SMS

Short messaging services in the Mizu VoIP Server

Contents

About

Users of the Mizu VoIP server can send and receive SMS messages from softphones, web portal or API. The server is capable to also convert simple SIP text messages to SMS if the target is a mobile phone number.
Incoming SMS requests can be converted to HTTP(S) or SMPP(S) requests and forwarded to an SMS gateway defined by the smsurl global config value.
The server can also use SMS for special purposes such as SMS code based sign-up verification (uauthverify) or administrative alters.

User interface

Once SMS is configured on your server, the endusers has various method to send SMS messages:

- Using the API. This can be easily integrated with any custom user interface or third party tool. See the API section below for the details.
- Customized softphones: softphones provided by Mizutech can have a "Send SMS" feature
- Third-party softphones: the server is cabable to convert standard SIP chat messages to SMS if the target is a mobile phone number, thus SMS can be sent from any SIP endpoint (sendsmsmessages global config option enabled by default)
- Web portal: Users can also send SMS messages from the enduser web control panel (SMS page)
- Server console: you can easily test SMS messages using the asendsms command. Example: asendsms,TONUMBER,Hello word

SMS Integration

To be able to offer SMS capabilities for your users, you will need to interconnect with an SMS service provider or with an SMS gateway.
The Mizu VoIP server has support for SMS over various protocols such as SIP IM, email, 3GPP SMS routing, SMPP or HTTP GET/POST with proprietary API's using clear text, JSON, XML or other text based payload types.
You can configure integration with any SMS service as described below.

Outbound SMS

The server can connect to SMS providers (such as clickatel , routomessaging or sight) and send SMS message via SMPP(S) or HTTP(S) requests.
If you don't have an SMS provider yet then please consider this list:

clickatell.com
www.sinch.com
infobip.com
messagingbay.com
www.routomessaging.com
hqsms.com
tm4b.com
smsbts.com
bulksms.co.uk
smsxchange.com
textmagic.com
truesenses.com
txtlocal.co.uk

We recommend clickatel and sight due to their affordable pricing and system stability.

The SMS API URL and it's parameters must be obtained from the SMS provider.
Set the URL in the global configuration under the smsurl key.
If the SMS is to be sent over SMPP, then the URL must have the following format:
SMPP;key1=value1;key2=value2;…etc. See the parameters below at the "SMPP parameters" section.
If the request have to be a HTTP POST (instead of GET) then you can set the data to be posted with the smssendata parameter or stored in the smssendata.txt file in the server app folder.
If some special HTTP header(s) are required (for example Authentication) then you can specify it with the sms smssendheader parameter or in the smssendheader.txt file.

You can use the following keywords in the smsurl, smssendata and smssendheader:
[fromid], [fromnum],[tonum],[message],[smscounter],[smsid],[gatewayid],[clientip] or any SQL query in brackets such as {SELECT …}.
These will be replaced automatically to their respective values for each session.
The delivery success is guessed by default from the response or you can configure the smssuccessstring and/or smsfailstring the keyword to look for (the response is parsed as simple text).
Unicode messages are also supported.

If you wish to route to SMS messages to multiple providers then you should follow the following steps:

- set the smsrouting global config option to 1
- add SMS GW users
- setup the routing
- setup billing
- HTTP POST data will be loaded from smssendata_ID.txt loaded file (where ID is the gateway id -tb_users.id)
- HTTP headers can be added from smssendheader_ID.txt file (where ID is the gateway id - tb_users.id)

You can find more details about SMS setup here.

Inbound SMS

You can also receive incoming SMS messages as HTTP callback requests.
To enable SMS receiving, set the smsreceive_route parameter to 4 (or any other value as described below).
For incoming SMS you will have to configure your SMS service provider callback URL to point to your server smsreceive API which looks like this:
http://domain.com/mvapireq/?apientry=smsreceive
(Replace the domain.com with your SIP server domain name or IP address and use HTTPS instead if you enabled TLS on your server)
For example a simple SMS send might look like this:
https://sip.mydomain.com/mvapireq/?apientry=smsreceive&from=1234567&to=98765432&message=Works&now=555
(You can also use POST instead of GET)
See the smsreceive_xxx parameters below for other incoming SMS related configuration options.

SMS routing and billing

After all SMS message a CDR record will be created with its type set to "SMS".

