



ΕΡΓΑΣΤΗΡΙΟ ΔΙΚΤΥΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 7: ΔΥΝΑΜΙΚΗ ΔΡΟΜΟΛΟΓΗΣΗ RIP



3 ΜΑΙΟΥ, 2023

ΘΟΔΩΡΗΣ ΑΡΑΠΗΣ – ΕΙ18028

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

Προετοιμασία στο σπίτι

1

Σταματάμε το frr.

```
root@R2:~ # service frr stop
/etc/rc.conf: to: not found
/etc/rc.conf: to: not found
Stopping staticd.
Stopping zebra.
```

2

Δημιουργούμε με “**touch ripd.conf**” το αρχείο όντας στο directory /usr/local/etc/frr.

3

Εκτελούμε “**chown frr:frr ripd.conf**”.

4

Τροποποιούμε κατάλληλα το αρχείο:

```
root@R2:/usr/local/etc/frr # cat /etc/rc.conf
sshd_enable="YES" # to enable the ssh daemon
hostname="R2" #to assign the host name
gateway_enable="YES"
frr_enable="YES"
frr_daemons="zebra staticd ripd"
```

5

Εκκινούμε την υπηρεσία:

```
root@R2:~ # service frr start
/etc/rc.conf: to: not found
/etc/rc.conf: to: not found
Checking zebra.conf
2022/04/18 00:56:03 ZEBRA: [EC 4043309111] Disabling MPLS support (no kernel support)
OK
Starting zebra.
2022/04/18 00:56:03 ZEBRA: [EC 4043309111] Disabling MPLS support (no kernel support)
Checking staticd.conf
OK
Starting staticd.
Checking ripd.conf
OK
Starting ripd.
```

6

Δημιουργούμε το αντίστοιχα .ova.

Z

Αποθηκεύουμε το αρχείο.

1.1

Εκτελούμε τις εξής εντολές στο PC1: “**vtysh**” → “**configure terminal**” → “**hostname PC1**” → “**interface em0**” → “**ip address 192.168.1.2/24**” → “**exit**” → “**ip route 0.0.0.0/0 192.168.1.1**”.

1.2

Αντίστοιχα εκτελούμε στο PC2: “**vtysh**” → “**configure terminal**” → “**hostname PC2**” → “**interface em0**” → “**ip address 192.168.2.2/24**” → “**exit**” → “**ip route 0.0.0.0/0 192.168.2.1**”.

1.3

Εκτελούμε στο R1: “**cli**” → “**configure terminal**” → “**hostname R1**” → “**interface em0**” → “**ip address 192.168.1.1/24**” → “**exit**” → “**interface em1**” → “**ip address 172.17.17.1/30**”.

1.4

Εκτελούμε στο Configuration mode του R1 την εντολή “**do show ip route**” και επιβεβαιώνουμε πως δεν υπάρχουν στατικές εγγραφές:

```
R1(config-if)# 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>* 192.168.1.0/24 is directly connected, em0
```

1.5

Εκτελούμε στο R1 την εντολή “**router ?**” οπότε και μας εμφανίζονται οι διάφορες επιλογές που έχουμε στο σημείο αυτό, οι οποίες και αντιστοιχούν στα διαθέσιμα πρωτόκολλα δρομολόγησης:

```
R1(config)# router
  babel  Babel
  bgp    BGP information
  isis   ISO IS-IS
  ospf   Start OSPF configuration
  ospf6  Open Shortest Path First (OSPF) for IPv6
  rip    RIP
  ripng  RIPng
```

1.6

Εκτελούμε “**router rip**” στο R1.

1.7

Εκτελούμε “?” και βλέπουμε τις διαθέσιμες εντολές.

```
R1(config-router)# ?
  default-information          Control distribution of default route
  default-metric              Set a metric of redistribute routes
  distance                     Administrative distance
  end                          End current mode and change to enable mode
  exit                         Exit current mode and down to previous mode
  list                         Print command list
  neighbor                     Specify a neighbor router
  network                      Enable routing on an IP network
  no                           Negate a command or set its defaults
  offset-list                 Modify RIP metric
  passive-interface            Suppress routing updates on an interface
  quit                         Exit current mode and down to previous mode
  redistribute                 Redistribute information from another routing protocol
  relaxed-recv-size-checks   Abide other treatments of RFC for received packets
  route                        RIP static route configuration
  route-map                   Route map set
  timers                       Adjust routing timers
  version                      Set routing protocol version
```

1.8

Με την εντολή “**version 2**”.

1.9

Εκτελούμε “**network 192.168.1.0/24**”.

1.10

Εκτελούμε “**network 172.17.17.0/30**”.

1.11

Παρατηρούμε πως δεν έχει αλλάξει κάτι.

1.12

Επαναλαμβάνουμε τα προηγούμενα βήματα για το R2. Μεταξύ άλλων, βλέπουμε πως όντως δεν υπάρχουν στατικές εγγραφές:

```
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, em1
C>* 192.168.2.0/24 is directly connected, em0
```

Φτάνοντας στο βήμα 1.11 βλέπουμε πως έχει αλλάξει κατά μία εγγραφή ο πίνακας δρομολόγησης:

```
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, em1
R>* 192.168.1.0/24 [120/21] via 172.17.17.1, em1, 00:00:07
C>* 192.168.2.0/24 is directly connected, em0
```

Τα PC1, PC2 επικοινωνούν κανονικά, όπως διαπιστώνουμε από το επιτυχημένο Ping μεταξύ τους.

```
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=62 time=0.604 ms

--- 192.168.2.2 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.604/0.604/0.604/0.000 ms
```

1.13

Όπως είδαμε και παραπάνω, όντας στο configuration mode του R2 εκτελούμε “**do show ip route**”.

1.14

Βλέπουμε με “**show ip rip**” σε privileged exec mode τις εγγραφές για τα υποδίκτυα 172.17.17.0/30, 192.168.1.0/24, 192.168.2.0/24.

```
R1# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

      Network          Next Hop           Metric From        Tag Time
C(i) 172.17.17.0/30    0.0.0.0            1 self          0
C(i) 192.168.1.0/24    0.0.0.0            1 self          0
R(n) 192.168.2.0/24    172.17.17.2         2 172.17.17.2  0 02:33
```

1.15

Η εγγραφή 0.0.0.0 ως Next Hop υποδηλώνει το ίδιο το μηχάνημα R1, δεδομένου ότι δε γνωρίζει την IP του.

1.16

Όσον αφορά τα 2 πρώτα υποδίκτυα είναι άμεσα συνδεδεμένα με το R1 (Connected), ενώ το 3ο υποδίκτυο έχει προστεθεί λόγω του πρωτοκόλλου RIP. Η μετρική δηλώνει τα hops που απαιτούνται μέχρι κάποιον στόχο του δεδομένου υποδικτύου. Για παράδειγμα στα 2 πρώτα υποδίκτυα έχει τιμή 1, καθώς το πακέτο θα φτάσει αμέσως στον στόχο του, ενώ για το LAN2 έχει τιμή 2, καθώς θα περάσει πρώτα από το R2.

1.17

Βλέπουμε τις παρακάτω 4 εγγραφές:

```
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, em1
R>* 192.168.1.0/24 [120/2] via 172.17.17.1, em1, 00:06:39
C>* 192.168.2.0/24 is directly connected, em0
```

1.18

Το γράμμα R στην αρχή υποδηλώνει προσθήκη μέσω του RIP.

1.19

Με το >.

1.20

Με το *.

1.21

Οι διαδρομές RIP έχουν διαχειριστική απόσταση 120, όπως βλέπουμε και από τον παραπάνω πίνακα [120/2]. Αντίστοιχα, το μήκος της διαδρομής είναι 2 και εξάγεται από το ίδιο σημείο.

1.22

Όντας σε Global configuration mode εκτελούμε “**do show ip rip status**”. Όπως διαβάζουμε στα αποτελέσματα, αποστέλλονται ενημερώσεις κάθε 30 δευτερόλεπτα, τιμή η οποία μπορεί, ωστόσο, να κυμαίνεται στο +-50% αυτής (15 – 45 δευτερόλεπτα).

1.23

Είναι ενεργοποιημένο σε αμφότερες τις em0, em1 και μετέχουν τα δίκτυα 172.17.17.0/30 και 192.168.1.0/24.

```

R1# show ip rip status
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 17 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Relaxed receiving size checks are off
  Redistributing:
  Default version control: send version 2, receive version 2
    Interface      Send   Recv   Key-chain
    em0            2       2
    em1            2       2
  Routing for Networks:
    172.17.17.0/30
    192.168.1.0/24
  Routing Information Sources:
    Gateway          BadPackets  BadRoutes  Distance  Last Update
    172.17.17.2           0          0        120     00:00:27
  Distance: (default is 120)

```

1.24

Όπως βλέπουμε από το παραπάνω στιγμιότυπο στο πεδίο “Routing Information Sources”, ο R1 λαμβάνει πληροφορίες από την em1 του R2 (172.17.17.2). Ο χρόνος τελευταίας ενημέρωσης (Last Update) δηλώνει πόσος χρόνος πέρασε από την τελευταία ενημέρωση από την πηγή αυτή.

1.25

Ο χρόνος που εμφανίζεται με “**do show ip rip**” δηλώνει τον χρόνο που απομένει μέχρι η διαδρομή να παύσει να ισχύει (timeout – 180 sec), για αυτό και βλέπουμε πως ξεκινάει εκ νέου από τα 3 λεπτά (180 sec) κάθε φορά που έρχεται νέο update, δηλαδή κάθε φορά που το timer φτάνει περίπου στα δυόμιση λεπτά.

1.26

Εκτελώντας στο μηχάνημα R1 “**netstat -r**” βλέπουμε τις παρακάτω εγγραφές, εκ των οπίων διακρίνουμε πως αυτή που αφορά το LAN2 δημιουργήθηκε δυναμικά, καθώς περιλαμβάνει το flag “1”, το οποίο δηλώνει πως δημιουργήθηκε από κάποιο πρωτόκολλο δρομολόγησης.

```

[root@router]# netstat -r
Routing tables

Internet:
Destination      Gateway          Flags   Refs      Use Netif Expire
localhost        link#3          UH        0    161  lo0
172.17.17.0/30   link#2          U         0      1  em1
172.17.17.1      link#2          UHS       0      0  lo0
192.168.1.0      link#1          U         0      5  em0
192.168.1.1      link#1          UHS       0      0  lo0
192.168.2.0      172.17.17.2    UG1       0      5  em1

```

Άσκηση 2: Λειτουργία του RIP

2.1

Εκτελούμε στο R1 “**tcpdump -vvvni em0**”.

