar configured). If the gateway is enabled, this also registers the voice port or group dial-peers.

Syntax:

```
SIP Config$[no] register           Enable SIP registration of peer numbers
```

2.2.22 [NO] REGISTRAR

Configures the IP address or the SIP domain name.

Registrar where the register messages are sent. If neither is configured, the messages are sent to the active proxy. If there is no proxy, the router, in server mode, acts as Registrar.

Syntax:

```
SIP Config$[no] registrar <word>    Set ip address or dns name for this host
```

2.2.23 [NO] RPORT

Enables the rport functionality described in the RFC3581. The SIP responses are returned to the port where the SIP messages arrived, instead of being sent to the port indicated in the SIP header. This functionality is useful in cases where there is NAT in the network. The **no rport** command disables this functionality. Default is disabled.

Syntax:

```
SIP Config$[no] rport              Use SIP rport defined in RFC3581
```

2.2.24 [NO] RPR

Configures the reliable provisional responses support, in accordance with RFC 3262. It retransmits the provisional responses (with a code greater than 100 and lower than 200) to an INVITE, while the remote end does not confirm reception for these by sending the PRACK method.

2.2.24.1 [NO] RPR UNSUPPORTED

Configures the device so that it doesn't support reliable provisional responses. This is the default configuration.

Syntax:

```
SIP Config$[no] rpr unsupported    Reliable provisional responses unsupported
```

2.2.24.2 [NO] RPR SUPPORTED

Configures the device to support the exchange of reliable provisional responses. With this configuration, if the remote end of the SIP dialog requests reliable provisional response exchanges, call establishment is supported in accordance with RFC 3262.

Syntax:

```
SIP Config$[no] rpr supported    Reliable provisional responses supported
```

2.2.24.3 [NO] RPR REQUIRED

Configures the device so that it requires the remote end of the SIP dialog to use reliable provisional responses.

Syntax:

```
SIP Config$[no] rpr required    Reliable provisional responses required
```

2.2.25 [NO] SDP

Helps configure advanced options of the SDP functionality when it is used as a negotiation protocol for SIP.

2.2.25.1 [NO] SDP 1xx-ignore

If this is configured, the SDP protocol received in the provisional SIP responses (with codes between 100 and 199) is ignored, behaving as if the SDPs are not included. By default, this command is deactivated and the SDP in the provisional responses is processed, allowing in-band audio to be set before establishing the calls.

Syntax:

```
SIP Config$[no] sdp 1xx-ignore
```

2.2.25.2 [NO] SDP CHECK-SESSVERSION

Specifies if, during the course of a single session, the device should check that the number for the received SDP session version increases when a new SDP message is received. This is active by default. Disabling it may prove useful when operating with devices that do not comply with the RFC, since they do not increase the session number in a Re-INVITE.

Syntax:

```
SIP Config$[no] sdp check-sessversion
```

2.2.25.3 [NO] SDP LAST-MEDIA-UNHOLD

When the device sends a Re-INVITE to recover a call on hold, it creates a new SDP offer with all the codecs supported in the call depending on the dial-peers' configuration, carrying out a new medium negotiation. If the command is configured, the Re-INVITE includes an SDP with the same medium negotiation before holding the call. It may be necessary to configure this command in order to interoperate with a manufacturer that doesn't support new negotiation on recovering a call. However, this means that some call transfers can fail where it's necessary to renegotiate the PSTN codec or address. This command is not configured by default.

Syntax:

```
SIP Config$[no] sdp last-media-unhold
```

2.2.25.4 [NO] SDP PTIME

Enables the ptime negotiation, which is an SDP Media Attribute that indicates the required time interval (expressed in milliseconds) between two consecutive RTP packets. In the SDP messages sent, the value for this field is calculated from the number of frames per RTP packet, configured in the dial-peers that intervene in the call. When this field is received in an SDP message from the remote end, the number of frames to encapsulate in each RTP packet that is going to be sent is extracted. This is disabled by default, meaning it isn't included in the SDP messages sent and ignored if received.

Syntax:

```
SIP Config$[no] sdp ptime
```

2.2.26 NO SIP

Deletes all the SIP protocol configuration.

Syntax:

```
SIP Config$ no sip        Clear all SIP configuration
```

2.2.27 [NO] STUN

Configures the use of the STUN protocol in SIP calls. As a parameter, you need to indicate the STUN domain (which must be previously configured).

Syntax:

```
SIP Config$[no] stun <id>     Value in the specified range
```

2.2.28 [NO] TCP-MSS

Configures the maximum segment size for TCP connections in bytes. Default value is 1280 bytes.

Syntax:

```
SIP Config$[no] tcp-mss <536..65535>    Value in the specified range
```

2.2.29 [NO] TIMERS

Configures the diverse timers affecting the running of the SIP protocol.

2.2.29.1 TIMERS B

Configures the maximum time the device waits between the start of an INVITE transaction and the receipt of an answer from the remote end. If this times out without a response, the call fails. This is configured in seconds. Default value is 32 times the T1 timer value. Therefore, default for all timers is 16 seconds.

Syntax:

```
SIP Config$[no] timers b <1s..1m4s>    Time value
```

2.2.29.2 TIMERS D

Time that a successfully completed client INVITE transaction waits (after the ACK is sent) before passing to a terminated state.

This is configured in seconds. Default is 32 seconds.

Syntax:

```
SIP Config$[no] timers d <16s..1m>    Time value
```

2.2.29.3 TIMERS F

Configures the maximum time the device waits between the beginning of a no INVITE transaction (e.g., REGISTER) and the reception of a response from the remote end. If this times out without a response, the transaction has failed. This is configured in seconds. Default is 32 times the value of the T1 timer. Therefore, default for all timers is 16 seconds.

Syntax:

```
SIP Config$[no] timers f <1s..1m4s>    Time value
```

2.2.29.4 TIMERS H

Time during which a server retransmits the end response of an INVITE transaction without having received the client ACK transaction.

This is configured in seconds. Default is 64*T1 seconds. Therefore, default for all timers is 32 seconds.

Syntax:

```
SIP Config$[no] timers h <1s..1m4s>    Time value
```

2.2.29.5 TIMERS J

Time a server waits before considering a no INVITE transaction as terminated. During this time, the response is retransmitted if a retransmission of the initial message is received.

