触发trySetupData

1：DcTracker(constructor)-registerAllEvents

->registerServiceStateTrackerEvents

->EVENT\_DATA\_CONNECTION\_ATTACHED,

->onDataConnectionAttached

->setupDataOnAllConnectableApns

接收到RILJ主动上报：UNSOL\_RESPONSE\_NETWORK\_STATE\_CHANGED，然后ServiceStateTracker发出network 状态发生变化， 通知其他模块，比如DcTracker)

此时SIM卡已经loaded完成，data也attached。isDataAllowed为true，可以setupdata了

选择一项。

4g log:

trySetupData for APN type default, reason: dataEnabled. Data allowed, reason: NORMAL

buildWaitingApns: E requestedApnType=default

com.android.internal.telephony.SubscriptionController.getSubId(SubscriptionController.java:1454)

com.android.internal.telephony.SubscriptionController.getSubIdUsingPhoneId(SubscriptionController.java:1888)

com.android.internal.telephony.Phone.getSubId(Phone.java:3135)

com.android.internal.telephony.dataconnection.DcTracker.getPreferredApn(DcTracker.java:3629)

com.android.internal.telephony.dataconnection.DcTracker.buildWaitingApns(DcTracker.java:3501)

com.android.internal.telephony.dataconnection.DcTracker.trySetupData(DcTracker.java:1467)

com.android.internal.telephony.dataconnection.DcTracker.applyNewState(DcTracker.java:2479)

com.android.internal.telephony.dataconnection.DcTracker.onEnableApn(DcTracker.java:2575)

com.android.internal.telephony.dataconnection.DcTracker.handleMessage(DcTracker.java:3800)

android.os.Handler.dispatchMessage(Handler.java:106)

android.os.Looper.loop(Looper.java:193)

android.app.ActivityThread.main(ActivityThread.java:6923)

TT log:

trySetupData for APN type default, reason: dataEnabled. Data disallowed, reasons: NOT\_ATTACHED

notifyOffApnOfAvailability type:default

com.android.internal.telephony.SubscriptionController.getSubId(SubscriptionController.java:1454)

com.android.internal.telephony.SubscriptionController.getSubIdUsingPhoneId(SubscriptionController.java:1888)

com.android.internal.telephony.Phone.getSubId(Phone.java:3135)

com.android.internal.telephony.DefaultPhoneNotifier.doNotifyDataConnection(DefaultPhoneNotifier.java:161)

com.android.internal.telephony.DefaultPhoneNotifier.notifyDataConnection(DefaultPhoneNotifier.java:156)

com.android.internal.telephony.Phone.notifyDataConnection(Phone.java:2158)

com.android.internal.telephony.dataconnection.DcTracker.notifyOffApnsOfAvailability(DcTracker.java:1526)

com.android.internal.telephony.dataconnection.DcTracker.trySetupData(DcTracker.java:1495)

com.android.internal.telephony.dataconnection.DcTracker.applyNewState(DcTracker.java:2479)

com.android.internal.telephony.dataconnection.DcTracker.onEnableApn(DcTracker.java:2575)

com.android.internal.telephony.dataconnection.DcTracker.handleMessage(DcTracker.java:3800)

android.os.Handler.dispatchMessage(Handler.java:106)

android.os.Looper.loop(Looper.java:193)

android.app.ActivityThread.main(ActivityThread.java:6923)

2:

DcTracker(constructor)

->update

->onUpdateIcc

->EVENT\_RECORDS\_LOADED

->onRecordsLoadedOrSubIdChanged

->setupDataOnAllConnectableApns.

这个流程触发，必须是SIM加载已经ready。虽然这个时候也会触发trySetupData，但是由于此时数据还没有attached，所以就不会触发setUpData。

Log:

05-18 14:26:23.874 D/QtiDCT-C( 2185): [0]trySetupData for APN type default, \*\*reason: simLoaded\*\*, requestType=NORMAL. \*\*Data disallowed, reasons: NOT\_ATTACHED\*\*

05-18 14:26:23.874 D/QtiDCT-C( 2185): [0]\*\*trySetupData failed.\*\* apnContext = [type=default, mState=IDLE, apnEnabled=true, mDependencyMet=true]

3:

DcTracker监听到registerServiceStateTrackerEvents

->EVENT\_DATA\_RAT\_CHANGED，此时数据也没有attached，所以也不会触发setUpData。

05-18 14:26:29.346 D/QtiDCT-C( 2185): [0]trySetupData for APN type default, reason: \*\*nwTypeChanged\*\*, requestType=NORMAL. \*\*Data disallowed, reasons: NOT\_ATTACHED\*\*

