

Εργαστηριακή Άσκηση 6 Εισαγωγή στο Quagga και FRRouting (FRR)

Άσκηση 1

- 1) Εκτελούμε τις οδηγίες
- 2) Εκτελούμε **dhclient em0**
- 3) Εκτελούμε **ping www.google.com** και επιτυγχάνει
- 4) Εκτελούμε την εντολή **pkg update**
- 5) Εκτελούμε **pkg install frr7**
- 6) Προσθέτουμε τη γραμμή
- 7) **mkdir /usr/local/etc/frr** και εκτελούμε στο **/usr/local/etc/frr touch vtysh.conf touch frr.conf**
- 8) Εκτελούμε **chown -R frr:frr /usr/local/etc/frr**
- 9) Εκτελούμε **chown frr:frr /usr/local/etc/frr/vtysh.conf**

```
root@PC:/usr/local/etc/frr # chown frr:frrvty /usr/local/etc/frr/vtysh.conf
root@PC:/usr/local/etc/frr # ls -la
total 8
drwxr-xr-x  2 frr  frr    512 Apr  2 12:12 .
drwxr-xr-x  6 root wheel  512 Apr  2 12:12 ..
-rw-r--r--  1 frr  frr      0 Apr  2 12:12 frr.conf
-rw-r--r--  1 frr  frrvty  0 Apr  2 12:12 vtysh.conf
```

10)

```
root@PC:/usr/local/etc/frr # cat /etc/rc.conf
hostname="PC"
#ifconfig_DEFAULT="DHCP inet6 accept_rtadv"
#growfs_enable="YES"
sshd_enable="YES"
syslogd_flags="-scc"
gateway_enable="YES"
frr_vtysh_boot="YES"
frr_enable="YES"
frr_daemons="zebra staticd"
```

11)

```
root@PC:/usr/local/etc/frr # cat /etc/csh.cshrc
# $FreeBSD$
#
# System-wide .cshrc file for csh(1).
setenv VTYSH_PAGER "more -EFX"
```

- 12) Εκτελούμε **rm /etc/resolv.conf** και **poweroff**
- 13) Βάζουμε την κάρτα LAN1 σε εσωτερική δικτύωση (Internal Network)

14)

```

root@PC:~ # service sshd status
sshd is running as pid 738.
root@PC:~ # service frr status
zebra is running as pid 501.
staticd is running as pid 503.

```

15) Εκτελούμε “history -c” και στη συνέχεια τα υπόλοιπα βήματα, με την επιλογή σχετικά με το MAC Address Policy να είναι στο “Include only NAT network adapter MAC addresses”.

16) Αποθηκεύουμε το αρχείο

1.1 Λαμβάνουμε το σφάλμα “Vty password is not set, Connection closed by foreign host” έχοντας εκτελέσει “telnet localhost 2601”

```

root@PC:~ # telnet localhost 2601
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Vty password is not set.
Connection closed by foreign host.

```

1.2 Με την εντολή “vtysh”

1.3 Βλέπουμε 23 εντολές:

```

PC#
add          Add registration
clear        clear
configure    Configuration from vty interface
copy         Apply a configuration file
debug        Debugging functions
disable      Turn off privileged mode command
enable       Turn on privileged mode command
end          End current mode and change to enable mode
exit         Exit current mode and down to previous mode
find         Find CLI command matching a regular expression
graceful-restart Graceful Restart commands
list         Print command list
mtrace       Multicast trace route to multicast source
no           Negate a command or set its defaults
output       Direct vtysh output to file
ping         Send echo messages
quit         Exit current mode and down to previous mode
rpki         Control rpki specific settings
show         Show running system information
terminal     Set terminal line parameters
traceroute   Trace route to destination
watchfrr     Watchfrr Specific sub-command
write        Write running configuration to memory, network, or terminal

```

1.4 Παρατηρούμε πως κάνει autocomplete

1.5 Εάν πληκτρολογήσουμε “co” και μετά TAB δε κάνει τίποτα, ωστόσο εάν ξαναπατήσουμε TAB εμφανίζει όλες τις διαθέσιμες εντολές που ξεκινάνε από “co” (configure, copy). Εάν, αφού έχουμε πατήσει τα 2 γράμματα, πληκτρολογήσουμε “?”, τότε εμφανίζεται μια μικρή περιγραφή για κάθε μία από τις διαθέσιμες εντολές που ξεκινάνε από “co”

1.6 Η “show version”

1.7 Γράφουμε “wr”, μετά TAB, μετά “t” και μετά ξανά TAB

1.8 Με την εντολή “show running-config” ή “write terminal”

1.9 Με την εντολή “configure terminal”

1.10 Εκτελούμε “hostname R1”, οπότε και μας αλλάζει το prompt.

```
PC(config)# hostname R1
R1(config)#
```

1.11 Εκτελούμε “password ntua”

1.12 2 φορές, η μία μας πάει στο Privileged EXEC και η άλλη στο αρχικό κέλυφος UNIX

1.13 Πλέον απαιτείται συνθηματικό.

1.14 Βρισκόμαστε σε επίπεδο User EXEC.

1.15 Βλέπουμε 10 εντολές:

```
PC>
debug      Debugging functions
enable     Turn on privileged mode command
exit       Exit current mode and down to previous mode
find       Find CLI command matching a regular expression
help       Description of the interactive help system
list       Print command list
quit       Exit current mode and down to previous mode
show       Show running system information
terminal   Set terminal line parameters
who        Display who is on vty
PC>
```

1.16 Προφανώς, εφόσον δεν είμαστε πλέον Privileged χρήστες έχουμε πολύ λιγότερες διαθέσιμες εντολές

1.17 Με την εντολή “show interface”

1.18 Με την εντολή “show ip forwarding”

```
PC> show ip forwarding
IP forwarding is on
```

1.19 Με την εντολή “show ip route”. Ο πίνακας είναι άδειος

1.20 Όχι δε μπορούμε, καθώς δεδομένου ότι δεν είμαστε προνομιούχος χρήστης δεν αναγνωρίζεται η εντολή μας.