2.2

Αφού αφήσουμε την καταγραφή να τρέχει για περίπου 2 λεπτά, βλέπουμε τα παρακάτω:

```
tcpdump: listening on em0, link-type EN10MB (Ethernet), capture size 65535 bytes
20:22:44.505592 IP (tos 0xc0, ttl 1, id 410, offset 0, flags [none], proto UDP (17), length 52)
    192.168.1.1.520 > 224.0.0.9.520: [bad udp cksum 0xa1e4 -> 0x58d9!]
        RIPv2, Request, length: 24
        0x0000: 0102 0000 0000 0000 0000 0000 0000 0000
        0x0010: 0000 0000 0000 0010
20:22:44.635004 IP (tos 0xc0, ttl 1, id 411, offset 0, flags [DF], proto IGMP (2), length 40, options [RA])
    192.168.1.1 > 224.0.0.22: igmp v3 report, 1 group record(s) [gaddr 224.0.0.9 to_ex { }]
20:22:44.835027 IP (tos 0xc0, ttl 1, id 412, offset 0, flags [DF], proto IGMP (2), length 40, options [RA])
    192.168.1.1 > 224.0.0.22: igmp v3 report, 1 group record(s) [gaddr 224.0.0.9 to_ex { }]
20:22:48.736048 IP (tos 0xc0, ttl 1, id 413, offset 0, flags [none], proto UDP (17), length 72)
    192.168.1.1.520 > 224.0.0.9.520: [bad udp cksum 0xa1f8 -> 0xd901!]
        RIPv2, Response, length: 44, routes: 2
        AFI IPv4, 172.17.17.0/30, tag 0x0000, metric: 1, next-hop: self
        AFI IPv4, 192.168.2.0/24, tag 0x0000, metric: 2, next-hop: self
        0x0000: 0202 0000 0002 0000 ac11 1100 ffff fffc
        0x0010: 0000 0000 0000 0001 0002 0000 c0a8 0200
        0x0020: ffff ff00 0000 0000 0000 0002
20:23:25.744671 IP (tos 0xc0, ttl 1, id 415, offset 0, flags [none], proto UDP (17), length 72)
    192.168.1.1.520 > 224.0.0.9.520: [bad udp cksum 0xa1f8 -> 0xd901!]
        RIPv2, Response, length: 44, routes: 2
        AFI IPv4, 172.17.17.0/30, tag 0x0000, metric: 1, next-hop: self
        AFI IPv4, 192.168.2.0/24, tag 0x0000, metric: 2, next-hop: self
        0x0000: 0202 0000 0002 0000 ac11 1100 ffff fffc
        0x0010: 0000 0000 0000 0001 0002 0000 c0a8 0200
        0x0020: ffff ff00 0000 0000 0000 0002
20:23:53.755039 IP (tos 0xc0, ttl 1, id 417, offset 0, flags [none], proto UDP (17), length 72)
    192.168.1.1.520 > 224.0.0.9.520: [bad udp cksum 0xa1f8 -> 0xd901!]
        RIPv2, Response, length: 44, routes: 2
        AFI IPv4, 172.17.17.0/30, tag 0x0000, metric: 1, next-hop: self
        AFI IPv4, 192.168.2.0/24, tag 0x0000, metric: 2, next-hop: self
        0x0000: 0202 0000 0002 0000 ac11 1100 ffff fffc
        0x0010: 0000 0000 0000 0001 0002 0000 c0a8 0200
        0x0020: ffff ff00 0000 0000 0000 0002
20:24:25.763920 IP (tos 0xc0, ttl 1, id 419, offset 0, flags [none], proto UDP (17), length 72)
    192.168.1.1.520 > 224.0.0.9.520: [bad udp cksum 0xa1f8 -> 0xd901!]
        RIPv2, Response, length: 44, routes: 2
        AFI IPv4, 172.17.17.0/30, tag 0x0000, metric: 1, next-hop: self
        AFI IPv4, 192.168.2.0/24, tag 0x0000, metric: 2, next-hop: self
        0x0000: 0202 0000 0002 0000 ac11 1100 ffff fffc
        0x0010: 0000 0000 0000 0001 0002 0000 c0a8 0200
        0x0020: ffff ff00 0000 0000 0000 0002
```

Παρατηρούμε ένα “RIPv2 Request” και στη συνέχεια μια ακολουθία από “RIPv2 Response”, ωστόσο, στο μεταξύ παρεμβάλλονται και κάποια “IGMPv3 Report” πακέτα.

2.3

Πηγή για τα μηνύματα και των 2 ειδών είναι η θύρα 520 της διεύθυνσης 192.168.1.1, ενώ προορισμό αποτελεί η θύρα 520 της διεύθυνσης 224.0.0.9., η οποία χρησιμοποιείται από το RIPv2 προκειμένου να σταλεί πληροφορία σχετικά με το routing του δικτύου σε όλους τους δρομολογητές του υποδικτύου που χρησιμοποιούν το συγκεκριμένο πρωτόκολλο (multicasting).

2.4

Όχι, δε καταγράψαμε RIP μηνύματα από τον R2.

2.5

Έχουν TTL = 1.

2.6

Βλέπουμε την τιμή “Proto UDP”, επομένως χρησιμοποιεί το πρωτόκολλο UDP και τη θύρα 520.

2.7

Στα παραπάνω RIP responses που παράγει ο R1 βλέπουμε πως διαφημίζονται 2 δίκτυα, το WAN1 (172.17.17.0/30) και το LAN2 (192.168.2.0/24), ενώ δεν υπάρχει διαφήμιση για το LAN1.

2.8

Βλέπουμε μηνύματα ανά διαστήματα που κυμαίνονται από περίπου 28 δευτερόλεπτα έως και περίπου 37, τιμές που ανήκουν στο διάστημα που αναφέραμε στο ερώτημα 1.22.

2.9

Εκτελούμε στο R1 “**tcpdump -vvvni em1**” και καταγράφουμε τα παρακάτω:

```
[root@R1]~# tcpdump -vvvni em1
tcpdump: listening on em1, link-type EN10MB (Ethernet), capture size 65535 bytes
21:02:49.847878 IP (tos 0xc0, ttl 1, id 656, offset 0, flags [none], proto UDP (17), length 52)
    172.17.17.1.520 > 224.0.0.9.520: [bad udp cksum 0x9d4d -> 0x5d70!]
        RIPv2, Request, length: 24
        0x0000: 0102 0000 0000 0000 0000 0000 0000 0000
        0x0010: 0000 0000 0000 0010
21:02:49.848129 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 172.17.17.1 tell 172.17.17.2, length 46
21:02:49.848143 ARP, Ethernet (len 6), IPv4 (len 4), Reply 172.17.17.1 is-at 08:00:27:24:fe:02, length 28
21:02:49.848224 IP (tos 0xc0, ttl 64, id 533, offset 0, flags [none], proto UDP (17), length 52)
    172.17.17.2.520 > 172.17.17.1.520: [udp sum ok]
        RIPv2, Response, length: 24, routes: 1
            AFI IPv4, 192.168.2.0/24, tag 0x0000, metric: 1, next-hop: self
            0x0000: 0202 0000 0002 0000 c0a8 0200 ffff ff00
            0x0010: 0000 0000 0000 0001
21:02:50.472151 IP (tos 0xc0, ttl 1, id 535, offset 0, flags [none], proto UDP (17), length 52)
    172.17.17.2.520 > 224.0.0.9.520: [udp sum ok]
        RIPv2, Response, length: 24, routes: 1
            AFI IPv4, 192.168.2.0/24, tag 0x0000, metric: 1, next-hop: self
            0x0000: 0202 0000 0002 0000 c0a8 0200 ffff ff00
            0x0010: 0000 0000 0000 0001
21:03:12.488927 IP (tos 0xc0, ttl 1, id 658, offset 0, flags [none], proto UDP (17), length 52)
    172.17.17.1.520 > 224.0.0.9.520: [bad udp cksum 0x9d4d -> 0x9bd3!]
        RIPv2, Response, length: 24, routes: 1
            AFI IPv4, 192.168.1.0/24, tag 0x0000, metric: 1, next-hop: self
            0x0000: 0202 0000 0002 0000 c0a8 0100 ffff ff00
            0x0010: 0000 0000 0000 0001
21:03:20.482618 IP (tos 0xc0, ttl 1, id 537, offset 0, flags [none], proto UDP (17), length 52)
    172.17.17.2.520 > 224.0.0.9.520: [udp sum ok]
        RIPv2, Response, length: 24, routes: 1
            AFI IPv4, 192.168.2.0/24, tag 0x0000, metric: 1, next-hop: self
            0x0000: 0202 0000 0002 0000 c0a8 0200 ffff ff00
            0x0010: 0000 0000 0000 0001
21:03:45.498362 IP (tos 0xc0, ttl 1, id 660, offset 0, flags [none], proto UDP (17), length 52)
    172.17.17.1.520 > 224.0.0.9.520: [bad udp cksum 0x9d4d -> 0x9bd3!]
        RIPv2, Response, length: 24, routes: 1
            AFI IPv4, 192.168.1.0/24, tag 0x0000, metric: 1, next-hop: self
            0x0000: 0202 0000 0002 0000 c0a8 0100 ffff ff00
            0x0010: 0000 0000 0000 0001
21:03:56.491551 IP (tos 0xc0, ttl 1, id 539, offset 0, flags [none], proto UDP (17), length 52)
    172.17.17.2.520 > 224.0.0.9.520: [udp sum ok]
        RIPv2, Response, length: 24, routes: 1
            AFI IPv4, 192.168.2.0/24, tag 0x0000, metric: 1, next-hop: self
            0x0000: 0202 0000 0002 0000 c0a8 0200 ffff ff00
            0x0010: 0000 0000 0000 0001
21:04:12.508606 IP (tos 0xc0, ttl 1, id 662, offset 0, flags [none], proto UDP (17), length 52)
    172.17.17.1.520 > 224.0.0.9.520: [bad udp cksum 0x9d4d -> 0x9bd3!]
        RIPv2, Response, length: 24, routes: 1
            AFI IPv4, 192.168.1.0/24, tag 0x0000, metric: 1, next-hop: self
            0x0000: 0202 0000 0002 0000 c0a8 0100 ffff ff00
            0x0010: 0000 0000 0000 0001
```

Βλέπουμε πως έχουν διαφημιστεί τα LAN1 (192.168.1.0/24) και LAN2 (192.168.2.0/24). Από τον R1 ειδικότερα έχει παραχθεί το αρχικό RIPv2 Request και στη συνέχεια κάποια RIPv2 Responses.

