

VegaStream

Information Note

Vega SIP messaging



This document provides an introduction to SIP messaging, highlighting how Vega configuration affects the contents of the messages sent.

- [SIP Registration](#)
- [SIP General call flow](#)
- [Modifying SIP message formats](#)
- [Matching call legs](#)

It also provides details on what features are supported and which features are not supported in Vega code.

- [RFC2833](#)
- [Supported features](#)
- [Unsupported features](#)

Also other useful information

- [Audio connection times](#)
- [RTCP](#)

Reference information:

The main RFCs that the Vega complies with are:

- RFC 2543 – SIP
- RFC 3261 – SIP
- RFC 3262 – PRACK
- RFC 3264 – Offer Answer Model
- RFC 3323 – Privacy header
- RFC 3325 – P-Asserted-Identity header
- RFC 3326 – Reason Header field
- RFC 3362 – T.38
- RFC 3515 (draft 4) – Refer method

- RFC 3891 – Replaces header
- RFC 3892 – Referred-by header
- RFC 4040 – Clear channel codec
- RFC 1889 – RTP / RTCP
- RFC 2327 – SDP
- RFC 2833 – out-of-band DTMF and hookflash events
- RFC 2976 – SIP INFO method

There are however lots of other RFCs and Vega gateways may either fully or partially adhere to those as well.

SIP Introduction

The VegaStream implementation of SIP was originally based on RFC 2543. As the code is extended, the SIP implementation is being migrated to be conformant with RFC 3261.

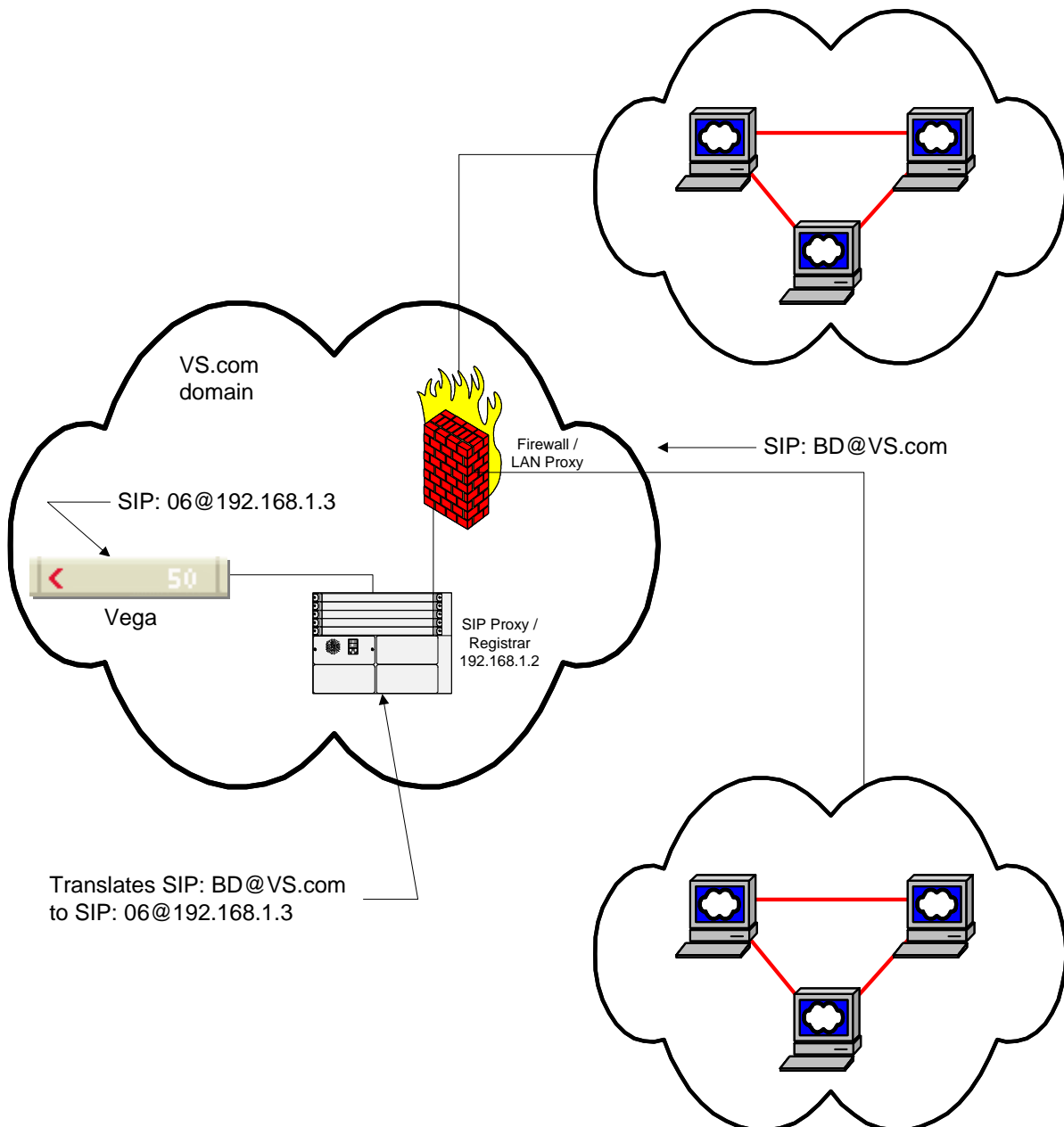
SIP signaling messages may be sent using UDP packets (default) or in a TCP data stream. The RTP (media) packets are sent using UDP.

SIP Registration

- Vega 50 FXS
- Vega 50 FXO from Release 4.1
- Vega 100 and Vega 50 BRI from Release 6

In SIP a user registers with the SIP registrar (often part of the proxy) so that the proxy knows to accept calls from that user and knows that it can send calls to that user.

