GOIP SMS Interface

Follow this API, you can develop your sms server to use GoIP doing something.

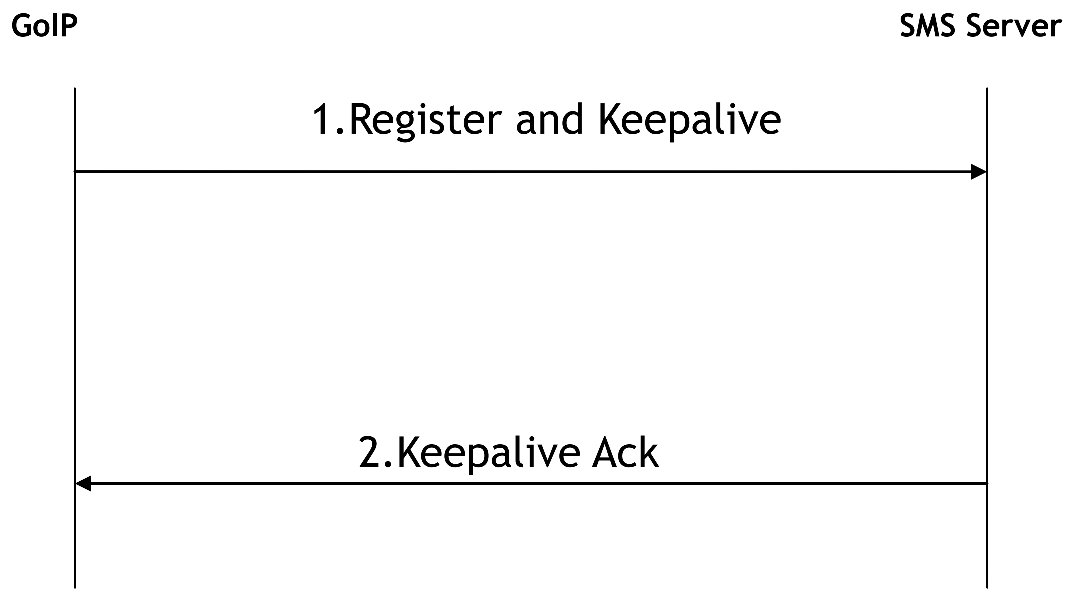
1.Initialization

Communication of this system use clear text for transmission over UDP transport layer. Please setup the retransmission mechanism for reduces the packet loss between the SMS server and GoIP.

SMS Server Initialization: Set the authentication ID and password for GOIP, and save the list of IDs and passwords. Open the UDP port then start listening.

GOIP Parameters: Enable “SMS Sender” on the configuration page, fills your SMS server address and port, authentication ID and password (refer to above “SMS Server Initialization” for setting).

2.Registration and Keepalive



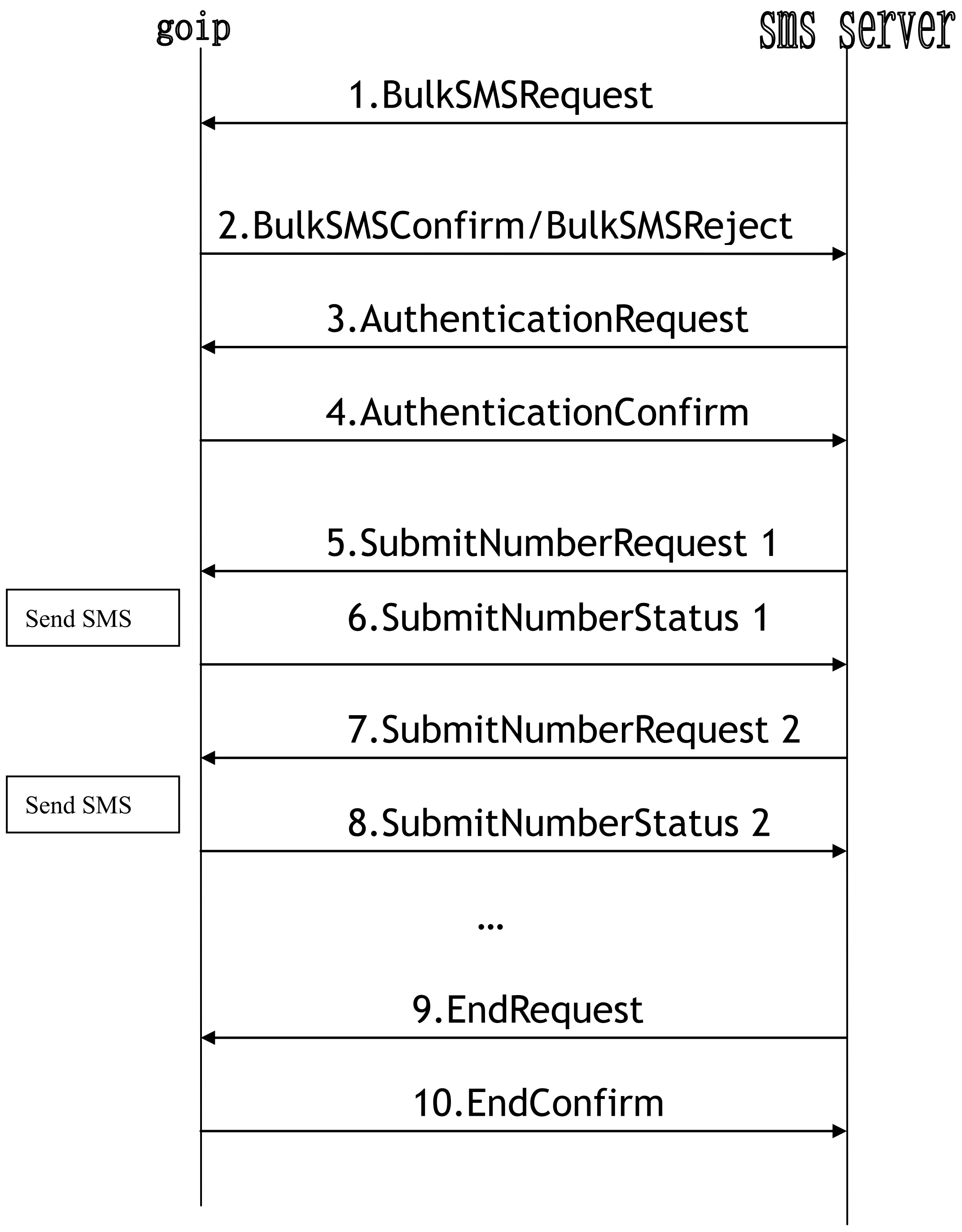
Step	Description
1.Re giste r and Keep	GOIP will send a keepalive packet to server every 30s. Format: req:$count;id:$id;pass:$password;num:$gsm_num;signal:$gsm_signal ;gsm_status:$gsm_status;voip_status:$voip_status;imei:$imei;imsi:$i

