

SH

Dear gpt,

please see below logs related to wifi aware test->data path (oob):open responder  
i believe it is network request failure-timed out.

can you give me more info why this failed from below logs

logs:-

```
02-13 14:09:54.709 14248 14248 D wpa_supplicant: nl80211: Regulatory information
- country=KR
02-13 14:09:54.709 14248 14248 D wpa_supplicant: nl80211: 2402-2482 @ 40 MHz 20
mBm
02-13 14:09:54.709 14248 14248 D wpa_supplicant: nl80211: 5170-5250 @ 80 MHz 20
mBm
02-13 14:09:54.709 14248 14248 D wpa_supplicant: nl80211: 5250-5330 @ 80 MHz 23
mBm (DFS)
02-13 14:09:54.709 14248 14248 D wpa_supplicant: nl80211: 5490-5730 @ 160 MHz
24 mBm (DFS)
02-13 14:09:54.709 14248 14248 D wpa_supplicant: nl80211: 5735-5835 @ 80 MHz 30
mBm
02-13 14:09:54.709 14248 14248 D wpa_supplicant: nl80211: Added 802.11b mode
based on 802.11g information
02-13 14:09:54.709 14248 14248 D wpa_supplicant: nl80211: Mode IEEE 802.11g:
2412 2417 2422 2427 2432 2437 2442 2447 2452 2457 2462 2467 2472
2484[DISABLED]
02-13 14:09:54.709 14248 14248 D wpa_supplicant: nl80211: Mode IEEE 802.11a:
5180 5200 5220 5240 5260[RADAR] 5280[RADAR] 5300[RADAR] 5320[RADAR]
5500[RADAR] 5520[RADAR] 5540[RADAR] 5560[RADAR] 5580[RADAR] 5600[RADAR]
5620[RADAR] 5640[RADAR] 5660[RADAR] 5680[RADAR]
02-13 14:09:54.709 14248 14248 D wpa_supplicant: nl80211: Mode IEEE 802.11b:
2412 2417 2422 2427 2432 2437 2442 2447 2452 2457 2462 2467 2472
2484[DISABLED]
02-13 14:09:54.710 1674 2046 D WifiVendorHal: onIfaceRemoved 3, name: wifi-
aware0
02-13 14:09:54.710 1674 2474 D HalDevMgr: onIfaceRemoved: type=3, name=wifi-
aware0
02-13 14:09:54.710 1114 14243 I WifiHAL : event received NL80211_CMD_VENDOR,
vendor_id = 0x1374, subcmd = 0xc
02-13 14:09:54.710 1114 14243 V WifiHAL : handleEvent: Vendor event:
vendor_id=0x1374 subcmd=12
```



