Εργαστήριο Δικτύων Υπολογιστών

Εργαστηριακή Άσκηση 6
Εισαγωγή στο Quagga και FRRouting (FRR)

Ονοματεπώνυμο: Νικόλαος Παγώνας, el18175	**Όνομα PC:** nick-ubuntu
Ομάδα: 1 (Τρίτη 10:45)	**Ημερομηνία Εξέτασης:** Τρίτη 05/04/2022

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

1

Ορίζουμε την πρώτη δικτυακή διεπαφή σε NAT.

2

Εκτελούμε `dhclient em0`.

3

Εκτελούμε `ping www.google.com`.

4

Εκτελούμε `pkg update`.

5

Εκτελούμε `pkg install frr7`.

6

Προσθέτουμε τη γραμμή `kern.ipc.maxsockbuf=16777216`.

7

Εκτελούμε `chown frr:frr /usr/local/etc/frr/`.

8

Εκτελούμε `touch /usr/local/etc/frr/{vtysh.conf,zebra.conf,staticd.conf}`.

9

Εκτελούμε `chown frr:frr /usr/local/etc/frr/{vtysh.conf,zebra.conf,staticd.conf}`

10

Στο αρχείο εκκίνησης `/etc/rc.conf` προσθέτουμε τις εντολές:

```
hostname="R0"
gateway_enable="YES"
frr_enable="YES"
frr_daemons="zebra staticd"
```

11

Στο αρχείο `/etc/csh.cshrc` προσθέτουμε τη γραμμή:

```
setenv VTYSH_PAGER "more -EFX"
```

12

Διαγράφουμε το αρχείο `/etc/resolv.conf` που δημιούργησε ο πελάτης DHCP και κλείνουμε το μηχάνημα με την εντολή `poweroff`.

13

Ενεργοποιούμε τις άλλες 3 κάρτες δικτύωσης, ρυθμίζουμε όλες τις κάρτες σε εσωτερική δικτύωση, επιλέγουμε Deny στο Promiscuous Mode και δίνουμε στα εσωτερικά δίκτυα διαφορετικά ονόματα LAN1,2,3,4.

14

Κάνουμε επανεκκίνηση του FreeBSD και βεβαιωνόμαστε ότι η υπηρεσία `sshd` και το `frr` τρέχουν με τις εντολές `ps aux | grep sshd` και `ps aux | grep frr`.

15

Εκτελούμε `history -c`, ύστερα `poweroff` και κάνουμε Export Appliance, δημιουργώντας ένα αρχείο `frr.ova`.

16

Αποθηκεύουμε το αρχείο `frr.ova` για μελλοντική χρήση.

1.1

Εκτελούμε `telnet localhost 2601`. Εμφανίζεται μήνυμα λάθους:

```
Vty password is not set.
Connection closed by foreign host.
```

1.2

Η εντολή `vtysh`.

1.3

Βλέπουμε 22 εντολές.

1.4

Παρατηρούμε ότι γίνεται αυτόματη συμπλήρωση της εντολής traceroute.

1.5

Πατώντας το tab εμφανίζονται οι εντολές που ξεκινάνε με ”co”, ενώ πατώντας το `"?"` εμφανίζονται επιπλέον σύντομες περιγραφές γι’ αυτές τις εντολές.

1.6

Είναι `show version`.

1.7

Είναι `w t`.

1.8

Με την εντολή `show running-config`.

1.9

Με την εντολή `configure terminal`

1.10

Εκτελούμε `configure terminal` και ύστερα `hostname R1`. Παρατηρούμε ότι αλλάζει το prompt, από `R0(config)` σε `R1(config)`.

1.11

Εκτελούμε `password ntua`.

1.12

Πρέπει να δώσουμε την εντολή ”exit” 2 φορές.

1.13

Δοκιμάζουμε πάλι `telnet localhost 2601`. Παρατηρούμε ότι αυτή τη φορά αντί να αποτυγχάνει το telnet, μας ζητείται να εισάγουμε το password που ορίσαμε.

1.14

Βρισκόμαστε σε επίπεδο User EXEC, όπως δείχνει και το prompt `R0>` αντί `R0#`.

1.15

Βλέπουμε 9 εντολές.