alive	msi;iccid:$iccid; Variable: $count: counter for sending packets. Initialize to 1 when GOIP power up and increase by 1 after a packet is sent out. $id: authentication id set in configuration page. $password: authentication password set in configuration page. $gsm_num: sim card number $gsm_signal: sim signal $gsm_status: LOGIN or LOGOUT $voip_status: status of VoIP, LOGIN or LOGOUT $imei: IMEI of GSM Module $imsi: IMSI of SIM Card $iccid: ICCID of SIM Card
2.keepalive ack	SMS server will verify the authentication id and password of the keepalive packet received. And send keepalive ack if the id and password are matched with the authentication list. Format: reg:$count;status:$status; Variable: $count: Integer, the same as Register and Keep alive packet of GoIP. $status: Integer, 200 means ok.

e.g.

GOIP send the keepalive with "req:10;id:goipid1;pass:password1; num:12345;signal:23;gsm_status:LOGIN;voip_status:LOGIN;" to SMS server.
SMS server send the keepalive ack with "reg:10;status:200;" (if goipid1 and password1 is valid.)

3.Send SMS


goip
sms server
1.BulkSMSRequest
2.BulkSMSConfirm/BulkSMSReject
3.AuthenticationRequest
4.AuthenticationConfirm
5.SubmitNumberRequest 1
Send SMS
6.SubmitNumberStatus 1
7.SubmitNumberRequest 2
Send SMS
8.SubmitNumberStatus 2
…
9.EndRequest
10.EndConfirm

