

?? IV. ???? ??

- [Chapter 34. □ □□□](#)

MAC

0:e0: 是 FreeBSD 的 MAC 地址。FreeBSD 的 MAC 地址 (通常 test0) 是 0:e0: 开头的。在 FreeBSD 中，MAC 地址通常由 (re0) 接口生成。Expire 是 FreeBSD 的默认值，通常设置为 30 天。在 FreeBSD 中，MAC 地址通常由 (re0) 接口生成。RIP(路由协议) 通常由 (re0) 接口生成。

subnet

FreeBSD 的 IP 地址通常是 10.20.30.255。在 FreeBSD 中，IP 地址通常由 10.20.30 子网生成。example.com 是 FreeBSD 的默认域名。link#1 是 FreeBSD 的默认接口名称。

在 FreeBSD 中，路由表通常由 routed(8) 生成。在 FreeBSD 中，路由表通常由 routed(8) 生成。

host

host1 是 FreeBSD 的默认主机名。在 FreeBSD 中，主机名通常由 host1 生成。FreeBSD 的默认主机名是 host1。lo0 是 FreeBSD 的默认本地接口名称。

在 FreeBSD 中，ifconfig(8) 通常用于配置接口。lo0 是 FreeBSD 的默认本地接口名称。link#1 是 FreeBSD 的默认接口名称。

224

在 FreeBSD 中，224 是默认的子网掩码。在 FreeBSD 中，224 是默认的子网掩码。

Flags 是 FreeBSD 的默认标志。在 FreeBSD 中，Flags 通常由 U、H、G、S 组成。在 FreeBSD 中，Flags 通常由 U、H、G、S 组成。

1. 在 FreeBSD 中，Flags 通常由 U、H、G、S 组成。

标志	描述
U	接口是活动的 (up)。
H	接口是主接口。
G	接口是广播接口。
S	接口是静态接口。

C	<p>FreeBSD 默认情况下，在 /etc/rc.conf 文件中，<code>defaultrouter</code> 变量被设置为空字符串。这导致系统不会设置默认网关。</p>
W	<p>为了设置默认网关，我们需要在 /etc/rc.conf 文件中，将 <code>defaultrouter</code> 变量设置为网关的 IP 地址。</p>
L	<p>设置完 <code>defaultrouter</code> 后，我们还需要在 /etc/rc.conf 文件中，将 <code>gateway_enable</code> 变量设置为 <code>YES</code>，以启用 IP 转发功能。</p>

FreeBSD 默认情况下，在 /etc/rc.conf 文件中，`defaultrouter` 变量被设置为空字符串。这导致系统不会设置默认网关。

```
defaultrouter="10.20.30.1"
```

route 命令用于添加、删除和显示路由表。要添加默认路由，我们可以使用以下命令：

```
$ route add default 10.20.30.1
```

要查看当前的路由表，我们可以使用 `route -n` 命令。这显示了路由表的详细视图，包括目标网络、网关和接口。

34.2.2. 配置 IP 转发

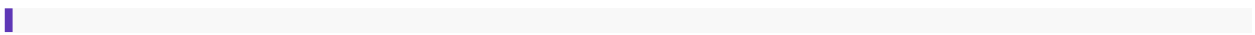
为了让 FreeBSD 能够作为网关，我们需要启用 IP 转发功能。这可以通过在 /etc/rc.conf 文件中设置 `gateway_enable` 变量为 `YES` 来实现。

此外，我们还需要在 /etc/rc.conf 文件中设置 `net.inet.ip.forwarding` 变量为 `1`，以启用 IP 转发。这可以通过以下配置实现：

```
gateway_enable="YES"           # Set to YES if this host will be a gateway
net.inet.ip.forwarding="1"    # Enable IP forwarding
```