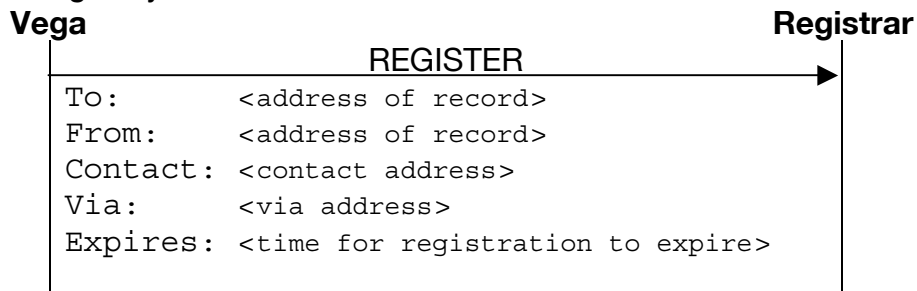
In the diagram below the domain VS.com is protected from the outside world by a firewall.



The public address of the registering user is BD@VS.com. When a SIP call is made to this user the call is routed to the VS.com domain and the Firewall / LAN proxy then forwards the call to the SIP Proxy.

In order to register as a user the Vega sends a REGISTER message to the Registrar as follows ...

Registration message key fields:



- <address of record> = public name of user
 - = sip: UserID@Registration_Domain
 - = UsernamePrefix Username UsernameSuffix@Registration_Domain
 - = or: UsernumberPrefix Usernumber UsernumberSuffix@Registration_Domain
 - e.g. sip: BD@VS.com
 - See also Note 7 and 8
- <contact address> = local name of user
 - = sip: DN@Host_Name_or_IP_address_of_Vega
 - e.g. sip: 05@192.168.1.3
- <via address> = Version of SIP Host_Name_or_IP_address_of_Vega:IP_port
 - e.g. sip: SIP/2.0/UDP 192.168.1.3:5060

Notes:

1. Vega 50 FXS and Vega 50 FXO gateways use per-port registration parameters. Vega 100 and Vega 50 BRI use per-gateway parameters.
2. The REGISTER message is sent to sip.reg_proxy; the Request line is:
REGISTER sip:Registration_Domain SIP/2.0
3. The IP port number that the registration request is sent to is sip.reg_remote_rx_port
4. When using Host_Name_or_IP_address_of_Vega, the Host Name is used if a DNS lookup of this Host Name resolves to the IP address of the Vega.
5. Passwords for authentication must be configured manually – when used for Authentication the password is not passed between the Vega and the Proxy, but is used as part of the hashing process to produce the “response” value to the challenge.

Challenges are supported for REGISTER and for INVITE, not currently supported for BYE.

6. After registration invites received by the Vega will be addressed to the <contact address>

INVITE sip: 05@192.168.1.3 - whole line is known as the Request line

“sip: 05@192.168.1.3” - “sip...” is known as the Request URI

7. For Vega 50 FXS and Vega 50 FXO, authentication userID may either be the same as the `usernumber` configured for registration, alternatively a separate set of configuration parameters may be used to provide the authentication UserID. By setting `_advanced.use_auth_user=0`, the authentication details are taken from the `usernumber` registration parameters. By setting `_advanced.use_auth_user=1`, the authentication details are taken from `auth_` versions of the `usernumber` parameters.
The password is defined on a per port basis.

Vega 100 and Vega 50 BRI always use separate authentication parameters to define the userID and these are configured in `sip.auth.user.n`.

8. If `reg_domain_from_header=1` then instead of using the Vega's IP address / DNS name as the `registration_domain` in the From: header, `sip.reg_domain` will be used.

Example: REGISTRATION – including challenge

```
SIP m:2631883 -1246742 00064--- TX --> To 192.168.1.2:5060
REGISTER sip:VS.com SIP/2.0
Via: SIP/2.0/UDP 192.168.1.3:5060
From: <sip:BD@VS.com>
To: <sip:BD@VS.com>
Call-ID: 0000-0008-69958FEF-99@192.168.1.3
CSeq: 1 REGISTER
Contact: <sip:05@192.168.1.3>
Expires: 600
Content-Length: 0
```

Registration Domain

```
SIP m:2632190 0307 00065<-- RX --- From 192.168.1.2:33359
SIP/2.0 100 Trying
To: <sip:BD@VS.com>
From: <sip:BD@VS.com>
CSeq: 1 REGISTER
Call-ID: 0000-0008-69958FEF-99@192.168.1.3
Via: SIP/2.0/UDP 192.168.1.3:5060;received=67.104.98.89
Content-Length: 0
```

```
SIP m:2632228 0038 00066<-- RX --- From 192.168.1.2:33359
```

```
SIP/2.0 401 Unauthorized
To: <sip:BD@VS.com>;tag=1.A2D9C8E11C5FCFC3F9D2952002D17BDD
From: <sip:BD@VS.com>
CSeq: 1 REGISTER
Call-ID: 0000-0008-69958FEF-99@192.168.1.3
Via: SIP/2.0/UDP 192.168.1.3;received=67.104.98.89
WWW-Authenticate: Digest realm="Ubiquity Software Corporation",
nonce="eaebff6197494216b118facaee8ef75e5684229a4", opaque="1234567890abcdef",
stale=false,
lgorithm=MD5
Content-Length: 0
```

Challenge information,
Nonce = random part
from registrar

```
SIP m:2632248 0020 00067--- TX --> To 192.168.1.2:5060
```

```
REGISTER sip:sipcenter.com SIP/2.0
Via: SIP/2.0/UDP 192.168.1.3
From: <sip:BD@VS.com>
To: <sip:BD@VS.com>;tag=1.A2D9C8E11C5FCFC3F9D295202D17BDD
Call-ID: 0000-0008-69958FEF-99@192.168.1.3
CSeq: 2 REGISTER
Contact: <sip:05@192.168.1.3>
Expires: 600
Authorization:Digest username="BD", realm="Ubiquity Software
Corporation", nonce="eaebff16197494216b118facaee8ef75e5684229a4",
uri="sip:VS.com", response="16b60a1616e40b531a28b5920a742829", opaque="1234567890abcdef"
Content-Length: 0
```

