

# GOIP SMS Interface

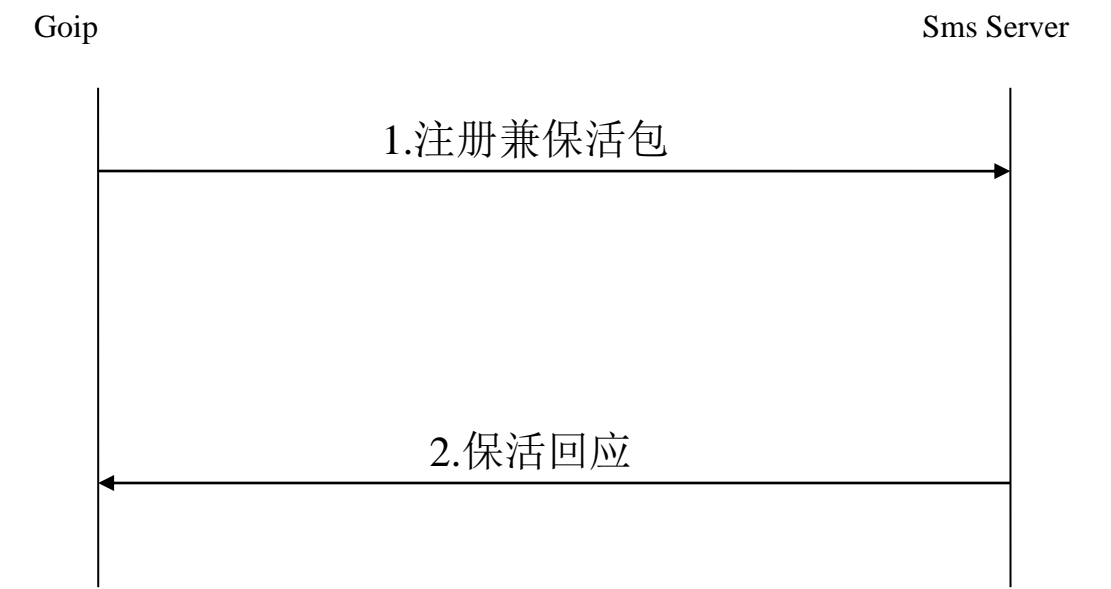
依照本文档，你可以开发一个 SMS server 来收发 GoIP 的短信以及进行获取 GoIP 状态和控制 GoIP 的操作。

## 1.初始化

本系统采用 udp 通讯，传输明文内容。请在 sms server 端建立 udp 的丢包重发机制。

sms server 初始化：设置 goip 的认证 id 和密码，保存 id 和密码列表。开启 udp 端口，开始 udp 监听  
goip 参数设置：在 goip 配置页面启用“短信发送客户端”，填写 sms server 的地址和端口，填写 sms server 设置好的认证 id 和密码，重启 goip 设备。

## 2.注册和保活交互



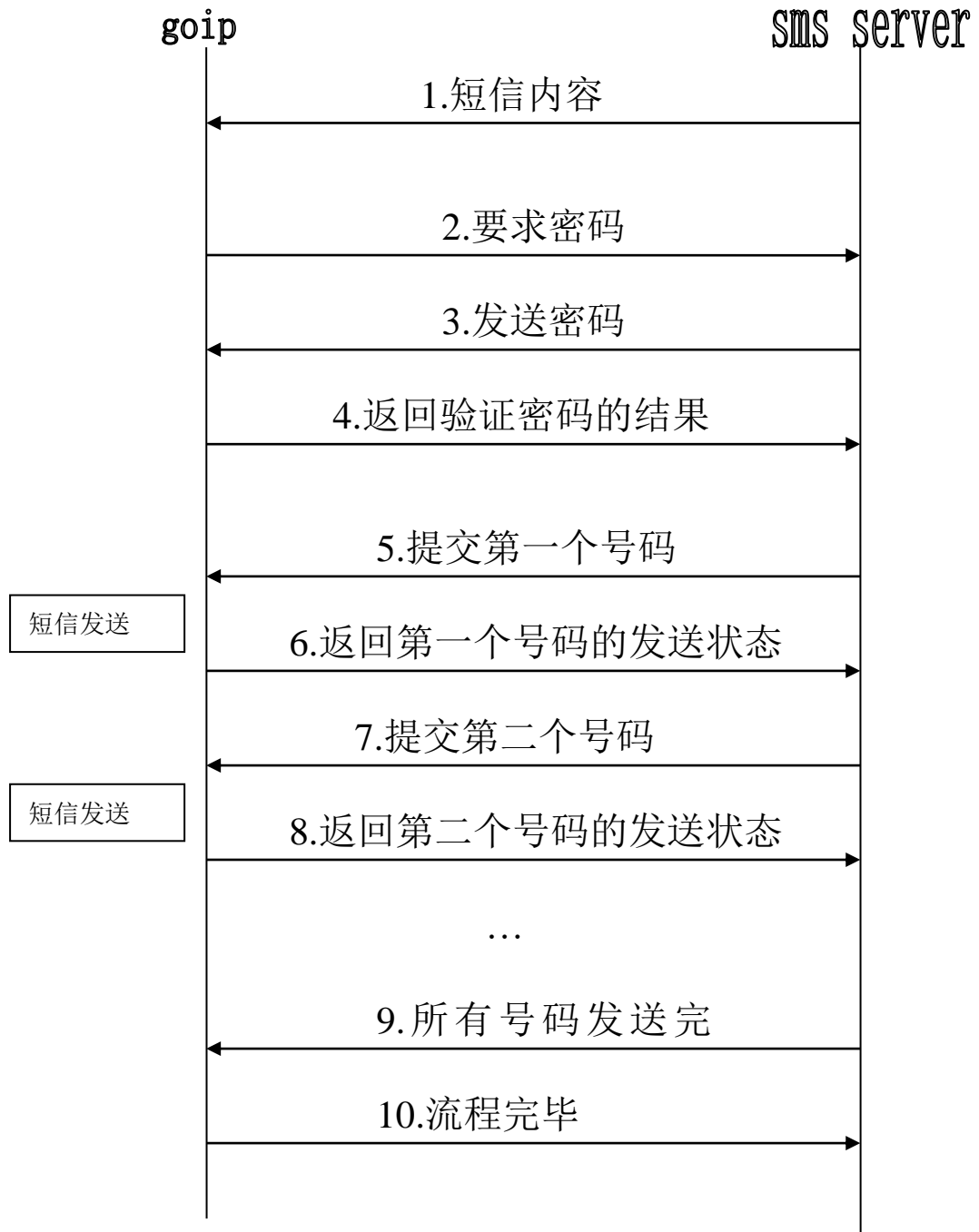
步骤	描述
1.注册兼保活包	Goip 每间隔 30 秒发送一个 udp 包，兼有注册和保活功能。 格式： req:\$count;id:\$id;pass:\$password;num:\$gsm_num;signal:\$gsm_signal;gsm_status:\$gsm_status;voip_status:\$voip_status;imei:\$imei;imsi:\$imsi;iccid:\$iccid; 变量： \$count:计数,goip 启动后由 1 开始递增，每发一个保活包增 1; \$id:在 goip 配置页面填写的注册的 id; \$password: 在 goip 配置页面填写的注册密码;

