

SH

hello i want you check below framework log

```
02-06 10:19:28.514 1711 5556 D WifiClientModelImpl[wlan0]:
getConativeWiFiQualityStatus keep previous WifiQuailityInfo
02-06 10:19:35.534 1711 2428 D ConnectivityService: [102 CELLULAR]
EVENT_NETWORK_INFO_CHANGED, going from CONNECTED to CONNECTED
02-06 10:19:35.535 1711 2428 D ConnectivityService: Adding iface epdg0 to
network 102
02-06 10:19:35.542 1711 2428 D ConnectivityService: Removing iface rmnet_data1
from network 102
02-06 10:19:35.564 1711 2428 E ConnectivityService: Exception in
networkRemoveRouteParcel: android.os.ServiceSpecificException: No such device
(code 19)

02-06 10:19:38.797 1711 2417 D WifiDataContinuityService:
isInternetCheckAvailable return false, reason isPoortNetworkEnabled is false
02-06 10:19:38.797 1711 2417 D WifiScoreReport: isDCFOffCondition 33
02-06 10:19:41.838 1711 2417 D WifiDataContinuityService:
isInternetCheckAvailable return false, reason isPoortNetworkEnabled is false
02-06 10:19:41.838 1711 2417 D WifiScoreReport: isDCFOffCondition 28
02-06 10:19:44.895 1711 2417 D WifiDataContinuityService:
isInternetCheckAvailable return false, reason isPoortNetworkEnabled is false
02-06 10:19:44.895 1711 2417 D WifiScoreReport: isDCFOffCondition 29

02-06 10:19:45.358 1711 2428 D ConnectivityService: [102 CELLULAR]
EVENT_NETWORK_INFO_CHANGED, going from CONNECTED to CONNECTED
02-06 10:19:45.358 1711 2428 D ConnectivityService: Adding iface rmnet_data2 to
network 102
02-06 10:19:45.363 1711 2428 D ConnectivityService: Removing iface epdg0 from
network 102
02-06 10:19:45.377 1711 2428 E ConnectivityService: Exception in
networkRemoveRouteParcel: android.os.ServiceSpecificException: No such device
(code 19)
```



This log is from an Android system and contains information about different events related to networking. Some key points from the log are:



Regenerate  
response

Generate  
PNG

Download  
PDF

Share  
Link



- The WifiClientModelImpl class is keeping a previous WifiQualityInfo.
- The ConnectivityService reports changes in network connectivity, such as adding and removing network interfaces (ifaces) from a network.
- The WifiDataContinuityService is checking if Internet check is available and returns false because the "isPoortNetworkEnabled" is false.
- The WifiScoreReport is reporting the value of "isDCFOffCondition".

SH

There are exceptions related to "No such device" in the ConnectivityService, which what is "ispoortnetworkenabled" indicate an error in removing a network route. what does it mean by false



"isPoortNetworkEnabled" is a variable in the code that determines whether a "poor network" is enabled or not. The value "false" indicates that the poor network feature is not enabled in the system. The exact meaning and purpose of the "poor network" feature can only be determined by looking at the code and context in which this variable is used.



SH

ok can you check below function and explain

```
public boolean isInternetCheckAvailable() {
    boolean isPoortNetworkEnabled;
    boolean isRoaming;
    boolean bRet = true;

    if (mDisableL3MonitoringTemporally) {
        Log.d(TAG, "isInternetCheckAvailable temporally disabled return false");
        return false;
    }
    isPoortNetworkEnabled =
(Settings.System.getInt(mContext.getContentResolver(),
        LgeServiceExtConstant.WIFI_DATA_CONTINUITY_ENABLED, 0) == 1);
    isRoaming = (mTelephonyManager != null &&
mTelephonyManager.isNetworkRoaming()) ? true : false;
    checkAndSetConnectivityInstance();
    if (isPoortNetworkEnabled == false) {
        boolean ignorePoorNetworkEnabled = false;
        if
(LGWiFiFeature.LGP_WIFI_STA_SKT_DCF_SWITCHING_POPUP_POLICY.isEnabled()) {
```

```
        // if wifi_multiple_connection_popup is 1, check wifi quality although dcf db
is off
        ignorePoorNetworkEnabled = showMultipleConnectionPopup();
    }
    if (ignorePoorNetworkEnabled == false) {
        Log.d(TAG, "isInternetCheckAvailable return false, reason
isPoortNetworkEnabled is false");
        return false;
    }
}

if (LGWiFiFeature.LGP_WIFI_STA_DCF.isEnabled() == false || isRoaming == true) {
    Log.d(TAG, "isInternetCheckAvailable return false");
    return false;
}

if (LGWiFiFeature.LGP_WIFI_STA_KDDI_DCF_ONLY_OPERATOR_AP.isEnabled()) {
    bRet = isCarrierSpecificAP();
}

if (bRet) {
    bRet = Do_CurrentWifiConfiguration_DCF_Test();
}
return bRet;
}
```