New request; same
call ID

Challenge response

```
SIP m:2632490 0242 00068<-- RX --- From 192.168.1.2:33445
```

```
SIP/2.0 100 Trying
To: <sip:BD@VS.com>;tag=1.A2D9C8E11C5FCFC3F9D295202D17BDD
From: <sip:BD@VS.com>
CSeq: 2 REGISTER
Call-ID: 0000-0008-69958FEF-99@192.168.1.3
Via: SIP/2.0/UDP 192.168.1.3;received=67.104.98.89
Content-Length: 0
```

System elapsed time
in milli seconds

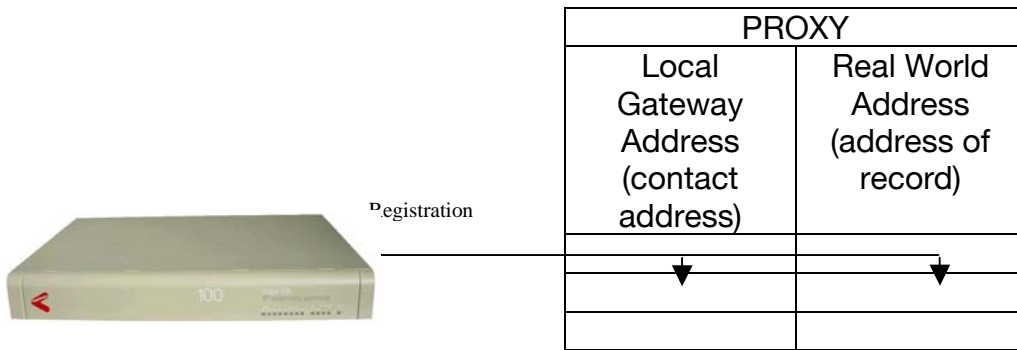
Milli second count
since last diagnostic
or debug message

```
SIP m:2632833 0343 00069<-- RX --- From 192.168.1.2:33445
```

```
SIP/2.0 200 OK
To: <sip:BD@VS.com>;tag=1.A2D9C8E11C5FCFC3F9D295202D17BDD
From: <sip:BD@VS.com>
CSeq: 2 REGISTER
Call-ID: 0000-0008-69958FEF-99@192.168.1.3
Via: SIP/2.0/UDP 192.168.1.3;received=67.104.98.89
Contact: <sip:05@192.168.1.3>;action=proxy;expires=599;q=0.5
Content-Length: 0
```

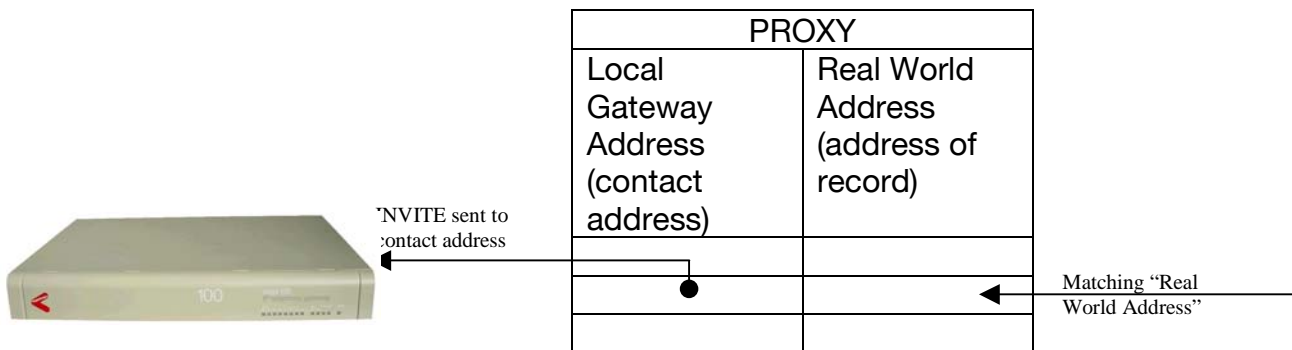
Incrementing count
of SIP messages
monitored

Proxy registration ... what it does:



At registration the Vega provides details to the proxy on the “Real World Address” – the phone number / URL that people in the wide world use to reach this end-point and also the “Local Gateway Address” that the Vega wants the proxy to use to make calls to it.

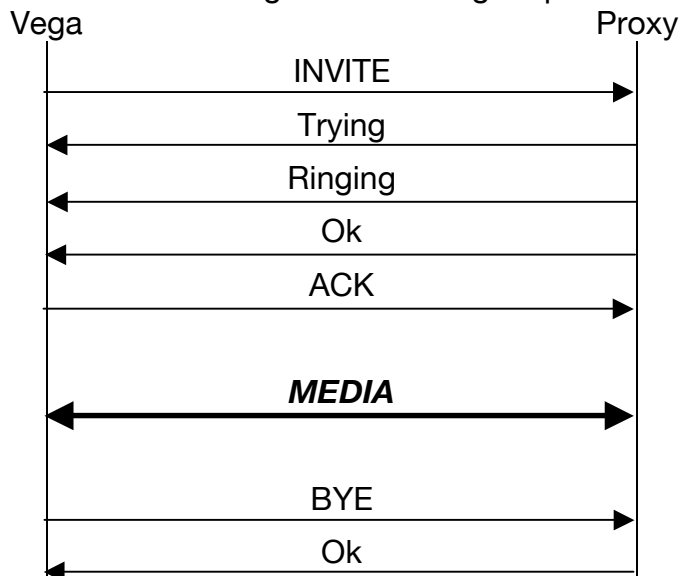
Vega 50 FXSs have the ability to register twice per port – with the idea that one registration will be provide a numeric (telephone number) “Real World Address” registration and the second a URL (text based) “Real World Address” registration.