1.21 Με την εντολή “enable”

1.22 Πλέον μπορούμε να δούμε την παραμετροποίηση του FRR, η οποία περιλαμβάνει και το password που ορίσαμε.

```
PC# write terminal

Current configuration:
!
frr version 8.5.4
frr defaults traditional
!
hostname PC
password ntua
!
!
!
!
no ipv6 forwarding
!
!
!
end
```

1.23 Βλέπουμε 18 διαθέσιμες εντολές:

```
PC#
clear          Clear stored data in all pthreads
configure      Configuration from vty interface
copy           Copy configuration
debug          Debugging functions
disable        Turn off privileged mode command
enable         Turn on privileged mode command
end            End current mode and change to enable mode.
exit           Exit current mode and down to previous mode
find           Find CLI command matching a regular expression
help           Description of the interactive help system
list           Print command list
logmsg         Send a message to enabled logging destinations
no             Negate a command or set its defaults
quit           Exit current mode and down to previous mode
show           Show running system information
terminal       Set terminal line parameters
who            Display who is on vty
write          Write running configuration to memory, network, or terminal
```

1.24 Εκτελούμε την εντολή “list”

1.25 Αρχικά θα εκτελέσουμε “configure terminal” για να εισέλθουμε σε λειτουργία Global Configuration Mode και στη συνέχεια θα εκτελέσουμε “enable password ntua”.

1.26 Με την εντολή “service password-encryption”

1.27 Με την εντολή “write memory” / “write file”. Ενημερώνεται το αρχείο /usr/local/etc/frr/zebra.conf

```
PC(config)# write memory
Configuration saved to /usr/local/etc/frr/zebra.conf
```

1.28 Αυτή τη φορά απαιτείται κωδικός για την είσοδο

1.29 Με την εντολή “vtysh” αποκτάω κατευθείαν πρόσβαση ως root απευθείας στο επίπεδο “Privileged EXEC”, χωρίς την απαίτηση κωδικού.

1.30 Με την εντολή “write memory” / “write file”. Αυτή τη φορά ενημερώνεται το αρχείο /usr/local/etc/frr/frr.conf. Το FRR χρησιμοποιεί μοναδικό configuration που βρίσκεται στο /etc/frr/frr.conf. Οπότε όταν ξεκινάει με την vtysh, καλείται να διαβάσει από αυτό το αρχείο config και να στείλει τα κατάλληλα τμήματα μόνο στους δαίμονες που ενδιαφέρονται για αυτά.

```
Building Configuration...
Integrated configuration saved to /usr/local/etc/frr/frr.conf
```

1.31 Θα προτιμούσαμε το ssh, καθώς μέσω telnet το συνθηματικό θα μεταφέρονταν ως Plaintext με αποτέλεσμα να είναι επιρρεπές σε υποκλοπές από τρίτους.

Άσκηση 2

2.1 Για το PC1: Αρχικά συνδεόμαστε με “vtysh” και όντας Privileged εκτελούμε “configure terminal”. Εκεί, εκτελούμε “hostname PC1” και στη συνέχεια “interface em0”. Αμέσως μετά εκτελούμε “ip address 192.168.1.2/24”.

Αντιστοίχα για το PC2 με hostname PC2 και ip: 192.168.2.2/24

2.2 Αρχικά συνδεόμαστε με “cli” και όντας Privileged εκτελούμε “configure terminal”. Εκεί, εκτελούμε “hostname R1” και στη συνέχεια “interface em0”. Αμέσως μετά εκτελούμε “ip address 192.168.1.1/24”. Κάνουμε “exit” ώστε να επιστρέψουμε στο Global configuration mode και εκεί εκτελούμε “interface em1”, και αμέσως μετά “ip address 192.168.2.1/24”. Κάνουμε “exit” για να επιστρέψουμε πίσω. Εκτελούμε do write file και αφού βγούμε από το cli config save, ώστε να παραμείνουν οι ρυθμίσεις μετά την επανακκίνηση.

2.3 Μεταφερόμαστε σε Privileged EXEC mode. Εκεί εκτελούμε “show interface” και βλέπουμε τα παρακάτω:

PC1 (Linked Base for PC1 and PC2) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
Interface em0 is up, line protocol is up
Link ups:      1      last: 2024/04/02 15:02:12.95
Link downs:    0      last: (never)
vrf: default
index 1 metric 1 mtu 1500 speed 1000
flags: <UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>
Type: Ethernet
HWaddr: 08:00:27:aa:5a:fa
inet 192.168.1.2/24
Interface Type Other
Interface Slave Type None
protodown: off
  input packets 0, bytes 0, dropped 0, multicast packets 0
  input errors 0
  output packets 0, bytes 0, multicast packets 0
  output errors 0
  collisions 0
Interface lo0 is up, line protocol is up
Link ups:      1      last: 2024/04/02 14:32:45.22
Link downs:    0      last: (never)
vrf: default
index 2 metric 1 mtu 16384 speed 0
flags: <UP,LOOPBACK,RUNNING,MULTICAST>
Type: Loopback
```

PC2 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
Interface em0 is up, line protocol is up
Link ups:      1      last: 2024/04/02 15:01:05.00
Link downs:    0      last: (never)
vrf: default
index 1 metric 1 mtu 1500 speed 1000
flags: <UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>
Type: Ethernet
HWaddr: 08:00:27:2c:f2:ae
inet 192.168.2.2/24
Interface Type Other
Interface Slave Type None
protodown: off
  input packets 0, bytes 0, dropped 0, multicast packets 0
  input errors 0
  output packets 0, bytes 0, multicast packets 0
  output errors 0
  collisions 0
Interface lo0 is up, line protocol is up
Link ups:      1      last: 2024/04/02 14:12:14.23
Link downs:    0      last: (never)
vrf: default
index 2 metric 1 mtu 16384 speed 0
flags: <UP,LOOPBACK,RUNNING,MULTICAST>
Type: Loopback
```

```

Interface em0 is up, line protocol detection is disabled
  index 1 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>
  HWaddr: 08:00:27:95:e9:95
  inet 192.168.1.1/24 broadcast 192.168.1.255
  inet6 fe80::a00:27ff:fe95:e995/64
    input packets 0, bytes 0, dropped 0, multicast packets 0
    input errors 0
    output packets 0, bytes 480, multicast packets 5
    output errors 0
    collisions 0
Interface em1 is up, line protocol detection is disabled
  index 2 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>
  HWaddr: 08:00:27:73:2b:8f
  inet 192.168.2.1/24 broadcast 192.168.2.255
  inet6 fe80::a00:27ff:fe73:2b8f/64
    input packets 0, bytes 0, dropped 0, multicast packets 0
    input errors 0
    output packets 0, bytes 480, multicast packets 5
    output errors 0
    collisions 0

