



---

## ΕΡΓΑΣΤΗΡΙΟ ΔΙΚΤΥΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

---

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 6: ΕΙΣΑΓΩΓΗ ΣΤΟ QUAGGA ΚΑΙ FRROUTING (FRR)



12 ΑΠΡΙΛΙΟΥ, 2023

ΘΟΔΩΡΗΣ ΑΡΑΠΗΣ – EL18028

Όνοματεπώνυμο: Θεodorής Αράπης	Ομάδα: 3
Όνομα PC/ΛΣ: DESKTOP-JGHL94V/ WINDOWS 10	Ημερομηνία: 5/4/2023

## Άσκηση 1 (προετοιμασία): Γνωριμία με το περιβάλλον του FRR

1

Εκτελούμε τις οδηγίες.

2

Εκτελούμε “dhclient em0”.

3

Εκτελούμε “ping www.google.com” και επιτυγχάνει.

4

Εκτελούμε την εντολή “pkg update”.

5

Εκτελούμε την εντολή “pkg install frr”.

6

Εκτελούμε την εντολή “sysctl kern.ipc.maxsockbuf=16777216”.

7

Εκτελούμε “chown frr:frr /usr/local/etc/frr” οπότε και βλέπουμε το αποτέλεσμα:

```
root@FRRrouter:/usr/local/etc # ls -al
total 32
drwxr-xr-x  6 root  wheel  512 Apr 10 18:31 .
drwxr-xr-x 10 root  wheel  512 Apr 10 18:31 ..
drwxr-xr-x  2 root  wheel  512 Apr 10 18:28 bash_completion.d
drwxr-xr-x  2 frr   frr    512 Apr 10 18:31 frr
drwxr-xr-x  5 root  wheel  512 Apr 10 18:28 periodic
-rw-r--r--  1 root  wheel 2392 Apr  2 01:07 pkg.conf
-rw-r--r--  1 root  wheel 2392 Apr  2 01:07 pkg.conf.sample
drwxr-xr-x  2 root  wheel  512 Apr 10 18:31 rc.d
```

## 8

Εκτελούμε στο /usr/local/etc/frr **"touch vtysh.conf zebra.conf staticd.conf"**.

## 9

Εκτελούμε όντας στο directory /usr/local/etc/frr **"chown frr:frr vtysh.conf"**, **"chown frr:frr zebra.conf"** και **"chown frr:frr staticd.conf"**.

## 10

Διαμορφώνουμε κατάλληλα το αρχείο:

```
sshd_enable="YES" # to enable the ssh daemon
hostname="R0" # to assign the host name
syslogd_flags="-scc" # to disable compression of repeated messages
gateway_enable="YES"
frr_enable="YES"
frr_daemons="zebra staticd"
```

## 11

Διαμορφώνουμε κατάλληλα το αρχείο:

```
# $FreeBSD: releng/12.4/bin/csh/csh.cshrc 337849 2018-08-15 14:41:24Z brd $
#
# System-wide .cshrc file for csh(1).
setenv UTYSH_PAGER "more -EFX"
```

## 12

Εκτελούμε **"rm /etc/resolv.conf"** και **"poweroff"**.

## 13

Βάζουμε όλες τις κάρτες σε εσωτερική δικτύωση στα LAN1, LAN2, LAN3 και LAN4.

## 14

Με τις εντολές **"service sshd onestatus"** και **"service frr status"** βλέπουμε αν οι υπηρεσίες αυτές είναι ενεργοποιημένες.

```
root@R0:~ # service sshd onestatus
sshd is running as pid 1120.
root@R0:~ # service frr status
zebra is running as pid 876.
staticd is running as pid 879.
```

## 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

Εκτελούμε “**telnet localhost 2601**” και λαμβάνουμε το σφάλμα “Vty password is not set, Connection closed by foreign host”:

```
root@R0:~ # 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 τις εξής 22:

```
R0#
add          Add registration
clear        Reset functions
configure    Configuration from vty interface
copy         Copy from one file to another
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
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, ωστόσο εν προκειμένω λαμβάνουμε και “Command Incomplete”.

#### 1.5

Εάν πληκτρολογήσουμε “co” και μετά TAB δε κάνει τίποτα, ωστόσο εάν ξανα πατήσουμε 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 “sh version”.

#### 1.7

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

#### 1.8

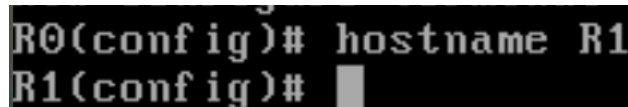
Με την εντολή “show running-config”.

#### 1.9

Με την εντολή “configure terminal”.