This is configured in seconds. Default is 64*T1 seconds. Therefore, default for all timers is 32 seconds.

Syntax:

```
SIP Config$[no] timers j <1s..1m4s>    Time value
```

2.2.29.6 TIMERS KEEPALIVE

Defines the time between two keep-alive packets. Default is 1 hour.

Syntax:

```
SIP Config$[no] timers keep-alive <30s.. 1d>    Time value
```

2.2.29.7 TIMERS NO-ANSWER

Configures the time that the SIP protocol waits for a definite response to a transaction after receiving a provisional response. In the typical case of an INVITE transaction, this is the maximum time the telephone will be ringing without the called user responding.

This is configured in seconds. Default is 180 seconds.

Syntax:

```
SIP Config$[no] timers no-answer <1s..5m>    Time value
```

2.2.29.8 TIMERS REGISTER

Defines the time between registers when voice-port or group dial-peers are registered in the Registrar or in the active proxy (if there is no Registrar configured).

This is configured in seconds. Default is 3600 seconds.

Syntax:

```
SIP Config$[no] timers register     Set register expiry time
```

2.2.29.9 TIMERS T1

Defines the minimum time the SIP protocol waits to receive a response before resending the request. Each time the request is resent, the wait time base T1 duplicates, until it reaches the maximum T2 value.

This is configured in milliseconds. Default is 500 milliseconds.

Syntax:

```
SIP Config$[no] timers t1 <1..20>    Timer value in 1/10secs.
```

2.2.29.10 TIMERS T2

Defines the maximum time the SIP protocol waits to receive a response before resending the request. Each time the request is resent, the wait time base T1 duplicates, until it reaches the maximum T2 value.

This is configured in seconds. Default is 4 seconds.

Syntax:

```
SIP Config$[no] timers t2 <1s..10s>    Time value
```

2.2.29.11 TIMERS T4

Time that a successfully completed transaction waits, once the ACK has been received, before passing to a terminated state. This applies to all transactions except for the client INVITE transaction, which uses timer D.

This is configured in seconds. Default value is 5 seconds.

Syntax:

```
SIP Config$[no] timers t4 <1s..10s>    Time value
```

2.2.29.12 TIMERS USE-T2-INVITE

Changes the behavior of the INVITE client transaction so the retransmission interval is never greater than the T2 time. Normal behavior is that the interval between retransmissions duplicates for each retransmission. T2 is the maximum interval between retransmissions for all transactions distinct to INVITE complying with the RFC3261 norm. Configuring this command means that T2 is also used as a limit in the INVITE transactions.

Syntax:

```
SIP Config$[no] timers use-t2-invite  Use T2 (max retransmission interval) for INVITE
```

2.2.30 [NO] TRANSPORT

Allows you to select the type of transport used by SIP for outgoing calls and registers, if the dial peers or the proxy configurations do not indicate another type of transport. The supported protocols are TCP, UDP and TLS. The default protocol is UDP. However, the server accepts all three types. To accept TLS connections, you must have a user certificate configured through the **crypto signaling default user** command.

2.2.30.1 TRANSPORT UDP

The protocol used is UDP.

Syntax:

```
SIP Config$protocol udp
```

2.2.30.2 TRANSPORT TCP

The protocol used is TCP.

Syntax:

```
SIP Config$protocol tcp
```

2.2.30.3 TRANSPORT TLS

The protocol used is TLS over TCP. To accept TLS connections, you must have a user certificate configured through the **crypto signaling default user** command.

Syntax:

```
SIP Config$protocol tls
```

2.2.31 [NO] UPDATE LEVEL-INDICATOR

Configures the updating for a level indicator, an element that can be associated with an NSLA filter (please see manual bintec-Dm754-I Network Service Level Advisor). The level indicator indexed by <id> is modified by a whole value <val> depending on the SIP registrations: if a SIP registration petition sent to an external server has a positive response, the level indicator increases by <val>; when the registration petitions do not receive a response, this value is subtracted.

The aim of this command is to have connectivity information at the SIP level with an external server in an NSLA poll; this poll can then be used, for example, so the local server automatically deactivates when the external server is accessible.

Syntax:

```
SIP Config$[no] update level-indicator <id> value <val> when-registrations-ok
```

Chapter 3 Monitoring

3.1 Accessing the Monitoring Menu

The SIP protocol monitoring commands must be entered in the monitoring menu associated with the SIP (*SIP+*). To access said menu, use the **protocol SIP** command from the general monitoring menu (+).

```
+protocol sip
SIP Mon
SIP Mon+
```

Once you have accessed the protocol SIP monitoring menu, you can introduce the following commands:

3.2 Monitoring Commands

3.2.1 CLEAR

3.2.1.1 CLEAR SIP-STATISTICS

Resets all the counters relative to the SIP packets to zero.

Syntax:

```
SIP Mon+clear sip-statistics
```

3.2.1.2 CLEAR UA-STATISTICS

Resets all the counters relative to the SIP calls to zero.

Syntax:

```
SIP Mon+clear ua-statistics
```

3.2.2 LIST

Lists information on the SIP protocol.

3.2.2.1 LIST ALL

Lists everything related to the SIP protocol.

Syntax:

```
SIP Mon+list all
```

Example:

```
SIP Mon+list all
```

3.2.2.2 LIST BACK-TO-BACK

Lists the information relative to the back-to-backup functionality (specifically on the active proxy, if there is one) and on the existing back-to-back transactions, as well as their state.

Syntax:

```
SIP Mon+list back-to-back
```

Example:

In the following example, you can see that the configured proxy is active, as well as an established back-to-back call.

```
SIP Mon+list back-to-back
Active proxy: sipserver.id.bintec.es

Leg Id: 876 State: 12 - Leg Id: 877 State: 12
```

```
SIP Mon+
```

3.2.2.3 LIST REGISTERED-USERS

Displays the users registered in the router as well as the rest of the register time.

Syntax:

```
SIP Mon+list registered-users
```

Example:

In the following example, you can see that there are two registered users (1202 and 1201) belonging to the asterisk.id.bintec.es domain. For each of these, their current contact address and the time left until the register expires are displayed.

```
SIP Mon+list registered-users
Dumping registered users, time since last purge 12

Locations for Registered User 1202@id.bintec.es:
   1202@172.24.100.131:5060 ttl:28

Locations for Registered User 1201@id.bintec.es:
   1201@172.24.100.130:5061 ttl:8

SIP Mon+
```

3.2.2.4 LIST SIP-STATISTICS

Displays information on the received/sent SIP packets.

