oSIP 协议栈浅析

1. oSIP介绍

oSIP 是按照 RFC3261(SIP)和 RFC2327(SDP)标准，并使用标准 c 编写的一个 SIP 协议栈。它是一个公开源码的免费协议栈。oSIP 协议栈结构简单而小巧，它并不提供高层的 SIP 会话控制的 API，它主要提供一些解析 SIP/SDP 消息的 API 和事务处理的状态机。

oSIP 支持线程安全，既可以用于多线程的编程模式，也可以用于单线程的编程模式；oSIP 可以用来开发 User Agent，IP soft-phone 和 SIP Proxy 等等。

oSIP 目前最后版本为 V 0.9.7，不久 oSIP 版本将升级至 oSIP2（V 1.99.7）。oSIP2 主要调整了一些函数和结构名称，以及一些头文件的名称、内容结构的调整，整体的构架和功能不变。

本文以下描述都基于 oSIP V0.9.6 版本。

2. oSIP结构分析

2.1 oSIP 结构

oSIP 主要包括三大部分的内容：状态机模块、解析器模块和工具模块。

状态机模块的功能：

完成对某个事务（注册过程，呼叫过程等等）状态记录，并在特定状态下触发相应的事件或回调函数。

解析器模块的功能：

该模块主要完成对 SIP 消息结构剖析、SDP 消息的结构剖析以及 URI 结构的剖析；

工具模块的功能：

该模块提供一些 SDP 等处理的一些工具。

oSIP 的模块结构图如下(图 2-1)：

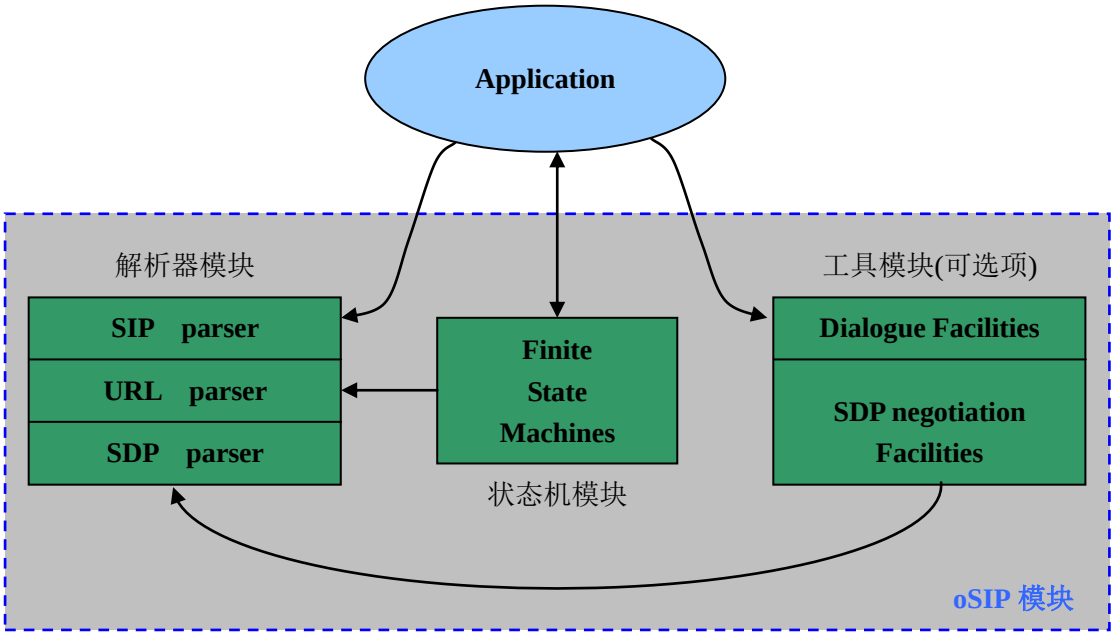

图 2-1 oSIP 结构

2.2 状态机(Finite State Machines)模块

2.2.1 概述

oSIP 状态机(Finite State Machines)主要分为四类，分别为：

- ICT -- Invite Client (outgoing) Transaction
- NICT -- Non-Invite Client (outgoing) Transaction
- IST -- Invite Server (incoming) Transaction
- NIST -- Non-Invite Server (incoming) Transaction