2.10

Βλέπουμε πως ο R1 διαφημίζει μέσω των Responses το LAN1 μόνο, επομένως δε διαφημίζεται το WAN1.

2.11

Αντίστοιχα, παρατηρήσαμε και μηνύματα RIP από τον R2, στα οποία διαφημίζεται το LAN2.

2.12

Παρατηρούμε πως όταν διαφημίζεται ένα δίκτυο, το RIPv2 response έχει μέγεθος 24 bytes, ενώ όταν διαφημίζονται 2 έχει μέγεθος 44 bytes. Επομένως, έχουμε τις εξισώσεις:

$$a + x = 24 \quad (1)$$

$$a + x + x = 44 \quad (2)$$

Από αυτές συνάγεται πως $a = 4$ και $x = 20$, όπου a είναι το RIP header, ενώ 20 bytes είναι το απαιτούμενο ανά εγγραφή μέγεθος. (Σημείωση: Το IP header έχει πεδίο Total Length με τιμή 52 bytes όταν έχουμε μία εγγραφή, το οποίο προκύπτει ως: 20 (IP header) + 8 (UDP header) + 24 (RIP))

2.13

Εκτελούμε στο R1 “**tcpdump -vvvni em0 udp port 520**”.

2.14

Από το R2, όντας σε Global Configuration Mode εκτελούμε “**router rip**” → “**no network 192.168.2.0/24**” και βλέπουμε πως στο LAN1 το δίκτυο 192.168.2.0/24 διαφημίζεται πλέον με κόστος 16, επομένως είναι πρακτικά unreachable.

```
21:17:34.787856 IP (tos 0xc0, ttl 1, id 747, offset 0, flags [none], proto UDP (17), length 72)
    192.168.1.1.520 > 224.0.0.9.520: [bad udp cksum 0xa1f8 -> 0xd901!]
        RIPv2, Response, length: 44, routes: 2
            AFI IPv4,    172.17.17.0/30, tag 0x0000, metric: 1, next-hop: self
            AFI IPv4,    192.168.2.0/24, tag 0x0000, metric: 2, next-hop: self
            0x0000: 0202 0000 0002 0000 ac11 1100 ffff fffc
            0x0010: 0000 0000 0000 0001 0002 0000 c0a8 0200
            0x0020: ffff ff00 0000 0000 0000 0002
21:17:36.840904 IP (tos 0xc0, ttl 1, id 750, offset 0, flags [none], proto UDP (17), length 52)
    192.168.1.1.520 > 224.0.0.9.520: [bad udp cksum 0xa1e4 -> 0x962d!]
        RIPv2, Response, length: 24, routes: 1
            AFI IPv4,    192.168.2.0/24, tag 0x0000, metric: 16, next-hop: self
            0x0000: 0202 0000 0002 0000 c0a8 0200 ffff ff00
            0x0010: 0000 0000 0000 0001
```

2.15

Επανεισάγουμε το LAN2 στο R2 και βλέπουμε πως πλέον διαφημίζεται η απόσταση με κόστος 2 αντί για 16.

```
192.168.1.1.520 > 224.0.0.9.520: [bad udp cksum 0xa1e4 -> 0x963b!]
    RIPv2, Response, length: 24, routes: 1
        AFI IPv4, 192.168.2.0/24, tag 0x0000, metric: 2, next-hop: self
        0x0000: 0202 0000 0002 0000 c0a8 0200 ffff ff00
        0x0010: 0000 0000 0000 0002
21:20:02.836180 IP (tos 0xc0, ttl 1, id 795, offset 0, flags [none], proto UDP (17), length 72)
    192.168.1.1.520 > 224.0.0.9.520: [bad udp cksum 0xa1f8 -> 0xd901!]
        RIPv2, Response, length: 44, routes: 2
            AFI IPv4, 172.17.17.0/30, tag 0x0000, metric: 1, next-hop: self
            AFI IPv4, 192.168.2.0/24, tag 0x0000, metric: 2, next-hop: self
            0x0000: 0202 0000 0002 0000 ac11 1100 ffff fffc
            0x0010: 0000 0000 0000 0001 0002 0000 c0a8 0200
            0x0020: ffff ff00 0000 0000 0000 0002
```

2.16

Εκτελούμε στο R2 “**tcpdump -vvvi em1 ‘udp port 520 and host 172.17.17.1’**”.

2.17

Ναι, η ενημέρωση για το κόστος της διαδρομής για το LAN1 παράχθηκε αμέσως

```
172.17.17.1.router > 224.0.0.9.router: [udp sum ok]
    RIPv2, Response, length: 24, routes: 1
        AFI IPv4, 192.168.1.0/24, tag 0x0000, metric: 1, next-hop: self
        0x0000: 0202 0000 0002 0000 c0a8 0100 ffff ff00
        0x0010: 0000 0000 0000 0001
21:25:22.477590 IP (tos 0xc0, ttl 1, id 832, offset 0, flags [none], proto UDP (17), length 52)
    172.17.17.1.router > 224.0.0.9.router: [udp sum ok]
        RIPv2, Response, length: 24, routes: 1
        AFI IPv4, 192.168.1.0/24, tag 0x0000, metric: 1, next-hop: self
        0x0000: 0202 0000 0002 0000 c0a8 0100 ffff ff00
        0x0010: 0000 0000 0000 0001
21:25:25.547254 IP (tos 0xc0, ttl 1, id 833, offset 0, flags [none], proto UDP (17), length 52)
    172.17.17.1.router > 224.0.0.9.router: [udp sum ok]
        RIPv2, Response, length: 24, routes: 1
        AFI IPv4, 192.168.1.0/24, tag 0x0000, metric: 16, next-hop: self
        0x0000: 0202 0000 0002 0000 c0a8 0100 ffff ff00
        0x0010: 0000 0000 0000 0010
```

2.18

Δε παράχθηκε RIP μήνυμα για τη διαγραφή του 192.168.1.0/24 στην καταγραφή που γίνεται στο LAN1, καθώς το R1 δεν ενημερώνεται από άλλους δρομολογητές, αφού είναι σε άμεση επαφή με αυτό.

2.19

Παρατηρούμε με “**show ip route**” πως διαγράφηκε άμεσα από τον πίνακα δρομολόγησης.

```
R1# 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>* 192.168.1.0/24 is directly connected, em0
```

2.20

Παρατηρούμε με “**show ip rip**” πως η εγγραφή για το LAN2 δε διαγράφηκε αμέσως, ωστόσο το κόστος έγινε 16. Μετά από περίπου 2 λεπτά παρατηρούμε πως η εγγραφή έχει διαγραφεί, καθώς αυτός είναι ο default χρόνος που απαιτείται για το garbage collection.

```
R1# show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

      Network          Next Hop          Metric From           Tag Time
C(i) 172.17.17.0/30    0.0.0.0            1 self             0
R(n) 192.168.2.0/24    172.17.17.2        16 172.17.17.2     0 00:33
```

2.21

Εκτελούμε αφού έχουμε μπει στο configuration του RIP “**network 192.168.1.0/24**” και “**network 192.168.2.0/24**” στα R1 και R2 αντίστοιχα.

2.22

Πρέπει να κάνουμε τις διεπαφές των R1 και R2 που είναι στα LAN1 και LAN2 αντίστοιχα να μην αποστέλλουν ενημερώσεις RIP, επομένως εκτελούμε “**passive-interface em0**” στο R1 και “**passive-interface em0**” στο R2.

2.23

Παρατηρούμε πως στο LAN1 στέλνεται αρχικά ένα RIP Request στην multicast διεύθυνση, χωρίς ωστόσο κάποια απόκριση.

Άσκηση 3: Εναλλακτικές διαδρομές

3.1

Εκτελούμε στο R1 “cli” → “configure terminal” → “interface em2” → “ip address 172.17.17.5/30” → “exit” → “router rip” → “network 172.17.17.4/30”.

3.2

Εκτελούμε στο R2 “cli” → “configure terminal” → “interface em2” → “ip address 172.17.17.9/30” → “exit” → “router rip” → “network 172.17.17.8/30”.

3.3

Εκτελούμε στο R3 “cli” → “configure terminal” → “hostname R3” → “interface em0” → “ip address 172.17.17.6/30” → “exit” → “router rip” → “network 172.17.17.4/30” → “exit” → “interface em1” → “ip address 172.17.17.10/30” → “exit” → “router rip” → “network 172.17.17.8/30”.

3.4

Ο R1 έχει μάθει μέσω του RIP τα LAN2 και WAN3.

```
R1(config-router)# do show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
    (n) - normal, (s) - static, (d) - default, (r) - redistribute,
    (i) - interface

      Network          Next Hop           Metric From        Tag Time
C(i) 172.17.17.0/30   0.0.0.0            1 self           0
C(i) 172.17.17.4/30   0.0.0.0            1 self           0
R(n) 172.17.17.8/30   172.17.17.2         2 172.17.17.2   0 02:39
C(i) 192.168.1.0/24   0.0.0.0            1 self           0
R(n) 192.168.2.0/24   172.17.17.2         2 172.17.17.2   0 02:39
```

3.5

Ο R2 έχει μάθει μέσω του RIP τα LAN1 και WAN2.

```
R2(config-router)# do show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
    (n) - normal, (s) - static, (d) - default, (r) - redistribute,
    (i) - interface

      Network          Next Hop           Metric From        Tag Time
C(i) 172.17.17.0/30   0.0.0.0            1 self           0
R(n) 172.17.17.4/30   172.17.17.1         2 172.17.17.1   0 02:45
C(i) 172.17.17.8/30   0.0.0.0            1 self           0
R(n) 192.168.1.0/24   172.17.17.1         2 172.17.17.1   0 02:45
C(i) 192.168.2.0/24   0.0.0.0            1 self           0
```

3.6

Ο R3 έχει μάθει μέσω του RIP τα LAN1, LAN2 και WAN1.

