| WifiManager.java | WifiServiceImpl.java | | WifiStateMachine.java | WifiNative.java | WificondControl.java | system/connectivity/wificond/
client_interface_binder.cpp | system/connectivity/wificond/
client_interface_impl.cpp | system/connectivity/wificond/
net/netlink_utils.cpp |

getConnectionInfo

mWifiStateMachine.syncRequestConnectionInfo()

WifiInfo result = new WifiInfo(mWifiInfo);
直接返回类的成员变量

连接WIFI以后mWifiInfo RSSI的赋值
L2ConnectedState CMD_RSSI_POLL
fetchRssiLinkSpeedAndFrequencyNative()

mWifiInfo.setRssi(newRssi);

signalPoll

signalPoll

IClientInterface iface = getClientInterface(ifaceName);
resultArray = iface.signalPoll()

ClientInterfaceBinder::signalPoll
impl_->SignalPoll

netlink_utils_->GetStationInfo

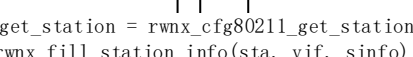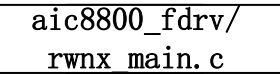NetlinkUtils::GetStationInfo

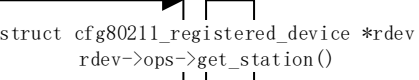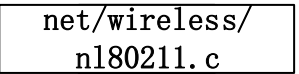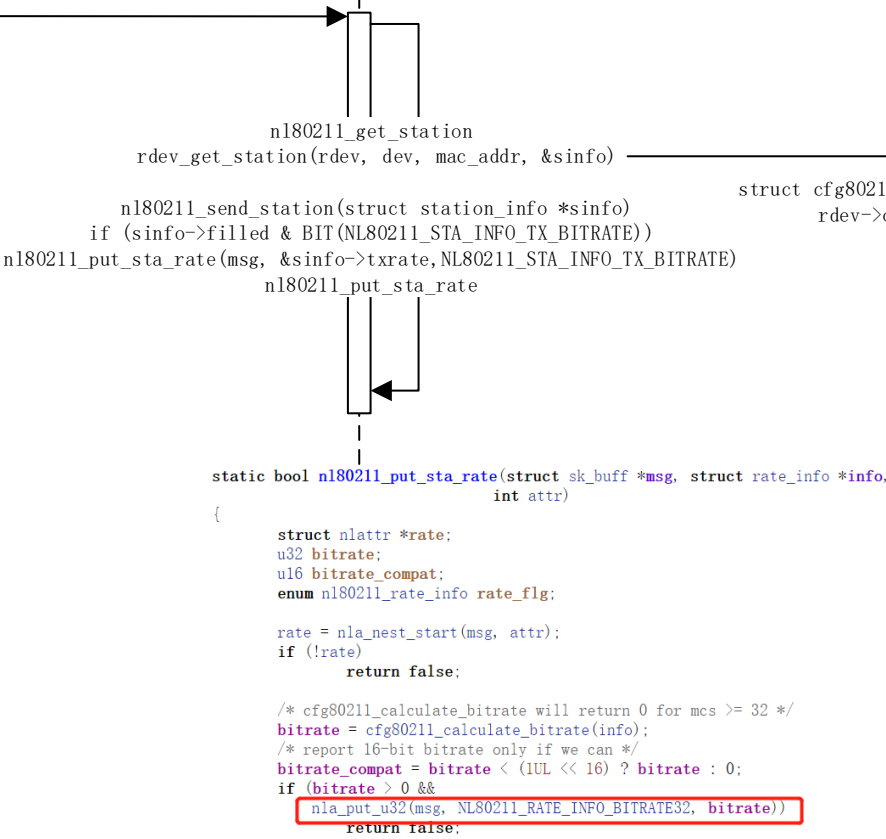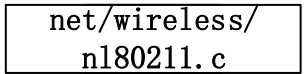fetchRssiLinkSpeedAndFrequencyNative 获取当前连接的RSSI 连接速度 信道频率

Enable rssi poll
private boolean mEnableRssiPolling = false

可以在WifiInfo.setRssi 方法里面添加Log的堆栈打印来解决
Log.d(TAG,Log.getStackTraceString(new Throwable()));

NL80211Packet get_station(NL80211_CMD_GET_STATION)
netlink_manager_->SendMessageAndGetSingleResponse(get_station,&response)
获得sta信息

| net/wireless/
nl80211.c | net/wireless/
nl80211.c | aic8800_fdrv/
rwnx_main.c |

NL80211NestedAttr sta_info(0);
response->GetAttribute(NL80211_ATTR_STA_INFO, &sta_info)
从response中获得sta_info
sta_info.GetAttribute(NL80211_STA_INFO_TX_BITRATE,&tx_bitrate_attr)
tx_bitrate_attr.GetAttributeValue(NL80211_RATE_INFO_BITRATE32,&tx_bitrate)
sta_info中获得该属性检查检查报错 return false

```
uint32_t tx_bitrate;
if (!tx_bitrate_attr.GetAttributeValue(NL80211_RATE_INFO_BITRATE32,
                                        &tx_bitrate)) {
    LOG(ERROR) << "Failed to get NL80211_RATE_INFO_BITRATE32";
    return false;
}
```

nl80211_get_station
rdev_get_station(rdev, dev, mac_addr, &sinfo)

struct cfg80211_registered_device *rdev
rdev->ops->get_station()

nl80211_send_station(struct station_info *sinfo)
if (sinfo->filled & BIT(NL80211_STA_INFO_TX_BITRATE))
nl80211_put_sta_rate(msg, &sinfo->txrate,NL80211_STA_INFO_TX_BITRATE)
nl80211_put_sta_rate

.get_station = rwnx_cfg80211_get_station
rwnx_fill_station_info(sta, vif, sinfo)

```
static bool nl80211_put_sta_rate(struct sk_buff *msg, struct rate_info *info,
                                  int attr)
{
    struct nlattr *rate;
    u32 bitrate;
    u16 bitrate_compat;
    enum nl80211_rate_info rate_flg;

    rate = nla_nest_start(msg, attr);
    if (!rate)
        return false;

    /* cfg80211_calculate_bitrate will return 0 for mcs >= 32 */
    bitrate = cfg80211_calculate_bitrate(info);
    /* report 16-bit bitrate only if we can */
    bitrate_compat = bitrate < (1UL << 16) ? bitrate : 0;
    if (bitrate > 0 &&
        nla_put_u32(msg, NL80211_RATE_INFO_BITRATE32, bitrate))
        return false;
```

aic8800D
sinfo->txrate.legacy = tx_legrates_lut[rate_info->mcsIndexTx];
        tx_legrates_lut是U16的一个数组
        rate_info->mcsIndexTx 是U32
sinfo->filled |= (BIT(NL80211_STA_INFO_TX_BITRATE) | BIT(NL80211_STA_INFO_TX_FAILED));

RTK8723du
sinfo->filled |= STATION_INFO_TX_BITRATE;
sinfo->txrate.legacy = rtw_get_cur_max_rate(padapter) U16类型的
if (psta) sinfo->filled |= STATION_INFO_TX_FAILED;