SIM8500_android10_网络配置

智能模块

芯讯通无线科技(上海)有限公司
上海市长宁区临虹路289号3号楼芯讯通总部大楼
电话：86-21-31575100
技术支持邮箱：support@simcom.com
官网：www.simcom.com

名称：	SIM8500_android10_网络配置
版本：	1.00
日期：	2022.03.11
状态：	已发布

版权声明

芯讯通无线科技(上海)有限公司
上海市长宁区临虹路289号3号楼芯讯通总部大楼
电话：86-21-31575100
邮箱：simcom@simcom.com
官网：www.simcom.com

了解更多资料，请点击以下链接：
http://cn.simcom.com/download/list-230-cn.html

技术支持，请点击以下链接：
http://cn.simcom.com/ask/index-cn.html 或发送邮件至 support@simcom.com

关于文档

版本历史

版本	日期	作者	备注
1.00	2022.03.11	王璇	第一版

适用范围

本文档适用于 SIM8500 系列：

目录

1 介绍

1.1 本文目的

Android 平台可支持的网卡类型，网口，WIFI，4G data，Android 源码在同一时刻，只能开启优先级最高的一路网卡。此文档介绍，如何同时开启三路网路，以及如何通过网口共享 WIFI 域 4G data 网络数据。

1.2 参考文档

1.3 术语和缩写

WIFI：全称 WirelessFidelity，属 IEEE802.11 标准的 802.11b 子集，是一种无线传输分组数据的标准。
Ifconfig：全称 network interfaces configuring，用于显示或配置网络设备（网络接口卡）的命令。
ip rule：用于管理路由规则。
ip route：用于管理静态路由表。
Iptables：实现对网络数据包进出设备及转发的控制，当数据包需要进入设备、从设备中流出或者经该设备转发、路由时，都可以使用 iptables 进行控制。

2 Android 网络优先级

2.1 Android 平台默认网卡优先级评分

以太网优先级定义，如下：
文件路径：
android/frameworks/opt/net/ethernet/java/com/android/server/ethernet/EthernetNetworkFactory.java

```
public class EthernetNetworkFactory extends NetworkFactory {
    private final static String TAG = EthernetNetworkFactory.class.getSimpleName();
    final static boolean DBG = true;


    private final static int NETWORK_SCORE = 70;
    private static final String NETWORK_TYPE = "Ethernet";
```

WIFI 优先级定义，如下：
文件路径：android/frameworks/base/core/java/android/net/NetworkAgent.java

```
/* centralize place where base network score, and network score scaling, will be
 * stored, so as we can consistently compare apple and oranges, or wifi, ethernet and LTE
 */
public static final int WIFI_BASE_SCORE = 60;
```

4G data 优先级定义，如下：
文件路径：
android/frameworks/opt/net/ethernet/java/com/android/server/ethernet/ethernetNetworkFactoryFactory.java

```
public class TelephonyNetworkFactory extends NetworkFactory {
    public final String LOG_TAG;
    protected static final boolean DBG = true;

    private static final int REQUEST_LOG_SIZE = 40;

    private static final int ACTION_NO_OP   = 0;
    private static final int ACTION_REQUEST = 1;
    private static final int ACTION_RELEASE = 2;

    private static final int TELEPHONY_NETWORK_SCORE = 50;
```

2.2 Android 平台默认网卡优先级

Android 默认网路优先级顺序：网口（70）> WIFI（60）> 4G data（50）。如 android 手机使用过程，4G data 联网状态下，开启 WIFI 后，网路就从 4G data 自动切换到 WIFI。

同时，网口优先级可以依据客户需求进调整。例如，将网口评分修改为 NETWORK_SCORE = 30,优先级从最高变更为最低。

3 Android 平台网路切换方法

3.1 Android 平台优先级切换逻辑

Android 网络评分切换函数：

文件路径：android/frameworks/base/core/java/android/net/NetworkFactory.java

```
private void evalRequest(NetworkRequestInfo n) {

    if (shouldNeedNetworkFor(n)) {
        if (VDBG) log("  needNetworkFor");
        needNetworkFor(n.request, n.score);
        n.requested = true;
    } else if (shouldReleaseNetworkFor(n)) {
        if (VDBG) log("  releaseNetworkFor");
        releaseNetworkFor(n.request);
        n.requested = false;
    } else {
        if (VDBG) log("  done");
    }
}
```

加载新网卡，评分切换条件函数：

```
private boolean shouldNeedNetworkFor(NetworkRequestInfo n) {
    // If this request is already tracked, it doesn't qualify for need
    return !n.requested
        // If the score of this request is higher or equal to that of this factory and some
        // other factory is responsible for it, then this factory should not track the request
        // because it has no hope of satisfying it.
        && ( n.score < mScore || n.factorySerialNumber == mSerialNumber)
        // If this factory can't satisfy the capability needs of this request, then it
        // should not be tracked.
        && n.request.networkCapabilities.satisfiedByNetworkCapabilities(mCapabilityFilter)
        // Finally if the concrete implementation of the factory rejects the request, then
        // don't track it.
        && acceptRequest(n.request, n.score);
}
```

由 evalRequest()函数可见，满足 shouldNeedNetworkFor()函数条件，即执行 needNetworkFor()函数，加载对应网卡。

卸载已加载网卡，评分切换条件函数：

```
private boolean shouldReleaseNetworkFor(NetworkRequestInfo n) {
    // Don't release a request that's not tracked.
    return n.requested
        // The request should be released if it can't be satisfied by this factory. That
        // means either of the following conditions are met :
        // - Its score is too high to be satisfied by this factory and it's not already
        //   assigned to the factory
        // - This factory can't satisfy the capability needs of the request
        // - The concrete implementation of the factory rejects the request
        && ((n.score > mScore && n.factorySerialNumber != mSerialNumber)
                || !n.request.networkCapabilities.satisfiedByNetworkCapabilities(
                        mCapabilityFilter)
                || !acceptRequest(n.request, n.score));
}
```

由 evalRequest()函数可见，满足 shouldReleaseNetworkFor()函数条件，即执行 releaseNetworkFor()函数，卸载对应网卡。

以上逻辑是网络评分系统关键方法，如果当前 NetworkFactory 所处的网络优先级高于其他网络的优先级，就会触发 needNetworkFor()流程，开启高优先级网络；否则触发 releaseNetworkFor()，关闭当前低优先级网络。

3.2 Android 平台修改默认使用 4G data

Android 默认网络优先级为，网口(70) > WIFI(60) > 4G data(50),如果 android 需要默认到 4G data 网卡。调整 4G data 评分为最大即可。

例如，直接将 4G data 评分修改为 100 即可，如下：

```
--- a/src/java/com/android/internal/telephony/dataconnection/TelephonyNetworkFactory.java
+++ b/src/java/com/android/internal/telephony/dataconnection/TelephonyNetworkFactory.java
@@ -57,7 +57,7 @@ public class TelephonyNetworkFactory extends NetworkFactory {
     private static final int ACTION_REQUEST = 1;
     private static final int ACTION_RELEASE = 2;

-    private static final int TELEPHONY_NETWORK_SCORE = 50;
+    private static final int TELEPHONY_NETWORK_SCORE = 100 // 50;

     private static final int EVENT_ACTIVE_PHONE_SWITCH              = 1;
```

按照如上修改，Andorid 网卡优先级，变更为：4G data(100) > 网口(70) > WIFI(60)。

4 三网同开方法

4.1 三网同开，修改方法

修改，禁止网路连接被断开，如下：

代码路径：android/frameworks/base/services/core/java/com/android/server/ConnectivityService.java

```
    private void teardownUnneededNetwork(NetworkAgentInfo nai) {
        if (nai.numRequestNetworkRequests() != 0) {
            for (int i = 0; i < nai.numNetworkRequests(); i++) {
                NetworkRequest nr = nai.requestAt(i);
                // Ignore listening requests.
                if (nr.isListen()) continue;
                loge("Dead network still had at least " + nr);
                break;
            }
        }
        nai.asyncChannel.disconnect();
    }
```

```
--- a/services/core/java/com/android/server/ConnectivityService.java
+++ b/services/core/java/com/android/server/ConnectivityService.java
@@ -6184,7 +6184,7 @@ public class ConnectivityService extends IConnectivityManager.Stub
                 break;
             }
         }
-        nai.asyncChannel.disconnect();
+      //  nai.asyncChannel.disconnect();
     }
```

修改，网络评分切换修改，如下：

代码路径：android/frameworks/base/core/java/android/net/NetworkFactory.java

```
diff --git a/core/java/android/net/NetworkFactory.java b/core/java/android/net/NetworkFactory.java
index 5b1d12c..1aa8c38 100644
--- a/core/java/android/net/NetworkFactory.java
+++ b/core/java/android/net/NetworkFactory.java
@@ -342,8 +342,8 @@ public class NetworkFactory extends Handler {
             n.requested = true;
         } else if (shouldReleaseNetworkFor(n)) {
             if (VDBG) log("  releaseNetworkFor");
-            releaseNetworkFor(n.request);
-            n.requested = false;
+            // releaseNetworkFor(n.request);
+            // n.requested = false;
         } else {
             if (VDBG) log("  done");
         }
@@ -355,7 +355,7 @@ public class NetworkFactory extends Handler {
             // If the score of this request is higher or equal to that of this factory and some
             // other factory is responsible for it, then this factory should not track the request
             // because it has no hope of satisfying it.
-            && (n.score < mScore || n.factorySerialNumber == mSerialNumber)
+            && (0 < mScore || n.factorySerialNumber == mSerialNumber)
             // If this factory can't satisfy the capability needs of this request, then it
             // should not be tracked.
             && n.request.networkCapabilities.satisfiedByNetworkCapabilities(mCapabilityFilter)
```

4.2 功能测试方法

```
msm8953_64:/ # ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope: Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:43 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3115 TX bytes:3115

dummy0    Link encap:Ethernet  HWaddr ae:86:a7:10:90:11
          inet6 addr: fe80::ac86:a7ff:fe10:9011/64 Scope: Link
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:103 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 TX bytes:17177

wlan0     Link encap:Ethernet  HWaddr 00:0a:f5:f7:c4:44  Driver wcnss_wlan
          inet addr:172.16.186.58  Bcast:172.16.186.255  Mask:255.255.255.0
          inet6 addr: fe80::20a:f5ff:fef7:c444/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13155 errors:0 dropped:0 overruns:0 frame:0
          TX packets:109 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1004139 TX bytes:18873

eth0      Link encap:Ethernet  HWaddr 00:08:dc:91:97:98  Driver wiznet-w5500
          inet addr:172.16.186.57  Bcast:172.16.186.255  Mask:255.255.255.0
          inet6 addr: fe80::208:dcff:fe91:9798/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:23782 errors:0 dropped:0 overruns:0 frame:0
          TX packets:285 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2554304 TX bytes:43429
          Interrupt:104
```

Ifconfig 查看网卡信息，wlan0（WIFI），eth0（网口）

可通过 ping 方式测试网口功能，只要 ping 通，即网卡已通：

```
msm8953_64:/ # ping -I eth0 172.16.186.35
PING 172.16.186.35 (172.16.186.35) from 172.16.186.57 eth0: 56(84) bytes of data.
64 bytes from 172.16.186.35: icmp_seq=1 ttl=128 time=152 ms
64 bytes from 172.16.186.35: icmp_seq=2 ttl=128 time=78.3 ms
64 bytes from 172.16.186.35: icmp_seq=3 ttl=128 time=28.4 ms
```

5 Android 网口共享网络方法

5.1 Android 平台默认网路修改

调整网口平分最低，即，WIFI（60）> 4G data（50）> 网口（40），如下：
文件路径：opt/net/Ethernet/java/com/android/server/ethernet/EthernetNetworkFactory.java

```
--- a/java/com/android/server/ethernet/EthernetNetworkFactory.java
+++ b/java/com/android/server/ethernet/EthernetNetworkFactory.java
@@ -64,7 +64,9 @@ public class EthernetNetworkFactory extends NetworkFactory {
     private final static String TAG = EthernetNetworkFactory.class.getSimpleName();
     final static boolean DBG = true;

-    private final static int NETWORK_SCORE = 70;
+
+    private final static int NETWORK_SCORE = 40;
+
     private static final String NETWORK_TYPE = "Ethernet";

     private final ConcurrentHashMap<String, NetworkInterfaceState> mTrackingInterfaces =
@@ -291,7 +293,7 @@ public class EthernetNetworkFactory extends NetworkFactory {
                     new TransportInfo(ConnectivityManager.TYPE_NONE, 1));
             // EthernetNetworkFactory.NETWORK_SCORE
             sTransports.put(NetworkCapabilities.TRANSPORT_ETHERNET,
-                    new TransportInfo(ConnectivityManager.TYPE_ETHERNET, 70));
+                    new TransportInfo(ConnectivityManager.TYPE_ETHERNET, 40));
             // BluetoothTetheringNetworkFactory.NETWORK_SCORE
```

禁止网路连接被断开：
文件路径：base/services/core/java/com/android/server/ConnectivityService.java

```
    private void teardownUnneededNetwork(NetworkAgentInfo nai) {
        if (nai.numRequestNetworkRequests() != 0) {
            for (int i = 0; i < nai.numNetworkRequests(); i++) {
                NetworkRequest nr = nai.requestAt(i);
                // Ignore listening requests.
                if (nr.isListen()) continue;
                loge("Dead network still had at least " + nr);
                break;
            }
        }
        nai.asyncChannel.disconnect();
    }
```

```
--- a/services/core/java/com/android/server/ConnectivityService.java
+++ b/services/core/java/com/android/server/ConnectivityService.java
@@ -6184,7 +6184,7 @@ public class ConnectivityService extends IConnectivityManager.Stub
                 break;
             }
         }
-        nai.asyncChannel.disconnect();
+      //  nai.asyncChannel.disconnect();
     }
```

5.2 Android 平台转发网络数据到网口

在路由选择时，按照网口 IP 信息对网口配置，以及将其它网卡数据转发到网口上。
如下是网口共享 4G data 网络数据方法：

修改点如下图函数，红框部分：

文件路径：system/netd/server/RouteController.cpp

```
WARN_UNUSED_RESULT int RouteController::modifyRoute(uint16_t action, const char* interface,
                                                    const char* destination, const char* nexthop,
                                                    TableType tableType) {
    uint32_t table;
    switch (tableType) {
        case RouteController::INTERFACE: {
            table = getRouteTableForInterface(interface);
            if (table == RT_TABLE_UNSPEC) {
                return -ESRCH;
            }
            break;
        }
        case RouteController::LOCAL_NETWORK: {
            table = ROUTE_TABLE_LOCAL_NETWORK;
            break;
        }
        case RouteController::LEGACY_NETWORK: {
            table = ROUTE_TABLE_LEGACY_NETWORK;
            break;
        }
        case RouteController::LEGACY_SYSTEM: {
            table = ROUTE_TABLE_LEGACY_SYSTEM;
            break;
        }
    }

    ALOGE("modifyRoute route action = %d, interface = %s   destination= %s  nexthop= %s", action, interface , destination , nexthop);
     char cmd[128]={0};
     if (strcmp(interface,"eth0") == 0) {
         ALOGE("modifyRoute interface11= %s", interface);
         system("ip rule del table 20");
         system("ip rule add to 192.168.1.0/24 table 20 prio 19");
         system("ip route 192.168.1.0/24 eth0");
         system("ip route add 192.168.1.0/24 dev eth0 proto kernel scope link table 20");
     } else {
         ALOGE("modifyRoute interface22= %s", interface);
         if (strcmp(interface,"seth_lte0") == 0)
         system("ip rule add from all lookup seth_lte0 prio 23000");
         system("echo 1 > /proc/sys/net/ipv4/ip_forward");
         system("iptables -I FORWARD 1 -d 192.168.1.0/24 -j ACCEPT");
         system("iptables -I FORWARD 1 -s 192.168.1.0/24 -j ACCEPT");
         snprintf(cmd, sizeof(cmd), "iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o %s -j MASQUERADE",interface);
         system(cmd);

    int ret = modifyIpRoute(action, table, interface, destination, nexthop);
    // Trying to add a route that already exists shouldn't cause an error.
    if (ret && !(action == RTM_NEWROUTE && ret == -EEXIST)) {
        return ret;
    }
```

修改 diff 差异点：

```
diff --git a/server/RouteController.cpp b/server/RouteController.cpp
old mode 100644
new mode 100755
index 6722372..564e3ff
--- a/server/RouteController.cpp
+++ b/server/RouteController.cpp
@@ -880,6 +880,24 @@ WARN_UNUSED_RESULT int RouteController::modifyRoute(uint16_t action, const char*
         }
     }

+    ALOGE("modifyRoute route action = %d, interface = %s   destination= %s  nexthop= %s", action, interface , destination , nexthop);
+     char cmd[128]={0};
+     if (strcmp(interface,"eth0") == 0) {
+         ALOGE("modifyRoute interface11= %s", interface);
+         system("ip rule del table 20");
+         system("ip rule add to 192.168.1.0/24 table 20 prio 19");
+         system("ip route 192.168.1.0/24 eth0");
+         system("ip route add 192.168.1.0/24 dev eth0 proto kernel scope link table 20");
+     } else {
+         ALOGE("modifyRoute interface22= %s", interface);
+         if (strcmp(interface,"seth_lte0") == 0)
+         system("ip rule add from all lookup seth_lte0 prio 23000");
+         system("echo 1 > /proc/sys/net/ipv4/ip_forward");
+         system("iptables -I FORWARD 1 -d 192.168.1.0/24 -j ACCEPT");
+         system("iptables -I FORWARD 1 -s 192.168.1.0/24 -j ACCEPT");
+         snprintf(cmd, sizeof(cmd), "iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o %s -j MASQUERADE",interface);
+         system(cmd);
+
     int ret = modifyIpRoute(action, table, interface, destination, nexthop);
     // Trying to add a route that already exists shouldn't cause an error.
     if (ret && !(action == RTM_NEWROUTE && ret == -EEXIST)) {
```

备注：如果 user 版本执行出现 su 权限问题，可参考异常原因添加权限。如果版本路由规则，有特殊调整，需要核实路由关系后进行调整。

使用 adb 实现网口共享其它网络数据方式。如下，网口共享 4G data 数据方法:

```
ip rule del table 20
ip rule add to 192.168.1.0/24 table 20 prio 19
ip route add 192.168.1.0/24 dev eth0 proto kernel scope link table 20
ip rule add from all lookup seth_lte0 prio 23000

echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -I FORWARD 1 -d 192.168.1.0/24 -j ACCEPT
iptables -I FORWARD 1 -s 192.168.1.0/24 -j ACCEPT
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o seth_lte0 -j MASQUERADE
```

5.3 测试方法：

查看网卡信息：

```
msm8953_64:/ # ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope: Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:43 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3115 TX bytes:3115

dummy0    Link encap:Ethernet  HWaddr ae:86:a7:10:90:11
          inet6 addr: fe80::ac86:a7ff:fe10:9011/64 Scope: Link
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:103 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 TX bytes:17177

wlan0     Link encap:Ethernet  HWaddr 00:0a:f5:f7:c4:44  Driver wcnss_wlan
          inet addr:172.16.186.58  Bcast:172.16.186.255  Mask:255.255.255.0
          inet6 addr: fe80::20a:f5ff:fef7:c444/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13155 errors:0 dropped:0 overruns:0 frame:0
          TX packets:109 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1004139 TX bytes:18873
```

```
eth0      Link encap:Ethernet  HWaddr 00:08:dc:91:97:98  Driver wiznet-w5500
          inet addr:172.16.186.57  Bcast:172.16.186.255  Mask:255.255.255.0
          inet6 addr: fe80::208:dcff:fe91:9798/64 Scope: Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:23782 errors:0 dropped:0 overruns:0 frame:0
          TX packets:285 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2554304 TX bytes:43429
          Interrupt:104
```