05-18 14:26:29.346 D/QtiDCT-C( 2185): [0]\*\*trySetupData failed\*\*. apnContext = [type=default, mState=IDLE, apnEnabled=true, mDependencyMet=true]

4：

开机创建数据连接的流程为：

DcTracker(constructor)

->registerForAllEvents

->registerServiceStateTrackerEvents

->DctConstants.EVENT\_DATA\_CONNECTION\_ATTACHED

->onDataConnectionAttached

->setupDataOnAllConnectableApns

->trySetupData->setupData

->DataConnection.bringUp

->DataConnection.EVENT\_CONNECT(DcInactiveState)

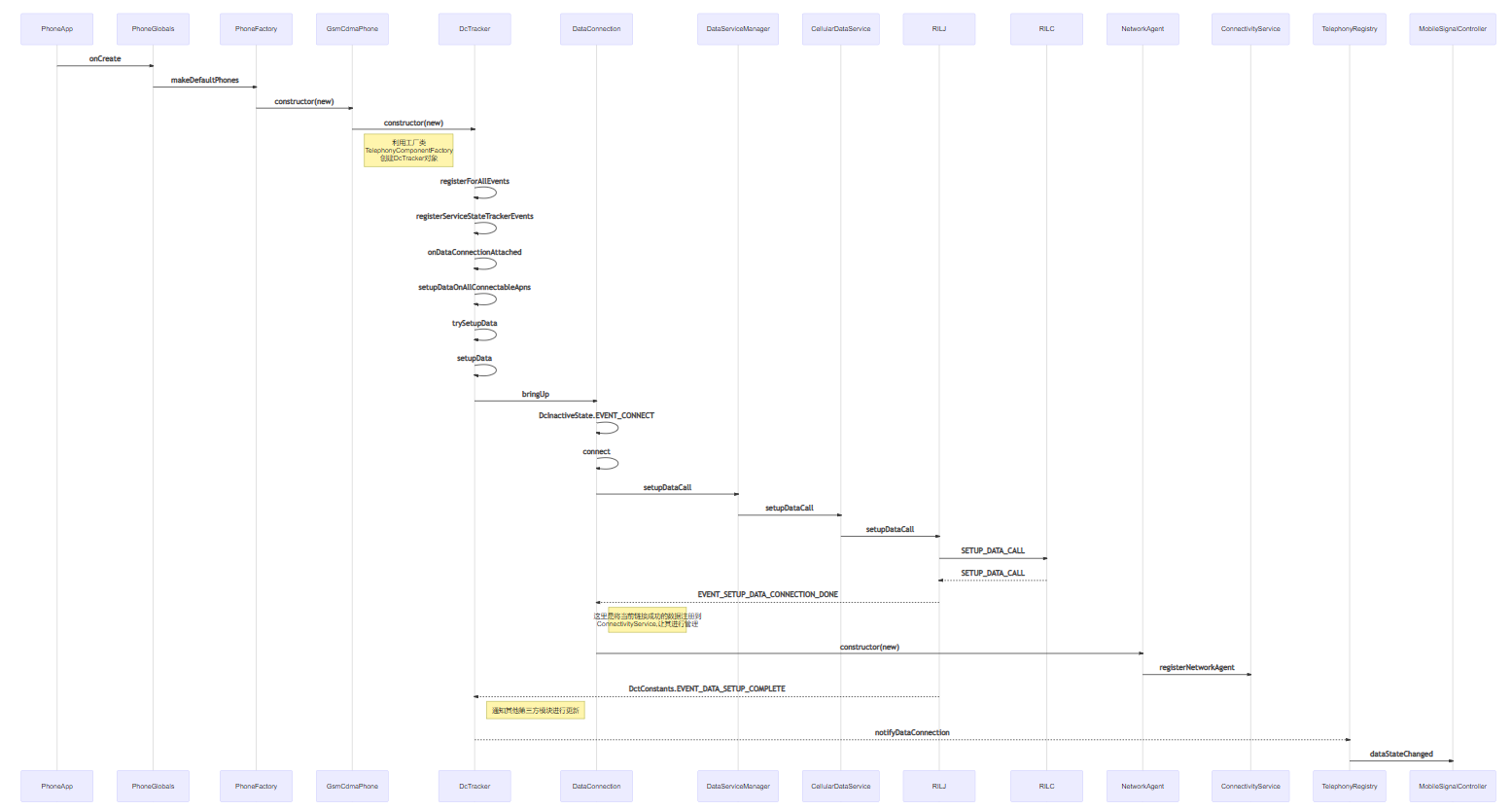
->connect(同时将状态切换到

mActivatingState)->DataServiceManager.setupDataCall

->CellularDataService.CellularDataServiceProvider.setupDataCall

->mPhone.mCi.setupDataCall

->RIL.setupDataCall.