```

2.4 Με την εντολή “show ip forwarding” βλέπουμε ότι είναι ενεργοποιημένη

```

router.ntua.lab# show ip forwarding
IP forwarding is on

```

2.5 Στο PC1 εκτελούμε σε global configuration mode: ip route 192.168.2.0/24 192.168.1.1

2.6 Στο PC2 εκτελούμε σε global configuration mode: ip route 192.168.1.0/24 192.168.2.1

2.7 Με την εντολή show ip route από το Privileged EXEC mode.

Για το PC1 (όμοια το το PC2) εμφανίζεται η διαδρομή προς το PC2 μέσω του R1 καθώς επίσης και η network address που είναι directly connected

```

C>* 192.168.1.0/24 [0/1] is directly connected, em0, 00:32:16
S>* 192.168.2.0/24 [1/0] via 192.168.1.1, em0, weight 1, 00:00:05

```

2.8 Στην αρχή της γραμμής βλέπουμε C για την απευθείας σύνδεση και S για την στατική

2.9 netstat -rn

Παρατηρούμε πως εδώ εμφανίζονται επιπλέον οι Hosts που ανήκουν σε κάποιο υποδίκτυο, ενώ παραπάνω είχαμε μόνο τα υποδίκτυα. Ωστόσο δεν μπορούμε να δούμε αν η εγγραφή 192.168.2.0 είναι static ή όχι.

```

root@PC1:~ # netstat -rn
Routing tables

Internet:
Destination      Gateway           Flags      Netif Expire
127.0.0.1         link#2           UH         lo0
192.168.1.0/24    link#1           U          em0
192.168.1.2       link#1           UHS        lo0
192.168.2.0/24    192.168.1.1     UG1        em0

```

2.10 Ναι επικοινωνούν κανονικά

2.11 Χρησιμοποιούμε την εξής αλληλουχία εντολών: “configure terminal”, “interface em0”, “ip address 192.168.1.200/24”. Όπως βλέπουμε, η νέα διεύθυνση που αναθέσαμε μπήκε ως secondary:

```

router.ntua.lab(config)# interface em0
router.ntua.lab(config-if)# ip address 192.168.1.200/24
router.ntua.lab(config-if)# show interface em0
% Unknown command.
router.ntua.lab(config-if)# do show interface em0
Interface em0 is up, line protocol detection is disabled
  index 1 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>
  HWaddr: 08:00:27:95:e9:95
  inet 192.168.1.1/24 broadcast 192.168.1.255
  inet 192.168.1.200/24 broadcast 192.168.1.255 secondary
  inet6 fe80::a00:27ff:fe95:e995/64
    input packets 15, bytes 1356, dropped 0, multicast packets 3
    input errors 0
    output packets 13, bytes 1054, multicast packets 5
    output errors 0
    collisions 0

```

2.12

Από το αρχικό κέλυφος UNIX βλέπουμε τις ίδιες ουσιαστικά πληροφορίες με “ifconfig em0”, χωρίς όμως να αναγράφεται ότι η νέα διεύθυνση είναι secondary:

```

[root@router]~# ifconfig em0
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
  options=9b<RXCSUM, TXCSUM, VLAN_MTU, VLAN_HWTAGGING, VLAN_HWCSUM>
  ether 08:00:27:95:e9:95
  inet6 fe80::a00:27ff:fe95:e995%em0 prefixlen 64 scopeid 0x1
  inet 192.168.1.1 netmask 0xfffff000 broadcast 192.168.1.255
  inet 192.168.1.200 netmask 0xfffff000 broadcast 192.168.1.255
  nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
  media: Ethernet autoselect (1000baseT <full-duplex>)
  status: active

```


2.13 Χρησιμοποιούμε την εξής αλληλουχία εντολών: “configure terminal”, “interface em0”, “no ip address 192.168.1.200/24”. Εκτελώντας την εντολή do show interface em0 βλέπουμε ότι η IP έχει αφαιρεθεί:

```
router.ntua.lab(config-if)# do show interface em0
Interface em0 is up, line protocol detection is disabled
  index 1 metric 1 mtu 1500
  flags: <UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>
  HWaddr: 08:00:27:95:e9:95
  inet 192.168.1.1/24 broadcast 192.168.1.255
  inet6 fe80::a00:27ff:fe95:e995/64
    input packets 15, bytes 1356, dropped 0, multicast packets 3
    input errors 0
    output packets 13, bytes 1054, multicast packets 5
    output errors 0
    collisions 0
```

2.14 Με την εντολή “write file”

```
router.ntua.lab# write file
Building Configuration...
Configuration saved to /usr/local/etc/quagga/zebra.conf
Configuration saved to /usr/local/etc/quagga/ripd.conf
Configuration saved to /usr/local/etc/quagga/ripngd.conf
Configuration saved to /usr/local/etc/quagga/ospfd.conf
Configuration saved to /usr/local/etc/quagga/ospf6d.conf
Configuration saved to /usr/local/etc/quagga/bgpd.conf
Configuration saved to /usr/local/etc/quagga/isisd.conf
[OK]
```

