



universidade de aveiro
theoria poiesis praxis

DEPARTAMENTO DE ELECTRÓNICA, TELECOMUNICAÇÕES E INFORMÁTICA
LICENCIATURA EM ENG. DE COMPUTADORES E INFORMÁTICA

REDES DE COMUNICAÇÕES I

COMMAND GUIDE

CONFIGURE PCs NETWORK INTERFACES

BASIC NETWORKING COMMANDS

CISCO ROUTERS COMMAND LINE

Purpose

- How to configure IP parameters on the PCs interfaces
- Basic guide on basic networking tools: ping, traceroute and nslookup
- Basic list of Cisco Command Line instructions
- DELL Switch operation

Contents

1. Using ipconfig, ifconfig, and arp Commands	4
1.1. ipconfig (Windows)	4
Basic Commands:	4
Checking MAC Address:	4
1.2. ifconfig (Linux/macOS).....	5
Basic Commands:	5
Alternative (ip command for Linux):	5
1.3. arp (Address Resolution Protocol).....	6
1.3.1. Windows:	6
1.3.2. Linux/macOS:	6
2. Basic Networking Tools: Ping, Traceroute, and Nslookup	7
2.1. Ping Command.....	7
2.1.1. Windows: ping <IP_address_or_domain>	7
2.1.2. Linux/macOS: ping <IP_address_or_domain>	7
2.2. Traceroute Command.....	8
2.2.1. Windows:	8
2.2.2. Linux/macOS:	8
2.3. Nslookup Command	8
2.3.1. Windows/Linux/macOS:.....	8
Advanced Usage	8
3. Simple CLI Guide for Cisco IOS	10
3.1. Basic Command Line Navigation	10
3.2. Using Auto-Complete and "?" in Cisco CLI.....	10
3.3. Cisco prompt and levels.....	10
3.3.1. User EXEC Mode (Privilege Level 1)	10
3.3.2. Privileged EXEC Mode (Privilege Level 15)	10
3.3.3. Global Configuration Mode	10
3.3.4. Other Configuration Modes	11
3.4. Global Configuration Commands	11
3.4.1. Enabling IPv6 Routing.....	11
3.4.2. Allow the Use of Subnet Zero	11
3.4.3. Enable and Disable IP Domain Lookup.....	11
3.5. Configure a DNS Server	11
3.6. Enable DHCP Service on a Cisco Router	11
3.7. Configuring IPv4 and IPv6 on Physical and VLAN Interfaces	12
3.7.1. IPv4 Configuration.....	12
3.7.2. IPv6 Configuration.....	12
3.7.3. VLAN Interface Configuration	12
3.7.4. Configuring Sub-Interfaces for VLANs (Single VLAN)	12

3.8.	Configuring a DHCP Server	12
3.8.1.	Enable and Configure a DHCP Server for IPv4	12
3.8.2.	Using IP Helper Address (Forward DHCP Requests)	13
3.9.	Routing.....	13
3.9.1.	Configuring Static Routing	13
3.9.2.	Configuring RIP	13
3.9.3.	Configuring Basic OSPF.....	13
3.10.	Cisco ESW: Routing vs. Switching Interfaces	13
3.10.1.	Step-by-Step VLAN Creation on VLAN Database Mode.....	13
3.10.2.	Assign VLANs to Interfaces	14
3.10.3.	Configuring a Trunk Interface	14
	Important Notes	14
4.	D-LINK Switch Operation.....	15

1. Using ipconfig, ifconfig, and arp Commands

This section provides an overview of `ipconfig`, `ifconfig`, and `arp` commands, their options, and how they can be used to troubleshoot network issues, view MAC addresses, configure IP settings, and manage network interfaces.

To configure an IP address, subnet mask, default gateway, and DNS server on Windows 10/11, Linux, and macOS using both GUI and command-line methods. Ensure administrative/root permissions when modifying network settings.

1.1. ipconfig (Windows)

`ipconfig` is a command-line tool used to display and manage IP configuration settings on a Windows machine.