SMS messages are routed according to your routing rules in case if you have multiple outbound routes and billed according to your pricing rules. SMS messages billing can be specified by selecting SMS as the service type on the Price setup form. The smstime is set to 60 sec by default.
If no pricing found, then the sms messages will be billed by smsprice value. For unified pricing you can use the smsprice and the smstime. (in this case all sms messages will be billed with the same price, regardless of the destination) Otherwise the pricing can be set on the "Price Setup" form the same way as for voice calls. The only difference is that all SMS messages appear with 60 sec duration.
In case if you wish to rewrite the outbound SMS number, check this FAQ point for the details.

Rewrite SMS number or prefix

Create a stored procedure named v_dialplansms with the following inputs:

- tonum varchar(256)
- fromid int
- fromnum varchar(64)
- clientip varchar(64)

This sp must return one field containing the rewritten target number ("tonum")

Set the "usesmsprefixsp" global config option to "true" and reload.

Note:
For each SMS message a CDR record is generated (you can see them on the „CDR" form).
Please note that the CDR generation might be delayed by about 10 seconds (because the server also sends a second requests to verify wether the SMS was sent successfully or not).

API

The following API entries are defined:
sms: old deprecated
sendsms: asynchronously (fast) send SMS for endusers (check user rights and billing)
sendsmssync: synchronously (wait for delivery result) send SMS for endusers (check user rights and billing)
asendsms: asynchronously (fast) send SMS for admins
asendsmssync: synchronously (wait for delivery result) send SMS for admins

API parameters:
Sender number: anum or from
Target number: bnum or to
Message text body: message or txt

Answer:
On success, the API will answer with message beginning with OK text.
On failure, the API will answer with message beginning with ERROR text.

Note:
By default the "asendsms" and "sendsms" API will send the SMS asynchronously. This means that you will receive a positive answer immediately once the send was initiated (which is then completed in a separate thread).
You can use the "asendsmssync" or "sendsmssync" API to send SMS synchronously. This will wait for the final status.
We recommend the "asendsms" and "sendsms" API to be used for high volume SMS since it is much faster. The "asendsmssync" and "sendsmssync" API should be used only if you need an accurate result (delivery succeed/failed) for the API. Otherwise you can always check the CDR for the delivery results.

Example:
http://yourserveraddress.com/mvapireq/?apientry=asendsms&authkey=xxx&authid=xxx&authmd5=xxx&authsalt=xxx&anum=FROMNUMBER&bnum=TONUMBER&message=MESSAGE&now=555

Configurations

All SMS related settings can be configured from the following places:

- Users and devices form: you might create multiple outbound SMS gateways as "SMS GW" user entries
- Routing form: you can create routing rule patter(s) for SMS messages by setting the "Type" to "SMS" for the directory definition
- Price Setup form: create billing rules for SMS messages by specifying "SMS" for the pricing packet(s) service type
- Global configuration: search for "sms" and/or "smpp" to list all SMS related settings (discussed here below)

The SMS related global config options are listed below:
enablesms: enable the SMS module (default is true. Set to false to disable)
smsurl: outbound SMS URL (it can be also an executable path instead of URL)
smssendata: HTTP data for outbound SMS if POST have to be used
smssendheader: HTTP headers for outbound SMS (for example Authentication header)
smsthread: 0: no separate thread (more reliable answer), 1: separate thread (better performance)
smsverifyurl: if a second request is request to verify the SMS success (currently harcoded to use "messageId" from the answer
smssendataverify: HTTP POST data for verify (otherwise HTTP GET will be used)
smsverifydelay: milliseconds to wait before calling the verify URL
smssignature: to add a "signature" text after all outbound SMS
smsadv: set to true for advertising –to insert and advertising message before each outbound SMS (tb_advertisment)
smsadvprefix: insert a text before each message
smsadvvalue: users can receive some credit if advertisement are sent to them
smssuccessstring: success keyword to look for in the response (depending on your SMS provider and payload type)
smsfailstring: failure keyword to look for in the response (depending on your SMS provider and payload type)
smsrouting: if SMS routing have to be used ("Routing" form): -1: auto guess, 0: global config only, 1=multiple sms gw
sendsmsmessages: specifiy wether to convert SIP IM to SMS. 0=no,1=to mobile only (default), 2=to all phone number, 3=to all destination
smsprice: default SMS price for a 60 second SMS (which is the default "duration" for an SMS)
smstime: SMS messages will be treated with this duration (60 seconds by default)
smsreceive_route: how to forward incoming SMS to users: 0: disable, 1:chat,2:email,3:chat or email,4:chat and email
smsreceive_ip: allow incoming SMS only from this IP address or IP address list separated by comma (if empty, then it will allow from anywhere)
smsreceive_check: any string to check if the message is incoming sms, otherwise will be dropped. usually empty
smsreceive_from: extact sms sender number from the incoming message. it can be a single word parameter name or fromstring1,fromstring2,tostring1,tostring2

smsreceive_to: extact sms target number (our user) from the incoming message. it can be a single word parameter name or fromstring1,fromstring2,tostring1,tostring2
smsreceive_message: extact the SMS message body from the incoming message. it can be a single word parameter name or fromstring1,fromstring2,tostring1,tostring2