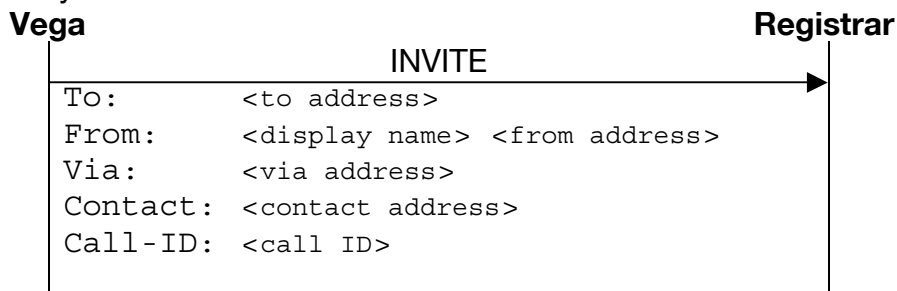
When a call is made via the SIP proxy, the “Real World Address” is matched against the table entries. When a match is found then the INVITE is sent to the Vega using the “Local Gateway Address” (contact address)

SIP General call flow

For a typical outbound SIP call from a Vega the following sequence of events are followed:



INVITE message key fields:



<to address>

= telephone number of destination user as known by local proxy

= sip: dest_TEL@Default Proxy Host Name/IP

e.g. sip: [784900](tel:784900)@192.168.1.2:5060

Note 1: If TA: is configured in the dial plan entry, the TA: value will be used instead of Default Proxy Host Name/IP.

Note 2: If _advanced.sip.reg_domain_to_header=1, the reg_domain will be used instead of Default Proxy Host Name/IP.

<display name>

= UserID of self

= UsernamePrefix Username UsernameSuffix

e.g. BD

<from address>

= public name of self

= <sip:UsernumberPrefix Usernumber UsernumberSuffix@Registration Domain>

e.g. BD <sip:0101@VS.com:5060>

OR

= Alphanumeric caller ID <sip:received caller ID@Registration Domain>
 e.g. CONTRA COSTA CT <sip:9256465540@vegastream.com:5060>
 <via address> = Version of SIP Host_Name_or_IP_address_of_Vega:IP_port
 e.g. sip: SIP/2.0/UDP 192.168.1.3:5060
 <contact address> = local name of self
 = sip: DN@Host_Name_or_IP_address_of_Vega;maddr=
 Host_Name_or_IP_address_of_Vega
 e.g. sip: 06@192.168.1.3:5060;maddr=192.168.1.3
 <call ID> = Unique ID @IP_address_of_Vega
 e.g. 0010-0003-AF40A7C9-0@192.168.1.3

Notes:

1. The INVITE message is sent to “Default Proxy Host Name/IP”; the Request Line is:
 INVITE sip:dest_TEL@Default_Proxy_Host_Name/IP:IP_port SIP/2.0
 Note 1: If TA: is configured in the dial plan entry, the TA: value will be used instead of Default Proxy Host Name/IP.
 Note 2: If _advanced.sip.reg_domain_to_header=1, the reg_domain will be used instead of Default Proxy Host Name/IP.
2. The IP port number that the INVITE is sent to is sip.remote_rx_port
3. When using Host_Name_or_IP_address_of_Vega, the Host Name is used if a DNS lookup of this Host Name resolves to the IP address of the Vega.
4. If the proxy that the INVITE is sent to fails to respond fully it is possible to configure the Vega to try other “backup” proxies – see primer for details.
5. Invites received by the Vega will be addressed to the <contact address>
 INVITE sip: 05@192.168.1.3 - whole line is known as the Request line
 “sip: 05@192.168.1.3” - “sip...” is known as the Request URI
6. When receiving an incoming INVITE the information before the @ in the Request URI (see 3. above) is passed to the router (Vega Dial Planner) as the source TEL:
7. The To: From: and Call-ID: fields define a call “dialogue”. The To: field and the From: field define the end points and represent the direction of the SIP Request. In a single dialogue (call) the Call-ID: must always be the same, and the endpoints must also remain constant – though in response to an invite the recipient may add a tag value to the To: field, which from that time onwards must be used as part of that field.

Example: CALL FLOW – including hookflash event

SIP m:0367091 14525 00038--- TX --> To 192.168.1.2:5060

```
INVITE sip:784900@192.168.1.2:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.3:5060
From: BD <sip:0101@VS.com:5060>
To: <sip:784900@192.168.1.2:5060>
Call-ID: 0010-0003-AF40A7C9-0@192.168.1.3
CSeq: 1 INVITE
Contact: <sip:06@192.168.1.3:5060;maddr=192.168.1.3>
Allow: INVITE,ACK,BYE,CANCEL,INFO,NOTIFY,OPTIONS
Accept-Language:en
Content-Type: application/sdp
Content-Length: 134
```

SDP
(Session
Description
Protocol)
payload

```
v=0
o=Vega50 4 1 IN IP4 192.168.1.3
s=Sip Call
c=IN IP4 192.168.1.3
t=0 0
m=audio 10006 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

Port number for
audio

SIP m:0367138 0047 00039<-- RX --- From 192.168.1.2:5060

```
SIP/2.0 100 Trying
Via: SIP/2.0/UDP 192.168.1.3:5060
From: BD <sip:0101@VS.com:5060>
To: <sip:784900@192.168.1.2:5060>;tag=0010-0005-1AB53F6A
Call-ID: 0010-0003-AF40A7C9-0@192.168.1.3
CSeq: 1 INVITE
Contact: <sip:784900@192.170.1.54:5060;maddr=192.170.1.54>
Content-Length: 0
```

SIP m:0367168 0030 00040<-- RX --- From 192.168.1.2:5060