#### 1.10

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



```
R0(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

Μόνο τις παρακάτω 9:

```
R0>
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
```

### 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”**, βλέπουμε πως είναι ενεργοποιημένη.

### 1.19

Με την εντολή **“show ip route”**, δε παίρνουμε καμία απάντηση, επομένως 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.

### 1.23

Εκτελούμε την εντολή **“list”**.

### 1.24

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

### 1.25

Με την εντολή **“service password-encryption”**.

### 1.26

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

## Άσκηση 2: Δρομολόγηση σε ένα βήμα

### 2.1

Εκτελούμε **“ifconfig em0 192.168.1.2/24”** στο PC1 και **“ifconfig em0 192.168.2.2/24”** στο PC2.

### 2.2

Αρχικά συνδεόμαστε με **“vtysh”** 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.

### 2.3

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

```
Interface em0 is up, line protocol is up
Link ups:      1      last: 2023/04/11 01:45:11.50
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:b0:4e:80
inet 192.168.1.1/24
Interface Type Other
Interface Slave Type None
  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 em1 is up, line protocol is up
Link ups:      1      last: 2023/04/11 01:45:31.37
Link downs:    0      last: (never)
vrf: default
index 2 metric 1 mtu 1500 speed 1000
flags: <UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST>
Type: Ethernet
HWaddr: 08:00:27:3c:e7:34
inet 192.168.2.1/24
Interface Type Other
Interface Slave Type None
  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
```

## 2.4

Με “show ip forwarding” βλέπουμε πως είναι ενεργοποιημένη.

## 2.5

Εκτελούμε στο PC1 “route add -net 192.168.2.0/24 192.168.1.1”.

## 2.6

Εκτελούμε στο PC2 “route add -net 192.168.1.0/24 192.168.2.1”.

## 2.7

Ναι, επικοινωνούν κανονικά.



## 2.8

Χρησιμοποιούμε την εξής αλληλουχία εντολών: “**configure terminal**”, “**interface em0**”, “**ip address 192.168.1.200/24**” και μετά “**exit**” και ξανά “**exit**” για να πάμε στο privileged EXEC mode. Όπως βλέπουμε, η νέα διεύθυνση που αναθέσαμε μπήκε ως secondary:

```
Interface em0 is up, line protocol is up
  Link ups:      1      last: 2023/04/11 01:45:11.50
  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:b0:4e:80
  inet 192.168.1.1/24
  inet 192.168.1.200/24 secondary
  Interface Type Other
  Interface Slave Type None
    input packets 2, bytes 158, dropped 0, multicast packets 1
    input errors 0
    output packets 3, bytes 182, multicast packets 0
    output errors 0
    collisions 0
```

## 2.9

Από το αρχικό κέλυφος unix βλέπουμε τις ίδιες ουσιαστικά πληροφορίες με “ifconfig em0”:

```
root@R0:~ # ifconfig em0
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    options=81009b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM,VLAN_HW
FILTER>
    ether 08:00:27:b0:4e:80
    inet 192.168.1.1 netmask 0xffffff00 broadcast 192.168.1.255
    inet 192.168.1.200 netmask 0xffffff00 broadcast 192.168.1.255
    media: Ethernet autoselect (1000baseT <full-duplex>)
    status: active
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
```

## 2.10

Αρχικά μεταφερόμαστε πάλι σε Global Configuration Mode. Όντας εκεί, εκτελούμε “**interface em0**” και στη συνέχεια “**no ip address 192.168.1.200/24**”. Επιστρέφουμε με “**exit**” στο Global Configuration Mode και εκεί εκτελούμε “**exit**” ώστε να μεταφερθούμε στο Privileged EXEC mode. Εκεί εκτελούμε “**show interface em0**” και βλέπουμε πως δεν έχει αφαιρεθεί η secondary address. Ωστόσο, εκτελώντας “**write terminal**” βλέπουμε πως όντως έχει αφαιρεθεί η διεύθυνση:

### 2.11

Με την εντολή “write file”.

```
R1# write terminal
Building configuration...

Current configuration:
?
frr version 7.5.1
frr defaults traditional
hostname R0
service password-encryption
no ipv6 forwarding
hostname R1
?
password 8 B6uqtW0yYgkU7M
enable password 8 1CX2urnxpm/X2
password ntua
?
interface em0
 ip address 192.168.1.1/24
?
interface em1
 ip address 192.168.2.1/24
?
line vty
?
end
```

### 2.12

Τα αρχεία /usr/local/etc/frr/zebra.conf και /usr/local/etc/frr/staticd.conf.