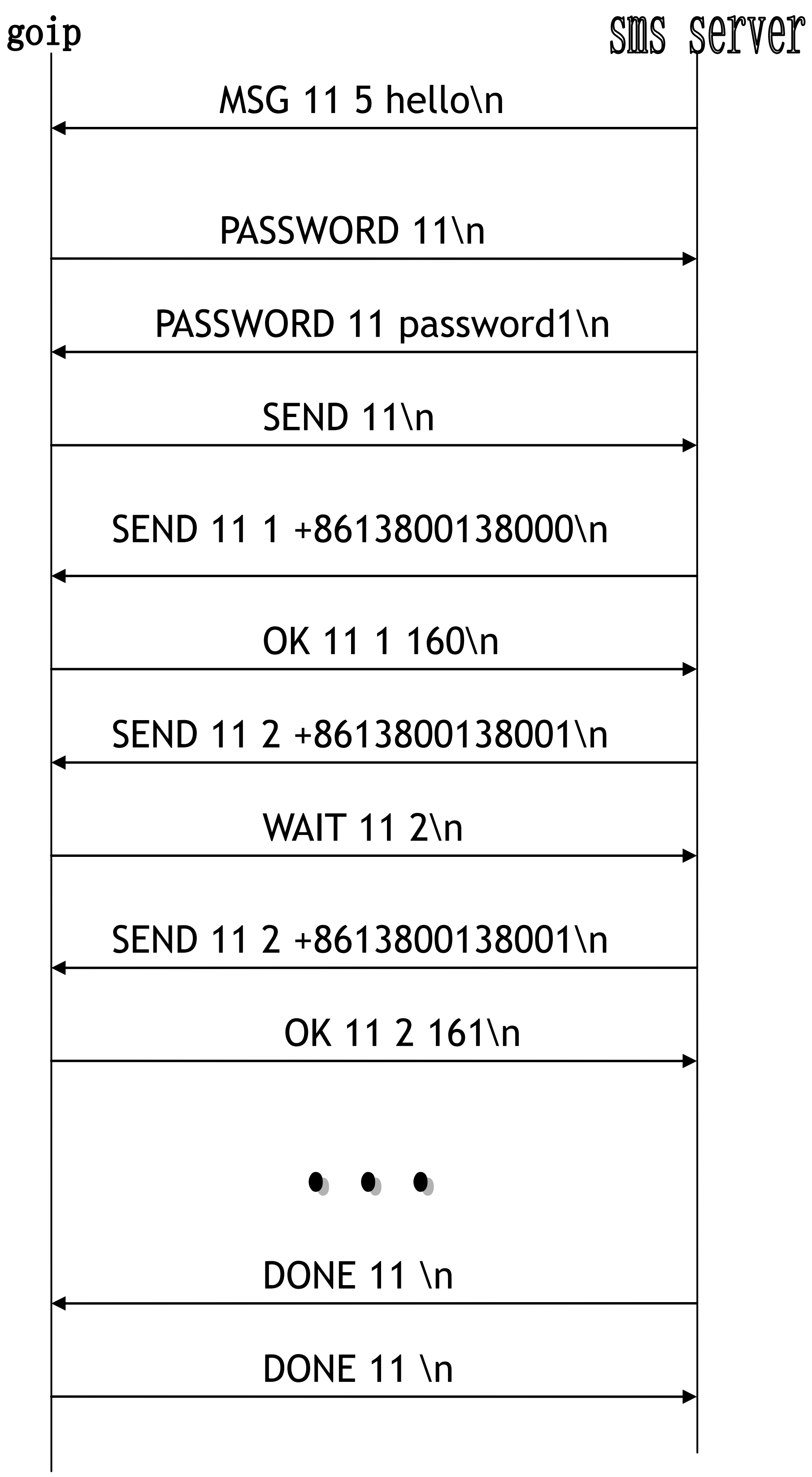
Step	Description
1. BulkSMSRequest	At the beginning, SMS server will send a bulk SMS Request to GOIP, include the SMS content and length in utf8 format. The SMS content should be limited to 3000 bytes or less. Format: MSG $sendid $length $msg\n Variable: $sendid: Integer, as a server packet identifier. Note, all messages described below should use the same "sendid" defined here if they are belong to a same bulk SMS session. $length: Integer in utf8 format, as the SMS content length $msg: SMS content in utf8 format
2.BulkSMSConfirm BulkSMSReject	GOIP will do the initialization for the bulk SMS when Bulk SMS Request received. Note, the other messages of this bulk SMS should be received in 90s after Bulk SMS Request received. Or, GOIP will stop the session and release. (1) When the initialization successful, GOIP will send a BulkRequest Confirm to server and request for authentication Format: PASSWORD $sendid\n Variable: $sendid: Integer, bulk SMS session identifier in BulkSMSRequest. (2) When the initialization failed, GOIP will send a BulkSMSReject to server, include the error message. Format: ERROR $sendid $errormsg\n Variable: $sendid: Integer, bulk SMS session identifier in BulkSMSRequest. $errormsg: String, Error message. Now too many request for bulk SMS received at the same time is the main reason of the failure. Note, GOIP only support 5 bulk SMS session at the same time. And, server should stop the session when receive bulkSMSRequest Reject. Then, try again later.
3.AuthenticationRequest	SMS server should send a AuthenticationRequest with the password to GOIP. Format: PASSWORD $sendid $password\n

	Variable: $sendid: Integer, bulk SMS session identifier in BulkSMSRequest. $password: The registration password of GOIP
4.AuthenticationConfirm AuthenticationReject	GOIP will verify the password in AuthenticationRequest and return the result to server. (1) When authentication successful, GOIP will send a AuthenticationConfirm to server. Format: SEND $sendid\n Variable: $sendid: Integer, bulk SMS session identifier in BulkSMSRequest. (2) When authentication failed, GOIP will send a AuthenticationReject to server and wait for the next AuthenticationRequest or the EndRequest Format: ERROR $sendid PASSWORD\n Variable: $sendid: Integer, bulk SMS session identifier in BulkSMSRequest.
5.SubmitNumberRequest	GOIP need about 2-5 seconds to send a SMS. So, to avoid packet loos, SMS server could send a SubmitNumberRequest to GOIP to get the sending status of an appointed number every serial seconds until the SMS is sent successfully.And, GOIP only save the sending status of the last 10 numbers in this group Format: SEND $sendid $telid $telnum\n Variable: $sendid: Integer, bulk SMS session identifier in BulkSMSRequest. $telid: Integer, the unique SubmitNumberRequest sequence number defined by server. $teinum: String, telephone number
6. SubmitNumberStatus	GOIP will send a SubmitNumberStatus to server when the SubmitNumberRequest received. (1) when sending SMS to appointed number successful, GOIP will send a SubmitNumberStatus with OK to server. Format: OK $sendid $telid $sms_no \n Variable: $sendid: Integer, bulk SMS session identifier in BulkSMSRequest.

	$telid: Integer, unique sequence number in SubmitNumberRequest. $sms_no: number count of SMS sending in GoIP (2) When sending failed, GOIP will send a SubmitNumberStatus with ERROR to server. Format: ERROR $sendid $telid errorstatus:$errorid\n Variable: $sendid: Integer, bulk SMS session identifier in BulkSMSRequest. $telid: Integer, unique sequence number in SubmitNumberRequest. $errorid: Integer, error code. Usually it is equal to 1. (3) When the telid in SubmitNumberRequest is not in the list of which save the recent 10 sending number, GOIP will save the telid in the waiting list and send a SubmitNumberStatus with WAIT. Format: WAIT $sendid $telid\n Variable: $sendid: Integer, bulk SMS session identifier in BulkSMSRequest. $telid: Integer, unique sequence number in SubmitNumberRequest.
7.SubmitNumberRequest	Note, SMS server should send the next SubmitNumberRequest to GOIP after the SubmitNumberStatus with OK or ERROR received Same as 5
8. SubmitNumberStatus	Same as 6
9. EndReqeust	SMS server could send a EndRequest to GOIP to finish the bulk SMS session. Format: DONE $sendid\n Variable: $sendid: Integer, bulk SMS session identifier in BulkSMSRequest.
10.End	GOIP will release the bulk SMS session resource when EndRequest received. And return a EndConfirm to server Format: DONE $sendid\n Variable: $sendid: Integer, bulk SMS session identifier in BulkSMSRequest.