The method `isInternetCheckAvailable()` is a boolean function in an Android system that determines if an internet check is available or not. The function checks a number of conditions and variables to determine the availability of the internet check.

The function starts by checking whether the `mDisableL3MonitoringTemporally` variable is set to true, in which case it returns false, indicating that the internet check is not available.

Next, the function retrieves the value of "WIFI\_DATA\_CONTINUITY\_ENABLED" from the system settings and assigns it to the variable `isPoortNetworkEnabled`. If `isPoortNetworkEnabled` is false, it checks if the `showMultipleConnectionPopup` function returns true or false. If it returns false, the function returns false, indicating that the internet check is not available.

The function then checks if the value of ``LGWiFiFeature.LGP_WIFI_STA_DCF.isEnabled()`` is false or if the device is roaming, and returns false in either of those cases, indicating that the internet check is not available.

If ``LGWiFiFeature.LGP_WIFI_STA_KDDI_DCF_ONLY_OPERATOR_AP.isEnabled()`` is true, the function checks if the device is connected to a carrier-specific access point using the ``isCarrierSpecificAP`` function. If the check returns true, the function calls the ``Do_CurrentWifiConfiguration_DCF_Test`` function and returns its result.

Finally, the function returns ``bRet``, which indicates whether the internet check is available or not.



SH

what is WIFI\_DATA\_CONTINUITY\_ENABLED



WIFI\_DATA\_CONTINUITY\_ENABLED is a setting in the device's system that determines if the Wi-Fi data continuity feature is enabled or not. The setting is stored in the device's content resolver and is retrieved using

```
`Settings.System.getInt(mContext.getContentResolver(),
```

```
LgeServiceExtConstant.WIFI_DATA_CONTINUITY_ENABLED, 0)`.
```

 If the retrieved value is equal to 1, then the Wi-Fi data continuity feature is enabled. If the value is 0, then the feature is disabled.

SH

what is the meaning of below

02-06 13:27:27.531 1711 5536 D ConnectivityService: Returning BLOCKED NetworkInfo to uid=10223



The log message "02-06 13:27:27.531 1711 5536 D ConnectivityService: Returning BLOCKED NetworkInfo to uid=10223" indicates that the ConnectivityService has returned a BLOCKED NetworkInfo to the process with UID 10223. This means that the process with UID 10223 has requested information about the network connection status and the ConnectivityService has determined that the network is

currently BLOCKED, likely meaning that the network connection is unavailable or not functioning properly.



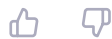
SH

Hi what is the meaning of below framework log

02-05 22:08:26.105 1711 2417 D WifiClientModelImpl[wlan0]: Network connection re established(L3 roaming), but it is same bssid



This log message is from the Wi-Fi module of an Android device and is indicating that the device has re-established a network connection (L3 roaming), but it is the same Basic Service Set Identifier (BSSID), which means the device is still connected to the same Wi-Fi network.



SH

what is l3 roaming?

what is the meaning of re-established a network connection (L3 roaming)?



L3 roaming refers to a type of roaming in wireless networking where the roaming device changes access points but remains within the same IP subnet.

The log message "Network connection re-established (L3 roaming), but it is same bssid" is indicating that the Wi-Fi client device has reconnected to the same wireless network access point (with the same BSSID), but the IP subnet has changed. This is considered L3 roaming.