2.15 Τα αρχεία που ενημερώνονται φαίνονται στο παραπάνω screenshot

2.16 Από το αρχικό κέλυφος UNIX εκτελούμε config save, ώστε να παραμείνουν οι ρυθμίσεις μετά την επανεκκίνηση

Άσκηση 3

3.1 Συνδέομαι με vtysh, και εκτελώ τις εντολές show interface em0 και show ip route. Οι ρυθμίσεις είναι αυτές που είχαμε ορίσει στα προηγούμενα ερωτήματα

3.2 Στο R1 ξεκινάμε με “cli”. Στη συνέχεια όντας σε Privileged EXEC mode εκτελούμε “configure terminal” και από εκεί “interface em1”, καθώς η em0 είναι ήδη κατάλληλα ρυθμισμένη. Όντας στο configuration της em1 εκτελούμε “ip address 172.17.17.1/30”. Παρατηρούμε ξανά πως η διεύθυνση που είχαμε αναθέσει προηγουμένως δεν έχει αφαιρεθεί από το “show interface em1”, οπότε εκτελούμε “no ip address 192.168.1.2/24”

3.3 Στο R2 ξεκινάμε με “cli”. Στη συνέχεια όντας σε Privileged EXEC mode εκτελούμε “configure terminal” και από εκεί “interface em0”, καθώς η em1 είναι ήδη κατάλληλα ρυθμισμένη. Όντας στο configuration της em0 εκτελούμε “ip address 172.17.17.2/30”. Παρατηρούμε ξανά πως η διεύθυνση που είχαμε αναθέσει προηγουμένως δεν έχει αφαιρεθεί από το “show interface em0”, οπότε εκτελούμε “no ip address 192.168.1.1/24”

3.4 Όντας σε configuration mode στο R1 εκτελούμε “ip route 192.168.2.0/24 172.17.17.2”

3.5 Όντας σε configuration mode στο R2 εκτελούμε “ip route 192.168.1.0/24 172.17.17.1”

3.6

```
root@PC1:~ # telnet 192.168.1.1 2601
Trying 192.168.1.1...
Connected to 192.168.1.1.
Escape character is '^]'.
Vty password is not set.
Connection closed by foreign host.
```

Πρέπει να ορίσουμε password στον R1. Οπότε στο R1, cli -> configure terminal -> password ntua.

3.7 Εκτελώντας “list” είτε σε απλό είτε σε privileged mode βλέπουμε πως δεν υπάρχει η εντολή “telnet”, επομένως δε μπορούμε να συνδεθούμε στο R2. Αν επιχειρήσουμε να εκτελέσουμε την εντολή “telnet 172.17.17.2 2601” τότε παίρνουμε σφάλμα περί “unknown command”.

3.8 Ναι μπορούμε κανονικά

3.9 Θα κάναμε telnet στη διεύθυνση 192.168.2.1, καθώς για το υποδίκτυο 192.168.2.0/24 έχουμε δρομολόγηση, σε αντίθεση με του 172.17.17.0/30.

3.10 Εκτελούμε από το PC2 “telnet 192.168.1.1 2601” και μας ζητείται κωδικός, οπότε και εισάγουμε το “ntua”. Με την εντολή “who” βλέπουμε τους απομακρυσμένους χρήστες και όχι αυτόν που συνδέθηκε τοπικά με vtysh.

```
router.ntua.lab> who
vty[18] connected from 192.168.2.2.
router.ntua.lab>
```

3.11 Όχι

3.12 Από την απομακρυσμένη στο R2 σύνδεση δεν αναγνωρίζεται ούτε η εντολή ping, αλλά ούτε και η traceroute. Από την τοπική μέσω vtysh σύνδεση στο R2 αναγνωρίζονται μεν οι εντολές, ωστόσο, δε λαμβάνουμε απάντηση από το PC1.

```
R2(config)# do ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
```

```
R2(config)# do traceroute 192.168.1.2
traceroute to 192.168.1.2 (192.168.1.2), 64 hops max, 40 byte packets
 1  172.17.17.1 (172.17.17.1)  0.357 ms  0.257 ms  0.252 ms
 2  * * *
 3  * * *
 4  * * *
```

3.13 Διότι τα PC1 και PC2 δε μπορούν να απαντήσουν στις διεπαφές των R2 και R1 αντίστοιχα, οι οποίες βρίσκονται στο WAN1. Αντίθετα τα PC1 ΚΑΙ PC2 έχουν στο routing table εγγραφές για τα LAN2 και LAN1 αντίστοιχα, οπότε και τα ping προς ή από PC1 και PC2 επιτυγχάνουν

3.14 Στο PC1: vtysh -> configure terminal -> no ip route 192.168.2.0/24 192.168.1.1 -> ip route 172.17.17.2/30 192.168.1.1

3.15 Στο PC2: vtysh -> configure terminal -> no ip route 192.168.1.0/24 192.168.2.1 -> ip route 172.17.17.1/30 192.168.2.1

3.16 Ναι, τώρα είναι επιτυχές

Άσκηση 4

4.1 Οι IP υπάρχουν ήδη από πριν στα PC, επομένως ορίζουμε τις προκαθορισμένες διαδρομές από το vtysh -> configure terminal με τις εντολές "ip route 0.0.0.0/0 192.168.1.1" και "ip route 0.0.0.0/0 192.168.2.1" στα PC1 και PC2 αντίστοιχα.

4.2 Με το cli βρισκόμαστε σε privileged EXEC mode και εκτελούμε τις εξής εντολές: "configure terminal" → "interface em0" → "ip address 192.168.1.1/24" → "exit" → "interface em1" → "ip address 172.17.17.1/30" → "exit" → "interface em2" → "ip address 172.17.17.5/30" → "exit". Επίσης hostname R1.