Example:

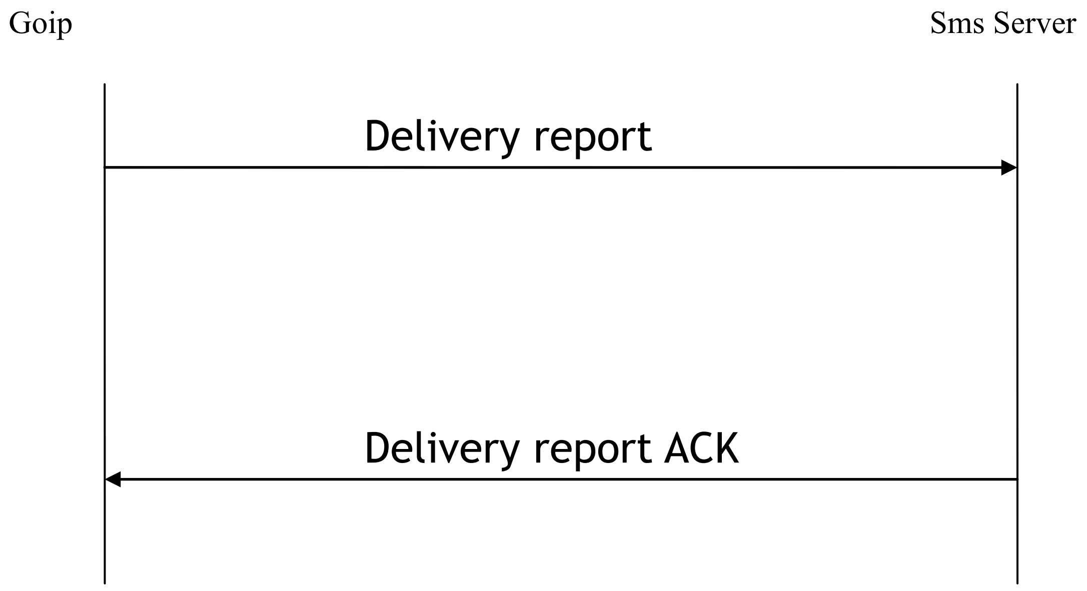
Assume: sendid: 11, SMS content: hello, Goip password:password1, send a sms to +8613800138000 and +8613800138001.



3. 1 Delivery Reports

If carrier report delivery of sms sending, goip will send

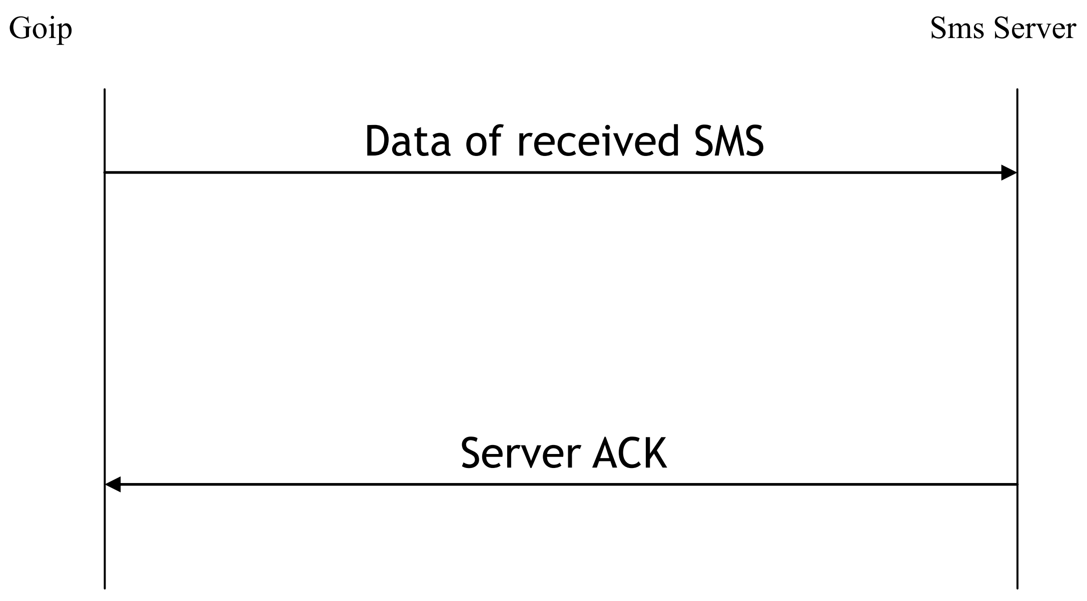
delivery report to sms server.