1.16

Σε επίπεδο User EXEC βλέπουμε 13 λιγότερες εντολές, όπως είναι λογικό, αφού το επίπεδο User EXEC έχει λιγότερες ελευθερίες από το επίπεδο Privileged EXEC.

1.17

Με την εντολή `show interface`.

1.18

Εκτελούμε `show ip forwarding`. Η προώθηση είναι ενεργοποιημένη.

1.19

Εκτελούμε `show ip route`.

1.20

Όχι, διότι η εντολή `show running-config / write terminal` ανήκει στο επίπεδο Privileged EXEC.

1.21

Με την εντολή `enable`.

1.22

Ναι, μπορούμε να δούμε την τρέχουσα παραμετροποίηση του FRR με `write terminal`. Το συνθηματικό εμφανίζεται στην έξοδο της εντολής.

1.23

Εκτελούμε `list`.

1.24

Σε Global Configuration Mode (δηλαδή αφού έχουμε εκτελέσει `configure terminal` σε λειτουργία Privileged EXEC), εκτελούμε `enable password ntua`.

1.25

Με την εντολή `service password-encryption`.

1.26

Θα προτιμούσαμε την επιλογή του ssh, καθώς ως γνωστόν το ssh αποτελεί πολύ ασφαλέστερη επιλογή από το telnet, για λόγους που έχουν συζητηθεί στο μάθημα των Δικτύων Υπολογιστών (το telnet στέλνει τα πακέτα χωρίς κανενός είδους κρυπτογράφηση -ακόμα και τους κωδικούς-, ενώ το ssh χρησιμοποιεί κρυπτογραφημένη επικοινωνία).

Άσκηση 2: Δρομολόγηση σε ένα βήμα

2.1

Με τις εντολές:

```
PC1: ifconfig em0 192.168.1.2/24
PC2: ifconfig em0 192.168.2.2/24
```

2.2

Εκτελούμε στον R1:

```
vtysh
configure terminal
hostname R1
interface em0
ip address 192.168.1.1/24
exit
interface em1
ip address 192.168.2.1/24
exit
```

2.3

Στον R1 εκτελούμε `do show interface`. Οι διευθύνσεις ορίστηκαν κανονικά.

2.4

Εκτελούμε `do show ip forwarding`. Αν δεν είναι ενεργοποιημένη, την ενεργοποιούμε με `ip forwarding`. Στη δική μας περίπτωση είναι ενεργοποιημένη.

2.5

Στο PC1 εκτελούμε `route add -net 192.168.2.0/24 192.168.1.1`.

2.6

Στο PC2 εκτελούμε `route add -net 192.168.1.0/24 192.168.2.1`.

2.7

Εκτελούμε `ping 192.168.2.2` από το PC1 στο PC2. Οι δύο υπολογιστές επικοινωνούν.

2.8

Στον R1 μέσω vtysh εκτελούμε:

```
interface em0
ip address 192.168.1.200/24
do show interface em0
```

Παρατηρούμε ότι η διεύθυνση `192.168.1.200/24` έχει οριστεί ως secondary, ενώ η προηγούμενη `192.168.1.1/24` δεν έχει σβηστεί.

2.9

Εκτελούμε `ifconfig em0` από τη γραμμή εντολών του R1. Βλέπουμε ότι η `em0` έχει όντως δύο διευθύνσεις IPv4, οι οποίες συμφωνούν με το προηγούμενο ερώτημα, ωστόσο σε αυτή την περίπτωση δεν αναγράφεται κάποια διεύθυνση ως secondary.

2.10

Στον R2 εκτελούμε:

```
no ip address 192.168.1.200/24
```

και ύστερα με `ifconfig em0` επιβεβαιώνουμε ότι η ζητούμενη διεύθυνση διαγράφτηκε.

2.11

Με την εντολή `do write memory` ή `do write file`.

2.12

Από την έξοδο της παραπάνω εντολής βλέπουμε ότι ενημερώνονται τα αρχεία `zebra.conf` και `staticd.conf` του φακέλου `/usr/local/etc/frr/`.

