Transparent SOCKSification

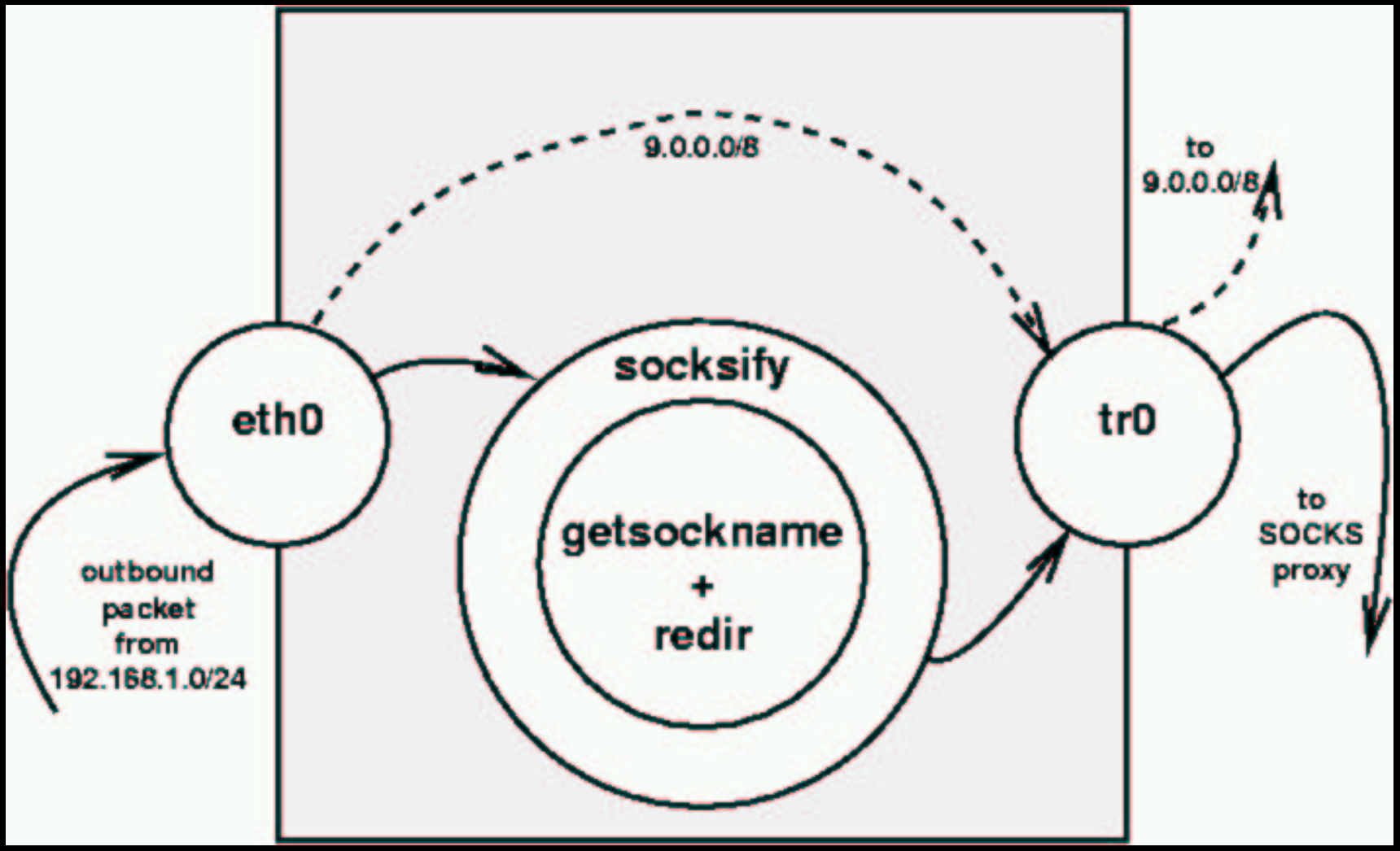
Martin Schwenke

IBM OzLabs - Linux Technology Center

Background

- A bunch of Linux hackers sitting in an IBM office.
- Want Internet access.
- Don’t want to know about SOCKS proxies.
- Have a router providing unroutable Ethernet network.
- How do we make the SOCKS proxy (relatively) invisible?

