Practica 4: Multiples Rutas

Oscar Andrés Rosas Hernández *Instituto Politécnico Nacional, Escuela Superior de Cómputo, CDMX*

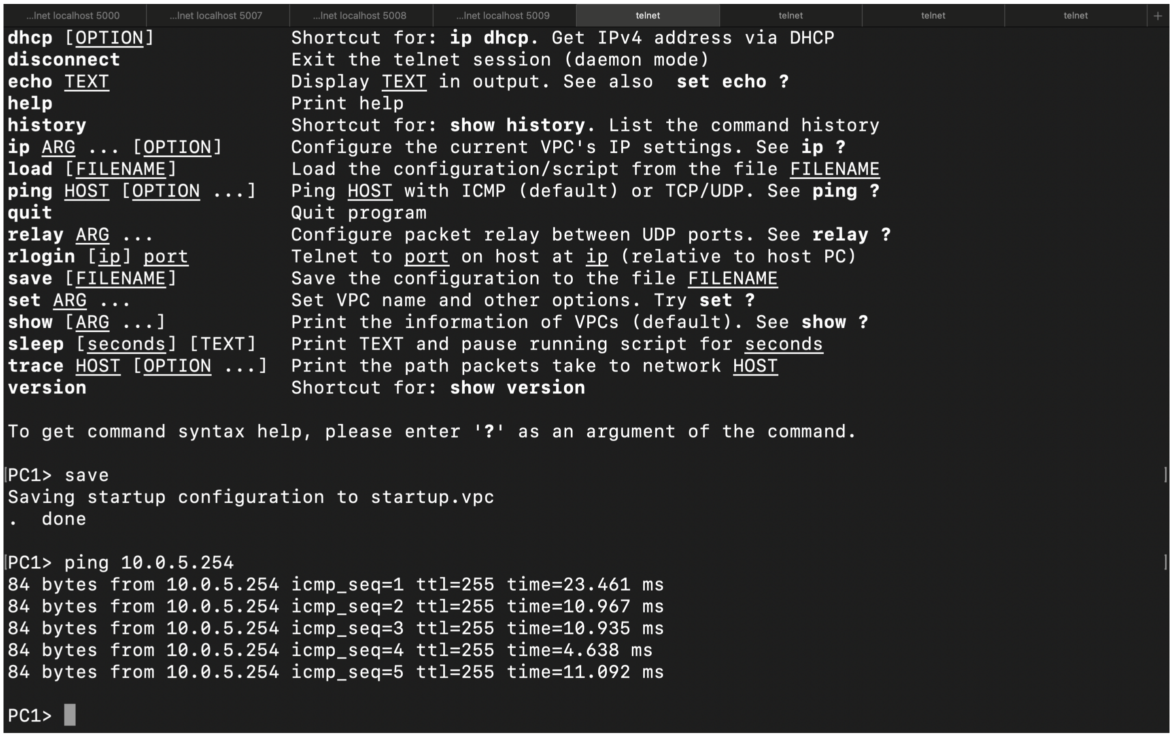
I. Practica 4

Muy buena practica, despues de haber creado la topologia basta con hacer una especia de crawler que vaya al R4, y captura su IP, de ahi se mueve a cada uno de los routes, y captura su IP de la misma manera, una vez que existan todos los datos basta con hacer los pasos para configurar las cosas manual (la mas facil de hacer), luego levantar RIP en los indicados y finalmente hacer lo mismo con OSPF.