配置完 `net.inet.ip.forwarding` 后，我们还需要重启网络服务，以使配置生效。这可以通过使用 `sysctl(8)` 命令来完成。

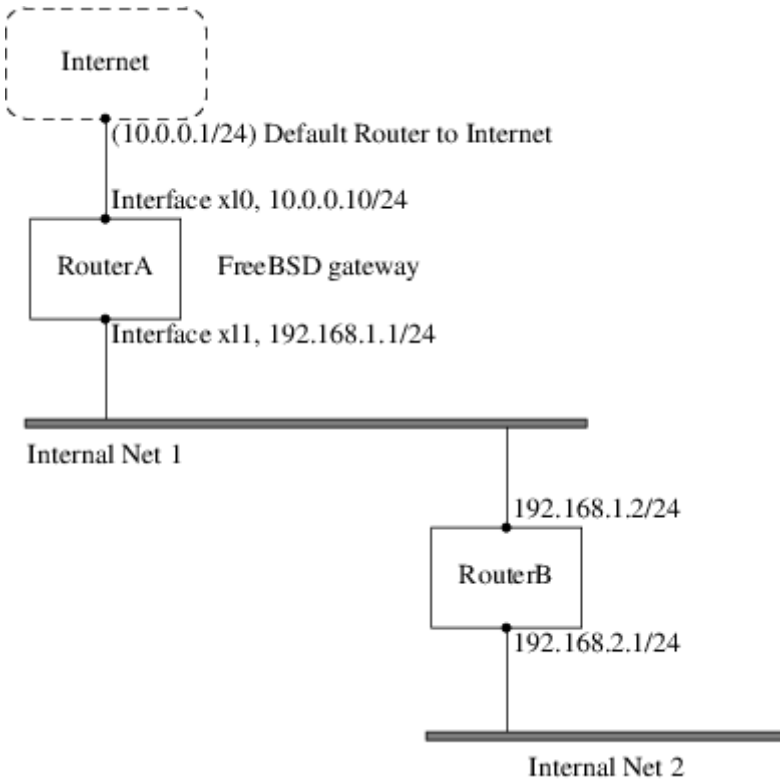
要验证配置是否正确，我们可以使用 `sysctl -n net.inet.ip.forwarding` 命令来检查 `net.inet.ip.forwarding` 的值是否为 `1`。如果配置正确，我们应该会看到输出 `1`。



Note

FreeBSD 的網路功能非常強大，支援多種網路協議。FreeBSD 支援 RIP, 1, 2, IRDP, BSD 的 [routed\(8\)](#)。BGP, OSPF 等。net/quagga 等。

網路架構如下：



RouterA 的 IP 地址是 10.0.0.1，RouterB 的 IP 地址是 192.168.1.1。

在 RouterA 上執行以下命令：

```
$ netstat -nr
Routing tables

Internet:
Destination      Gateway          Flags   Refs      Use  Netif  Expire
default          10.0.0.1        UGS     0         49378  x10
127.0.0.1        127.0.0.1      UH      0          6     lo0
10.0.0.0/24      link#1          UC      0          0     x10
192.168.1.0/24   link#2          UC      0          0     x11
```

在 A 的 192.168.2.0/24 网络中，添加一条静态路由，指向 192.168.1.2。这通常是在 FreeBSD 的 `/etc/rc.conf` 文件中配置。

```
$ route add -net 192.168.2.0/24 192.168.1.2
```

在 FreeBSD 的 `/etc/rc.conf` 文件中，添加以下配置以持久化静态路由：

```
# Add Internal Net 2 as a persistent static route
static_routes="internalnet2"
route_internalnet2="-net 192.168.2.0/24 192.168.1.2"
```

配置 `static_routes` 和 `route_internalnet2` 以添加静态路由。

在 `static_routes` 中添加 `net1` 和 `net2`，并配置 `route_net1` 和 `route_net2`。

```
static_routes="net1 net2"
route_net1="-net 192.168.0.0/24 192.168.0.1"
route_net2="-net 192.168.1.0/24 192.168.1.1"
```

34.2.3. ?? ??

在配置静态路由时，需要指定目标网络和下一跳 IP 地址。这通常是在路由表中定义的。

配置静态路由时，需要指定目标网络和下一跳 IP 地址。这通常是在路由表中定义的。

配置静态路由时，需要指定目标网络和下一跳 IP 地址。这通常是在路由表中定义的。

配置静态路由时，需要指定目标网络和下一跳 IP 地址。这通常是在路由表中定义的。

配置静态路由时，需要指定目标网络和下一跳 IP 地址。这通常是在路由表中定义的。

34.2.4. ?????? ?? ??

FreeBSD 1000000 1000000 1000000 1000000 1000000 1000000 1000000 . 1000000
1000000 FreeBSD 1000000 1000000 1000000 1000000 1000000 1000000 . 1000000 1000000
1000000 1000000 1000000 1000000 1000000 1000000 :

```
options MROUTING
```

1000000 1000000 1000000 mouted net/mouted 1000000 1000000 1000000 1000000 1000000 . 1000000
1000000 DVMP 1000000 1000000 1000000 1000000 DVMP 1000000 1000000
/usr/local/etc/mouted.conf 1000000 1000000 . mouted 1000000 map-mbone 1000000 minfo
1000000 1000000 1000000 1000000 . 1000000 1000000 1000000 .

