

netctl (简体中文)

From ArchWiki

翻译状态： 本文是英文页面 Netctl 的翻译，最后翻译时间：2013-11-14，[点击这里 \(https://wiki.archlinux.org/index.php?title=Netctl&diff=0&oldid=281047\)](https://wiki.archlinux.org/index.php?title=Netctl&diff=0&oldid=281047) 可以查看翻译后英文页面的改动。

Netctl 是基于命令行的网络管理器，支持场景配置。它是 Arch 的项目，取代了原来的 netcfg。

相关文章

Network configuration

Wireless network configuration

NetworkManager

Wicd

Bridge with netctl

Contents

- 1 安装
- 2 建议阅读
- 3 配置
 - 3.1 自动化操作
 - 3.1.1 基本方法
 - 3.1.2 自动切换配置
 - 3.2 从 netcfg 迁移的方法
 - 3.3 密码加密(256-bit PSK)
- 4 提示与技巧
 - 4.1 替代'netcfg current'
 - 4.2 Eduroam
 - 4.3 Bonding
 - 4.3.1 均衡负载
 - 4.3.2 有线 -> 无线故障切换
 - 4.4 移除过时的dhcpcd租期
 - 4.5 DHCP超时问题
- 5 参见

安装

netctl (<https://www.archlinux.org/packages/?name=netctl>) 软件包已经进入官方软件仓库。

netctl (<https://www.archlinux.org/packages/?name=netctl>) 和 netcfg (<https://aur.archlinux.org/packages/netcfg/>)相互冲突，安装 netctl 会替换掉netcfg (<https://aur.archlinux.org/packages/netcfg/>)。如果配置不正确，可能导致连接中断。

Note: 请使用 `systemctl --type=service` 确保其它可以配置网络的服务都没有运行，同时使用多个网络配置工具会导致冲突。

建议阅读

建议用户使用前阅读如下文档：

- netctl (<https://github.com/joukewitteveen/netctl/blob/master/docs/netctl.1.txt>)
- netctl.profile (<https://github.com/joukewitteveen/netctl/blob/master/docs/netctl.profile.5.txt>)
- netctl.special (<https://github.com/joukewitteveen/netctl/blob/master/docs/netctl.special.7.txt>)

配置

netctl 使用配置文件来管理网络连接，这些配置文件保存在 `/etc/netctl/`。软件提供了一些配置文件的示例以供参考。这些示例位于 `/etc/netctl/examples/`。配置文件通常包括：

- ethernet-dhcp
- ethernet-static
- wireless-wpa
- wireless-wpa-static

若要使用上述示例配置文件，只需将其从 `/etc/netctl/examples/` 复制到 `/etc/netctl/` 然后按照自己的需求进一步配置即可，如：

```
# cp /etc/netctl/examples/wireless-wpa /etc/netctl/profile
```

Tip: 如要配置无线网络，可以使用 `wifi-menu -o` 来自动在 `/etc/netctl/` 中生成配置文件。

Warning: 使用 `wifi-menu -o` 而在 `/etc/netctl/` 中生成的文件名中带有 `'-'`。这将在自动设置中带来问题，建议重新为配置文件命名。

在创建好你自己的配置文件之后，可以尝试运行：

```
# netctl start profile
```

来使用新的配置文件创建一个连接。

Note: 其中profile是文件名，不包含完整路径，使用完整的路径将使 netctl 异常退出。

如果上述命令出错，可使用 `journalctl -xn` 和 `netctl status profile` 来获得对错误的详细描述，对配置文件做必要的更正后重试。

自动化操作

如果使用单一配置文件（对每个网络接口）或想人工切换配置文件，请看 [基本方法](#)。使用这种配置方法的多为服务器、工作站和路由器等。

如果需要方便的在多个配置文件间切换，请看 [自动切换配置](#)。笔记本多用此种方式。

基本方法

使用此种方式，可以为每个网络接口固定地启用一个配置文件。首先需手动检查该配置文件可以正常使用，那么可以用下面的命令启用它：

```
# netctl enable profile
```

这将创建并启用一个随计算机启动而自动运行的 `systemd` 服务。对配置文件本身的修改无法自动关联到前述创建的服务文件，修改之后，需用下述命令重新启用该配置。

```
# netctl reenable profile
```