Άσκηση 3: Δρομολόγηση σε περισσότερα βήματα

3.1

Εκτελούμε:

```
PC1:
ifconfig em0 192.168.1.2/24
route add -net 192.168.2.0/24 192.168.1.1

PC2:
ifconfig em0 192.168.2.2/24
route add -net 192.168.1.0/24 192.168.2.1
```

3.2

Εκτελούμε στον R1:

```
vtysh
configure terminal
hostname R1
interface em0
ip address 192.168.1.1/24
interface em1
ip address 172.17.17.1/30
```

3.3

Εκτελούμε στον R2:

```
vtysh
configure terminal
hostname R2
interface em0
ip address 172.17.17.2/30
interface em1
ip address 192.168.2.1/24
```

3.4

Εκτελούμε στον R1 μέσω `vtysh`:

```
configure terminal
ip route 192.168.2.0/24 172.17.17.2
```

3.5

Εκτελούμε στον R2 μέσω `vtysh`:

```
configure terminal
ip route 192.168.1.0/24 172.17.17.1
```

3.6

Εκτελούμε `telnet 192.168.1.1 2601`. Η σύνδεση αποτυγχάνει με μήνυμα:

```
Vty password is not set.
Connection closed by foreign host.
```

Για να γίνει δυνατή η παραπάνω σύνδεση πρέπει να ορίσουμε κωδικό στον R1 για την πιστοποίηση απομακρυσμένων συνδέσεων (πχ `password ntua`).

3.7

Εκτελούμε ? και βλέπουμε ότι δεν υπάρχει διαθέσιμη εντολή σχετική με σύνδεση στην υπηρεσία zebra άλλου μηχανήματος, άρα δεν μπορούμε να συνδεθούμε στην υπηρεσία zebra του R2 μέσω του path PC1→R1→R2.

3.8

Θα κάναμε `telnet 192.168.2.1 2601` διότι το PC1 έχει εγγραφή στον πίνακα δρομολόγησης μόνο για το υποδίκτυο `192.168.2.0/24` και όχι για το `172.17.17.0/30`.

3.9

Με την εντολή `who`. Δεν εμφανίζεται ο χρήστης που έχει εισέλθει τοπικά μέσω `vtysh` στον R2, μόνο ο PC2 (`192.168.2.2`).

3.10

Όχι, διότι οι εντολές `ping` και `traceroute` δεν εμφανίζονται όταν πατάμε το πλήκτρο `"?"`. Αντίθετα, από την τοπική σύνδεση μπορούμε.

3.11

Εκτελούμε `traceroute 192.168.1.2` από τον R2 και `traceroute 192.168.2.2` από τον R1. Τα traceroute δεν ολοκληρώνονται διότι δεν έχει τεθεί στον πίνακα δρομολόγησης των PC1, PC2 εγγραφή το δίκτυο `172.17.17.0/30`, οπότε δεν μπορούν τα PC1,2 να απαντήσουν με ”ICMP udp port unreachable”.

3.12

Μπορούμε να κάνουμε:

```
PC1: route add -net 172.17.17.0/30 192.168.1.1
PC2: route add -net 172.17.17.0/30 192.168.2.1
```

Άσκηση 4: Εναλλακτικές διαδρομές

4.1

Εκτελούμε:

```
PC1:
ifconfig em0 192.168.1.2/24
route add default 192.168.1.1

PC2:
ifconfig em0 192.168.2.2/24
route add default 192.168.2.1
```

4.2

Εκτελούμε στον R1:

```
cli
configure terminal
hostname R1
interface em0
ip address 192.168.1.1/24
interface em1
ip address 172.17.17.1/30
interface em2
ip address 172.17.17.5/30
```

4.3

Εκτελούμε στον R1 μέσω cli:

```
ip route 192.168.2.0/24 172.17.17.2
```

4.4

Εκτελούμε στον R1 μέσω cli:

```
do show ip route
```

Εμφανίζονται:

- 127.0.0.0/8 → lo0
- 172.17.17.0/30 → em1
- 172.17.17.4/30 → em2
- 192.168.1.0/24 → em0
- 192.168.2.0/24 → em1

4.5

Με τη φράση `is directly connected` στην έξοδο της εντολής.

4.6

Μέσω του flag ”S” στην αρχή της γραμμής:

```
===> S>* 192.168.2.0/24 [1/0] via 172.17.17.2, em1
```

4.7

Εκτελούμε `netstat -rn`. Οι εγγραφές συμφωνούν, απλώς η `netstat` έχει κάποιες επιπλέον εγγραφές που αφορούν την `lo0`.