```
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP 192.168.1.3:5060
From: BD <sip:0101@VS.com:5060>
To: <sip:784900@192.168.1.2:5060>;tag=0010-0005-1AB53F6A
Call-ID: 0010-0003-AF40A7C9-0@192.168.1.3
CSeq: 1 INVITE
Contact: <sip:784900@192.170.1.54:5060;maddr=192.170.1.54>
Content-Type: application/sdp
Content-Length: 134
v=0
o=Vega50 6 1 IN IP4 192.170.1.54
s=Sip Call
c=IN IP4 192.170.1.54
t=0 0
m=audio 10008 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

SIP m:0368846 1678 00041<-- RX --- From 192.168.1.2:5060

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.3:5060
From: BD <sip:0101@VS.com:5060>
To: <sip:784900@192.168.1.2:5060>;tag=0010-0005-1AB53F6A
Call-ID: 0010-0003-AF40A7C9-0@192.168.1.3
CSeq: 1 INVITE
Contact: <sip:784900@192.170.1.54:5060;maddr=192.170.1.54>
Allow: INVITE,ACK,BYE,CANCEL,INFO,NOTIFY,OPTIONS,PRACK
Accept: application/sdp, application/none
Accept-Language:en
Content-Type: application/sdp
Content-Length: 134
v=0
o=Vega50 7 1 IN IP4 192.170.1.54
s=Sip Call
c=IN IP4 192.170.1.54
t=0 0
m=audio 10008 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

SIP m:0368883 0037 00042--- TX --> To 192.168.1.2:5060
ACK sip:784900@192.168.1.2:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.3:5060
From: BD <sip:0101@VS.com:5060>
To: <sip:784900@192.168.1.2:5060>;tag=0010-0005-1AB53F6A
Call-ID: 0010-0003-AF40A7C9-0@192.168.1.3
CSeq: 1 ACK
Contact: <sip:06@192.168.1.3:5060;maddr=192.168.1.3>
Content-Length: 0

SIP m:0374284 5401 00043<-- RX --- From 192.168.1.2:5060
INFO sip:0101@192.168.1.3:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.2:5060
From: <sip:784900@192.168.1.2:5060>;tag=0010-0005-1AB53F6A
To: BD <sip:0101@VS.com:5060>
Call-ID: 0010-0003-AF40A7C9-0@192.168.1.3
CSeq: 2 INFO
Content-Type: application/none
Content-Length: 17
event flashhook

SIP m:0374308 0024 00044--- TX --> To 192.168.1.2:5060
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.2:5060
From: <sip:784900@192.168.1.2:5060>;tag=0010-0005-1AB53F6A
To: BD <sip:0101@VS.com:5060>
Call-ID: 0010-0003-AF40A7C9-0@192.168.1.3
CSeq: 2 INFO
Contact: <sip:0101@192.168.1.3:5060;maddr=192.168.1.3>
Content-Length: 0

SIP m:0391440 17132 00045<-- RX --- From 192.168.1.2:5060
BYE sip:0101@192.168.1.3:5060 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.2:5060
From: <sip:784900@192.168.1.2:5060>;tag=0010-0005-1AB53F6A
To: BD <sip:0101@VS.com:5060>
Call-ID: 0010-0003-AF40A7C9-0@192.168.1.3
CSeq: 3 BYE
Content-Length: 0

SIP m:0391469 0029 00046--- TX --> To 192.168.1.2:5060
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.168.1.2:5060
From: <sip:784900@192.168.1.2:5060>;tag=0010-0005-1AB53F6A
To: BD <sip:0101@VS.com:5060>
Call-ID: 0010-0003-AF40A7C9-0@192.168.1.3
CSeq: 3 BYE
Contact: <sip:0101@192.168.1.3:5060;maddr=192.168.1.3>
Content-Length: 0

System elapsed time
in milli seconds

Milli second count
since last diagnostic
or debug message

Incrementing count
of SIP messages
monitored

Invite from FXO with caller ID (Agilent trace)

```
----- SIP Header -----
SIP: Message Type = Request
SIP: Method = INVITE
SIP: Request URI = sip:4444@64.85.227.9:5060
SIP: SIP Version = SIP/2.0
SIP: Via = SIP/2.0/UDP/64.85.241.31:5060 (Path Taken By Request Till Now)
SIP: From = CONTRA COSTA CT <sip:9256465540@vegastream.com:5060>;tag=0010-0002-CDBE3BF1
(Request Initiator)
SIP: To = <sip:4444@64.85.227.9:5060> (Recipient Of Request)
SIP: Call-ID = 0010-0002-71A460E5-0@64.85.241.31 (Unique Identifier)
SIP: Cseq = 1 INVITE (Command Sequence Number)
SIP: Contact = <sip:9256465540@64.85.241.31:5060;maddr=64.85.241.31> (Contact Details)
SIP: Allow = INVITE,ACK,BYE,CANCEL,INFO,NOTIFY,OPTIONS (Supported Methods)
SIP: Accept-Language =en (Acceptable Languages)
SIP: Content-Type = application/sdp (Media Type Of Message Body)
SIP: Content-Length = 128 (Message Body Length In Octets)

----- SDP Header -----
SDP: Version = 0
SDP: Origin Field :
SDP:   User Name = Vega50
SDP:   Session Identifier = 3
SDP:   Session Version = 1
SDP:   Network Type = IN
SDP:   Address Type = IP4
SDP:   Address = 64.85.241.31
SDP: Session Name Field :
SDP:   Session Name = Sip Call
SDP: Connection Field :
SDP:   Network Type = IN
SDP:   Address Type = IP4
SDP:   Address = 64.85.241.31
SDP: Time Field :
SDP:   Start Time = 0
SDP:   Stop Time = 0
SDP: Media Field :
SDP:   Media Type = audio
SDP:   Port Number = 10004
SDP:   Transport Protocol = RTP/AVP
SDP:   Media Format = PCMU (0)
SDP: Attributes Field :
SDP:   rtpmap=0 PCMU/8000
```

The diagram shows two lines originating from the SIP header. One line starts at the 'SIP: From' field, which contains the alphanumeric string 'CONTRA COSTA CT', and points to a box labeled 'Alphabetic caller ID'. The other line starts at the 'SIP: To' field, which contains the numeric string '<sip:4444@64.85.227.9:5060>', and points to a box labeled 'Numeric caller ID'.