4.3 Εκτελούμε όντας στο Configuration mode "ip route 192.168.2.0/24 172.17.17.2"

4.4

```
R1(config)# do show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo0
C>* 172.17.17.0/30 is directly connected, em1
C>* 172.17.17.4/30 is directly connected, em2
C>* 192.168.1.0/24 is directly connected, em0
S>* 192.168.2.0/24 [1/0] via 172.17.17.2, em1
```

4.5

Routing tables

Internet:

Destination	Gateway	Flags	Refs	Use	Netif	Expire
127.0.0.1	link#4	UH	0	219	lo0	
172.17.17.0/30	link#2	U	0	0	em1	
172.17.17.1	link#2	UHS	0	0	lo0	
172.17.17.4/30	link#3	U	0	0	em2	
172.17.17.5	link#3	UHS	0	0	lo0	
192.168.1.0/24	link#1	U	0	0	em0	
192.168.1.1	link#1	UHS	0	0	lo0	
192.168.2.0/24	172.17.17.2	UG1	0	0	em1	

Παρατηρούμε πως εδώ εμφανίζονται επιπλέον οι Hosts που ανήκουν σε κάποιο υποδίκτυο, ενώ παραπάνω είχαμε μόνο τα υποδίκτυα.

4.6 Εμφανίζονται οι σημαίες UG1, οι οποίες σημαίνουν πως η διαδρομή είναι ενεργή (U), ο προορισμός είναι πύλη, η οποία θα αποφασίσει για την περαιτέρω προώθηση των πακέτων (G) και ότι εμπλέκεται συγκεκριμένο πρωτόκολλο δρομολόγησης (Protocol specific routing flag #1 (1)).

4.7 Με το cli βρισκόμαστε σε privileged EXEC mode και εκτελούμε τις εξής εντολές: “configure terminal” → “interface em0” → “ip address 172.17.17.2/30” → “exit” → “interface em1” → “ip address 192.168.2.1/24” → “exit” → “interface em2” → “ip address 172.17.17.9/30” → “exit”. Επίσης hostname R2.

4.8 Εκτελούμε όντας στο Configuration mode “ip route 192.168.1.0/24 172.17.17.1”.

4.9 Με το cli βρισκόμαστε σε privileged EXEC mode και εκτελούμε τις εξής εντολές: “configure terminal” → “interface em0” → “ip address 172.17.17.2/30” → “exit” → “interface em1” → “ip address 192.168.2.1/24” → “exit” → “interface em2” → “ip address 172.17.17.9/30” → “exit”. Επίσης hostname R3.

4.10 Σε configuration mode: “ip route 192.168.1.0/24 172.17.17.5” και “ip route 192.168.2.0/24 172.17.17.9”.

4.11 Ναι, είναι επιτυχές καθώς η ip em2 του R1 ανήκει στο ίδιο υποδίκτυο με τη διεπαφή R3 στο WAN2

4.12 Όχι, δεν είναι επιτυχές καθώς αρχικά το προωθεί στη default gateway R2 και από εκεί η R2 αποτυγχάνει με “No route to host” καθώς δεν ανήκει στο ίδιο υποδίκτυο με τη διεπαφή του R3 στο WAN2

4.13 Ακολουθούν τη διαδρομή PC1 → R1 → R2 → PC2

```

root@PC1:~ # traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  0.388 ms  0.300 ms  0.272 ms
 2  172.17.17.2 (172.17.17.2)  0.805 ms  0.526 ms  0.443 ms
 3  192.168.2.2 (192.168.2.2)  0.637 ms  0.565 ms  0.543 ms

```

Άσκηση 5

5.1 Η διαδρομή από το R1 προς το LAN2 μέσω του R2 ορίστηκε στατικά με τη default διαχειριστική απόσταση, επομένως είχε κόστος 1. Ορίζουμε εναλλακτική διαδρομή από το R1 προς το LAN2 μέσω του R3 με κόστος 2 με τις εντολές “configure terminal” → “ip route 192.168.2.0/24 172.17.17.6 2”. Έχουμε τον παρακάτω πίνακα δρομολόγησης:

```

router.ntua.lab(config)# do show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo0
C>* 172.17.17.0/30 is directly connected, em1
C>* 172.17.17.4/30 is directly connected, em2
C>* 192.168.1.0/24 is directly connected, em0
S   192.168.2.0/24 [2/0] via 172.17.17.6, em2
S>* 192.168.2.0/24 [1/0] via 172.17.17.2, em1

```

5.2 Δώσαμε την τιμή 2 ως αμέσως μεγαλύτερη της 1

5.3 Αντίστοιχα από το R2 εκτελούμε “ip route 192.168.1.0/24 172.17.17.10 2”.

5.4 Υπάρχουν οι παρακάτω εγγραφές για τα LAN1 και LAN2 των R1 και R2 αντίστοιχα:

```

S   192.168.2.0/24 [2/0] via 172.17.17.6, em2
S>* 192.168.2.0/24 [1/0] via 172.17.17.2, em1

```

```

S   192.168.1.0/24 [2/0] via 172.17.17.10, em2
S>* 192.168.1.0/24 [1/0] via 172.17.17.1, em0