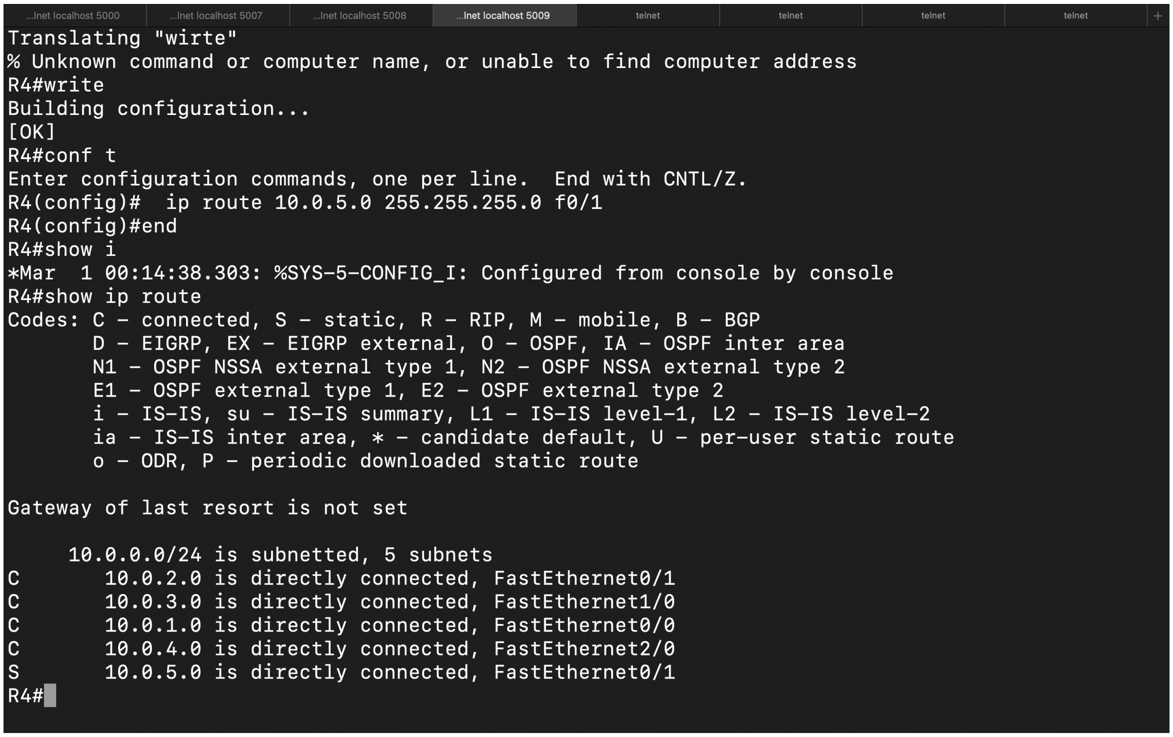
Lista de pasos realizados:

```
! RANDOM COMMANDS
enable
configure terminal
interface fx/x
no shutdown
exit
no shutdown
write memory
end

! ==== Router 4 ====
  interface FastEthernet 0/0 !--> Computadora MV
  ip address 10.0.1.254 255.255.255.0

  interface FastEthernet 0/1 !--> Router1
  ip address 10.0.2.254 255.255.255.0

  interface FastEthernet 1/0 !--> Router2
  ip address 10.0.3.254 255.255.255.0

  interface FastEthernet 2/0 !--> Router3
  ip address 10.0.4.254 255.255.255.0

  !MANUAL ip route 10.0.5.0 255.255.255.0 f0/1

! ==== Router 1 ====
  interface FastEthernet 0/0 !--> Computadora MV
  ip address 10.0.5.254 255.255.255.0

  interface FastEthernet 0/1 !--> Router4
  ip address 10.0.2.253 255.255.255.0

  !MANUAL ip route 10.0.1.0 255.255.255.0 f0/1
  !MANUAL ip route 10.0.3.0 255.255.255.0 f0/1
  !MANUAL ip route 10.0.4.0 255.255.255.0 f0/1
  !MANUAL ip route 10.0.6.0 255.255.255.0 f0/1
  !MANUAL ip route 10.0.7.0 255.255.255.0 f0/1


! ==== Router 2 ====
  interface FastEthernet 0/0 !--> Computadora MV
  ip address 10.0.6.254 255.255.255.0

  interface FastEthernet 0/1 !--> Router4
  ip address 10.0.3.253 255.255.255.0

! ==== Router 3 ====
  interface FastEthernet 0/0 !--> Computadora MV
  ip address 10.0.7.254 255.255.255.0

  interface FastEthernet 0/1 !--> Router4
  ip address 10.0.4.253 255.255.255.0


! ==== Config PC ====
  ip 10.0.5.1 /24 10.0.5.254 !PC1
  ip 10.0.6.1 /24 10.0.6.254 !PC2
  ip 10.0.7.1 /24 10.0.7.254 !PC3



! ==== Config SSH ====
  enable
