

# 网络配置

中央网络配置由uci 网络子系统处理，并存储在文件中 `/etc/config/network`。这个uci系统负责定义交换机 *VLAN*，接口配置和网络路由。

在任何网络配置更改（通过uci或其他方式）后，您需要通过编写netifid守护程序来重新加载网络配置：

```
root @ lede: /# 服务网络重新加载
```

由于netifd（网络接口守护程序），更改的接口将自动重新启动，以实际应用更改。重新启动路由器是不必要的，但也是强制配置重新加载的另一种方式。

这里是一个用于TL-WR1043ND的默认设置的网络uci子系统示例

```
root @ lede: /# uci显示网络
network.loopback =界面
network.loopback.ifname = 'L0'
network.loopback.proto = '静态'
network.loopback.ipaddr = '127.0.0.1'
network.loopback.netmask = '255.0.0.0'
network.globals =全局
network.globals.ula_prefix = 'fd27: 70FA: 5c1d :: / 48'
network.lan =界面
network.lan.type = '桥'
network.lan.ifname = 'eth0.1'
network.lan.proto = '静态'
network.lan.netmask = '255.255.255.0'
network.lan.ip6assign = '60'
network.lan.ipaddr = '192.168.1.1'
network.wan =界面
network.wan.ifname = 'eth0.2'
network.wan.proto = 'DHCP'
network.wan6 =界面
network.wan6.ifname = 'eth0.2'
network.wan6.proto = '的DHCPv6'
网络.@切换[0] =开关
网络.@切换[0].名称= 'switch0'
网络.@切换[0].reset段= '1'
网络.@切换[0].enable_vlan = '1'
网络.@ switch_vlan [0] = switch_vlan
网络.@ switch_vlan [0].device = 'switch0'
网络.@ switch_vlan [0].vlan = '1'
网络.@ switch_vlan [0].ports = '1 2 3 4 5t'
网络.@ switch_vlan [1] = switch_vlan
网络.@ switch_vlan [1].device = 'switch0'
网络.@ switch_vlan [1].vlan = '2'
网络.@ switch_vlan [1].ports = '0 5t'
```

并且这里写入相同的设置 /etc/config/network

```
root @ lede: /#cat / etc / config / network
```

配置界面'loopback'

```
选项ifname'lo'  
选项proto'static'  
选项ipaddr'127.0.0.1'  
option netmask'255.0.0.0'
```

配置全局变量

```
选项ula_prefix'fd27: 70fa: 5c1d :: / 48'
```

配置界面'lan'

```
选项类型'bridge'  
选项ifname'eth0.1'  
选项proto'static'  
option netmask'255.255.255.0'  
选项ip6assign'60'  
选项ipaddr'192.168.1.1'
```

配置界面'wan'

```
选项ifname'eth0.2'  
选项proto'dhcp'
```

配置界面'wan6'

```
选项ifname'eth0.2'  
选项proto'dhcpv6'
```

配置开关

```
选项名称'switch0'  
选项复位'1'  
选项enable_vlan'1'
```

```
config switch_vlan
```

```
选项设备'switch0'  
选项vlan'1'  
选项端口1 2 3 4 5t'
```

```
config switch_vlan
```

```
选项设备'switch0'  
选项vlan'2'  
选项端口'0 5t'
```

查看一个接口列表

```
root @ lede: /#ubus list network.interface.*
```

要查看有关特定接口的所有信息（UCI名称，而不是物理接口），请写入ifstatus <interface\_name>

```
root @ lede: /#ifstatus lan
```

# 第

路由器的最小网络配置通常由至少两个接口（lan 和 wan）和切换部分组成（如果适用）。

## 全局

该 `globals` 部分通常包含影响网络配置的独立于接口的选项。

名称	类型	需要	默认	描述
<code>ula_prefix</code>	IPv6 前缀	没有	（没有）	IPv6 ULA ( <a href="https://en.wikipedia.org/wiki/Unique_local_address">https://en.wikipedia.org/wiki/Unique_local_address</a> ) -Prefix为该设备

## 接口

类型的章节 `interface` 声明了用作IP地址设置，别名，路由，物理接口名称和防火墙规则的容器的逻辑网络- 它们在LEDE配置概念中起核心作用。

最小的界面声明由以下几行组成：


### UCI:

```
network.wan =界面
network.wan.ifname = 'eth0.2'
network.wan.proto = 'DHCP'
```

配置文件：

```
config'interface''wan'
    选项'proto''dhcp'
    选项'ifname''eth0.2'
```

- `wan` 是一个独特的逻辑接口名称
- `dhcp` 指定接口协议，DHCP在这个例子中
- `eth0.2` 是与本节相关联的物理接口

 系统将物理接口名称长度限制为15个字符，包括为某些协议添加的自动添加的前缀（例如“6in4-”，“pppoe-”，“pppoe-”）或桥接状态（“br-”）。

根据协议类型，逻辑接口名称可能仅限于9个字符。Eg'abcde67890'是使用dhcp的普通接口的有效接口名称，但不是用于pppoe接口的最终名称为“pppoe-abcde67890”（即> 15个字符）的接口名称。

使用太长的名称可能会导致错误，因为网络，防火墙或dhcp配置中的某些设置可能未被应用。

该接口协议可以是下列之一：

协议	描述	程序
static	具有固定地址和网络掩码的静态配置	ip / ifconfig
dhcp	地址和网络掩码由DHCP分配	udhcpd (Busybox的)
dhcpv6	地址和网络掩码由DHCPV6分配	odhcpd6c
ppp	PPP协议 - 拨号调制解调器连接	pppd
pppoe	以太网PPP - DSL宽带连接	pppd + plugin rp-pppoe.so
pppoa	PPP over ATM - DSL连接使用内置调制解调器	pppd + 插件...
3g	CDMA, UMTS或GPRS连接使用AT式3G调制解调器	comgt
qmi	USB调制解调器使用QMI协议	uqmi
ncm	USB调制解调器使用NCM协议	comgt-ncm +?
wwan	具有协议自动检测的USB调制解调器	wwan
hnet	自我管理家庭网络 (HNCP)	hnet-full
pptp	通过PPTP VPN连接	?
6in4	与隧道经纪人一起使用的IPv6-in-IPv4隧道, 如HE.net	?
aiccu	任何东西在任何隧道	aiccu
6to4	无状态IPv6 over IPv4传输	?
6rd	IPv6快速部署	6rd
dslite	双栈Lite	ds-lite
l2tp	PPP over L2TP伪线隧道	xl2tpd
relay	中继伪桥	relayd
gre , gretap	GRE over IPv4	gre + kmod-gre
gre6 , grev6tap	GRE over IPv6	gre + kmod-gre6
vti	VTI over IPv4	vti + kmod-ip_vti
vtiv6	VTI over IPv6	vti + kmod-ip6_vti
none	未指定协议, 因此所有其他接口设置将被忽略 (如禁用配置)	-

根据所使用的接口协议, 完整的接口声明可能需要几个其他选项。下面列出了每个协议的相应选项。必须在“必需”列中标记为“是”的选项在接口部分中定义, 如果使用相应的协议, 则可以定义标记为“否”的选项, 但也可以省略。

⚠️如果接口部分没有定义协议（不均匀 `none`），则其他设置将被完全忽略。结果是，如果接口部分提到物理网络接口（即`eth0`），即使连接了电缆（原型“无”接口已经启动），这将会关闭。

## 选项对所有协议类型都有效

名称	类型	需要	默认	描述
<code>ifname</code>	接口名称	是 (*)	(没有)	要分配给此部分的物理接口名称，如果设置了类型 <code>bridge</code> ，接口列表。 (* ) 该选项可以是空的或丢失如果只有一个无线接口的引用此网络，或如果协议类型是 <code>pptp</code> ， <code>pppoa</code> 或 <code>6in4</code>
<code>type</code>	串	没有	(没有)	如果设置为“桥”，包含给定的桥梁 <code>ifnames</code> 创建 <code>WLAN</code> 接口的名称是不可预测的， <b>therefore</b> 你不能直接在网络配置中引用它们 ( <a href="https://forum.openwrt.org/viewtopic.php?pid=203784#p203784">https://forum.openwrt.org/viewtopic.php?pid=203784#p203784</a> )
<code>stp</code>	布尔	没有	0	仅对类型“ <code>bridge</code> ”有效，启用生成树协议
<code>bridge_empty</code>	布尔	没有	0	仅对类型“ <code>bridge</code> ”有效，才能创建空桥
<code>igmp_snooping</code>	布尔	没有	1	仅对类型“ <code>bridge</code> ”有效，设置 <code>bridge</code> 的 <code>multicast_snooping</code> 内核设置
<code>macaddr</code>	MAC 地址	没有	(没有)	覆盖此接口的 <code>MAC</code> 地址
<code>mtu</code>	数	没有	(没有)	覆盖此接口的默认 <code>MTU</code>
<code>auto</code>	布尔	没有	0 对于原型 <code>none</code> ，否则 1	指定是否启动界面
<code>ipv6</code>	布尔	没有	1	指定是否在此界面上启用 (1) 或禁用 (0) <code>IPv6</code> (障碍断路器及更高版本)
<code>accept_ra</code>	布尔	没有	1 对于协议 <code>dhcp</code> ，否则 0	指定是否在此接口上接受 <code>IPv6</code> 路由器通告 弃用:
<code>send_rs</code>	布尔	没有	1 对于协议 <code>static</code> ，否则 0	指定是否在此接口上发送路由器请求 弃用:
<code>force_link</code>	布尔	没有	1 对于协议 <code>static</code> ，否则 0	指定是否将 <code>IP</code> 地址，路由和可选网关分配给接口，无论链路处于活动状态 ('1') 还是仅在链路激活 (0) 之后); 当设置为 '1' 时，载波侦听事件不会调用热插拔处理程序
<code>enabled</code>	布尔	没有	1	启用或禁用接口部分
<code>ip4table</code>	串	没有	(没有)	( <code>ipv4</code> ) 路由表，用于该接口的路由。例如，当 <code>proto = dhcp</code> 时， <code>dhcp</code> 客户端将添加到该表的路由

ip6table	串	没有	(没有)	(ipv6) 路由表，用于该接口的路由。例如，当 proto = dhcp6时，dhcp6客户端将向该表添加路由
----------	---	----	------	--

## 网络管理

网络配置可以通过运行重新应用 `/etc/init.d/network restart`。

单个接口可以带有或关闭，名称对应于相应部分的逻辑接口名称。一个意味着一个先前的，所以没有必要在重新加载一个接口时调用它们。 `ifup name ifdown name config`

```
interface ifup ifdown
```

请注意，无线接口在外部进行管理， `ifup` 可能会破坏与现有网桥的关系。在这种情况下，需要运行 `wifi up` 后 `ifup`，以重新建立桥式连接。

## 脚本友好界面

对于需要获取网络相关信息的脚本，您可以使用 `/lib/functions/network.sh` 库存LEDE固件中的功能。请参阅该文件中的源和注释，以获取有关可用内容和如何调用的更多信息。

一个简单的例子：

```
root @ lede: /# source /lib/functions/network.sh; 如果network_get_ipaddr addr“lan”;  
那么echo“IP是$ addr”; 科幻  
IP为192.168.1.1
```

要获得一个Linux界面名称，像 `eth1` 一个逻辑网络名称，就像 `wan` 你可以这样做：

```
root @ lede: /# uci get network.wan.ifname  
eth0.2
```

🕒 最后一次修改：2017/01/01 16:56 通过bobafetthotmail

除非另有说明，本维基的内容将根据以下许可证获得许可：CC Attribution-Share Alike 4.0 International  
(<http://creativecommons.org/licenses/by-sa/4.0/>)