フォーワード設定

カーネルパラメータ

```
net.ipv4.ip_forward = 1
```

にすればフォワードされる。方法は以下のどちらでもよい。

```
echo 1 > /proc/sys/net/ipv4/ip_forward
sysctl -w net.ipv4.ip_forward=1
```

上記設定をすれば、/etc/sysconfig/network に

```
FORWARD_IPV4=yes
```

を記述する必要はない。
（そもそも CentOS6 以降は FORWARD_IPV4=yes は効かない）

恒久的に設定するには
/etc/sysctl.conf
に以下を記述する。

```
net.ipv4.ip_forward = 1
```

```
iptables -L -n -v
iptables -L -n -v -t nat
```

などのコマンドで設定した内容を確認できる。

iptables 設定

■ IPv4

```
#!/bin/sh

IPTABLES="/sbin/iptables"
#/etc/init.d/iptables stop
local0Dev="eno1"
local0IP="192.168.0.0/24"
local1Dev="enp13s0"
local1IP="192.168.1.0/24"
ppp0Dev="ppp0"
ppp1Dev="ppp1"

$IPTABLES -t filter -F  #フィルター　クリア
$IPTABLES -t nat -F     #NAT クリア

$IPTABLES -P FORWARD DROP   #全てのフォワードをドロップ
$IPTABLES -P INPUT DROP     #全てのインバウンドをドロップ
$IPTABLES -P OUTPUT ACCEPT  #全てのアウトバウンドを許可

#指定ポートのインバウンドを許可
#$IPTABLES -A INPUT -i ${ppp0Dev} -p tcp --dport 2780 -j ACCEPT
# $IPTABLES -A INPUT -i ${ppp0Dev} -p tcp --dport 2722 -j ACCEPT
#$IPTABLES -A INPUT -i ${ppp0Dev} -p tcp --dport 2723 -j ACCEPT

#ループバック（127.0.0.1）を許可
$IPTABLES -A INPUT -i lo -j ACCEPT
$IPTABLES -A OUTPUT -o lo -j ACCEPT
```

```
# ICMP echo
$IPTABLES -A INPUT -p icmp --icmp-type 8 -j ACCEPT
# ICMP echo reply
$IPTABLES -A INPUT -p icmp --icmp-type 0 -j ACCEPT


#確立しているコネクションとそれに関連したインバウンドを許可
$IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#フォワード、マスカレイド設定
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

### ローカル ####
#指定アドレスからのインバウンドを許可
$IPTABLES -A INPUT -i ${local0Dev} -s ${local0IP} -j ACCEPT
#フォワード、NAT
$IPTABLES -A FORWARD -s ${local0IP} -o ${ppp0Dev} -j ACCEPT
$IPTABLES -A FORWARD -s ${local0IP} -o ${ppp1Dev} -j ACCEPT
$IPTABLES -t nat -A POSTROUTING -o ${ppp0Dev} -s ${local0IP} -j MASQUERADE
$IPTABLES -t nat -A POSTROUTING -o ${ppp1Dev} -s ${local0IP} -j MASQUERADE
# 無線からの Synergy を許可
# $IPTABLES -A FORWARD -s ${local1IP} -o ${local0Dev} -p tcp --dport 24800 -j ACCEPT

### 無線 ###
#DNS の使用許可
$IPTABLES -A INPUT  -p udp -i ${local1Dev} -s ${local1IP} --dport 53 -j ACCEPT
# $IPTABLES -A INPUT -i ${local1Dev} -p tcp --dport 2723 -j ACCEPT
$IPTABLES -A INPUT  -p tcp -m state --state NEW -i ${local1Dev} -s ${local1IP} --dport 53 -j ACCEPT
#フォワード、NAT
#$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -d ${local1IP} -j ACCEPT
$IPTABLES -A FORWARD -s ${local0IP} -o ${local1Dev} -j ACCEPT
$IPTABLES -A FORWARD -s ${local1IP} -o ${ppp0Dev} -j ACCEPT
$IPTABLES -t nat -A POSTROUTING -o ${ppp0Dev} -s ${local1IP} -j MASQUERADE
#$IPTABLES -t nat -A POSTROUTING -o ${ppp1Dev} -s ${local1IP} -j MASQUERADE

# interface port toHost  toPort returnHost
# eth0 8022 192.168.1.100 22 192.168.0.1
function setPortforward(){
 #パケットの送信先変換
 $IPTABLES -t nat -A PREROUTING -i $1 -p tcp --dport $2 -j DNAT --to $3:$4
 #パケットフォワード
 $IPTABLES -A FORWARD -i $1 -p tcp -d $3 --dport $4 -j ACCEPT
 #パケットの送信元変換
 $IPTABLES -t nat -A POSTROUTING -p tcp -d $3 --dport $4 -j SNAT --to $5
}
#指定ポートへのアクセスをローカル PC のポートへ転送
 # setPortforward ppp0 3000 192.168.0.2 80 192.168.0.1

################################################
#Outgoing packet should be real internet Address
################################################
$IPTABLES -A OUTPUT -o ${ppp0Dev} -d 10.0.0.0/8 -j DROP
$IPTABLES -A OUTPUT -o ${ppp0Dev} -d 176.16.0.0/12 -j DROP
$IPTABLES -A OUTPUT -o ${ppp0Dev} -d 192.168.0.0/16 -j DROP
$IPTABLES -A OUTPUT -o ${ppp0Dev} -d 127.0.0.0/8 -j DROP

$IPTABLES -A OUTPUT -o ${ppp1Dev} -d 10.0.0.0/8 -j DROP
$IPTABLES -A OUTPUT -o ${ppp1Dev} -d 176.16.0.0/12 -j DROP
$IPTABLES -A OUTPUT -o ${ppp1Dev} -d 192.168.0.0/16 -j DROP
$IPTABLES -A OUTPUT -o ${ppp1Dev} -d 127.0.0.0/8 -j DROP

# カーネルパラメータの設定
#echo 1 > /proc/sys/net/ipv4/ip_forward
sysctl -w net.ipv4.ip_forward=1
```