Syntax:

```
SIP Mon+list sip-statistics
```

Example:

```
SIP Mon+list sip-statistics
```

3.2.2.5 LIST SSL-SESSIONS

Displays information on the SIP active ssl sessions.

Syntax:

```
SIP Mon+list ssl-sessions
```

3.2.2.6 LIST UA-STATISTICS

Displays information on the received/executed calls.

Syntax:

```
SIP Mon+list ua-statistics
```

Example:

```
SIP Mon+list ua-statistics
```

3.2.3 EXIT

Allows you to return to the general monitoring menu.

Syntax:

```
SIP Mon+exit
```

Chapter 4 Examples

4.1 SIP Server in Emergency Mode

A company wishes to install SIP telephones in all their offices so calls between their branches and their headquarters, between two of their branches, or between their branches and any exterior telephone number, are routed through a SIP central server, with dns id.bintec.es.

The company also wants, in cases where IP connectivity between the branch and the central drops or when there is excessive delay, the SIP telephones to carry on working in order to execute internal calls and that external calls are routed through an external line connected to a voice gateway. This gateway executes the SIP-POTS conversion.

Below, a diagram has been included representing a branch and the central office:

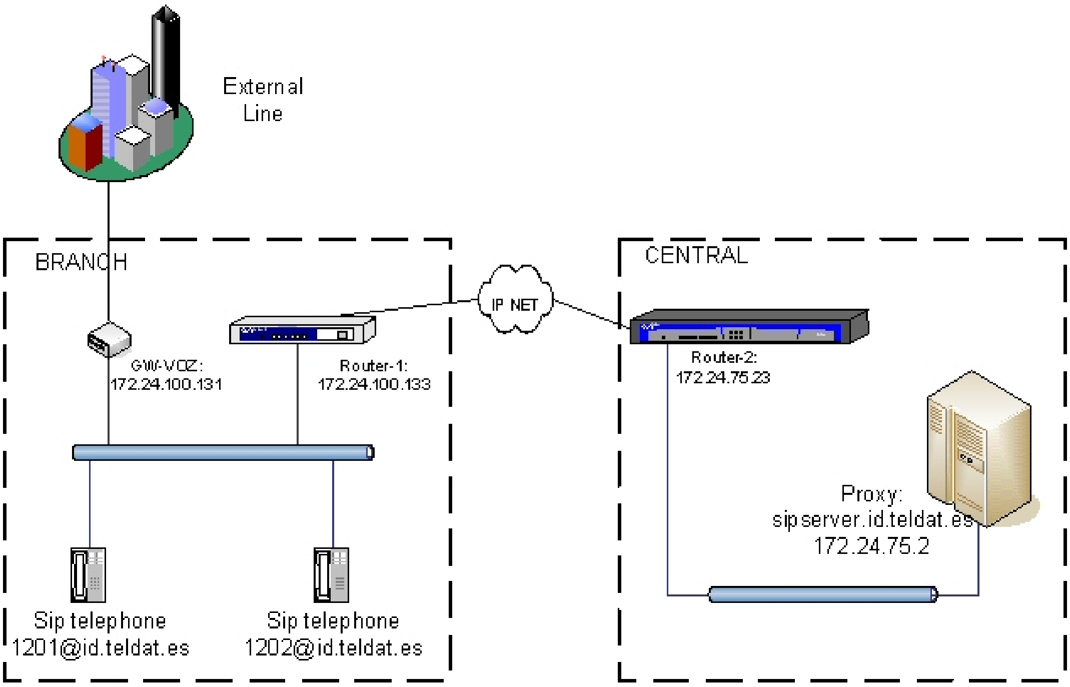

Fig. 1: SIP Server in Emergency Mode

To monitor the IP connection quality, an nsm poll is configured to monitor the delay between GW-1 and GW-2. If this delay surpasses 300 milliseconds, the proxy is considered unreachable and the dial-peer activates to route all external calls via GW-VOZ, said external calls take patterns 9........ or 6......... corresponding to landlines and mobile numbers respectively. When this dial-peer is inactive the external calls cannot find an outgoing dial-peer and consequently are sent to the proxy.

Configuration:

Below, you can see the GW-1 device configuration. This device behaves as backup should the central server go down.

The WAN line configuration has not been detailed, since it depends on the type of interface used (Frame-Relay, ADSL, etc.). You need to bear in mind that, in this configuration and in order to measure the connection quality, bandwidth reservation must be enabled to ensure that both SIP traffic and the traffic generated by the poll are given priority treatment.

You need to configure two dial-peers, the first should only be active when the SIP proxy goes down and said dial-peer consequently routes the external calls over GW-VOZ.

The second dial-peer targets the SIP proxy, and consequently is only active when the proxy is active. It is in charge of routing all the calls between two LAN telephones (numbers with pattern 12..) to the proxy.

Please note that if this dial-peer is not present, the calls between the two LAN telephones are not routed to the proxy (even if it is active). In this case however, the GW-1 itself acts as the Back-to-Back server. If this is the behavior required, simply eliminate said dial-peer.

```
; Showing System Configuration ...
```

```
log-command-errors
no configuration
telephony
; -- Telephony configuration –
   dial-peer 1 sip
      destination-pattern 6........
      destination-pattern 9........
      target ipv4 172.24.100.131
      track nsla 1
   exit
;
   dial-peer 3 sip
      destination-pattern 12..
      target sip-proxy
   exit
exit
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.24.100.133 255.255.255.248
;
   exit
;
protocol ip
; -- Internet protocol user configuration --
   internal-ip-address 172.24.100.133
;
exit
;
protocol sip
;
   application address 172.24.100.133
   application server default
   proxy id.bintec.es default
   proxy id.bintec.es track nsla-advisor 2
;
   realm id.bintec.es
exit
;
feature dns
; -- DNS resolver user configuration --
   server 172.24.51.36
   no cache enable
exit
;
feature nsm
; -- Network Service Monitor configuration --
   operation 1
; -- NSM Operation configuration --
      type echo ipicmp 172.24.75.23
      frequency 5
      timeout 1000
   exit
;
   schedule 1 start-time now
exit
;
feature nsla
; -- Feature Network Service Level Advisor --
   enable
;
   filter 1 nsm-op 1 rtt
   filter 1 significant-samples 1
   filter 1 activation threshold 300
   filter 1 activation sensibility 80
   filter 1 activation stabilization-time 25
   filter 1 deactivation threshold 200
   filter 1 deactivation sensibility 80
   filter 1 deactivation stabilization-time 25
;
   alarm 1 filter-id 1
;
   advisor 1 alarm-id 1
;
   advisor 2 not alarm-id 1
;
exit
;
dump-command-errors
end
; --- end ---
```

4.2 SIP Server with TLS and SRTP Encryption

In the previous example, we want to use TLS/SRTP encryption to encrypt conversations between the telephones and the media gateways. To do this, you need to configure the transport mode as tls and load two certificates (i.e., the bintecca.cer and the user.cer).

