

OSIP 结构状态机部分文档

雷

OSIP 的核心是系统状态机，在不同情况下，系统处于不同的状态，在某一状态下当系统发生某一个动作后（如接受或者发送了消息），状态机做相应的跳转。的状态机在不同的状态下，对某一动作的响应也是不一样的。

由于 SIP 的复杂性，为了降低代码实现难度，也为了能够保证代码的模块化，Osip 把整个 SIP 的实现分成了四个状态机，分别是：

- (1) ICT （带 invit 事件的 out 处理）
- (2) IST （带 invit 事件的 in 处理）
- (3) NCT（不带 invit 事件的 out 处理）
- (4) NST（不带 invit 事件的 in 处理）

实际上，1 和 2 是一队连接，1 负责一个 callout，2 负责一个相应的 callin，3 和 4 也是一样，不同之处是他们负责的是不带 invite 的请求，比如 NOTIFY, SUBSCRIBE 等等。下面对这些状态机做说明。

- (1) ICT callout 的状态机，当系统 make a call out 的时候，系统处于这个状态机，这个状态机处理 1xx,2xx,3xxx 等消息，围绕这些消息，系统的状态机切换状态。
- (2) IST callin 的状态机，当系统接收到一个 invite 请求后，系统处于这个状态机下，在这个状态机下，系统发送 1xx,2xx 消息。可以看到，状态机 1 和 2 负责电话呼叫流程的控制，而 3 和 4 负责其它的非呼叫的状态控制。
- (3) NCT 无 invite 的 out 请求的状态机，实际上就是初了 invit 呼叫的其它状态处理，比如 NOTIFY 等等。
- (4) NST 无 invite 的 in 请求的状态机，和上面的对应，没什么好讲。

这四个状态机的实现是基本一致的，你理解了其中一个，理解其他三个就没有任何问题了，在这里我详细的将第一个，其它的您可以举一反三。如果看这些东西有困难，建议您先看下 RTC3261，熟悉下 SIP 呼叫流程，这也许对您理解 OSIP 状态机有帮助。

状态机 ICT（Invite client callout）说明

1、说明：

本文件的作用是解释终端 callout 的时候的状态控制，它是 osip 中的四个状态机中最重要的一个，处理 callout 的各种事件，比如 180，200，timeout 等等。文件中的函数是系收发消息的核心处理函数，为状态机切换服务。其它三个状态机和此类似。

包含文件：fsm.h、ict_fsm.c、ist_fsm.c、ict.c、ist.c。

状态五种状态机：

```
ICT_PRE_CALLING
ICT_CALLING
ICT_PROCEEDING
ICT_COMPLETED
ICT_TERMINATED
```

ICT_PRE_CALLING:

呼叫前的状态，系统处于准备状态。系统调用 osip_transaction_init 函数后，系统初始化，处于这个状态。在这个状态下，系统调用 ict_snd_invite，发送 invte 请求出去后，

系统跳转到 ICT_CALLING 状态。

ICT_CALLING:

已经成功发送 invit 请求，但还没有收到远方的任何响应的时候，系统处于此状态。在这个状态下，当 timeouta 时间到达后，系统会冲发消息，以确保对方可以收到请求包。当 timeb 时间到达后，系统认为对方无响应，状态直接跳到 ICT_TERMINATED 状态，释放资源完成此次呼叫。

当系统在此状态下接收到 1xx 消息后，认为对方有响应，则系统跳到 ICT_PROCEEDING 状态。

ICT_PROCEEDING:

当系统发送 invite 后，接收到对方的响应 (100 trying)，系统跳到此状态。正常接收到 200 OK 后，系统跳到 ICT_TERMINATED，认为完成一次请求。如果接收到 3456xxx 等请求，系统跳到 ICT_COMPLETED，发送 ACK 给对方（有些不需要发送 ACK），设置 timeoutd 时间，等待 D 时间到来。

ICT_COMPLETED:

当系统接收到 3456xx 的时候，系统发送 ACK 给对方，系统处于此状态，当 timeoutd 时间到达后，系统跳到 ICT_TERMINATED 状态，完成一次呼叫。

ICT_TERMINATED:

完成一次呼叫，这是状态机的必经的最后一个状态。状态到达这里，就表示了一个状态机的完结，也就表示了一个呼叫的完结，无论成功与否，最后都会到达这个终结状态。

系统状态切换的函数见状态图，图中其它几个 timeout 事件函数没有添加，需要了解的请自己查阅文件。

2、总结：

个人认为该说的就这么多，具体函数实现请自己看代码，没什么说的比较，没有什么很复杂的东西在里面。如果你了解 SIP 的流程，那么看这些状态机会很轻松，如果不了解建议你先看下 RTC 中的描述，这能够让你很快明白代码或者状态机的变换。

——雷

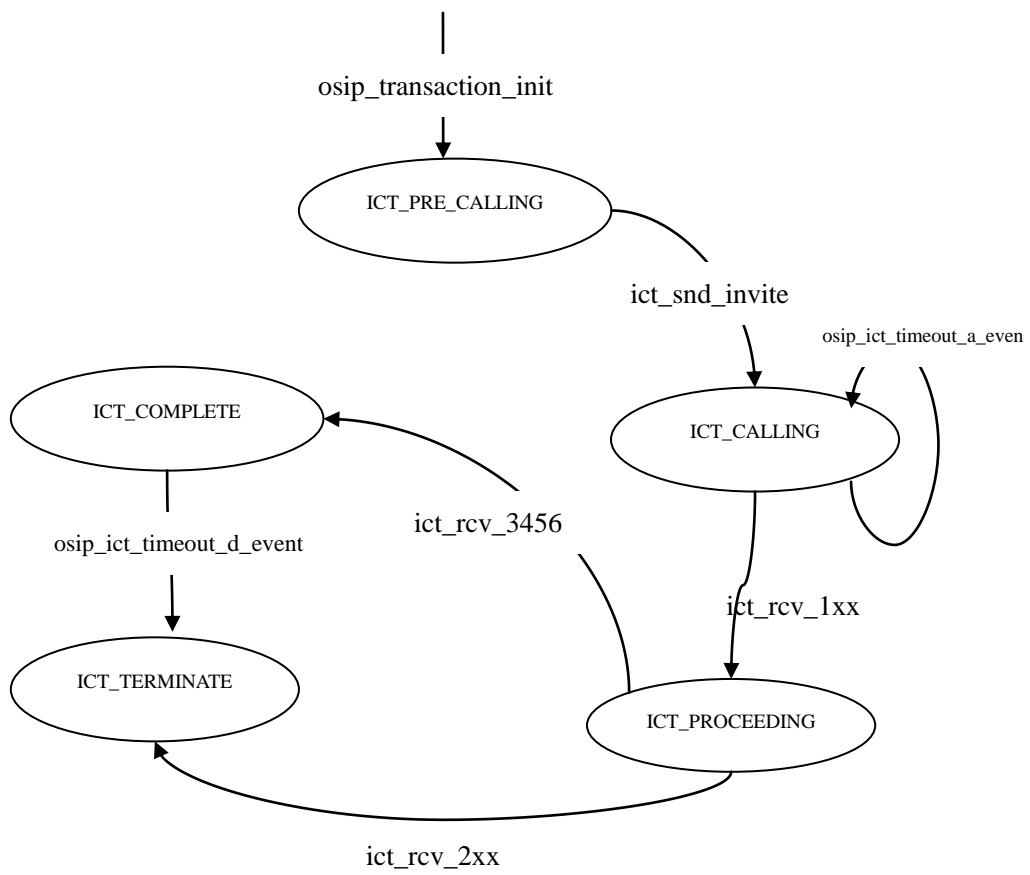


图 1 ICT 状态机