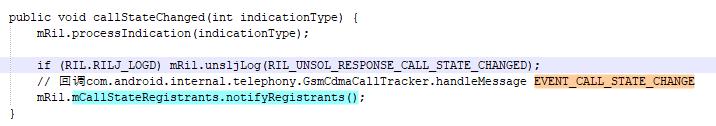
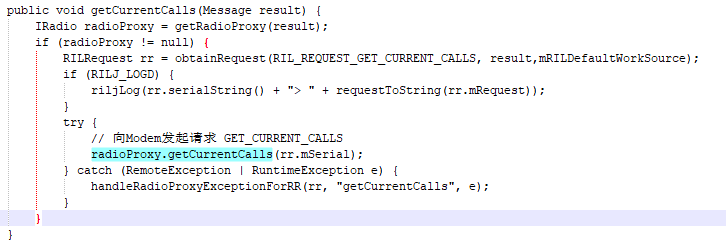
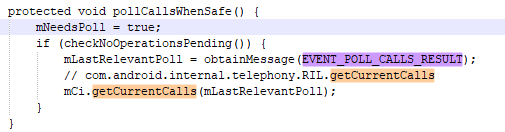
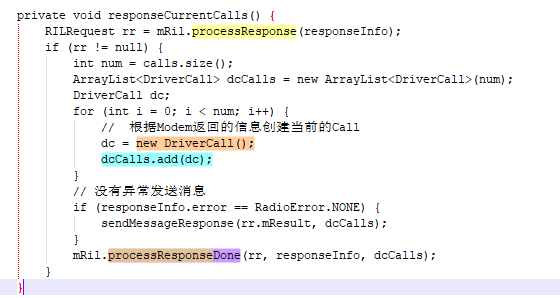
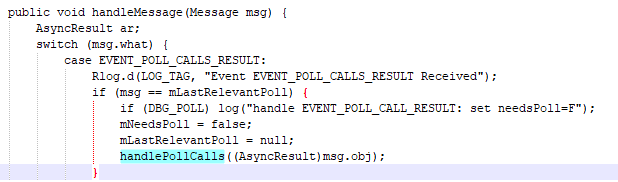
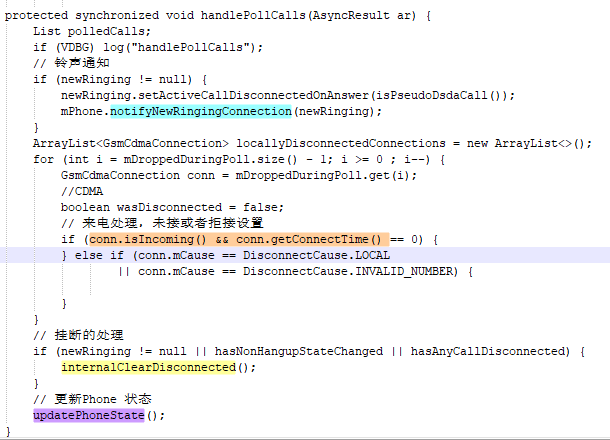
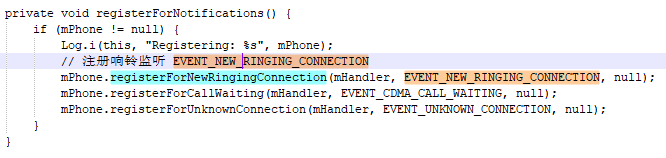
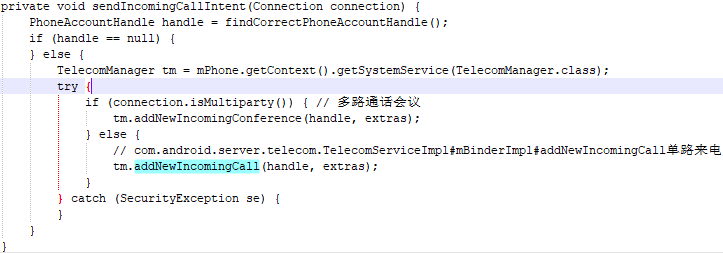
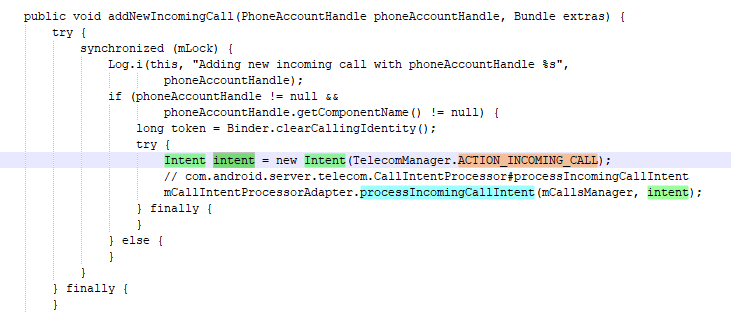
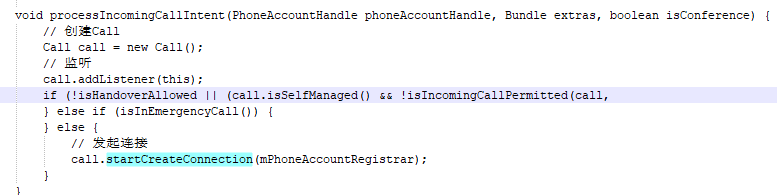
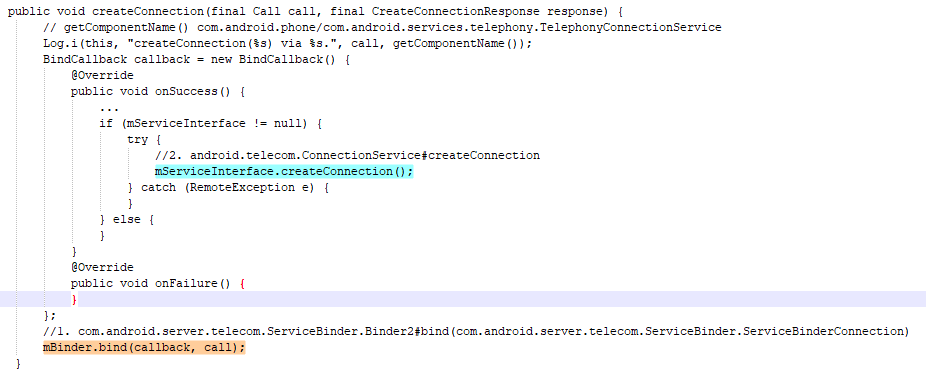
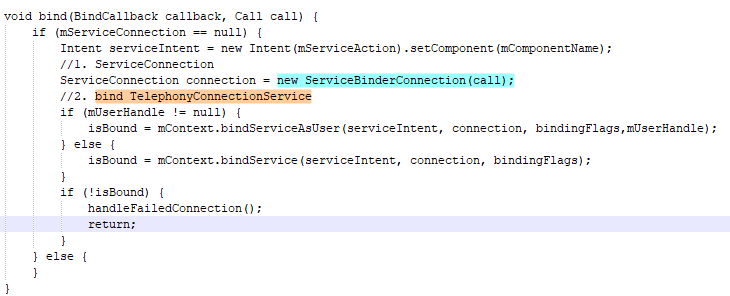
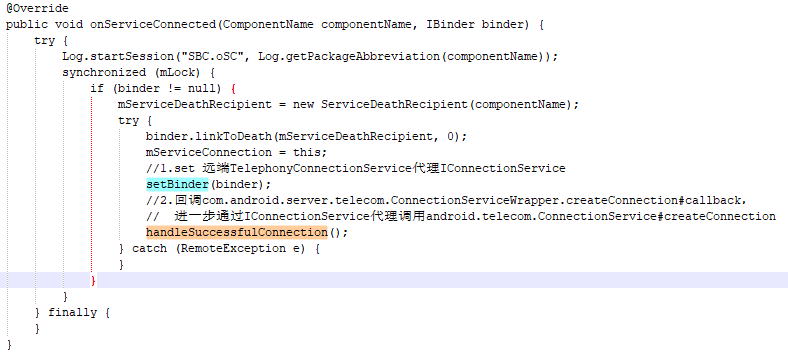
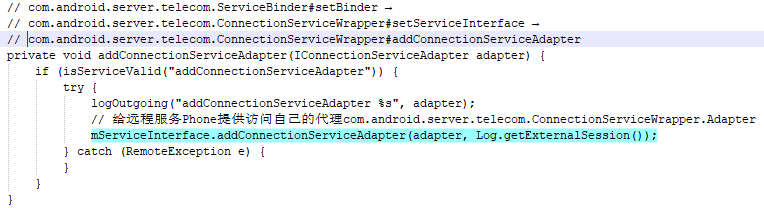
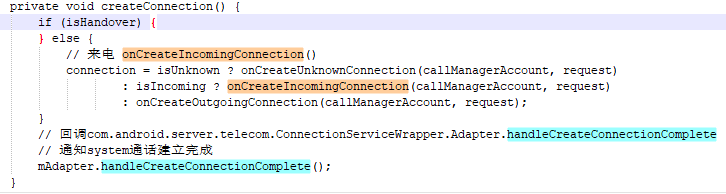
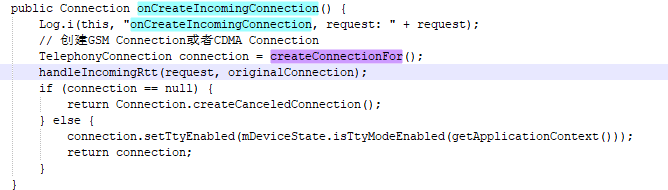
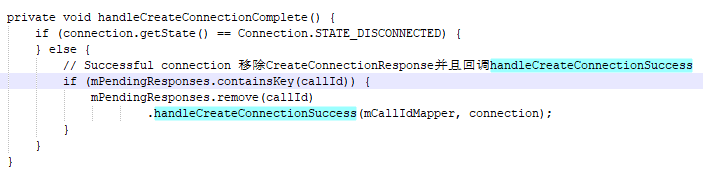
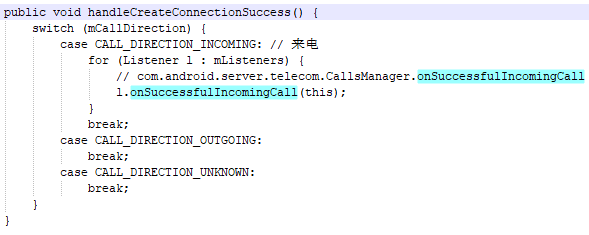
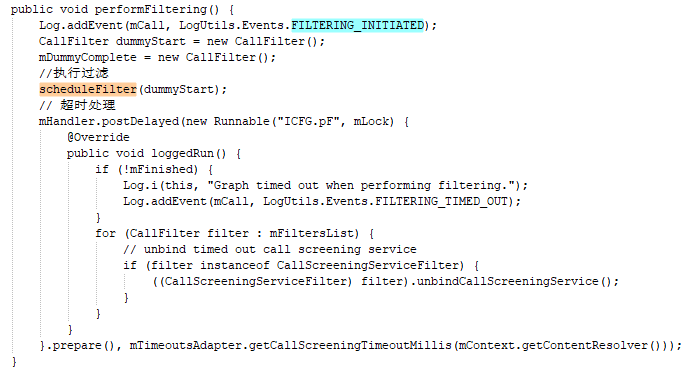
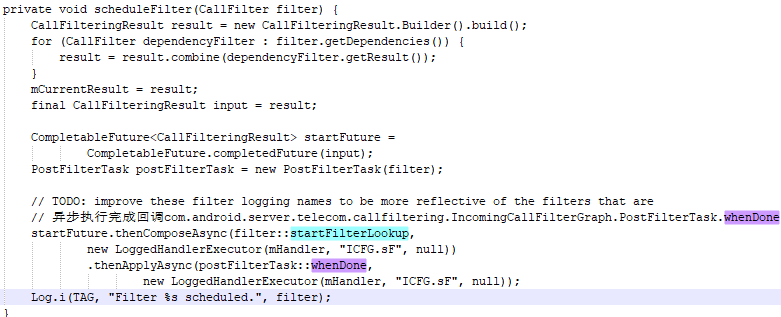
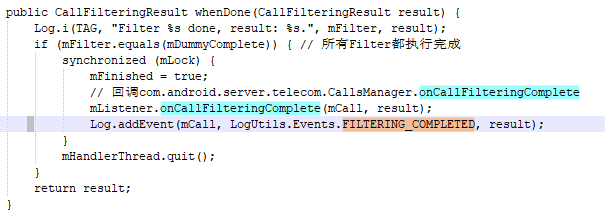
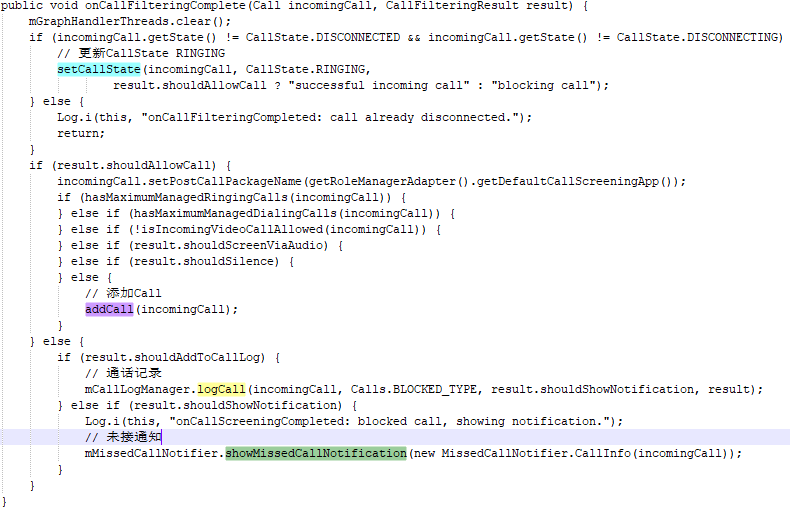
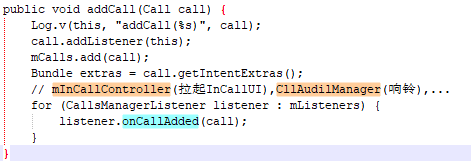
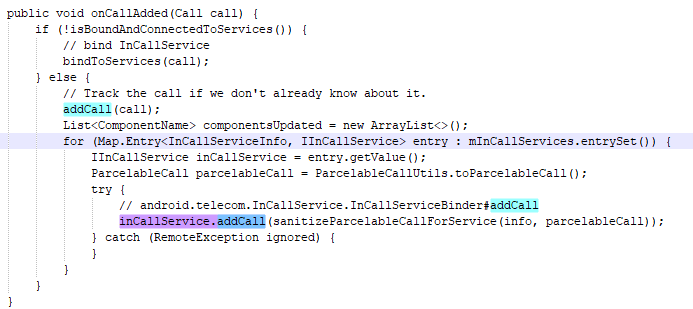
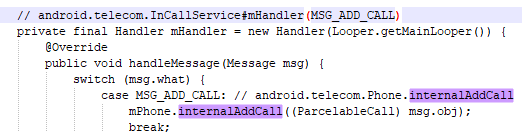
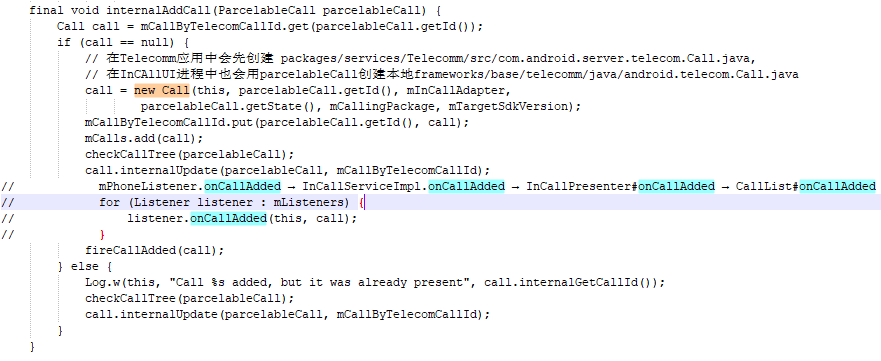