Regenerate  
response

Generate  
PNG

Download  
PDF

Share  
Link



```
02-13 14:09:54.710 1114 14243 V WifiHAL : 10 00 02 00 91 01 0c 00
02-13 14:09:54.710 1114 14243 V WifiHAL : 00 00 30 02 00 00 00 00
02-13 14:09:54.710 1114 14243 V WifiHAL : *****HexDump End*****
02-13 14:09:54.711 1114 14243 V WifiHAL : handleEvent: Subcmd=12 Vendor data
len received:16
02-13 14:09:54.711 1114 14243 V WifiHAL : *****HexDump len:16*****
02-13 14:09:54.711 1114 14243 V WifiHAL : 10 00 02 00 91 01 0c 00
02-13 14:09:54.711 1114 14243 V WifiHAL : 00 00 30 02 00 00 00 00
02-13 14:09:54.711 1114 14243 V WifiHAL : *****HexDump End*****
02-13 14:09:54.711 1114 14243 V WifiHAL : handleNanResponse called
0xb4000075f2a8c850
02-13 14:09:54.711 1114 14243 D WifiHAL : NanErrorTranslation: Status: 0 Error
Info[value 0]: NAN status success
02-13 14:09:54.711 1674 2523 D HalDevMgr: Current RttController is valid, Don't try
to create a new one
02-13 14:09:54.711 1114 14243 I WifiHAL : handleNanResponse ret:0 status:0
value:NAN status success response_type:1
02-13 14:09:54.711 1659 1659 I wificond: 2.4Ghz frequencies: 2412 2417 2422 2427
2432 2437 2442 2447 2452 2457 2462 2467 2472
02-13 14:09:54.711 1659 1659 I wificond: 5Ghz non-DFS frequencies: 5180 5200
5220 5240 5745 5765 5785 5805 5825
02-13 14:09:54.711 1659 1659 I wificond: 5Ghz DFS frequencies: 5260 5280 5300
5320 5500 5520 5540 5560 5580 5600 5620 5640 5660 5680 5700 5720
02-13 14:09:54.711 1659 1659 I wificond: 6Ghz frequencies:
02-13 14:09:54.711 1659 1659 I wificond: 60Ghz frequencies:
02-13 14:09:54.712 1114 14243 E android.hardware.wifi@1.0-service: Callback
invoked on an invalid object
02-13 14:09:54.712 1114 14243 I WifiHAL : event received NL80211_CMD_VENDOR,
vendor_id = 0x1374, subcmd = 0xc
02-13 14:09:54.712 1114 14243 V WifiHAL : handleEvent: Vendor event:
vendor_id=0x1374 subcmd=12
02-13 14:09:54.712 1114 14243 V WifiHAL : handleEvent: Vendor data len
received:16
02-13 14:09:54.712 1114 14243 V WifiHAL : *****HexDump len:16*****
02-13 14:09:54.712 1114 14243 V WifiHAL : 10 00 02 00 a1 01 0c 00
02-13 14:09:54.712 1114 14243 V WifiHAL : 00 00 d8 01 00 00 00 00
02-13 14:09:54.712 1114 14243 V WifiHAL : *****HexDump End*****
02-13 14:09:54.712 1114 14243 V WifiHAL : handleEvent: Subcmd=12 Vendor data
len received:16
02-13 14:09:54.712 1114 14243 V WifiHAL : *****HexDump len:16*****
02-13 14:09:54.712 1114 14243 V WifiHAL : 10 00 02 00 a1 01 0c 00
02-13 14:09:54.712 1114 14243 V WifiHAL : 00 00 d8 01 00 00 00 00
```

```
02-13 14:09:54.712 1114 14243 V WifiHAL : *****HexDump End*****
02-13 14:09:54.712 1114 14243 V WifiHAL : handleNanIndication msg_id:7
02-13 14:09:54.712 1114 14243 D WifiHAL : NanErrorTranslation: Status: 0 Error
Info[value 0]: NAN status success
02-13 14:09:54.712 1114 14243 E android.hardware.wifi@1.0-service: Callback
invoked on an invalid object
02-13 14:09:54.713 1114 14243 I WifiHAL : event received NL80211_CMD_VENDOR,
vendor_id = 0x1374, subcmd = 0xc
02-13 14:09:54.713 1114 14243 V WifiHAL : handleEvent: Vendor event:
vendor_id=0x1374 subcmd=12
02-13 14:09:54.713 1114 14243 V WifiHAL : handleEvent: Vendor data len
received:16
02-13 14:09:54.713 1114 14243 V WifiHAL : *****HexDump len:16*****
02-13 14:09:54.713 1114 14243 V WifiHAL : 10 00 02 00 91 01 0c 00
02-13 14:09:54.713 1114 14243 V WifiHAL : 00 00 ff ff 02 00 00 00
02-13 14:09:54.713 1114 14243 V WifiHAL : *****HexDump End*****
02-13 14:09:54.713 1114 14243 V WifiHAL : handleEvent: Subcmd=12 Vendor data
len received:16
02-13 14:09:54.713 1114 14243 V WifiHAL : *****HexDump len:16*****
02-13 14:09:54.713 1114 14243 V WifiHAL : 10 00 02 00 91 01 0c 00
02-13 14:09:54.713 1114 14243 V WifiHAL : 00 00 ff ff 02 00 00 00
02-13 14:09:54.713 1114 14243 V WifiHAL : *****HexDump End*****
02-13 14:09:54.713 1114 14243 V WifiHAL : handleNanResponse called
0xb4000075f2a8c850
02-13 14:09:54.713 1114 14243 D WifiHAL : NanErrorTranslation: Status: 1 Error
Info[value 0]: NAN Discovery engine failure
02-13 14:09:54.713 1114 14243 I WifiHAL : handleNanResponse ret:0 status:1
value:NAN Discovery engine failure response_type:1
02-13 14:09:54.713 1114 14243 E android.hardware.wifi@1.0-service: Callback
invoked on an invalid object
```



The logs indicate a Wi-Fi Aware failure during a data path (OOB) request. "OOB" stands for "Out of Band", which means the data is being transmitted outside the normal data channel.

The logs show that the regulatory information for the country (South Korea) is being loaded into wpa\_supplicant, which is a Wi-Fi client utility that implements the core logic of the Wi-Fi Protected Access (WPA) protocol. The logs also show the available Wi-Fi modes (IEEE 802.11a, IEEE 802.11b, and IEEE 802.11g), frequency ranges, and maximum transmit power levels.

However, it seems that the Wi-Fi Aware interface was removed, as indicated by the following log message: "Interface was destroyed:

mWifiNanIface=android.hardware.wifi@1.4::IWifiNanIface@Proxy, active=false."

Finally, a vendor event with a subcommand value of 12 was received and processed by the Wi-Fi HAL, which is the Hardware Abstraction Layer for Wi-Fi. The vendor data received has a length of 16 and contains binary data (represented in hexadecimal form). However, without more context or information about the system or application being used, it is difficult to determine the exact cause of the network request failure.

Overall, it seems that the failure may have been caused by a problem with the Wi-Fi Aware interface or some other issue related to the Wi-Fi connectivity.



SH

ok see more logs below:-

```
02-13 14:09:54.711 1114 14243 D WifiHAL : NanErrorTranslation: Status: 0 Error Info[value 0]: NAN status success
```

```
02-13 14:09:54.711 1674 2523 D HalDevMgr: Current RttController is valid, Don't try to create a new one
```

```
02-13 14:09:54.711 1114 14243 I WifiHAL : handleNanResponse ret:0 status:0 value:NAN status success response_type:1
```

```
02-13 14:09:54.711 1659 1659 I wificond: 2.4Ghz frequencies: 2412 2417 2422 2427 2432 2437 2442 2447 2452 2457 2462 2467 2472
```

```
02-13 14:09:54.711 1659 1659 I wificond: 5Ghz non-DFS frequencies: 5180 5200 5220 5240 5745 5765 5785 5805 5825
```

```
02-13 14:09:54.711 1659 1659 I wificond: 5Ghz DFS frequencies: 5260 5280 5300 5320 5500 5520 5540 5560 5580 5600 5620 5640 5660 5680 5700 5720
```

```
02-13 14:09:54.711 1659 1659 I wificond: 6Ghz frequencies:
```

```
02-13 14:09:54.711 1659 1659 I wificond: 60Ghz frequencies:
```

```
02-13 14:09:54.712 1114 14243 E android.hardware.wifi@1.0-service: Callback invoked on an invalid object
```

```
02-13 14:09:54.712 1114 14243 I WifiHAL : event received NL80211_CMD_VENDOR, vendor_id = 0x1374, subcmd = 0xc
```

```
02-13 14:09:54.712 1114 14243 V WifiHAL : handleEvent: Vendor event: vendor_id=0x1374 subcmd=12
```

```
02-13 14:09:54.712 1114 14243 V WifiHAL : handleEvent: Vendor data len received:16
```

```
02-13 14:09:54.712 1114 14243 V WifiHAL : *****HexDump len:16*****
```

```
02-13 14:09:54.712 1114 14243 V WifiHAL : 10 00 02 00 a1 01 0c 00
```

```
02-13 14:09:54.712 1114 14243 V WifiHAL : 00 00 d8 01 00 00 00 00
02-13 14:09:54.712 1114 14243 V WifiHAL : *****HexDump End*****
02-13 14:09:54.712 1114 14243 V WifiHAL : handleEvent: Subcmd=12 Vendor data
len received:16
02-13 14:09:54.712 1114 14243 V WifiHAL : *****HexDump len:16*****
02-13 14:09:54.712 1114 14243 V WifiHAL : 10 00 02 00 a1 01 0c 00
02-13 14:09:54.712 1114 14243 V WifiHAL : 00 00 d8 01 00 00 00 00
02-13 14:09:54.712 1114 14243 V WifiHAL : *****HexDump End*****
02-13 14:09:54.712 1114 14243 V WifiHAL : handleNanIndication msg_id:7
02-13 14:09:54.712 1114 14243 D WifiHAL : NanErrorTranslation: Status: 0 Error
Info[value 0]: NAN status success
02-13 14:09:54.712 1114 14243 E android.hardware.wifi@1.0-service: Callback
invoked on an invalid object
02-13 14:09:54.713 1114 14243 I WifiHAL : event received NL80211_CMD_VENDOR,
vendor_id = 0x1374, subcmd = 0xc
02-13 14:09:54.713 1114 14243 V WifiHAL : handleEvent: Vendor event:
vendor_id=0x1374 subcmd=12
02-13 14:09:54.713 1114 14243 V WifiHAL : handleEvent: Vendor data len
received:16
02-13 14:09:54.713 1114 14243 V WifiHAL : *****HexDump len:16*****
02-13 14:09:54.713 1114 14243 V WifiHAL : 10 00 02 00 91 01 0c 00
02-13 14:09:54.713 1114 14243 V WifiHAL : 00 00 ff ff 02 00 00 00
02-13 14:09:54.713 1114 14243 V WifiHAL : *****HexDump End*****
02-13 14:09:54.713 1114 14243 V WifiHAL : handleEvent: Subcmd=12 Vendor data
len received:16
02-13 14:09:54.713 1114 14243 V WifiHAL : *****HexDump len:16*****
02-13 14:09:54.713 1114 14243 V WifiHAL : 10 00 02 00 91 01 0c 00
02-13 14:09:54.713 1114 14243 V WifiHAL : 00 00 ff ff 02 00 00 00
02-13 14:09:54.713 1114 14243 V WifiHAL : *****HexDump End*****
02-13 14:09:54.713 1114 14243 V WifiHAL : handleNanResponse called
0xb4000075f2a8c850
02-13 14:09:54.713 1114 14243 D WifiHAL : NanErrorTranslation: Status: 1 Error
Info[value 0]: NAN Discovery engine failure
02-13 14:09:54.713 1114 14243 I WifiHAL : handleNanResponse ret:0 status:1
value:NAN Discovery engine failure response_type:1
02-13 14:09:54.713 1114 14243 E android.hardware.wifi@1.0-service: Callback
invoked on an invalid object
```



The logs you provided seem to be related to the Wi-Fi functionality of an Android device. The log messages indicate that the Wi-Fi hardware abstraction layer (HAL) is

processing various Wi-Fi related events and messages, such as vendor-specific events, and handling NAN (Neighbor Awareness Networking) responses.

There are some log messages that indicate errors, such as "Callback invoked on an invalid object" and "WifiHAL : event received NL80211\_CMD\_VENDOR, vendor\_id = 0x1374, subcmd = 0xc." These errors may indicate that something is not working as expected with the Wi-Fi functionality on the device.

It would be best to seek assistance from the device manufacturer or a technical support professional to further diagnose and resolve any issues that might be present.



SH

ok see more below and try to make some sense

```
02-13 14:09:54.669 1114 1114 E WifiHAL : ack_handler_nan: called
02-13 14:09:54.669 14248 14248 D wpa_supplicant: nl80211: Mode IEEE 802.11a:
5180 5200 5220 5240 5260[RADAR] 5280[RADAR] 5300[RADAR] 5320[RADAR]
5500[RADAR] 5520[RADAR] 5540[RADAR] 5560[RADAR] 5580[RADAR] 5600[RADAR]
5620[RADAR] 5640[RADAR] 5660[RADAR] 5680[RADAR]
02-13 14:09:54.669 1114 1114 V WifiHAL : NanCommand 0xb4000075f2a8da30
destroyed
02-13 14:09:54.669 14248 14248 D wpa_supplicant: nl80211: Mode IEEE 802.11b:
2412 2417 2422 2427 2432 2437 2442 2447 2452 2457 2462 2467 2472
2484[DISABLED]
02-13 14:09:54.669 1114 1114 V WifiHAL : NAN_DP_INTERFACE_DELETE
02-13 14:09:54.669 1114 1114 V WifiHAL : WifiVendorCommand
0xb4000075f2a8da30 created vendor_id:0x1374 subcmd:81
02-13 14:09:54.670 1114 1114 V WifiHAL :
requestEvent:Handle:0xb400007632a95150 Socket Value:0xb400007572a8d880
02-13 14:09:54.671 1114 1114 E WifiHAL : error_handler_nan: Error code:-22 (Invalid
argument)
02-13 14:09:54.671 1114 1114 E WifiHAL : nan_data_interface_delete: requestEvent
Error:-5
02-13 14:09:54.671 1114 1114 V WifiHAL : NanCommand 0xb4000075f2a8da30
destroyed
02-13 14:09:54.671 1659 1659 I wificond: 2.4Ghz frequencies: 2412 2417 2422 2427
2432 2437 2442 2447 2452 2457 2462 2467 2472
02-13 14:09:54.671 1114 1114 V WifiHAL : NAN_DP_INTERFACE_DELETE
02-13 14:09:54.671 1114 1114 V WifiHAL : WifiVendorCommand
0xb4000075f2a8da30 created vendor_id:0x1374 subcmd:81
02-13 14:09:54.673 1114 1114 V WifiHAL :
```

```
requestEvent:Handle:0xb400007632a95150 Socket Value:0xb400007572a8d880
02-13 14:09:54.673 1114 1114 E WifiHAL : error_handler_nan: Error code:-22 (Invalid
argument)
02-13 14:09:54.673 1114 1114 E WifiHAL : nan_data_interface_delete: requestEvent
Error:-5
02-13 14:09:54.673 1114 1114 V WifiHAL : NanCommand 0xb4000075f2a8da30
destroyed
02-13 14:09:54.706 1674 2439 D WifiNative: interfaceLinkStateChanged:
ifaceName=wifi-aware0, mInterfaceId = 0, mIfaceMgr={0=Iface:
{Name=wlan0,Id=0,Type=STA_CONNECTIVITY}}
02-13 14:09:54.706 1674 2439 V WifiNative: Received iface link up/down notification
on an invalid iface=wifi-aware0
```



The log entries you've provided are from the Android operating system and are related to the Wi-Fi functionality of a device.

The log mentions the following:

- "WifiHAL" (Wi-Fi Hardware Abstraction Layer) events, including calls to the `ack_handler_nan` and `error_handler_nan` functions, as well as the creation and destruction of "NanCommand" objects.
- "wpa\_supplicant" events, including the detection of available 802.11a and 802.11b Wi-Fi modes.
- "wificond" events, including the listing of 2.4Ghz frequencies.
- "WifiNative" events, including the notification of an interface link state change for a Wi-Fi interface named "wifi-aware0." The log indicates that this change is not valid because the "wifi-aware0" interface is not recognized by the device.

In summary, these logs show the Wi-Fi-related activities of the device, including the detection of Wi-Fi modes and frequencies, the creation and destruction of Wi-Fi-related objects, and notifications of Wi-Fi interface state changes.



SH

```
02-13 14:09:54.586 2554 2671 D [SystemUI]NetworkController:
[NetworkControllerImpl.java:2242:refreshCarrierLabel()]carrierlabel = LG U+(false)
02-13 14:09:54.590 18475 20075 D ConnectivityManager: NetworkCallback was
already unregistered
02-13 14:09:54.591 18475 20075 E DataPathOutOfBandTestCase:
```



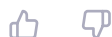
```
executeTestResponder: network request rejected - ON_UNAVAILABLE
02-13 14:09:54.591 1674 3341 V WifiAwareService: disconnect: uid=10342,
clientId=41
02-13 14:09:54.592 1674 2523 V WifiAwareDataPathStMgr: deleteAllInterfaces
02-13 14:09:54.592 1674 2523 V WifiAwareDataPathStMgr:
onAwareDownCleanupDataPaths
02-13 14:09:54.592 1674 2523 V WifiAwareNativeApi:
deleteAwareNetworkInterface: transactionId=559, interfaceName=aware_data0
02-13 14:09:54.593 1114 1114 V WifiHAL : NAN_DP_INTERFACE_DELETE
```



This log output appears to be from the system logs of an Android device. The logs show various events and status updates from different components of the system, such as the SystemUI, ConnectivityManager, and WifiAwareService.

- The first log message shows the status of the carrier label, indicating that the carrier is "LG U+" and the value of "false".
- The next few log messages show that a NetworkCallback was already unregistered, and a DataPathOutOfBandTestCase encountered an error because the network request was rejected as unavailable.
- The subsequent log messages show that the WifiAwareService is disconnecting, with a client ID of 41, and that the WifiAwareDataPathStMgr is cleaning up its data paths.
- The final log message shows a WifiHAL event, indicating that a NAN (Neighbor Awareness Networking) Data Path interface has been deleted.

Overall, this log output provides a snapshot of various events and updates related to network connectivity and the Wifi service on an Android device.



SH

hello check below logs and function.tell what is causing problem.

```
02-13 14:44:11.588 2005 9478 V WifiAwareService: connect: uid=10341, clientId=1,
configRequestConfigRequest [mSupport5gBand=true, mSupport6gBand=false,
mMasterPreference=0, mClusterLow=0, mClusterHigh=65535,
mDiscoveryWindowInterval=[-1, -1, -1]], notifyOnIdentityChanged=false
02-13 14:44:11.591 2005 6260 V WifiAwareStateManager: mergeConfigRequests():
mClients=[{}], configRequest=ConfigRequest [mSupport5gBand=true,
mSupport6gBand=false, mMasterPreference=0, mClusterLow=0,
```



```
mClusterHigh=65535, mDiscoveryWindowInterval=[-1, -1, -1]]
```

```
02-13 14:44:11.591 2005 6260 D WifiAwareNativeManager: tryToGetAware:
```

```
mWifiNativeInterface=null, mReferenceCount=0, requestorWs=WorkSource{10341  
com.android.cts.verifier}
```

```
02-13 14:44:11.603 2005 6260 E HalDevMgr: createInterfacePossible: Failed to create  
interface for interfaceType=4, requestorWs=WorkSource{10341 com.android.cts.verifier}
```

```
@Override
```

```
public void connect(final IBinder binder, String callingPackage, String  
callingFeatureId,
```

```
    IWifiAwareEventCallback callback, ConfigRequest configRequest,  
    boolean notifyOnIdentityChanged, Bundle extras) {  
    enforceAccessPermission();  
    enforceChangePermission();
```

```
    final int uid = getMockableCallingUid();  
    mWifiPermissionsUtil.checkPackage(uid, callingPackage);
```

```
    if (callback == null) {  
        throw new IllegalArgumentException("Callback must not be null");  
    }
```

```
    if (binder == null) {  
        throw new IllegalArgumentException("Binder must not be null");  
    }
```

```
    if (extras == null) {  
        throw new IllegalArgumentException("extras bundle must not be null");  
    }
```

```
    if (notifyOnIdentityChanged) {  
        enforceNearbyOrLocationPermission(callingPackage, callingFeatureId,  
            getMockableCallingUid(), extras, "Wifi Aware attach");  
    }
```

```
    if (configRequest != null) {  
        enforceNetworkStackPermission();  
    } else {  
        configRequest = new ConfigRequest.Builder().build();  
    }  
    configRequest.validate();
```

```
int pid = getCallingPid();

final int clientId;
synchronized (mLock) {
    clientId = mNextClientId++;
}

if (mDbg) {
    Log.v(TAG, "connect: uid=" + uid + ", clientId=" + clientId + ", configRequest"
        + configRequest + ", notifyOnIdentityChanged=" +
notifyOnIdentityChanged);
}

IBinder.DeathRecipient dr = new IBinder.DeathRecipient() {
    @Override
    public void binderDied() {
        if (mDbg) Log.v(TAG, "binderDied: clientId=" + clientId);
        binder.unlinkToDeath(this, 0);

        synchronized (mLock) {
            mDeathRecipientsByClientId.delete(clientId);
            mUidByClientId.delete(clientId);
        }

        mStateManager.disconnect(clientId);
    }
};

try {
    binder.linkToDeath(dr, 0);
} catch (RemoteException e) {
    Log.e(TAG, "Error on linkToDeath - " + e);
    try {
        callback.onConnectFail(NanStatusType.INTERNAL_FAILURE);
    } catch (RemoteException e1) {
        Log.e(TAG, "Error on onConnectFail()");
    }
    return;
}

synchronized (mLock) {
    mDeathRecipientsByClientId.put(clientId, dr);
}
```

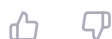
```
mUidByClientId.put(clientId, uid);  
}  
  
mStateManager.connect(clientId, uid, pid, callingPackage, callingFeatureId,  
callback,  
    configRequest, notifyOnIdentityChanged, extras);  
}
```



Based on the logs, the error appears to be related to the creation of an interface for the Wi-Fi Aware (NAN) feature. The message "Failed to create iface for ifaceType=4, requestorWs=WorkSource{10341 com.android.cts.verifier}" suggests that the creation of the interface has failed.

The function appears to be connecting a client to the Wi-Fi Aware service, and checking for various conditions before making the connection. For example, the function enforces various permissions and checks the validity of the input parameters.

The root cause of the problem could be due to a variety of reasons, such as insufficient permissions, incorrect input parameters, or an internal failure in the Wi-Fi Aware service. Further investigation, such as looking at additional logs or debugging the code, would be required to determine the exact cause of the problem.



SH

ok retry and focus on below part

```
if (notifyOnIdentityChanged) {  
    enforceNearbyOrLocationPermission(callingPackage, callingFeatureId,  
        getMockableCallingUid(), extras, "Wifi Aware attach");  
}
```

also tell what this code does



This code checks if the "notifyOnIdentityChanged" flag is set. If it is, it calls the "enforceNearbyOrLocationPermission" method with several arguments.

The purpose of the "enforceNearbyOrLocationPermission" method is to enforce that the calling package has the proper permissions to access location data or nearby device discovery functionality.

The arguments include the "callingPackage" (a string representing the package name of the caller), "callingFeatureId" (a string representing a feature ID of the caller), "getMockableCallingUid" (the user ID of the caller), "extras" (a Bundle of additional information), and a message "Wifi Aware attach". The method uses these arguments to determine whether the calling package has the required permissions for the requested functionality. If the package does not have the required permissions, the method throws a security exception.



SH

what does below code do

```
856  /**
857   * Place a request to reconfigure Aware. No additional input - intended to use
      current
858   * power settings when executed. Thus possibly entering or exiting power
      saving mode if
859   * needed (or do nothing if Aware is not active).
860   */
861   public void reconfigure() {
862       Message msg = mSm.obtainMessage(MESSAGE_TYPE_COMMAND);
863       msg.arg1 = COMMAND_TYPE_RECONFIGURE;
864       mSm.sendMessage(msg);
865   }
```

02-13 15:18:51.030 2005 2005 V WifiAwareStateManager: BroadcastReceiver:  
action=android.intent.action.SCREEN\_ON



This code defines a method `reconfigure()` in the `WifiAwareStateManager` class. When this method is called, it obtains a message object from the state machine (`mSm`) and sets the message type to `MESSAGE_TYPE_COMMAND` and the argument to `COMMAND_TYPE_RECONFIGURE`. Then, it sends this message to the state machine.

The purpose of this method is to request the reconfiguration of the Wi-Fi Aware protocol. This reconfiguration is done by sending a command message to the state machine. The exact behavior of this reconfiguration is not specified in this code, but the comments suggest that it is meant to use the current power settings and possibly enter or exit power saving mode if needed.