```

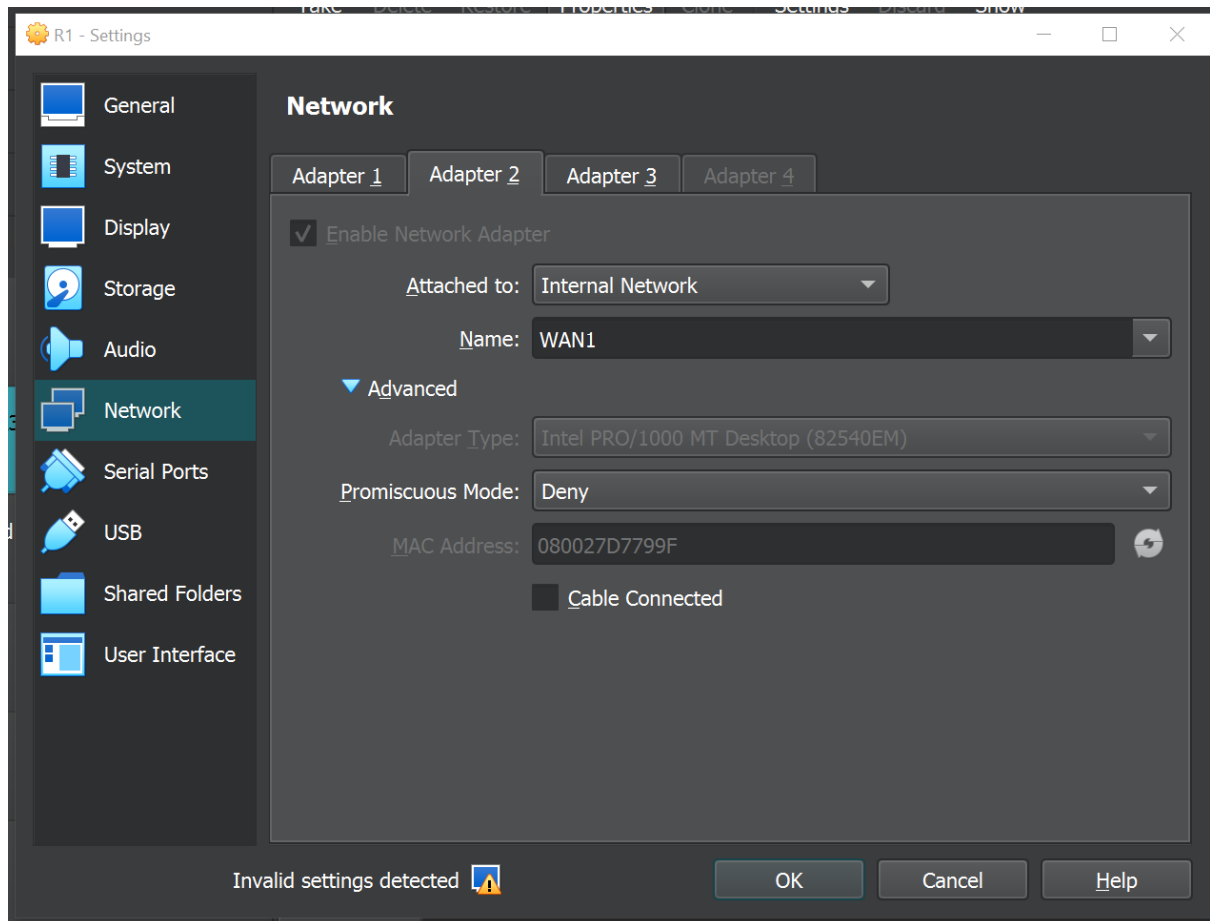
5.5 Ενεργοποιημένη είναι η διαδρομή μέσω του R2, πράγμα που υποδηλώνεται από το βελάκι (> - Selected route) και τον αστερίσκο (* - Forward Information Base (FIB) route).

5.6 Εμφανίζεται εντός των αγκυλών, αμέσως μετά τον προορισμό, όπως φαίνεται παραπάνω.

5.7 Είναι ενεργοποιημένη η διαδρομή μέσω R1

5.8 Εκτελούμε στον R1: “configure terminal” → “interface em1” → “link-detect”. Αντίστοιχα στον R2: “configure terminal” → “interface em0” → “link-detect”

5.9 Θα αποεπιλέξουμε την επιλογή “Cable connected”:



5.10 Η διαδρομή μέσω του R3:

```
router.ntua.lab(config)# do show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo0
C>* 172.17.17.4/30 is directly connected, em2
C>* 192.168.1.0/24 is directly connected, em0
S>* 192.168.2.0/24 [2/0] via 172.17.17.6, em2
S   192.168.2.0/24 [1/0] via 172.17.17.2 inactive
```

5.11 Υπάρχει η ένδειξη “inactive”

5.12 Ναι, η διαδρομή προς το LAN2 από το R1 γίνεται μέσω του R3:

```
Routing tables

Internet:
Destination      Gateway           Flags      Refs      Use  Netif  Expire
127.0.0.1         link#4            UH          0         243   lo0
172.17.17.0/30    link#2            U           0           0   em1
172.17.17.1       link#2            UHS         0           0   lo0
172.17.17.4/30    link#3            U           0           4   em2
172.17.17.5       link#3            UHS         0           0   lo0
192.168.1.0/24    link#1            U           0          15   em0
192.168.1.1       link#1            UHS         0           0   lo0
192.168.2.0/24    172.17.17.6      UG1         0           0   em2
```

5.13 Στο R2 παραμένει ενεργοποιημένη η διαδρομή μέσω του R1 για το LAN1, αφού δεν απενεργοποιήσαμε το καλώδιο της em0 του R2.

```
R2(config)# do show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo0
C>* 172.17.17.0/30 is directly connected, em0
C>* 172.17.17.8/30 is directly connected, em2
S   192.168.1.0/24 [2/0] via 172.17.17.10, em2
S>* 192.168.1.0/24 [1/0] via 172.17.17.1, em0
C>* 192.168.2.0/24 is directly connected, em1
```

5.14 Ναι, η διαδρομή προς το LAN1 από το R2 γίνεται μέσω του R3:

```

R2(config)# do show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo0
C>* 172.17.17.8/30 is directly connected, em2
S>* 192.168.1.0/24 [2/0] via 172.17.17.10, em2
S   192.168.1.0/24 [1/0] via 172.17.17.1 inactive
C>* 192.168.2.0/24 is directly connected, em1

```

5.15

```

root@PC1:~ # traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  0.677 ms  0.303 ms  0.286 ms
 2  172.17.17.6 (172.17.17.6)  0.799 ms  0.523 ms  0.500 ms
 3  172.17.17.9 (172.17.17.9)  0.902 ms  0.522 ms  0.487 ms
 4  192.168.2.2 (192.168.2.2)  0.349 ms  0.671 ms  1.076 ms

```

5.16 Εκτελούμε από το PC2 “ssh lab@192.168.1.2” και ως password “ntua”. Επαναφέροντας και τα δύο καλώδια, η σύνδεση δε διακόπτεται.