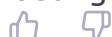
SH

can you tell me why this happens



L3 roaming typically occurs when a device switches from one Wi-Fi access point to another while still connected to the same network. This can happen when the device moves out of range of the current access point and into range of another access

point that provides better signal quality. The network connection is re-established to ensure that the device remains connected to the network. The "L3" in "L3 roaming" refers to the layer 3 of the OSI model, which is the network layer responsible for routing data between different networks.



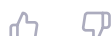
SH

ok what is meaning of below log

```
02-06 10:08:37.371 1711 5556 D ConnectivityService: Returning BLOCKED
NetworkInfo to uid=10229
```



The log message "Returning BLOCKED NetworkInfo to uid=10229" is indicating that the ConnectivityService is returning a BLOCKED status for the network information to the application with UID 10229. This means that the network connection is currently not available for use by the application. The specific reason for the network being blocked may not be specified in this log message and would need to be determined from the broader context of the logs or the code.



SH

what is menaing of below android framework 12 log

```
02-06 10:47:25.244 1711 2417 I WifiClientModelImpl: Default Gateway MAC address
of bc:5b:d5:d4:36:16 from routes is : bc:5b:d5:d4:36:10
02-06 10:47:54.425 1711 2428 D ConnectivityService: handleLingerComplete for
[112 CELLULAR]
```



The first log message shows that the WifiClientModelImpl has determined the MAC address of the default gateway based on the routing information, and it is recorded as "bc:5b:d5:d4:36:10".

The second log message shows that the ConnectivityService has completed processing for a lingering network type 112, which is CELLULAR. The "handleLingerComplete" message indicates that the service is no longer waiting for the connection to be re-established and the network type is now considered inactive.



explain below

```
02-06 11:08:39.614 1711 2859 D ConnectivityService: Returning BLOCKED
NetworkInfo to uid=10330
```

```
02-06 11:08:39.616 1711 2516 D ConnectivityService: requestNetwork for
uid/pid:10330/22834 activeRequest: null callbackRequest: 535 [NetworkRequest [
REQUEST id=536, [ Capabilities:
```

```
INTERNET&NOT_RESTRICTED&TRUSTED&NOT_VCN_MANAGED Uid: 10330
RequestorUid: 10330 RequestorPkg: com.google.android.apps.youtube.music
UnderlyingNetworks: Null] ]] callback flags: 0 order: 2147483647
```

```
02-06 11:08:39.621 1711 2428 D ConnectivityService: NetReassign [536 : null → 113]
[c 0] [a 1] [i 3]
```



The log messages you've posted are from the Android ConnectivityService, which is responsible for managing the device's network connections.

The log message "Returning BLOCKED NetworkInfo to uid=10330" indicates that the ConnectivityService has received a network request from an app with UID 10330, but the network request has been blocked.

The subsequent log message "requestNetwork for uid/pid:10330/22834 activeRequest: null callbackRequest: 535 [NetworkRequest [ REQUEST id=536, [ Capabilities: INTERNET&NOT\_RESTRICTED&TRUSTED&NOT\_VCN\_MANAGED Uid: 10330 RequestorUid: 10330 RequestorPkg: com.google.android.apps.youtube.music UnderlyingNetworks: Null] ]] callback flags: 0 order: 2147483647" shows the details of the network request made by the app with UID 10330. The log indicates that this is the first network request from this app, and the request is for a network with the capabilities of INTERNET, NOT\_RESTRICTED, TRUSTED, and NOT\_VCN\_MANAGED.

The final log message "NetReassign [536 : null → 113] [c 0] [a 1] [i 3]" indicates that the ConnectivityService has assigned the network request from the app with ID 536 to network 113. The numbers "c 0" and "a 1" indicate the number of calls and active requests, respectively, while "i 3" refers to the identity of the network assigned.



SH

can you check below logs

Ap2 validation passed to validation fail logs below.