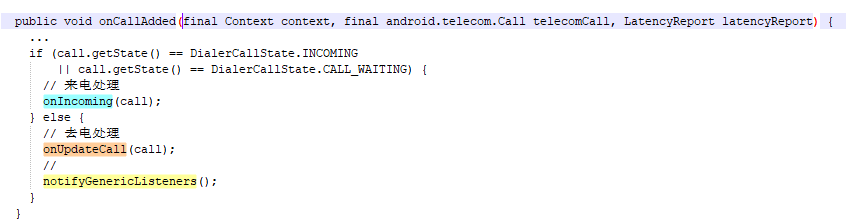


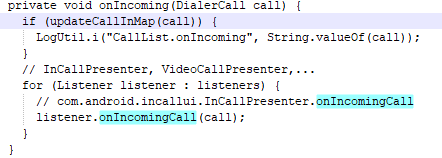
1. com.android.internal.telephony.RadioIndication#callStateChanged
2. com.android.internal.telephony.GsmCdmaCallTracker.handleMessage(EVENT\_CALL\_STATE\_CHANGE) → com.android.internal.telephony.CallTracker#pollCallsWhenSafe → com.android.internal.telephony.RIL#getCurrentCalls
3. Modem返回GET\_CURRENT\_CALLS请求结果com.android.internal.telephony.RadioResponse#getCurrentCallsResponse → com.android.internal.telephony.RadioResponse#responseCurrentCalls
4. GsmCdmaCallTracker的handleMessage中,之前发送请求的时候有发送EVENT\_POLL\_CALLS\_RESULT
5. com.android.internal.telephony.GsmCdmaCallTracker#handlePollCalls对Modem返回的消息解析，获取通话状态，判断如果来电发出响铃通知notifyNewRingingConnection，然后处理如果通话断开的操作及更新Phone状态。
6. notifyNewRingingConnection响铃消息，注册者Registrant，com.android.services.telephony.PstnIncomingCallNotifier#registerForNotifications
