

# Configuração de rede

O sistema UNIX usa como protocolo nativo o TCP/IP (esta família de protocolos foi criada para a interconexão de máquinas UNIX). Desta forma, os conceitos de redes TCP/IP são aplicáveis às redes UNIX sem esforço. Neste módulo veremos como ajustar as configurações básicas de rede em uma máquina Linux.

## Interfaces de rede

Cada máquina possui uma ou mais interfaces de rede, geralmente no padrão *ethernet*, com endereços IP associados que as identificam para o protocolo IP. Além disso, cada interface de rede também possui um endereço físico (MAC) que a identifica na rede local na camada de enlace.

A relação das interfaces de rede físicas presentes no sistema (ativas e inativas) pode ser obtida através do comando `ifconfig -a`:

```
$ ifconfig -a
eth0      Link encap:Ethernet  Endereço de HW 84:2b:2b:7a:c2:07
          inet end.: 200.134.10.120  Bcast:200.134.10.255  Masc:255.255.255.0
          endereço inet6: fe80::862b:2bff:fe7a:c207/64  Escopo:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
          pacotes RX:1561589  erros:0  descartados:0  excesso:0  quadro:0
          Pacotes TX:726651  erros:0  descartados:0  excesso:0  portadora:0
          colisões:0  txqueuelen:1000
          RX bytes:2232242561 (2.2 GB)  TX bytes:64919270 (64.9 MB)
          IRQ:42  Endereço de E/S:0x4000

lo        Link encap:Loopback Local
          inet end.: 127.0.0.1  Masc:255.0.0.0
          endereço inet6: ::1/128  Escopo:Máquina
          UP LOOPBACK RUNNING  MTU:16436  Métrica:1
          pacotes RX:4950  erros:0  descartados:0  excesso:0  quadro:0
          Pacotes TX:4950  erros:0  descartados:0  excesso:0  portadora:0
          colisões:0  txqueuelen:0
          RX bytes:839090 (839.0 KB)  TX bytes:839090 (839.0 KB)

vboxnet0  Link encap:Ethernet  Endereço de HW 0a:00:27:00:00:00
          BROADCAST MULTICAST  MTU:1500  Métrica:1
          pacotes RX:0  erros:0  descartados:0  excesso:0  quadro:0
          Pacotes TX:0  erros:0  descartados:0  excesso:0  portadora:0
          colisões:0  txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

virbr0    Link encap:Ethernet  Endereço de HW 82:66:a0:7f:25:48
          inet end.: 192.168.122.1  Bcast:192.168.122.255  Masc:255.255.255.0
          endereço inet6: fe80::8066:a0ff:fe7f:2548/64  Escopo:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
          pacotes RX:0  erros:0  descartados:0  excesso:0  quadro:0
          Pacotes TX:725  erros:0  descartados:0  excesso:0  portadora:0
          colisões:0  txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:101310 (101.3 KB)
```

Informação similar também pode ser ou nas mensagens de inicialização (*boot*) do sistema, visualizadas nos

arquivos de log (/var/log/messages) ou através do comando `dmesg`. Eis um trecho das mensagens de inicialização que indicam as interfaces de rede disponíveis em um sistema típico:

```
...
Intel(R) PRO/100 Network Driver - version 2.3.38-k1
eth0: Intel(R) PRO/100 Network Connection
e100: eth0 NIC Link is Up 100 Mbps Half duplex
Intel(R) PRO/1000 Network Driver - version 5.2.30.1-k1
eth1: Intel(R) PRO/1000 Network Connection
e1000: eth1 NIC Link is Up 100 Mbps Full Duplex
...
```