02-06 10:47:24.826 1711 2428 D ConnectivityService: reportNetworkConnectivity(113, I  
02-06 10:47:25.192 1711 2428 D ConnectivityService: [113 WIFI] validation passed  
02-06 10:47:25.195 1711 2428 D ConnectivityService: NetReassign [no changes] [c 1] [a  
02-06 10:47:25.213 1711 2428 D ConnectivityService: Setting DNS servers for network 1  
02-06 10:47:25.213 1711 2428 D DnsManager: sendDnsConfigurationForNetwork(113, [I  
02-06 10:47:25.244 1711 2417 I WifiClientModelImpl: Default Gateway MAC address of t  
02-06 10:47:25.977 1711 22273 D WifiInternetCheck: runDnsCheck() - dnsResultValue: t  
02-06 10:47:26.024 1711 3912 V WifiInternetCheck: isHttpConnectionAvailable - We got  
02-06 10:47:54.425 1711 2428 D ConnectivityService: handleLingerComplete for [112 C  
02-06 10:58:22.020 1711 2417 D WifiActiveModeWarden: Switching all client mode man  
  
02-06 11:22:40.140 1711 2417 D WifiClientModelImpl: isPingSuccessToGwServer:false  
02-06 11:22:40.141 1711 2417 D WifiClientModelImpl: Roaming, Gw address is changed,  
02-06 11:22:40.142 1711 2428 D ConnectivityService: Removing iface wlan0 from netwo  
02-06 11:22:40.142 1711 2417 D WifiClientModelImpl[wlan0]: ClientModelImpl: Leaving (I  
02-06 11:22:40.168 1711 23625 D WifiClientModelImpl: isPingSuccessToGwServer:false/I  
02-06 11:22:40.175 1711 2428 D ConnectivityService: Invalid tcpBufferSizes string: null,  
02-06 11:22:40.178 1711 2428 E ConnectivityService: Exception in networkRemoveRout  
02-06 11:22:40.182 1711 2428 D ConnectivityService: Setting DNS servers for network 1



02-06 11:22:40.182 1711 2428 D DnsManager: sendDnsConfigurationForNetwork(113, [

02-06 11:22:40.224 1711 2428 E ConnectivityService: BUG: NetworkAgentInfo{network|  
created=2023-02-06T18:47:23.025Z Score(Policies :  
TRANSPORT\_PRIMARY&EVER\_EVALUATED&IS\_UNMETERED&ACCEPT\_UNVALIDATED&E  
163433999 firstValidated 163434802 lastValidated 163434802 explicitlySelected accept  
null MTU: 0 Routes: [ ]}} nc{[ Transports: WIFI Capabilities:  
NOT\_METERED&INTERNET&NOT\_RESTRICTED&TRUSTED&NOT\_VPN&VALIDATED&NOT  
LinkUpBandwidth>=12000Kbps LinkDnBandwidth>=30000Kbps Specifier: <WifiNetwork  
mMatchLocalOnlySpecifiers=false]> TransportInfo: <SSID: "ZiPLY-3610", BSSID: 00:00:00:(  
ASSOCIATED, Wi-Fi standard: 4, RSSI: -72, Link speed: 390Mbps, Tx Link speed: 390Mbps,  
speed: 144Mbps, Frequency: 5500MHz, Net ID: 3, Metered hint: false, score: 55, isUsable:  
Ephemeral: false, OEM paid: false, OEM private: false, OSU AP: false, FQDN: <none>, Pro  
pskMLO Information: , AP MLD Address: <none>, AP MLO Link Id: <none>, AP MLO Affilia  
"ZiPLY-3610" UnderlyingNetworks: Null]] factorySerialNumber=7} lost immutable capabi  
3610", BSSID=bc:5b:d5:d4:36:16, band=2, mMatchLocalOnlySpecifiers=false] -> WifiNetw  
BSSID=bc:5b:d5:d4:36:12, band=2, mMatchLocalOnlySpecifiers=false]