7. com.android.services.telephony.PstnIncomingCallNotifier#mHandler#handleMessage(EVENT\_NEW\_RINGING\_CONNECTION) → com.android.services.telephony.PstnIncomingCallNotifier#handleNewRingingConnection → com.android.services.telephony.PstnIncomingCallNotifier#sendIncomingCallIntent()
8. com.android.server.telecom.TelecomServiceImpl#mBinderImpl#addNewIncomingCall
9. com.android.server.telecom.CallIntentProcessor#processIncomingCallIntent → com.android.server.telecom.CallsManager#processIncomingCallIntent
10. com.android.server.telecom.Call#startCreateConnection(与MO绑定Phone流程一样) → CreateConnectionProcessor.process → CreateConnectionProcessor.attemptNextPhoneAccount
11. com.android.server.telecom.ConnectionServiceWrapper#createConnection，与com.android.phone/com.android.services.telephony.TelephonyConnectionService连接com.android.server.telecom.ServiceBinder.Binder2#bindcom.android.server.telecom.ServiceBinder.ServiceBinderConnection#onServiceConnected

* **给远程服务Phone提供访问自己的代理**
* **利用远程接口创建一个通话链接**

1. android.telecom.ConnectionService#mBinder#createConnection线程切换binder线程回到Phone主线程
2. android.telecom.ConnectionService#onCreateIncomingConnection → com.android.services.telephony.TelephonyConnectionService#onCreateIncomingConnection