Basic Commands:

Display basic IP configuration: `ipconfig`

This shows IPv4/IPv6 addresses, subnet mask, and default gateway.

Display detailed network configuration: `ipconfig /all`

This provides additional details such as MAC address, DHCP, DNS, and lease expiration.

Release current IP configuration (for DHCP networks): `ipconfig /release`

This disconnects the IP address assigned by the DHCP server.

Renew IP configuration (request a new address from DHCP): `ipconfig /renew`

Flush the DNS resolver cache: `ipconfig /flushdns`

Clears the DNS cache, resolving DNS-related issues.

Display DNS cache: `ipconfig /displaydns`

Show all active TCP/IP connections and ports: `netstat -an`

Checking MAC Address:

Find the MAC address of all network interfaces: `ipconfig /all | findstr "Physical"`

The **Physical Address** field corresponds to the MAC address.

To configure the Gateway of the PC, execute the following command (administrator):

```
route add <NETWORK> mask <MASK> <GW IPaddress>
```

The command `route print` shows the routing table of the PC and the configured *Gateways*.

1.2. ifconfig (Linux/macOS)

`ifconfig` is used to configure network interfaces in Linux and macOS. On newer Linux distributions, `ip` is preferred over `ifconfig`.

Basic Commands:

Display network interfaces and IP configuration: `ifconfig`

Shows interface names, IP addresses, and MAC addresses.

Show detailed information for a specific interface: `ifconfig eth0`

Replace `eth0` with the relevant interface name.

Enable a network interface: `sudo ifconfig eth0 up`

Disable a network interface: `sudo ifconfig eth0 down`

Assign a static IP address: `sudo ifconfig eth0 192.168.1.100 netmask 255.255.255.0`

Change the MAC address (requires root privileges): `sudo ifconfig eth0 hw ether 00:1A:2B:3C:4D:5E`

View MAC addresses of all interfaces: `ifconfig | grep ether`

The **ether** field shows the MAC address.

View only active network interfaces: `ifconfig -a`

To configure the Gateway of the PC, execute the following command:

```
sudo route add default gw <IPaddress>
```

The command `route -n` shows the routing table of the PC and the configured *Gateways*.

Alternative (`ip` command for Linux):

Since `ifconfig` is deprecated in some Linux distributions, the `ip` command is recommended:

Show all interfaces: `ip a`

Bring an interface up/down:

```
sudo ip link set eth0 up
```

```
sudo ip link set eth0 down
```

Show MAC address: `ip link show eth0`

1.3. arp (Address Resolution Protocol)

The `arp` command is used to view and manipulate the ARP table, which maps IP addresses to MAC addresses.

1.3.1. Windows:

Display the ARP table: `arp -a`

Lists known IP-to-MAC address mappings.

Clear the ARP cache: `netsh interface ip delete arpcache`

Add a static ARP entry: `arp -s <IP_address> <MAC_address>`

Example: `arp -s 192.168.1.100 00-14-22-01-23-45`

Delete a specific ARP entry: `arp -d <IP_address>`

Example: `arp -d 192.168.1.100`

1.3.2. Linux/macOS:

Display the ARP table: `arp -a`

Delete a specific ARP entry: `sudo arp -d <IP_address>`

Add a static ARP entry: `sudo arp -s <IP_address> <MAC_address>`

Example: `sudo arp -s 192.168.1.100 00:14:22:01:23:45`

View ARP entries for a specific interface: `arp -i eth0 -a`

Show detailed ARP table information: `ip neigh show`

2. Basic Networking Tools: Ping, Traceroute, and Nslookup

This section provides a detailed explanation of the `ping`, `traceroute`, and `nslookup` commands across different operating systems, including how to modify relevant parameters for troubleshooting and analysis. Ensure you have appropriate permissions to execute these commands effectively.

2.1. Ping Command

The `ping` command is used to test network connectivity between devices by sending ICMP (Internet Control Message Protocol) echo requests and waiting for a reply.