Step	Description
1.Send delivery report.	When received delivery report, Goip will send it to Server. Goip will resend the data if Server no response in 3 seconds, most resend 3 times. Format: DELIVER:$recvid;id:$id;password:$password;sms_no:$sms_no;state:$deliver_state;num:$send_num Variable: $recvid:int, count with the current time stamp decreasing; $id: authentication id set in configuration page. $password: authentication password set in configuration page. $sms_no: int, goip create when send sms (see 3 send sms) $deliver_state: 0 if sms received $send_num: the number sms send to
2.delivery report ACK	Sms server will receive the report and check the goip id and password, then send a ACK Successful Format：DELIVER $recvid OK\n Error Format：DELIVER $recvid ERROR $errmsg Variable:$recvid: int, the goip count; $errmsg:error massge

4. RECEIVE SMS

Data of received SMS



Step	Description
1.Send data of receiverSMS to Server.	When received SMS, Goip will relay the SMS to Server. Goip will resend the data if Server no response in 3 seconds, most resend 3 times. Format: RECEIVE:$recvid;id:$id;password:$password;srcnum:$srcnum;msg:$msg Variable: $recvid:int, count with the current time stamp decreasing; $id: authentication id set in configuration page. $password: authentication password set in configuration page. $srcnum: Source mobile number $msg: Content of SMS，utf8 format
2.Server ACK	Sms server will receive SMS and check the goip id and password, then send a ACK. Successful Format：RECEIVE $recvid OK\n Error Format：RECEIVE $recvid ERROR $errmsg Variable:$recvid: int, the goip count; $errmsg:error massge

Example:

Assume: A Goip(id:goipid1, password:password1)received a SMS “just a test” from mobile “+8613513415667”. And it got a count 1270197307, then will send to SMS Server like this:

”RECEIVE:1270197307;id:goipid1;pass:password1;srcnum:+8613513415667;msg:just a test”

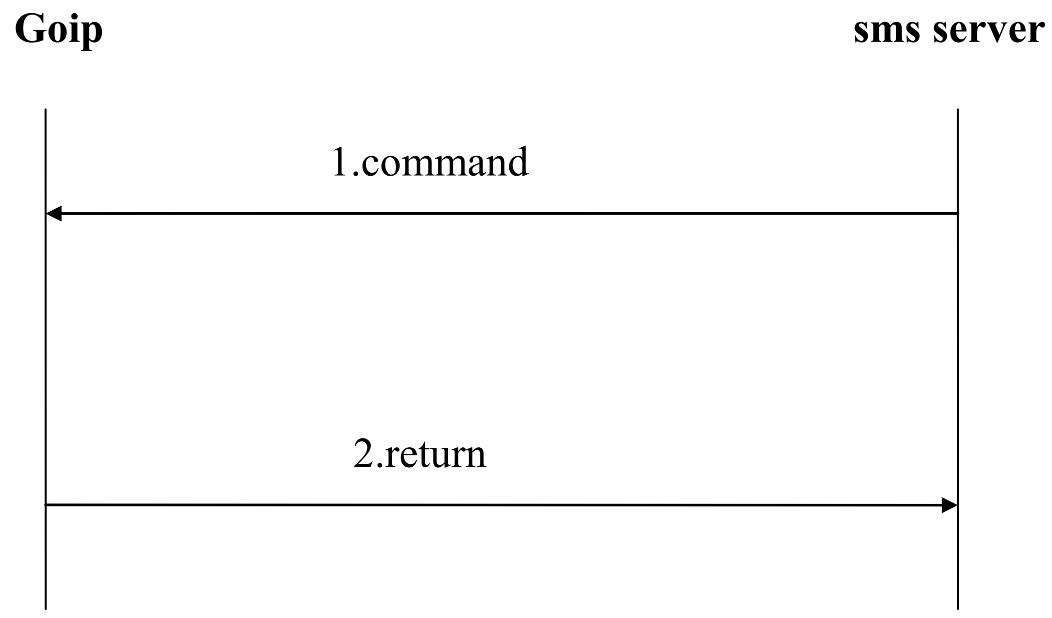
Sms server check goip id and password, Saved the SMS data, and reply:

"RECEIVE 1270197307 OK"

5. Get status of GoIP and control Goip.

5.1 Server send a command to Get status of GoIP or control Goip

Server should resend packet several times when cannot receive reply from goip.



5.1.1 Get GSM number

Step	Description
1.server send command to get GSM number	Format: get_gsm_num $sendid $password Variable: $sendid: integer, the id of server packet increase by 1; $password: The registration password of GOIP;
2.Goip 回应	Goip return OK Format: get_gsm_num $sendid $gsmnum Error Format: ERROR $sendid $errmsg Variable: $sendid:the same as server packet; $gsmnum:GSM number;

	$errmsg: string of error message

e.g:

server send

"get_gsm_num 879901 password1"

Goip send

"get_gsm_num 879901 13800138000"

5.1.2 Set GSM number

Step	Description
1.server send	Format: set_gsm_num $sendid $gsmnum $password Variable: $sendid: integer, the id of server packet increase by 1; $gsmnum:the gsm number which want to set $password: The registration password of GOIP;
2.Goip return	OK Format: set_gsm_num $sendid $gsmnum ok Error Format: ERROR $sendid $gsmnum $errmsg Variable: $sendid:the same as server packet; $gsmnum: the gsm number which want to set; $errmsg: string of error message;

e.g.

server send

"set_gsm_num 879902 13800138001 password1"

Goip return

"set_gsm_num 879902 13800138001 ok"