The bintecca.cer is the certification authority in charge of verifying that the certificates sent by the devices that want to connect to the router through TLS are permitted. These certificates must be signed by the bintecca.cer certification authority.

The user.cer is the user certificate used by the router when a TLS client or server asks it to authenticate. This certificate is signed by the bintecca.cer certification authority. For the user.cer certificate to be acknowledged as valid, the bintecca.cer certificate must be configured in said clients/servers as belonging to a valid certification authority.

Additionally, you need to configure srtp in fallback mode. This way, if the conversation cannot be established through srtp, it will start without encryption.

```
; Showing System Configuration ...

log-command-errors
no configuration
telephony
; -- Telephony configuration –
   srtp mode fallback
   dial-peer 1 sip
      destination-pattern 6........
      destination-pattern 9........
      target ipv4 172.24.100.131
      track nsla 1
   exit
;
   dial-peer 3 sip
      destination-pattern 12..
      target sip-proxy
   exit
exit
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.24.100.133 255.255.255.248
;
   exit
;

protocol ip
; -- Internet protocol user configuration --
   internal-ip-address 172.24.100.133
;
      ipsec
; -- IPSec user configuration --
         key rsa file add 0x341DC332F23BD75492E8583A82F10A8CFCA4349F531563EF
         key rsa file add 0xCA002248A79CFCFE24FBBCE0FA9C3BF99B02C316C9C5EB44
         key rsa file add 0xA2630B28F04183766A79C93BD967380425955159D32B7035
         key rsa file add 0x448A8DB2954FA8E53132CDAFE7365FED6BAE5CA55ED8809E
         key rsa file add 0x89191F4762B0850603BE8A61AFD3CC786EC5ED1EB08F191C
         key rsa file add 0xCF7B8FD6AF0C37BDF33BEC201C6B58B1FAC419DF8F68F525
         key rsa file add 0x31F285C22AD9896587FE9095A8355C6F35075CDFAE7E2485
         key rsa file add 0x6E75FC669194A00DED45B8AFDF3B99B6A162F7FE14B0F3B8
         key rsa file add 0xDC0E4D442F9EC916D05F161BABA2D3D803AABADF64A8E6EC
         key rsa file add 0x1CBD2973DC158D77A872B75FE4E99277E877E49C117FC6D9
         key rsa file add 0xC8E29F6D1D84030B955E1A6A15E2F15386CA5F6598148876
         key rsa file add 0x0C27B15CF5D812C2922706CF25C7D42DE09DCB4330125C2B
         key rsa file add 0x911BC4C084B9ADE1D6D5B7DDDDD030692F2EC9E66E0E7D74
         key rsa file add 0x782F99AC347A1BCAC42CEBCEF011F1D25646465BED83ABE9
         key rsa file add 0xBCC82DFBE5BA79E4D024AA7F6AB05D03101335AD37882784
         key rsa file add 0xF5F01429118037178894D1823871D8F498F9B2B5C1EB488D
         key rsa file add 0x9D6349C59E11A617F5622FFC33D8D6272D7C13C6F9B494CE
         key rsa file add 0x3148B09CB12A6B438EC87E272FBC4A0C629CD9DDCAC1B9A3
         key rsa file add 0x8DFEC5C76588A09F6417A94ACF76313C89198FE0D7B5E1B2
         key rsa file add 0x02249303A5A3731A0162B207950C41E18A3952DC84415084
         key rsa file add 0x41E09EF3908BD169243C9A611D1EA318FCEF7A2BD0378108
         key rsa file add 0xFD2E886FE114EAFBB1F18892F67FEA2173D8F05B5ED54B67
         key rsa file add 0x0D649530B2392230C9AA2D9974777147DFBCCD2067222A11
         key rsa file add 0x8C49A3D60E22901AC5103313CE5CC0B9FA1A2F1607BC55EE
         key rsa file add 0xA6EB05FB527E786CD4529F1388F6E66AFBFA41234902488E
         key rsa file end 0xB4303ABF65069D25D17145D8695CBD88EC0C92EECC210B36
         cert
; -- Cert user configuration --
            file new SAMPLE.CER
            file add 0x308203DB308202C3A003020102020101300D06092A864886F70D010104050030
            file add 0x818F310B3009060355040613025350310F300D060355040813064D6164726964
            file add 0x311430120603550407130B547265732043616E746F7331143012060355040A13
            file add 0x0B54656C64617420532E412E311B3019060355040B131249502054656C657068
            file add 0x6F6E792047726F7570312630240603550403131D54656C6461742043657274696966
            file add 0x69636174696F6E20417574686F72697479301E170D3037303631313135323731
            file add 0x335A170D3137303630383135323731335A3073310B3009060355040613025350
            file add 0x310F300D060355040813064D616472696431143012060355040A130B54656C64
            file add 0x617420532E412E311B3019060355040B131249502054656C6570686F6E792047
            file add 0x726F75703120301E0603550403131753616D706C652055736572204365727469
            file add 0x666963617465305C300D06092A864886F70D0101010500034B003048024100D7
            file add 0xE5733321E3FFFEBC36011D2AF3A2EE4D2DA995C62E064AD1D625AB451DDB0A7E
            file add 0x2614ED2E832A382607D01CCA2DA1687F9BC9D0EA0F6A4C0C23CCBDFC94463102
            file add 0x03010001A38201233082011F30090603551D1304023000302C06096086480186
            file add 0xF842010D041F161D4F70656E53534C2047656E657261746564204365727466966
            file add 0x6963617465301D0603551D0E0416041423F7EC38E8281BE6E95D1C4D09DFB04D
            file add 0x53909A1F3081C40603551D230481BC3081B980142C24D56B5333A7AFB4718F81
            file add 0x5F76204B28E4AFF4A18195A4819230818F310B3009060355040613025350310F
            file add 0x300D060355040813064D616472696431143012060355040713OB547265732043
            file add 0x616E746F7331143012060355040A130B54656C64617420532E412E311B301906
            file add 0x0355040B131249502054656C6570686F6E792047726F75703126302406035504
            file add 0x03131D54656461742043657274696669636174696F6E20417574686F72697479