R3(config)# do show ip rip						
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP						
Sub-codes:						
(n) - normal, (s) - static, (d) - default, (r) - redistribute, (i) - interface						
Network	Next Hop	Metric	From	Tag	Time	
R(n) 172.17.17.0/30	172.17.17.5	2	172.17.17.5	0	02:35	
C(i) 172.17.17.4/30	0.0.0.0	1	self	0		
C(i) 172.17.17.8/30	0.0.0.0	1	self	0		
R(n) 192.168.1.0/24	172.17.17.5	2	172.17.17.5	0	02:35	
R(n) 192.168.2.0/24	172.17.17.9	2	172.17.17.9	0	02:50	

3.7

Ναι, 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=62 time=0.608 ms

--- 192.168.2.2 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.608/0.608/0.608/0.000 ms
```

3.8

Όντας σε Global Configuration Mode στο R3 εκτελούμε “**interface em2**” → “**ip address 192.168.3.1/24**”.

3.9

Όχι, δε παρατηρούμε καμία μεταβολή στις εγγραφές.

3.10

Εκτελούμε στο R3 από Global Configuration Mode “**router rip**” → “**network 192.168.3.0/24**”.

3.11

Ναι, αυτή τη φορά έχουν αλλάξει οι εγγραφές στους R1 και R2 αντίστοιχα:

```
R1(config-router)# do show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
    (n) - normal, (s) - static, (d) - default, (r) - redistribute,
    (i) - interface

      Network          Next Hop          Metric From        Tag Time
C(i) 172.17.17.0/30  0.0.0.0           1 self            0
C(i) 172.17.17.4/30  0.0.0.0           1 self            0
R(n) 172.17.17.8/30  172.17.17.2       2 172.17.17.2   0 02:58
C(i) 192.168.1.0/24  0.0.0.0           1 self            0
R(n) 192.168.2.0/24  172.17.17.2       2 172.17.17.2   0 02:58
R(n) 192.168.3.0/24  172.17.17.6       2 172.17.17.6   0 02:59
```

```
R2(config-router)# do show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
    (n) - normal, (s) - static, (d) - default, (r) - redistribute,
    (i) - interface

      Network          Next Hop          Metric From        Tag Time
C(i) 172.17.17.0/30  0.0.0.0           1 self            0
R(n) 172.17.17.4/30  172.17.17.1       2 172.17.17.1   0 02:30
C(i) 172.17.17.8/30  0.0.0.0           1 self            0
R(n) 192.168.1.0/24  172.17.17.1       2 172.17.17.1   0 02:30
C(i) 192.168.2.0/24  0.0.0.0           1 self            0
R(n) 192.168.3.0/24  172.17.17.10      2 172.17.17.10  0 02:56
```

3.12

Ναι, οι πίνακες άλλαξαν άμεσα.

3.13

Εκτελούμε στο R3 “**no network 172.17.17.4/30**” → “**no network 172.17.17.8/30**” → “**no network 192.168.3.0/24**” → “**network 0.0.0.0/0**”. Το δίκτυο 0.0.0.0/0 ουσιαστικά περιλαμβάνει όλες τις IPv4 διευθύνσεις, άρα εν προκειμένω ενεργοποιήσαμε το RIP σε κάθε διεπαφή του R3.

3.14

Εκτελούμε στο R3 “**do show ip rip status**”:

```
R3(config-router)# do show ip rip status
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 10 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Relaxed receiving size checks are off
  Redistributing:
    Default version control: send version 2, receive any version
      Interface      Send   Recv   Key-chain
        em0          2      1 2
        em1          2      1 2
        em2          2      1 2
        lo0          2      1 2
  Routing for Networks:
    0.0.0.0/0
  Routing Information Sources:
    Gateway          BadPackets  BadRoutes  Distance  Last Update
      172.17.17.5           0          0       120  00:00:02
      172.17.17.9           0          0       120  00:00:07
  Distance: (default is 120)
```

Βλέπουμε πως το RIP είναι ενεργοποιημένο σε όλες τις διεπαφές του R3, em0, em1, em2 και lo0. Στη δρομολόγηση βλέπουμε πως μετέχει το δίκτυο 0.0.0.0/0, άρα πρακτικά όλα.

3.15

Οι δυναμικές εγγραφές δεν έχουν αλλάξει (“**do show ip route rip**”).

3.16

Εκτελώντας στο R3 “**tcpdump -vvvi em0**” βλέπουμε πως διαφημίζει τα WAN3, LAN2, LAN3.

```
19:34:27.097226 IP (tos 0xc0, ttl 1, id 442, offset 0, flags [none], proto UDP (17), length 92)
  172.17.17.6.router > 224.0.0.9.router: [bad udp cksum 0x9d7a -> 0x1ab7?]
    RIPv2, Response, length: 64, routes: 3
      AFI IPv4,      172.17.17.8/30, tag 0x0000, metric: 1, next-hop: self
      AFI IPv4,      192.168.2.0/24, tag 0x0000, metric: 2, next-hop: self
      AFI IPv4,      192.168.3.0/24, tag 0x0000, metric: 1, next-hop: self
      0x0000: 0202 0000 0002 0000 ac11 1108 ffff fffc
      0x0010: 0000 0000 0001 0001 0002 0000 c0a8 0200
      0x0020: ffff ff00 0000 0000 0000 0002 0000 0000
      0x0030: c0a8 0300 ffff ff00 0000 0000 0000 0001
```

3.17

Όπως είδαμε παραπάνω, το R3 δε διαφημίζει το 192.168.1.0/24. Αυτό συμβαίνει λόγω του μηχανισμού αποφυγής βρόχων “διαιτρεμένου ορίζοντα (split horizon)”, ο οποίος αποτρέπει τον δρομολογητή από το να διαφημίσει μια διαδρομή στη διεπαφή από όπου την έμαθε (εν προκειμένω ο R3 έμαθε για το 192.168.1.0/24 από το R1).

3.18

Συμπεραίνουμε πως με την εισαγωγή του δικτύου 0.0.0.0/0 στην RIP δρομολόγηση, ο δρομολογητής ενημερώνει για κάθε υποδίκτυο που γνωρίζει.

3.19

Παρατηρούμε (“tcpdump -vvvi em1” και “tcpdump -vvvi em2”) από τους παρακάτω πίνακες πως το διαφημιζόμενο κόστος (σε hops) προς το WAN3 (172.17.17.8/30) είναι 1 από το R2 αλλά και από το R3.

```
19:56:07.584721 IP (tos 0xc0, ttl 1, id 608, offset 0, flags [none], proto UDP (17), length 92)
    172.17.17.2.router > 224.0.0.9.router: [bad udp cksum 0x9d76 -> 0x1abb!]
        RIPv2, Response, length: 64, routes: 3
            AFI IPv4,    172.17.17.8/30, tag 0x0000, metric: 1, next-hop: self
            AFI IPv4,    192.168.2.0/24, tag 0x0000, metric: 1, next-hop: self
            AFI IPv4,    192.168.3.0/24, tag 0x0000, metric: 2, next-hop: self
            0x0000: 0202 0000 0002 0000 ac11 1108 ffff fffc
            0x0010: 0000 0000 0000 0001 0002 0000 c0a8 0200
            0x0020: ffff ff00 0000 0000 0000 0001 0002 0000
            0x0030: c0a8 0300 ffff ff00 0000 0000 0000 0002
```

```
19:55:18.518340 IP (tos 0xc0, ttl 1, id 752, offset 0, flags [none], proto UDP (17), length 92)
    172.17.17.6.router > 224.0.0.9.router: [bad udp cksum 0x9d7a -> 0x1ab7!]
        RIPv2, Response, length: 64, routes: 3
            AFI IPv4,    172.17.17.8/30, tag 0x0000, metric: 1, next-hop: self
            AFI IPv4,    192.168.2.0/24, tag 0x0000, metric: 2, next-hop: self
            AFI IPv4,    192.168.3.0/24, tag 0x0000, metric: 1, next-hop: self
            0x0000: 0202 0000 0002 0000 ac11 1108 ffff fffc
            0x0010: 0000 0000 0000 0001 0002 0000 c0a8 0200
            0x0020: ffff ff00 0000 0000 0000 0002 0002 0000
            0x0030: c0a8 0300 ffff ff00 0000 0000 0000 0001
```

Ο R1 έχει επιλέξει όπως φαίνεται τη διαδρομή μέσω του R2:

```
R1(config)# do show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
    (n) - normal, (s) - static, (d) - default, (r) - redistribute,
    (i) - interface

      Network          Next Hop           Metric From      Tag Time
C(i) 172.17.17.0/30    0.0.0.0             1 self          0
C(i) 172.17.17.4/30    0.0.0.0             1 self          0
R(n) 172.17.17.8/30    172.17.17.2         2 172.17.17.2  0 02:51
C(i) 192.168.1.0/24    0.0.0.0             1 self          0
R(n) 192.168.2.0/24    172.17.17.2         2 172.17.17.2  0 02:51
R(n) 192.168.3.0/24    172.17.17.6         2 172.17.17.6  0 02:48
```

3.20

Καταγράφοντας την κίνηση στο WAN1 και στο WAN2 βλέπουμε πως το 172.17.17.8/30 διαφημίζεται από το R1 στο WAN2.