5.1.3 Get expiry time of out call of a channel

Step	Description
1.server send	Format: get_exp_time $sendid $password Variable: $sendid: integer, the id of server packet increase by 1; $password: The registration password of GOIP;
2.Goip return	OK Format: get_exp_time $sendid $exptime Error Format: ERROR $sendid $errmsg Variable:

	$sendid:the same as server packet; $exptime: expiry time of out call of a channel (minute); $errmsg: string of error message;

5.1.4 Set expiry time of out call of a channel

Step	Description
1.server send	Format: set_exp_time $sendid $exptime $password Variable: $sendid: integer, the id of server packet increase by 1; $exptime: expiry time (minute) witch want to set $password: The registration password of GOIP;
2.Goip return	OK Format: set_exp_time $sendid $exptime ok Error Format: ERROR $sendid $exptime $errmsg Variable: $sendid:the same as server packet; $exptime: expiry time of out call of a channel (minute); $errmsg: string of error message;

5.1.5 Get Remain time of out call

Step	Description
1.server send	Format: get_remain_time $sendid $password Variable: $sendid: integer, the id of server packet increase by 1; $password: The registration password of GOIP;
2.Goip return	OK Format: get_remain_time $sendid $remaintime Error Format: ERROR $sendid $errmsg Variable: $sendid:the same as server packet; $remaintime: remain time of out call(minute) $errmsg: string of error message;

5.1.6 Reset remain time of out call to expiry time

Step	Description
1.server send	Format:

	reset_remain_time $sendid $password Variable: $sendid: integer, the id of server packet increase by 1; $password: The registration password of GOIP;
2.Goip return	OK Format: reset_remain_time $sendid ok Error Format: ERROR $sendid $errmsg Variable: $sendid:the same as server packet; $errmsg: string of error message;

5.1.7 Get status of channel

Step	Description
1.server send	Format: get_gsm_state $sendid $password Variable: $sendid: integer, the id of server packet increase by 1; $password: The registration password of GOIP;
2.Goip return	OK Format: get_gsm_state $sendid $state Error Format: ERROR $sendid $errmsg Variable: $sendid:the same as server packet; $state: status of line，string(IDLE or ACTIVE) $errmsg: string of error message;

5.1.8 Drop call

Step	Description
1.server send	Format: svr_drop_call $sendid $password Variable: $sendid: integer, the id of server packet increase by 1; $password: The registration password of GOIP;
2.Goip return	GoIP will try to drop the current call. OK Format: svr_drop_call $sendid $ok Error Format: ERROR $sendid $errmsg Variable: $sendid:the same as server packet; $errmsg: string of error message;

5.1.9 Reboot channel

Step	Description
1.server send	Format: svr_reboot_module $sendid $password Variable: $sendid: integer, the id of server packet increase by 1; $password: The registration password of GOIP;
2.Goip return	Goip will try to reboot channel OK Format: svr_reboot_module $sendid $ok Error Format: ERROR $sendid $errmsg Variable: $sendid:the same as server packet; $errmsg: string of error message;

5.1.10 Reboot GoIP

Step	Description
1.server send	Format: svr_reboot_dev $sendid $password Variable: $sendid: integer, the id of server packet increase by 1; $password: The registration password of GOIP;
2.Goip return	Goip will try to reboot OK Format: svr_reboot_dev $sendid $ok Error Format: ERROR $sendid $errmsg Variable: $sendid:the same as server packet; $errmsg: string of error message;

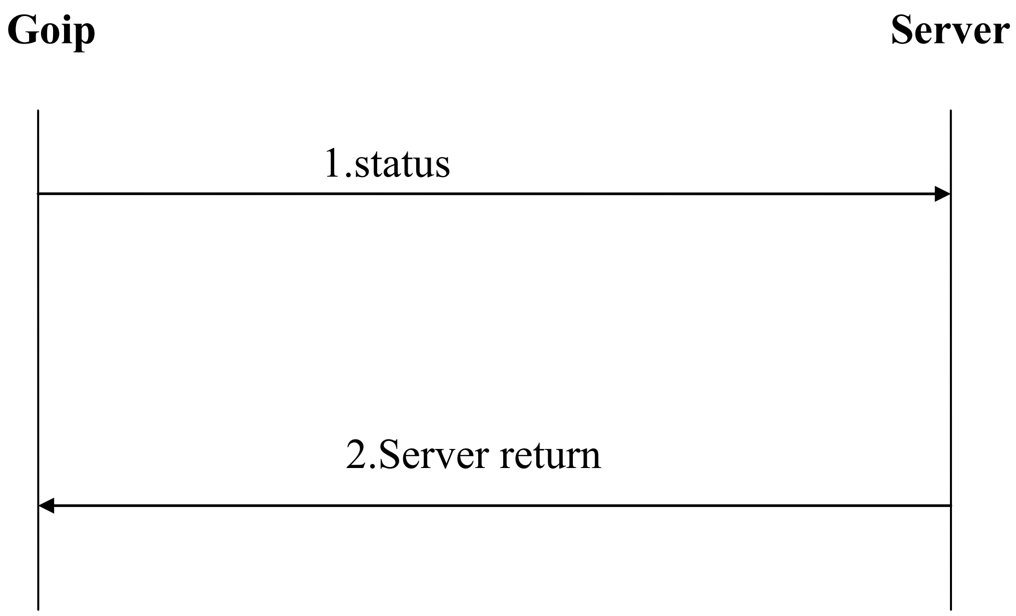
5.1.11 Set GSM call forward

Step	Description

1.server send	Format: CF $sendid $password $reason $mode $num $ftime Variable: $sendid: integer, the id of server packet increase by 1; $password: The registration password of GOIP; $reason: type of call forward. 0: unconditional， 1: busy， 2: noreply， 3: noreachable, 4: all， 5:busy,noreply,noreachable； $mode: enable or disable forward。3:enable，4:disable。 $num: forward to this number $ftime: timeout (second) of noreply forward type. Other types set to 0.
2.Goip return	Goip will try to set call forward. OK Format: CFOK $sendid Error Format: CFERROR $sendid
3. server return	Format: DONE $sendid

