W DEBIAN głównym pakietem narzędzi sieciowych jest **iproute2**, oferuje on większe możliwości oraz bardziej ujednolicony interfejs obsługi w stosunku do narzędzi z pakietu **net-tools** (**ifconfig**, **route**, **arp**, **ifup**, **ifdown**). Głównym programem pakietu **iproute2** jest program **ip** zawierający funkcjonalność kilku starszych narzędzi w jednym.

Konfiguracja interfejsów

Aby wyświetlić konfigurację interfejsów użyjemy polecenia **ip addr**.

```
# ip addr
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
   inet6 ::1/128 scope host
      valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
   link/ether 00:50:8d:e1:e8:83 brd ff:ff:ff:ff:ff:ff
   inet 10.0.0.1/24 brd 10.0.0.255 scope global eth0
   inet 10.0.0.22/24 brd 10.0.0.255 scope global secondary eth0
   inet6 fe80::250:8dff:fee1:e883/64 scope link
      valid_lft forever preferred_lft forever
3: sit0: <NOARP> mtu 1480 qdisc noop
   link/sit 0.0.0.0 brd 0.0.0.0
```

Polecenie wyświetliło listę dostępnych interfejsów, każdy z nich oznaczony jest numerem porządkowym. W większości wypadków najbardziej interesujące są dla nas interfejsy fizyczne (eth0 na powyższym przykładzie). Interfejsy *lo* oraz *sit0*, są "wirtualnymi" interfejsami, pierwszy z nich to interfejs pętli zwrotnej (loopback), drugi służy do tunelowania protokołu IPv6 wewnątrz IPv4.

Tryb pracy urządzenia jest wyświetlany wewnątrz trójkątnych nawiasów, oto kilka oznaczeń:

- UP - urządzenie działa
- LOOPBACK - interfejs pętli zwrotnej
- BROADCAST - urządzenie ma możliwość wysyłania komunikatów rozgłoszeniowych

- MULTICAST - interfejs może być używany do transmisji typu multicast
- PROMISC - tryb nasłuchiwania, używany przez monitory sieci i sniffery
- NO-CARRIER - brak nośnej, komunikat spotykany zwykle w wypadku braku fizycznego połączenia z siecią

Poniżej omawianego wiersza wyświetlone zostały informacje o adresach związanych z urządzeniem (adresy IP i maski podsieci są przedstawione w notacji CIDR):

-link/ether - adres fizyczny karty sieciowej (MAC)
-inet - dane protokołu IPv4
-inet6 - dane protokołu IPv6

Zarządzanie interfejsami

Do najczęstszych operacji tego typu należy włączanie i wyłączanie interfejsów, w tym celu użyjemy następującego polecenia:

ip link set {$interfejs} {up/down}

```
# ip link set eth0 up
# ip link set eth0 down
```

Dodawanie/usuwanie adresów IP interfejsu:

ip addr {add/del} {$adresIP}/{$maska} dev {$interfejs}

```
# ip addr add 10.1.1.1/24 dev eth0
# ip addr del 10.1.1.1/24 dev eth0
```

Adresy sprzętowe (MAC)

Do odczytania adresu sprzętowego lokalnych kart sieciowych możemy użyć opisanego wcześniej polecenia **ip addr**. Aby odczytać MAC zdalnej maszyny użyjemy programu **arping {$nazwa/$IP}** np.:

```
# arping 10.0.0.100
ARPING 10.0.0.100 from 10.0.0.1 eth0
Unicast reply from 10.0.0.100 [00:04:ED:07:25:F8]  4.250ms
Unicast reply from 10.0.0.100 [00:04:ED:07:25:F8]  0.976ms
Unicast reply from 10.0.0.100 [00:04:ED:07:25:F8]  0.929ms
```

Dowolny adres MAC karty sieciowej ustawimy poleceniem:

ip link set {$interfejs} address {$MAC-adres}

```
# ip link set eth0 address 01:01:01:01:01:01
```

Aby adres sprzętowy za każdym razem był ustawiany dla interfejsu, powinniśmy dokonać odpowiedniego wpisu w pliku konfiguracyjnym interfejsu. W przypadku urządzenia eth0 musimy zmodyfikować plik */etc/sysconfig/interfaces/ifcfg-eth0*, i dodać zmienną MACADDR np.:

```
MACADDR="01:01:01:01:01:01"
```

ARP

Aby wyświetlić tablicę ARP należy użyć polecenia **ip neighbour**:

```
# ip neighbour
10.0.0.140            ether   00:E0:7D:A1:8B:E2   C                     eth0
10.0.0.2              ether   4C:00:10:54:19:50   C                     eth0
10.0.0.100            ether   00:04:ED:07:25:F8   C                     eth0
```

Tablica ARP wypełnia się w miarę komunikowania się z innymi hostami, aby wymusić zbadanie działania protokołu ARP wystarczy zainicjować komunikację. Możemy użyć do tego programu **ping**.

Możemy stworzyć własną, statyczną tablicę ARP, posłuży nam do tego plik */etc/sysconfig/static-arp* w którym umieszczamy wpisy zawierające kolejno: interfejs, adres MAC, adres IP oraz status wpisu.

```
eth0 00:80:48:12:c2:3c 192.168.10.10 permanent
eth0 00:c0:df:f9:4e:ac 10.0.0.7 permanent
```

Opcja permanent oznacza, że wpis nigdy nie wygasa.