Desired router architecture



Desired iptables rules

```
np=9999

for p in ssh smtp whois http rsync cvspserver ircd ; do
  $IPTABLES -t nat -A PREROUTING -j ACCEPT \
    -i eth0 -p tcp -d 9.0.0.0/8 --dport $p
  $IPTABLES -t nat -A PREROUTING -j REDIRECT \
    -i eth0 -p tcp --dport $p --to-ports $np
done
```

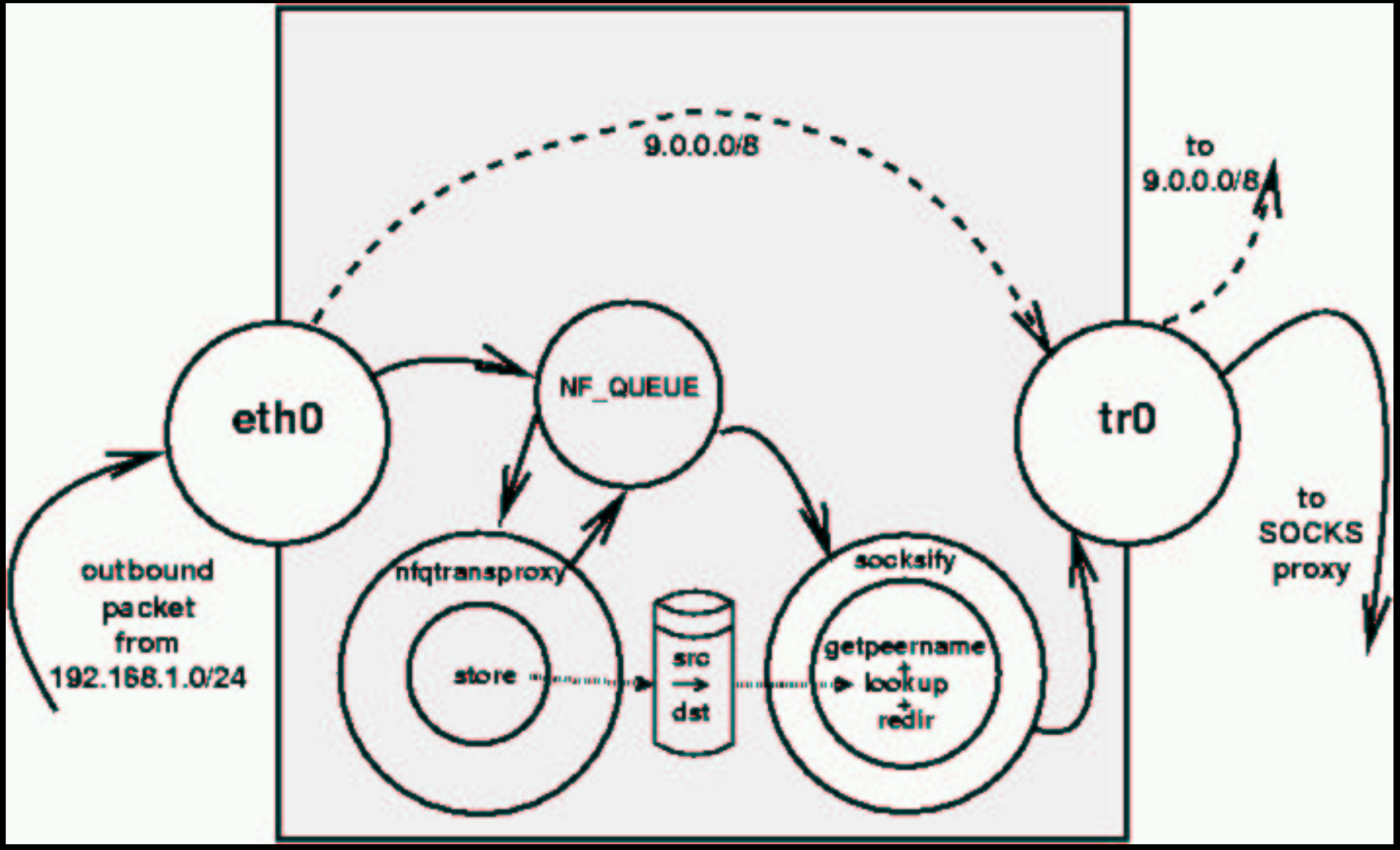
- Could just do this for all TCP ports, but that might upset the SOCKS proxy admins.
- This way we’re still controlling external access.