05-18 14:26:21.489 W/TelephonyComponentFactory( 2185): Injection jar is not protected , path: /system/framework/qti-telephony-common.jarstatvfs failed: ENOENT (No such file or directory)

05-18 14:26:21.519 I/PhoneFactory( 2185): Cdma Subscription set to 0

05-18 14:26:21.519 I/PhoneFactory( 2185): Network Mode set to 10

05-18 14:26:21.521 D/RILJ ( 2185): RIL: init preferredNetworkType=10 cdmaSubscription=0) [SUBnull]

05-18 14:26:21.541 D/RILJ ( 2185): Unsol response received; Sending ack to ril.cpp [SUB0]

05-18 14:26:21.541 D/RILJ ( 2185): [UNSL]< UNSOL\_RIL\_CONNECTED [SUB0]

05-18 14:26:21.542 D/RILJ ( 2185): [0002]> RADIO\_POWER on = false [SUB0]

//Create GsmCdmaCallTracker

05-18 14:26:21.612 D/TelephonyComponentFactory( 2185): makeGsmCdmaCallTracker

//Create DcTracker

05-18 14:26:21.763 D/QtiTelephonyComponentFactory( 2185): makeQtiDcTracker

//init Apn on DcTracker constructor

05-18 14:26:21.784 D/ ( 2185): [0]initApnContexts: E

05-18 14:26:21.784 D/ ( 2185): [0]initApnContexts: skipping unknown type=1

05-18 14:26:21.784 D/ ( 2185): [0]initApnContexts: apnContext={mApnType=default mState=IDLE mWaitingApns={null} mApnSetting={null} mReason=dataEnabled mDataEnabled=false mDependencyMet=true}

05-18 14:26:21.784 D/ ( 2185): [0]initApnContexts: apnContext={mApnType=mms mState=IDLE mWaitingApns={null} mApnSetting={null} mReason=dataEnabled mDataEnabled=false mDependencyMet=true}

05-18 14:26:21.784 D/ ( 2185): [0]initApnContexts: apnContext={mApnType=supl mState=IDLE mWaitingApns={null} mApnSetting={null} mReason=dataEnabled mDataEnabled=false mDependencyMet=true}

05-18 14:26:21.784 D/ ( 2185): [0]initApnContexts: apnContext={mApnType=dun mState=IDLE mWaitingApns={null} mApnSetting={null} mReason=dataEnabled mDataEnabled=false mDependencyMet=true}

05-18 14:26:21.784 D/ ( 2185): [0]initApnContexts: apnContext={mApnType=hipri mState=IDLE mWaitingApns={null} mApnSetting={null} mReason=dataEnabled mDataEnabled=false mDependencyMet=true}

05-18 14:26:21.784 D/ ( 2185): [0]initApnContexts: apnContext={mApnType=fota mState=IDLE mWaitingApns={null} mApnSetting={null} mReason=dataEnabled mDataEnabled=false mDependencyMet=true}

05-18 14:26:21.784 D/ ( 2185): [0]initApnContexts: apnContext={mApnType=ims mState=IDLE mWaitingApns={null} mApnSetting={null} mReason=dataEnabled mDataEnabled=false mDependencyMet=true}

05-18 14:26:21.784 D/ ( 2185): [0]initApnContexts: apnContext={mApnType=cbs mState=IDLE mWaitingApns={null} mApnSetting={null} mReason=dataEnabled mDataEnabled=false mDependencyMet=true}

05-18 14:26:21.784 D/ ( 2185): [0]initApnContexts: skipping unknown type=7

05-18 14:26:21.784 D/ ( 2185): [0]initApnContexts: skipping unknown type=9

05-18 14:26:21.784 D/ ( 2185): [0]initApnContexts: apnContext={mApnType=emergency mState=IDLE mWaitingApns={null} mApnSetting={null} mReason=dataEnabled mDataEnabled=false mDependencyMet=true}

//Update ImsPhone

05-18 14:26:21.886 D/Phone ( 2185): updateImsPhone mImsServiceReady=true

//Create ImsPhoneCallTracker

05-18 14:26:21.887 D/TelephonyComponentFactory( 2185): makeImsPhoneCallTracker

//Register NetworkFactory to ConnectivityService, which will triggle trySetupData, but will failed in this case.