  configure terminal

  # Name stuff
  hostname R#
  ip domain-name R#.LOCAL

  # Config line
  line vty 0 4
  transport input ssh
  login local
  exit

  line console 0
  logging synchronous
  login local
  exit

  # Usernames
  username admin priv 15 password admin
  enable secret 12345678
  service password-encryption

  # SSH versions
  crypto key generate rsa
  ip ssh time-out 15
  ip ssh authentication-retries 5
  ip ssh version 2

  # Generate keys
  crypto key generate rsa
```

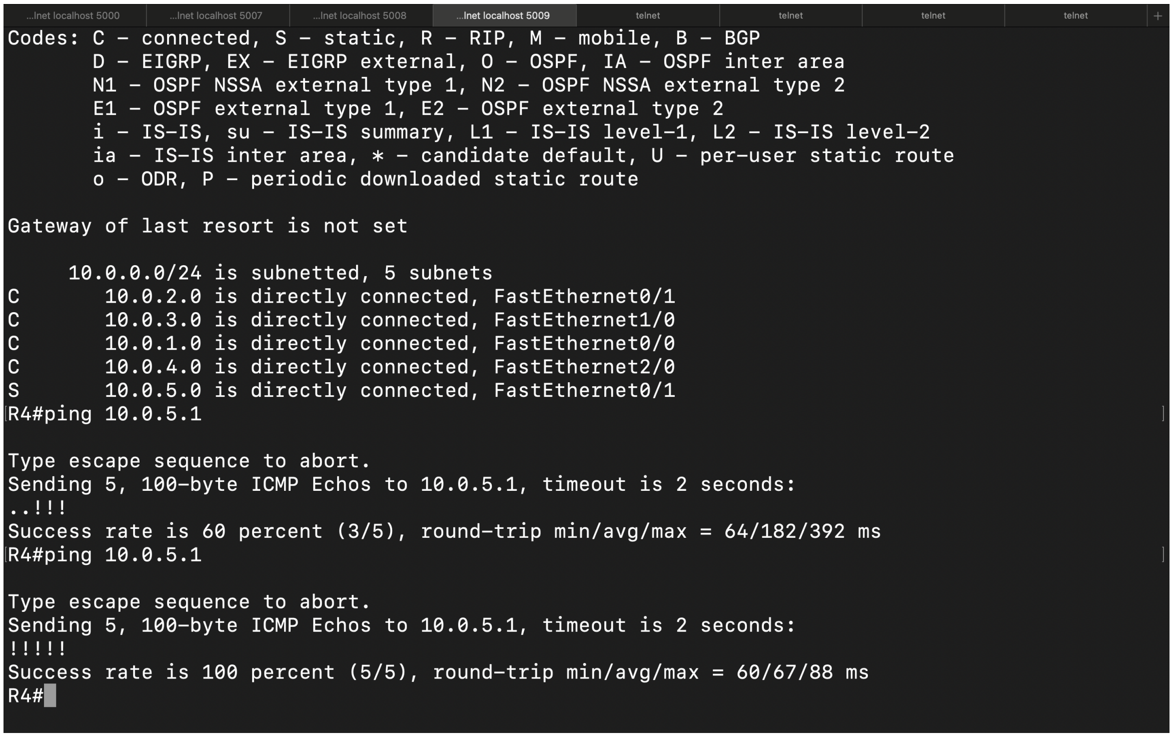
Evidencias



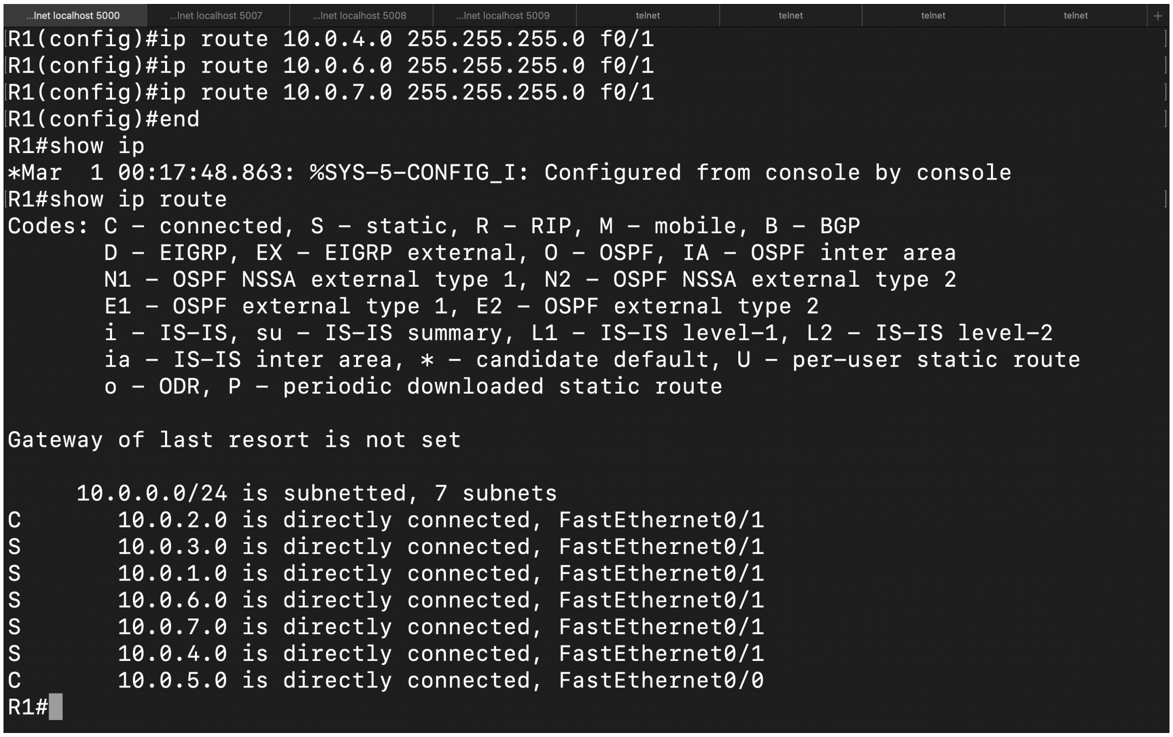
```
dhcp [OPTION]             Shortcut for: ip dhcp. Get IPv4 address via DHCP
disconnect                Exit the telnet session (daemon mode)
echo TEXT                 Display TEXT in output. See also  set echo ?
help                      Print help
history                   Shortcut for: show history. List the command history
ip ARG ... [OPTION]       Configure the current VPC's IP settings. See ip ?
load [FILENAME]           Load the configuration/script from the file FILENAME
ping HOST [OPTION ...]    Ping HOST with ICMP (default) or TCP/UDP. See ping ?
quit                      Quit program
relay ARG ...             Configure packet relay between UDP ports. See relay ?
rlogin [ip] port          Telnet to port on host at ip (relative to host PC)
save [FILENAME]           Save the configuration to the file FILENAME
set ARG ...               Set VPC name and other options. Try set ?
show [ARG ...]            Print the information of VPCs (default). See show ?