Modifying SIP message formats

INVITE Message Session Description

Some systems require the “c=” line to be in in the SDP media description, others require it in the SDP session description. Vegas can support either requirement based on the configuration of the parameter:

If:

```
_advanced.sip.sdp.ssess_desc_connection=0 ;(Default=0)
```

the “c=” line appears in the SDP media description. For example:

```
v=0
o=Vega50 7 1 IN IP4 136.170.208.245
s=Sip Call
t=0 0
m=audio 10012 RTP/AVP 0
c=IN IP4 136.170.208.245
a=rtpmap:0 PCMU/8000
```

If:

```
_advanced.sip.sdp.ssess_desc_connection=1
```

the “c=” line appears in the SDP session description. For example:

```
v=0
o=Vega50 8 1 IN IP4 136.170.208.245
s=Sip Call
c=IN IP4 136.170.208.245
t=0 0
m=audio 10014 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

User Agent Header

From Release 4, by default a user agent header was added to Vega messages, e.g.:

```
User-Agent: Vega50-Wisc/04.02.04xT034
```

If this causes problems and needs to be disabled, set

```
_advanced.sip.user_agent_header=0
```

Media headers for T.38

By default, when T.38 is enabled the Vega will provide multiple SIP “m=” headers in the SDP request – providing “m=” lines for audio as well as image codecs.

If this does not work when requesting t.38 the audio “m=” headers can be removed (leaving just a single t.38 “m=” header) by setting:

```
_advanced.sip. t38_single_media=1
```

Matching call legs

When matching messages with calls the Vega uses:

- the From: header – including any tag field,
- the To: header – including any tag field, and
- the Call-ID.

All fields must match exactly.

If

```
_advanced.sip.match_req_uri=1
```

then also include the request URI in the matching process.

RFC2833

RFC2833 is implemented to support the transmission of DTMF tones (only; not modem or MF tones which the specification also has methods for handling).

RFC2833 operation can be seen in the INVITE message sent to or from the Vega. An “a=” with a codec type “telephone-event...” will be seen. In the example below the RTP_AVP type is 96, and this can be seen in the offered codecs “m=” list.

```
INVITE sip:...
...
c=IN IP4 136.170.208.167
m=audio 8766 RTP/AVP 0 96 8
a=rtpmap:0 pcmu/8000/1
a=rtpmap:96 telephone-event/8000/1
a=rtpmap:8 pcma/8000/1
```

If the call is from / via a SIP proxy to the Vega, it may be desired that the RFC2833 DTMF messages be sent to the proxy whilst the audio is sent directly to the endpoint. In this case two sets of connection information may be sent. In the example below pcm u-law (0) or pcm a-law (8) are being offered and the chosen audio format should be sent to 136.170.208.167. RFC2833 DTMF data (telephone-event data) should be sent to 136.170.208.165.