	\$gsm_num: 在 goip 配置页面填写的 gsm sim 卡号码 \$gsm_signal: sim 卡的信号强度 \$gsm_status: gsm 的注册状态, 分为 LOGIN 和 LOGOUT \$voip_status: SIP 或 H323 的注册状态, 分为 LOGIN 和 LOGOUT \$imei:GSM 模块的 IMEI 号 \$imsi: SIM 卡的 IMSI \$iccid: SIM 卡的 ICCID
2 . 保 活 回 应	Sms server 收到保活包后, 匹配 id 列表中的 id 和密码, 并发出的回应 格式: reg:\$count;status:\$status; 变量: \$count:计数;\$status:整数, sms server 定义的 goip 注册状态, 以 0 为已 匹配上。Goip 暂不对此值进行处理;

例如: goip 向 sms server 发送: " req:10;id:goipid1;pass:password1;"

Sms server 匹配 id 和密码成功, 返回:" reg:10;status:0;"

### 3.群发短信流程



步骤	描述
1.发起一组发送	<p>Sms server 发起一组群发，其中<b>包含短信 utf8 内容和长度，内容要求在 3000 字节内</b></p> <p>格式：MSG \$sendid \$length \$msg\n</p> <p>变量：\$sendid:整数,sms server 定义的短期内不重复的发送序号，以下都使用该序号来识别本组发送;\$length:整数，短信长度;\$msg:短信内容,utf8 编码</p>
2.要求密码	<p>Goip 为本组群发进行初始化。<b>如果 goip90 秒内没收到由关本次群发的通讯，将结束本组发送，释放资源</b></p> <p>(1) 初始化成功，要求密码以确认是否是有权限的发送请求。</p> <p>格式：PASSWORD \$sendid\n</p> <p>变量：\$sendid:整数,sms server 定义的发送序号</p> <p>(2) <b>初始化失败</b>，主要是正在进行多个群发导致失败，错误信息：“too_many_sends_in_server”（goip 现在支持 5 个群发同时进行），sms server 收到此信息应当结束群发，稍后再试。</p> <p>格式：ERROR \$sendid \$errmsg\n</p> <p>变量：\$sendid:整数,sms server 定义的发送序号;\$errmsg:字符串,出错信息</p>
3.发送密码	<p>Sms server 发送密码</p> <p>格式：PASSWORD \$sendid \$password\n</p> <p>变量：\$sendid:整数,sms server 定义的发送序号;\$password:goip 注册密码</p>
4.返回验证密码的结果	<p>Goip 验证密码，返回结果。</p> <p>(1) 成功</p> <p>格式：SEND \$sendid\n</p> <p>变量：\$sendid:整数,sms server 定义的发送序号;</p> <p>(2) <b>失败，等待 sms server 再次传来密码或传来结束信息</b></p> <p>格式：ERROR \$sendid PASSWORD\n</p> <p>变量：\$sendid:整数,sms server 定义的发送序号;</p>
5.提交第一个号码	<p>Sms server 提交第一个号码，得到它的发送状态。Goip 发送一条短信用时约为 2-5 秒。为保证 udp 不丢包可以隔数秒发送本指令获取状态。Goip 保存本组最近的 10 个号码的发送状态，可以反复发送本指令获取。</p> <p>格式：SEND \$sendid \$telid \$telnum\n</p> <p>变量：\$sendid:整数,sms server 定义的发送序号;\$telid:整数，sms server 定义的一组发送中唯一的序号;\$telnum:字符串,电话号码;</p>
6.返回第一个号码的发送状态	<p>Goip 接收到“SEND \$sendid \$telid \$telnum\n”指令时，如果该\$telid 不在最近 10 个发送 ID 中，将把该号码放入待发送列表,立即返回下列第三个状态，发送完毕再返回下列第一或第二种状态。如果该\$telid 在最近 10 个发送 ID 中，则立即返回下列状态之一：</p> <p>(1) 发送完成，已经成功的对该号码发送了短信</p>