sleep [seconds] [TEXT]    Print TEXT and pause running script for seconds
trace HOST [OPTION ...]   Print the path packets take to network HOST
version                   Shortcut for: show version

To get command syntax help, please enter '?' as an argument of the command.

PC1> save
Saving startup configuration to startup.vpc
.  done

PC1> ping 10.0.5.254
84 bytes from 10.0.5.254 icmp_seq=1 ttl=255 time=23.461 ms
84 bytes from 10.0.5.254 icmp_seq=2 ttl=255 time=10.967 ms
84 bytes from 10.0.5.254 icmp_seq=3 ttl=255 time=10.935 ms
84 bytes from 10.0.5.254 icmp_seq=4 ttl=255 time=4.638 ms
84 bytes from 10.0.5.254 icmp_seq=5 ttl=255 time=11.092 ms

PC1>
```



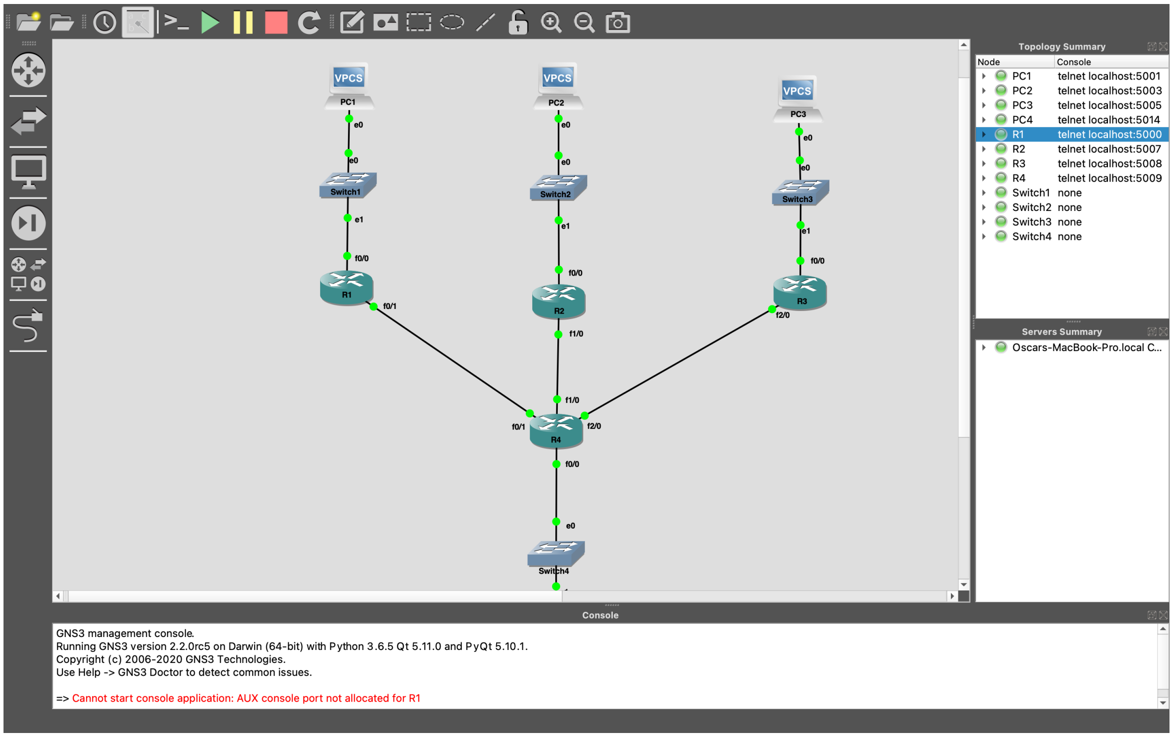
```
Translating "wirte"
% Unknown command or computer name, or unable to find computer address
R4#write
Building configuration...
[OK]
R4#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#  ip route 10.0.5.0 255.255.255.0 f0/1
R4(config)#end
R4#show i
*Mar  1 00:14:38.303: %SYS-5-CONFIG_I: Configured from console by console
R4#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 5 subnets
C       10.0.2.0 is directly connected, FastEthernet0/1
C       10.0.3.0 is directly connected, FastEthernet1/0
C       10.0.1.0 is directly connected, FastEthernet0/0
C       10.0.4.0 is directly connected, FastEthernet2/0
S       10.0.5.0 is directly connected, FastEthernet0/1
R4#
```



```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 5 subnets
C       10.0.2.0 is directly connected, FastEthernet0/1