02-06 11:22:40.235 1711 2428 D ConnectivityService: Adding iface wlan0 to network 11  
02-06 11:22:40.244 1711 2428 D ConnectivityService: Setting DNS servers for network 1  
02-06 11:22:40.244 1711 2428 D DnsManager: sendDnsConfigurationForNetwork(113, [  
02-06 11:22:40.262 1711 2428 D ConnectivityService: reportNetworkConnectivity(113, l  
02-06 11:22:40.262 1711 2428 D ConnectivityService: Invalid tcpBufferSizes string: null,  
02-06 11:22:40.269 1711 2428 E ConnectivityService: Exception in networkRemoveRout  
02-06 11:22:40.270 1711 2428 D ConnectivityService: Setting DNS servers for network 1  
02-06 11:22:40.270 1711 2428 D DnsManager: sendDnsConfigurationForNetwork(113, [  
02-06 11:22:40.364 1711 2428 D ConnectivityService: [113 WIFI] validation failed  
02-06 11:22:41.985 1711 2417 E WifiClientModelImpl: Network connection candidate wi  
02-06 11:22:42.024 1711 2428 D DnsManager: sendDnsConfigurationForNetwork(113, [  
02-06 11:22:42.037 1711 2428 D ConnectivityService: [113 WIFI] EVENT\_NETWORK\_INF

02-06 11:22:42.050 1711 2496 E WifiWatchdogStateMachine: transitionTo called while t  
com.android.server.wifi.WifiWatchdogStateMachine\$OnlineWatchState@1e332f0, new l  
02-06 11:22:42.051 1711 2428 E ConnectivityService: BUG: NetworkAgentInfo{network|  
created=2023-02-06T18:47:23.025Z Score(Policies :  
TRANSPORT\_PRIMARY&EVER\_EVALUATED&IS\_UNMETERED&ACCEPT\_UNVALIDATED&E  
firstValidated 163434802 explicitlySelected acceptUnvalidated acceptPartialConnectivit  
fe80::b87f:b54d:c40d:1410/64,192.168.1.57/24 ] DnsAddresses: [ /192.168.1.1 ] Domains

```
524288,1048576,2097152,262144,524288,1048576 Routes: [ fe80::/64 -> :: wlan0 mtu 0,
Transports: WIFI Capabilities:
NOT_METERED&INTERNET&NOT_RESTRICTED&TRUSTED&NOT_VPN&NOT_ROAMING&
LinkUpBandwidth>=12000Kbps LinkDnBandwidth>=30000Kbps Specifier: <WifiNetwork
band=2, mMatchLocalOnlySpecifiers=false]> TransportInfo: <SSID: "ZiPLY-3610", BSSID: b
state: COMPLETED, Wi-Fi standard: 4, RSSI: -72, Link speed: 390Mbps, Tx Link speed: 390
Rx Link speed: 144Mbps, Frequency: 5500MHz, Net ID: 3, Metered hint: false, score: 55, is
Restricted: false, Ephemeral: false, OEM paid: false, OEM private: false, OSU AP: false, FC
3610" wpa2-psk MLO Information: , AP MLD Address: <none>, AP MLO Link Id: <none>, AP
SSID: "ZiPLY-3610" UnderlyingNetworks: Null]] factorySerialNumber=7} lost immutable c
SSID="Omon_ra iphone ", BSSID=bc:5b:d5:d4:36:12, band=2, mMatchLocalOnlySpecifiers
BSSID=bc:5b:d5:d4:36:12, band=1, mMatchLocalOnlySpecifiers=false]
```

---

The logs suggest that a WIFI network with SSID "ZiPLY-3610" and BSSID "bc:5b:d5:d4:36:16" was connected and validated on 02-06 10:47:25, with DNS servers set to "/10.17.254.254". The network was later removed due to a failure in the routing configuration at 11:22:40. The logs indicate that the device failed to remove a route and threw a "No such device (code 19)" error. Additionally, the logs also mention that the internet connectivity was checked and that the device received a valid response of 204 at 10:47:54. However, later at 11:22:40, the device failed to successfully ping to the gateway server.



SH

what could be the problem



From the logs provided, it appears that the WIFI connectivity has encountered some issues. Some possible problems based on the logs include:

- Wifi gateway address has changed which triggered the client to leave connected state (11:22:40.142)
- The device is unable to connect to the internet as the response from `isHttpConnectionAvailable` was 204 (not a success response) (10:47:26.024)
- The device failed to ping to the gateway server (11:22:40.140 and 11:22:40.168)
- The network removal has resulted in an exception "No such device" (11:22:40.178)

Without more context or further logs, it is not possible to determine the exact cause of the problem.