As informações sobre as interfaces de rede ativas podem ser visualizadas através do comando `ifconfig`. Eis um exemplo de saída desse comando:

```
$ ifconfig
eth0      Link encap:Ethernet  Endereço de HW 84:2b:2b:7a:c2:07
          inet end.: 200.134.10.120  Bcast:200.134.10.255  Masc:255.255.255.0
          endereço inet6: fe80::862b:2bff:fe7a:c207/64  Escopo:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
          pacotes RX:1561644  erros:0  descartados:0  excesso:0  quadro:0
          Pacotes TX:726702  erros:0  descartados:0  excesso:0  portadora:0
          colisões:0  txqueuelen:1000
          RX bytes:2232247900 (2.2 GB)  TX bytes:64949454 (64.9 MB)
          IRQ:42  Endereço de E/S:0x4000

lo        Link encap:Loopback Local
          inet end.: 127.0.0.1  Masc:255.0.0.0
          endereço inet6: ::1/128  Escopo:Máquina
          UP LOOPBACK RUNNING  MTU:16436  Métrica:1
          pacotes RX:4952  erros:0  descartados:0  excesso:0  quadro:0
          Pacotes TX:4952  erros:0  descartados:0  excesso:0  portadora:0
          colisões:0  txqueuelen:0
          RX bytes:839287 (839.2 KB)  TX bytes:839287 (839.2 KB)

virbr0    Link encap:Ethernet  Endereço de HW 82:66:a0:7f:25:48
          inet end.: 192.168.122.1  Bcast:192.168.122.255  Masc:255.255.255.0
          endereço inet6: fe80::8066:a0ff:fe7f:2548/64  Escopo:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
          pacotes RX:0  erros:0  descartados:0  excesso:0  quadro:0
          Pacotes TX:728  erros:0  descartados:0  excesso:0  portadora:0
          colisões:0  txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:101662 (101.6 KB)
```

A interface `lo` indica o dispositivo virtual chamado *loopback*, que permite ao host acessar seus próprios serviços TCP/IP. O endereço IP dessa interface é sempre 127.0.0.1, com máscara 255.0.0.0. Por ser um dispositivo virtual, ele não possui endereço físico (MAC). Na maioria dos dispositivos e sistemas essa interface é definida na configuração default do sistema, por isso não é necessário configurá-la.

Além de listar as interfaces ativas, o comando `ifconfig` pode ser usado para ligar/desligar/configurar uma determinada interface, ou para definir novas interfaces de rede na máquina. A sintaxe básica desse comando é a seguinte:

```
ifconfig <interface> [<ip-address>]
                [netmask <mask>]
```

```
[broadcast <bcast-address>]
[up|down]
```

Eis alguns exemplos de uso:

Para desativar a interface eth0 (primeira interface *ethernet*):

```
$ ifconfig eth0 down
```

Para ativar a interface eth0 com uma configuração pré-definida (default do sistema):

```
$ ifconfig eth0 up
```

Para ativar a interface eth0 com uma configuração específica:

```
$ ifconfig eth0 10.0.0.17 netmask 255.0.0.0 up
```

Para associar um segundo endereço IP (virtual) à interface eth0:

```
$ ifconfig eth0:1 66.193.175.173 netmask 255.255.254.0
```

## Configuração de rotas

A tabela de roteamento indica quais as rotas ativas em uma máquina. Ela pode ser obtida através do comando `route`. Eis uma saída típica desse comando:

```
$ route
Tabela de Roteamento IP do Kernel
Destino      Roteador      MáscaraGen.   Opções Métrica Ref   Uso Iface
200.134.10.0 *              255.255.255.0 U       1       0       0 eth0
192.168.122.0 *             255.255.255.0 U       0       0       0 virbr0
link-local   *             255.255.0.0   U      1000    0       0 eth0
default      metcalfe.dainf. 0.0.0.0       UG      0       0       0 eth0

$ route -n
Tabela de Roteamento IP do Kernel
Destino      Roteador      MáscaraGen.   Opções Métrica Ref   Uso Iface
200.134.10.0 0.0.0.0       255.255.255.0 U       1       0       0 eth0
192.168.122.0 0.0.0.0       255.255.255.0 U       0       0       0 virbr0
169.254.0.0   0.0.0.0       255.255.0.0   U      1000    0       0 eth0
0.0.0.0       200.134.10.254 0.0.0.0       UG      0       0       0 eth0
```