C       10.0.3.0 is directly connected, FastEthernet1/0
C       10.0.1.0 is directly connected, FastEthernet0/0
C       10.0.4.0 is directly connected, FastEthernet2/0
S       10.0.5.0 is directly connected, FastEthernet0/1
R4#ping 10.0.5.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.5.1, timeout is 2 seconds:
..!!!
Success rate is 60 percent (3/5), round-trip min/avg/max = 64/182/392 ms
R4#ping 10.0.5.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.5.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 60/67/88 ms
R4#
```



```
R1(config)#ip route 10.0.4.0 255.255.255.0 f0/1
R1(config)#ip route 10.0.6.0 255.255.255.0 f0/1
R1(config)#ip route 10.0.7.0 255.255.255.0 f0/1
R1(config)#end
R1#show ip
*Mar  1 00:17:48.863: %SYS-5-CONFIG_I: Configured from console by console
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 7 subnets
C       10.0.2.0 is directly connected, FastEthernet0/1
S       10.0.3.0 is directly connected, FastEthernet0/1
S       10.0.1.0 is directly connected, FastEthernet0/1
S       10.0.6.0 is directly connected, FastEthernet0/1
S       10.0.7.0 is directly connected, FastEthernet0/1
S       10.0.4.0 is directly connected, FastEthernet0/1
C       10.0.5.0 is directly connected, FastEthernet0/0
R1#
```



Node	Console
PC1	telnet localhost:5001
PC2	telnet localhost:5003
PC3	telnet localhost:5005
PC4	telnet localhost:5014
R1	telnet localhost:5000
R2	telnet localhost:5007
R3	telnet localhost:5008
R4	telnet localhost:5009
Switch1	none
Switch2	none
Switch3	none
Switch4	none

```
GNS3 management console.