05-18 14:26:21.904 D/TelephonyNetworkFactory[0]( 2185): Registering NetworkFactory

05-18 14:26:21.905 D/ConnectivityService( 1407): Got NetworkFactory Messenger for TelephonyNetworkFactory[0]

//Notify SSTracker to notify DcTracker the data attached.

05-18 14:26:29.210 D/RILJ ( 2185): [UNSL]< UNSOL\_RESPONSE\_NETWORK\_STATE\_CHANGED [SUB0]

05-18 14:26:29.210 D/NRM-C ( 2185): onNetworkStateChanged

05-18 14:26:29.210 D/SST ( 2185): [0] Poll ServiceState done: oldSS=[{mVoiceRegState=0(IN\_SERVICE), mDataRegState=1(OUT\_OF\_SERVICE), mChannelNumber=38950, duplexMode()=0, mCellBandwidths=[], mVoiceOperatorAlphaLong=CMCC, mVoiceOperatorAlphaShort=CMCC, mDataOperatorAlphaLong=CMCC, mDataOperatorAlphaShort=CMCC, isManualNetworkSelection=false(automatic), getRilVoiceRadioTechnology=14(LTE), getRilDataRadioTechnology=0(Unknown), mCssIndicator=unsupported, mNetworkId=-1, mSystemId=-1, mCdmaRoamingIndicator=-1, mCdmaDefaultRoamingIndicator=-1, mIsEmergencyOnly=false, isUsingCarrierAggregation=false, mLteEarfcnRsrpBoost=0, mNetworkRegistrationInfos=[NetworkRegistrationInfo{ domain=PS transportType=WWAN registrationState=NOT\_REG\_OR\_SEARCHING roamingType=NOT\_ROAMING accessNetworkTechnology=UNKNOWN rejectCause=0 emergencyEnabled=false availableServices=[] cellIdentity=null voiceSpecificInfo=null dataSpecificInfo=android.telephony.DataSpecificRegistrationInfo :{ maxDataCalls = 20 isDcNrRestricted = false isNrAvailable = false isEnDcAvailable = false LteVopsSupportInfo : mVopsSupport = 1 mEmcBearerSupport = 1 mIsUsingCarrierAggregation = false } nrState=NONE}, NetworkRegistrationInfo{ domain=CS transportType=WWAN registrationState=HOME roamingType=NOT\_ROAMING accessNetworkTechnology=LTE rejectCause=0 emergencyEnabled=false availableServices=[VOICE,SMS,VIDEO] cellIdentity=CellIdentityLte:{ mCi=21765889 mPci=89 mTac=4421 mEarfcn=38950 mBandwidth=2147483647 mMcc=460 mMnc=00 mAlphaLong=CMCC mAlphaShort=CMCC} voiceSpecificInfo=VoiceSpecificRegistrationInfo { mCssSupported=false mRoamingIndicator=-1 mSystemIsInPrl=-1 mDefaultRoamingIndicator=-1} dataSpecificInfo=null nrState=NONE}], mNrFrequencyRange=-1, mOperatorAlphaLongRaw=CMCC, mOperatorAlphaShortRaw=CMCC, mIsIwlanPreferred=false}] newSS=[{mVoiceRegState=0(IN\_SERVICE), mDataRegState=0(IN\_SERVICE), mChannelNumber=38950, duplexMode()=2, mCellBandwidths=[], mVoiceOperatorAlphaLong=CMCC, mVoiceOperatorAlphaShort=CMCC, mDataOperatorAlphaLong=CMCC, mDataOperatorAlphaShort=CMCC, isManualNetworkSelection=false(automatic), getRilVoiceRadioTechnology=14(LTE), getRilDataRadioTechnology=14(LTE), mCssIndicator=unsupported, mNetworkId=-1, mSystemId=-1, mCdmaRoamingIndicator=-1, mCdmaDefaultRoamingIndicator=-1, mIsEmergencyOnly=false, isUsingCarrierAggregation=false, mLteEarfcnRsrpBoost=0, mNetworkRegistrationInfos=[NetworkRegistrationInfo{ domain=CS transportType=WWAN registrationState=HOME roamingType=NOT\_ROAMING accessNetworkTechnology=LTE rejectCause=0 emergencyEnabled=false availableServices=[VOICE,SMS,VIDEO] cellIdentity=CellIdentityLte:{ mCi=21765889 mPci=89 mTac=4421 mEarfcn=38950 mBandwidth=2147483647 mMcc=460 mMnc=00 mAlphaLong=CMCC mAlphaShort=CMCC} voiceSpecificInfo=VoiceSpecificRegistrationInfo { mCssSupported=false mRoamingIndicator=1 mSystemIsInPrl=-1 mDefaultRoamingIndicator=-1} dataSpecificInfo=null nrState=NONE}, NetworkRegistrationInfo{ domain=PS transportType=WWAN registrationState=HOME roamingType=NOT\_ROAMING accessNetworkTechnology=LTE rejectCause=0 emergencyEnabled=false availableServices=[DATA] cellIdentity=CellIdentityLte:{ mCi=21765889 mPci=89 mTac=4421 mEarfcn=38950 mBandwidth=2147483647 mMcc=460 mMnc=00 mAlphaLong=CMCC mAlphaShort=CMCC} voiceSpecificInfo=null dataSpecificInfo=android.telephony.DataSpecificRegistrationInfo :{ maxDataCalls = 20 isDcNrRestricted = false isNrAvailable = false isEnDcAvailable = false LteVopsSupportInfo : mVopsSupport = 2 mEmcBearerSupport = 3 mIsUsingCarrierAggregation = false } nrState=NONE}], mNrFrequencyRange=-1, mOperatorAlphaLongRaw=CMCC, mOperatorAlphaShortRaw=CMCC, mIsIwlanPreferred=false}] oldMaxDataCalls=20 mNewMaxDataCalls=20 oldReasonDataDenied=0 mNewReasonDataDenied=0