```
INVITE sip:...
...
c=IN IP4 136.170.208.167
m=audio 8766 RTP/AVP 0 8
a=rtpmap:0 pcmu/8000/1
a=rtpmap:8 pcma/8000/1
c=IN IP4 136.170.208.165
m=audio 9766 RTP/AVP 96
a=rtpmap:96 telephone-event/8000/1
```

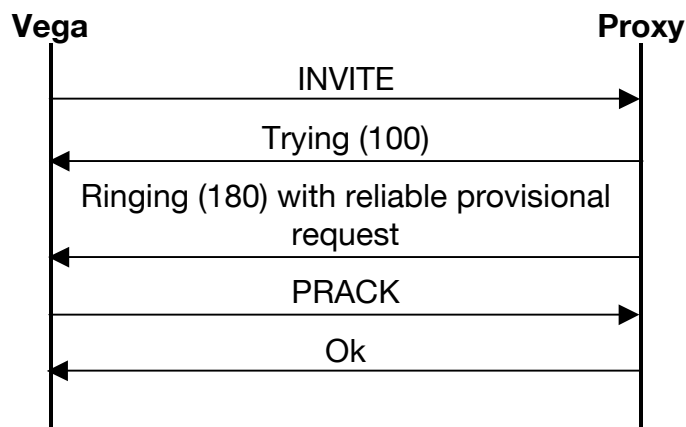
Supported Features

For more details see the SIP user guide

- Prack – Provisional Response ACKnowledge

In general in SIP it is up to the initiator to ensure reliable delivery of messages, Requests are retransmitted, Responses are not (it is up to the originator to run a timer and if no Response is received then re-send the Request). Prack is triggered by a “reliable provisional” request in the Provisional response (180 series of responses (180, 181, 182 ...) e.g. Ringing (180)).

Example message sequence:



- Forking – When a call is made through a proxy, the proxy may duplicate the INVITE to multiple destinations. When a destination receives the INVITE it responds trying / ringing with an additional tag field in the To: address (to uniquely identify the end point). When the first endpoint accepts the call (responds Ok to the IVITE) the Proxy sends CANCEL to all the endpoints that the original INVITE was sent to. All endpoints except the one that OK'd the IVITE then cancel the call request.
- REFER is supported (- a method for call transfer) ... the Vega also supports the Bye with Also (clear call then restart it) and the Re-Invite (for media transfer) methods.
- RFC2833 is supported
- SIP Annex D (UDP T.38) is supported
- Authentication – Vegas support digest authentication (using MD5 hashing) on REGISTER, INVITE, ACK and BYE messages.
- TCP transport for SIP signaling messages is supported
- RPID- Remote party ID, used for carrying called and calling party number details, presentation and screening information and original called number / redirecting number.
- DNS-SRV records are supported to provide alternate IP addresses for Registrars / proxies

Unsupported features

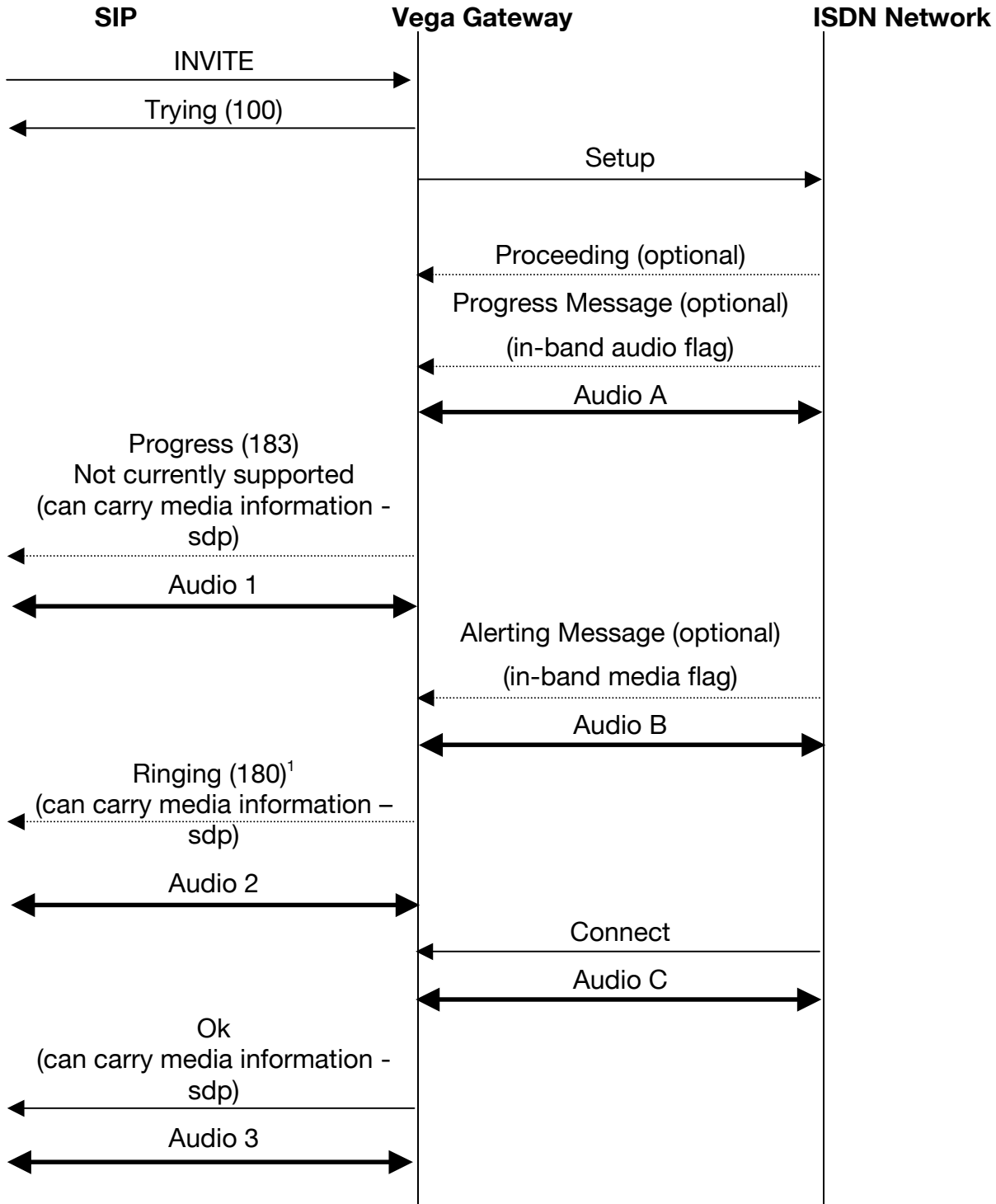
- Multicast invites – used to invite multiple parties to a conference – is not supported as the Vega does not have an audio conferencing capability.
- Multicast search for proxys is not supported
- PROPOSE method is not supported
- COMET – no longer in SIP spec see UPDATE

Other Notes

- If DNS is to be used (to get IP addresses from names) then lan.name must be configured to be the Vega's DNS hostname.

Audio Connection times

Call SIP to ISDN



¹ To return sdp information in the ringing message and generate ringback tone the Vega needs to have `Tones.net.ring=1`

Audio can connect at various stages depending upon settings / Vega implementation. To get a clear audio path both an Audio <number> and an Audio <letter> must be made.

Setup message carries details of the codecs supported.

Audio 1 – audio gets connected at this point if Progress (183) is supported.

Audio 2 – audio gets connected at this point if the ringing message flags media available (In the Vega, the sdp is provided with ringing if tones.net.ring=1)

