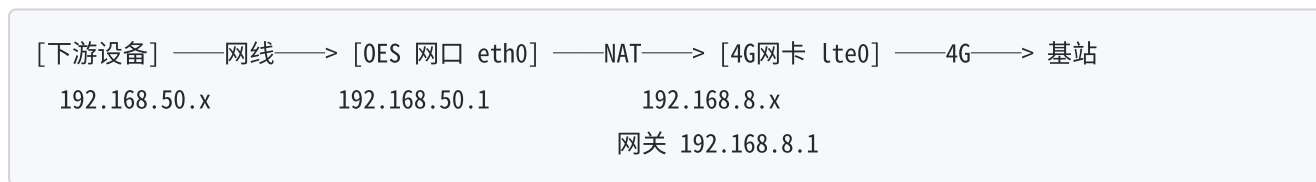


网心云 OES (A311D) + 飞牛 fnOS: 4G 网卡共享上网配置

适用条件

项目	你的情况
硬件	网心云 OES 一代（A311D / 4G内存 / 8G eMMC）
接口	1 个千兆网口、1 个 USB 2.0、 无 HDMI
系统	飞牛 fnOS ARM 版
4G 网卡	华为 E8372h-155（HiLink 路由模式，免驱）
目标	插在网口上的设备，通过 4G 正常上网

网络结构



E8372h-155 自己会拨号、自己发 DHCP、自己做 NAT，飞牛只需要再做一层转发。

⚠️ 动手前的三条底线

1. **网段必须用 192.168.50.0/24** —— 绝对不能用 192.168.8.x，那是 E8372 自己的网段，会冲突。
2. **不要开 OVS** —— 单网口 + 无 HDMI，OVS 一旦崩就是彻底失联，只能拆机救。
3. **先备好 U 盘启动盘** —— 另找一只 U 盘用 Rufus 写入飞牛 ARM 镜像贴标签收好，这是唯一的救命通道。

第 0 步：打开 SSH

飞牛后台 → 系统设置 → SSH → 打开。

电脑上用 FinalShell / MobaXterm / PowerShell 登录飞牛 IP，登录后先提权：

```
sudo -i
```

之后所有命令都在 root 下执行。

第 1 步：搞清楚两个接口叫什么

```
ip -o link show | awk -F: ' '{print $2} '
```

你会看到一串名字，典型输出：

```
lo
eth0          ← 有线网口（也可能是 end0）
enx0c5b8f279a64 ← 4G 网卡（enx 开头，后面是 MAC）
ovs-system
docker0
```

记下这两个名字，全文后面用 `<网口名>` 和 `<4G名>` 代替，你替换成实际的。

再确认 4G 网卡有没有拿到 IP：

```
ip addr show enx0c5b8f279a64
```

看到 `inet 192.168.8.x/24` 就说明 E8372 已经正常工作。

第 2 步：给 4G 网卡钉个固定名字

E8372 每次上电 MAC 会随机变，接口名 `enx...` 跟着变，重启后 NAT 规则就会失效。先固定成 `lte0`：

```
cat > /etc/udev/rules.d/70-huawei-lte.rules <<'EOF'
SUBSYSTEM=="net", ACTION=="add", ENV{ID_VENDOR_ID}=="12d1", NAME="lte0"
EOF

udevadm control --reload-rules
udevadm trigger
```

拔掉 4G 网卡再插上（或重启一次），然后确认：

```
ip addr show lte0
```

看到 `192.168.8.x` 就成了。这一步做完，后面所有命令里的 4G 网卡名统一用 `lte0`。

第 3 步：确认飞牛自己能上 4G 网

```
ping -I lte0 -c 3 192.168.8.1
ping -I lte0 -c 3 223.5.5.5
```

第一条通 = 网卡和 E8372 之间正常；第二条通 = 4G 真的能出网。

两条都通才继续。**不通的话先别往下做**，检查 SIM 卡、流量、APN（浏览器访问 `192.168.8.1` 进华为后台看）。

如果 ping 不通外网但能 ping 通 192.168.8.1，多半是默认路由没走 4G。去 **系统设置 → 网络设置** 编辑 `lte0`，IPv4 选"自动(DHCP)"，勾选"设为默认网关"，DNS 填 `223.5.5.5`。

第 4 步：开启共享（方案 A，推荐）

先看网口对应的连接名：

```
nmcli -t -f NAME,DEVICE con show
```

输出类似 `网口 1:eth0`，冒号前面是**连接名**（可能是中文），后面是设备名。

把连接名填进去执行：

```
nmcli con mod "网口 1" ipv4.method shared ipv4.addresses 192.168.50.1/24
nmcli con up "网口 1"
```