	格式: OK \$sendid \$telid\n 变量: \$sendid:整数,sms server 定义的发送序号;\$steiid:sms server 定义的号码序号; (2) 发送失败 格式: ERROR \$sendid \$telid errorstatus:\$errorid\n 变量: \$errorid:整数,错误号,一般是 1, 模块发送短信失败。 (3) 等待发送, 正在发送, 或者模块正发送其他短信 格式: WAIT \$sendid \$telid\n 变量:
7.提交第二个号码	Sms server 收到 OK \$sendid \$telid\n 或 ERROR \$sendid \$telid errorstatus:\$errorid\n, 再发送下一个号码。 同 5
8.返回第二个号码的发送状态	同 6
9.所有号码发送完毕	Sms server 处理完了一组群发中的所有号码, 向 goip 发送本组群发结束的指令 格式: DONE \$sendid\n 变量: \$sendid:整数,sms server 定义的发送序号;
10.流程完毕	Goip 释放资源, 流程结束 格式: DONE \$sendid\n 变量: \$sendid:整数,sms server 定义的发送序号;

例如: 假设 sms server 定义 sendid 为 11, 发送短信内容为”hello”;

Sms server 发起一次群发: ”MSG 11 5 hello\n”;

Goip 群发初始化成功, 要求密码 “PASSWORD 11\n”

Sms server 传递密码 “PASSWORD 11 password1\n”

Goip 对比密码成功,可以接收电话号码了 “SEND 11\n”

Sms Server 开始传递电话号码 “SEND 11 1 +8613800138000\n”

Goip 回应已接收请等待 “WAIT 11 1\n”

Goip 发送本号码完毕 “OK 11 1\n”

Sms Server 传递第二个电话号码 “SEND 11 2 +8613800138001\n”

Goip 回应已接收请等待 “WAIT 11 2\n”

Sms Server 一个重发的包“SEND 11 2 +8613800138001\n”

Goip 回应依然在等待状态 “WAIT 11 2\n”

Goip 发送本号码完毕 “OK 11 2\n”

Sms Server 又一个重发的包“SEND 11 2 +8613800138001\n”

Goip 回应已发送完毕 “OK 11 2\n”

Sms Server 传递第三个电话号码 “SEND 11 3 +8613800138002\n”

...

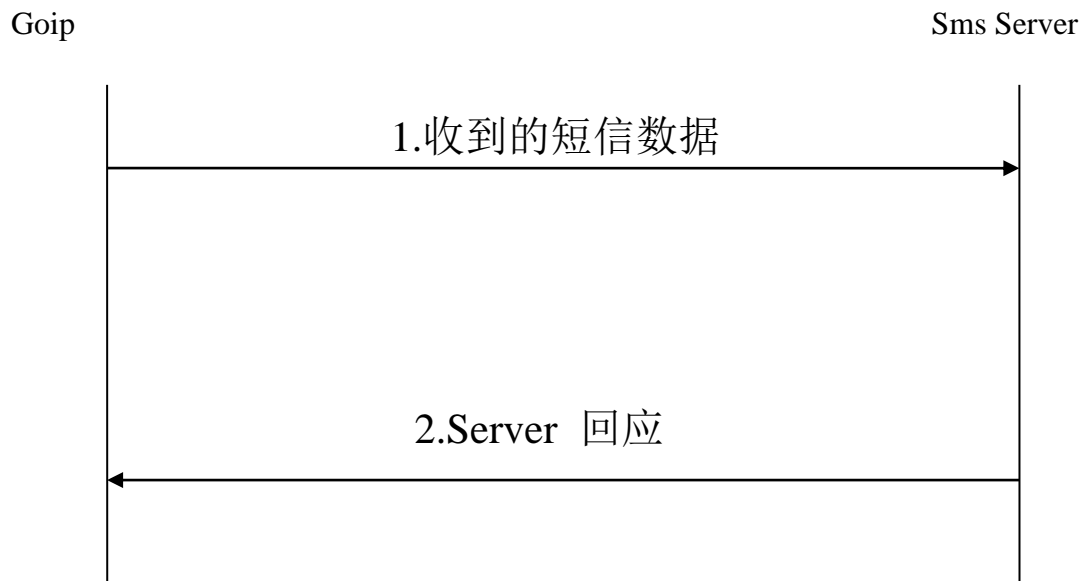
...

...

Sms Server 处理完全部号码 “DONE 11\n”

Sms Server 释放资源，回应完毕 “DONE 11\n”

## 4.接收短信流程



步骤	描述
1. 收到的短信数据	<p>Goip 把接受到短信转发给 server，默认 3 秒 server 无响应则重发，共重发 3 次。</p> <p>格式： RECEIVE:\$recvid;id:\$id;password:\$password;srcnum:\$srcnum;msg:\$msg</p> <p>变量： \$recvid:计数,GOIP 以当前时间戳递减计数; \$id:在 goip 配置页面填写的注册的 id; \$password:在 goip 配置页面填写的注册密码; \$srcnum:短信来源的手机号码 \$msg 短信内容，utf8 编码</p>
2.Server 回应	<p>Sms server 收到短信后，查找 id 列表中的 id 和密码，并发出的回应</p> <p>成功格式：RECEIVE \$recvid OK\n</p> <p>失败格式：RECEIVE \$recvid ERROR \$errmsg</p> <p>变量：\$recvid:计数; \$errmsg:出错信息</p>