A tabela acima indica que:

- A rede 200.17.98.0 pode ser acessada através da interface *ethernet* eth0.
- O *gateway* default é o dispositivo `router.pucpr.br`

O comando `route` também é usado para configurar as rotas usadas pelo sistema. Sua sintaxe básica de uso é a seguinte:

```
route [-v] add [-net|-host] Alvo
      [netmask Nm]
      [gw Gw]
      [metric N]
```

```
[mss M]
>window W]
>irtt I]
>reject] [mod] [dyn] [reinststate]
>[[dev] If]
```

ou

```
route [-v] del [-net|-host] Alvo
>gw Gw]
>netmask Nm]
>metric N]
>[[dev] If]
```

Eis alguns exemplos de uso desse comando:

Para adicionar a rota de *loopback* (com máscara default 255.0.0.0 e dispositivo lo):

```
$ route add -net 127.0.0.0
```

Para adicionar/remover uma rota para a rede 192.120.10.0 através do dispositivo eth0:

```
$ route add -net 192.120.10.0 netmask 255.255.255.0 dev eth0
```

```
$ route del -net 192.120.10.0 netmask 255.255.255.0 dev eth0
```

Para adicionar/remover uma rota default usando router02 como *gateway*:

```
$ route add default gw router02
```

```
$ route del default gw router02
```

Para adicionar uma rota para o host 10.0.0.125 através do dispositivo eth0:

```
$ route add -host 10.0.0.125 dev eth0
```

Para adicionar uma rota para a rede 200.192.112.0 através do dispositivo eth1:

```
$ route add -net 200.192.112.0 netmask 255.255.255.0 dev eth1
```

Além de rotear seus próprios pacotes, um host pode também rotear pacotes recebidos em outras interfaces, agindo portanto como *gateway* para outras redes. Essa função é denominada "IP forwarding".

Para verificar se o "IP forwarding" está ativado em uma máquina:

```
$ cat /proc/sys/net/ipv4/ip_forward
```

Para ativar ou desativar o IP forwarding:

```
$ echo 1 > /proc/sys/net/ipv4/ip_forward
```

```
$ echo 0 > /proc/sys/net/ipv4/ip_forward
```

## Resolução de endereços físicos

Os endereços físicos (da camada de enlace) são resolvidos (associados a endereços IP) através do protocolo ARP - *Address Resolution Protocol*. O comando `arp` permite visualizar o cache de endereços ARP na máquina local. Eis um exemplo de uso:

```
$ arp
Endereço          TipoHW  EndereçoHW          Flags Máscara  Iface
mccarthy.dainf.ct.utfpr ether    08:00:20:a3:de:a3   C
dainf.ct.utfpr.edu.br ether    00:1c:c0:39:71:ae   C
metcalfe.dainf.ct.utfpr ether    00:08:54:b3:75:79   C
```

## Status da atividade de rede

A comunicação em TCP/IP pode ocorrer através de serviços com ou sem conexão, providos respectivamente por TCP e UDP. Cada serviço é identificado por uma porta, na qual um processo tem um *socket* aberto para receber solicitações. Os principais serviços TCP/UDP estão listados no arquivo `/etc/services`, do qual apresentamos uma pequena parte:

```
# service      port/protocol  aliases      description
netstat        15/tcp
qotd           17/tcp         quote
msp           18/tcp         # message send protocol
msp           18/udp         # message send protocol
chargen       19/tcp         ttytst source
chargen       19/udp         ttytst source
ftp-data      20/tcp
ftp           21/tcp
fsp           21/udp         fspd
ssh           22/tcp         # SSH Remote Login Protocol
ssh           22/udp         # SSH Remote Login Protocol
telnet        23/tcp
smtp          25/tcp         mail
time          37/tcp         timeserver
...
```