```

20:09:55.424725 IP (tos 0xc0, ttl 1, id 713, offset 0, flags [none], proto UDP (17), length 92)
    172.17.17.1.router > 224.0.0.9.router: [bad udp cksum 0x9d75 -> 0x1bc0!]
        RIPv2, Response, length: 64, routes: 3
            AFI IPv4,      172.17.17.4/30, tag 0x0000, metric: 1, next-hop: self
            AFI IPv4,      192.168.1.0/24, tag 0x0000, metric: 1, next-hop: self
            AFI IPv4,      192.168.3.0/24, tag 0x0000, metric: 2, next-hop: self
            0x0000: 0202 0000 0002 0000 ac11 1104 ffff fffc
            0x0010: 0000 0000 0000 0001 0002 0000 c0a8 0100
            0x0020: ffff ff00 0000 0000 0000 0001 0002 0000
            0x0030: c0a8 0300 ffff ff00 0000 0000 0000 0002

```



```

20:06:10.363343 IP (tos 0xc0, ttl 1, id 642, offset 0, flags [none], proto UDP (17), length 112)
    172.17.17.5.router > 224.0.0.9.router: [bad udp cksum 0x9d8d -> 0x5f7d!]
        RIPv2, Response, length: 84, routes: 4
            AFI IPv4,      172.17.17.0/30, tag 0x0000, metric: 1, next-hop: self
            AFI IPv4,      172.17.17.8/30, tag 0x0000, metric: 2, next-hop: self
            AFI IPv4,      192.168.1.0/24, tag 0x0000, metric: 1, next-hop: self
            AFI IPv4,      192.168.2.0/24, tag 0x0000, metric: 2, next-hop: self
            0x0000: 0202 0000 0002 0000 ac11 1100 ffff fffc
            0x0010: 0000 0000 0000 0001 0002 0000 ac11 1108
            0x0020: ffff fffc 0000 0000 0000 0002 0002 0000
            0x0030: c0a8 0100 ffff ff00 0000 0000 0000 0001
            0x0040: 0002 0000 c0a8 0200 ffff ff00 0000 0000
            0x0050: 0000 0002

```

Είδαμε πως ο R1 έχει επιλέξει τη διαδρομή μέσω του R2. Ωστόσο, η μετρική που λαμβάνει από το R3 έχει επίσης ίδια τιμή με αυτή από το R2. Σε αυτή την περίπτωση, προστίθεται στον πίνακα η νέα εγγραφή (πιο πρόσφατη με μετρική ίδια της υπάρχουσας) σε περίπτωση που το timeout της ήδη υπάρχουσας εγγραφής έχει παρέλθει τουλάχιστον κατά το ήμισυ του χρόνου λήξης.

Άσκηση 4: Αλλαγές στην τοπολογία, σφάλμα καλωδίου και RIP

4.1

Εκτελούμε στο PC3 “**vtysh**” → “**configure terminal**” → “**hostname PC3**” → “**interface em0**” → “**ip address 192.168.3.2/24**” → “**exit**” → “**ip route 0.0.0.0/0 192.168.3.1**”.

4.2

Ναι.

4.3

Ναι.

4.4

Ναι.

4.5

Παρουσιάζονται κατά σειρά οι πίνακες δρομολόγησης των R1, R2, R3:

```
R1(config)# do show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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
R>* 172.17.17.8/30 [120/2] via 172.17.17.2, em1, 01:18:48
C>* 192.168.1.0/24 is directly connected, em0
R>* 192.168.2.0/24 [120/2] via 172.17.17.2, em1, 01:21:20
R>* 192.168.3.0/24 [120/2] via 172.17.17.6, em2, 00:57:21
```

```
R2(config)# do show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - 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
R>* 172.17.17.4/30 [120/2] via 172.17.17.1, em1, 01:20:13
C>* 172.17.17.8/30 is directly connected, em2
R>* 192.168.1.0/24 [120/2] via 172.17.17.1, em1, 01:20:13
C>* 192.168.2.0/24 is directly connected, em0
R>* 192.168.3.0/24 [120/2] via 172.17.17.10, em2, 00:55:53
```

```
R3(config)# do show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       0 - OSPF, I - IS-IS, B - BGP, A - Babel,
       > - selected route, * - FIB route

C>* 127.0.0.0/8 is directly connected, lo0
R>* 172.17.17.0/30 [120/2] via 172.17.17.5, em0, 00:56:19
C>* 172.17.17.4/30 is directly connected, em0
C>* 172.17.17.8/30 is directly connected, em1
R>* 192.168.1.0/24 [120/2] via 172.17.17.5, em0, 00:56:19
R>* 192.168.2.0/24 [120/2] via 172.17.17.9, em1, 00:56:19
C>* 192.168.3.0/24 is directly connected, em2
```

4.6

Εκτελούμε “link-detect” στις διεπαφές:

- em1 και em2 του R1
- em1 και em2 του R2
- em0 και em1 του R3

4.7

Απενεργοποιούμε τα κατάλληλα καλώδια του WAN1 και καταγράφουμε εκ νέου τους πίνακες δρομολόγησης των R1, R2, R3 κατά σειρά παρακάτω. Παρατηρούμε τις εξής αλλαγές:

- Στον R1 έχει διαγραφεί η εγγραφή για το WAN1 (172.17.17.0/30) μέσω em1 και η δρομολόγηση για το WAN3 (172.17.17.8/30) και το LAN2 (192.168.2.0/24) γίνεται πλέον μέσω της 172.17.17.6 (em0 του R3) αντί της 172.17.17.2 (em1 του R2).
- Στον R2 έχει διαγραφεί η εγγραφή για το WAN1 μέσω em1 και η δρομολόγηση για το WAN2 (172.17.17.4/30) και το LAN1 (192.168.1.0/24) γίνεται πλέον μέσω της 172.17.17.10 (em1 του R3) αντί της 172.17.17.1 (em1 του R1)
- Στον R3 έχει διαγραφεί η εγγραφή για το WAN1 μέσω της em0.

```
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>* 172.17.4/30 is directly connected, em2
R>* 172.17.17.8/30 [120/2] via 172.17.17.6, em2, 00:01:37
C>* 192.168.1.0/24 is directly connected, em0
R>* 192.168.2.0/24 [120/3] via 172.17.17.6, em2, 00:01:37
R>* 192.168.3.0/24 [120/2] via 172.17.17.6, em2, 01:04:30
```

```
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
R>* 172.17.4/30 [120/2] via 172.17.17.10, em2, 00:01:42
C>* 172.17.17.8/30 is directly connected, em2
R>* 192.168.1.0/24 [120/3] via 172.17.17.10, em2, 00:01:42
C>* 192.168.2.0/24 is directly connected, em0
R>* 192.168.3.0/24 [120/2] via 172.17.17.10, em2, 01:04:35
```

```
R3(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, em0
C>* 172.17.17.8/30 is directly connected, em1
R>* 192.168.1.0/24 [120/2] via 172.17.17.5, em0, 01:04:59
R>* 192.168.2.0/24 [120/2] via 172.17.17.9, em1, 01:04:59
C>* 192.168.3.0/24 is directly connected, em2
```

4.8

Ναι, συνεχίζουν να επικοινωνούν κανονικά.

4.9

Αφού επαναφέρουμε το WAN1, απενεργοποιούμε το WAN2 και βλέποντας τους νέους πίνακες δρομολόγησης, παρατηρούμε πως πλέον δεν υπάρχουν οι εγγραφές για το WAN2 και επιπλέον όσες δρομολογήσεις γίνονταν άμεσα μεταξύ των R1, R3 γίνονται πλέον μέσω του 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
R>* 172.17.17.8/30 [120/2] via 172.17.17.2, em1, 00:02:01
C>* 192.168.1.0/24 is directly connected, em0
R>* 192.168.2.0/24 [120/2] via 172.17.17.2, em1, 00:02:01
R>* 192.168.3.0/24 [120/3] via 172.17.17.2, em1, 00:02:01
```

```
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, em1
C>* 172.17.17.8/30 is directly connected, em2
R>* 192.168.1.0/24 [120/2] via 172.17.17.1, em1, 00:02:05
C>* 192.168.2.0/24 is directly connected, em0
R>* 192.168.3.0/24 [120/2] via 172.17.17.10, em2, 01:11:41
```

```
R3(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
R>* 172.17.17.0/30 [120/2] via 172.17.17.9, em1, 00:02:08
C>* 172.17.17.8/30 is directly connected, em1
R>* 192.168.1.0/24 [120/3] via 172.17.17.9, em1, 00:01:34
R>* 192.168.2.0/24 [120/2] via 172.17.17.9, em1, 01:12:02
C>* 192.168.3.0/24 is directly connected, em2
```

4.10

Ναι, επικοινωνούν κανονικά.

4.11

Αφού επαναφέρουμε το WAN2, απενεργοποιούμε το WAN3 και βλέποντας τους νέους πίνακες δρομολόγησης, παρατηρούμε πως πλέον δεν υπάρχουν οι εγγραφές για το WAN3 και επιπλέον όσες δρομολογήσεις γινόταν άμεσα μεταξύ των R2, R3 γίνονται πλέον μέσω του R1.

```
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
R>* 192.168.2.0/24 [120/2] via 172.17.17.2, em1, 00:04:32
R>* 192.168.3.0/24 [120/2] via 172.17.17.6, em2, 00:00:31
```

```
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, em1
R>* 172.17.17.4/30 [120/2] via 172.17.17.1, em1, 00:01:03
R>* 192.168.1.0/24 [120/2] via 172.17.17.1, em1, 00:04:37
C>* 192.168.2.0/24 is directly connected, em0
R>* 192.168.3.0/24 [120/3] via 172.17.17.1, em1, 00:00:34
```

```
R3(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
R>* 172.17.17.0/30 [120/2] via 172.17.17.5, em0, 00:00:54
C>* 172.17.17.4/30 is directly connected, em0
R>* 192.168.1.0/24 [120/2] via 172.17.17.5, em0, 00:00:54
R>* 192.168.2.0/24 [120/3] via 172.17.17.5, em0, 00:00:54
C>* 192.168.3.0/24 is directly connected, em2
```

4.12

Ναι, επικοινωνούν κανονικά.

4.13

Το τελευταίο πετυχημένο Ping είχε icmp_seq = 16, ενώ το πρώτο νέο πετυχημένο έχει icmp_seq = 42, που σημαίνει πως μεσολάβησαν 26 αποτυχημένα ping και δεδομένου πως το καθένα διαρκεί περίπου 0.8 ms, πέρασαν περίπου 20 δευτερόλεπτα.

```
64 bytes from 192.168.2.2: icmp_seq=14 ttl=62 time=0.672 ms
64 bytes from 192.168.2.2: icmp_seq=15 ttl=62 time=0.676 ms
64 bytes from 192.168.2.2: icmp_seq=16 ttl=62 time=0.629 ms
64 bytes from 192.168.2.2: icmp_seq=42 ttl=61 time=0.738 ms
64 bytes from 192.168.2.2: icmp_seq=43 ttl=61 time=0.809 ms
```

4.14

Παρατηρούμε ότι προτού αποκατασταθεί η παλιά διαδρομή το TTL είχε τιμή 61 (PC2 → R2 → R3 → R1 → PC1), ενώ αφού εγκαταστάθηκε η νέα διαδρομή, το TTL έγινε 62 (PC2 → R2 → R1 → PC1).