05-18 14:26:29.210 D/SST ( 2185): [0] pollStateDone: hasRegistered = false hasDeregistered = false hasDataAttached = {1=true} hasDataDetached = {1=false} hasDataRegStateChanged = {1=true} hasRilVoiceRadioTechnologyChanged = false hasRilDataRadioTechnologyChanged = {1=true} hasDataTransportPreferenceChanged = false hasChanged = true hasVoiceRoamingOn = false hasVoiceRoamingOff = false hasDataRoamingOn =false hasDataRoamingOff = false hasLocationChanged = false has4gHandoff = false hasMultiApnSupport = false hasLostMultiApnSupport = false hasCssIndicatorChanged = false hasNrFrequencyRangeChanged = false hasNrStateChanged = false hasAirplaneModeOnlChanged = false

\*\*05-18 14:26:29.347 D/QtiDCT-C( 2185): [0]onDataConnectionAttached\*\*

05-18 14:26:29.347 D/QtiDCT-C( 2185): [0]setupDataOnAllConnectableApns: dataAttached emergency:[state=IDLE,enabled=false] mms:[state=IDLE,enabled=false] supl:[state=IDLE,enabled=false] hipri:[state=IDLE,enabled=false] dun:[state=IDLE,enabled=false] fota:[state=IDLE,enabled=false] ims:[state=IDLE,enabled=false] default:[state=IDLE,enabled=true] cbs:[state=IDLE,enabled=false]

05-18 14:26:29.347 D/QtiDCT-C( 2185): [0]isConnectable() call trySetupData

05-18 14:26:29.352 D/QtiDCT-C( 2185): [0]trySetupData for APN type default, reason: dataAttached, requestType=NORMAL. Data allowed, reason: NORMAL

05-18 14:26:29.357 D/QtiDCT-C( 2185): [0]setupData: apnContext={mApnType=default mState=IDLE mWaitingApns={[[ApnSettingV7] ä¸­å›½ç§»åŠ¨ (China Mobile) GPRS, 2775, 46007, cmnet, , null, , null, null, 3, supl | hipri | default, IPV4V6, IPV4V6, true, 0, false, 0, 0, 0, 0, null, , false, 0, 0, -1, -1]} mApnSetting={null} mReason=dataAttached mDataEnabled=true mDependencyMet=true}, requestType=NORMAL

05-18 14:26:29.361 D/QtiDCT-C( 2185): [0]setupData: dc={DC-QtiDC-1: State=DcInactiveState mApnSetting=null RefCount=0 mCid=-1 mCreateTime=0 mLastastFailTime=0 mLastFailCause=0 mTag=0 mLinkProperties={LinkAddresses: [ ] DnsAddresses: [ ] Domains: null MTU: 0 Routes: [ ]} linkCapabilities=[ Transports: CELLULAR Capabilities: NOT\_RESTRICTED&TRUSTED&NOT\_VPN&NOT\_ROAMING&NOT\_CONGESTED LinkUpBandwidth>=15000Kbps LinkDnBandwidth>=30000Kbps Specifier: <0>] mRestrictedNetworkOverride=false mApnContexts={}} apnSetting=[ApnSettingV7] ä¸­å›½ç§»åŠ¨ (China Mobile) GPRS, 2775, 46007, cmnet, , null, , null, null, 3, supl | hipri | default, IPV4V6, IPV4V6, true, 0, false, 0, 0, 0, 0, null, , false, 0, 0, -1, -1 gen#=1