5.2 Goip send status to server

When status of Goip changed or goip in a call， Goip sends state to server。



5.2.1 When status of channel of goip changed, goip send the

status to server.

Step	Description
1.Goip send status of channel	Format: STATE:$recvid;id:$goipid;password:$password;gsm_remain_state:$state Variable: $recvid: :int, count with the current time stamp decreasing; $goipid: authentication id set in configuration page. $password: authentication password set in configuration page. $state:String of status (IDEL, BUSY)
2.Server return	Goip return to goip. OK format: STATE $recvid OK Error format: STATE $recvid $errmsg Variable: $recvid: the same as goip send. $errmsg: error message, defined of you.

e.g
goip send to server:

STATE:7568;id:goip1;password:password1;gsm_remain_state:IDLE

Server return :

STATE 7568 OK

5.2.2 When Goip in a call, goip send status of call to server。

Step	Description
1.Goip send status of call to server	Format: RECORD:$recvid;id:$goipid;password:$password;dir:$dir;num:$num Variable: $recvid: int, count with the current time stamp decreasing; $goipid: authentication id set in configuration page. $password: authentication password set in configuration page. $dir: int，means direction of call. 1:INCOMING，2:OUTGOING
2.Server return	Goip return to goip. OK format: RECORD $recvid OK Error format: RECORD $recvid $errmsg Variable: $recvid: the same as goip send. $errmsg: error message, defined of you

e.g
goip send to server:

RECORD:7565;id:goip2;password:password2;dir:2;num:10086

Server return :

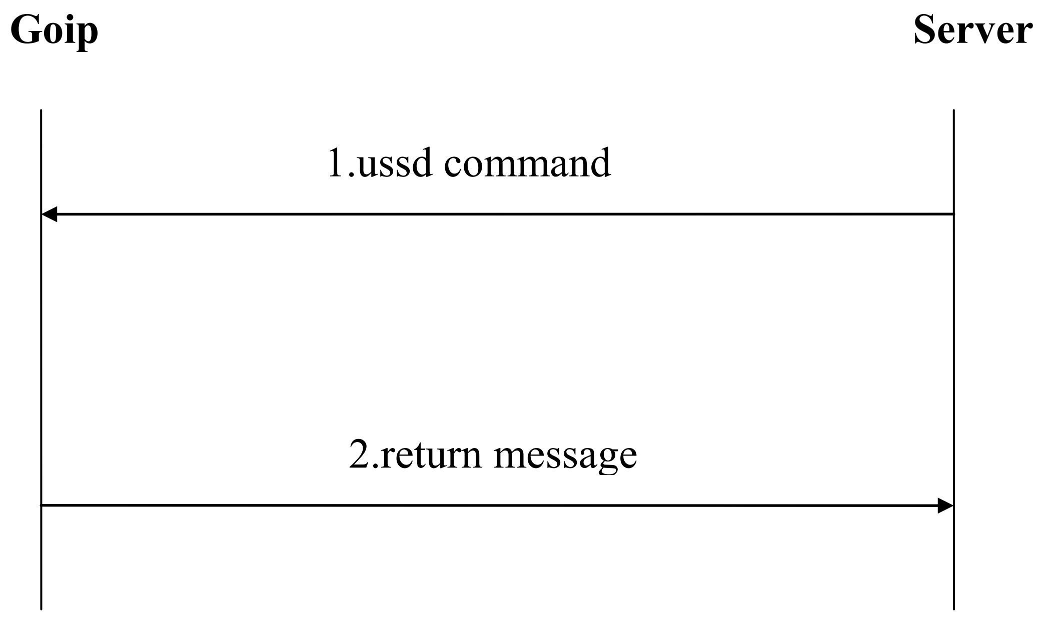
RECORD 7565 OK

5.2.3 after each call ,goip send remain time to server.

Step	Description
1.Goip send remain tiem	Format: REMAIN:$recvid;id:$goipid;password:$password;gsm_remain_time:$time Variable: $recvid: int, count with the current time stamp decreasing; $goipid: authentication id set in configuration page. $password: authentication password set in configuration page. $time:remain time of a channel of out call, minutes。
2.Server 回应	OK format: REMAIN $recvid OK Error format: REMAIN $recvid $errmsg Variable: $recvid: the same as goip send. $errmsg: error message, defined of you

6. Send USSD

It can be used for recharging SIMS (We need to recharge sims by sending EX: (*150*543164649761# Call. For api, just send USSD command "*150*543164649761#") .