4.15

Η μετρική για τα 172.17.17.0/30 και 192.168.2.0/24 είναι 1 και 2 αντίστοιχα.

```
R1(config)# do show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface

      Network          Next Hop           Metric From        Tag Time
C(i) 172.17.17.0/30    0.0.0.0            1 self           0
C(i) 172.17.17.4/30    0.0.0.0            1 self           0
R(n) 172.17.17.8/30    172.17.17.6         2 172.17.17.6   0 02:53
C(i) 192.168.1.0/24    0.0.0.0            1 self           0
R(n) 192.168.2.0/24    172.17.17.2         2 172.17.17.2   0 02:48
R(n) 192.168.3.0/24    172.17.17.6         2 172.17.17.6   0 02:53
```

4.16

Ο χρόνος που εμφανίζεται με “**do show ip rip**” δηλώνει τον χρόνο που απομένει μέχρι η διαδρομή να παύσει να ισχύει (timeout – 180 sec).

4.17

Αποσυνδέουμε το WAN1 και βλέπουμε τα παρακάτω:

```
R1(config)# do show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
    (n) - normal, (s) - static, (d) - default, (r) - redistribute,
    (i) - interface

      Network          Next Hop          Metric From        Tag Time
R(n) 172.17.17.0/30    172.17.17.6      16 172.17.17.6    0 01:57
C(i) 172.17.17.4/30   0.0.0.0           1 self            0
R(n) 172.17.17.8/30    172.17.17.6      2 172.17.17.6    0 02:30
C(i) 192.168.1.0/24   0.0.0.0           1 self            0
R(n) 192.168.2.0/24   172.17.17.2      16 172.17.17.2    0 01:48
R(n) 192.168.3.0/24   172.17.17.6      2 172.17.17.6    0 02:30
```

Πλέον η απόσταση για το 172.17.17.0/30 έγινε 16 (unreachable) και η διάρκεια ζωής της εν λόγω εγγραφής ανήλθε στα 2 λεπτά, που είναι ο προκαθορισμένος χρόνος για το garbage collection. Αντίστοιχα για το 192.168.2.0/24 η απόσταση έγινε επίσης 16 και ο χρόνος ανήλθε επίσης στα 2 λεπτά.

4.18

Λίγο πιο μετά, βλέπουμε τον παρακάτω πίνακα δρομολογήσεων, ο οποίος μας δείχνει ότι η απόσταση προς το 192.168.2.0/24 έγινε 3 και είναι reachable από το R1 μέσω του R3.

```
R1(config)# do show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
    (n) - normal, (s) - static, (d) - default, (r) - redistribute,
    (i) - interface

      Network          Next Hop          Metric From        Tag Time
R(n) 172.17.17.0/30    172.17.17.6      16 172.17.17.6    0 01:50
C(i) 172.17.17.4/30   0.0.0.0           1 self            0
R(n) 172.17.17.8/30    172.17.17.6      2 172.17.17.6    0 02:58
C(i) 192.168.1.0/24   0.0.0.0           1 self            0
R(n) 192.168.2.0/24   172.17.17.6      3 172.17.17.6    0 02:58
R(n) 192.168.3.0/24   172.17.17.6      2 172.17.17.6    0 02:58
```

4.19

Η εγγραφή για το 172.17.17.0/30 έχει διαγραφεί εντελώς.

```
R1(config)# do show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
    (n) - normal, (s) - static, (d) - default, (r) - redistribute,
    (i) - interface

      Network          Next Hop        Metric From           Tag Time
C(i) 172.17.17.4/30   0.0.0.0          1 self            0
R(n) 172.17.17.8/30   172.17.17.6       2 172.17.17.6    0 02:48
C(i) 192.168.1.0/24   0.0.0.0          1 self            0
R(n) 192.168.2.0/24   172.17.17.6       3 172.17.17.6    0 02:48
R(n) 192.168.3.0/24   172.17.17.6       2 172.17.17.6    0 02:48
```

4.20

Όπως ήδη αναφέραμε, ο χρόνος εκείνος ήταν ο χρόνος για το garbage collection, μετά το πέρας του οποίου διαγράφονται οι μη ισχύουσες εγγραφές, όπως και έγινε για το 172.17.17.0/30, ενώ η εγγραφή για το 192.168.2.0/24 ενημερώθηκε μέσω του R3 οπότε και δε διαγράφηκε.

4.21

Εκτελούμε σε 2 διαφορετικές κονσόλες στο R1 “**tcpdump -vvvi em1**” και “**tcpdump -vvvi em2**” και καταγράφουμε τα παρακάτω αντίστοιχα:

```
21:09:59.905531 IP (tos 0xc0, ttl 1, id 1269, offset 0, flags [none], proto UDP
(17), length 112)
    172.17.17.1.router > 224.0.0.9.router: [bad udp cksum 0x9d89 -> 0x5e7d!]
        RIPv2, Response, length: 84, routes: 4
            AFI IPv4,      172.17.17.4/30, tag 0x0000, metric: 1, next-hop: self
            AFI IPv4,      172.17.17.8/30, tag 0x0000, metric: 2, next-hop: self
            AFI IPv4,      192.168.1.0/24, tag 0x0000, metric: 1, next-hop: self
            AFI IPv4,      192.168.3.0/24, tag 0x0000, metric: 2, next-hop: self
0x0000: 0202 0000 0002 0000 ac11 1104 ffff ffcc
0x0010: 0000 0000 0000 0001 0002 0000 ac11 1108
0x0020: ffff ffcc 0000 0000 0000 0002 0002 0000
0x0030: c0a8 0100 ffff ff00 0000 0000 0000 0001
0x0040: 0002 0000 c0a8 0300 ffff ff00 0000 0000
0x0050: 0000 0002
```

```
21:10:30.915315 IP (tos 0xc0, ttl 1, id 1332, offset 0, flags [none], proto UDP
(17), length 92)
    172.17.17.5.router > 224.0.0.9.router: [bad udp cksum 0x9d79 -> 0x1cc0!]
        RIPv2, Response, length: 64, routes: 3
            AFI IPv4,      172.17.17.0/30, tag 0x0000, metric: 1, next-hop: self
            AFI IPv4,      192.168.1.0/24, tag 0x0000, metric: 1, next-hop: self
            AFI IPv4,      192.168.2.0/24, tag 0x0000, metric: 2, next-hop: self
0x0000: 0202 0000 0002 0000 ac11 1100 ffff ffcc
0x0010: 0000 0000 0000 0001 0002 0000 c0a8 0100
0x0020: ffff ff00 0000 0000 0000 0001 0002 0000
0x0030: c0a8 0200 ffff ff00 0000 0000 0000 0002
```

Όπως βλέπουμε, το R1 διαφημίζει το 172.17.17.8/30 στο WAN1, διότι μαθαίνει για το δίκτυο αυτό από τη διεύθυνση 172.17.17.6 (R3) που είναι στο WAN2, επομένως λόγω διαιρεμένου ορίζοντα για αποφυγή βρόχων, δε διαφημίζεται ξανά στο WAN2, αφού από εκεί το έχει μάθει

Άσκηση 5: Τοπολογία με πολλαπλές WAN διασυνδέσεις

5.1

Εκτελούμε σε όλους τους δρομολογητές όντας σε Global Configuration Mode “**router rip**” → “**version 2**” → “**network 0.0.0.0/0**”.

5.2

Ο πίνακας δρομολόγησης του R1 περιέχει 7 δυναμικές εγγραφές.

```
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

R>* 10.0.0.0/30 [120/2] via 10.0.1.6, em2, 00:05:35
C>* 10.0.1.0/30 is directly connected, em1
C>* 10.0.1.4/30 is directly connected, em2
R>* 10.0.2.0/30 [120/2] via 10.0.1.2, em1, 00:05:11
R>* 10.0.2.4/30 [120/2] via 10.0.1.6, em2, 00:05:35
C>* 127.0.0.0/8 is directly connected, lo0
C>* 172.22.1.1/32 is directly connected, lo0
R>* 172.22.1.2/32 [120/2] via 10.0.1.2, em1, 00:05:11
R>* 172.22.2.1/32 [120/2] via 10.0.1.6, em2, 00:05:35
R>* 172.22.2.2/32 [120/2] via 10.0.1.6, em2, 00:01:05
C>* 192.168.1.0/24 is directly connected, em0
R>* 192.168.2.0/24 [120/3] via 10.0.1.6, em2, 00:05:45
```

5.3

Ο πίνακας δρομολόγησης του R2 περιέχει 7 δυναμικές εγγραφές.

```
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

R>* 10.0.0.0/30 [120/2] via 10.0.2.6, em2, 00:06:22
R>* 10.0.1.0/30 [120/2] via 10.0.2.2, em1, 00:05:58
R>* 10.0.1.4/30 [120/2] via 10.0.2.6, em2, 00:06:22
C>* 10.0.2.0/30 is directly connected, em1
C>* 10.0.2.4/30 is directly connected, em2
C>* 127.0.0.0/8 is directly connected, lo0
R>* 172.22.1.1/32 [120/3] via 10.0.2.6, em2, 00:06:32
R>* 172.22.1.2/32 [120/2] via 10.0.2.2, em1, 00:05:58
C>* 172.22.2.1/32 is directly connected, lo0
R>* 172.22.2.2/32 [120/2] via 10.0.2.6, em2, 00:01:52
R>* 192.168.1.0/24 [120/3] via 10.0.2.6, em2, 00:06:32
C>* 192.168.2.0/24 is directly connected, em0
```

5.4

Ο πίνακας δρομολόγησης του C1 περιέχει 7 δυναμικές εγγραφές.

```
C1(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>* 10.0.0.0/30 is directly connected, em2
C>* 10.0.1.0/30 is directly connected, em0
R>* 10.0.1.4/30 [120/2] via 10.0.1.1, em0, 00:07:29
C>* 10.0.2.0/30 is directly connected, em1
R>* 10.0.2.4/30 [120/2] via 10.0.2.1, em1, 00:07:29
C>* 127.0.0.0/8 is directly connected, lo0
R>* 172.22.1.1/32 [120/2] via 10.0.1.1, em0, 00:07:29
C>* 172.22.1.2/32 is directly connected, lo0
R>* 172.22.2.1/32 [120/2] via 10.0.2.1, em1, 00:07:29
R>* 172.22.2.2/32 [120/2] via 10.0.0.2, em2, 00:03:08
R>* 192.168.1.0/24 [120/2] via 10.0.1.1, em0, 00:07:29
R>* 192.168.2.0/24 [120/2] via 10.0.2.1, em1, 00:07:29
```