把 "网口 1" 换成你实际的连接名（引号别丢）。

这一步会自动完成三件事：开启 IP 转发、启动 DHCP 服务、配好 NAT 规则，并且重启后依然生效。

验证：

```
ip addr show eth0
```

看到 `inet 192.168.50.1/24` 就成功了。

现在把设备的网线插到 OES 网口，设备会自动拿到 `192.168.50.x`，网关和 DNS 都指向 OES，应该已经能上网了。跳到第 6 步继续加固。

第 5 步：方案 B（仅当方案 A 失败时）

ARM 版飞牛上 nmcli 共享模式偶尔不生效，用 iptables 手动来：

```
echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/99-ipforward.conf
sysctl -p /etc/sysctl.d/99-ipforward.conf
```

把 LAN 口改成静态 IP（连接名换成你的）：

```
nmcli con mod "网口 1" ipv4.method manual ipv4.addresses 192.168.50.1/24
nmcli con up "网口 1"
```

加 NAT 规则：

```
iptables -t nat -A POSTROUTING -o lte0 -j MASQUERADE
iptables -A FORWARD -i eth0 -o lte0 -j ACCEPT
iptables -A FORWARD -i lte0 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -t mangle -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu
```

`eth0` 换成你的实际网口名。最后那条 TCPMSS 是治"微信能上、网页打不开"的，必加。

保存规则：

```
apt install -y iptables-persistent
netfilter-persistent save
```

如果 apt 装不上，手动存一份：

```
iptables-save > /etc/iptables/rules.v4
```

第 6 步：开机自愈脚本（强烈建议加）

4G 网卡重启后可能晚于系统就绪，NAT 规则会漏掉。加个自动补规则的钩子：

```
cat > /etc/NetworkManager/dispatcher.d/90-4g-nat <<'EOF'
#!/bin/bash
[ "$1" = "lte0" ] || exit 0
[ "$2" = "up" ] || exit 0

iptables -t nat -C POSTROUTING -o lte0 -j MASQUERADE 2>/dev/null || \
iptables -t nat -A POSTROUTING -o lte0 -j MASQUERADE

iptables -C FORWARD -i eth0 -o lte0 -j ACCEPT 2>/dev/null || \
iptables -A FORWARD -i eth0 -o lte0 -j ACCEPT

exit 0
EOF

chmod +x /etc/NetworkManager/dispatcher.d/90-4g-nat
```

如果你的网口名不是 `eth0`，把脚本里的 `eth0` 改掉。

第 7 步：给下游设备配 IP

方案 A 下不用管，设备自动获取。

方案 B 下二选一：

只接一台设备 —— 设备上手动设静态 IP 最省事：

项	值
IP	192.168.50.10
子网掩码	255.255.255.0
网关	192.168.50.1
DNS	223.5.5.5

要接多台 —— 装 dnsmasq 发 DHCP：

```
apt install -y dnsmasq

cat > /etc/dnsmasq.d/lan4g.conf <<'EOF'
interface=eth0
bind-interfaces
port=0
dhcp-range=192.168.50.10,192.168.50.100,255.255.255.0,24h
dhcp-option=3,192.168.50.1
dhcp-option=6,223.5.5.5,114.114.114.114
EOF

systemctl restart dnsmasq
systemctl enable dnsmasq
```

port=0 是让 dnsmasq 只发 IP、不抢 DNS，避免和飞牛自带服务冲突。

第 8 步（可选）：端口映射，让 WiFi 设备访问网口设备

为什么需要

配置完成后网络分成两块，中间隔着飞牛的 NAT：

谁	IP 段	能访问谁
连 E8372 WiFi 的手机/平板	192.168.8.x	访问飞牛（用 lte0 的 IP），访问不了网口设备
插在 OES 网口的设备	192.168.50.x	能出网，访问不了 WiFi 设备

要让手机访问网口上那台设备（比如它的网页服务、SSH、远程桌面），得加 DNAT 映射。

访问飞牛自己不需要映射：飞牛在 lte0 上也有个 192.168.8.x 地址，手机直接访问 `http://192.168.8.x:5666` 即可。

先查飞牛在 WiFi 侧的地址

```
ip addr show lte0 | grep inet
```

假设输出 `inet 192.168.8.100/24`，那手机就统一访问 192.168.8.100。

建议去 E8372 后台（192.168.8.1）给飞牛做个 DHCP 静态绑定，免得重启后地址变。

添加映射（通用模板）

```
iptables -t nat -A PREROUTING -i lte0 -p tcp --dport <外部端口> \
    -j DNAT --to-destination <设备IP>:<设备端口>

iptables -A FORWARD -i lte0 -o eth0 -p tcp -d <设备IP> \
    --dport <设备端口> -j ACCEPT
```