Running GNS3 version 2.2.0rc5 on Darwin (64-bit) with Python 3.6.5 Qt 5.11.0 and PyQt 5.10.1.
Copyright (c) 2006-2020 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.

=> Cannot start console application: AUX console port not allocated for R1
```

Script realizado:

```
import json
import netifaces
import os

from pexpect import pxssh, spawn
from ipaddress import IPv4Address
from graphviz import Digraph
from threading import Lock, Thread

username, password = "admin", "firulais"
algorithm, cipher = "diffie-hellman-group1-sha1", "aes256-cbc"
options = {"KexAlgorithms": algorithm, "Ciphers": cipher}

routers = {}
lock = Lock()

try:
    os.system("rm /home/soyoscarrh/.ssh/known_hosts")
except:
    pass


def crawling_from(ip):
    print("\nLogging into", ip)

    child = pxssh.pxssh(options=options)
    child.login(ip, username, password, auto_prompt_reset=False)

    name = child.before[-2:].decode("utf-8")
    if name == "":
        return

    print(f"[{name}] logged into")

    with lock:
        if name in routers and routers[name] != "seen":
            print(f"[{name}] data already here, logging out")
            return

    child.sendline("show cdp neighbors")
    child.expect(f"{name}#")

    data = child.before.decode("utf-8").split("\n")[5:-1]
    connections = {}

    for line in data:
        info = line.split()
        connection_name, interface = info[0][:2], info[2]
        connections[interface] = connection_name

    print(f"[{name}] neigbours found: {connections}")

    child.sendline("show ip interface brief")
    child.expect(f"{name}#")

    data = child.before.decode("utf-8").split("\r\n")
    data = [line for line in data if ’Fast’ in line]

    terminals = {}
    for line in data:
        info = line.split()
        interface, network_ip = info[0][-3:], info[1]
        not_a_router = interface not in connections
        switch_connection = network_ip[0] != "8"
        if (not_a_router or switch_connection) and network_ip != "unassigned":
            network_ip = network_ip[:-1] + "0"
            terminal_ip = str(IPv4Address(int(IPv4Address(network_ip)) + 10))
            if terminal_ip[0] != "8":
                terminals[interface] = terminal_ip

    print(f"[{name}] terminals found: {terminals}")

    with lock:
        routers[name] = {"terminals": terminals, "neigbours": connections}
    child.sendline("show ip route connected")
    child.expect(f"{name}#")

    next_jumps = []
    data = child.before.decode("utf-8").split("\r\n")
    data = [line for line in data if line != "" and line[0] == "C"]
    for line in data:
        info = line.split()
        network_ip, interface = info[1], info[-1][-3:]
        if interface in connections:
            next_ip1 = str(IPv4Address(int(IPv4Address(network_ip)) + 1))
            next_ip2 = str(IPv4Address(int(IPv4Address(network_ip)) + 2))
            connection_name = connections[interface]
            next_jumps.append((connection_name, next_ip1, next_ip2))

    child.sendline("conf t")
    child.sendline("username pirata priv 15 password pirata")
    child.sendline("end")

    print(f"[{name}] next jumps: {next_jumps}")
    for connection_name, next_ip1, next_ip2 in next_jumps:
        with lock:
            if connection_name in routers:
                continue
            routers[connection_name] = "seen"

        print(f"[{name}] searching {connection_name}")
        crawling_from(next_ip1)
        crawling_from(next_ip2)


gateways = netifaces.gateways()
default_gateway = gateways[’default’][netifaces.AF_INET][0]
ip = default_gateway
crawling_from(ip)

dot = Digraph(comment=’Topology of network’, format=’png’)
for router in routers:
    if routers[router] == "seen":
        continue

    print("\n", router)
    print("\t neigbours:", routers[router]["neigbours"])
    print("\t terminals:", routers[router]["terminals"])
    dot.node(router, router)

    for interface in routers[router]["neigbours"]:
        jump_name = routers[router]["neigbours"][interface]
        dot.edge(router, jump_name)

    for interface in routers[router]["terminals"]:
        ip = routers[router]["terminals"][interface]
        dot.node(ip, ip, shape=’plaintext’)
        dot.edge(router, ip)
        dot.edge(ip, router)

dot.render(’./net’)
```