3. android.telecom.ConnectionServiceAdapter#handleCreateConnectionComplete → com.android.server.telecom.ConnectionServiceWrapper.Adapter#handleCreateConnectionComplete，从Phone进程通过IConnectionServiceAdapter的Binder代理回调到Telecomm进程ConnectionServiceWrapper
4. com.android.server.telecom.CreateConnectionResponse#handleCreateConnectionSuccess → com.android.server.telecom.Call#handleCreateConnectionSuccess
5. com.android.server.telecom.CallsManager#onSuccessfulIncomingCall → com.android.server.telecom.callfiltering.IncomingCallFilterGraph#performFiltering
6. com.android.server.telecom.callfiltering.IncomingCallFilterGraph#scheduleFilter
7. com.android.server.telecom.callfiltering.IncomingCallFilterGraph.PostFilterTask#whenDone
8. com.android.server.telecom.CallsManager#onCallFilteringComplete
9. com.android.server.telecom.CallsManager#addCall
10. com.android.server.telecom.CallAudioManager#onCallAdded → com.android.server.telecom.CallAudioManager#addCall → com.android.server.telecom.CallAudioManager#onCallEnteringState → com.android.server.telecom.CallAudioManager#onCallEnteringRinging 控制电话Audio，耳机或外放等等
11. com.android.server.telecom.InCallController#onCallAdded
12. android.telecom.InCallService.InCallServiceBinder#addCall → android.telecom.InCallService#mHandler(MSG\_ADD\_CALL)
13. android.telecom.Phone#internalAddCall 创建一个本地的Telecom Call (不同于system进程的Call, 下面简称TCall)，通知 InCallService
14. com.android.incallui.call.CallList#onCallAdded,CallList 维护电话列表(CallList里面的Call简称LCall，区别与Phone进程的TCall)

* TCall是由Phone进程维护
* LCall是由InCallUI进程InCallList维护
* TCall和LCAll一一对应



1. com.android.incallui.call.CallList#onIncoming
2. InCallPresenter监听来电状态，com.android.incallui.InCallPresenter#onCallListChange
3. com.android.incallui.InCallPresenter#startOrFinishUi 界面显示。

MO是从InCallUI发起，MT是从Modem转发给Phone发起。两个流程最大的差异是如何开始创建Call和Connection。当开始创建和创建完成后，MO和MT在流程上没多大差异，都是Ccall和Connection控制，然后通过InCallUI显示出来。

**RIL → TelephonyFramework → TelephonyService → TelecomService → TelecomFramework → TelecomService → TelecomFramework → InCallUI**