SMPP parameters:
SMPP parameters can be set in the global configuration with an "smpp_" prefix (for example "smpp_server") or inside the SMS URL as key=value pairs.
Example URL or smpp_allparameters:

- Basic: SMPP;server=11.22.33.44;username=smppuser;password=xxx;serverport=2775
- Advanced: SMPP;server=55.66.77.88;username=smppuser;password=xxx;serverport=2785;version=50;timeout=30;msgpriority=3;servicetype=CMT;SourceTON=1

Possible parameters:

- smpp_allparameters: You can store all the below parameters in this single field using the above format
- server: SMPP server address to connect to. The instant messaging server must contain a valid SMPP service application address. Usually an IP address.
- username: User ID value to be used for identification with the SMPP service
- password: This is the user's password
- serverport: SMPP server port (default is 2775)
- version: the SMPP version to be used. Possible values: 33: v.3.3, 34: v.3.4, 50: v.5.0. Default is 34 (v.3.4)
- secure: set to 1 if you need SSL/TLS connection
- systemtype: Some SMS servers might require suplying the system type which is usually an abbreviation string of maximum 12 characters. Default is empty.
- servicetype: type of service. Can be set globally or by message. Possible values: 0: default, 1: CMT, 2: CPT, 3: VMN, 4: VMA, 5: WAP, 6: USSD, 7: CBS. Default is 0.
- msgpriority: message priority. Can be set globally or by message. Possible values: -1: default, 0: low, 1: normal, 2: high, 3: urgent. Default is 1 (normal).
- msgexpire: optional MC expiration time in YYMMDDhhmmsstnnp format. p set to "R" means relative. (validity period of the current message). Default is empty. t means tenths of a second (0-9). nn means quarter-hour time difference between local time and UTC time (00-48)
- timeout: timeout in seconds for inter-packet inactivity. Default is 60. You might set the AbsoluteTimeout config to true to interpret this as total timeout.
- recipients: one destination phone number or multiple destinations separated by ; (max 255)
- recipienttype: 0: normal SMS number or ipv4, 1: namelist
- isdata: set to 1 if you wish to send the paypload as data instead of a message. Only one recipient is supported in this case
- extraconfig: additional exta configuration parameters as key=value;key2=value2;
    - SenderAddress: the SMPP protocol allows an External Short Messaging Entity (ESME) to specify its address, whether it is a phone number or an IP address. If SenderAddress is not set, the component will default to the value in LocalHost
    - SourceTON: Type of Number for the ESME. 0: unknown, 1: international, 2: national, 3: network specific, 4: subscriber number without prefixes, 5: alphanumeric, 6: abbreviated
    - SourceNPI: Number Planning Indicator for the ESME used to specify the numbering plan. 0: unknown, 1: ISDN, 3: data, 4: telex, 6: landline, 8: national, 9: private, 10: ERMES, 14: internet, 18: WAP. Default is 1 which since mobiles are usually covered by ISDN
    - DestinationTON: Type of Number for the destination ESME. Possible values: same as for SourceTON.

- DestinationNPI: Number Planning Indicator for the desination ESME. Possible values: same as for SourceNPI.
- DataCoding: data encoding mechanism to be used for the current message. Possible values: 0: MC Specific encoding, 1: IA5 (CCITT T.50)/ASCII (ANSI X3.4), 2: Octet unspecified (8-bit binary), 3: Latin 1 (ISO-8859-1), 4: Octet unspecified (8-bit binary), 5: JIS (X 0208-1990), 6: Cyrillic (ISO-8859-5), 7: Latin/Hebrew (ISO-8859-8), 8: UCS2 (ISO/IEC-10646), 9: Pictogram Encoding, 10: ISO-2022-JP (Music Codes), 11: Reserved, 12: Reserved 2, 13: Extended Kanji JIS (X 0212-1990), 14: KS C 5601,
- HexString: a hex-encoded binary string to be sent to the current recipient
- InBufferSize: size in bytes of the incoming queue of the socket.
- OutBufferSize: size in bytes of the outgoing queue of the socket.
- AbsoluteTimeout: determines whether timeout means inactivity timeout or absolute timeout