5.17 Κάνοντας ξανά traceroute από το PC1 βλέπουμε πως ακολουθείται η διαδρομή PC1 → R1(em0) → R2(em0) → PC2

```

root@PC1:~ # traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  0.402 ms  0.289 ms  0.236 ms
 2  172.17.17.2 (172.17.17.2)  0.945 ms  0.534 ms  0.573 ms
 3  192.168.2.2 (192.168.2.2)  0.708 ms  0.724 ms  0.549 ms

```

Άσκηση 6

6.1

- R1: “interface lo0” → “ip address 172.22.22.1/32”
- R2: “interface lo0” → “ip address 172.22.22.2/32”
- R3: “interface lo0” → “ip address 172.22.22.3/32”

6.2 Από το PC1 επιτυγχάνει το Ping στην lo0 του R1 μόνο, ενώ από το PC2 στην lo0 του R2 μόνο. Τα υπόλοιπα ping απαντάνε με “Destination host unreachable” Ο λόγος είναι ότι το PC1 προωθεί το πακέτο στη default gateway δηλαδή την R1 η οποία γνωρίζει μόνο τη διεύθυνση loopback του R1. Ομοια για το ping από το PC2 στο loopback του R2

6.3 Εκτελούμε στο R1 σε configuration mode “ip route 172.22.22.2/32 172.17.17.2” → “ip route 172.22.22.3/32 172.17.17.6”

6.4 Εκτελούμε στο R2 σε configuration mode “ip route 172.22.22.1/32 172.17.17.1” → “ip route 172.22.22.3/32 172.17.17.10”.

6.5 Εκτελούμε στο R3 σε configuration mode “ip route 172.22.22.1/32 172.17.17.5” → “ip route 172.22.22.2/32 172.17.17.9”.

6.6 Ναι, όλα τα ping είναι επιτυχή

6.7 Στο R1 τα ICMP echo requests έχουν αποστολέα την IP 172.17.17.6 (em0 του R3), ενώ στο R2 την 172.17.17.10 (em1 του R3).

6.8 Θα προσθέσουμε στο ping το όρισμα “-S 172.22.22.3”.

6.9 Θα έπρεπε να ορίζονται ξεχωριστές εγγραφές για κάθε προορισμό, αντί να ορίσουμε μόνο default gateway. Έτσι, δε θα μπορούσαν να δρομολογηθούν πακέτα για τα οποία δεν υπάρχουν εγγραφές στον πίνακα δρομολόγησης

6.10

Θα πετύχαιναν τα ping

- PC1 → R1
- PC1 → R3
- PC2 → R2
- PC2 → R3

Θα αποτυγχάνανε τα

- PC1 → R2
- PC2 → R1

6.11 “ip route 172.22.22.2/32 172.17.17.6 ” και “ip route 172.22.22.3/32 172.17.17.2 ”

6.12 “ip route 172.22.22.1/32 172.17.17.10 ” και “ip route 172.22.22.3/32 172.17.17.1 ”

6.13 “ip route 172.22.22.1/32 172.17.17.9 ” και “ip route 172.22.22.2/32 172.17.17.5 ”

6.14 Βλέπουμε πως έχει επιλεγθεί η συντομότερη, δηλαδή R1 → R2:

```

router.ntua.lab(config)# do show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo0
C>* 172.17.17.0/30 is directly connected, em1
C>* 172.17.17.4/30 is directly connected, em2
C>* 172.22.22.1/32 is directly connected, lo0
S>* 172.22.22.2/32 [1/0] via 172.17.17.2, em1
                        via 172.17.17.6, em2
S>* 172.22.22.3/32 [1/0] via 172.17.17.6, em2
                        via 172.17.17.2, em1
C>* 192.168.1.0/24 is directly connected, em0
S   192.168.2.0/24 [2/0] via 172.17.17.6, em2
S>* 192.168.2.0/24 [1/0] via 172.17.17.2, em1

```

6.15 Παρατηρούμε πως είναι inactive:

```

router.ntua.lab(config)# do show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo0
C>* 172.17.17.4/30 is directly connected, em2
C>* 172.22.22.1/32 is directly connected, lo0
S> 172.22.22.2/32 [1/0] via 172.17.17.2 inactive
    *                via 172.17.17.6, em2
S>* 172.22.22.3/32 [1/0] via 172.17.17.6, em2
                        via 172.17.17.2 inactive
C>* 192.168.1.0/24 is directly connected, em0
S>* 192.168.2.0/24 [2/0] via 172.17.17.6, em2
S   192.168.2.0/24 [1/0] via 172.17.17.2 inactive

```

6.16 Δε μας εμφανίζει τις διαδρομές του WAN2 ως inactive, καθώς δεν έχουμε ορίσει τις διεπαφές (R1-em2) και (R3-em0) στο WAN2 να κάνουν link-detect.

Άσκηση 7

7.1 Εκτελούμε στο C1 σε configuration mode “ip route 192.168.1.0/24 10.0.1.1” και “ip route 192.168.1.0/24 10.0.0.2 2”. Επίσης, “ip route 192.168.2.0/24 10.0.2.1” και “ip route 192.168.2.0/24 10.0.0.2 2”.

7.2 Εκτελούμε στο C2 σε configuration mode “ip route 192.168.1.0/24 10.0.1.5” και “ip route 192.168.1.0/24 10.0.0.1 2”. Επίσης, “ip route 192.168.2.0/24 10.0.2.5” και “ip route 192.168.2.0/24 10.0.0.1 2”.

7.3 Εκτελούμε στο R1 “ip route 192.168.2.0/24 10.0.1.2” και “ip route 192.168.2.0/24 10.0.1.6 2”.

7.4 Εκτελούμε στο R2 “ip route 192.168.1.0/24 10.0.2.2” και “ip route 192.168.1.0/24 10.0.2.6 2”.