2.1.1. Windows: `ping <IP_address_or_domain>`

Example: `ping 8.8.8.8`

Change packet size: `ping -l <size> <IP_address_or_domain>`

Example: `ping -l 1500 google.com`

Set number of packets: `ping -n <count> <IP_address_or_domain>`

Example: `ping -n 5 google.com`

Modify Time-To-Live (TTL): `ping -i <TTL_value> <IP_address_or_domain>`

Example: `ping -i 64 google.com`

2.1.2. Linux/macOS: `ping <IP_address_or_domain>`

Example: `ping google.com`

Change packet size: `ping -s <size> <IP_address_or_domain>`

Example: `ping -s 1500 google.com`

Set number of packets: `ping -c <count> <IP_address_or_domain>`

Example: `ping -c 5 google.com`

Modify TTL: `ping -t <TTL_value> <IP_address_or_domain>`

Example: `ping -t 64 google.com`

Ping using IPv6: `ping6 <IPv6_address_or_domain>`

Example: `ping6 ipv6.google.com`

2.2. Traceroute Command

The `traceroute` (or `tracert` in Windows) command is used to track the route packets take to reach a destination.

2.2.1. Windows:

tracert <IP_address_or_domain>

Example: `tracert google.com`

Modify number of hops: `tracert -h <max_hops> <IP_address_or_domain>`

Example: `tracert -h 30 google.com`

Change timeout value: `tracert -w <timeout_ms> <IP_address_or_domain>`

Example: `tracert -w 500 google.com`

2.2.2. Linux/macOS:

traceroute <IP_address_or_domain>

Example: `traceroute google.com`

Modify number of hops: `traceroute -m <max_hops> <IP_address_or_domain>`

Example: `traceroute -m 30 google.com`

Use ICMP packets (similar to Windows tracert): `traceroute -I <IP_address_or_domain>`

Example: `traceroute -I google.com`

Traceroute using IPv6: `traceroute6 <IPv6_address_or_domain>`

Example: `traceroute6 ipv6.google.com`

2.3. Nslookup Command

The `nslookup` command queries the DNS (Domain Name System) to resolve domain names to IP addresses and vice versa.

2.3.1. Windows/Linux/macOS:

`nslookup <domain>`

Example: `nslookup google.com`

Advanced Usage

Find Mail Exchange (MX) records: `nslookup -query=mx <domain>`

Example: `nslookup -query=mx gmail.com`

Find Name Server (NS) records: `nslookup -query=ns <domain>`

Example: `nslookup -query=ns google.com`

Find Start of Authority (SOA) records: `nslookup -query=soa <domain>`

Example: `nslookup -query=soa example.com`

Use a specific DNS server for queries: `nslookup <domain> <DNS_server>`

Example: `nslookup google.com 8.8.8.8`

3. Simple CLI Guide for Cisco IOS

3.1. Basic Command Line Navigation

```
enable // Enter privileged EXEC mode
configure terminal // Enter global configuration mode
exit // Exit to previous mode
write memory // Save configuration
show running-config // Display current configuration
show ip interface brief // Show status of interfaces
```

3.2. Using Auto-Complete and "?" in Cisco CLI

```
<Tab> // Auto-completes a partially typed command
? // Displays possible commands or parameters
```

Example:

```
Router# show ip ? // Displays possible subcommands for "show ip"
```

3.3. Cisco prompt and levels

Cisco IOS has different command-line interface (CLI) prompt levels, known as **privilege levels**, that control access to various commands. The three most common levels are:

3.3.1. User EXEC Mode (Privilege Level 1)

Prompt: Router>

Access: Basic monitoring and connectivity commands (e.g., ping, show version)

Restrictions: Cannot modify the configuration or view sensitive details

To enter: Automatically accessed after login

Exit command: logout or exit

3.3.2. Privileged EXEC Mode (Privilege Level 15)

Prompt: Router#

Access: Full access to view and troubleshoot the device (e.g., show running-config)

To enter: Type enable in User EXEC mode (requires a password if configured)

Exit command: disable (returns to User EXEC mode)

3.3.3. Global Configuration Mode

Prompt: Router(config)#

Access: Allows modification of the device configuration

To enter: Type configure terminal or conf t in Privileged EXEC mode

Exit command: `exit` (returns to Privileged EXEC mode)

3.3.4. Other Configuration Modes

Interface Configuration Mode (`Router(config-if)#`) – Configures specific interfaces like Ethernet or serial ports.