Step	Description

1.server send ussd command	If server cannot receive client's answer for 10 seconds, resend command once. Format: Common USSD command： USSD $sendid $password $ussdcmd Exit USSD command： USSDEXIT $sendid $password Variable: $sendid: int, count of the sending packets increase by 1; $password: The registration password of GOIP; $ussdcmd: USSD command
2.Goip return	Goip received the command , send the command to provider，then renturn the reply to server。 Format: USSD $sendid $msg Error format：USSDERROR $sendid $errmsg USSD disconneted：USSDEXIT $sendid Variable： $sendid: the same as server send; $msg: the message of provider return , utf8 code $errmsg: error message

e.g.

Server send to goip "USSD 111 password1 557*0112220248*10#"

Then goip renturn to server "USSD 111 You are going to transfer 10.00 EGP to 20112220248. The service fee is 2% with a minimum of 0.20 EGP. Press 1 to confirm, or any key to cancel"

7. IMEI

7.1 Get IMEI

Step	Description
1.server send	Format: get_imei $sendid $password Variable: $sendid: integer, the id of server packet increase by 1; $password: The registration password of GOIP;
2.Goip return	OK Format: get_imei $sendid $imei Error format: ERROR $sendid $errmsg Variable: $sendid:the same as server packet; $imei:IMEI number

	$errmsg: string of error message;

7.2 Set IMEI

Step	Description
1.server send	Format: set_imei $sendid $imei $password Variable: $sendid: integer, the id of server packet increase by 1; $imei: IMEI number，15 digits. $password: The registration password of GOIP;
2.Goip return	OK Format: set_imei $sendid $imei ok Error format: ERROR $sendid $errmsg Variable: $sendid:the same as server packet; $imei:IMEI number $errmsg: string of error message;

8. Out call interval

8.1 Get out call interval

Step	Description
1.server send	Format: get_out_call_interval $sendid $password Variable: $sendid: integer, the id of server packet increase by 1;; $password: The registration password of GOIP;
2.Goip return	OK Format: get_out_call_interval $sendid $interval Error format: ERROR $sendid $errmsg Variable: $sendid:the same as server packet; $interval: out call interval (second) $errmsg: string of error message;

8.2 Set out call interval

Step	Description
1.server send	Format: set_out_call_interval $sendid $interval $password Variable: $sendid: integer, the id of server packet increase by 1;; $interval: out call interval (second)

	$password: The registration password of GOIP;
2.Goip return	OK format: set_out_call_interval $sendid $interval ok Error format: ERROR $sendid $errmsg Variable: sendid:the same as server packet; $interval: out call interval (second) $errmsg: string of error message;

9 enable/disable module

9.1 enable/disable this module

Step	Description
1.server send	Format: module_ctl_i $sendid $value $password Variable: $sendid: integer, the id of server packet increase by 1; $value: 1 to enable, 2 to disable. $password: The registration password of GOIP;
2.Goip return	OK format: module_ctl_i $sendid $ok Error format: ERROR $sendid $errmsg Variable: sendid:the same as server packet; $errmsg: string of error message;

9.2 enable/disable all modules

Step	Description
1.server send	Format: module_ctl $sendid $value $password Variable: $sendid: integer, the id of server packet increase by 1; $value: 1 to enable, 0 to disable, 2 to not change, each digit for each channel. For example 10120121, means channel 1 enable, channel 2 disable, channel 3 enable, channel 4 not change, channel 5 disable, channel 6 enable, channel 7 not change and channel 8 enable. $password: The registration password of GOIP;
2.Goip return	OK format: module_ctl $sendid $ok Error format: ERROR $sendid $errmsg Variable: sendid:the same as server packet; $errmsg: string of error message;

10.Cells

10.1 GoIP send cells list to server

Step	Description
1.goip send cells list to server when goip read cells.	Format: CELLS:$recvid;id:$id;password:$password;lists:$cell_list Variable: $recvid: int, count with the current time stamp decreasing; $id: authentication id set in configuration page. $password:The registration password of GOIP; $cell_list: cells list strings, each cell as a number split by ‘,’, for example 123,456,789,10,
2.Server return	OK format: CELLS $recvid OK\n Error format: CELLS $recvid ERROR $errmsg Variable: $recvid:the same as goip packet; $errmsg: string of error message;

10.2 Set cell

Step	Description
1.server send command to goip to set cell	Format: set_base_cell $sendid $cell_id $password Variable: $sendid: integer, the id of server packet increase by 1; $cell_id: cell id $password: The registration password of GOIP;
2.Goip return	OK format: set_base_cell $sendid $cell_id ok Error format: ERROR $sendid $errmsg Variable: sendid:the same as server packet; $errmsg: string of error message; $cell_id: cell id

10.3 Get cells list

Step	Description
1.server send command	Format: get_cells_list $sendid $password Variable: $sendid: integer, the id of server packet increase by 1; $password: The registration password of GOIP;
2.Goip return	GoIP get the command ， then rtuen OK message. GoIP will get the cells list from operator then goip send a command with cells list

	information to server.(command in 10.1) OK format: get_cells_list $sendid ok Error format: ERROR $sendid $errmsg Variable: $sendid: the same as server packet; $errmsg: string of error message;

10.4 Get current cell

Step	Description
1.server send	Format: CURCELL $sendid $password Variable: $sendid: integer, the id of server packet increase by 1; $password: The registration password of GOIP;
2.Goip return	OK format: CURCELL $sendid $curcellid Error format: ERROR $sendid $errmsg Variable: $sendid: the same as server packet; $curcellid: goip current cell id $errmsg: string of error message;