上图可见网卡信息，wlan0（WIFI），eth0（网口）。

通过网口对端，ping 方式测试网口功能，可测试内网 ip，外网 ip。ping 通外网说明共享成功，如下图：

```
C:\Users\lenovo>ping 10.62.71.17

正在 Ping 10.62.71.17 具有 32 字节的数据:
来自 10.62.71.17 的回复: 字节=32 时间=1ms TTL=64
来自 10.62.71.17 的回复: 字节=32 时间=1ms TTL=64

10.62.71.17 的 Ping 统计信息:
    数据包: 已发送 = 2, 已接收 = 2, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 1ms, 最长 = 1ms, 平均 = 1ms
Control-C
^C
C:\Users\lenovo>ping 8.8.8.8

正在 Ping 8.8.8.8 具有 32 字节的数据:
来自 8.8.8.8 的回复: 字节=32 时间=91ms TTL=113
来自 8.8.8.8 的回复: 字节=32 时间=71ms TTL=113
来自 8.8.8.8 的回复: 字节=32 时间=73ms TTL=113

8.8.8.8 的 Ping 统计信息:
    数据包: 已发送 = 3, 已接收 = 3, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 71ms, 最长 = 91ms, 平均 = 78ms
Control-C
^C
C:\Users\lenovo>ping www.baidu.com

正在 Ping www.wshifen.com [103.235.46.39] 具有 32 字节的数据:
来自 103.235.46.39 的回复: 字节=32 时间=258ms TTL=44
来自 103.235.46.39 的回复: 字节=32 时间=296ms TTL=44

103.235.46.39 的 Ping 统计信息:
    数据包: 已发送 = 2, 已接收 = 2, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 258ms, 最长 = 296ms, 平均 = 277ms
```

6 网口常见问题

6.1 以太网驱动名称命名

新增以太网网卡驱动时，需要注意驱动网口名，定义为“eth0”。因为 android 系统默认以太网网口名为"eth0",android 以太网网卡服务加载时，按照驱动名称进行检测匹配。

6.2 以太网网口共享其它网络，对端网卡 IP 设置要点

网口对端 IP 信息配置，如网口对端连接 pc 机，pc 机 IP 信息配置参考：
PC IP ：192.168.1.* ;(* ,自己 ip 末位，自己定义)；
PC 掩码 ：255.255.255.0
PC 网关 ：192.168.1.100（此 IP，需要是板子以太网，网卡 IP）
PC DNS ：按照需求设置，电信，阿里等

板子以太网网口 IP 信息：可以使用 ifconfig 配置，也可以由对端通过 DHCP 服务自动下发：
Ifconfig 配置方法：ifconfig 192.168.1.100/24

6.3 以太网网口共享其它网络，以太网优先级评分修改点

如下图，需要修改两点：

```
--- a/java/com/android/server/ethernet/EthernetNetworkFactory.java
+++ b/java/com/android/server/ethernet/EthernetNetworkFactory.java
@@ -64,7 +64,9 @@ public class EthernetNetworkFactory extends NetworkFactory {
     private final static String TAG = EthernetNetworkFactory.class.getSimpleName();
     final static boolean DBG = true;

-    private final static int NETWORK_SCORE = 70;
+
+    private final static int NETWORK_SCORE = 40;
+
     private static final String NETWORK_TYPE = "Ethernet";

     private final ConcurrentHashMap<String, NetworkInterfaceState> mTrackingInterfaces =
@@ -291,7 +293,7 @@ public class EthernetNetworkFactory extends NetworkFactory {
                 new TransportInfo(ConnectivityManager.TYPE_NONE, 1));
             // EthernetNetworkFactory.NETWORK_SCORE
             sTransports.put(NetworkCapabilities.TRANSPORT_ETHERNET,
-                    new TransportInfo(ConnectivityManager.TYPE_ETHERNET, 70));
+                    new TransportInfo(ConnectivityManager.TYPE_ETHERNET, 40));
             // BluetoothTetheringNetworkFactory.NETWORK_SCORE
```