7.5 Επικοινωνούν κανονικά

7.6 Ναι, συνεχίζουν να επικοινωνούν κανονικά

7.7 Με traceroute από το PC1 στο PC2 βλέπουμε να ακολουθείται η διαδρομή PC1 → R1 → C1 → C2 → R2 → PC2:

```
root@PC1:~ # traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  0.323 ms  0.341 ms  0.270 ms
 2  10.0.1.2 (10.0.1.2)  0.556 ms  0.524 ms  0.522 ms
 3  10.0.1.6 (10.0.1.6)  0.639 ms  0.630 ms  0.506 ms
 4  10.0.2.5 (10.0.2.5)  0.668 ms  0.684 ms  0.715 ms
 5  192.168.2.2 (192.168.2.2)  0.938 ms  0.679 ms  0.867 ms
```

Από το PC2 στο PC1 έχουμε PC2→R2->C2 → R1 → PC1:

```
root@PC2:~ # traceroute 192.168.1.2
traceroute to 192.168.1.2 (192.168.1.2), 64 hops max, 40 byte packets
 1  192.168.2.1 (192.168.2.1)  0.383 ms  0.312 ms  0.272 ms
 2  10.0.2.6 (10.0.2.6)  0.529 ms  0.473 ms  0.431 ms
 3  10.0.1.1 (10.0.1.1)  0.843 ms  0.690 ms  0.645 ms
 4  192.168.1.2 (192.168.1.2)  0.827 ms  0.776 ms  0.809 ms
```

7.8 Από το προηγούμενο ερώτημα βλέπουμε το traceroute. Παρατηρώ ότι υπάρχει μια αναντιστοιχία σε διεύθυνση από την οποία διέρχονται τα πακέτα, καθώς δεν επιστρέφονται από αυτή. Αρχικά, το PC1 στέλνει τα πακέτα, το R1 τα προωθεί μέσω της 10.0.1.1 και απαντάει στο PC1 μέσω της **192.168.1.1**. Στη συνέχεια, πάνε μέσω της κύριας διαδρομής για το LAN2, δηλαδή στο C1, το οποίο απαντάει πίσω από την **10.0.1.2**, αλλά προωθεί μέσω της 10.0.0.1. Από εκεί, τα λαμβάνει το C2, το οποίο απαντάει μέσω της **10.0.1.6** και προωθεί μέσω της 10.0.2.6. Μετά, λαμβάνονται από το R2, το οποίο απαντάει μέσω της δευτερεύουσας διαδρομής, δηλαδή προς το C2 από την **10.0.2.5** δεδομένου ότι έχουμε κόψει το WAN2. Τέλος, απαντάει ο παραλήπτης PC2 από την **192.168.2.2**. Επομένως, η αναντιστοιχία ουσιαστικά βρίσκεται στο βήμα C1 → C2, όπου λαμβάνουμε απάντηση από την em1 του C2 αντί της em0. Αυτό είναι λογικό, αφού στον πίνακα δρομολόγησης του C2 υπάρχει εγγραφή που οδηγεί τα πακέτα με προορισμό το δίκτυο 192,168,1,0/24 στη διεπαφή του R1 μέσω του WAN3

7.9 Απέχει 4 βήματα από το PC1. Αντίστοιχα, το PC2 στέλνει, τα λαμβάνει η **192.168.2.1** του R2, η οποία και απαντάει και τα προωθεί η 10.0.2.5 στο C2. Το C2 τα συλλαμβάνει από την

10.0.2.6, οπότε και απαντάει από αυτήν και τα προωθεί από την 10.0.1.6 στην 10.0.1.5 του R1. Το R1, παρόλο που παρέλαβε από την 10.0.1.5 απαντάει από την **10.0.1.1**, αφού η κύρια διαδρομή από το R1 για το LAN2 ορίσαμε να είναι μέσω του C1. Τέλος, αφού το R1 προωθήσει το πακέτο μέσω της 192.168.1.1, το PC1 απαντάει από την **192.168.1.2**

7.10 Συνεχίζουν να επικοινωνούν και η διαδρομή των πακέτων είναι

(PC1 → R1 → C1 → C2 → R2 → PC2) ή

(PC2 → R2 → C2 → C1 → R1 → PC1)

```
root@PC1:~ # traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  0.671 ms  0.305 ms  0.320 ms
 2  10.0.1.2 (10.0.1.2)  0.891 ms  0.597 ms  0.488 ms
 3  10.0.0.2 (10.0.0.2)  0.947 ms  0.704 ms  0.654 ms
 4  10.0.2.5 (10.0.2.5)  3.029 ms  0.848 ms  0.798 ms
 5  192.168.2.2 (192.168.2.2)  1.211 ms  0.982 ms  0.874 ms
```

```
root@PC2:~ # traceroute 192.168.1.2
traceroute to 192.168.1.2 (192.168.1.2), 64 hops max, 40 byte packets
 1  192.168.2.1 (192.168.2.1)  0.384 ms  0.306 ms  0.282 ms
 2  10.0.2.6 (10.0.2.6)  0.518 ms  0.454 ms  0.476 ms
 3  10.0.0.1 (10.0.0.1)  0.730 ms  0.590 ms  2.663 ms
 4  10.0.1.1 (10.0.1.1)  0.793 ms  0.854 ms  0.729 ms
 5  192.168.1.2 (192.168.1.2)  0.980 ms  1.047 ms  0.914 ms
```

7.11 Το ping αποτυγχάνει

7.12 Το μεγαλύτερο μειονέκτημα μιας τέτοια τοπολογίας εταιρικού δικτύου είναι η πολυπλοκότητα. Στο παράδειγμα μας έχουμε μόνο 2 LAN δίκτυα. Με την προσθήκη πολλών δικτύων LAN δικτύων, τα πράγματα γίνονται αρκετά σύνθετα στη διαχείριση και προσθήκη εγγραφών στα routers.