Audio 3 – audio gets connected at this point if not connected before

Audio A – audio should connect here if the ISDN progress message indicates in-band audio

Audio B – audio should connect here if the ISDN alerting message indicates in-band audio

Audio C – audio connects here on the connect message if it is not connected before.

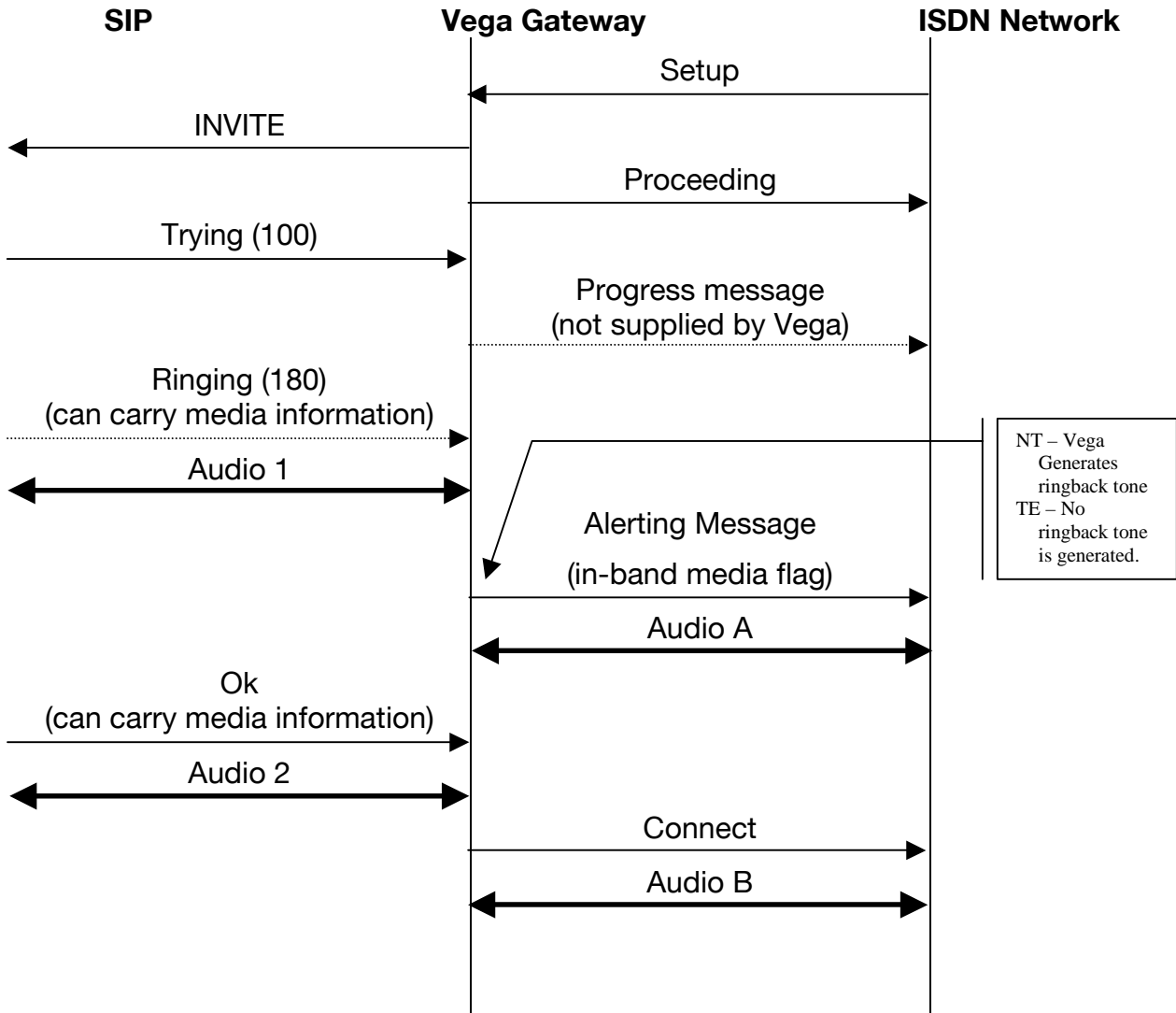
FXS receives “Connect” as the handset is lifted – there is no possibility of earlier in-band media indication.

FXO receives “Connect” on seizing the outgoing line, so the caller will hear the PBX / CO switch dial tone, the Vega outdialing the DTMF tones and any following call progress tones.

Note:

1. Audio connection in SIP side is carried out by the one end saying “send the audio to me at this IP address/port number using this codec(s)”. The other device has to receive this and start sending the audio. Thus the actual time of audio connection depends on the operation of the units at both ends. For bi-directional audio, both ends must have sent their sdp and received and processed the sdp from the other end.
2. When initiating a call, Vegas always send sdp information in the INVITE.

Call ISDN to SIP



Audio can connect at various stages depending upon settings / Vega implementation. To get a clear audio path both an Audio <number> and an Audio <letter> must be made.

Setup message carries details of the codecs supported.

Audio 1 – audio gets connected at this point if the ringing message flags media available
Audio 2 – audio gets connected at this point if not connected before

Audio A – audio connects here if the ISDN alerting message indicates in-band audio
Audio B – audio connects here on the connect message if it is not connected before.

FXS has no connect state, so the audio is connected on the telecoms side as soon as the line is seized.

FXO Vega will trigger an INVITE when it detects inbound ringing. It answers the call (seizes the line) when the SIP Ok is received, so the "Connect" occurs on call answer.

Notes:

1. See notes on page 18.
2. For calls ISDN to SIP, if the Vega trunk is configured as TE, the variable `_advanced.user_progress` controls whether progress tones are generated by the Vega. (The Vega needs to generate the tones if the Network does not.)

RTCP

Real Time Control Protocol is the control protocol that runs alongside RTP that allows monitoring of the data delivery. RTP uses even numbered IP ports and the associated RTCP stream uses the next higher (odd) numbered port. Both streams of data are between the same two IP end points.

In Vegas, an RTCP packet will be sent every 5 seconds. Details of RTCP content defined in RFC 1889 - RTP: A Transport Protocol for Real-Time Applications.

Contact Details

Email: support@vegastream.com

Web: <http://www.vegastream.com>

EMEA Office
VegaStream Limited
Western Centre
Western Road
Bracknell
Berks RG12 1RW
UK

+44 (0) 1344 784900

USA Office
VegaStream Inc.
6200 Stoneridge Mall Road
3rd Floor
Pleasanton
California 94588
USA

+1 925 399 6428