“ Note
1000000 1000000 1000000 DVMP 1000000 PIM 1000000 1000000 . 1000000
1000000 [pim\(4\)](#) 1000000 .

34.3. ?? ???

1000000 1000000 1000000 1000000 1000000 1000000 1000000 FreeBSD 1000000 1000000 .
1000000 1000000 1000000 1000000 1000000 1000000 .

1000000 1000000 1000000 1000000 "1000000 " 1000000 1000000 , "1000000 " 1000000 1000000 1000000 .
1000000 1000000 1000000 1000000 1000000 1000000 **/etc/rc.conf** 1000000 1000000 1000000 1000000 :

```
$ sysrc ifconfig_fxp0_alias0="inet xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx"
```

alias 1000000 alias0, alias1 1000000 1000000 1000000 alias0 1000000 1000000 . 1000000 1000000
1000000 1000000 1000000 1000000 .

1000000 1000000 1000000 1000000 . 1000000 1000000 1000000 1000000 1000000 1000000
1000000 1000000 1000000 . 1000000 1000000 1000000 1000000 255.255.255.255 1000000 0xffffffff
1000000 1000000 1 1000000 1000000 1000000 .

이제, fxp0 인터페이스에 IP 주소와 마스크를 설정하고, 10.1.1.1 ~ 10.1.1.5와 202.0.75.17 ~ 202.0.75.20 범위의 IP 주소와 마스크를 설정합니다. (10.1.1.2 ~ 10.1.1.5와 202.0.75.18 ~ 202.0.75.20)는 255.255.255.255 마스크를 사용합니다.

이제 `/etc/rc.conf` 파일을 편집하여 다음과 같이 설정합니다 :

```
$ sysrc ifconfig_fxp0="inet 10.1.1.1 netmask 255.255.255.0"
$ sysrc ifconfig_fxp0_alias0="inet 10.1.1.2 netmask 255.255.255.255"
$ sysrc ifconfig_fxp0_alias1="inet 10.1.1.3 netmask 255.255.255.255"
$ sysrc ifconfig_fxp0_alias2="inet 10.1.1.4 netmask 255.255.255.255"
$ sysrc ifconfig_fxp0_alias3="inet 10.1.1.5 netmask 255.255.255.255"
$ sysrc ifconfig_fxp0_alias4="inet 202.0.75.17 netmask 255.255.255.240"
$ sysrc ifconfig_fxp0_alias5="inet 202.0.75.18 netmask 255.255.255.255"
$ sysrc ifconfig_fxp0_alias6="inet 202.0.75.19 netmask 255.255.255.255"
$ sysrc ifconfig_fxp0_alias7="inet 202.0.75.20 netmask 255.255.255.255"
```

이제 이 설정을 적용하고, IP 주소와 마스크를 설정합니다. 이 명령을 사용하여 설정을 적용합니다.

```
$ sysrc ifconfig_fxp0_aliases="inet 10.1.1.1-5/24 inet 202.0.75.17-20/28"
```

34.4. ?? ?? ??

FreeBSD는 다양한 보안 프로토콜을 지원합니다. 이 섹션에서는 WPA와 EAP에 대해 알아보겠습니다.

이 섹션에서는 WPA와 EAP의 기본 개념과 설정 방법을 알아보겠습니다.

34.4.1. EAP-TLS? ??? WPA

WPA는 802.1X 프로토콜을 사용하여 네트워크에 접근하는 클라이언트를 인증합니다. WPA는 WPA-Enterprise와 WPA-Personal로 나뉘며, WPA-Enterprise는 EAP(Extensible Authentication Protocol)을 사용합니다.

EAP은 다양한 인증 프로토콜을 지원합니다. EAP-TLS, EAP-TTLS, EAP-PEAP은 EAP의 하위 프로토콜입니다.

EAP-TLS(Extensible Authentication Protocol - Transport Layer Security)는 Wi-Fi Alliance에서 인증하는 EAP 프로토콜입니다. EAP-TLS는 CA(Certificate Authority)를 사용하여 인증을 수행합니다.

1. `ca_cert` `ca_cert` `ca_cert` `ca_cert` EAP `ca_cert` `ca_cert` .
2. ID `ca_cert` `ca_cert` TLS `ca_cert` `ca_cert` EAP `ca_cert` `ca_cert` ID `ca_cert` `ca_cert` .
3. password `ca_cert` EAP `ca_cert` `ca_cert` `ca_cert` `ca_cert` .
4. ca_cert `ca_cert` CA `ca_cert` `ca_cert` `ca_cert` `ca_cert` `ca_cert` . `ca_cert` `ca_cert` `ca_cert` `ca_cert` `ca_cert` `ca_cert` .
5. `ca_cert` `ca_cert` TLS `ca_cert` `ca_cert` `ca_cert` `ca_cert` `ca_cert` `ca_cert` . `ca_cert` `ca_cert` MD5-Challenge `ca_cert` `ca_cert` EAP `ca_cert` `ca_cert` . " `ca_cert` `ca_cert` " `ca_cert` `ca_cert` "phase2" `ca_cert` `ca_cert` .

`ca_cert` `/etc/rc.conf` `ca_cert` `ca_cert` `ca_cert` :

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

`ca_cert` `ca_cert` `ca_cert` `ca_cert` `ca_cert` :

```
$ service netif start
Starting wpa_supplicant.
DHCPCREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPCREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPCREQUEST on wlan0 to 255.255.255.255 port 67 interval 21
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 00:11:95:d5:43:62
    inet 192.168.0.254 netmask 0xfffff00 broadcast 192.168.0.255
    media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
    status: associated
    ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
    country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
    AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
    bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
    wme burst roaming MANUAL
```

34.4.3. EAP-PEAP? ??? WPA

<Note>

PEAPv0/EAP-MSCHAPv2 `ca_cert` `ca_cert` PEAP `ca_cert` . `ca_cert` `ca_cert` `ca_cert` `ca_cert` PEAP `ca_cert` `ca_cert` .

Protected EAP(PEAP) EAP-TTLS EAP-TLS PEAP EAP-TLS EAP . PEAP EAP-TLS .

PEAP TLS . PEAP EAP-TTLS . PEAP TLS EAP-TTLS . EAP-TTLS TLS .

EAP-PEAP /etc/wpa_supplicant.conf :

```
network={
  ssid="freebsdap"
  proto=RSN
  key_mgmt=WPA-EAP
  eap=PEAP - 1
  identity="test" - 2
  password="test" - 3
  ca_cert="/etc/certs/cacert.pem" - 4
  phase1="peaplabel=0" - 5
  phase2="auth=MSCHAPV2" - 6
}
```

1. EAP .
2. ID TLS EAP ID .
3. password EAP .
4. ca_cert CA . .
5. TLS . . peaplabel=0 . [wpa_supplicant.conf\(5\)](#) .
6. TLS . PEAP auth=MSCHAPV2 .

/etc/rc.conf :

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

```
❏ ❏ ❏❏❏❏ ❏❏❏❏ :
```

```
$ service netif start
Starting wpa_supplicant.
DHCPCREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPCREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPCREQUEST on wlan0 to 255.255.255.255 port 67 interval 21
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 00:11:95:d5:43:62
    inet 192.168.0.254 netmask 0xfffff00 broadcast 192.168.0.255
    media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
    status: associated
    ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
    country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
    AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
    bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
    wme burst roaming MANUAL
```

34.5. ?? Ad-hoc ??

```
❏❏ ❏❏❏❏ ❏❏ IBSS ❏❏❏ ❏❏ ❏❏❏❏ ❏❏ ❏❏❏❏❏❏❏ . ❏❏ ❏❏ , ❏❏ A❏ B❏❏❏
❏❏ ❏❏❏❏ ❏❏❏❏ ❏❏❏❏ IP ❏❏❏❏ ❏❏❏❏ SSID❏❏❏❏❏❏❏ .
```

A❏❏ :

```
$ ifconfig wlan0 create wlandev ath0 wlanmode adhoc
$ ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap
$ ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
❏ ether 00:11:95:c3:0d:ac
❏ inet 192.168.0.1 netmask 0xfffff00 broadcast 192.168.0.255
❏ media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <adhoc>
❏ status: running
❏ ssid freebsdap channel 2 (2417 Mhz 11g) bssid 02:11:95:c3:0d:ac
```