```
R1# write file
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Configuration saved to /usr/local/etc/frr/zebra.conf
Configuration saved to /usr/local/etc/frr/staticd.conf
```

## Άσκηση 3: Δρομολόγηση σε περισσότερα βήματα

### 3.1

Τα PC είναι κατάλληλα ρυθμισμένα.

### 3.2

Στο R1 ξεκινάμε με **“vtysh”**. 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”**, αλλά το **“write terminal”** εμφανίζει τη σωστή παραμετροποίηση.

### 3.3

Όντας στο αρχικό shell του μηχανήματος R2, εκτελούμε **“vtysh”** → **“configure terminal”** → **“interface em1”** → **“ip address 172.17.17.2/30”** → **“exit”** → **“interface em0”** → **“ip address 192.168.2.1/24”** → **“exit”** → **“exit”** και ξαναβρισκόμαστε σε Privileged EXEC mode.

### 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

Κάνουμε από το PC1 **“telnet 192.168.1.1 2601”** και μας ζητείται password, οπότε και εισάγουμε το **“ntua”**, καθώς είχαμε προηγουμένως ορίσει αυτό. Αν δεν είχαμε ορίσει password δεν θα μπορούσαμε να συνδεθούμε.

### 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

Θα κάνουμε 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.

### 3.9

Εκτελούμε από το PC2 **"telnet 192.168.2.1 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.

```
R2> who
vty[28] connected from 192.168.2.2.
```

### 3.10

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)  1.274 ms  0.950 ms  0.728 ms
 2  * * *
 3  * * *
 4  * * *
```

### 3.11

Διότι τα PC1 και PC2 δε μπορούν να απαντήσουν στις διεπαφές των R2 και R1 αντίστοιχα, οι οποίες βρίσκονται στο WAN1.

### 3.12

Πρέπει να εκτελέσουμε τις εντολές **"route add -net 172.17.17.0/30 192.168.1.1"** στο PC1 και **"route add -net 172.17.17.0/30 192.168.2.1"** στο PC2. Πλέον, επιτυγχάνουν κανονικά:

```
R2# 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)  0.324 ms  0.153 ms  0.133 ms
 2  192.168.1.2 (192.168.1.2)  0.297 ms  0.334 ms  0.332 ms
```

## Άσκηση 4: Εναλλακτικές διαδρομές

### 4.1

Οι IP υπάρχουν ήδη από πριν στα PC, επομένως ορίζουμε τις προκαθορισμένες διαδρομές με τις εντολές **"route add default 192.168.1.1"** και **"route add default 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"**.

### 4.3

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

### 4.4

Εκτελούμε όντας στο Configuration mode **"do show ip route"** και βλέπουμε διαδρομές προς τα παρακάτω υποδίκτυα:

- 127.0.0.0/8 μέσω της loopback (lo0)
- 172.17.17.0/30 μέσω της em1
- 172.17.17.4/30 μέσω της em2
- 192.168.1.0/24 μέσω της em0
- 192.168.2.0/24 μέσω της em1 με ενδιάμεσο βήμα το 172.17.17.2

```
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 [1/0] via 172.17.17.2, em1
```

#### 4.5

Με το μήνυμα “is directly connected”.

#### 4.6

Βλέπουμε στα αριστερά “S>\*” αντί για “C>\*”.

#### 4.7

Εμφανίζονται οι παρακάτω πληροφορίες:

Internet:						
Destination	Gateway	Flags	Refs	Use	Netif	Expire
localhost	link#4	UH	0	171	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	link#1	U	0	0	em0	
192.168.1.1	link#1	UHS	0	0	lo0	
192.168.2.0	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.

#### 4.8

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))

#### 4.9

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

#### 4.10

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

#### 4.11

Με το cli βρισκόμαστε σε privileged EXEC mode και εκτελούμε τις εξής εντολές: “configure terminal” → “interface em0” → “ip address 172.17.17.6/30” → “exit” → “interface em1” → “ip address 172.17.17.10/30” → “exit”

#### 4.12

Όντας 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**”.

#### 4.13

Με “**show ip forwarding**” βλέπουμε πως είναι ενεργή.

#### 4.14

Βλέπουμε πως 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)  0.687 ms  0.211 ms  0.165 ms
 2  172.17.17.2 (172.17.17.2)  0.392 ms  0.323 ms  0.246 ms
 3  192.168.2.2 (192.168.2.2)  0.840 ms  0.465 ms  0.441 ms
```

### Άσκηση 5: Σφάλμα καλωδίου και αυτόματη αλλαγή στη δρομολόγηση

#### 5.1

Ε 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**” όντας σε privileged EXEC mode. Έχουμε πλέον τον παρακάτω πίνακα δρομολόγησης:

```
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 [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, όντας σε Global configuration mode 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, em1
```

## 5.5

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

## 5.6

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

## 5.7

Είναι ενεργοποιημένη η διαδρομή μέσω R1.

## 5.8

Όντας σε privileged exec mode 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", στην συγκεκριμένη διεπαφή.

## 5.10

Βλέπουμε πως είναι πλέον ενεργοποιημένη η διαδρομή μέσω του R3:

```
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”, η οποία μας ενημερώνει πως η διαδρομή δεν είναι διαθέσιμη.