■ IPv6

とりあえず必要最低限

```
#!/bin/sh

IPTABLES="/sbin/ip6tables"
local0Dev="eno1"
local0IP="192.168.0.0/24"
local1Dev="enp13s0"
local1IP="192.168.1.0/24"
ppp0Dev="enp7s0"
#ppp1Dev="ppp1"
```

```
$IPTABLES -t filter -F  # フィルター　クリア
$IPTABLES -t nat -F     #NAT クリア

$IPTABLES -P FORWARD DROP   # 全てのフォワードをドロップ
$IPTABLES -P INPUT DROP     # 全てのインバウンドをドロップ
$IPTABLES -P OUTPUT ACCEPT  # 全てのアウトバウンドを許可

# ループバック（127.0.0.1）を許可
$IPTABLES -A INPUT -i lo -j ACCEPT
$IPTABLES -A OUTPUT -o lo -j ACCEPT

$IPTABLES -A INPUT -p ipv6-icmp -j ACCEPT

# 確立しているコネクションとそれに関連したインバウンドを許可
$IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# フォワード、マスカレイド設定
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

#sysctl -w net.ipv6.conf.all.forwarding=1
```

設定反映

■ iptables service

iptables コマンドでルールを設定したあとで

```
serivce iptables save
```

で設定を保存できる。
保存先

```
/etc/sysconfig/iptables
/etc/sysconfig/ip6tables
```

また、iptables-save で設定を標準出力に表示できる。

```
iptables-save > iptables
ip6tables-save > ip6tables
```

等で出力してから編集して、/etc/sysconfig にコピーしても良い。

■ OS 起動時にルールを設定するシェルを実行する (systemd)

systemd のサービスを作成し、ネットワークや iptables の起動後に設定用のシェルが実行されるようにする。
/etc/systemd/system/startup.service

```
[Unit]
Description=Funa startup service(oneshot)
After=network-online.target iptables.service nss-lookup.target

[Service]
Type=oneshot
ExecStart=/root/bin/startup.sh
RemainAfterExit=no

[Install]
WantedBy=multi-user.target
```

```
systemctl daemon-reload
systemctl enable startup
```

■ OS 起動時にルールを設定するシェルを実行する (rc.local)

この方法場合、

1.iptables 起動
2./etc/sysconfig/iptables を参照
3.rc.local で上記シェルの実行

となる。（rc.* の実行順序で rc.local の実行順序が iptables より後ならば）
なので、/etc/sysconfig/iptables の設定は全ての通信を遮断するか
上記シェルを実行後に

```
/etc/init.d/iptables save
```

を実行し、/etc/sysconfig/iptables と上記シェルと内容を合わせた方が無難。

補足

SNAT について

http://www.mazn.net/blog/2008/07/18/99.html
ポートフォワード時以下の条件を満たす場合は、SNAT の設定は不要。

・フォワード先のデフォルトゲートウェイが設定されている場合
・フォワード先からの戻りパケットが NAT を行ったマシンを正しく経由する場合

フォワード先からの戻りパケットを正しく返すには、SNAT により送信元を変更する。