2.2.2 ICT 状态机

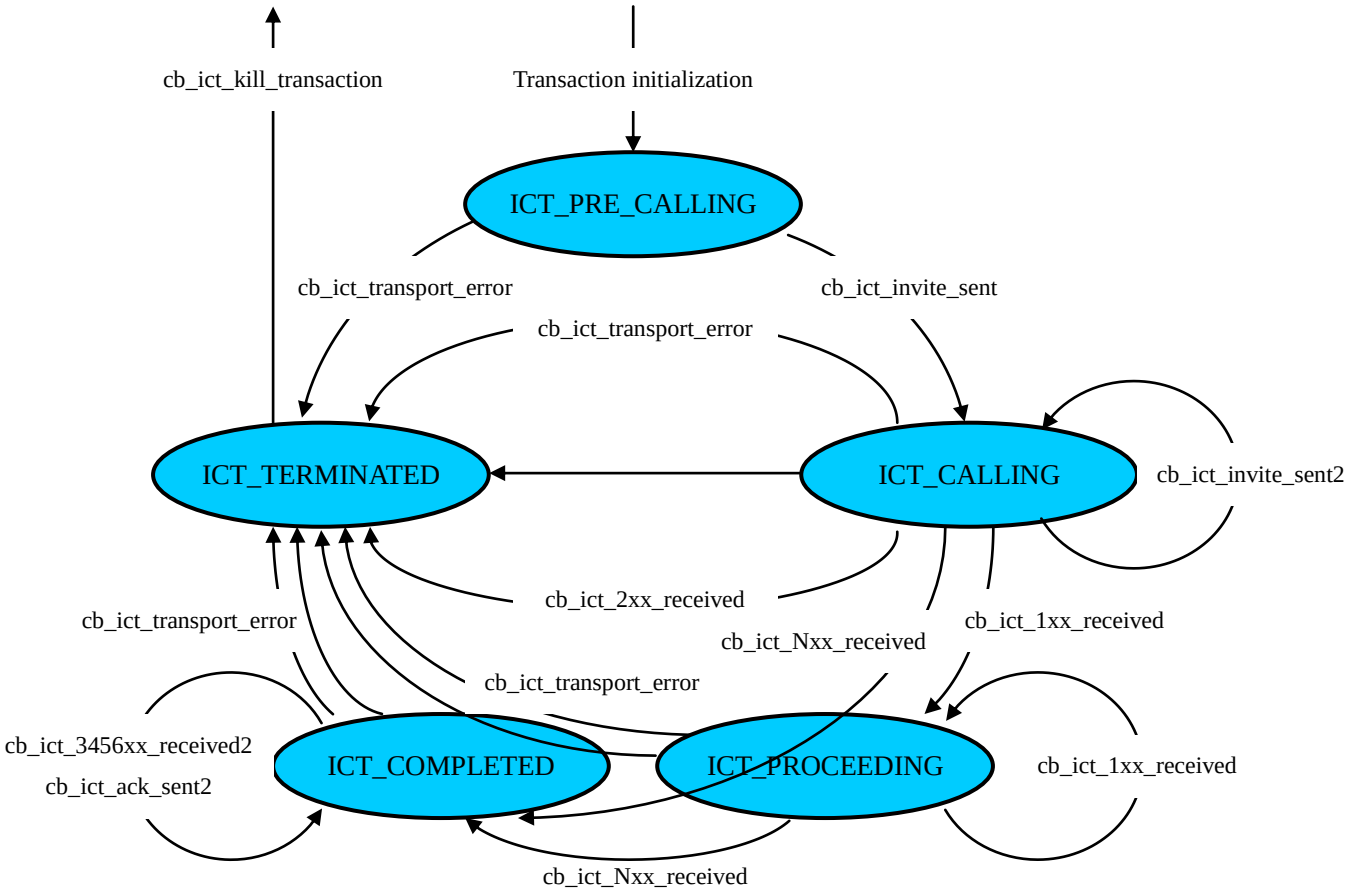

图 2-2: ICT State Machine

注：

—— cb_ict_xxx_xxxx ⟶ 表示从一种状态转换到另一种状态时将调用该回调函数

⟶ 表示从一种状态转换到另一种状态时不调用任何回调函数

cb_ict_Nxx_received：其中 N 表示一下几个值

3 -- cb_ict_3xx_received
4 -- cb_ict_4xx_received
5 -- cb_ict_5xx_received
6 -- cb_ict_6xx_received