            file add 0x82090088C278A06A7220C5300D06092A864886F70D01010405000382010100 7A
            file add 0xABF460D76CEECA2C4B6AE2203F9A1546B6DC646DC54F3D92D8EE57371496AA89
            file add 0x170A6BF836D7A40BD608480B73D53F8C38B27C110532B1B2B8E0967F3BB67DA3
            file add 0x7503EB26B08417D17246190E1D0584F325C1CA691748730AF1B1B9EB6E8F06B6
            file add 0x85F8A4600927F5F68E08D042DEBE97E62500C3D4AE19A3302B085FF572D0E5C8
            file add 0x68C56B6D1923EE9E33A50222A9D48A0515B44C2D324C6CDFB8E1B0F3D5E56FC1
            file add 0xEF618B5898E613E4EFC194D782B8241C1267523A6D8A02449D6AA07A609D4279
            file add 0x1739E27DA61804160075C9617E8D5C585A8F0E0400DD2650FC372EE8F7B22F3D
            file end 0xEEA5B7701DD27E44870E49F1486930B28E6D45874AA62D3F4F1BEFA4EAEF84
            file new BINTECCA.CER
            file add 0x3082049B30820383A003020102020900 88C278A06A7220C5300D06092A864886
            file add 0xF70D010104050030818F310B3009060355040613025350310F300D0603550408
            file add 0x13064D616472696431143012060355040713 0B547265732043616E746F733114
            file add 0x3012060355040A130B54656C64617420532E412E311B3019060355040B131249
            file add 0x502054656C6570686F6E792047726F7570312630240603550403131D54656461
            file add 0x742043657274696669636174696F6E20417574686F72697479301E170D303730
            file add 0x363131313531343136 5A170D3137303630383135313431365A30818F310B3009
            file add 0x060355040613025350310F300D060355040813064D6164726964311430120603
            file add 0x550407130B547265732043616E746F7331143012060355040A130B54656C6461
            file add 0x7420532E412E311B3019060355040B131249502054656C6570686F6E79204772
            file add 0x6F7570312630240603550403131D546564617420436572746966696361746F
            file add 0x6E20417574686F72697479308201223000D06092A864886F70D01010105000382
            file add 0x010F003082010A028201010099246D67DC070A4B0F03391D32FA98C9402739E7
            file add 0x095B0FC06A2EBB4CE6D7862FDAE30E533A76DFF0B2A3B0659A9A4B0C53AA7B8A
            file add 0xFEE7E2B77A2AACABD6AFD2D723FD5DF10D8957EC9AEDDDABD7A1922A60E62C70
            file add 0x135EB101DA465C9A5DA1BF333A453088A68989BF0BB13137F2EA5A036895FBE6
            file add 0xD048830A7266B8AE794F5B5C27F6300998CC1330A093EEF4FB39F704215CC1EC
            file add 0xFD515BDA614CC990B124D383B48C951C53374736596DB5E9A2CE3B9A91C2FB78
            file add 0xC737285D15586E6D9A063ECBB9C8EC3AB82811DB1C9A06BC90AECF61D7ADC6AB
            file add 0x62D67E917E33F605798052B2A1F75737EB57DACEFFA9371A4A48BD2143C19BFA
            file add 0x15313F28D92534D6F998A1FB0203010001A381F73081F4301D0603551D0E0416
            file add 0x04142C24D56B5333A7AFB4718F815F76204B28E4AFF43081C40603551D230481
            file add 0xBC3081B980142C24D56B5333A7AFB4718F815F76204B28E4AFF4A18195A48192
            file add 0x30818F310B3009060355040613025350310F300D060355040813064D61647269
            file add 0x643114301206035504071 30B5472657320436 16E746F7331143012060355040A
            file add 0x130B54656C64617420532E412E311B3019060355040B131249502054656C6570
            file add 0x686F6E792047726F7570312630240603550403131D546564617420436572 7469
            file add 0x6669636174696F6E20417574686F7269747982090088C278A06A7220C5300C06
            file add 0x03551D13040530030101FF300D06092A864886F70D0101040500038201010047
            file add 0x5E868ECD7A4AACF4E2FFD125E45BF9F71B2AF020CCEDEC5A9532630CD4FE3E8B
            file add 0xED1E5755B822997E05437AB19EB35E617093B5900CF323162A99D7D5DF590E01
            file add 0xAA66C395D0DAE3952180624DF2BDD33970D1174292C390A86337C6A0783E3CC8
            file add 0xD5A9053CDB4C393F8D05C27E4B45DCBF77AA907F5BBEAD682E786DBD4BA231EC
            file add 0x91C3D7078A7380DED019CB7FC15CA0D8A35C31A4084E639EB18F90AADF13D10C
            file add 0x59A26B27F25773E9CEC702846EAE69C1C7F482769A9FC0C3274BB6A9FF9CBEB8
            file add 0xF36E44EC3789BF06596AAAE88DFF6D4AB1824804C160FC39D9181CA34957987C
            file end 0xD332635C7E7F8EA7F5866C0A574ED043E5B110DC5DB5CE8507721D9A08F7DF
;
            certificate SAMPLE.CER load
            certificate BINTECCA.CER load
         exit
;
      exit
;
exit
;
protocol sip
;
   application address 172.24.100.133
   application server default
   proxy id.bintec.es default
   proxy id.bintec.es track nsla-advisor 2
;
   crypto signaling default ca bintecca.cer
   crypto signaling default user sample.cer
   transport tls
   realm id.bintec.es
exit
;
feature dns
; -- DNS resolver user configuration --
   server 172.24.51.36
   no cache enable
exit
;
feature nsm
; -- Network Service Monitor configuration --
   operation 1
; -- NSM Operation configuration --
      type echo ipicmp 172.24.75.23
      frequency 5
      timeout 1000
   exit
;
   schedule 1 start-time now
exit
```

```
;
feature nsla
; -- Feature Network Service Level Advisor --
   enable
;
   filter 1 nsm-op 1 rtt
   filter 1 significant-samples 1
   filter 1 activation threshold 300
   filter 1 activation sensibility 80
   filter 1 activation stabilization-time 25
   filter 1 deactivation threshold 200
   filter 1 deactivation sensibility 80
   filter 1 deactivation stabilization-time 25
;
   alarm 1 filter-id 1
;
   advisor 1 alarm-id 1
;
   advisor 2 not alarm-id 1
;
exit
;
dump-command-errors
end
; --- end ---
```