05-18 14:26:29.368 D/DC-QtiDC-1( 2185): bringUp: apnContext={mApnType=default mState=CONNECTING mWaitingApns={[[ApnSettingV7] ä¸­å›½ç§»åŠ¨ (China Mobile) GPRS, 2775, 46007, cmnet, , null, , null, null, 3, supl | hipri | default, IPV4V6, IPV4V6, true, 0, false, 0, 0, 0, 0, null, , false, 0, 0, -1, -1]} mApnSetting={[ApnSettingV7] ä¸­å›½ç§»åŠ¨ (China Mobile) GPRS, 2775, 46007, cmnet, , null, , null, null, 3, supl | hipri | default, IPV4V6, IPV4V6, true, 0, false, 0, 0, 0, 0, null, , false, 0, 0, -1, -1} mReason=dataAttached mDataEnabled=true mDependencyMet=true} onCompletedMsg={ when=-24s962ms what=270336 obj=Pair{{mApnType=default mState=CONNECTING mWaitingApns={[[ApnSettingV7] ä¸­å›½ç§»åŠ¨ (China Mobile) GPRS, 2775, 46007, cmnet, , null, , null, null, 3, supl | hipri | default, IPV4V6, IPV4V6, true, 0, false, 0, 0, 0, 0, null, , false, 0, 0, -1, -1]} mApnSetting={[ApnSettingV7] ä¸­å›½ç§»åŠ¨ (China Mobile) GPRS, 2775, 46007, cmnet, , null, , null, null, 3, supl | hipri | default, IPV4V6, IPV4V6, true, 0, false, 0, 0, 0, 0, null, , false, 0, 0, -1, -1} mReason=dataAttached mDataEnabled=true mDependencyMet=true} 1} target=com.qualcomm.qti.internal.telephony.dataconnection.QtiDcTracker }

05-18 14:26:29.368 D/DC-QtiDC-1( 2185): connect: carrier='ä¸­å›½ç§»åŠ¨ (China Mobile) GPRS' APN='cmnet' proxy='' port='-1'

05-18 14:26:29.378 D/DSM-C ( 2185): setupDataCall

05-18 14:26:29.379 D/RILJ ( 2185): [0250]> SETUP\_DATA\_CALL,dataRat=14,isRoaming=false,allowRoaming=false,DataProfile=-1/2/3/cmnet///0/0/0/0/true/21/2/0/0/false/true [SUB0]

05-18 14:26:29.755 D/RILJ ( 2185): [0250]< SETUP\_DATA\_CALL DataCallResponse: { cause=0 retry=-1 cid=0 linkStatus=2 protocolType=2 ifname=rmnet\_data1 addresses=[10.8.126.27/29] dnses=[/221.179.155.193, /221.179.155.209] gateways=[/10.8.126.28] pcscf=[] mtu=0} [SUB0]

05-18 14:26:29.755 D/DSM-C ( 2185): onSetupDataCallComplete. resultCode = 0, response = DataCallResponse: { cause=0 retry=-1 cid=0 linkStatus=2 protocolType=2 ifname=rmnet\_data1 addresses=[10.8.126.27/29] dnses=[/221.179.155.193, /221.179.155.209] gateways=[/10.8.126.28] pcscf=[] mtu=0}

05-18 14:26:29.757 D/DC-QtiDC-1( 2185): DcActivatingState: msg={what=EVENT\_SETUP\_DATA\_CONNECTION\_DONE when=-1ms obj={mTag=2 mApnContext={mApnType=default mState=CONNECTING mWaitingApns={[[ApnSettingV7] ä¸­å›½ç§»åŠ¨ (China Mobile) GPRS, 2775, 46007, cmnet, , null, , null, null, 3, supl | hipri | default, IPV4V6, IPV4V6, true, 0, false, 0, 0, 0, 0, null, , false, 0, 0, -1, -1]} mApnSetting={[ApnSettingV7] ä¸­å›½ç§»åŠ¨ (China Mobile) GPRS, 2775, 46007, cmnet, , null, , null, null, 3, supl | hipri | default, IPV4V6, IPV4V6, true, 0, false, 0, 0, 0, 0, null, , false, 0, 0, -1, -1} mReason=dataAttached mDataEnabled=true mDependencyMet=true} mProfileId=-1 mRat=14 mOnCompletedMsg={what=0x42000 when=-25s351ms obj=Pair{{mApnType=default mState=CONNECTING mWaitingApns={[[ApnSettingV7] ä¸­å›½ç§»åŠ¨ (China Mobile) GPRS, 2775, 46007, cmnet, , null, , null, null, 3, supl | hipri | default, IPV4V6, IPV4V6, true, 0, false, 0, 0, 0, 0, null, , false, 0, 0, -1, -1]} mApnSetting={[ApnSettingV7] ä¸­å›½ç§»åŠ¨ (China Mobile) GPRS, 2775, 46007, cmnet, , null, , null, null, 3, supl | hipri | default, IPV4V6, IPV4V6, true, 0, false, 0, 0, 0, 0, null, , false, 0, 0, -1, -1} mReason=dataAttached mDataEnabled=true mDependencyMet=true} 1} target=Handler (com.qualcomm.qti.internal.telephony.dataconnection.QtiDcTracker) {5996016} replyTo=null} mRequestType=NORMAL mSubId=1} target=Handler (com.android.internal.util.StateMachine$SmHandler) {9c33f6d} replyTo=null}