例如：goip 收到手机+8613513415667 的短信”just a test’，而当前的接收短信计数为 1270197307，则向 sms server 发送：

”RECEIVE:1270197307;id:goipid1;pass:password1;srcnum:+8613513415667;

msg:just a test”  
Sms server 匹配 id 和密码成功，保存短信，返回：  
”RECEIVE 1270197307 OK”

5.Goip 信息读取和控制

5.1 Server 发起获取信息或控制命令

Server 向 goip 发出一条命令,用来获取修改 goip 信息或控制 goip.此功能 Server 端应建立无应答超时和重发机制，推荐 5 秒 goip 无响应则重发，重发 2 次。



5.1.1 GSM 号码获取

步骤	描述
1.server 发送 GSM 号码获取命令	格式： get_gsm_num \$sendid \$password  变量： \$sendid:数字字符计数, server 设定的通信包 id; \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	Goip 收到命令后，验证密码，取得 GSM 号码并发出的回应 成功格式：get_gsm_num \$sendid \$gsmnum 失败格式：ERROR \$sendid \$errmsg 变量：

	\$sendid:计数; \$gsmnum:GSM 号码; \$errmsg:出错信息字符串
--	--

### 5.1.2 设置 GSM 号码

步骤	描述
1.server 发送 设置 GSM 号码的命令	格式: set_gsm_num \$sendid \$gsmnum \$password  变量: \$sendid:数字字符计数, server 设定的通信包 id; \$gsmnum:欲设置的号码 \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	Goip 收到命令后, 验证密码, 设置 GSM 号码并发出的回应 成功格式: set_gsm_num \$sendid \$gsmnum ok 失败格式: ERROR \$sendid \$gsmnum \$errmsg 变量: \$sendid:计数; \$gsmnum:GSM 号码; \$errmsg:出错信息

### 5.1.3 取线路总通话限制时间

步骤	描述
1.server 发送 线路总通话时间获取命令	格式: get_exp_time \$sendid \$password  变量: \$sendid:数字字符计数, server 设定的通信包 id; \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	Goip 收到命令后, 验证密码, 取得线路总通话限制时间并发出的回应 成功格式: get_exp_time \$sendid \$exptime 失败格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$exptime:线路总通话时间 (分钟); \$errmsg:出错信息字符串



### 5.1.5 设置线路总通话限制时间

步骤	描述
1.server 发送 线路总通话 时间设置命 令	格式： set_exp_time \$sendid \$exptime \$password  变量： \$sendid:数字字符计数, server 设定的通信包 id; \$exptime:将设置的时间（分钟） \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	Goip 收到命令后，验证密码，设置线路总通话限制时间并发出的回应 成功格式：set_exp_time \$sendid \$exptime ok 失败格式：ERROR \$sendid \$exptime \$errmsg 变量： \$sendid:计数; \$exptime:线路总通话时间（分钟）; \$errmsg:出错信息字符串

### 5.1.4 取线路剩余通话时间

步骤	描述
1.server 发送 命令,以获取 线路剩余通 话时间	格式： get_remain_time \$sendid \$password  变量： \$sendid:数字字符计数, server 设定的通信包 id; \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	Goip 收到命令后，验证密码，取得线路剩余通话限制时间并发出的回应 成功格式：get_remain_time \$sendid \$remain_time 失败格式：ERROR \$sendid \$errmsg 变量： \$sendid:计数; \$remain_time:线路剩余通话时间（分钟）; \$errmsg:出错信息字符串

### 5.1.6 重置线路剩余通话时间

步骤	描述
1.server 发送 命令,以重置	格式： reset_remain_time \$sendid \$password

线路剩余通话时间	变量: \$sendid:数字字符计数, server 设定的通信包 id; \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	Goip 收到命令后, 验证密码, 重置线路剩余通话限制时间并发出的回应 成功格式: reset_remain_time \$sendid ok 失败格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$errmsg:出错信息字符串

### 5.1.7 获取线路状态

步骤	描述
1.server 发送命令, 获取线路状态	格式: get_gsm_state \$sendid \$password  变量: \$sendid:数字字符计数, server 设定的通信包 id; \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	Goip 收到命令后, 验证密码, 取得线路状态并发出的回应 成功格式: get_gsm_state \$sendid \$state 失败格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$state:线路状态, 字符串(一般是 IDLE 或 ACTIVE) \$errmsg:出错信息字符串

### 5.1.8 终止正进行的通话

步骤	描述
1.server 发送命令, 终止当前正进行着的通话	格式: svr_drop_call \$sendid \$password  变量: \$sendid:数字字符计数, server 设定的通信包 id; \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	Goip 收到命令后, 验证密码, 终止通话并发出的回应 成功格式: svr_drop_call \$sendid \$ok 失败格式: ERROR \$sendid \$errmsg 变量:

	\$sendid:计数; \$errmsg:出错信息字符串
--	----------------------------------

### 5.1.9 重启本线路

步骤	描述
1.server 发送命令,以重启本线路(本模块)	格式: svr_reboot_module \$sendid \$password  变量: \$sendid:数字字符计数, server 设定的通信包 id; \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	Goip 收到命令后, 验证密码, 终止通话并发出的回应 成功格式: svr_reboot_module \$sendid \$ok 失败格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$errmsg:出错信息字符串

### 5.1.10 重启设备

步骤	描述
1.server 发送命令,以重启整个 GOIP 设备	格式: svr_reboot_dev \$sendid \$password  变量: \$sendid:数字字符计数, server 设定的通信包 id; \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	Goip 收到命令后, 验证密码, 终止通话并发出的回应 成功格式: svr_reboot_dev \$sendid \$ok 失败格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$errmsg:出错信息字符串