4.3 Gateway SIP

A company wishes to provide voice and data services between the Madrid headquarters and one of its new branches in Barcelona using a PPP link over a serial line at 128 Kbps.

The main office has SIP telephones and a router acting as the SIP server where said telephones are registered. In order to carry out external calls, the router has two VoIP cards, one of which has its four lines in FXO while the other has two lines in FXO and two in FXS. The six FXO lines are connected to the PSTN lines while one of the FXS lines is connected to the FAX and the other to a telephone.

Therefore, up to six calls to external telephones can be simultaneously managed from the headquarters.

The branch has another router with a VoIP card which has one line configured in FXO and connected to a PSTN line. The other three lines are configured in FXS, two connected to telephones and the third to a FAX.

If the FXO line is busy and you wish to route a second call, the ISDN base interface connected to an external ISDN line is used.

The proposed numeration plan for the company is as follows:

- The company internal numbers follow pattern 8.. with the branch numbers taking pattern 89.
- Non-corporate telephones can be fixed, with pattern 9........ or mobiles with pattern 6........
- External calls directed to the Madrid or Barcelona extensions take pattern 913458.. where the last three digits correspond to the user's internal extension number. Consequently the 91345 prefix must be eliminated; this can be executed by configuring a translation.

From the central SIP telephones, calls to the outside are directly routed through one of the FXO lines, except those calls corresponding to Barcelona numbers, which take pattern 93........ and are preferably diverted to the branch router so they can be billed as local calls.

If the PPP line is down, all the calls are routed over the FXO line.

For the branch, calls to external numbers are routed to the central over VoIP, as the Central's FXO lines have lower rates, with the exception of those calls to Barcelona numbers which are preferably routed over the FXO line. If this line is busy, the calls are routed over ISDN.

To check if the PPP line is active, we use a poll to measure the quality of service. If the delay is more than 300 milliseconds the line is considered unsuitable for VoIP. Consequently, the calls are not routed through the dial-peers using said PPP line.

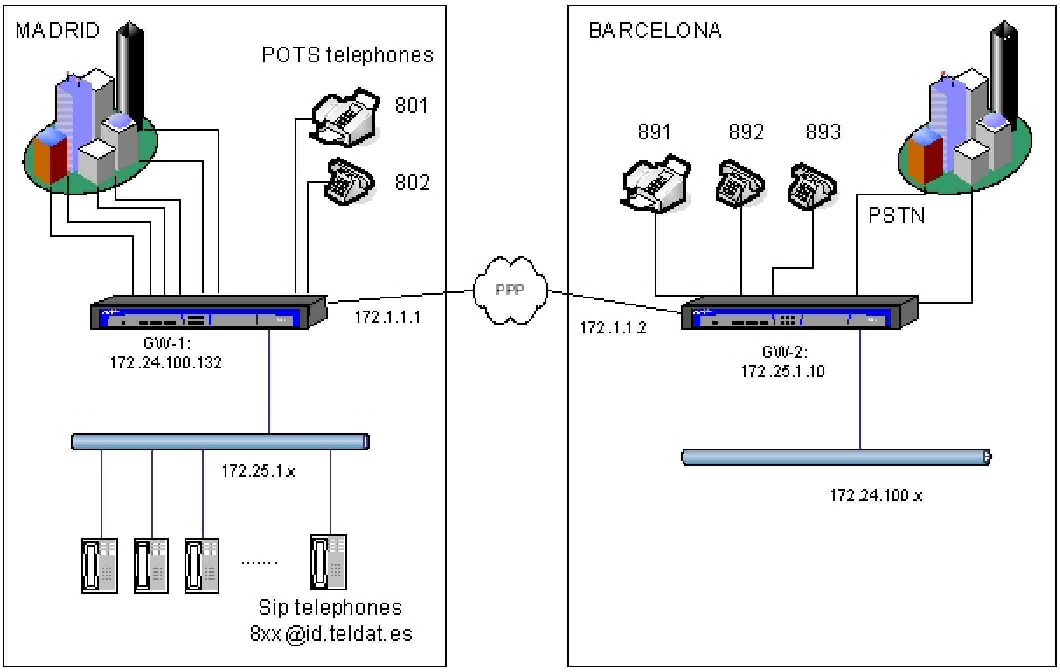

Fig. 2: Gateway SIP

Proposed Configuration:

PPP serial line:

Data is transmitted using the PPP protocol over a serial line at 128 Kbps. RTP (CRTP) header compression is enabled without including the UDP checksum in order to save bandwidth in voice transmission. To reduce delay in the voice packets, fragmentation is also enabled in packets of 512 bytes.

Bandwidth Reservation:

With the aim of preventing voice cutouts when there is a lot of traffic, you need to prioritize voice traffic over data. To do this, use BRS and assign a higher priority to voice than to the rest of the traffic. As multilink is configured, the class for voice over IP must be real-time to avoid it being encapsulated in multilink instead of PPP. This voice traffic is characterized by having the VoIP routers' IP address (172.25.1.10 or 172.24.100.132) as its source or destination IP. A 32-element queue is used when there is little traffic and reduced to 5 when traffic is heavy.

Dial-Peers:

At the main office, a dial-peer is configured for each FXO line, so all nine digit numbers are sent there. A dial-peer is also configured with pattern 93....... so all Barcelona numbers are sent over SIP. Additionally a dial-peer is configured for the fax and another one for the analog telephone. As we want to use the T38 protocol for fax calls, we need to configure the t38-detect mode in the SIP dial-peer of both routers.