2.2.3 NICT 状态机

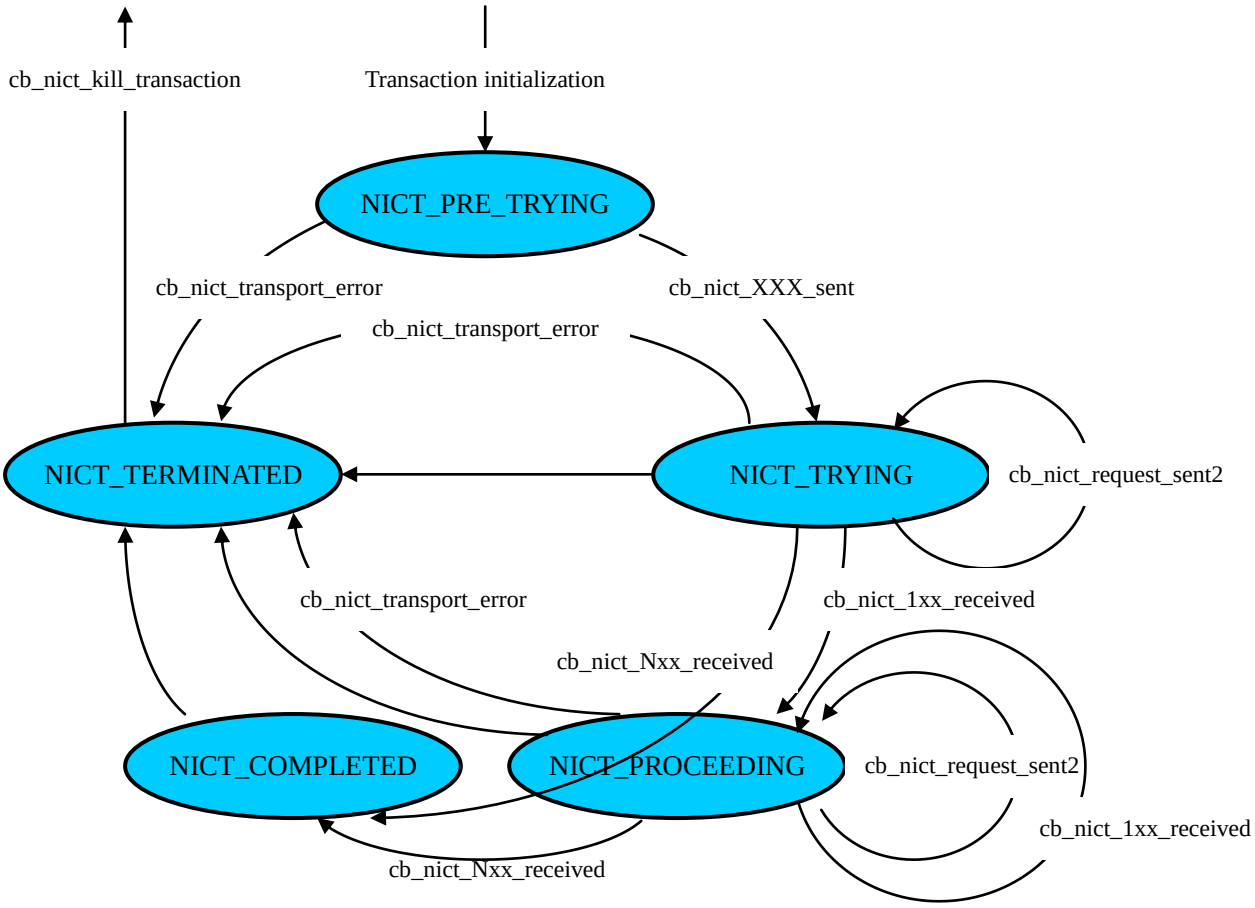

图 2-3: NICT State Machine

注：

—— cb_nict_xxx_xxxx ——→ 表示从一种状态转换到另一种状态时将调用该回调函数

——————————→ 表示从一种状态转换到另一种状态时不调用任何回调函数

cb_nict_XXX_sent：其中 XXX 表示一下几种消息类型，

register	--	cb_nict_register_sent
bye	--	cb_nict_bye_sent
options	--	cb_nict_options_sent
info	--	cb_nict_info_sent
cancel	--	cb_nict_cancel_sent
notify	--	cb_nict_notify_sent
subscribe	--	cb_nict_subscribe_sent
unknown	--	cb_nict_unknown_sent

cb_nict_Nxx_received：其中 N 表示一下几个值

2	--	cb_nict_2xx_received

3 -- cb_nict_3xx_received
4 -- cb_nict_4xx_received
5 -- cb_nict_5xx_received
6 -- cb_nict_6xx_received