- ssl config (to be set only if you need to use secure SMPPS):
  - sslaccept: set to "ANY" or "ALL" to accept any certificate presented by the server. set to - to cler it and force cert validation.
  - sslencodig: the PEM/base64 encoded SSL certificate
  - sslcertstore: name of the certificate store for the client certificate. Possible values: MY: personal cert store, CA: authority certs, ROOT: root certs, SPC: publisher cert, filename: PFX or java cert store or cert+private key file
  - sslpwd: password for the certificate store (if any)
  - sslstoretype: type of certificate store for the client certificate. Possible values: 0: user, 1: machine, 2: PFX file, 3: PFX blob, 4: PEM cert and key
  - sslsubject: subject of the certificate used for client authentication. can be set to * for any

Debug

In case if you encounter any issue with SMS sending, you can find out the problem following these steps:

1. Verify the SMS related configuration described above. Search for “sms” in the global config on the “Configurations” form.
2. Note that the Mizu server SMPP module will bind as a transmitter. Receiver and trasreceiver bind is not supported.
3. SMPP connectivity and message delivery can be tested with the mnetutils.exe (found in your VoIP server app folder).
   You can also test with third party tools such as SMPPCli.
4. Check the SMS CDR records (CDR form). Delivered SMS messages has a positive “duration” set after the smstime global config (60 default). Failed SMS messages has 0 duration and you can find details in the disconnect reason and comment fields (with “All Fields” checkbox checked).
   Note: if a separate API request is used for SMS verification then the CDR records are generated with around 6 seconds delay.
5. If the problem is not clear from the CDR, have a look into the server logs (MizuManage -> Control -> Logs -> Show -> Log Folder). Make sure that the logs are enabled (Set to “On” or loglevel is set to 5) and search for “sms” in the logs (or for “sendsms” or for your SMS number or message text).

SMS callback

Using SMS callback you can initiate VoIP calls with SMS messages.
If you subscribe to an SMS provider (like clickatel) the server can receive SMS messages (usually on http port 8084) and initiate actions regarding the content.

By sending an sms messages users can initiate callback, register a new A number for CLI authentication, initiate phone to phone call or add credit to your account.

There are different formats for all functions (short/long - depending on the authentication method). The server can use A number based authentication for the incoming sms messages. If the CLI is hided, or the A number is not specified then the users have to send their username and password or a pincode.
Instead of SMS callback we can recommend the phone to phone functionality. (In this mode the users don't have to work with the IVR. the 2 phone number can be interconnected immediate)

The authentication of the requests is done based on “callingcardauth” global setting.
Pincode can represent the concatenated username and password (if allowed by callingcardauth)

The following formats are defined:

general
username password number1, number2
pincode number
number
pincode
b (pincode based authentication. the A number will be called back)

general requests will result in a p2p call when at least 2 number is known. If only one number is know, than it will result in generating a callback to that number. Two numbers can be supplied, one by sms and the another number can be known from the sender CLI.

callback:
b username password number (username and password based authentication and the number to be called)
b pincode number (pincode based authentication and the number to be called)
b number (pincode based authentication only the number to be called have to be sent)
b (pincode based authentication. the A number will be called back)

register new number:
n username password number
n pincode number
n number

phone to phone:
p username password number1 number2
p pincode number1 number2
p number1 number2
p number2

add credit:
c username password rechargecode number2
c pincode rechargecode number2
c rechargecode number2
c rechargecode

Example http request:

```
POST / HTTP/1.1
User-Agent: Clickatell MO Callback
Host: 99.99.99.99:8084
Pragma: no-cache
Accept:
Referer: 99.99.99.99
Content-Length: 157
```

```
Content-Type: application/x-www-form-urlencoded

api_id=3144985&from=3611111111&to=3622222222&timestamp=2009-07-04 21:39:40&text=Test+1+2+3&charset=ISO-8859-1&udh=&moMsgId=5jv45t5tb42bavu34drftr
```

For callbacks you must define the campaign which will have the proper IVR content by the **defcallbackivr** global configuration setting.

You can set a httpsmscallbackip parameter if you wish to enable callbacks only from a specific IP address.

Additional resources

- Server admin guide
- Documentations
- Support