Os processos que respondem pelos serviços são normalmente chamados *daemons*, e são executados em *background* (segundo plano). Vários deles, sobretudo os que gerenciam serviços mais importantes, são lançados pelos scripts de inicialização do sistema, contidos no diretório `/etc/rc.d`, cuja estrutura é definida a seguir:

```
:
|
|-- rc.d
|   |-- init.d  scripts de inicialização de serviços
:   |-- rc0.d   scripts a executar ao entrar no run-level 0
|   |-- rc1.d   idem, 1
|   |-- rc2.d   idem, 2
|   |-- rc3.d   idem, 3
|   |-- rc4.d   idem, 4
|   |-- rc5.d   idem, 5
```

```
`-- rc6.d idem, 6
```

O comando `netstat` permite obter uma série de informações sobre os serviços de rede em uma máquina UNIX. Suas principais opções são:

- `-r` : informações de roteamento (similar ao comando `route`)
- `-i` : informações sobre interfaces (similar ao comando `ifconfig`)
- `-a` : indica também as portas TCP e UDP abertas no host)
- `-p` : indica quais os processos associados a portas de rede abertas
- `-g` : informações sobre grupos multicast conhecidos no host
- `-n` : saída numérica (sem converter IPs e números de porta em seus nomes respectivos)

A operação default mostra as conexões e *sockets* ativos, como vemos na listagem a seguir:

```
$ netstat -n
Conexões Internet Ativas (sem os servidores)
Proto Recv-Q Send-Q Endereço Local          Endereço Remoto          Estado
tcp      0      0 200.134.10.120:55769    74.125.234.103:80      ESTABELECIDA
tcp      38      0 200.134.10.120:39320    199.47.218.160:443
ESPERANDO_FECHAR
tcp      0      0 200.134.10.120:37098    199.47.217.149:80      ESTABELECIDA
tcp      0      0 200.134.10.120:44811    38.104.86.204:443      ESTABELECIDA
tcp      0      0 200.134.10.120:55458    200.134.10.1:80        TIME_WAIT
tcp      0      0 127.0.0.1:34668         127.0.0.1:40751        ESTABELECIDA
tcp      0      0 127.0.0.1:40751         127.0.0.1:34668        ESTABELECIDA
tcp      38      0 200.134.10.120:45837    23.21.220.122:443
ESPERANDO_FECHAR
tcp      0      0 200.134.10.120:45950    74.125.47.125:443      ESTABELECIDA
tcp      0      0 200.134.10.120:38835    157.55.56.142:40042    ESTABELECIDA
tcp      0      0 200.134.10.120:58365    38.104.86.203:443      ESTABELECIDA
tcp      38      0 200.134.10.120:44763    199.47.217.177:443
ESPERANDO_FECHAR
tcp      0      0 200.134.10.120:55457    200.134.10.1:80        TIME_WAIT

Domain sockets UNIX ativos (sem os servidores)
Proto RefCnt Flags      Type      State      I-Node  Caminho
unix  2      [ ]      DGRAM          6534      @/org/kernel/udev/udev
unix  2      [ ]      DGRAM          11123
@/org/freedesktop/hal/udev_event
unix  2      [ ]      DGRAM          8753      @
unix  21     [ ]      DGRAM          7929      /dev/log
unix  3      [ ]      STREAM         CONECTADO   600248
unix  3      [ ]      STREAM         CONECTADO   600247
unix  3      [ ]      STREAM         CONECTADO   600211      @/tmp/dbus-D0ZmiXx0PC
...
```

O comando `netstat` possui uma série de opções úteis para verificar o status corrente das operações de rede. Consulte a página de manual do comando para maiores informações.

From:

<https://wiki.inf.ufpr.br/maziero/> - **Prof. Carlos Maziero**

Permanent link:

[https://wiki.inf.ufpr.br/maziero/doku.php?id=espec:configuracao\\_de\\_rede\\_em\\_uni\\_x](https://wiki.inf.ufpr.br/maziero/doku.php?id=espec:configuracao_de_rede_em_uni_x)

Last update: **2020/08/18 22:14**