05-18 14:26:29.757 D/DC-QtiDC-1( 2185): onSetupConnectionCompleted: resultCode=0, response=DataCallResponse: { cause=0 retry=-1 cid=0 linkStatus=2 protocolType=2 ifname=rmnet\_data1 addresses=[10.8.126.27/29] dnses=[/221.179.155.193, /221.179.155.209] gateways=[/10.8.126.28] pcscf=[] mtu=0}

05-18 14:26:29.757 D/DC-QtiDC-1( 2185): onSetupConnectionCompleted received successful DataCallResponse

05-18 14:26:29.772 D/QtiDCT-C( 2185): [0]completeConnection: successful, notify the world apnContext={mApnType=default mState=CONNECTED mWaitingApns={[[ApnSettingV7] ä¸­å›½ç§»åŠ¨ (China Mobile) GPRS, 2775, 46007, cmnet, , null, , null, null, 3, supl | hipri | default, IPV4V6, IPV4V6, true, 0, false, 0, 0, 0, 0, null, , false, 0, 0, -1, -1]} mApnSetting={[ApnSettingV7] ä¸­å›½ç§»åŠ¨ (China Mobile) GPRS, 2775, 46007, cmnet, , null, , null, null, 3, supl | hipri | default, IPV4V6, IPV4V6, true, 0, false, 0, 0, 0, 0, null, , false, 0, 0, -1, -1} mReason=connected mDataEnabled=true mDependencyMet=true}

05-18 14:26:29.772 D/GsmCdmaPhone( 2185): [0] getDataConnectionState apnType=default ret=CONNECTED

05-18 14:26:29.776 D/ConnectivityService( 1407): registerNetworkAgent NetworkAgentInfo{ ni{[type: MOBILE[LTE], state: CONNECTED/CONNECTED, reason: connected, extra: cmnet, failover: false, available: true, roaming: false]} network{100} nethandle{432902426637} lp{{InterfaceName: rmnet\_data1 LinkAddresses: [ 10.8.126.27/29 ] DnsAddresses: [ /221.179.155.193,/221.179.155.209 ] Domains: null MTU: 1500 TcpBufferSizes: 524288,1048576,8388608,262144,524288,4194304 Routes: [ 0.0.0.0/0 -> 10.8.126.28 rmnet\_data1,10.8.126.24/29 -> 0.0.0.0 rmnet\_data1 ]}} nc{[ Transports: CELLULAR Capabilities: SUPL&INTERNET&NOT\_RESTRICTED&TRUSTED&NOT\_VPN&NOT\_ROAMING&FOREGROUND&NOT\_CONGESTED&NOT\_SUSPENDED LinkUpBandwidth>=15000Kbps LinkDnBandwidth>=30000Kbps Specifier: <1>]} Score{10} everValidated{false} lastValidated{false} created{false} lingering{false} explicitlySelected{false} acceptUnvalidated{false} everCaptivePortalDetected{false} lastCaptivePortalDetected{false} captivePortalValidationPending{false} partialConnectivity{false} acceptPartialConnectivity{false} clat{mBaseIface: null, mIface: null, mState: IDLE} }

05-18 14:26:29.780 D/QtiDCT-C( 2185): [0]onDataSetupComplete: SETUP complete type=default

***APN流程:***

public DcTracker(Phone phone) {

.......

//每个Phone对象有自己DcTracker

//每个DcTracker加载各自卡可用的APN

mPhone = phone;

.......

//1、监听卡载入

mUiccController = UiccController.getInstance();

mUiccController.registerForIccChanged(this, DctConstants.EVENT\_ICC\_CHANGED, null);

.......