### 5.12

Ναι, αντίθετα με πριν, πλέον η διαδρομή προς το LAN2 από το R1 γίνεται μέσω του R3.

### 5.13

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

### 5.14

Αποσυνδέουμε το καλώδιο και βλέπουμε μέσω του πίνακα δρομολόγησης του R2 πως η αλλαγή έγινε κανονικά.

```
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
```

### 5.15

Επιβεβαιώνουμε τα παραπάνω: (PC1 → R1(em0) → R3(em0) → R2(em2) → 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)  0.488 ms  0.187 ms  0.142 ms
 2  172.17.17.6 (172.17.17.6)  0.554 ms  0.321 ms  0.316 ms
 3  172.17.17.9 (172.17.17.9)  0.766 ms  0.405 ms  0.367 ms
 4  192.168.2.2 (192.168.2.2)  0.724 ms  0.590 ms  0.508 ms
```

### 5.16

Εκτελούμε από το PC2 “ssh lab@192.168.1.2” και ως password εισάγουμε το “ntua”. Επαναφέροντας τα 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)  0.259 ms  0.182 ms  0.170 ms
 2  172.17.17.2 (172.17.17.2)  0.315 ms  0.343 ms  0.238 ms
 3  192.168.2.2 (192.168.2.2)  0.479 ms  0.407 ms  0.396 ms
```

## Άσκηση 6: Διευθύνσεις διαχείρισης (loopback)

### 6.1

Όντας σε configuration mode σε κάθε δρομολογητή εκτελούμε:

- 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".

### 6.3

Εκτελούμε στο R1 όντας 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 όντας 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 όντας 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

Πλέον όλα τα Ping επιτυγχάνουν.

### 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 προσθέταμε στο ping το όρισμα **"-S 172.22.22.3"**.

### 6.9

Θα ορίζαμε ξεχωριστές εγγραφές για κάθε προορισμό, αντί να ορίσουμε μόνο default gateway, θα είχαμε μεγαλύτερη πιθανότητα για σφάλμα, καθώς και περισσότερες παραμέτρους να ελέγχουμε κατά το debugging.

### 6.10

Θα πετύχαιναν τα ping

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

ενώ θα αποτυγχάνανε τα

- PC1 → R2
- PC2 → R1

### 6.11

Εκτελούμε όντας σε configuration mode **"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

Εκτελούμε όντας σε configuration mode **"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

Εκτελούμε όντας σε configuration mode **"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 επιλεχθεί η συντομότερη, δηλαδή R1 → R2:

```
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>* 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 πλέον inactive:

```
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>* 172.17.17.4/30 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) να κάνουν link-detect.

## Άσκηση 7: Ένα εταιρικό δίκτυο

### 7.1

Εκτελούμε στο C1 όντας 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 όντας 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:

```
root@PC1:~ # ping -c 1 192.168.2.2
PING 192.168.2.2 (192.168.2.2): 56 data bytes
64 bytes from 192.168.2.2: icmp_seq=0 ttl=61 time=1.949 ms

--- 192.168.2.2 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.949/1.949/1.949/0.000 ms
```

### 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)  0.303 ms  0.156 ms  0.143 ms
 2  10.0.1.2 (10.0.1.2)  0.308 ms  0.343 ms  0.290 ms
 3  10.0.1.6 (10.0.1.6)  0.421 ms  0.380 ms  0.324 ms
 4  10.0.2.5 (10.0.2.5)  1.288 ms  0.522 ms  0.443 ms
 5  192.168.2.2 (192.168.2.2)  0.654 ms  0.969 ms  0.868 ms
```

Αντίστοιχα 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)  0.274 ms  0.230 ms  0.197 ms
 2  10.0.2.6 (10.0.2.6)  0.341 ms  0.314 ms  0.276 ms
 3  10.0.1.1 (10.0.1.1)  0.560 ms  0.559 ms  0.558 ms
 4  192.168.1.2 (192.168.1.2)  0.662 ms  0.592 ms  0.606 ms
```

## **7.8**

Βλέπουμε το traceroute παραπάνω. Εμφανίζεται μια αναντιστοιχία την οποία και θα δούμε αναλυτικά. 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 που βλέπουμε. 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. Ύστερα, 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 αυτού, διότι η κύρια διαδρομή από το C2 για το LAN1 είναι μέσω του R1, οπότε και ακολουθείται αυτή.

## **7.9**

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, καθώς 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).

## **7.11**

Το ping θα αποτύχει και θα λάβουμε ως απάντηση από το C1 “Time to live exceeded”, καθώς τα πακέτα θα εγκλωβιστούν σε ένα loop μεταξύ των C1 και C2.

## **7.12**

Παρουσιάζεται μεγάλη πολυπλοκότητα κατά την προσθήκη εγγραφών σε περίπτωση πολλών LAN δικτύων.