2.2.4 IST 状态机

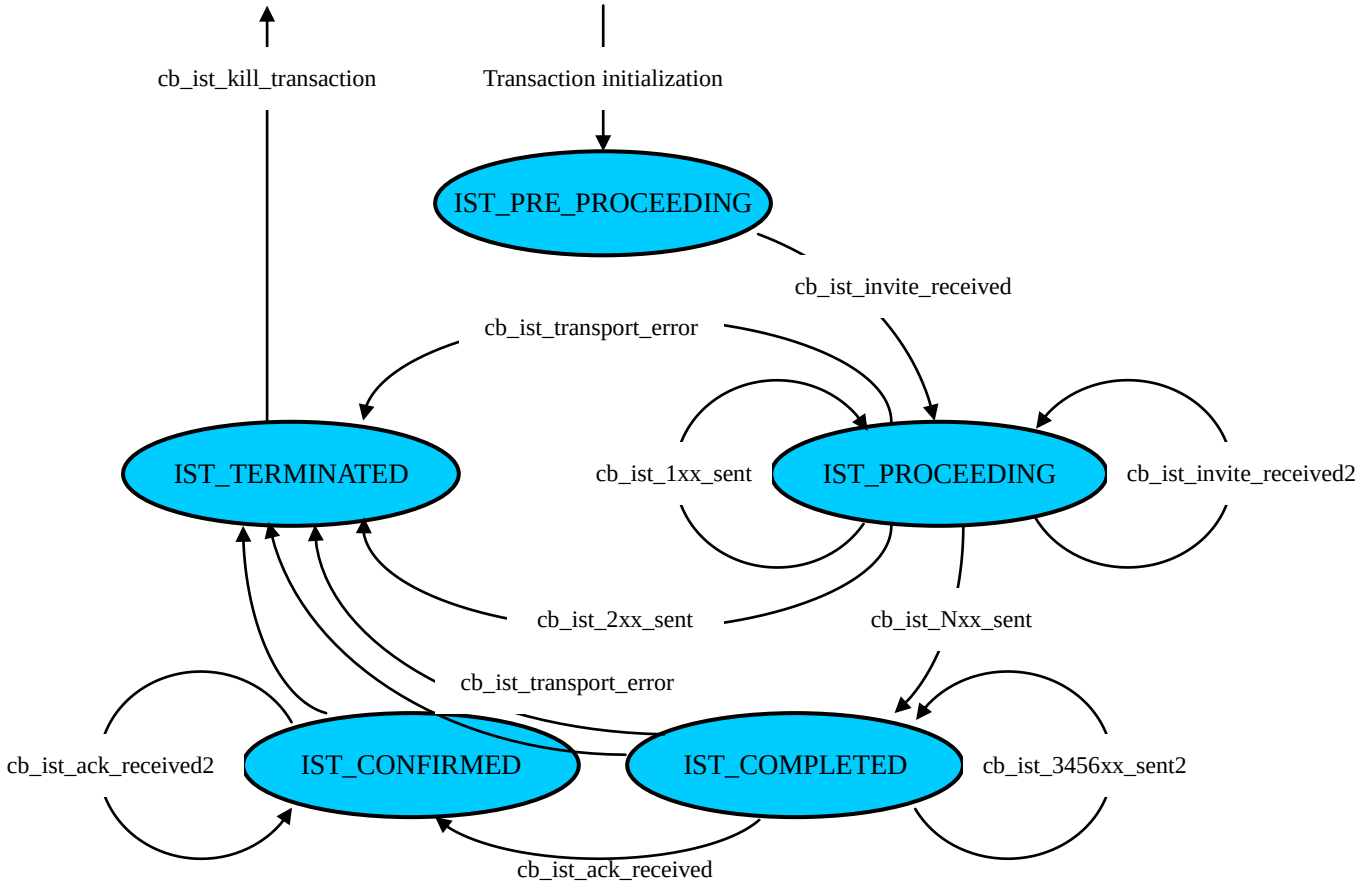

图 2-4: IST State Machine

注：

—— cb_ist_xxx_xxxx ——→ 表示从一种状态转换到另一种状态时将调用该回调函数

——————→ 表示从一种状态转换到另一种状态时不调用任何回调函数

cb_ist_Nxx_sent：其中 N 表示一下几个值，

3 -- cb_ist_3xx_sent
4 -- cb_ist_4xx_sent
5 -- cb_ist_5xx_sent
6 -- cb_ist_6xx_sent