eth0 换成你的实际网口名。第二条放行规则不能省——默认只有“已建立的连接”能从外向内走，新建连接被挡住。

常用示例

```
# 手机访问网口设备的网页服务（设备 192.168.50.10 的 80 端口）
# 手机浏览器打开 http://192.168.8.100:8080
iptables -t nat -A PREROUTING -i lte0 -p tcp --dport 8080 \
    -j DNAT --to-destination 192.168.50.10:80
iptables -A FORWARD -i lte0 -o eth0 -p tcp -d 192.168.50.10 --dport 80 -j ACCEPT

# SSH: 手机连 192.168.8.100:2222 → 设备 22
iptables -t nat -A PREROUTING -i lte0 -p tcp --dport 2222 \
    -j DNAT --to-destination 192.168.50.10:22
iptables -A FORWARD -i lte0 -o eth0 -p tcp -d 192.168.50.10 --dport 22 -j ACCEPT

# Windows 远程桌面: 192.168.8.100:3389 → 设备 3389
iptables -t nat -A PREROUTING -i lte0 -p tcp --dport 3389 \
    -j DNAT --to-destination 192.168.50.10:3389
iptables -A FORWARD -i lte0 -o eth0 -p tcp -d 192.168.50.10 --dport 3389 -j ACCEPT
```

外部端口别用 **80 / 443 / 5666**，飞牛自己占着。

保存

```
netfilter-persistent save      # 或 iptables-save > /etc/iptables/rules.v4
```

查看 / 删除

```
# 查看
iptables -t nat -L PREROUTING -n --line-numbers | grep DNAT

# 删除（把 1 换成实际行号）
iptables -t nat -D PREROUTING 1
iptables -L FORWARD -n --line-numbers      # 对应放行规则也要删
```

做不到的事

场景	能不能用 DNAT	说明
网页服务、SSH、远程桌面、VNC	✓	单端口，直接映射

场景	能不能用 DNAT	说明
Windows 文件共享 SMB	⚠️ 勉强	需要 445 + 139 + 多个 UDP 端口，容易缺
投屏（DLNA / AirPlay）	❌	靠组播广播发现设备，跨网段天然不通
智能家居发现、打印机搜索	❌	同上，依赖广播
从外面（公网）访问	❌	4G 是运营商大内网，没有公网 IP

投屏这类需求，建议直接把设备连 E8372 的 WiFi，跟手机同网段就行。

验证清单

按顺序跑一遍，全绿就完工：

```
# 1. 4G 网卡有 IP
ip addr show lte0 | grep inet

# 2. 网口是 192.168.50.1
ip addr show eth0 | grep inet

# 3. IP 转发已开（应输出 1）
cat /proc/sys/net/ipv4/ip_forward

# 4. NAT 规则存在
iptables -t nat -L POSTROUTING -n -v | grep MASQUERADE

# 5. 飞牛自己能上外网
ping -c 3 223.5.5.5
```

最后在下游设备上打开浏览器试试，并访问 **192.168.50.1:5666** 确认还能进飞牛后台。

出问题怎么查

现象	原因	处理
设备拿不到 IP	方案 A 的 DHCP 没起来 / 方案 B 没配 DHCP	用方案 A；或给设备设静态 IP
能上微信打不开网页	DNS 没下发	设备手动指定 DNS 223.5.5.5

现象	原因	处理
小网页通、大网页卡	MTU 问题	补第 5 步的 TCPMSS 规则
飞牛后台进不去	网口 IP 变了	用旧 IP 或去路由器查新 IP
4G 频繁掉线	USB 2.0 供电不足	换带独立供电的 USB Hub
重启后失效	缺自愈脚本	补第 6 步

怎么回滚

```
# 方案 A 回滚：网口恢复自动获取
nmcli con mod "网口 1" ipv4.method auto
nmcli con up "网口 1"

# 方案 B 回滚：删规则
iptables -t nat -D POSTROUTING -o lte0 -j MASQUERADE
iptables -D FORWARD -i eth0 -o lte0 -j ACCEPT
iptables -D FORWARD -i lte0 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
netfilter-persistent save

# 删掉自愈脚本和 udev 规则
rm -f /etc/NetworkManager/dispatcher.d/90-4g-nat
rm -f /etc/udev/rules.d/70-huawei-lte.rules
```

真失联了怎么救（U 盘大法）

你之前刷飞牛时执行过 `fw_setenv xl_softmode "usermode"` 和 `fw_setenv upgrade_step 3`，所以 OES 支持 U 盘优先启动，救砖不用拆机短接。

1. 断电，**拔掉 4G 网卡**（唯一的 USB 口要让给 U 盘）
2. 插上启动 U 盘、插网线、上电，等 2-3 分钟
3. 路由器后台找新上线的 `debian` 设备，记下 IP，SSH 进去后 `sudo -i`
4. 挂载 eMMC 上的原系统改配置：

```
lsblk
mkdir -p /mnt/emmc
mount /dev/mmcblk1p2 /mnt/emmc          # p2 是根分区，按 lsblk 实际输出调整

# 让 OVS 永远别再接管网络
rm -f /mnt/emmc/etc/systemd/system/multi-user.target.wants/openvswitch-switch.service

# 清理出问题的网络配置
rm -f /mnt/emmc/etc/NetworkManager/system-connections/*.nmconnection

umount /mnt/emmc
```

5. 断电、拔 U 盘、插回 4G 网卡、上电

飞牛会用默认 DHCP 重新起来，网络即恢复。