At the branch, a dial-peer is configured for each FXS extension, another dial-peer which sends 93....... numbers over the FXO line and a third which sends pattern 9......... numbers over FXO if the proxy is down. An identical configuration is executed with another two dial-peers to use the ISDN line should the FXO be busy.

In addition a dial-peer is configured so all calls to pattern 8........ telephone numbers (Madrid extensions) are sent to the proxy.

The codec used in all calls is G723 at 53 Kbps, one frame per RTP and VAD packet. If codecs are not configured in the dial-peers, all supported codecs are negotiated with G723 having the highest priority.

Configuring the lines:

Six of the GW-1 lines are configured as FXO and the rest (lines 3 and 4 from the second VoIP card) as FXS. In the GW-2 router, line one is in FXO mode and the other three in FXS.

Configurations:

Device-Gw1 (central):

CONFIGURATION:

```
; Showing System Configuration ...
```

```
log-command-errors
no configuration
set hostname gw-1
add device ppp 1
add device voip-isdn 100
set data-link sync serial0/0
set data-link x25 serial0/1
set data-link x25 serial0/2
global-profiles dial
; -- Dial Profiles Configuration --
   profile audio default
   profile audio dialout
   profile audio isdn-type audio
;
exit
;
telephony
; -- Telephony configuration --
   translation 1
      rule 1 913458 any 8 unknown
   exit
;
   dial-peer 1 voice-port
      description "fxs fax extension"
      destination-pattern 801
      target voice-port voip2/0 3
   exit
;
   dial-peer 2 voice-port
      description "fxs telephone extension"
      destination-pattern 802
      target voice-port voip2/0 4
   exit
;
   dial-peer 3 voice-port
      description "external calls through fxo extension"
      destination-pattern .........
      incoming called translation 1
      target voice-port voip1/0 1
   exit
;
   dial-peer 4 voice-port
      description "external calls through fxo extension"
      destination-pattern .........
      incoming called translation 1
      target voice-port voip1/0 2
   exit
;
   dial-peer 5 voice-port
      description "external calls through fxo extension"
      destination-pattern .........
      incoming called translation 1
      target voice-port voip1/0 3
   exit
;
   dial-peer 6 voice-port
      description "external calls through fxo extension"
      destination-pattern .........
      incoming called translation 1
      target voice-port voip1/0 4
   exit
;
   dial-peer 7 voice-port
      description "external calls through fxo extension"
      destination-pattern .........
      incoming called translation 1
      target voice-port voip2/0 1
   exit
;
   dial-peer 8 voice-port
      description "external calls through fxo extension"
      destination-pattern .........
      incoming called translation 1
      target voice-port voip2/0 2
   exit
;
   dial-peer 9 sip
      description "calls to barcelona through gw-2"
      destination-pattern 93.......
      incoming called number 8..
      fax mode t38-detect
      target ipv4 172.25.1.10
      track nsla 2
   exit
;
exit
;
network serial0/0
; -- Interface Synchronous Serial Line. Configuration --
   speed 128000
exit
;
network voip1/0
; -- VoIP interface Configuration --
   line 1 interface-type fxo
;
   line 2 interface-type fxo
;
   line 3 interface-type fxo
;
   line 4 interface-type fxo
;
exit
;
network voip2/0
; -- VoIP interface Configuration --
   line 1 interface-type fxo
;
   line 2 interface-type fxo
;
exit
;
network ppp1
; -- Generic PPP User Configuration --
   ppp
; -- PPP Configuration --
      ccp lzs-dcp seq-lcb process-uncompressed history-count 1
      multilink enable
      multilink endpoint ip 172.1.1.1
      multilink fragmentation 512
   exit
;
   base-interface
; -- Base Interface Configuration --
      base-interface serial0/0 link
;
   exit
;
exit
;
network voip100
; -- VoIP interface Configuration --
   isdn bearer-cap speech
   base-interface
```

```
; -- Base Interface Configuration --
      base-interface bri0/0 255 link
      base-interface bri0/0 255 profile audio
      base-interface bri0/0 255 number-of-circuits 1
;
   exit
;
exit
;
event
; -- ELS Config --
   enable trace subsystem SIP ALL
   enable trace subsystem TLPHY ALL
   enable trace subsystem VOIP ALL
exit
;
protocol ip
; -- Internet protocol user configuration --
   internal-ip-address 172.24.100.132
;
   address ethernet0/0 172.24.100.132 255.255.255.248
   address ppp1 172.1.1.2 255.255.255.0
;
;
   route 172.25.1.0 255.255.255.0 172.1.1.1
   route 0.0.0.0 0.0.0.0 172.24.100.129
;
;
;
exit
;
protocol sip
   application gateway
   application server default
   realm bintec.es
exit
;
feature nsm
; -- Network Service Monitor configuration --
   operation 1
; -- NSM Operation configuration --
      type echo ipicmp 172.1.1.1
      frequency 5
      timeout 1000
   exit
;
   schedule 1 start-time now
exit
;
feature nsla
; -- Feature Network Service Level Advisor --
   enable
;
   filter 1 nsm-op 1 rtt
   filter 1 significant-samples 1
   filter 1 activation threshold 300
   filter 1 activation sensibility 80
   filter 1 activation stabilization-time 25
   filter 1 deactivation threshold 200
   filter 1 deactivation sensibility 80
   filter 1 deactivation stabilization-time 25
;
   alarm 1 filter-id 1
;
   advisor 1 alarm-id 1
;
   advisor 2 not alarm-id 1
```

```
;
exit
;
dump-command-errors
end
; --- end ---
```

Device-Gw2 (branch):