2.2.5 NIST 状态机

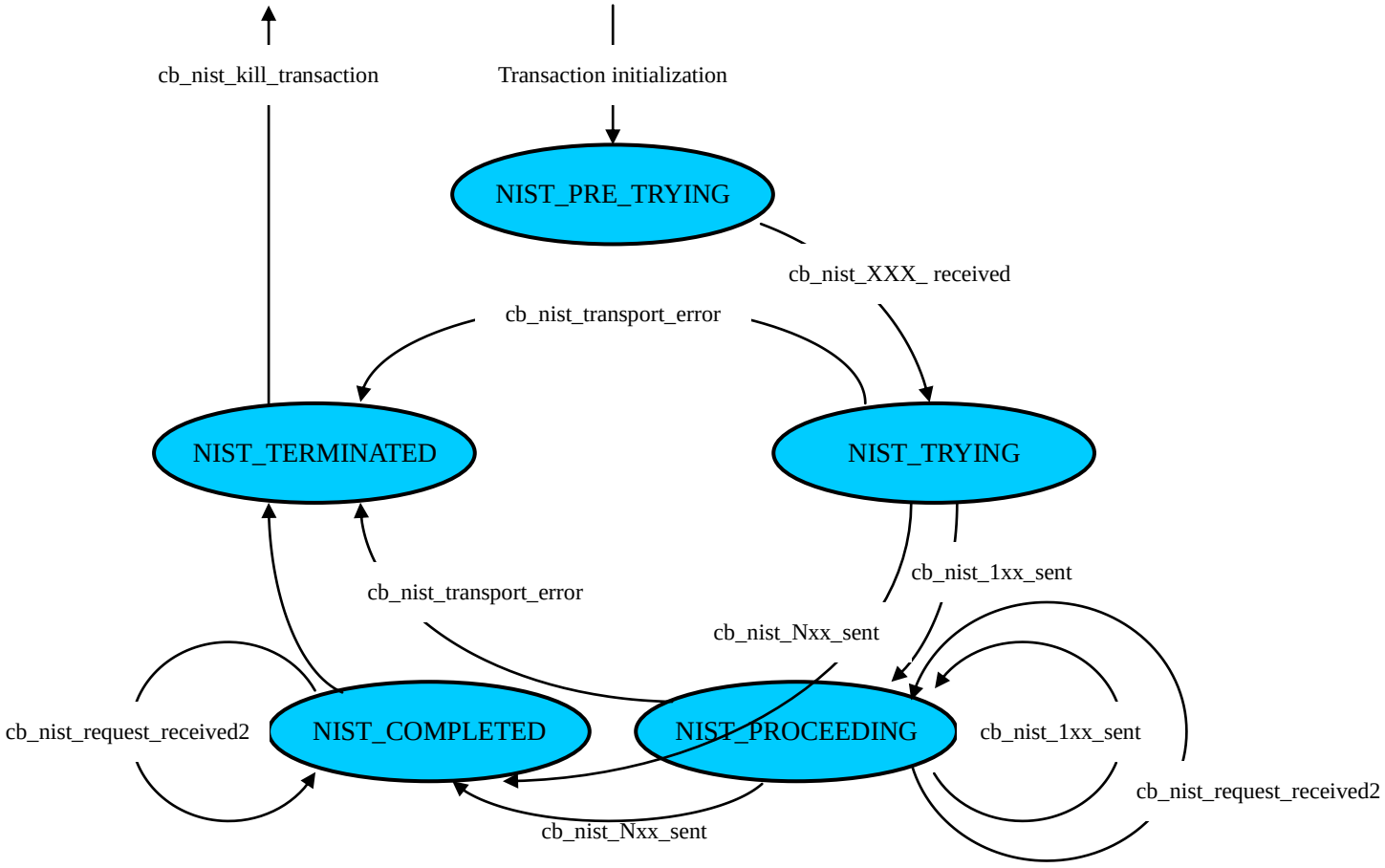

图 2-5: NIST State Machine

注：

—— cb_nist_xxx_xxxx ——▶ 表示从一种状态转换到另一种状态时将调用该回调函数

——————▶ 表示从一种状态转换到另一种状态时不调用任何回调函数

cb_nist_XXX_ received：其中 XXX 表示一下几种消息类型，

register	--	cb_nist_register_received
bye	--	cb_nist_bye_received
options	--	cb_nist_options_received
info	--	cb_nist_info_received
cancel	--	cb_nist_cancel_received
notify	--	cb_nist_notify_received
subscribe	--	cb_nist_subscribe_received
unknown	--	cb_nist_unknown_received

cb_nist_Nxx_ sent：其中 N 表示一下几个值

2 -- cb_nist_2xx_sent
3 -- cb_nist_3xx_sent
4 -- cb_nist_4xx_sent
5 -- cb_nist_5xx_sent
6 -- cb_nist_6xx_sent

2.3 解析器(Parsers)模块

2.3.1 SIP Parser

oSIP 的 SIP Parser 处理的 SIP 头域(SIP Header fields)及其相应的操作功能列表如下：

SIP Header(头域)	Functions(函数名称—简写)	
Accept	set(),get()	☆
Accept-Encoding	set(),get(),init(),parse(),2char(),free(),clone() Getelement(),setelement()	☆
Accept-Language	set(),get()	☆
Alert-Info	set(),get()	☆
Allow	set(),get()	☆
Authentication-Info		
Authorization	Init(),set(),parse(),get(),getauth_type(),setauth_type(), Getusername(),setusername(),getrealm(),setrealm(), Getnonce(),setnonce(),geturi(),seturi(),getresponse(), Setresponse(),getdigest(),setdigest(),getalgorithm(), Setalgorithm(),getcnonce(),setcnonce(),getopaque(), Setopaque(),getmessage_qop(),setmessage_qop(), getnonce_count(),setnonce_count(),2char(),free(), clone()	☆
Call-ID	set(),get(),parse(),2char(),free(),clone(),getnumber(), setnumber(),gethost(),sethost()	☆
Call-Info	set(),get(),init(),parse(),2char(),free(),clone(), geturi(),seturi()	☆
Contact	set(),get(),init(),parse(),2char(),free(),clone()	☆
Content-Disposition	set(),get(),parse()	☆
Content-Encoding	set(),get()	☆
Content-Language		
Content-Length	set(),get(),init(),parse(),2char(),free(),clone()	☆
Content-Type	set(),get(),init(),parse(),2char(),free(),clone()	☆
CSeq	set(),get(),init(),parse(),2char(),free(),clone(), getnumber(),setnumber(),getmethod(),setmethod()	☆
Date		
Error-Info	set(),get()	☆
Expires		
From	set(),get(),init(),parse(),2char(),free(),clone(),	☆

	getdisplayname(),setdisplayname(),geturl(),seturl(), param_get(),param_parseall(),param_setvalue(), param_getvalue(),param_getname(),param_setname(), compare()	
In-Reply-To		
Max-Forwards		
Min-Expires		
MIME-Version	set(),get()	☆
Organization		
Priority		
Proxy-Authenticate	set(),get()	☆
Proxy-Authorization	set(),get()	☆
Proxy-Require		
Record-Route	set(),get(),init(),parse(),2char(),free()	☆
Reply-To		
Require		
Retry-After		
Route	set(),get(),init(),parse(),2char(),free()	☆
Server		
Subject		
Supported		
Timestamp		
To	set(),get(),init(),parse(),2char(),free(),clone()	☆
Unsupported		
User-Agent		
Via	set(),append(),get(),init(),free(),parse(),2char(), setversion(),getversion(),setprotocol(),getprotocol(), sethost(),gethost(),setport(),getport(),setcomment(), getcomment(),clone()	☆
Warning		
WWW-Authenticate	Init(),set(),quoted_string_set(),token_set(),parse(), get(),getauth_type(),setauth_type(),getrealm(),setrealm(), getdomain(),setdomain(),getnonce(),setnonce(),getstale(), setstale(),getopaque(),setopaque(),getalgorithm(), setalgorithm(),getqop_options(),setqop_options(),2char(), free(),clone()	☆

注：标示“☆”表示 oSIP 支持该头域(SIP Header fields)解析处理，未注的表示目前还没有解析处理（这些被保存在字符串中，可自行处理分析），可能会在后续版本中逐步补充。

2.3.2 SDP Parser

SDP 的格式一般为：

<type>=<value>

type 通常为一个英文字母，其取值如下（按照 RFC2327,带“*”的表示为可选条目）：

Session description

v= (protocol version)
o= (owner/creator and session identifier).
s= (session name)
i=* (session information)
u=* (URI of description)
e=* (email address)
p=* (phone number)
c=* (connection information - not required if included in all media)
b=* (bandwidth information)
One or more time descriptions (see below)
z=* (time zone adjustments)
k=* (encryption key)
a=* (zero or more session attribute lines)
Zero or more media descriptions (see below)

Time description

t= (time the session is active)
r=* (zero or more repeat times)

Media description

m= (media name and transport address)
i=* (media title)
c=* (connection information - optional if included at session-level)
b=* (bandwidth information)
k=* (encryption key)
a=* (zero or more media attribute lines)

在 oSIP 中处理的 type 和相应操作功能如下：

type(类型)	Functions(函数名称—简写)
v	version_set(),version_get()
o	origin_set(),username_get(),sess_id_get(), sess_version_get(),nettype_get(),addrtype_get(), addr_get()
s	name_set(),name_get()
i	info_set(),info_get()
u	uri_set(),uri_get()
e	email_add(),email_get()
p	phone_add(),phone_get()

c	connection_add(),connection_get(),nettype_get(), addrtype_get(),addr_get(),addr_multicast_ttl_get(), addr_multicast_int_get()
b	bandwidth_add(),bwtype_get(),bandwidth_get()
t	time_descr_add(),start_time_get(),stop_time_get()
r	repeat_add(),repeat_get()
z	adjustments_set(),adjustments_get()
k	key_set(),keytype_get(),keydata_get()
a	attribute_add(),att_field_get(),att_value_get()
m	media_add(),media_get(),port_get(),number_of_port_get(), proto_get(),payload_add(),payload_get(),

另外,oSIP 还包含对 SDP 包的一些基本操作[set(), get(), init(), parse(), 2char(), free(), clone()],及对各类 type 的 init()和 free()操作

2.3.3 URL Parser

这里的 URL 是指 SIP 中的 URI，URI 有很多参数格式，在 RFC3261 中列举了一些比较例子：

The URIs within each of the following sets are equivalent:

sip:alice@atlanta.com;transport=TCP
sip:alice@AtLanTa.CoM;Transport=tcp

sip:carol@chicago.com
sip:carol@chicago.com;newparam=5
sip:carol@chicago.com;security=on

sip:biloxi.com;transport=tcp;method=REGISTER?to=sip:bob@biloxi.com
sip:biloxi.com;method=REGISTER;transport=tcp?to=sip:bob@biloxi.com

sip:alice@atlanta.com?subject=project x&priority=urgent
sip:alice@atlanta.com?priority=urgent&subject=project x

The URIs within each of the following sets are not equivalent:

SIP:ALICE@AtLanTa.CoM;Transport=udp (different usernames)
sip:alice@AtLanTa.CoM;Transport=UDP

sip:bob@biloxi.com (can resolve to different ports)
sip:bob@biloxi.com:5060

sip:bob@biloxi.com (can resolve to different transports)
sip:bob@biloxi.com;transport=udp

sip:bob@biloxi.com (can resolve to different port and transports)
sip:bob@biloxi.com:6000;transport=tcp

sip:carol@chicago.com (different header component)
sip:carol@chicago.com?Subject=next%20meeting

sip:bob@phone21.boxesbybob.com (even though that's what
sip:bob@192.0.2.4 phone21.boxesbybob.com resolves to)

在 oSIP 中处理 SIP URI 有比较多的操作函数提供，主要有对 host,port,username,password, scheme 的 get()和 set()，以及对参数的初始化设置和剖析处理。详细函数名称请参考源代码中的 url.h。

2.4 工具(Facilities)模块

2.4.1 SDP negotiator

SDP 协商工具(SDP negotiator)帮助 end point 提供协商 codec 等功能

2.4.2 Dialog management

对话管理工具(Dialog management)是 oSIP 提供的一个比较强大的辅助工具，主要用于有能力应答呼叫的 end point。

对话管理工具(Dialog management)能够帮助记录请求和响应消息，利用这个工具使 end point 能够快速准确的作出应答。

3. oSIP特点

3.1 oSIP 的优点

- Osip 没有给开发者限定在特定的某个执行模式下，能够使开发者选定一个比较适合自己的模式。
- Osip 的各个模块是相对清晰、独立的，因而去掉某个模块时也比较容易。

- Osip 的解析器提供了较为完善的 API，包含了消息的构造、修改和产生等。

3.2 oSIP 的缺点

- oSIP 目前版本源代码结构、定义比较混乱，并且缺乏文档，阅读比较困难；该问题将在 oSIP2 中得到改善。
- oSIP 不提供任何快速产生请求消息和响应消息的方法，所有请求消息和响应消息的形成必须调用一组 sip message api 来手动组装完成，关于这方面的缺陷，osip 作者可能在以后会开发一个 eXoSIP 的 API 来完成。
- 由于 oSIP 结构简单，外围相关模块需要用户自己开发，如 SIP 消息的接收和发送，RTP/RTCP 的语音数据的处理等。

4. oSIP应用结构图

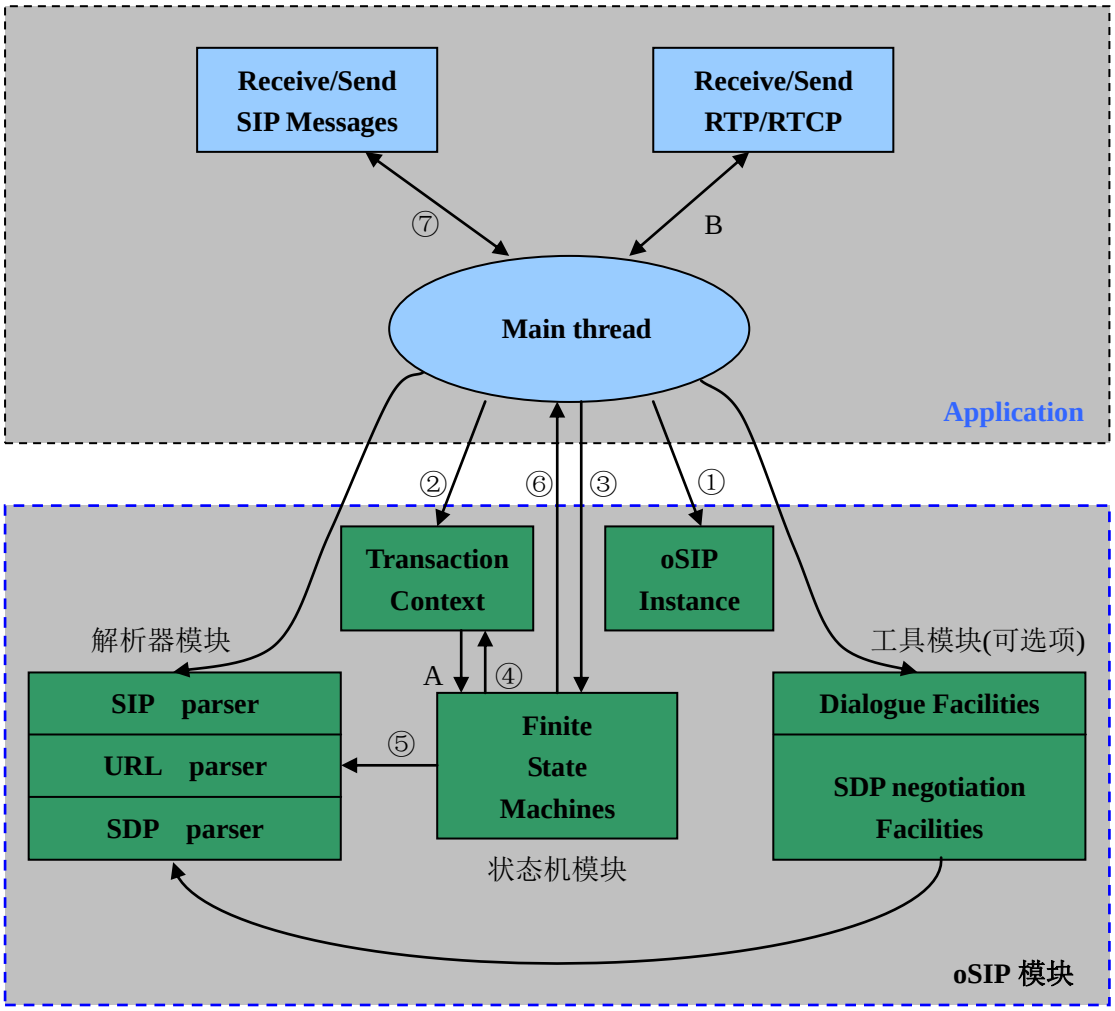

图 4-1 oSIP 应用

其中：

①：初始化 oSIP 和注册 CALL BACK 函数；

②：添加事件 A;

③：执行事务

④：取消事件 A

⑤：解析消息

⑥：触发 CALL BACK 函数

⑦：接收/发送消息

A：保存状态

B：接收/发送语音包

5. oSIP使用概述

5.1 初始化 oSIP

在使用 oSIP 前必须先初始化 oSIP，主要调用函数 osip_global_init()和 osip_init()，具体操作代码如下：

```
osip_t *osip;
  // initialise internal element first
  if (0!=osip_global_init())
     return -1;
  // allocate a global osip element.
  if (0!=osip_init(&osip))
     return -1;
```

5.2 注册 CALL BACK 函数

需要注册的 call back 函数主要包含发送消息、结束事务、发送失败、4 个状态机(ICT、NICT、IST、NIST)相关函数。

注册发送消息的 CALL BACK 函数：

```
osip_setcb_send_message(osip, &application_cb_snd_message);
```

注册结束事务的 CALL BACK 函数：

```
osip_setcb_ict_kill_transaction(osip,&application_cb_ict_kill_transaction);
osip_setcb_ist_kill_transaction(osip,&application_cb_ist_kill_transaction);
osip_setcb_nict_kill_transaction(osip,&application_cb_nict_kill_transaction);
osip_setcb_nist_kill_transaction(osip,&application_cb_nist_kill_transaction);
```

注册发送失败的 CALL BACK 函数：

```
osip_setcb_ict_transport_error(osip,&application_cb_transport_error);
osip_setcb_ist_transport_error(osip,&application_cb_transport_error);
osip_setcb_nict_transport_error(osip,&application_cb_transport_error);
osip_setcb_nist_transport_error(osip,&application_cb_transport_error);
```

注册 ICT、NICT、IST、NIST CALL BACK 函数

```
osip_setcb_ict_2xx_received2(osip,&application_cb_rcvresp_retransmission);
osip_setcb_ict_3456xx_received2(osip,&application_cb_rcvresp_retransmission);
osip_setcb_ict_invite_sent2(osip,&application_cb_sndreq_retransmission);
osip_setcb_ist_2xx_sent2(osip,&application_cb_sndresp_retransmission);
osip_setcb_ist_3456xx_sent2(osip,&application_cb_sndresp_retransmission);
osip_setcb_ist_invite_received2(osip,&application_cb_rcvreq_retransmission);
osip_setcb_nict_2xx_received2(osip,&application_cb_rcvresp_retransmission);
osip_setcb_nict_3456xx_received2(osip,&application_cb_rcvresp_retransmission);
osip_setcb_nict_request_sent2(osip,&application_cb_sndreq_retransmission);
osip_setcb_nist_2xx_sent2(osip,&application_cb_sndresp_retransmission);
osip_setcb_nist_3456xx_sent2(osip,&application_cb_sndresp_retransmission);
osip_setcb_nist_request_received2(osip,&application_cb_rcvreq_retransmission);
osip_setcb_ict_invite_sent    (osip,&application_cb_sndinvite);
osip_setcb_ict_ack_sent       (osip,&application_cb_sndack);
osip_setcb_nict_register_sent(osip,&application_cb_sndregister);
osip_setcb_nict_bye_sent       (osip,&application_cb_sndbye);
osip_setcb_nict_cancel_sent    (osip,&application_cb_sndcancel);
osip_setcb_nict_info_sent      (osip,&application_cb_sndinfo);
osip_setcb_nict_options_sent (osip,&application_cb_sndoptions);
osip_setcb_nict_subscribe_sent (osip,&application_cb_sndoptions);
osip_setcb_nict_notify_sent (osip,&application_cb_sndoptions);
osip_setcb_nict_unknown_sent(osip,&application_cb_sndunkrequest);
osip_setcb_ict_1xx_received(osip,&application_cb_rcv1xx);
osip_setcb_ict_2xx_received(osip,&application_cb_rcv2xx);
osip_setcb_ict_3xx_received(osip,&application_cb_rcv3xx);
osip_setcb_ict_4xx_received(osip,&application_cb_rcv4xx);
osip_setcb_ict_5xx_received(osip,&application_cb_rcv5xx);
osip_setcb_ict_6xx_received(osip,&application_cb_rcv6xx);
osip_setcb_ist_1xx_sent(osip,&application_cb_snd1xx);
osip_setcb_ist_2xx_sent(osip,&application_cb_snd2xx);
osip_setcb_ist_3xx_sent(osip,&application_cb_snd3xx);
osip_setcb_ist_4xx_sent(osip,&application_cb_snd4xx);
osip_setcb_ist_5xx_sent(osip,&application_cb_snd5xx);
```

```
osip_setcb_ist_6xx_sent(osip,&application_cb_snd6xx);
osip_setcb_nict_1xx_received(osip,&application_cb_rcv1xx);
osip_setcb_nict_2xx_received(osip,&application_cb_rcv2xx);
osip_setcb_nict_3xx_received(osip,&application_cb_rcv3xx);
osip_setcb_nict_4xx_received(osip,&application_cb_rcv4xx);
osip_setcb_nict_5xx_received(osip,&application_cb_rcv5xx);
osip_setcb_nict_6xx_received(osip,&application_cb_rcv6xx);
osip_setcb_nist_1xx_sent(osip,&application_cb_snd1xx);
osip_setcb_nist_2xx_sent(osip,&application_cb_snd2xx);
osip_setcb_nist_3xx_sent(osip,&application_cb_snd3xx);
osip_setcb_nist_4xx_sent(osip,&application_cb_snd4xx);
osip_setcb_nist_5xx_sent(osip,&application_cb_snd5xx);
osip_setcb_nist_6xx_sent(osip,&application_cb_snd6xx);
osip_setcb_ist_invite_received      (osip,&application_cb_rcvinvite);
osip_setcb_ist_ack_received         (osip,&application_cb_rcvack);
osip_setcb_ist_ack_received2        (osip,&application_cb_rcvack2);
osip_setcb_nist_register_received(osip,&application_cb_rcvregister);
osip_setcb_nist_bye_received        (osip,&application_cb_rcvbye);
osip_setcb_nist_cancel_received    (osip,&application_cb_rcvcancel);
osip_setcb_nist_info_received       (osip,&application_cb_rcvinfo);
osip_setcb_nist_options_received (osip,&application_cb_rcvoptions);
osip_setcb_nist_subscribe_received(osip,&application_cb_rcvoptions);
osip_setcb_nist_notify_received    (osip,&application_cb_rcvoptions);
osip_setcb_nist_unknown_received (osip,&application_cb_rcvunkrequest);
```

5.3 Transaction 操作

在注册完 CALL BACK 函数后，应用程序可以建立 Transaction 来调用 oSIP 的解析器和状态机模块的操作，来实现不同应用程序的需求。

6. 参考

[1] SIP -- RFC3261 (http://www.ietf.org)

[2] SDP -- RFC2327(http://www.ieft.org)

[3] oSIP Library -- http://www.gnu.org/software/osip/

[4] oSIP mailing list -- http://www.atosc.org/pipermail/public/osip/