5.5

Ο πίνακας δρομολόγησης του C2 περιέχει 7 δυναμικές εγγραφές.

```

C2(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>* 10.0.0.0/30 is directly connected, em2
R>* 10.0.1.0/30 [120/2] via 10.0.1.5, em0, 00:08:01
C>* 10.0.1.4/30 is directly connected, em0
R>* 10.0.2.0/30 [120/2] via 10.0.2.5, em1, 00:08:01
C>* 10.0.2.4/30 is directly connected, em1
C>* 127.0.0.0/8 is directly connected, lo0
R>* 172.22.1.1/32 [120/2] via 10.0.1.5, em0, 00:08:01
R>* 172.22.1.2/32 [120/2] via 10.0.0.1, em2, 00:07:24
C>* 172.22.2.1/32 is directly connected, lo0
C>* 172.22.2.2/32 is directly connected, lo0
R>* 192.168.1.0/24 [120/2] via 10.0.1.5, em0, 00:08:01
R>* 192.168.2.0/24 [120/2] via 10.0.2.5, em1, 00:08:01

```

5.6

Βλέπουμε πως συμμετέχει με το δίκτυο 0.0.0.0/0, το οποίο πρακτικά σημαίνει πως συμμετέχει με το WAN1 (10.0.1.0/30), WAN3 (10.0.1.4/30), LAN1 (192.168.1.0/24).

```

R1(config)# do show ip rip status
Routing Protocol is "rip"
  Sending updates every 30 seconds with +/-50%, next due in 3 seconds
  Timeout after 180 seconds, garbage collect after 120 seconds
  Outgoing update filter list for all interface is not set
  Incoming update filter list for all interface is not set
  Default redistribution metric is 1
  Relaxed receiving size checks are off
  Redistributing:
  Default version control: send version 2, receive version 2
    Interface      Send  Recv  Key-chain
    em0           2     2
    em1           2     2
    em2           2     2
    lo0           2     2
  Routing for Networks:
    0.0.0.0/0
  Routing Information Sources:
    Gateway      BadPackets BadRoutes Distance Last Update
    10.0.1.6          0         0        120   00:00:15
    10.0.1.2          0         0        120   00:00:26
  Distance: (default is 120)

```

5.7

Εκτελούμε στον R1 “**tcpdump -vvvi em2 udp port 520**” και βλέπουμε πως ο R1 διαφημίζει στο LAN1 τα παρακάτω δίκτυα:

- 10.0.0.0/30 (CORE)
- 10.0.1.0/30 (WAN1)
- 10.0.1.4/30 (WAN3)
- 10.0.2.0/30 (WAN2)
- 10.0.2.4/30 (WAN4)
- 172.22.1.1/32 (R1)
- 172.22.1.2/32 (C1)
- 172.22.2.1/32 (R2)
- 172.22.2.2/32 (C2)
- 192.168.2.0/24 (LAN2)

```
192.168.1.1.router > 224.0.0.9.router: [bad udp cksum 0xa298 -> 0xa657!]
    RIPv2, Response, length: 204, routes: 10
      AFI IPv4,          10.0.0.0/30, tag 0x0000, metric: 2, next-hop: self
      AFI IPv4,          10.0.1.0/30, tag 0x0000, metric: 1, next-hop: self
      AFI IPv4,          10.0.1.4/30, tag 0x0000, metric: 1, next-hop: self
      AFI IPv4,          10.0.2.0/30, tag 0x0000, metric: 2, next-hop: self
      AFI IPv4,          10.0.2.4/30, tag 0x0000, metric: 2, next-hop: self
      AFI IPv4,          172.22.1.1/32, tag 0x0000, metric: 1, next-hop: self
      AFI IPv4,          172.22.1.2/32, tag 0x0000, metric: 2, next-hop: self
      AFI IPv4,          172.22.2.1/32, tag 0x0000, metric: 2, next-hop: self
      AFI IPv4,          172.22.2.2/32, tag 0x0000, metric: 2, next-hop: self
      AFI IPv4,          192.168.2.0/24, tag 0x0000, metric: 3, next-hop: self
```

5.8

Στον πίνακα δρομολόγησης του PC1 δε βλέπουμε αντίστοιχες εγγραφές:

```
PC1(config)# do show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
C>* 192.168.1.0/24 [0/1] is directly connected, em0, 00:20:05
```

5.9

Εκτελούμε στο PC1 όντας σε Global Configuration Mode “**router rip**” → “**version 2**” → “**network em0**”.

5.10

Πλέον και ο πίνακας δρομολόγησης του PC1 περιέχει επίσης 10 δυναμικές εγγραφές.

```

PC1(config)# do show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

R>* 10.0.0.0/30 [120/3] via 192.168.1.1, em0, weight 1, 00:01:09
R>* 10.0.1.0/30 [120/2] via 192.168.1.1, em0, weight 1, 00:01:09
R>* 10.0.1.4/30 [120/2] via 192.168.1.1, em0, weight 1, 00:01:09
R>* 10.0.2.0/30 [120/3] via 192.168.1.1, em0, weight 1, 00:01:09
R>* 10.0.2.4/30 [120/3] via 192.168.1.1, em0, weight 1, 00:01:09
R>* 172.22.1.1/32 [120/2] via 192.168.1.1, em0, weight 1, 00:01:09
R>* 172.22.1.2/32 [120/3] via 192.168.1.1, em0, weight 1, 00:01:09
R>* 172.22.2.1/32 [120/3] via 192.168.1.1, em0, weight 1, 00:01:09
R>* 172.22.2.2/32 [120/3] via 192.168.1.1, em0, weight 1, 00:01:09
C>* 192.168.1.0/24 [0/1] is directly connected, em0, 00:22:40
R>* 192.168.2.0/24 [120/4] via 192.168.1.1, em0, weight 1, 00:01:09

```

5.11

Αντίστοιχα εκτελούμε στο PC2 όντας σε Global Configuration Mode “**router rip**” → “**version 2**” → “**network em0**”.

5.12

Υπάρχουν 2 διαδρομές ελαχίστου κόστους μεταξύ των LAN1 και LAN2, οι εξής:

- LAN1 – WAN1 – WAN2 – LAN2
- LAN1 – WAN3 – WAN4 – LAN2

5.13

Κάνουμε traceroute από το PC1 στο PC2 και βλέπουμε πως η διαδρομή που ακολουθείται είναι: PC1 → R1 → 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)  0.043 ms  0.173 ms  0.159 ms
 2  10.0.1.6 (10.0.1.6)  0.341 ms  0.406 ms  0.331 ms
 3  10.0.2.5 (10.0.2.5)  0.441 ms  0.449 ms  0.394 ms
 4  192.168.2.2 (192.168.2.2)  0.636 ms  0.555 ms  1.307 ms

```

5.14

Κάνουμε traceroute από το PC2 στο PC1 και βλέπουμε πως η διαδρομή που ακολουθείται είναι: 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)  0.277 ms  0.236 ms  0.228 ms
 2  10.0.2.6 (10.0.2.6)  0.307 ms  0.278 ms  0.424 ms
 3  10.0.1.5 (10.0.1.5)  0.429 ms  0.374 ms  0.381 ms
 4  192.168.1.2 (192.168.1.2)  0.606 ms  0.495 ms  0.584 ms

```

5.15

Ναι, χρησιμοποιείται η ίδια διαδρομή.

5.16

Εκτελούμε Ping από το PC1 προς κάθε loopback διαχείρισης και παρατηρούμε πως εκτελείται κανονικά.

5.17

Εκτελούμε Ping από το PC2 προς κάθε loopback διαχείρισης και παρατηρούμε πως εκτελείται κανονικά.

5.18

Μπορεί να αποκοπεί οποιοδήποτε από αυτά χωρίς να διακοπεί η σύνδεση.

5.19

Μπορούν να αποκοπούν όλα τα δίκτυα του C1 χωρίς να διακοπεί η σύνδεση.

5.20

Δε μπορούν να αποκοπούν τα WAN1 και WAN3, καθώς τότε δε μπορεί το R1 να επικοινωνήσει με το υπόλοιπο δίκτυο.

5.21

Μπορούν να αποκοπούν τα WAN2 και WAN3 χωρίς κανένα πρόβλημα.

5.22

Δε μπορούν να αποκοπούν τα WAN2 και WAN4, καθώς τότε δε μπορεί το R2 να επικοινωνήσει με το υπόλοιπο δίκτυο.

5.23

Μπορούν να αποκοπούν όλα τα δίκτυα του C2 χωρίς κανένα πρόβλημα.

5.24

Μπορούν να αποκοπούν τα WAN1 και WAN4 χωρίς κανένα πρόβλημα.

5.25

Αποσυνδέοντας το CORE, το ping συνεχίζει να επιτυγχάνει κανονικά, Ωστόσο με την αποσύνδεση και του WAN3 λαμβάνουμε το μήνυμα “ping: sendto: No route to host”. Παρατηρώντας τον πίνακα δρομολόγησης του R1 βλέπουμε πως δρομολογεί πακέτα στην loopback του C2 άμεσα μέσω του WAN3, επομένως η διακοπή του CORE δεν επέφερε

καμία αλλαγή, ενώ η διακοπή του WAN3 ανάγκασε το R1 να βρει εναλλακτική διαδρομή για το C2, όπως και βλέπουμε ότι έκανε.

```
64 bytes from 172.22.2.2: icmp_seq=29 ttl=63 time=0.529 ms
64 bytes from 172.22.2.2: icmp_seq=30 ttl=63 time=0.522 ms
ping: sendto: No route to host
64 bytes from 172.22.2.2: icmp_seq=42 ttl=61 time=0.873 ms
64 bytes from 172.22.2.2: icmp_seq=43 ttl=61 time=0.813 ms
```

```
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>* 10.0.1.0/30 is directly connected, em1
R>* 10.0.2.0/30 [120/2] via 10.0.1.2, em1, 00:40:33
R>* 10.0.2.4/30 [120/3] via 10.0.1.2, em1, 00:01:12
C>* 127.0.0.0/8 is directly connected, lo0
C>* 172.22.1.1/32 is directly connected, lo0
R>* 172.22.1.2/32 [120/2] via 10.0.1.2, em1, 00:40:33
R>* 172.22.2.1/32 [120/3] via 10.0.1.2, em1, 00:01:12
R>* 172.22.2.2/32 [120/4] via 10.0.1.2, em1, 00:01:34
C>* 192.168.1.0/24 is directly connected, em0
R>* 192.168.2.0/24 [120/3] via 10.0.1.2, em1, 00:01:12
```

5.26

Χρειάστηκαν περίπου 10 δευτερόλεπτα για την αποκατάσταση του δικτύου.