CONFIGURATION:

```
; Showing System Configuration ...

log-command-errors
no configuration
set hostname gw-2
add device ppp 1
add device voip-isdn 100
set data-link sync serial0/0
set data-link x25 serial0/1
set data-link x25 serial0/2
global-profiles dial
; -- Dial Profiles Configuration --
   profile audio default
   profile audio dialout
   profile audio isdn-type audio
;
exit
;
telephony
; -- Telephony configuration --
   dial-peer 1 voice-port
      description "local telephones"
      destination-pattern 93.......
      target voice-port voip1/0 1
   exit
;
   dial-peer 2 voice-port
      description "local telephones"
      destination-pattern 93.......
      target voice-port voip100 1
   exit
;
   dial-peer 3 sip
      description "external telephone calls to central"
      destination-pattern .........
      incoming called number 8..
      fax mode t38-detect
      target ipv4 172.24.100.132
      track nsla 2
   exit
;
   dial-peer 4 voice-port
      description "external telephone calls when no central access"
      destination-pattern .........
      target voice-port voip1/0 1
      track nsla 1
   exit
;
   dial-peer 5 voice-port
      description "external telephone calls when no central access"
      destination-pattern .........
      target voice-port voip100 1
      track nsla 1
   exit
;
   dial-peer 6 voice-port
      description "fxs fax port"
      destination-pattern 891
      target voice-port voip1/0 2
   exit
;
   dial-peer 7 voice-port
      description "fxs telephone1 port"
      destination-pattern 892
      target voice-port voip1/0 3
   exit
;
   dial-peer 8 voice-port
      description "fxs telephone2 port"
      destination-pattern 893
      target voice-port voip1/0 4
   exit
;
exit
;
network serial0/0
; -- Interface Synchronous Serial Line. Configuration --
   speed 128000
exit
;
network voip1/0
; -- VoIP interface Configuration --
   line 1 interface-type fxo
;
exit
;
network ppp1
; -- Generic PPP User Configuration --
   ppp
; -- PPP Configuration --
      ccp lzs-dcp seq-lcb process-uncompressed history-count 1
      multilink enable
      multilink endpoint ip 172.1.1.1
      multilink fragmentation 512
   exit
;
   base-interface
; -- Base Interface Configuration --
      base-interface serial0/0 link
;
   exit
;
exit
;
network voip100
; -- VoIP interface Configuration --
   isdn bearer-cap speech
   base-interface
; -- Base Interface Configuration --
      base-interface bri0/0 255 link
      base-interface bri0/0 255 profile audio
      base-interface bri0/0 255 number-of-circuits 1
;
   exit
;
exit
;
event
; -- ELS Config --
   enable trace subsystem SIP ALL
   enable trace subsystem VOIP ALL
   enable trace subsystem TLPHY ALL
exit
;
```

```
protocol ip
; -- Internet protocol user configuration --
   internal-ip-address 172.25.1.10
;
   address ethernet0/0 172.25.1.10 255.255.255.0
   address ppp1 172.1.1.1 255.255.255.0
;
;
   route 0.0.0.0 0.0.0.0 172.1.1.2
;
;
;
exit
;
protocol sip
   application gateway
   proxy 172.24.100.132 default
   proxy 172.24.100.132 track nsla-advisor 2
;
   realm bintec.es
exit
;
feature nsm
; -- Network Service Monitor configuration --
   operation 1
; -- NSM Operation configuration --
      type echo ipicmp 172.1.1.2
      frequency 5
      timeout 1000
   exit
;
   schedule 1 start-time now
exit
;
feature nsla
; -- Feature Network Service Level Advisor --
   enable
;
   filter 1 nsm-op 1 rtt
   filter 1 significant-samples 1
   filter 1 activation threshold 300
   filter 1 activation sensibility 80
   filter 1 activation stabilization-time 25
   filter 1 deactivation threshold 200
   filter 1 deactivation sensibility 80
   filter 1 deactivation stabilization-time 25
;
   alarm 1 filter-id 1
;
   advisor 1 alarm-id 1
;
   advisor 2 not alarm-id 1
;
exit
;
dump-command-errors
end
; --- end ---
```

Testing when there is IP connectivity:

External calls whose source is an FXO line in the main office with destination to a SIP telephone at the headquarters:

In this case, the number called has pattern 913458.. The input dial-peer eliminates the first six digits so the called number is 8.., which corresponds to a SIP telephone in the LAN where this is registered. If this is so, the call is routed to said telephone. However, if the call does not match any registered SIP user, it is rejected.

External calls whose source is an FXO line in the main office with destination to a gw extension at the branch

In this case, the number called has pattern 913458.. The input dial-peer eliminates the first six digits so the called number is 8.., which corresponds to one of the gw extensions in the branch (891 to 894). If this is so, the call is routed to said telephone. However, if the call does not match any registered SIP user, the call is rejected.

Calls from a SIP telephone in the main office with destination to an external Barcelona number:

In this case, the called number has pattern 93....... and is diverted by the SIP dial-peer to the Barcelona router, which will route the call over the FXO or the ISDN line if the FXO is busy.

Calls between a LAN SIP telephone at the main office to an internal extension:

In this case, the called number has pattern 8.. If there is a registered SIP user with this number, either in Madrid or in Barcelona, the call will be routed to said user.

Calls between an FXS extension from the Barcelona router to an internal extension in Madrid:

In this case, the called number has pattern 8.. and as it does not match any dial-peer, it is routed to the proxy.

Testing when there is no IP connectivity:

External calls whose source is an FXO line in the main office with destination to a SIP telephone at the headquarters:

In this case, the number called has pattern 913458.. and the input dial-peer eliminates the first six digits so the called number is 8.., which corresponds to a SIP telephone in the LAN where this is registered. If this is so, the call is routed to said telephone. However, if the call does not match any registered SIP user, the call is rejected.

External calls whose source is an FXO line in the main office with destination to a gw extension at the branch

In this case, the number called has pattern 913458.. and since there is no IP connectivity with the branch, the call is rejected.

Calls from a SIP telephone in the main office with destination to an external Barcelona number:

In this case, the called number has pattern 93....... As the SIP dial-peer is disabled due to the nsla poll, the call is routed over one of the FXO lines in the same way as any other external call.

Calls between a LAN SIP telephone at the main office to an internal extension:

In this case, the called number has pattern 8.. If there is a registered SIP user with this number in Madrid, the call will be routed in the normal way. If the registered user is in Barcelona, the call will not be routed as there is no IP connectivity with Barcelona.

Calls between an FXS extension from the Barcelona router to an internal extension in Madrid:

In this case, the called number has pattern 8.. and, as it does not match any dial-peer and there is no proxy active, the call is released.

Chapter 5 Annex A

5.1 Third Party Software

When it comes to TLS negotiation, CIT uses the OpenSSL library code.

Please see a copy of the OpenSSL license below: