

Sieci komputerowe

Kamil NOWAK

Katedra Informatyki

Wydział Informatyki i Zarządzania Politechniki Wrocławskiej

kamil.nowak@pwr.edu.pl

Network layer of the ISO-OSI model

- IPv4 and IPv6
- Configuration in MS Windows and Linux
- ICMP protocol
- DNS protocol
- Support services: ping, traceroute, nslookup

Version 0.1-190325

Słowa kluczowe

IPv4, IPv6, adresacja IPv4, adresacja IPv6, maska, podsieć, adres sieci, adres hosta, adres broadcast, adres prywatny, adres publiczny, adres multicast, VLSM, CIDR, adresacja klasowa, adresacja bezklasowa

Wprowadzenie

Najbardziej powszechnym i podstawowym protokołem sieciowym występującym na świecie jest protokół IP (Internet Protocol). Protokół ten znajduje się w trzeciej warstwie modelu ISO-OSI lub drugiej modelu TCP/IP. Popularna nazwa globalnej sieci – Internet wywodzi się od nazwy tego protokołu. Pierwowzór sieci Internet – sieć ARPANET powstał 1969r. na Uniwersytecie Kalifornijskim w Los Angeles (UCLA). Był to eksperymentalny projekt testowany dla potrzeb wojska, który miał funkcjonować pomimo uszkodzeń pewnej części sieci. Sieć testowa przekształciła się wkrótce w sieć użytkową. Do sieci dołączają się kolejne uniwersytety i użytkownicy. Stworzenie standardu prezentacji informacji html i serwisów WWW sprawiło, że sieć zaczęła się rozrastać w zaskakującym tempie. Początkowo sieć bazowała na adresacji klasowej. Szybko okazało się, że przewidziana z ogromnym nadmiarem pula adresów zaczyna się wyczerpywać. W celu lepszego zagospodarowania dostępnych adresów stworzono adresację bezklasową, wydzielono pule adresów prywatnych i zaczęto powszechnie wykorzystywać usługę NAT. Dla coraz szybciej rozrastającego się Internetu było to niewystarczające, dlatego zaczęto opracowywać nowe wersje tego protokołu. Obecna, stabilna wersja nosi nazwę IPv6. Protokół ten w niewyobrażalny sposób zwiększył pulę dostępnych adresów sieciowych oraz zlikwidował część wad poprzedniej wersji.

Spis treści

Warstwa sieciowa modelu ISO-OSI

Konfiguracja protokołów IPv4 i IPV6 w systemach MS Windows i Linux

Protokół ICMP

Protokół DNS

Usługi pomocnicze

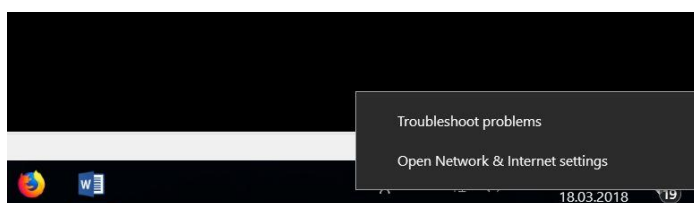
Konfiguracja protokołu IP.

Konfiguracja protokołu IP w systemie MS Windows

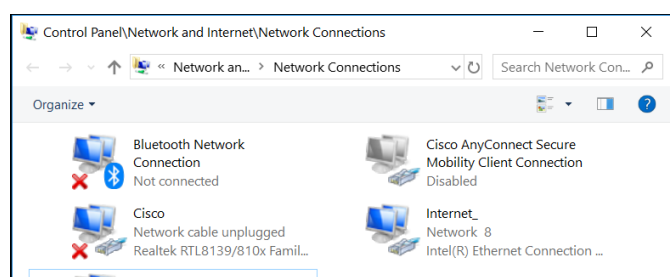
Konfiguracja w trybie GUI

Poniższe obrazy przedstawiają proces ustawiania konfiguracji IPv4 i IPv6 w systemie Windows.

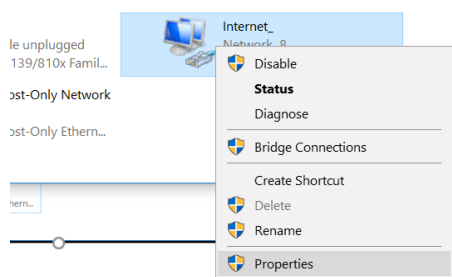
Uwaga: W zależności od wersji systemu Windows (stacja robocza, serwer, inna) oraz bieżącej aktualizacji poszczególne etapy i okna mogą się różnić od zamieszczonych poniżej.



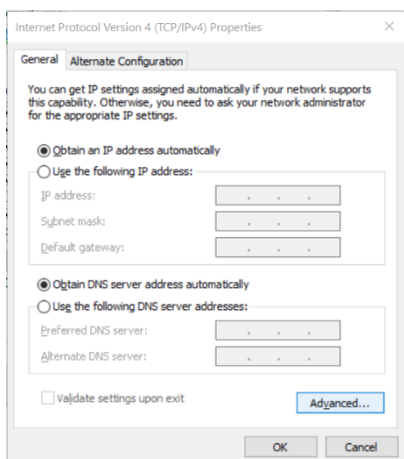
*Rysunek 1 - Konfiguracja protokołu IP w systemie Windows.
Szybkie dojście do opcji otwierającej konfigurację sieciową
w Windows.*



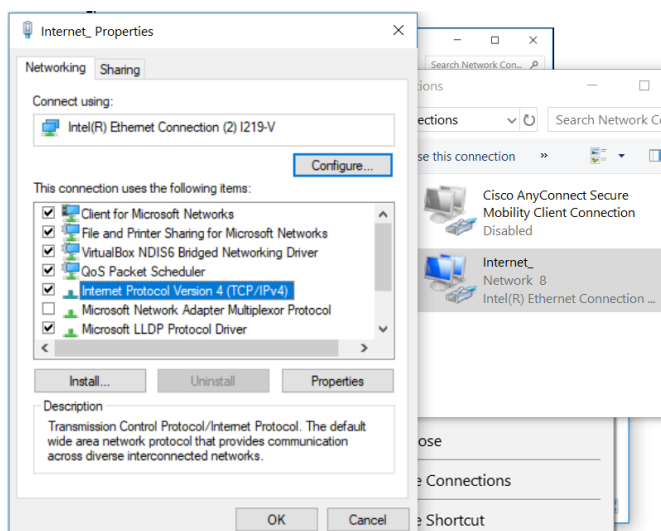
*Rysunek 2 - - Konfiguracja protokołu IP w systemie Windows.
Okno z interfejsami sieciowymi.*



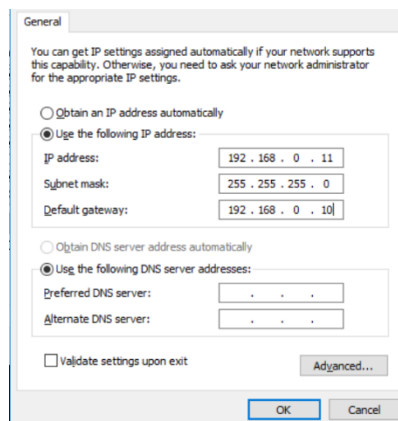
Rysunek 3 - - Konfiguracja protokołu IP w systemie Windows. Wybór właściwości karty sieciowej.



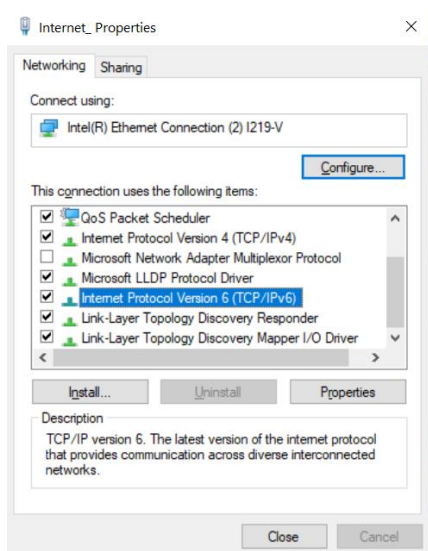
Rysunek 4 - Konfiguracja protokołu IP w systemie Windows. Wybór opcji automatycznej konfiguracji przez serwer DHCP.



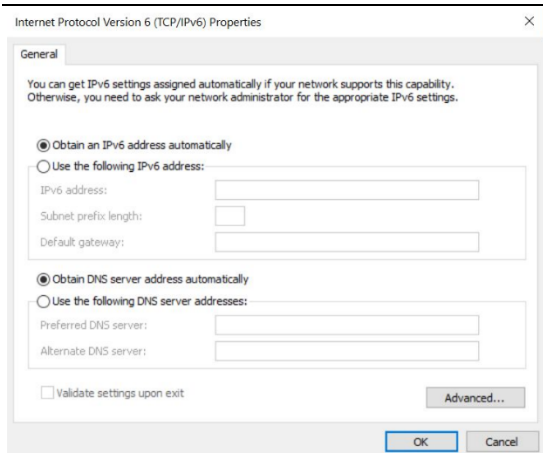
Rysunek 5 - Konfiguracja protokołu IP w systemie Windows. Wybór właściwości protokołu IPv4.



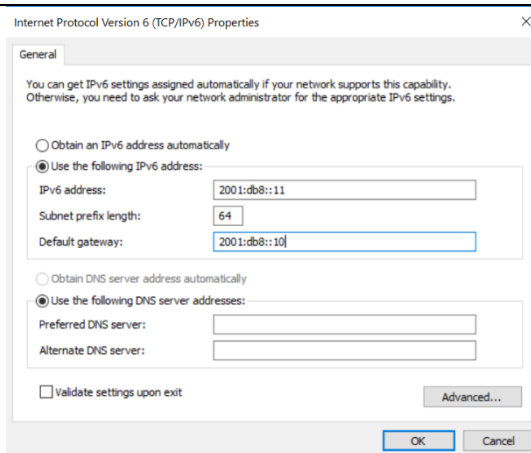
Rysunek 6 - Konfiguracja protokołu IP w systemie Windows. Konfiguracja ręczna adresu IPv4.



Rysunek 7 - Konfiguracja protokołu IP w systemie Windows. Wybór właściwości protokołu IPv6.



Rysunek 9 - Konfiguracja protokołu IP w systemie Windows. Wybór opcji automatycznej konfiguracji adresu IPv6.



Rysunek 8 - Konfiguracja protokołu IP w systemie Windows. Konfiguracja manualna adresu IPv6.

Aby sprawdzić poprawność konfiguracji protokołów IP w systemie Windows. Można skorzystać z interfejsu graficznego lub z komendy `ipconfig` w oknie `cmd` (Command Line) Druga opcja jest pewniejsza, gdyż interfejs graficzny czasami może pokazać adres niezatwierdzony przez system. Komendę można wykonać z dodatkowymi parametrami:

- `ipconfig /all` - pokazuje dostępne opcje
- `ipconfig /all` - pokazuje dodatkowe informacje konfiguracyjne dotyczące np. adresu MAC, czy konfiguracji IPv6.
- `ipconfig /release` - zwalnia adres IP przydzielony dynamicznie.
- `ipconfig /renew` - ponownie pobiera adres IP na interfejsach skonfigurowanym dynamicznie.

```

C:\Users\tmp>ipconfig

Windows IP Configuration

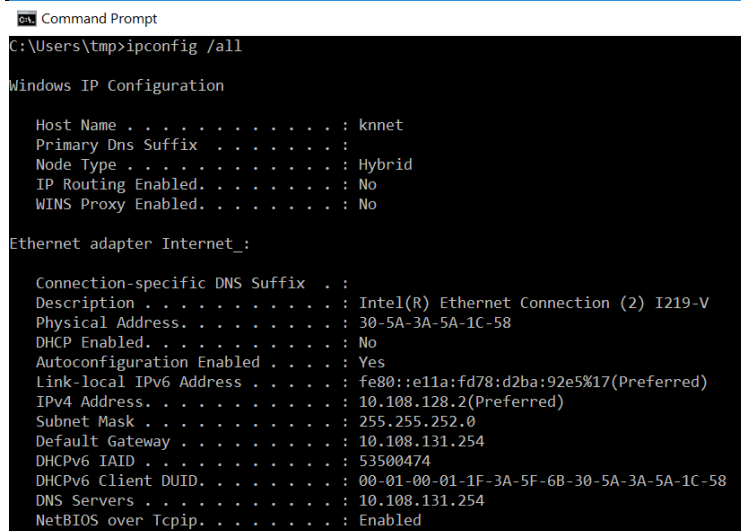
Ethernet adapter Internet_:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::e11a:fd78:d2ba:92e5%17
    IPv4 Address. . . . . : 10.108.128.2
    Subnet Mask . . . . . : 255.255.252.0
    Default Gateway . . . . . : 10.108.131.254

Ethernet adapter Cisco:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
  
```

Rysunek 10 - Weryfikacja konfiguracji adresu IP komendą `ipconfig`.



```

C:\Users\tmp>ipconfig /all

Windows IP Configuration

Host Name . . . . . : knnet
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter Internet_:

Connection-specific DNS Suffix . :
Description . . . . . : Intel(R) Ethernet Connection (2) I219-V
Physical Address. . . . . : 30-5A-3A-5A-1C-58
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::e11a:fd78:d2ba:92e5%17(Preferred)
IPv4 Address. . . . . : 10.108.128.2(Preferred)
Subnet Mask . . . . . : 255.255.252.0
Default Gateway . . . . . : 10.108.131.254
DHCPv6 IAID . . . . . : 53500474
DHCPv6 Client DUID. . . . . : 00-01-00-01-1F-3A-5F-6B-30-5A-3A-5A-1C-58
DNS Servers . . . . . : 10.108.131.254
NetBIOS over Tcpip. . . . . : Enabled

```

Rysunek 11 - Zawansowana weryfikacja konfiguracji karty sieciowej komendą `ipconfig /all`.

Konfiguracja w trybie CLI

W systemie Windows adres IP można ustawić również w oknie command line (cmd). Należy użyć narzędzia *netsh*. Można go używać w trybie wsadowym (wpisując wszystkie opcje w linii zaraz po poleceniu *netsh*) lub jak shelu konfiguracyjnego (uruchamiamy *netsh*, a następnie poruszamy się po różnych poziomach tego skryptu).

Tryb interaktywny

```
netsh
netsh>/?
```

Tryb wsadowy

Konfiguracja adresu statycznego na karcie Cisco

Konfiguracja pełna

```
netsh interface ipv4 set address name="Cisco" source=static address=192.168.0.X1
mask=255.255.255.0 gateway=192.168.0.X0
```

Konfiguracja skrócona

Satyczna konfiguracja adresu IP

```
netsh in ip set address "Cisco" static 192.168.0.X1 255.255.255.0 192.168.0.X0
```

```
netsh in ipv4 set dnsservers "Cisco" static 192.168.0.X0 primary
```

Dynamiczna konfiguracja adresu IP (DHCP na karcie Internet)

```
netsh in ipv4 set address name="Internet" dhcp
```

```
netsh in ip set dns "Internet" dhcp
```

Instalacja, uruchamianie, podstawowa obsługa systemu Linux

Instalacja

Istnieje wiele różnych dystrybucji systemu Linux. Wersją, którą można nazwać odpowiednią do tworzenia stabilnych routerów programowych jest dystrybucja Linux Debian. W dystrybucji tej wszystkie usługi są przez długi okres poddawane testom zanim wejdą do tak zwanej dystrybucji stabilnej. Urządzenie sieciowe nie musi posiadać najnowszych wersji aplikacji użytkowych

Dla celów laboratorium sieciowego wystarczające jest zainstalowanie systemu Linux z najmniejszego rodzaju obrazu. Następnie należy doinstalować podstawowe usługi i aplikacje. Może się również przydać środowisko

graficzne. Nie trzeba jednak instalować standardowych środowisk (Gnome czy KDE), wystarczy minimalistyczny menadżer okien. W Debianie jest dostępny „lekki” menadżer **LXDE**, można jednak nie instalować żadnego z menadżerów domyślnych tylko zainstalować jakiś minimalistyczny WM typu **Fluxbox**.

Dla potrzeb instalacji sieciowej (minimalnej) można pobrać plik:

<http://ftp.nl.debian.org/debian/dists/stretch/main/installer-amd64/current/images/netboot/mini.iso>

Szczegółowa dokumentacja znajduje się na stronie (23.03.2019):

<https://www.debian.org/doc/>

Uruchamianie

W laboratorium system Linux powinien być zainstalowany i gotowy do użycia. System zainstalowany jest na oddzielnej partycji. W celu jego uruchomienia należy zresetować programowo komputer i wybrać system Linux w boot loaderze przy starcie komputera.

Istnieją dwie możliwości uruchomienia systemu. Jeżeli istnieje zainstalowany menadżer okien wraz ze środowiskiem graficznym, system najprawdopodobniej uruchomi się w trybie graficznym. Jeżeli to środowisko nie zostało zainstalowane, system uruchomi się w trybie terminala tekstowego.

Dla potrzeb laboratorium sieciowego studenci powinni posługiwać się głównie terminalem tekstowym. Dlatego po zalogowaniu się w trybie graficznym, należy uruchomić terminal tekstowy i w nim pracować. Środowisko graficzne na pewno przyda się do uruchomienia programu Wireshark.

Terminal graficzny można pominąć już przy starcie lub w trakcie pracy. Należy wcisnąć skrót klawiszy:

Alt +F1 lub CTRL+Alt+F1 lub Alt+1

W zależności od środowiska (wirtualne/rzeczywiste/graficzne/tekstowe) i systemu operacyjnego konieczne może być użycie różnego rodzaju skrótów.

Logowanie do systemu Linux

Nie wchodząc zbytnio w szczegóły, aby móc przeprowadzić konfigurację w systemie Linux użytkownik powinien posiadać prawa administratora (root). W celach bezpieczeństwa zaleca się przeważnie zablokowanie bezpośredniej możliwości logowania do systemu na konto root. Logujemy się wówczas na konto zwykłego użytkownika, a następnie wykonujemy polecenia z prawami administratora, lub zmieniamy właściwości naszej sesji i logujemy się jako root.

Aktualizacja i instalacja aplikacji

Po zalogowaniu do systemu na konto administratora (root) powinniśmy zaktualizować system i zainstalować brakujące aplikacje. Wykonujemy następującą sekwencję komend:

`apt-get update` – aktualizacja archiwum pakietów.

`apt-get install aptitude` – instalacja programu aptitude, który lepiej niż apt-get zarządza instalacją pakietów.

`aptitude -full-upgrade` – aktualizacja systemu.

`aptitude install -R mc` – instalacja tekstowego menadżera plików.

`aptitude install -R xorg` – instalacja serwera graficznego X Window.

`aptitude install -R eterm` – instalacja dodatkowego terminala graficznego.

`aptitude install -R fluxbox` – instalacja „lekkiego” menadżera okien.

`aptitude install -R Wireshark` – instalacja programu Wireshark.

Aplikacja mc (Midnight Commander)

Aplikacja przypomina wyglądem i działaniem popularny w systemach DOS program Norton Commander, którego wersja windowsowa nosi nazwę Total Commander.

Większość działań wykonujemy za pomocą klawiszy funkcyjnych F1 – F9. W celu wejścia do menu konfiguracyjnego należy użyć klawisza F9. W menu można zmienić domyślny edytor tekstowy na własny (MC).

Podstawowe znaczenie klawiszy:

F2 – zmiana nazwy pliku.

F3 – podgląd zawartości pliku.

F4 – edycja pliku.

F5 – kopiowanie pliku.

F6 – przeniesienie pliku.

F8 – kasowanie pliku.

F9 – wejście do menu.

Poniżej przykładowe zrzuty ekranu z aplikacji Midnight Commander.

```

Left      File      Command      Options      Right
<- ~      .[^]>        <- /etc/network .[^]>
.n      Name      Size      Modify      time      .n      Name      Size      Modify      time
/..      UP--DIR      mar 23 19:01      /..      UP--DIR      mar 23 20:42
/.cache  4096      mar 23 20:42      /if-down.d  4096      mar 23 18:59
/.config 4096      mar 23 20:42      /if-post-down.d 4096      sty 30 2017
/.local  4096      mar 23 20:42      /if-pre-up.d  4096      sty 30 2017
.bashrc  570      sty 31 2010      /if-up.d      4096      mar 23 20:31
.profile 148      sie 17 2015      /interfaces.d 4096      sty 30 2017
                                     interfaces  317      mar 23 18:59

/.cache 3128M/3967M (78%)      interfaces 3128M/3967M (78%)

```

Porada: cytowanie znaku można uzyskać przez Ctrl-q i\ odpowiedni znak.

root@debian9-0:/etc/network# _

1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit

```

Left      File      Command      Options      Right
<- ~      .[^]>        <- /etc/network .[^]>
.n      Name      Size      Mod          Configuration...
/..      UP--DIR      mar 23 20:42      Layout...
/.cache  4096      mar 23 18:59      Panel options...
/.config 4096      mar 23 18:59      Confirmation...
/.local  4096      mar 23 18:59      Appearance...
.bashrc  570      sty 30 2017      Display bits...
.profile 148      sie 17 2015      Learn keys...
                                     Virtual FS...
                                     Save setup

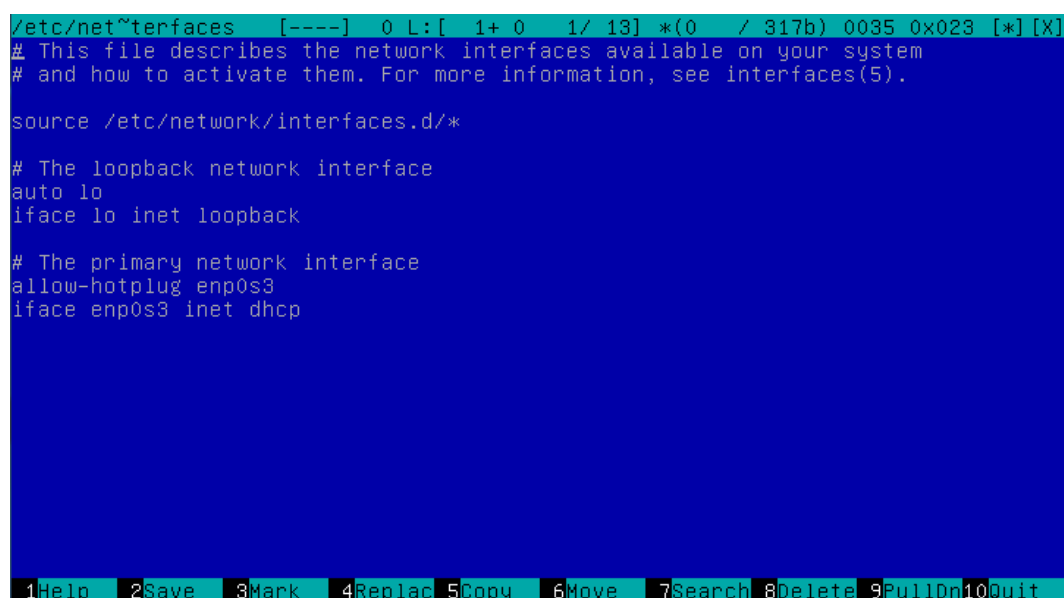
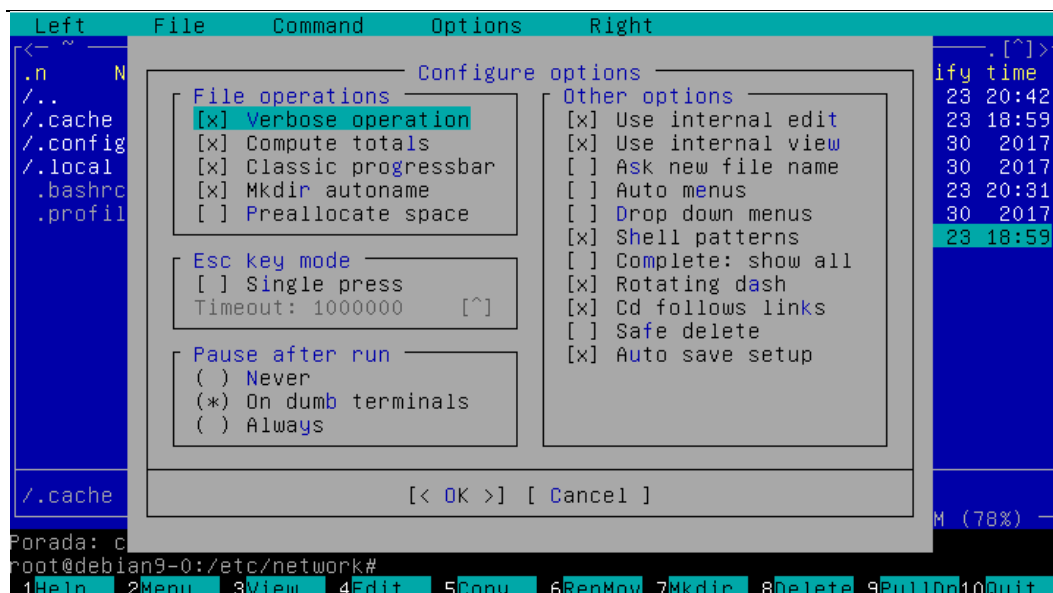
/.cache 3128M/3967M (78%)      interfaces 3128M/3967M (78%)

```

Porada: cytowanie znaku można uzyskać przez Ctrl-q i\ odpowiedni znak.

root@debian9-0:/etc/network# _

1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit



Konfiguracja adresu IP w systemie Linux

Uwaga: Konfiguracja protokołu IP może zależeć od rodzaju systemu Linux oraz jej wersji.

Podobnie jak w systemie MS Windows, w systemie Linux można skonfigurować adres IP w trybie GUI lub CLI. Administratorzy systemu konfigurują przeważnie protokół edytując tekstowy plik konfiguracyjny. Tryb graficzny nie jest polecany ze względu na różnorodność nakładek graficznych i ich nieprecyzyjne działanie. W trybie tekstowym wszystko powinno zadziałać poprawnie, jednak tu również czekają na użytkownika niespodzianki. W różnych dystrybucjach Linuxa adres IP można konfigurować w różnych miejscach (różnych plikach) i w różny sposób.

W obecnych wersjach Linuxa (2019) domyślnym narzędziem do konfiguracji sieci jest **iproute2**.

Ma ono zastąpić popularny wśród administratorów pakiet **net-tools** (**ifconfig**, **vconfig**, **route**, **arp** etc.). Wiele osób wciąż jest jednak przyzwyczajonych do używania poprzednich komend i nazw interfejsów: **ifconfig**, **eth0**, **eth1**, ... Obecne nazewnictwo interfejsów uwzględnia położenie karty sieciowej w slotcie komputera.

Pierwszemu interfejsowi **eth0** odpowiada teraz nazwa **enp3s0**.

Edycja konfiguracji w tekstowym pliku konfiguracyjnym

Poniżej podany jest przykład konfiguracji dla systemu Linux Debian. Konfigurację sieciową zmieniamy w pliku `/etc/network/interfaces`

```
/etc/network/interfaces
#Automatic activation during startup
auto lo eth0

#Configuration of loopback interface
iface lo inet loopback

#Configuration of static IPv4 address
iface enp3s0 inet static
address 192.168.0.X2
network 192.168.1.0
netmask 255.255.255.0
broadcast 192.168.1.255
gateway 192.168.0.X0
dns-nameservers 156.17.5.2

#Configuration of dynamic IPv4 address
auto enp3s0
iface enp3s0 inet dhcp
```

Konfiguracja w trybie CLI (terminal tekstowy)

W wersji Linux Debian 9, podstawowym narzędziem konfiguracji jest narzędzie **iproute2**. Poniżej sposób użycia komendy **ip** wchodzącej w skład tego pakietu.

Podstawowe komendy:

```
ip address show [dev enp3s0]

ip addr del dev enp3s0 192.168.0.13/24

ip address add 192.168.0.10/24 dev enp3s0

ip link show
ip link list

ip link set dev enp3s0 down

ip link set dev enp3s0 up

ip link set dev enp3s0 name lan0

ip route
ip route show
ip route list

ip route add 192.168.2.0/24 via 10.1.2.2
ip route add 192.168.2.0/24 dev enp3s0

ip route add default via 10.1.2.2 [dev enp3s0]
ip route add default dev enp3s0
```

Aktywowanie i sprawdzenie ustawień

```
dhclient eth0
ifdown eth0
ifup eth0
ifdown eth1
ifup eth1
ifconfig
```

Literatura

Materiały dydaktyczne Akademii Cisco CCNA 1

Dokumentacja systemu Linux Debian, 23.03.2019: <https://www.debian.org/doc/>

Debian, konfiguracja sieci, 24.03.2019: <http://qref.sourceforge.net/Debian/reference/ch-gateway.pl.html>

Konfiguracja iproute2, 24.03.2019: <https://baturin.org/docs/iproute2/>

Laboratory tasks

Network layer of the ISO-OSI model

- IPv4 and IPv6
- Configuration in MS Windows and Linux
- ICMP protocol
- DNS protocol
- Support services: ping, traceroute, nslookup

Basic information

Variables and symbols

- X - the variable X corresponds to the device number, as well as the laboratory group number.
- Y – universal variable correspond to device number of any other group.
- K, L, M, N, ... - variables used interchangeably with the variables X and Y. Correspond to the devices and groups numbers.
- Note: The variables K, L, M, N, ... are used when the order of the connected devices is important. For example, when configuration on the R_K router differs from the R_L router configuration. The variable L should correspond to the number of the current device (current laboratory group number), variable K indicates the “left” device and M the “right” device.
- Variables X and Y are used when the order of the connected devices is not important. X means local device and Y means any remote device.
- A, B, C, ... - variables corresponding to the addresses of the networks created in the labs.
- [1.2.3.4] – the number of the CCNA lab corresponding to the current exercise.
- {Variable} - marks the required variable.
- [Variable] – marks an alternative variable.
- 2 points - task difficulty measure.
- COLORS - can be used to express some additional meaning:
 - # **red - means extremely important things and obligatory tasks;**
 - # orange - means additional/alternative tasks;
 - # gray - means content that can be omitted in standard mode;
 - # green – means additional information, explanations;
 - # blue – means information to note to be checked by the instructor.
- FONT
 - # **bold - highlighted information.**
 - # Courier New - configuration commands.

IP addressing and device names used in the exercise

PCs

Name: PC_X1

Net: LAN_X (Cisco)
IP PC_L1: 192.168.X.X1/24
fc00:X::X1/64
fe80::X1
Gateway: 192.168.X.X0
fc00:X::X0/64

Name: PC_X2
Net: LAN_X (Cisco)
IP PC_L2: 192.168.X.X2/24
fc00:X::X2/64
fe80::X2
Gateway: 192.168.X.X0
fc00:X::X0/64

Switch

Name: S_X
Net: LAN_X1
Interface: VLAN 1
IP: 192.168.X.X9/24
fc00:X::X9/64
fe80::X9

Native VLAN: VLAN 1
Gateway: 192.168.X.X0
fc00:X::X0/64

Router

Name: R_X
Net: LAN_X
Interface: Fa0/0 (Fa0/2/0, or another connected)
IP: 192.168.X.X0/24
fc00:X::X0/64
fe80::X0

Net: WAN_L1
Interface: Serial 0/0/0 DCE (Serial 0/3/0 or another connected)
IP: 10.L.M.1/30
fc00:L:M::1/64
fe80::1
clock rate: **128000**

Net: WAN_L2
Interface: Serial 0/0/1 DTE (Serial 0/1/0, 0/3/1 or another connected)
IP: 10.K.L.2/30
fc00:K:L::2/64
fe80::2
clock rate: not applicable

Net: WAN_L3

Interface:	Serial 0/3/0 DCE (DTE) (Serial 0/3/0 or another connected)
IP:	10.X.Y.1(2)/ 30 2001:db8:L:N::1(2)/64 fe80::1(2)
clock rate:	128000/not applicable, depend on topology used

Gateway: Loopback 1 (virtual interface)

X, L - the number of the current device.

K – the number of the previous device.

M – the number of the next device.

y – the number of the remote device.

Base topology of a single laboratory group

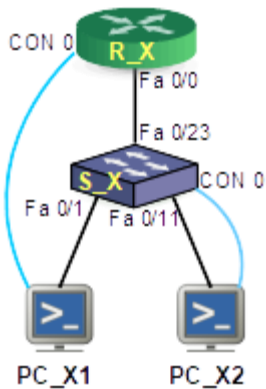


Figure 12 - Base topology of a single laboratory group.

PC_X1 intern NIC port - connected with a straight cable to S_X Fa 0/1 port.

PC_X1 serial COM port - connected with a serial cable to R_X console port.

PC_X2 intern NIC port - connected with a straight cable to S_X port Fa 0/11.

PC_X2 serial COM port - connected with a serial cable to S_X console port.

R_X port Fa 0/0 - connected with a straight cable to S_X Fa 0/23 port.



Configure the computers using the IP addresses given in the instruction. Set the static IP address on the internal NIC ("Cisco") that leads to the lab set. Set the dynamic IP address on the external NIC ("Internet"). Check the correctness of the configuration.

References to the CCNA 1 v. 6.0 labs

- [7.3.2.7] - Testing Network Connectivity with Ping and Traceroute
- [7.3.2.8] - Mapping the Internet
- [10.2.28] - Observing DNS Resolution

- [7.3.2.5] - Verifying IPv4 and IPv6 Addressing
- [7.3.2.6] - Pinging and Tracing to Test the Path
- [7.3.2.9] - Troubleshooting IPv4 and IPv6 Addressing
- [7.4.1.2] - Skills Integration Challenge

Note: tasks can be performed in two-person groups. Each task can be performed jointly or divided among group members.

Task 1 – Connectivity test in IPv4 network. ICMP protocol

Points: 1

Objectives

- IPv4 configuration under MS Windows system using *netsh* command.
- Connectivity test in IPv4 network using *ping* command.
- Analysis of the ICMP protocol in the Wireshark program.

Practical task

Administrator w firmie został poproszony o przeanalizowanie adresacji pod kątem nieprawidłowych adresów IPv4. W firmie występują błędy łączności. Z niektórych miejsc sieci Intranet nie można połączyć się z serwerami usług. Administrator dostał zadanie przeprowadzenia testów łączności i ich analizy w programie Wireshark.

Exercise scenario

1. Create a text document in the user's home directory (eg: C:\cisco\users\user.name) and [save the course of exercise](#) in it.
2. Configure the statics IP addresses on internal interfaces (Cisco) under MS Windows system. Use the addresses given in the instruction. Use the *netsh* command. [Save the exact configuration formula](#). Check the correctness of configuration using *ipconfig* command.
3. Start the Wireshark application on internal NIC (Cisco).
4. Make the connectivity test between PC_X1 and PC_X2 computers. Use the *ping* command.
5. Stop the Wireshark. [Save](#) the result in the user folder.
6. Find inside the Wireshark listing ICMP packets exchanged between PC_X1 and PC_X2. [Note](#) the frames IDs.
7. Analyze the IPv4 and ICMP headers. Find and [save](#) the following fields values:
 - [Source IPv4 address](#);
 - [Destination IPv4 address](#);
 - [Value of TTL field](#);
 - [Types and names of ICMP packets](#);
 - [Size and content of ICMP data field](#).
8. Make the connectivity test to google.com server. Enlarge the size of ping packet and set the non fragmentation bit.
[Check and save](#) the maximum size of ping packet.
9. Start the Wireshark on internal NIC. Capture the transmitted frames with increased size. [Save](#) the result in the user folder.
10. Analyse captured frames. Use the frame filter:
ICMP and ip.addr==google.com
[Find and save the values of the following fields](#):
 - [Source IPv4 address](#);
 - [Destination IPv4 address](#);
 - [Value of TTL field](#);
 - [Value of the non fragment bit](#);
 - [Types and names of ICMP packets](#);
 - [Maximal size and content of ICMP data field](#).

11. You can continue solving the next tasks or ask the instructor to check the results.

Task 2 – Connectivity test in IPv6 network. ICMPv6 protocol.

Points: 1

Objectives

- IPv6 configuration under MS Windows system using GUI interface.
- Connectivity test in IPv6 network using *ping* command.
- Analysis of the ICMPv6 protocol in the Wireshark program.

Practical task

Administrator w firmie został poproszony o przeanalizowanie adresacji pod kątem nieprawidłowych adresów IPv6. W firmie występują błędy łączności. Z niektórych miejsc sieci Intranet nie można połączyć się z serwerami usług. Administrator dostał zadanie przeprowadzenia testów łączności i ich analizy w programie Wireshark.

Exercise scenario

1. Create a text document in the user's home directory and [save the course of exercise](#) in it.
2. Configure the statics IPv6 addresses on internal interfaces (Cisco) under MS Windows system. Use the addresses given in the instruction. Check the correctness of configuration using command:


```
c:\>ipconfig /all
```
3. Start the Wireshark application on internal NIC (Cisco).
4. Make the connectivity test between PC_X1 and PC_X2 computers. Use the *ping* command.


```
PC_X1:
c:\>ping fc00:X::X2
PC_X2:
c:\>ping fc00:X::X1
```
5. Stop the Wireshark. [Save](#) the result in the user folder.
6. Find inside the Wireshark listing ICMPv6 packets exchanged between PC_X1 and PC_X2. [Note](#) the frames IDs.
7. Analyze the IPv4 and ICMP headers. Find and [save](#) the following fields values:
 - [Source IPv6 address;](#)
 - [Destination IPv6 address;](#)
 - [Value of TTL field;](#)
 - [Types and names of ICMPv6 packets;](#)
 - [Size and content of ICMPv6 data field.](#)
8. Make the connectivity test to PC_X2. Enlarge the size of ping packet and set the non fragmentation bit. [Check and save](#) the maximum size of ping packet.

[Note: Packet fragmentation in the IPv6 protocol works on a different way than in the IPv4 protocol.](#)
9. Start the Wireshark on internal NIC. Capture the transmitted frames with increased size. [Save](#) the result in the user folder.
10. Analyse captured frames. Use the frame filter:


```
ICMPv6 and ipv6.addr==fc00:X::X2
```

[Find and save the values of the following fields:](#)

- Source IPv6 address;
- Destination IPv6 address;
- Value of TTL field ?;
- Value of the non fragment bit ?;
- Types and names of ICMPv6 packets;
- Maximal size and content of ICMPv6 data field.

11. You can continue solving the next tasks or ask the instructor to check the results.
12. Skonfiguruj statycznie adresy IPv6 na interfejsach wewnętrznych („Cisco”) komputerów PC w systemie MS Windows. Użyj adresów podanych w instrukcji.
13. Uruchom program Wireshark. Przechwyć ramki na interfejsie wewnętrznym („Cisco”).

Task 3 – Tracking the path in IPv4 network. Traceroute (tracert) command

Points: 1

Objectives

- Configuring IPv4 address under MS Windows system using GUI interface.
- Analyzing routing paths to remote hosts using the traceroute (tracert) command.
- Capture traceroute packets in Wireshark.
- Analysis of the traceroute service.

Practical task

Administrator w firmie został poproszony o przeanalizowanie ścieżek routowania. Istnieje podejrzenie, że w firmie został źle skonfigurowany routing i pojawiły się pętle. Niektóre podsieci nie są dostępne z wybranych miejsc. Z kilku sieci zablokowany jest również dostęp do Internetu. Administrator postanowił przeanalizować ścieżki routowania przy pomocy aplikacji traceroute.

Exercise scenario

1. Create a text document in the user's home directory and [save the course of exercise](#) in it.
2. Configure the dynamic IPv4 addresses on external NIC (Internet) under MS Windows system. Use the GUI interface. Check the correctness of configuration using *ipconfig* command.
3. Start the Wireshark application on external NIC (Internet).
4. In MS Windows system trace the network path to the any chosen Web server using *tracert* command.
5. Stop the Wireshark. [Save](#) the result in the user folder.
6. Find inside the Wireshark listing *traceroute* packets exchanged between PC_X1 and Web server. [Note](#) the frames IDs.
7. Analyze the headers of first six sent *traceroute* packets. Find and [save](#) the values of the following fields:
 - Source IPv4 address;
 - Destination IPv4 address;
 - Value of TTL field;
 - Types and names of ICMP packets;
 - Size and content of ICMP data field.

8. Analyze the headers of first six received *traceroute* packets. Find and [save](#) the values of the following fields:
 - Source IPv4 address;
 - Destination IPv4 address;
 - Value of TTL field;
 - Types and names of ICMP packets;
 - Size and content of ICMP data field.
9. [Answer the question](#): Why do intermediate nodes respond to *traceroute* packets even though they are not the destinations of these packets?
10. You can continue solving the next tasks or ask the instructor to check the results.

Task 4 – Connectivity test in IPv4 network between Linux hosts. ICMP protocol

Points: 1

Objectives

- Configuration IPv4 addresses under Linux using CLI command: *iproute2*.
- Connectivity test between Linux system using *ping* command.
- Analysis of ICMP Linux packets in Wireshark application.

Practical task

Administrator w firmie został poproszony o przeanalizowanie adresacji pod kątem nieprawidłowych adresów IPv4. W firmie występują błędy łączności. Z niektórych miejsc sieci Intranet nie można połączyć się z serwerami usług. Administrator dostał zadanie przeprowadzenia testów łączności i ich analizy w programie Wireshark. Testy powinny objąć komputery z systemem Linux.

Exercise scenario

1. Create a text document in the user's home directory and [save the course of exercise](#) in it.
2. Start the computer with Linux Debian system. Login with root account. Use MS Windows Administrator password. To simplify the task, run PC_X1 in MS Windows and PC_X2 in Linux.
3. Check the operation of NICs. Try to use different command from *iproute2* packet.

```
ip address show [dev enp3s0]
ip link show
ip link list
```

```
ip link set dev {enp3s0} up
```

```
ip route
ip route show
ip route list
```

Alternative commands:

```
dhclient {eth0}
ifdown {eth0}
ifup {eth0}
```

```
ifdown {eth1}
ifup {eth1}
ifconfig
```

4. Check the update status and availability of the application.

```
apt-get update – updating the package archive.
apt-get install aptitude – installation of the aptitude program, the reliable packet manager.
aptitude -full-upgrade – system update.
aptitude install -R mc – instalation of CLI file manager.
aptitude install -R Wireshark – Wireshark installation.
```

5. Configure static IPv4 address on internal NIC in Linux Debian. Use the addresses given in the instruction. Configuration should be done inside the text terminal using **ip** command. **Save** the correct command formula. Check the configuration with command:

```
ip address show
```

6. In the Linux graphical mode start the terminal and Wireshark aplication in the background:

```
root#Wireshark &
```

Capture the frames on internal NIC.

7. Make the connectivity test between PC_X1 and PC_X2 using *ping* command.
8. Make the connectivity test to google.com Web Server.
9. Stop the Wireshark. Save the captured frames into user folder: /root/users/user_name

```
root#md users/user_name
```

10. Find in Wireshark listing ICMP packets exchanged between PC_X1 and PC_X2. **Save** the frames IDs.

11. Analyze the IPv4 and ICMP headers. Find and **save** the values of the following fields:

- Source IPv4 address;
- Destination IPv4 address;
- Value of TTL field;
- Types and names of ICMP packets;
- Size and content of ICMP data field.

12. Find and write down the differences in the structure of Linux and MS Windows ICMP packets.

13. You can continue solving the next tasks or ask the instructor to check the results.

Task 5 – Tracking the IPv4 network path from Linux OS. Traceroute command

Points: 1

Objectives

- Configuring IPv4 address under Linux Debian inside the network configuration file: /etc/network/interfaces.
- Analyzing routing paths to remote hosts using the traceroute command.
- Capture traceroute packets in Wireshark.
- Analysis of the traceroute service under Linux.

Practical task

Administrator w firmie został poproszony o przeanalizowanie ścieżek routowania. Istnieje podejrzenie, że w firmie został źle skonfigurowany routing i pojawiły się pętle. Niektóre podsieci nie są dostępne z wybranych miejsc. Z kilku sieci zablokowany jest również dostęp do Internetu. Administrator postanowił przeanalizować ścieżki routowania przy pomocy aplikacji traceroute. Testy mają zostać wykonane w systemach Linux.

Exercise scenario

1. Create a text document in the user's home directory and [save the course of exercise](#) in it.
2. Start the computer with Linux Debian system. Login with root account. Use MS Windows Administrator password. To simplify the task, run PC_X1 in MS Windows and PC_X2 in Linux.
3. Check the operation of NICs. Try to use different command from *iproute2* packet.

```
ip address show [dev enp3s0]
ip link show
ip link list
```

```
ip link set dev {enp3s0} up
```

```
ip route
ip route show
ip route list
```

Alternative commands:

```
dhclient {eth0}
ifdown {eth0}
ifup {eth0}
ifdown {eth1}
ifup {eth1}
ifconfig
```

4. Create the Linux network configuration inside the configuration file:
`/etc/networks/interfaces`
5. To edit the file, you can use the built-in text editor from the **mc** program. Configure the static IPv4 address on internal NIC (Cisco). Configure dynamic IPv4 address on external NIC (Internet). Use the addresses given in the instructions. [Save the created configuration](#). Reset the interfaces. Check the correctness of the new configuration with command:

```
ip address show
```

6. Start the Wireshark on external NIC.
7. On a Linux system, trace the route to the any chosen web server using the traceroute command.
8. Stop the Wireshark. Save the captured frames into the user folder:
`/root/users/user_name/file_name.txt`
9. Find inside the Wireshark listing *traceroute* packets exchanged between PC_X1 and Web server. **Note** the frames IDs. Use the frame filter.
10. Analyze the headers of first six sent *traceroute* packets. Find and [save](#) the values of the following fields:
 - Source IPv4 address;
 - Destination IPv4 address;
 - Value of TTL field;
 - Types and names of ICMP packets;

- Size and content of ICMP data field.

11. Analyze the headers of first six received *traceroute* packets. Find and [save](#) the values of the following fields:
 - Source IPv4 address;
 - Destination IPv4 address;
 - Value of TTL field;
 - Types and names of ICMP packets;
 - Size and content of ICMP data field.
12. [Write down the differences in the operation of the tracert \(MS Windows\) and traceroute \(Linux\) programs.](#)
13. [Answer the question:](#) Why do intermediate nodes respond to *traceroute* packets even though they are not the destinations of these packets?
14. Check the *mtr* program operation. Write down the differences in the operation of *traceroute* and *mtr*. In the absence of a program, try to install it:

```
aptitude install -R mtr
```
15. You can continue solving the next tasks or ask the instructor to check the results.

Zadania laboratoryjne w środowisku wirtualnym symulatora Packet Tracer

- Warstwa sieciowa modelu ISO-OSI
- Protokół IPv4 i IPv6
- Konfiguracja w systemie MS Windows i Linux
- Protokół ICMP
- Protokół DNS
- Usługi pomocnicze: ping, traceroute, nslookup

Środowisko programu Packet Tracer umożliwia analizę działania i konfigurację różnych urządzeń sieciowych. Konfiguracja IP jest możliwa w wirtualnym środowisku na różne sposoby:

- Przy użyciu graficznego interfejsu użytkownika
- Przy użyciu interfejsu CLI (Command Line Interface)