Desired nf_redir script on port 9999

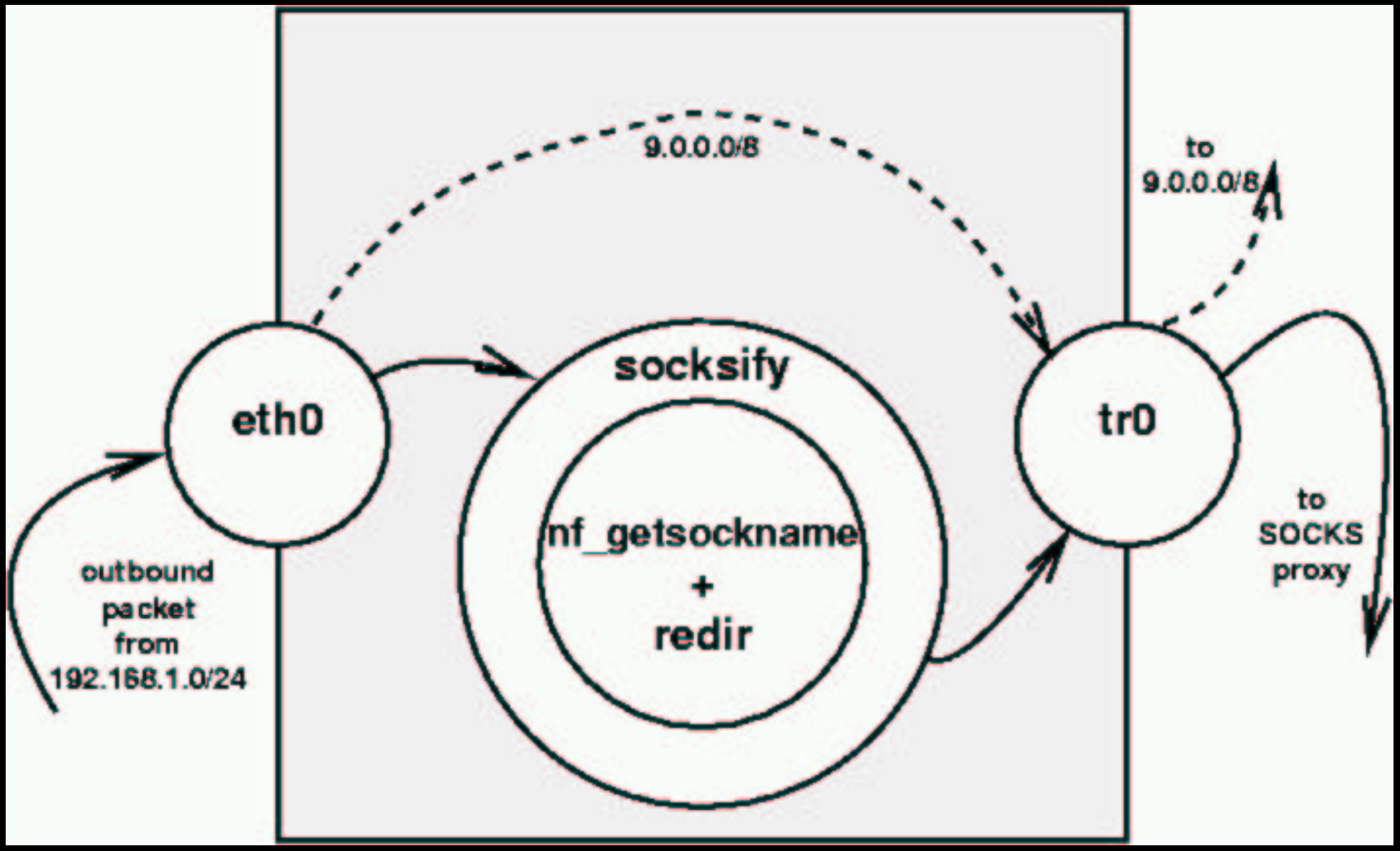
```
#!/bin/sh

set -- ‘/usr/local/sbin/getsockname -n‘
dstaddr="$1"
dstport="$2"

exec /usr/bin/socksify /usr/bin/redir --inetd \
   --caddr=${dstaddr} --cport=${dstport}
```

- getsockname is hacked version of getpeername from tcputils.

Intermediate router architecture



Final router architecture



Final nf_redir script on port 9999

```
#!/bin/sh

set -- ‘/usr/local/sbin/nf_getsockname -n‘
dstaddr="$1"
dstport="$2"

exec /usr/bin/socksify /usr/bin/redir --inetd \
   --caddr=${dstaddr} --cport=${dstport}
```