Serwery nazw (DNS)

Do odpytywania serwerów DNS możemy użyć programu **host** z pakietu **bind-utils**. Pozwala na szybkie sprawdzenie poprawności konfiguracji domeny (strefy).

host {$nazwa/$IP} {$dns_nazwa/$dns_IP}

Pierwszy parametr to nazwa domeny lub IP maszyny, drugi parametr nie jest obowiązkowy - wskazuje na serwer nazw, który chcemy odpytać. Jeśli nie podamy drugiego parametru użyty zostanie serwer zdefiniowany w pliku */etc/resolv.conf*.

Odpytanie serwera DNS o domenę, wpisanego do pliku */etc/resolv.conf*

```
$ host debian-linux.org
debian-linux.org has address 217.149.246.8
```

Odpytanie o adres IP (odwzorowanie odwrotne)

```
$ host 217.149.246.8
8.246.149.217.in-addr.arpa domain name pointer webmachine.debian-linux.org.
```

Odpytanie dowolnego serwera DNS

```
$ host debian-linux.org ns4.debian-linux.org.
Using domain server:
Name: ns4.debian-linux.org.
Address: 217.149.246.7#53
Aliases:

debian-linux.org has address 217.149.246.8
```

Aby odczytać konkretny rekord domeny użyjemy parametru "-t". Dla przykładu spróbujemy określić rekord MX dla domeny debian-linux.org:

```
$ host -t mx debian-linux.org
debian-linux.org mail is handled by 0 a.mx.debian-linux.org.
```

W celu poznania szczegółów zarejestrowanej domeny (np. obsługujące ją serwery nazw) możemy posłużyć się programem **whois**:

```
$ whois debian-linux.org
```

Wydajność sieci

Aby sprawdzić przepustowość sieci pomiędzy dwoma hostami możemy użyć programu **iperf**. Jest to program typu klient-serwer sprawdzający na kilka sposobów przepustowość połączenia. Rozpoczynamy od instalacji programu na obu interesujących nas maszynach, następnie na jednej z maszyn uruchamiamy program w trybie serwera:

```
$ iperf -s
```

a na drugiej w trybie klienta - **iperf -c {$adres_serwera}** np.:

```
$ iperf -c 192.168.0.5
```

Po chwili odczytujemy wyniki testu.

Monitorowanie sieci

Nawiązane połączenia i otwarte porty protokołu TCP/IP możemy kontrolować za pomocą programu **netstat** np.:

```
# netstat -tua

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:blackjack             *:*                     LISTEN
tcp        0      0 *:ircs                  *:*                     LISTEN
tcp        0      0 *:netbios-ssn           *:*                     LISTEN
tcp        0      0 *:ipp                   *:*                     LISTEN
tcp        0      0 *:microsoft-ds          *:*                     LISTEN
```

```
tcp        0      0 gargamel:ircs           192.168.1.3:rapidmq-reg ESTABLISHED
tcp        0      0 gargamel:imaps          192.168.1.6:ttc-etap-ds ESTABLISHED
tcp        0      0 gargamel:td-postman     host-92.gadugadu.p:8074 ESTABLISHED
tcp        0      0 gargamel:cma            chrome.pl:5223          ESTABLISHED
tcp        0      0 *:1024                  *:*                     LISTEN
udp        0      0 gargamel:netbios-ns     *:*
udp        0      0 *:netbios-ns            *:*
udp        0      0 *:ipp                   *:*
```

Na powyższym przykładzie zostały wyświetlone dane dotyczące protokołu TCP (opcja: -t) oraz UDP (-u). Dodatkowo zostały wyświetlone gniazda nasłuchujące (-a). Warte uwagi są jeszcze dwa parametry: -n i -p, pierwszy wyświetla porty i adresy w postaci liczb, drugi zaś wyświetla nazwy programów korzystających z danych gniazd.

Do analizowania wielkości ruchu sieciowego na interfejsie polecam program **nload**, jest to proste narzędzie rysujące wykresy pomocą znaków ASCII. Podstawowe statystyki możemy przeglądać dzięki programowi **ip** np. **ip -s link**, dokładniejsze dane otrzymamy dzięki programowi **iptraf**.

Wygodnym sposobem śledzenia wybranego ruchu sieciowego jest użycie regułek linuksowego filtra pakietów. Do śledzenia ruchu służy cel "-j LOG" np.:

```
# iptables -A INPUT -p TCP --dport 80  -s 10.0.0.3 -j LOG
```

Powyższy wpis doda do łańcucha INPUT regułkę rejestrującą połączenia TCP z hosta 10.0.0.3 na port 80 danej maszyny. Aby te regułki działały z wpisami odrzucającymi pakiety (DROP/REJECT) muszą być analizowane wcześniej, w przeciwnym wypadku pakiet nigdy nie dotrze do regułki rejestrującej. W przypadku dopasowania pakietu do regułki następuje odnotowanie tego zdarzenia, wpisy te można odczytywać za pomocą programu **dmesg**, lub z plików syslog-a - zwykle z */var/log/kernel* i */var/log/messages*.

Dużo bardziej szczegółowe informacje o ruchu sieciowym otrzymamy dzięki programowi **tcpdump**, jest to rozbudowany sniffer i program do analizy pakietów.

http://www.ubuntugeek.com/iproute2-networking-and-traffic-control-tools.html