

## Konfiguracja sieci w systemie Linux Debian

Jedną z najważniejszych funkcjonalności dla serwera jest oczywiście połączenie z siecią lokalną/rozległą. System Linux od samego początku budowany był z myślą o sieci – zaraz przy starcie systemu inicjowane jest połączenie sieciowe, ustalone adresy IP itp.; dzieje się tak dlatego, że już w samo jądro wkompileowany jest sterownik karty sieciowej. Rozwiązanie to jest zupełnie inne niż w systemie Microsoftu – tam połączenie sieciowe jest inicjowane na wyraźne życzenie usług/aplikacji zainstalowanych w systemie (tzw. Winsocket) pod warunkiem, że w systemie jest zainstalowany odpowiedni sterownik komunikujący się z jądrem systemu.

Sama konfiguracja sieci różni się w zależności od posiadanej wersji systemu (np. plik konfiguracji w systemie Red Hat będzie różnił się od konfiguracji w Debianie) czy też zainstalowanych pakietów (przykładowo system Linux z nakładką graficzną posiada niezależną konfigurację sieci tylko dla niej; odpowiada za nią pakiet NetworkManager).

W naszym wypadku dostęp do pliku konfiguracyjnego:

```
/etc/network/interfaces
```

gdzie interfaces to właśnie plik konfiguracyjny.

Otwieramy go naszym ulubionym edytorem tekstowym (np. nano). Jego zawartość może być różna, w zależności ile posiadamy kart sieciowych. Przykładowy zrzut z naszej wersji Debiana:

```
GNU nano 2.2.6          Plik: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp

# Secondary network interface
auto eth1

iface eth1 inet dhcp

[ Wczytano 15 linii ]
^G Pomoc      ^O Zapisz     ^R Wczyt.plik ^Y Poprz.str. ^K Wytnij     ^C Bież.poz.
^X Wyjdź     ^J Wyjustuj  ^W Wyszukaj  ^V Nast.str. ^U Wklej     ^T Pisownia
```

Proszę zauważyć, że tutaj, podobnie jak przy repozytoriach, znak „#” oznacza linię komentarza. Cokolwiek nie znajdzie się w danej linii po tym znaku nie będzie liczone do konfiguracji!

Pierwsze dwie linie komentarza wyjaśniają zastosowanie edytowanego pliku oraz podpowiadają gdzie w systemie można uzyskać więcej informacji - dokładnie trzeba w linii poleceń wpisać

```
man interfaces
```

Pierwszym konfigurowanym interfejsem sieciowym jest... pętla zwrotna. Należy ona do interfejsów wirtualnych chociaż w tym konkretnym wypadku skonfigurowano ją jako interfejs fizyczny (jak większość interfejsów zresztą). W systemach Unix/Linux pętla zwrotna posiada nazwę lo.

**WAŻNE:** Słowo 'auto' oznacza, że konfigurowujemy interfejs fizyczny. Wymienione po nim nazwy

interfejsów (można wpisać kilka nazw oddzielonych spacjami) będą domyślnie podnoszone przez system; warunkiem jest posiadanie tychże kart sieciowych

Następna linia, zaczynająca się od **iface**, określa konfigurację danego interfejsu (w tym przypadku lo). Słowo **inet** oznacza, że konfigurujemy protokół IPv4 (dla IPv6 byłoby **inet6**). Słowo **loopback** oznacza, że konfiguracja dotyczy pętli zwrotnej (przydzielenie adresu 127.0.0.1).

Przyjrzyjmy się teraz dwóm liniom pod komentarzem

```
#The primary network configuration
```

Przy interfejsie eth0 nie ma słowa auto a jednak jest przez nasz system widziany i skonfigurowany. Tutaj wykorzystana została komenda **allow-hotplug**. Pozwala ona na skonfigurowanie interfejsu w przypadku gdy zostanie on podłączony do komputera (wykryty przez jądro systemu); w innym wypadku jego konfiguracja zostanie pominięta. Parametr pozwala na dodanie wielu interfejsów w ramach jednego polecenia lecz konfiguruje tylko jeden interfejs, resztę traktując jako kopię zapasową. Przykładowo gdybyśmy zadeklarowali interfejs eth0 oraz eth1 to konfigurowanym w pierwszej kolejności interfejsem byłby eth0. Jednak gdyby coś się z nim stało to został by on przez system wyłączony, a podłączonym interfejsem byłby eth1.

Zakładając, że zawsze będziemy mieli ten interfejs podłączony możemy zamienić tę komendę na **auto eth0**.

**INFORMACJA:** Demon sieciowy nie zna komendy **auto** lecz **allow-auto** jednak dla wygody administratora wymyślono jej synonim.

Podstawową zmianą w drugiej linii opisywanego przypadku jest pojawienie się wyrazu **DHCP**. Informuje ona demona sieciowego iż ma żądać dla tego interfejsu ustawień sieciowych (adres, brama, DNS) od serwera/serwerów DHCP znajdujących się w obrębie sieci lokalnej/rozległej.

W ten oto sposób dokonywana jest podstawowa konfiguracja sieci w systemie Linux. Jednak zdarza się, że jako administratorzy będziemy musieli skonfigurować dodatkowy interfejs sieciowy. W tym momencie można by pomyśleć by dodać dodatkowy interfejs do linii

```
allow-hotplug eth0 eth1
```

I w zasadzie byłoby to poprawne. Z tym, że system skonfiguruje pierwszy zgłoszony przez jądro interfejs, resztę pozostawiając nieskonfigurowaną (przykład powyżej!). Ponieważ nie jest to dla nas korzystne dodajemy nowe linie do ustawień (drugi komentarz):

```
auto eth1  
iface eth1 inet dhcp
```

Dzięki temu nasz drugi interfejs NA PEWNO zostanie przez system podniesiony (w przypadku braku fizycznego interfejsu otrzymamy komunikat błędu/ostrzeżenia przy wczytywaniu konfiguracji).

Teraz jednak chcielibyśmy, aby eth1 miał statyczny adres ponieważ ma on być interfejsem serwera w sieci lokalnej (interfejs eth0 zapewnia serwerowi dostęp do sieci rozległej). Założmy, że komputer ma posiadać adres 192.168.1.70/24, bramę 192.168.1.254 a DNS 8.8.8.8 oraz 192.168.1.254. Konfiguracja będzie wyglądać następująco:

```
#Secondary network interface
auto eth1
iface eth1 inet static
address 192.168.1.70
netmask 255.255.255.0
gateway 192.168.1.254
dns-nameservers 8.8.8.8 192.168.1.254
```

Całość takiej konfiguracji będzie wyglądać zatem następująco:

```
GNU nano 2.2.6      Plik: /etc/network/interfaces      Zmieniony
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp

# Secondary network interface
auto eth1
iface eth1 inet static
address 192.168.1.70
netmask 255.255.255.0
gateway 192.168.1.254
dns-nameservers 8.8.8.8 192.168.1.254

^G Pomoc      ^O Zapisz      ^R Wczyt.plik ^Y Poprz.str.  ^K Wytnij      ^C Bież.poz.
^X Wyjdź      ^J Wyjustuj   ^W Wyszukaj   ^V Nast.str.  ^U Wklej      ^T Pisownia
```

**WAŻNE!!** Proszę nie stosować **PODANEJ KONFIGURACJI NA ZAJĘCIACH!** Może to spowodować nieokreślone konflikty sieciowe! W celu zastosowania konfiguracji proszę spojrzeć do **ZADAŃ!**

Aby zrestartować działanie sieci należy podać komendę:

```
/etc/init.d/networking restart
```

Następnie konfigurację sprawdzić poleceniem

```
ifconfig
```

Powinniśmy uzyskać informacje podobne do tych na zrzucie (przedstawiono fragment):

```
eth0      Link encap:Ethernet  HWaddr 08:00:27:fd:cc:3f
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:febd:cc3f/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2  errors:0  dropped:0  overruns:0  frame:0
          TX packets:8  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0  txqueuelen:1000
          RX bytes:1180 (1.1 KiB)  TX bytes:1152 (1.1 KiB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:b4:61:49
          inet addr:192.168.1.70  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:feb4:6149/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

## RĘCZNE USTAWIANIE KONFIGURACJI

System pozwala także na ręczne ustawianie adresu sieci jednorazowo – na chwilę bądź do restartu systemu. Służy do tego wspomniane wcześniej polecenie **ifconfig**. Dzięki niemu możemy np. wyłączać lub włączać wskazany interfejs, zmieniać adres i maskę sieci czy też zmieniać pewne opcje interfejsu. Możliwości jest bardzo dużo dlatego opisane zostaną tylko najczęściej wykorzystywane opcje, o reszcie można dowiedzieć się z podręcznika polecenia (**man ifconfig**).

a) sprawdzenie aktualnej konfiguracji dla danego interfejsu (eth1):

```
ifconfig eth1
```

b) wyłączenie interfejsu eth1

```
ifconfig eth1 down
```

c) włączenie

```
ifconfig eth1 up
```

d) zmiana adresu IP wraz ze zmianą maski oraz MTU (Maximum Transmission Unit)

```
ifconfig eth1 192.168.1.71 netmask 255.255.2550 mtu 1455
```

e) zmiana adresu MAC:

```
ifconfig eth1 hw ether 00:12:27:b5:61:50
```

**WAŻNE!** Zmiana sprzętowego adresu (MAC) powinno dokonywać się tylko w uzasadnionych przypadkach! Zmiany w celu np. wykonywania operacji sieciowych w „imieniu” innego komputera mogą być traktowane jak przestępstwo.

Zmiany bramy dokonuje się poprzez inne polecenie – **route**. Wpisując polecenie otrzymamy tabelę z aktualną tablicą routingu. Jeżeli zmienimy adres IP i będziemy chcieli zmienić domyślną bramę na inną musimy wydać stosowne polecenie:

```
route add default gw 192.168.1.254
```

Od tego momentu naszą domyślną bramą wyjściową będzie podany wyżej adres (serwera bądź routera).

Zmiana ustawień DNS najszybciej dokona się poprzez plik **/etc/resolv.conf**. Wystarczy w plik wpisać dowolne adresy, z jakich chcemy skorzystać (otwieramy poprzez polecenie **nano /etc/resolv.conf**):

```
nameserver 8.8.8.8  
nameserver 8.8.4.4
```

Po zapisaniu adresy te będą natychmiast zaakceptowane przez serwer. Trzeba jednak pamiętać, że ulegną one zmianie zaraz po restarcie systemu (jak zresztą wszystkie ustawienia z tego rozdziału). Więcej o adresach DNS - <https://wiki.archlinux.org/index.php/resolv.conf>

## TWORZENIE WIELU INTERFEJSÓW SIECIOWYCH W RAMACH JEDNEGO ADRESU FIZYCZNEGO

System Linux posiada jeszcze jedną, nad wyraz użyteczną możliwość – tworzenie w ramach jednego interfejsu jednego bądź kilku interfejsów wirtualnych. Dzięki temu możemy odseparowywać poszczególne sieci od siebie, filtrować ruch pomiędzy kilkoma podsieciami a nawet, w dalszej perspektywie, umożliwiać wielu podsieciom uzyskiwania osobnych adresów IP.

Założmy, że chcemy uzyskać dwa interfejsy sieciowe w ramach karty eth1. Pierwsza wirtualna karta ma posiadać aktualną konfigurację, druga ma być serwerem i bramą dla podsieci 192.168.100.0/24. Plik konfiguracyjny musi się więc zmienić z:

```
#Secondary network interface
auto eth1
iface eth1 inet eth1
address 192.168.1.70
netmask 255.255.255.0
gateway 192.168.1.254
dns-nameservers 8.8.8.8 192.168.1.254
```

na

```
#Secondary network interface
auto eth1:0
iface eth1:0 inet static
address 192.168.1.70
netmask 255.255.255.0
gateway 192.168.1.254
dns-nameservers 8.8.8.8 192.168.1.254
```

```
auto eth1:1
iface eth1:1 inet static
address 192.168.100.1
netmask 255.255.255.0
broadcast 192.168.100.255
```

Oczywiście polecenie broadcast jest opcjonalne (ten adres jest przeważnie rozgłoszeniem, ale można poinformować system o tym fakcie). I... to wszystko! Po restarcie interfejsów (polecenie w pierwszym rozdziale) wszystko będzie działać.

**UWAGA!** Interfejsy wirtualne będą ograniczane przepustowością fizycznego interfejsu. Dlatego tego typu operację powinno stosować się jedynie w konkretnych wypadkach, wymagającego tego podejścia.

ZADANIA (OPIS I WYKONANIE):

- 1) W jaki sposób dokonywane jest wczytanie konfiguracji z pliku /etc/network/interfaces ?
- 2) Czy istnieje możliwość stałego dodania konkretnej trasy pakietów (route) bez zapisywania jej w konfiguracji pliku interfaces?
- 3) W jaki sposób można zapobiec zamianie adresów DNS po restarcie systemu bez wpisywania ich statycznie w pliku interfaces.
- 4) Jakie użyteczne opcje posiada polecenie ifconfig oraz route nie wymienione w tekście?
- 5) Proszę przetestować ustawienia wirtualnych interfejsów sieciowych (w parach). Opisać

konfigurację oraz pozyskane rezultaty.

**DLA INTERFEJSÓW NALEŻY USTAWIAĆ ADRESY Z KLASY A (10.0.0.0/8), z czego trzeci i/lub drugi człon powinien posiadać numer z dziennika. Ma to zapobiec konfliktom sieciowym!**