Line Configuration Mode (`Router(config-line)#`) – Configures terminal lines (e.g., console, vty).

Router Configuration Mode (`Router(config-router)#`) – Configures dynamic routing protocols like OSPF, EIGRP.

3.4. Global Configuration Commands

3.4.1. Enabling IPv6 Routing

```
ipv6 unicast-routing // Enable global IPv6 routing
```

3.4.2. Allow the Use of Subnet Zero

By default, older Cisco routers do not allow the use of **subnet zero**. To enable it:

```
Router(config)# ip subnet-zero
```

This command allows the use of the first subnet (e.g., 192.168.1.0/26).

3.4.3. Enable and Disable IP Domain Lookup

By default, Cisco routers attempt to resolve mistyped commands as domain names, causing delays. To **disable IP domain lookup**:

```
Router(config)# no ip domain-lookup
```

To **enable IP domain lookup** (if disabled):

```
Router(config)# ip domain-lookup
```

3.5. Configure a DNS Server

To configure a DNS server on the router:

```
Router(config)# ip name-server 8.8.8.8 8.8.4.4 ! Set Google DNS servers
Router(config)# ip domain-lookup ! Ensure domain lookup is enabled
Router(config)# ip domain-name example.com ! Define a default domain name
```

To test DNS resolution:

```
Router# ping google.com
```

3.6. Enable DHCP Service on a Cisco Router

By default, the DHCP service is enabled on Cisco routers, but if it has been disabled, you can enable it with the following command:

```
Router(config)# service dhcp
```

To **disable** the DHCP service, use:

```
Router(config)# no service dhcp
```

To verify that DHCP is running, use:

```
Router# show ip dhcp server statistics
Router# show running-config | include dhcp
```

3.7. Configuring IPv4 and IPv6 on Physical and VLAN Interfaces

3.7.1. IPv4 Configuration

```
interface GigabitEthernet0/1    // Enter interface configuration mode
ip address 192.168.1.1 255.255.255.0 // Assign IPv4 address
no shutdown    // Enable the interface
exit
```

3.7.2. IPv6 Configuration

```
interface GigabitEthernet0/1    // Enter interface configuration mode
ipv6 address 2001:db8::1/64    // Assign IPv6 address
ipv6 enable    // Enable IPv6 on the interface
no shutdown
exit
```

3.7.3. VLAN Interface Configuration

```
interface vlan 10
ip address 192.168.10.1 255.255.255.0
no shutdown
exit
```

3.7.4. Configuring Sub-Interfaces for VLANs (Single VLAN)

```
interface GigabitEthernet0/1.10 // Create sub-interface for VLAN 10
encapsulation dot1Q 10          // Assign VLAN 10 to the sub-interface
ip address 192.168.10.1 255.255.255.0 // Assign IP address
exit
interface GigabitEthernet0/1
no shutdown
```

3.8. Configuring a DHCP Server

3.8.1. Enable and Configure a DHCP Server for IPv4

To configure the router as a **DHCP server**, follow these steps:

```
Router(config)# ip dhcp excluded-address 192.168.1.1 192.168.1.10 ! Exclude
addresses (e.g., reserved for static IPs)
Router(config)# ip dhcp pool MYPOOL ! Create a DHCP pool
Router(dhcp-config)# network 192.168.1.0 255.255.255.0 ! Define subnet
Router(dhcp-config)# default-router 192.168.1.1 ! Set default gateway
Router(dhcp-config)# dns-server 8.8.8.8 8.8.4.4 ! Define DNS servers
Router(dhcp-config)# lease 7 ! Lease time (7 days)
Router(dhcp-config)# exit
```