//2、监听卡信息变化

mSubscriptionManager = SubscriptionManager.from(mPhone.getContext());

mSubscriptionManager.addOnSubscriptionsChangedListener(mOnSubscriptionsChangedListener);

.......

//监听APN数据库变化

mApnObserver = new ApnChangeObserver();

phone.getContext().getContentResolver().registerContentObserver(

Telephony.Carriers.CONTENT\_URI, true, mApnObserver);

.............

//初始化不同APN类型对应的网络能力，后文介绍

initApnContexts();

.............

// Add Emergency APN to APN setting list by default to support EPDN in sim absent cases

initEmergencyApnSetting();

addEmergencyApnSetting();

...............

}

插卡或卡发生变化后，就要创建当前卡可用的 APN，同时设置初始时使用的 APN

private void onRecordsLoadedOrSubIdChanged() {

..............

//1、创建当前卡可用的APN

createAllApnList();

//2、设置初始使用的APN

setInitialAttachApn();

if (mPhone.mCi.getRadioState().isOn()) {

if (DBG) log("onRecordsLoadedOrSubIdChanged: notifying data availability");

notifyOffApnsOfAvailability(Phone.REASON\_SIM\_LOADED);

}

//卡变化也会触发拨号流程；不过若此时数据开关未开，那么拨号是不会成功的

setupDataOnConnectableApns(Phone.REASON\_SIM\_LOADED);

}

创建卡对应的 APN 的过程：

private void createAllApnList() {

//表示mvno是否匹配

//mvno也是APN的一种属性，代表该APN适用于虚拟运营商，目前用的比较少

mMvnoMatched = false;

//用于保存结果

mAllApnSettings = new ArrayList<ApnSetting>();

//得到当前卡的信息

IccRecords r = mIccRecords.get();

//得到卡对应的MCC/MNC

String operator = (r != null) ? r.getOperatorNumeric() : "";

if (operator != null) {

//构造SQL语句

String selection = "numeric = '" + operator + "'";

String orderBy = "\_id";

...............

//查询MCC/MNC对应的APN

Cursor cursor = mPhone.getContext().getContentResolver().query(

Telephony.Carriers.CONTENT\_URI, null, selection, null, orderBy);

if (cursor != null) {

if (cursor.getCount() > 0) {

//1、利用数据创建APN

mAllApnSettings = createApnList(cursor);

}

cursor.close();

}

}

//2、添加emergencyApnSettings

addEmergencyApnSetting();

//3、去除重复的APN

dedupeApnSettings();

if (mAllApnSettings.isEmpty()) {

mPreferredApn = null;

} else {

//4、得到用户偏爱的APN (用户在UI界面主动选择的)

mPreferredApn = getPreferredApn();

if (mPreferredApn != null && !mPreferredApn.numeric.equals(operator) {

mPreferredApn = null;

//用户偏爱的与当前卡不匹配，删除数据库中对应信息

setPreferredApn(-1);

}

}

//5、在需要的情况下，构造APN文件发送给modem

setDataProfilesAsNeeded();

}

private void createAllApnList() {

//表示mvno是否匹配

//mvno也是APN的一种属性，代表该APN适用于虚拟运营商，目前用的比较少

mMvnoMatched = false;

//用于保存结果

mAllApnSettings = new ArrayList<ApnSetting>();

//得到当前卡的信息

IccRecords r = mIccRecords.get();

//得到卡对应的MCC/MNC

String operator = (r != null) ? r.getOperatorNumeric() : "";

if (operator != null) {

//构造SQL语句

String selection = "numeric = '" + operator + "'";

String orderBy = "\_id";

...............

//查询MCC/MNC对应的APN

Cursor cursor = mPhone.getContext().getContentResolver().query(

Telephony.Carriers.CONTENT\_URI, null, selection, null, orderBy);

if (cursor != null) {

if (cursor.getCount() > 0) {

//1、利用数据创建APN

mAllApnSettings = createApnList(cursor);

}

cursor.close();

}

}

//2、添加emergencyApnSettings

addEmergencyApnSetting();

//3、去除重复的APN

dedupeApnSettings();

if (mAllApnSettings.isEmpty()) {

mPreferredApn = null;

} else {

//4、得到用户偏爱的APN (用户在UI界面主动选择的)

mPreferredApn = getPreferredApn();

if (mPreferredApn != null && !mPreferredApn.numeric.equals(operator) {

mPreferredApn = null;

//用户偏爱的与当前卡不匹配，删除数据库中对应信息

setPreferredApn(-1);

}

}

//5、在需要的情况下，构造APN文件发送给modem

setDataProfilesAsNeeded();

}