### 5.1.10 同时获取多个参数

Sever 发出一个命令, Goip 同时返回 GSM 号码, 线路状态, 总限制时间, 剩余限制时间四个数据

步骤	描述
1.server 发送命令, 以读取多个参数	<p>格式:</p> <p>GSM \$sendid \$password</p> <p>变量:</p> <p>\$sendid:数字字符计数, server 设定的通信包 id;</p> <p>\$password:在 goip 配置页面填写的注册密码;</p>
2.Goip 回应	<p>Goip 收到命令后, 验证密码, 读取数据并发出的回应</p> <p>成功格式:</p> <p>GSM:\$sendid;get_gsm_num:</p> <p>\$gsmnum;get_exp_time:\$exptime;get_remain_time:\$remain_time;get_gsm_state:\$state</p> <p>失败格式: GSMERROR \$sendid \$errmsg</p> <p>变量:</p> <p>\$sendid:计数;</p> <p>\$gsmnum: GSM 号码;</p> <p>\$exptime:线路总通话时间 (分钟);</p> <p>\$remain_time:线路剩余通话时间 (分钟);</p> <p>\$state:线路状态, 字符串(一般是 IDLE 或 ACTIVE)</p> <p>\$errmsg:出错信息字符串</p>

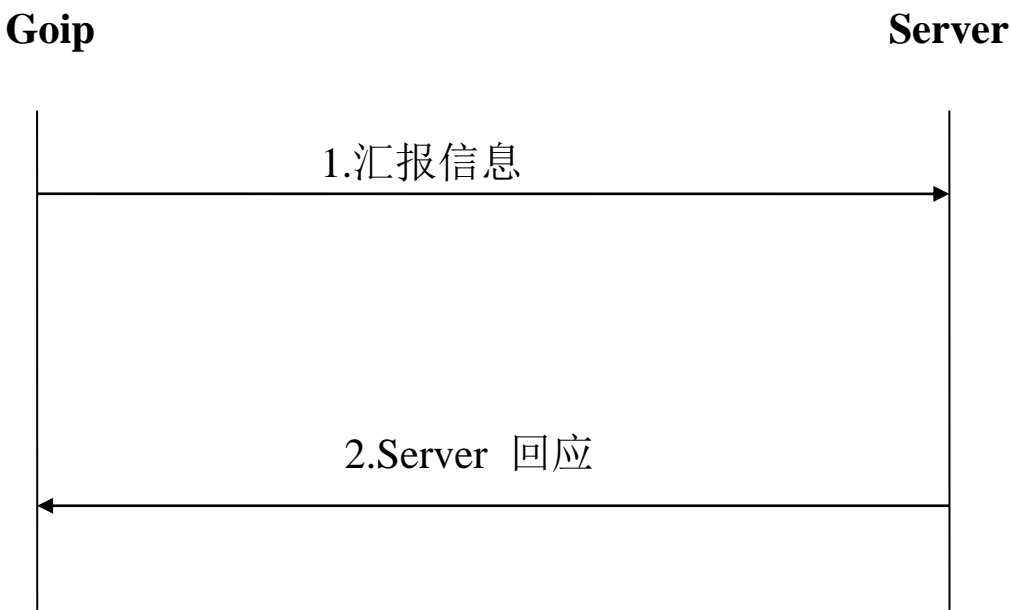
### 5.1.11 设置呼叫转移

步骤	描述
1.server 发送命令, 设置 sim 卡的呼叫转移	<p>格式:</p> <p>CF \$sendid \$password \$reason \$mode \$num \$ftime</p> <p>变量:</p> <p>\$sendid:数字字符计数, server 设定的通信包 id;</p> <p>\$password:在 goip 配置页面填写的注册密码;</p> <p>\$reason:转移的类型, 0 为无条件, 1 为遇忙, 2 为无应答, 3 为不可及, 4 为所有呼叫前传, 5 为所有有条件的呼叫前转;</p> <p>\$mode:启用或禁用该类型的呼叫转移。3 为启用, 4 为禁用。</p> <p>\$num:呼叫转移到的号码</p> <p>\$ftime:无应答转移中的无应答超时时间(秒), 如果非无应答转移请填写 0.</p>
2.Goip 回应	<p>Goip 收到命令后, 验证密码, 设置呼叫转移并发出的回应</p> <p>成功格式:</p> <p>CFOK \$sendid</p> <p>失败格式:</p> <p>CFERROR \$sendid</p>

3. server 最终回应	格式： DONE \$sendid
----------------	----------------------

## 5.2 Goip 主动向 Server 汇报信息。

当 Goip 线路状态改变或进行一个通话时，Goip 会向 server 发送信息。



### 5.2.1 Goip 线路状态改变时向服务器发送当前状态。

步骤	描述
1.Goip 汇报线路当前状态	格式： STATE:\$udpid;id:\$goipid;password:\$password;gsm_remain_state:\$state 变量： \$udpid:数字字符计数, Goip 设定的通信包 id; \$goipid: 在 goip 配置页面填写的注册 id; \$password:在 goip 配置页面填写的注册密码; \$state:线路状态字符串
2.Server 回应	Goip 收到命令后，验证密码，取得 GSM 号码并发出的回应 成功格式：STATE \$udpid OK 失败格式：STATE \$udpid \$errmsg 变量： \$udpid: :数字字符计数, Goip 设定的通信包 id; \$errmsg:出错信息

### 5.2.2 Goip 进行通话时向服务器发送本次通话信息。

步骤	描述
1.Goip 发送 通话信息	格式： RECORD:\$udpid;id:\$goipid;password:\$password;dir:\$dir;num:\$num 变量： \$udpid:数字字符计数, Goip 设定的通信包 id; \$goipid: 在 goip 配置页面填写的注册 id; \$password:在 goip 配置页面填写的注册密码; \$dir:数字字符, 代表通话方向 1 为 INCOMING, 2 为 OUTGOING
2.Server 回 应	Goip 收到命令后, 验证密码, 取得 GSM 号码并发出的回应 成功格式: RECORD \$udpid OK 失败格式: RECORD \$udpid \$errmsg 变量： \$udpid: :数字字符计数, Goip 设定的通信包 id; \$errmsg:出错信息

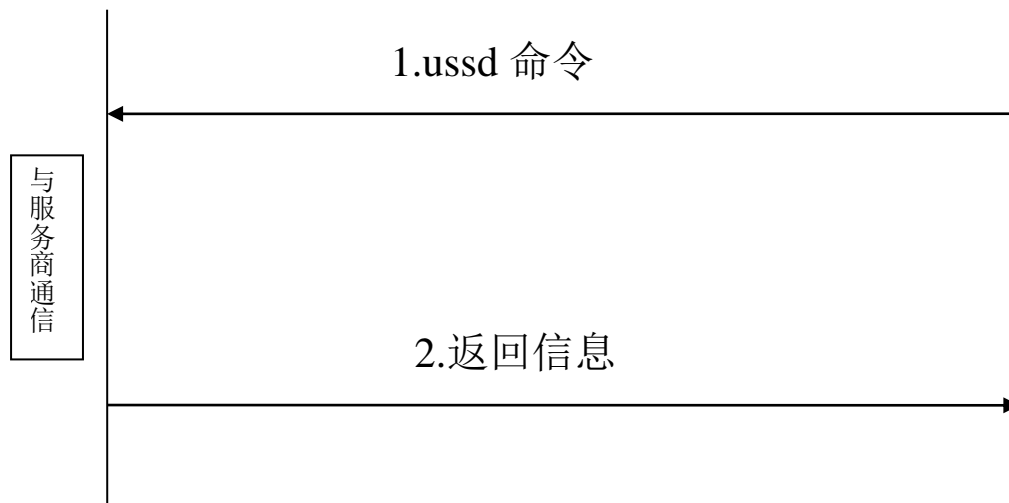
### 5.2.3 每次通话结束后向服务器汇报剩余通话时间。

步骤	描述
1.Goip 汇 报 线 路 当 前 状 态	格式： REMAIN:\$udpid;id:\$goipid;password:\$password;gsm_remain_time:\$time 变量： \$udpid:数字字符计数, Goip 设定的通信包 id; \$goipid: 在 goip 配置页面填写的注册 id; \$password:在 goip 配置页面填写的注册密码; \$time:线路剩余通话时间, 数字字符串, 单位分钟。
2.Server 回 应	Goip 收到命令后, 验证密码, 取得 GSM 号码并发出的回应 成功格式: REMAIN \$udpid OK 失败格式: REMAIN \$udpid \$errmsg 变量： \$udpid: :数字字符计数, Goip 设定的通信包 id; \$errmsg:出错信息

## 6.USSD

**Goip**

**Server**



步骤	描述
1.server 发送 ussd 命令	<p>Server 端建立无应答超时重传， 考虑到 USSD 的传输速度， 这个超时时间应该设置不低于 10 秒。推荐超时时间 10 秒，重发一次。</p> <p>Ussd 普通命令格式：  USSD \$sendid \$password \$ussdcmd  断开命令格式：  USSDEXIT \$sendid \$password  变量：  \$sendid:数字字符计数, server 设定的唯一通信包 id，递增;  \$password:在 goip 配置页面填写的注册密码;  \$ussdcmd:电信服务商可以识别的 ussd 指令</p>
2.Goip 回应	<p>Goip 收到命令后，验证密码，尝试与电信服务商通话并得到回应信息，然后将信息传回 server。Goip 会保存最新的\$sendid 和最新的回应信息，如果收到\$sendid 小于或等于已保存的\$sendid, 将直接传回保存的回应信息。</p> <p>返回信息格式： USSD \$sendid \$msg  出错格式： USSDERROR \$sendid \$errmsg  USSD 已断开： USSDEXIT \$sendid  变量：  \$sendid:计数;  \$msg:显示给客户的信息 utf8 编码字符串  \$errmsg:出错信息字符串</p>

## 7.IMEI

### 7.1 获取 IMEI

步骤	描述
1.server 发送	格式:

获取 IMEI 命令	get_imei \$sendid \$password 变量: \$sendid:数字字符计数, server 设定的唯一通信包 id, 递增; \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	返回信息格式: get_imei \$sendid \$imei 出错格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$imei:IMEI 号 \$errmsg:出错信息字符串

## 7.2 设置 IMEI

步骤	描述
1.server 发送设置 IMEI 命令	格式: set_imei \$sendid \$imei \$password 变量: \$sendid:数字字符计数, server 设定的唯一通信包 id, 递增; \$imei: IMEI 号码, 15 个字节 \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	返回信息格式: set_imei \$sendid \$imei ok 出错格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$imei:IMEI 号 \$errmsg:出错信息字符串

## 8.呼出间隔时间

### 8.1 获取呼出间隔时间

步骤	描述
1.server 发送获取呼出间隔时间命令	格式: get_out_call_interval \$sendid \$password 变量: \$sendid:数字字符计数, server 设定的唯一通信包 id, 递增; \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	返回信息格式: get_out_call_interval \$sendid \$interval 出错格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$interval:呼出间隔时间



	\$errmsg:出错信息字符串
--	------------------

## 8.2 设置呼出间隔时间

步骤	描述
1.server 发送 设置呼出间隔时间命令	格式: set_out_call_interval \$sendid \$interval \$password 变量: \$sendid:数字字符计数, server 设定的唯一通信包 id, 递增; \$interval: 呼出间隔时间 \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	返回信息格式: set_out_call_interval \$sendid \$interval ok 出错格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$interval: 呼出间隔时间 \$errmsg:出错信息字符串

## 9 启用禁用模块

### 9.1 启用禁用本线路的模块

步骤	描述
1.server 发送 启用禁用本模块的命令	格式: module_ctl_i \$sendid \$value \$password 变量: \$sendid:数字字符计数, server 设定的唯一通信包 id, 递增; \$value: 1 表示启用, 0 表示禁用 \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	Goip 收到命令后, 验证密码, 终止通话并发出的回应 成功格式: module_ctl_i \$sendid \$ok 失败格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$errmsg:出错信息字符串

### 9.2 启用禁用全部线路的模块

步骤	描述
1.server 发送 启用禁用所有模块的命令	格式: module_ctl \$sendid \$value \$password 变量: \$sendid:数字字符计数, server 设定的唯一通信包 id, 递增; \$value: 1 表示启用, 0 表示禁用, 2 表示保持原样, 每路一个数字, 比如 10120121, 表示第一路启用, 第二路禁用, 第三路启用, 第

	四路不变，第五路禁用，第六路启用，第七路不变，第八路启用。 \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	Goip 收到命令后，验证密码，终止通话并发出的回应 成功格式: module_ctl \$sendid \$ok 失败格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$errmsg:出错信息字符串

## 10.基站

### 10.1 发送基站列表

步骤	描述
1.goip 主动发送基站列表命令给服务器	格式: CELLS:\$recv_id;\$id;\$password;\$cell_list 变量: \$recv_id:计数,GOIP 以当前时间戳递减计数;\$password:在 goip 配置页面填写的注册密码; \$id:在服务器上设置的 goip 的 id \$password:在服务器上设置的 goip 的密码 \$cell_list:基站列表，每个基站一个数字字符串表示，用','号隔开，比如 123,456,789,10,
2.服务器回应	Sms server 收到信息后，查找 id 列表中的 id 和密码，并发出的回应 成功格式: CELLS \$recv_id OK\n 失败格式: CELLS \$recv_id ERROR \$errmsg 变量: \$recv_id:计数; \$errmsg:出错信息

### 10.2 设置基站

步骤	描述
1.server 发送设置基站命令	格式: set_base_cell \$sendid \$cell_id \$password 变量: \$sendid:数字字符计数, server 设定的唯一通信包 id，递增; \$cell_id: 基站 id \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	返回信息格式: set_base_cell \$sendid \$cell_id ok 出错格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$cell_id: 基站 id \$errmsg:出错信息字符串

### 10.3 获取基站列表

步骤	描述
1.server 发送获取基站列表命令	格式: get_cells_list \$sendid \$password 变量: \$sendid:数字字符计数, server 设定的唯一通信包 id, 递增; \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	GoIP 获取命令后, 返回 OK 信息, 并且向运营商发出获取基站列表的指令, 稍后运营商返回后, GoIP 向服务器发送 10.1 中的指令。 返回信息格式: get_cells_list \$sendid ok 出错格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$errmsg:出错信息字符串

### 10.4 获取当前基站

步骤	描述
1.server 发送获取当前基站的命令	格式: CURCELL \$sendid \$password 变量: \$sendid:数字字符计数, server 设定的唯一通信包 id, 递增; \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	返回信息格式: CURCELL \$sendid \$curcellid 出错格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$curcellid: 当前使用的基站 id \$errmsg:出错信息字符串

### 10.5 发送当前基站信息

步骤	描述
1.goip 注册上基站后, 主动发送基站信息给服务器	格式: CELLINFO:\$recvid;id:\$id;password:\$password;info:\$cell_info 变量: \$recvid:计数,GOIP 以当前时间戳递减计数;\$password:在 goip 配置页面填写的注册密码; \$id:在服务器上设置的 goip 的 id \$password:在服务器上设置的 goip 的密码 \$cell_info:基站信息, 字符串

2.服务器回应	<p>Sms server 收到信息后，查找 id 列表中的 id 和密码，并发出的回应</p> <p>成功格式：CELLINFO \$recvvid OK\n</p> <p>失败格式：CELLINFO \$recvvid ERROR \$errmsg</p> <p>变量：\$recvvid:计数; \$errmsg:出错信息</p>
---------	--

## 10.6 发送附近基站（neighbor cells）信息

步骤	描述
1.goip 注册上基站后,当不在通话中时,几分钟检查一下附近基站,发送信息给服务器	<p>格式:</p> <p>BCCH:\$recvvid;id:\$id;password:\$password;bcch:\$bcch_info</p> <p>变量:</p> <p>\$recvvid:计数,GOIP 以当前时间戳递减计数;\$password:在 goip 配置页面填写的注册密码;</p> <p>\$id:在服务器上设置的 goip 的 id</p> <p>\$password:在服务器上设置的 goip 的密码</p> <p>\$bcch_info:附近基站信息，字符串</p>
2.服务器回应	<p>Sms server 收到信息后，查找 id 列表中的 id 和密码，并发出的回应</p> <p>成功格式：BCCH \$recvvid OK\n</p> <p>失败格式：BCCH \$recvvid ERROR \$errmsg</p> <p>变量：\$recvvid:计数; \$errmsg:出错信息</p>

## 11.AT 指令交互

Goip 和 SMS Server 可以交互 AT 指令。

### 11.1 开启 AT 指令交互

步骤	描述
1.server 发送 AT 指令交互的启动命令	<p>格式:</p> <p>ATSTART \$sendid \$password</p> <p>变量:</p> <p>\$sendid:数字字符计数, server 设定的唯一通信包 id</p> <p>\$password:在 goip 配置页面填写的注册密码;</p>
2.Goip 回应	<p>GoIP 收到此指令后，将打开和 SMS Server 的 AT 指令交互，随后将把 AT 指令的所有读写信息发送给 SMS Server,发送格式见 11.2</p> <p>返回信息格式：ATSTART \$sendid OK</p> <p>出错格式：ERROR \$sendid \$errmsg</p> <p>变量:</p> <p>\$sendid:计数;</p> <p>\$errmsg:出错信息字符串</p>

### 11.2 GoIP 发送 AT 指令的读写信息给 Server

步骤	描述
1.goip 主动发送 AT 指令的读写信息服务器	<p>格式： AT:\$sendid;count:\$count;id:\$id;password:\$password;receive:\$at_msg</p> <p>变量： \$sendid:通讯包 id, 与 11.1 中收到的一致; \$count:计数, 从打开 at 指令交互开始, 从 1 开始计数, GoIP 每发一个包, 计数加 1; \$id:在服务器上设置的 goip 的 id \$password:在 goip 配置页面填写的注册密码; \$at_msg:AT 指令的读写信息, 以 write:开头的为写的信息, 其余为读出的信息。</p>
2.服务器回应	无需回应

### 11.3 关闭 AT 指令交互

步骤	描述
1.server 发送 AT 指令交互的关闭命令	<p>格式： ATSTOP \$sendid \$password</p> <p>变量： \$sendid:数字字符计数, server 设定的唯一通信包 id \$password:在 goip 配置页面填写的注册密码;</p>
2.Goip 回应	<p>GoIP 收到此指令后, 将关闭和 SMS Server 的 AT 指令交互。</p> <p>返回信息格式: ATSTOP \$sendid OK 出错格式: ERROR \$sendid \$errmsg</p> <p>变量： \$sendid:计数; \$errmsg:出错信息字符串</p>

### 11.4 保持 AT 指令交互

步骤	描述
1.server 发送 AT 指令交互的保持命令	<p>Sever 必须每隔 30 秒发送该命令给 GoIP, 以保持 AT 指令的交互状态。如果 GoIP 超过 60 秒没有收到该命令, 将关闭 AT 指令交互。</p> <p>格式： ATALIVE \$sendid \$password</p> <p>变量： \$sendid:数字字符计数, server 设定的唯一通信包 id \$password:在 goip 配置页面填写的注册密码;</p>
2.Goip 回应	<p>返回信息格式: ATALIVE \$sendid OK 出错格式: ERROR \$sendid \$errmsg</p> <p>变量： \$sendid:计数; \$errmsg:出错信息字符串</p>

### 11.5 Server 发送 AT 指令给 GoIP

步骤	描述
1.server 发送 AT 指令给 GoIP	格式: ATCMD \$sendid \$atcmd \$password 变量: \$sendid:数字字符计数, server 设定的唯一通信包 id \$atcmd: 要写入的 AT 指令 \$password:在 goip 配置页面填写的注册密码;
2.Goip 回应	GoIP 收到该指令后, 将会向模块写入该 AT 指令 返回信息格式: ATCMD \$sendid OK 出错格式: ERROR \$sendid \$errmsg 变量: \$sendid:计数; \$errmsg:出错信息字符串

## 12.发送 AT+CGATT 状态

### 12.1 发送 AT+CGATT 状态

步骤	描述
1.当 goip 注册上 GSM 时,主动发送 CGATT 状态给服务器	格式: CGATT:\$recv;id:\$id;password:\$password;cgatt:\$cgatt_state 变量: \$recv:计数,GOIP 以当前时间戳递减计数;\$password:在 goip 配置页面填写的注册密码; \$id:在服务器上设置的 goip 的 id \$password:在服务器上设置的 goip 的密码 \$cgatt_state:cgatt 状态, 0 或 1
2.服务器回应	Sms server 收到信息后, 查找 id 列表中的 id 和密码, 并发出的回应 成功格式: CGATT \$recv OK 失败格式: CGATT \$recv ERROR \$errmsg 变量: \$recv:计数; \$errmsg:出错信息

## 13.发送 hangup cause

### 13.1 发送 hangup cause

步骤	描述
1. 当 呼 叫 hangup 时, 发 送 hangup	格式: HANGUP:\$recv;id:\$id;password:\$password;num:\$num,cause:\$cause 变量: \$recv:计数,GOIP 以当前时间戳递减计数;\$password:在 goip 配置

cause 给服务器	页面填写的注册密码; \$id:在服务器上设置的 goip 的 id \$password:在服务器上设置的 goip 的密码 \$num:对方号码 \$cause: 字符串 User busy, Out of service, Normal call clearing 等其中之一
2. 服务器回应	Sms server 收到信息后, 查找 id 列表中的 id 和密码, 并发出的回应 成功格式: HANGUP \$recvid OK\n 失败格式: HANGUP \$recvid ERROR \$errmsg 变量: \$recvid:计数; \$errmsg:出错信息