To verify DHCP leases:

```
Router# show ip dhcp binding
```

3.8.2. Using IP Helper Address (Forward DHCP Requests)

```
interface vlan 10
 ip helper-address 192.168.2.1 // Forward DHCP requests to DHCP server
```

3.9. Routing

3.9.1. Configuring Static Routing

```
ip route 192.168.2.0 255.255.255.0 192.168.1.2 // Static route to network
192.168.2.0
```

For IPv6:

```
ipv6 route 2001:db8:1::/64 2001:db8::2 // Static route for IPv6
```

3.9.2. Configuring RIP

3.9.2.1. RIP for IPv4

```
router rip
 version 2 // Enable RIPv2
 network 192.168.1.0
 network 192.168.2.0
 no auto-summary
 redistribute connected subnets
 redistribute static subnets
```

3.9.2.2. RIP for IPv6

```
ipv6 router rip MYRIP
 redistribute connected
 redistribute static
 interface GigabitEthernet0/1
 ipv6 rip MYRIP enable
```

3.9.3. Configuring Basic OSPF

3.9.3.1. OSPF for IPv4

```
router ospf 1 // OSPF process ID
 network 192.168.1.0 0.0.0.255 area 0 // Define OSPF area
 network 192.168.2.0 0.0.0.255 area 0
 redistribute connected subnets
 redistribute static subnets
```

3.9.3.2. OSPF for IPv6

```
ipv6 router ospf 1
 router-id 1.1.1.1
 interface GigabitEthernet0/1
 ipv6 ospf 1 area 0
```

3.10. Cisco ESW: Routing vs. Switching Interfaces

In Cisco IOS 12.1, VLANs were traditionally configured using the **VLAN database mode** instead of the modern **config mode** used in newer IOS versions. Here's how you can create VLANs using the **VLAN database mode** in IOS 12.1:

3.10.1. Step-by-Step VLAN Creation on VLAN Database Mode

Enter VLAN database mode:

```
Router# vlan database
```

The prompt will change to:

```
Router(vlan) #
```

Create a VLAN and assign a name (optional):

```
Router(vlan) # vlan 10 name SALES
Router(vlan) # vlan 20 name HR
Router(vlan) # vlan 30 name IT
```

- o `vlan 10 name SALES` creates **VLAN 10** and names it **SALES**.
- o You can create multiple VLANs in the same way.

Apply and save the VLAN configuration:

```
Router(vlan) # exit
```

This saves the VLAN database to **NVRAM**.

Verify VLAN creation:

```
Router# show vlan      or   Router# show vlan brief
```

3.10.2. Assign VLANs to Interfaces

Once VLANs are created, you need to assign them to interfaces:

Enter interface configuration mode:

```
Router(config) # interface FastEthernet1/0
```

Set the interface as an access port and assign a VLAN:

```
Router(config-if) # switchport mode access
Router(config-if) # switchport access vlan 10
```

Repeat for other interfaces and VLANs as needed.

3.10.3. Configuring a Trunk Interface

```
Router(config) # interface FastEthernet1/14
Router(config-if) # switchport mode trunk // Set interface as trunk
Router(config-if) # switchport trunk allowed vlan 10,20,30 //Allow specific
VLANs (optional)
Router(config-if) #exit
```

Important Notes

VLAN database mode was **deprecated in later IOS versions** (after 12.2) in favor of configuring VLANs in **global configuration mode**.

VLANs stored in the **VLAN database** are not affected by `write erase` and `reload` unless manually deleted.

To delete a VLAN in VLAN database mode: `Router(vlan) # no vlan 10`

4. D-LINK Switch Operation

Connect the Switch to the PC. After a while, the Switch prompt will appear:

```
#
```

To configure the IP address of the Switch, execute the following command:

```
#config ipif System ipaddress 192.1.1.21/24  
#show ipif
```

To show the switching table of the switch:

```
#show fdb
```

To create a default gateway on the switch:

```
#create iproute default 192.1.1.11 1  
#show iproute
```

Execution of command *ping*

```
#ping 192.1.1.1 times 4
```