Note: 只有在计算机启动时（或服务启动时），配置文件可以被成功地启用，连接才可以建立。特别地，对于有线连接，需插好电缆；对于无线连接，需处于信号覆盖范围内。

Tip: 若要不视线缆是否插好，强制对有线网络接口启用一个固定IP的配置，可以在配置文件中加入 `SkipNoCarrier=yes` 。

自动切换配置

`netctl` 提供了两个特殊的 `systemd` 服务来自动切换配置：

- 有线网络：`netctl-ifplugd@interface.service`。使用它，将在你插入和拔出电缆时自动切换配置文件。
- 无线网络：`netctl-auto@interface.service`。使用它，将在你从一个无线网络的覆盖范围进入另一个无线网络的范围时自动切换配置文件。

Note: `netcfg` 使用 `net-auto-wireless.service` 和 `net-auto-wired.service` 来实现相同的目的。

首先 安装 下列软件包：

- `wpa_actiond` (https://www.archlinux.org/packages/?name=wpa_actiond) 用来支持使用 `netctl-auto@interface.service` 命令。
- `ifplugd` (<https://www.archlinux.org/packages/?name=ifplugd>) 用来支持使用 `netctl-ifplugd@interface.service` 命令。

然后将 `netctl-auto@interface.service` 或 `netctl-ifplugd@interface.service` 需要启动的所有配置文件编辑好。

如果希望某些无线网络配置不要被 `netctl-auto@interface.service` 自动启用，需要专门在该配置文件中加入 `ExcludeAuto=yes` 。存在多个可用配置时，可以通过设置 `Priority=` 来为某些文件设置优先级。 `netctl-ifplugd@interface.service` 优先启用使用了 DHCP 的配置文件，如要优先启用一个使用固定IP的配置，可以设置 `AutoWired=yes` 。详细内容参见 `netctl.profile(5)` 。

Warning: 不能通过选项 `Security=wpa-config` 来让 `netctl-auto` 自动选择一个启用了WPA的配置文件。请使用 `Security=wpa-configsection` 来代替。

在设置好你的配置文件并且确认它们可以正常工作后，使用 `systemctl` 启动这些服务：

```
# systemctl enable netctl-auto@interface.service
# systemctl enable netctl-ifplugd@interface.service
```

Warning:

- 如果任何一个配置文件包含错误，例如包含空变量 `Key=`，这个文件将无法在开机时自动启用。
- 本方法与 基本方法 矛盾。如果你之前已经通过 `netctl` 启用了配置文件，运行 `netctl disable profile` 来防止这个配置在计算机启动时被启用两次。

从 `netctl 1.3` 开始，允许在不停止 `netctl-auto` 服务的情况下手工控制一个不受 `netctl-auto` 管理的网络接口。这可以通过 `netctl-auto` 命令完成。若要获得可用行为的列表，运行：

```
# netctl-auto --help
```

从 netcfg 迁移的方法

`netctl` 使用 `/etc/netctl` 保存配置，而不是 `/etc/network.d`（`netcfg` 保存配置的位置）。

从 `netcfg` 迁移，至少需下列操作：

- 禁用 `netcfg` 服务：`# systemctl disable netcfg.service`。
- 删除 `netcfg`，然后安装 `netctl` (<https://www.archlinux.org/packages/?name=netctl>)
- 将老的网络配置移动到新目录
- 根据 `netctl.profile(5)` 修改配置文件中的变量名称(主要是大小写变化，例如 `CONNECTION` 变成 `Connection`)。
- 配置固定IP时，`Address` 变量中IP后面要带有子网掩码（例如示例文件中的 `Address=('192.168.1.23/24' '192.168.1.87/24')`）。
- 如果你依照 `wireless-wpa-configsection` 中的例子建立了一个无线网络配置，注意这将覆盖以上括号中的 `wpa_supplicant` 选项的设置。要连接到一个隐藏的无线网络，在 `wireless-wpa-configsection` 中加入 `scan_ssid=1`；`Hidden=yes` 在这里不起作用。
- 接口变量和其他变量不再需要引号(这仅仅是风格变化)。
- 为每一个迁移过来的配置执行 `netctl enable profile`，**last** 工作方式有变化，参阅 `netctl.special(7)`。
- 使用 `netctl list / netctl start <profile>` 替换 `netcfg-menu`。`wifi-menu` 依然保留。
- 不同于 `netcfg`，当 `netctl` 没有连接到一块已经通电的网卡时，它默认不会自己启动一块。要解决这个问题，只需在你的 `/etc/netctl/profile` 文件中加上 `SkipNoCarrier=yes`。

密码加密(256-bit PSK)

Note: 尽管进行了“加密”，你写进配置文件的密码依然可以让你连接到一个WPA-PSK网络。也即，这个加密过程只用来将密码变得“让人看不懂”，但无法阻止对这个文件有阅读权限的人使用它来连接网络。而且将这个密码再用到其他的地方变得极其不安全，你应当考虑是否有这样做的必要。

那些**不想**让自己的无线网络密钥以“明文”的形式保存的用户可以选择使用256位预先共享密钥(PSK)来代替，它由密钥和SSID经由标准算法计算而成。

- 方法 1: 使用 `wifi-menu -o` 在目录 `/etc/netctl/` 中生成一个配置文件。
- 方法 2: 按照如下方法手工配置。

无论采用何种方法，建议运行 `chmod 600 /etc/netctl/<config_file>` 来禁止其他用户访问密码。

用`wpa_passphrase`中的方法生成你的256位PSK:

```
$ wpa_passphrase your_essid passphrase

network={
    ssid="your_essid"
    #psk="passphrase"
    psk=64cf3ced850ecef39197bb7b7b301fc39437a6aa6c6a599d0534b16af578e04a
}
```

Note: 运行后不要关闭终端，这里的结果将在之后的配置中用到。

在另一个终端窗口中，将示例文件 `wireless-wpa` 从 `/etc/netctl/examples` 复制到 `/etc/netctl`：

```
# cp /etc/netctl/examples/wireless-wpa /etc/netctl/wireless-wpa
```

你需要用常用的文本编辑器编辑 `/etc/netctl/wireless-wpa`，将你刚才生成的PSK码添加到 `Key` 变量。

完成后，你的配置文件 `wireless-wpa` 应当像下面这样：

```
/etc/netctl/wireless-wpa

Description='A simple WPA encrypted wireless connection using 256-bit PSK'
Interface=wlp2s2
Connection=wireless
Security=wpa
IP=dhcp
ESSID=your_essid
Key=\"64cf3ced850ecef39197bb7b7b301fc39437a6aa6c6a599d0534b16af578e04a
```

Note:

- 注意对变量 `Key` 使用 **special quoting rules**，详细的描述参见`netctl.profile(5)` (<https://github.com/joukewitteveen/netctl/blob/master/docs/netctl.profile.5.txt>)

末尾。

- 如果密码没有起作用，从变量 `Key` 中删除 `\` 。

提示与技巧

替代'`netcfg current`'

如果之前使用 `netcfg current`，对于通过 `netctl-auto` 建立的连接，可以使用 `# netctl-auto current` 来作为代替(自`netctl-1.3`起有此功能)。

若要人工分析连接，使用：

```
# netctl list | awk '/*/ {print $2}'
```

Eduroam

有些大学使用一个叫做"Eduroam"的系统管理其无线网络。（鉴于未发现此节对与中文用户的作用，不做翻译）For this system, a WPA config-section profile with the following format is often useful:

```
/etc/netctl/wlan0-eduroam

Description='Eduroam-profile for <user>'
Interface=wlan0
Connection=wireless
Security=wpa-configsection
IP=dhcp
WPAConfigSection=(
  'ssid="eduroam"'
  'proto=RSN'
  'key_mgmt=WPA-EAP'
  'pairwise=CCMP'
  'auth_alg=OPEN'
  'eap=PEAP'
  'identity="<user>"'
  'password="<password>"'
)
```

Tip: To prevent storing your password as plaintext, you can generate a password hash with `$ echo -n <password> | iconv -t utf16le | openssl md4`. Then use it as `'password=hash:<hash>'`.

For TTLS and certified universities this setup works:

```
/etc/netctl/wlan0-eduroam

Description='Eduroam university'
Interface=wlan0
Connection=wireless
Security=wpa-configsection
IP=dhcp
ESSID=eduroam
```

```

WPAConfigSection=(
    'ssid="eduroam"'
    'key_mgmt=WPA-EAP'
    'eap=TTLS'
    'group=TKIP'
    'anonymous_identity="anonymous@domain_university"'
    'identity="XXX@domain_university"'
    'password="XXX"'
    'ca_cert="Path/to/the/certificate"'
    'phase2="auth=PAP"'
)

```

Bonding

引自 kernel documentation

(<https://www.kernel.org/doc/Documentation/networking/bonding.txt>):

The Linux bonding driver provides a method for aggregating multiple network interfaces into a single logical "bonded" interface. The behavior of the bonded interfaces depends on the mode. Generally speaking, modes provide either hot standby or load balancing services. Additionally, link integrity monitoring may be performed.

(Linux bonding驱动提供了一个途径来把多个网络接口“绑定”成一个逻辑上的接口。绑定后的接口的行为取决于绑定的模式，一般来说，提供“随时收发”和“均衡负载”两种模式。另外，可以提供对连接的总体情况的监测功能。)

均衡负载

要用netctl配合bonding，需要从官方软件源安装 ifenslave

(<https://www.archlinux.org/packages/?name=ifenslave>)。

复制 /etc/netctl/examples/bonding 到 /etc/netctl/bonding 然后进行编辑。例如：

```

/etc/netctl/bonding

Description='Bond Interface'
Interface='bond0'
Connection=bond
BindsToInterfaces=('eth0' 'eth1')
IP=dhcp
IP6=stateless

```

现在你可以停用之前的配置文件。然后设置bonding为自动启动，切换到新的配置。例如：

```
# netctl switch-to bonding
```

Note: 这将使用 bonding 驱动的默认策略round-robin。更多信息参见 官方文档 (<https://www.kernel.org/doc/Documentation/networking/bonding.txt>)。

Tip: 查看状态和绑定模式：


```
$ cat /proc/net/bonding/bond0
```

有线 -> 无线故障切换

这一部分探讨怎样用bonding来实现当有线以太网无法工作时自动切换至无线网络。我们假设所有的网络接口默认启动dhcpcd服务。

你需要从官方源安装软件包：~~ifplugd~~ (<https://www.archlinux.org/packages/?name=ifplugd>), ifenslave (<https://www.archlinux.org/packages/?name=ifenslave>) 和 wpa_supplicant (https://www.archlinux.org/packages/?name=wpa_supplicant).

首先设置 bonding 驱动使用 active-backup：

```
/etc/modprobe.d/bonding.conf

options bonding mode=active-backup
options bonding miimon=100
options bonding primary=eth0
options bonding max_bonds=0
```

max_bonds 选项避免了 Interface bond0 already exists 错误。如果使用了MAC过滤, 应当添加设置 fail_over_mac=active。

接下来, 编写一个netctl配置文件来绑定两个网络接口：

```
/etc/netctl/failover

Description='A wired connection with failover to wireless'
Interface='bond0'
Connection=bond
BindsToInterfaces=('eth0' 'wlan0')
IP='dhcp'
SkipNoCarrier='no'
```

设置该配置文件自启动：

```
# netctl enable failover
```

将 wpa_supplicant 配置为关联一个已知网络, 可以通过 netctl profile (记得设置 IP='no'), 和一个长期运行的 wpa_supplicant 服务或者 wpa_cli 命令实现。具体方法请访问 wpa_supplicant 页面。想要长期运行 wpa_supplicant 服务, 创建一个wpa_supplicant配置文件 /etc/wpa_supplicant/wpa_supplicant-wlan0.conf 然后运行：

```
# systemctl enable wpa_supplicant@wlan0
```

在有线网络配置中设置 IP='no'。IP地址应当只被分配到bond0接口。

如果你有一个有线连接和无线连接连接到同一个网络, 现在可以将有线网络断开连接, 然后重新连接而依然保持网络通畅。在大多数情况下, 甚至连流媒体音乐都不会卡顿。

移除过时的dhcpcd租期

```
# rm /var/lib/dhcpcd/dhcpcd-wlan0.lease
```

DHCP超时问题

如果在通过DHCP申请租期时有超时问题，你可以设置超时的时限大于netctl默认的30秒。Create a file in 在 /etc/netctl/hooks/ 或者 /etc/netctl/interfaces/ 中创建一个文件，在其中加入 TimeoutDHCP=40 可以使时限设置为40秒。然后为这个文件加上可执行权限。

参见

- 官方陈述页面 (<https://bbs.archlinux.org/viewtopic.php?id=157670>)
- 在AUR中有一个可用的cinnamon applet: cinnamon-applet-netctl-systray-menu (<https://aur.archlinux.org/packages/cinnamon-applet-netctl-systray-menu/>)

Retrieved from "https://wiki.archlinux.org/index.php?title=Netctl_(简体中文)&oldid=340854"

Category: Networking (简体中文)

-
- This page was last modified on 20 October 2014, at 13:14.
 - Content is available under GNU Free Documentation License 1.3 or later unless otherwise noted.