Άσκηση 6: RIP και αναδιανομή διαδρομών

6.1

Εκτελούμε στο C1 όντας σε Global Configuration Mode “**ip route 4.0.0.0/8 172.22.1.2**”.

6.2

Παρατηρούμε πως έχει προστεθεί η παραπάνω εγγραφή στον πίνακα δρομολόγησης.

```

C1(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

S>* 4.0.0.0/8 [1/0] via 172.22.1.2, lo0
C>* 10.0.0.0/30 is directly connected, em2
C>* 10.0.1.0/30 is directly connected, em0
R>* 10.0.1.4/30 [120/2] via 10.0.1.1, em0, 00:29:05
C>* 10.0.2.0/30 is directly connected, em1
R>* 10.0.2.4/30 [120/2] via 10.0.2.1, em1, 00:29:05
C>* 127.0.0.0/8 is directly connected, lo0
R>* 172.22.1.1/32 [120/2] via 10.0.1.1, em0, 00:29:05
C>* 172.22.1.2/32 is directly connected, lo0
R>* 172.22.2.1/32 [120/2] via 10.0.2.1, em1, 00:29:05
R>* 172.22.2.2/32 [120/2] via 10.0.0.2, em2, 00:28:45
R>* 192.168.1.0/24 [120/2] via 10.0.1.1, em0, 00:29:05
R>* 192.168.2.0/24 [120/2] via 10.0.2.1, em1, 00:29:05

```

6.3

Εκτελώντας “**do show ip route**” σε κάθε άλλο δρομολογητή/PC βλέπουμε πως δεν έχει προστεθεί η παραπάνω εγγραφή στους πίνακες δρομολόγησης τους.

6.4

Εκτελούμε στο C1 “**router rip**” → “**redistribute static**” χωρίς, ωστόσο, να παρατηρούμε κάποια αλλαγή στον πίνακα δρομολόγησης του C1.

6.5

Προστέθηκε πλέον η διαδρομή για το 4.0.0.0/8 στους πίνακες δρομολόγησης των άλλων μηχανημάτων, ως δυναμική εγγραφή.

6.6

Εκτελούμε στο C2 “**ip route 0.0.0.0/0 172.22.2.2**”.

6.7

Ναι, προστέθηκε στον πίνακα του C2 ως στατική εγγραφή.

6.8

Όχι, δε προστέθηκε η εγγραφή στους υπόλοιπους κόμβους του δικτύου.

6.9

Εκτελούμε στο C2 “**router rip**” → “**default-information originate**” και βλέπουμε πως δεν έχει αλλάξει κάτι στον πίνακα δρομολόγησης του C2.

6.10

Στους υπόλοιπους δρομολογητές και υπολογιστές του δικτύου έχει προστεθεί η εγγραφή για το 0.0.0.0/0 ως δυναμική. Πρακτικά, για τους δρομολογητές έγινε προκαθορισμένη πύλη το C2, ενώ για τα PC1, PC2 έγιναν τα R1 και R2 αντίστοιχα, τα οποία συνδέονται άμεσα με το C2.

6.11

Εκτελούμε στο C2 “**router rip**” → “**no default-information originate**”. Μετά εκτελούμε στο C1 “**ip route 0.0.0.0/0 10.0.0.2**” → “**router rip**” → “**default-information originate**”.

6.12

Στον πίνακα δρομολόγησης του C2 προστίθεται μια νέα δυναμική εγγραφή για το 0.0.0.0/0 μέσω της 10.0.0.1, η οποία, ωστόσο δε χρησιμοποιείται, καθώς χρησιμοποιείται η στατική εγγραφή για το 0.0.0.0/0 μέσω της 172.22.2.2.

```
C2(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

R  0.0.0.0/0 [120/2] via 10.0.0.1, em2, 00:00:57
S>* 0.0.0.0/0 [1/0] via 172.22.2.2, lo0
R>* 4.0.0.0/8 [120/2] via 10.0.0.1, em2, 00:33:17
C>* 10.0.0.0/30 is directly connected, em2
R>* 10.0.1.0/30 [120/2] via 10.0.1.5, em0, 01:08:23
C>* 10.0.1.4/30 is directly connected, em0
R>* 10.0.2.0/30 [120/2] via 10.0.0.1, em2, 01:08:23
C>* 10.0.2.4/30 is directly connected, em1
C>* 127.0.0.0/8 is directly connected, lo0
R>* 172.22.1.1/32 [120/2] via 10.0.1.5, em0, 01:08:23
R>* 172.22.1.2/32 [120/2] via 10.0.0.1, em2, 01:08:23
R>* 172.22.2.1/32 [120/3] via 10.0.0.1, em2, 00:00:01
C>* 172.22.2.2/32 is directly connected, lo0
R>* 192.168.1.0/24 [120/2] via 10.0.1.5, em0, 01:08:23
```

6.13

Εκτελούμε στο C2 “**no ip route 0.0.0.0/0 172.22.2.2**” και πλέον επιλέγεται ως προκαθορισμένη πύλη η ανωτέρω δυναμική εγγραφή που αναφέραμε.

6.14

Παρατηρούμε πως έχει μέγεθος 14 εγγραφών:

```

C2(config-if)# no ip address 172.22.2.1/32
% Can't find address
C2(config-if)# exit
C2(config)# interface lo0
C2(config-if)# no ip address 172.22.2.1/32
C2(config-if)# 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

R>* 0.0.0.0/0 [120/2] via 10.0.0.1, em2, 00:08:35
R>* 4.0.0.0/8 [120/2] via 10.0.0.1, em2, 00:40:55
C>* 10.0.0.0/30 is directly connected, em2
R>* 10.0.1.0/30 [120/2] via 10.0.1.5, em0, 01:16:01
C>* 10.0.1.4/30 is directly connected, em0
R>* 10.0.2.0/30 [120/2] via 10.0.0.1, em2, 01:16:01
C>* 10.0.2.4/30 is directly connected, em1
C>* 127.0.0.0/8 is directly connected, lo0
R>* 172.22.1.1/32 [120/2] via 10.0.1.5, em0, 01:16:01
R>* 172.22.1.2/32 [120/2] via 10.0.0.1, em2, 01:16:01
R>* 172.22.2.1/32 [120/3] via 10.0.0.1, em2, 00:00:01
C>* 172.22.2.2/32 is directly connected, lo0
R>* 192.168.1.0/24 [120/2] via 10.0.1.5, em0, 01:16:01
R>* 192.168.2.0/24 [120/2] via 10.0.2.5, em1, 01:16:01
C2(config-if)#

```

6.15

Κάνοντας ping από το PC1 στο 4.4.4.4 λαμβάνουμε μήνυμα Time-To-Live Exceeded, ενώ κάνοντας traceroute παρατηρούμε πως ακολουθείται η διαδρομή PC1 → R1 → C1 → C1 → C1 ... → C1. Αυτό που συμβαίνει είναι πως αφού φτάσει το πακέτο στο C1, δρομολογείται στο δίκτυο 4.0.0.0/8 στέλνοντάς το στην Loopback του C1, άρα κάνοντας το ουσιαστικά να πάει ξανά στο C1 με αποτέλεσμα να δημιουργείται αυτό το loop.

6.16

Κάνοντας ping από το PC1 στο 5.5.5.5 λαμβάνουμε επίσης TTL exceeded, ενώ με traceroute βλέπουμε πως ακολουθείται η διαδρομή PC1 → R1 → C1 → C2 → C1 → C2 ... μέχρι να μηδενιστεί το TTL. Αναλυτικότερα, το PC1 προωθεί το πακέτο στην προκαθορισμένη πύλη του, το R1 και το R1 κάνει το ίδιο με τη δικιά του πύλη, δηλαδή το C1. Το C1 με τη σειρά του προωθεί το πακέτο στη δική του default gateway, την 10.0.0.2 του C2. Το C2 με τη σειρά του προωθεί το πακέτο στη δικιά του default gateway, δηλαδή το 10.0.0.1 του C1, προκαλώντας loop το οποίο θα λήξει με το μηδενισμό του TTL.

6.17

Εκτελούμε στο R1 “**access-list private permit 192.168.0.0/16**” → “**access-list private deny any**”.

6.18

Εκτελούμε όντας σε configuration mode στον R1 “**password ntua**” → “**exit**” → “**exit**”.

6.19

Εκτελούμε στο PC2 “**telnet 10.0.1.1 2602**”.

6.20

Εκτελούμε στο PC2, το οποίο τώρα χειρίζεται τον RIP daemon του R1 “**enable**” → “**configure terminal**” → “**router rip**” → “**distribute-list private out em0**”.

6.21

Δε παρατηρούμε άμεσα κάποια αλλαγή στον πίνακα δρομολόγησης του PC1. Ωστόσο, 3 λεπτά μετά παρατηρούμε πως διαγράφονται όλες οι δυναμικές εγγραφές του PC1 εκτός αυτής των 192.168.1.0/24 και 192.168.2.0/24, διότι είναι υποδίκτυα του 192.168.0.0/16.

6.22

Παρατηρούμε πως αντίστοιχα οι δυναμικές εγγραφές πλην αυτής του LAN2 διαγράφηκαν περίπου 2 λεπτά μετά την παρέλευση των 3 λεπτών. Συγκεκριμένα, μετά από τα 3 πρώτα λεπτά, το κόστος δρομολόγησης έγινε ίσο με 16, ενώ αφού πέρασαν 2 επιπλέον λεπτά οι εγγραφές διαγράφηκαν εντελώς.