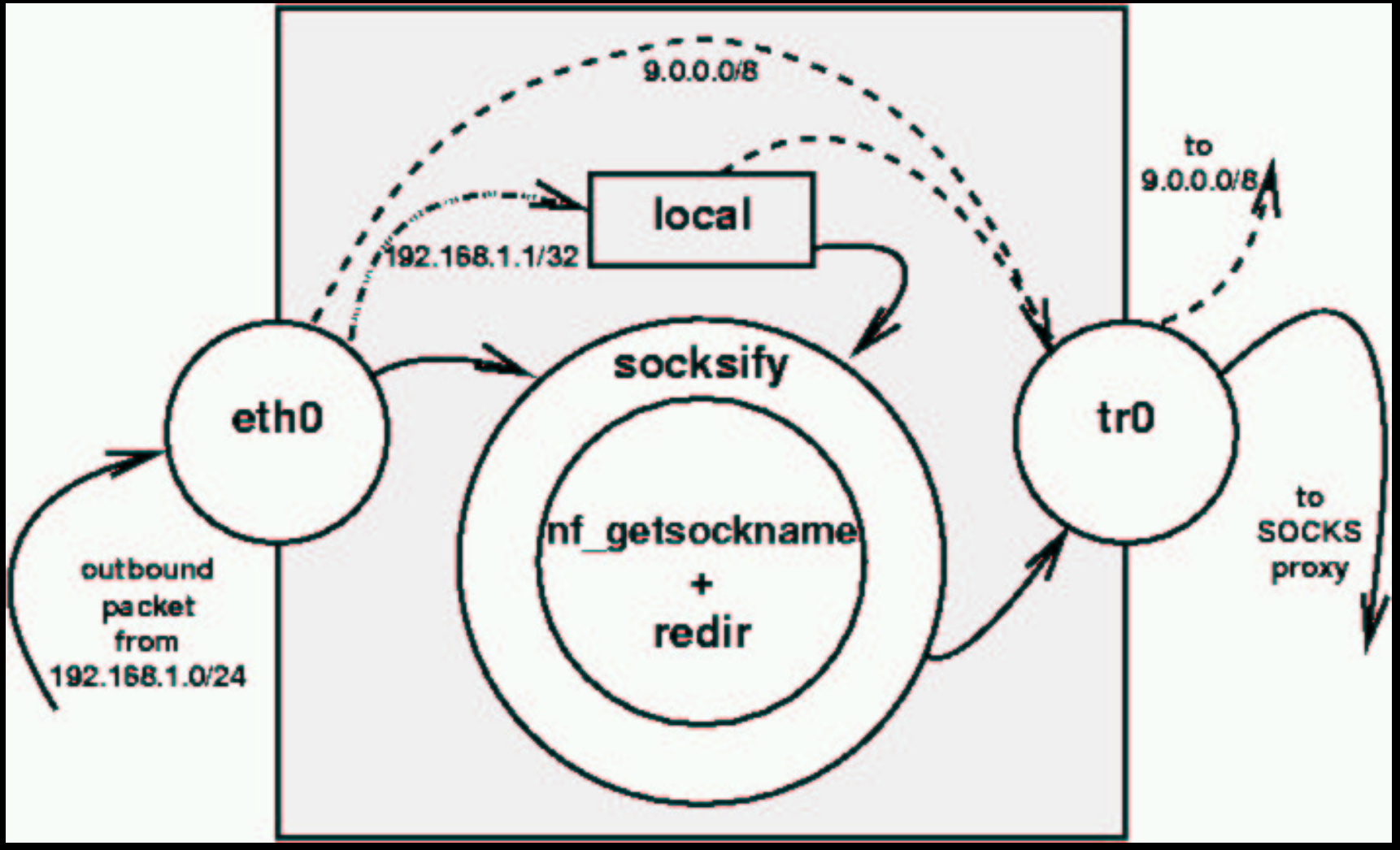
- nf_getsockname is hacked version of getpeername from tcputils with some code posted to the netfilter mailing list. This does

```
getsockopt(fd, SOL_IP, SO_ORIGINAL_DST, sa, salen)
```

instead of

```
getsock+name(fd, sa, salen)
```

Final router architecture... really!



Final iptables rules

```
np=9999

for p in ssh smtp whois http rsync cvspserver ircd ; do
  $IPTABLES -t nat -A PREROUTING -j ACCEPT \
    -i eth0 -p tcp -d 192.168.1.0/24 --dport $p
  $IPTABLES -t nat -A PREROUTING -j ACCEPT \
    -i eth0 -p tcp -d 9.0.0.0/8 --dport $p
  $IPTABLES -t nat -A PREROUTING -j REDIRECT \
    -i eth0 -p tcp --dport $p --to-ports $np
  $IPTABLES -t nat -A OUTPUT -j ACCEPT \
    -o tr0 -p tcp -d 9.0.0.0/8 --dport $p
  $IPTABLES -t nat -A OUTPUT -j REDIRECT \
    -o tr0 -p tcp --dport $p --to-ports $np
done
```

Result?