SH

can tell about the ConnectivityService: BUG ... more



The message "ConnectivityService: BUG Binder call to networkStack failed" is an error message from the Android operating system that indicates a problem with the ConnectivityService component in the Android framework. This message is often seen in Android system logs and indicates that a call from the ConnectivityService component to the networkStack component has failed.

This error can have several causes, including issues with the device's network stack, hardware problems with the device's network interface, or problems with the ConnectivityService component itself. In some cases, this error may be related to a specific app or service that is running on the device, while in others it may be related to a broader problem with the Android operating system.

If you are encountering this error, it is recommended that you restart your device and see if the problem resolves itself. If the error persists, you may want to try clearing the cache and data for the ConnectivityService and networkStack components, or consider resetting your device to its factory settings. If the problem continues to persist, you may want to seek further technical assistance from your device manufacturer or a professional repair service.



SH

i want to understand below section of dump from framework

DPP config:

IP config:

IP assignment: DHCP

Proxy settings: NONE

cuid=1000 cname=android.uid.system:1000 luid=1000

lname=android.uid.system:1000 lcuid=1000 allowAutojoin=true

noInternetAccessExpected=false mostRecentlyConnected=false

recentFailure: Association Rejection code: 0

ShareThisAp: false

wifi id: 1

, 1=ID: 1 SSID: "ShawMobileHotspot" PROVIDER-NAME: null BSSID: null FQDN: null

HOME-PROVIDER-NETWORK: false PRIO: 0 HIDDEN: true PMF: falseCarrierId: -1 SSID

Bridging - auto connection enabled: false  
NetworkSelectionStatus NETWORK\_SELECTION\_ENABLED  
hasEverConnected: true  
validatedInternetAccess trusted  
macRandomizationSetting: 0  
mRandomizedMacAddress: 12:c1:fc:3a:1d:f1  
randomizedMacExpirationTimeMs: <none>  
KeyMgmt: WPA\_EAP IEEE8021X Protocols: WPA RSN  
AuthAlgorithms:  
PairwiseCiphers: TKIP CCMP  
GroupCiphers: TKIP CCMP  
GroupMgmtCiphers: BIP\_GMAC\_256  
SuiteBCiphers: ECDHE\_ECDSA  
PSK/SAE:  
Enterprise config:  
subject\_match NULL  
altsubject\_match NULL  
proactive\_key\_caching 1  
client\_cert NULL  
key\_id NULL  
wapi\_cert\_suite NULL  
plmn NULL  
domain\_suffix\_match NULL  
anonymous\_identity  
"2AAUTPnm031ug/PbcO8Cwkf@wlan.mnc490.mcc302.3gppnetwork.org"  
password <removed>  
engine NULL  
engine\_id NULL  
identity "0302490215424499@wlan.mnc490.mcc302.3gppnetwork.org"  
ca\_path NULL  
realm NULL  
ca\_cert NULL  
eap\_method: AKA  
ocsp: 0



This section of the dump is related to the configuration of a Wi-Fi network on an Android device. The Wi-Fi network has the SSID "ShawMobileHotspot". The IP address is obtained via DHCP, and the proxy settings are not set. The Wi-Fi network is saved on the device, and it has ever connected to it in the past. The Wi-Fi network uses the WPA\_EAP (Wi-Fi Protected Access Enterprise) security protocol and requires

authentication using the AKA (Authentication and Key Agreement) method. The identity used for authentication is

"0302490215424499@wlan.mnc490.mcc302.3gppnetwork.org", and the password is hidden.



SH

what is this DPP config,when is it used



DPP stands for Device Provisioning Protocol. It is a standards-based protocol used to provision and configure network devices, such as access points, routers, and other IoT devices, in an automated and secure manner.

In this particular context, the "DPP config" section of the dump from the framework may refer to the configuration settings for a DPP-enabled device, such as the type of IP assignment (DHCP), the proxy settings (NONE), the connection details (SSID, BSSID, KeyMgmt, etc.), and the enterprise configuration (subject\_match, anonymous\_identity, password, eap\_method, etc.).

The DPP config is typically used in networks where large numbers of devices need to be provisioned quickly, securely, and consistently, such as in large-scale enterprise networks, service provider networks, or industrial IoT applications.