4.8

Έχουν δηλωθεί οι σημαίες ”UG1”.

- ”U”: σημαίνει ότι η διαδρομή είναι ενεργή (up).
- ”G”: σημαίνει ότι ο προορισμός είναι πύλη, που θα αποφασίσει για το πώς θα προωθήσει τα πακέτα περαιτέρω.
- ”1”: μια σημαία που εξαρτάται από το πρωτόκολλο δρομολόγησης (Protocol specific routing flag #1)

4.9

Εκτελούμε στον R2 μέσω cli:

```
configure terminal
hostname R2
interface em0
ip address 172.17.17.2/30
interface em1
ip address 172.17.17.9/30
interface em2
ip address 192.168.2.1/24
```

4.10

Εκτελούμε στον R2 μέσω cli:

```
ip route 192.168.1.0/24 172.17.17.1
```

4.11

Εκτελούμε στον R3 μέσω cli:

```
hostname R3
interface em0
ip address 172.17.17.6/30
interface em1
ip address 172.17.17.10/30
```

4.12

Εκτελούμε στον R3 μέσω cli:

```
ip route 192.168.1.0/24 172.17.17.5
ip route 192.168.2.0/24 172.17.17.9
```

4.13

Εκτελούμε στον R3 μέσω cli:

```
do show ip forwarding
```

Εμφανίζεται μήνυμα ”IP forwarding is **on**”.

4.14

Στο PC1 εκτελούμε `traceroute 192.168.2.2`. Τα πακέτα ακολουθούν τη διαδρομή:

PC1 $\rightarrow$ R1 $\rightarrow$ R2 $\rightarrow$ PC2.

Άσκηση 5: Σφάλμα καλωδίου και αυτόματη αλλαγή στη δρομολόγηση

5.1

Μέσω cli στον R1:

```
ip route 192.168.2.0/24 172.17.17.6 2
```

5.2

Δώσαμε την τιμή 2, για να είναι μεγαλύτερη (και άρα λιγότερο προτιμητέα) από την προηγούμενη σχετική στατική εγγραφή, που έχει απόσταση 1.

5.3

Μέσω cli στον R2:

```
ip route 192.168.1.0/24 172.17.17.10 2
```

5.4

Στους R1 και R2 μέσω cli:

```
do show ip route
```

R1 σχετικά με το LAN2:

```
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
```

R2 σχετικά με το LAN1:

```
S   192.168.1.0/24 [2/0] via 172.17.17.10, em1
S>* 192.168.1.0/24 [1/0] via 172.17.17.1, em0
```

5.5

Στον R1 είναι ενεργοποιημένη η διαδρομή μέσω του R2 (διεπαφή `172.17.17.2`). Αυτό φαίνεται από το σύμβολο `">"` στην αντίστοιχη εγγραφή του ερωτήματος `5.4`.

5.6

Η διαχειριστική απόσταση είναι ο πρώτος αριθμός μέσα σε αγκύλες (πχ `[2/0]` → distance = 2).

5.7

Η διαδρομή μέσω του R1 (διεπαφή `172.17.17.1`).

5.8

Εκτελούμε μέσω cli:

```
R1:
interface em1
link-detect

R2:
interface em0
link-detect
```

5.9

Κάνουμε δεξί κλικ στο εικονίδιο δικτύου του εικονικού μηχανήματος R1 και ύστερα αποεπιλογή του ”Connect Network Adapter 2”.

5.10

Η διαδρομή μέσω του R3 (διεπαφή `172.17.17.6`).

5.11

Ναι υπάρχει, είναι `inactive`.

5.12

Στον R1 εκτελούμε `netstat -rn` και παρατηρούμε ότι έχει γίνει η αντίστοιχη αλλαγή στον πίνακα δρομολόγησης:

```
192.168.2.0/24 172.17.17.6
```

5.13

Στον R2 εκτελούμε `do show ip route` και βλέπουμε ότι είναι ενεργοποιημένη η διαδρομή μέσω του R1 (`172.17.17.1`), διότι το αντίστοιχο καλώδιο δεν έχει αποσυνδεθεί στον R2 και έτσι αυτός δεν αντιλαμβάνεται την διακοπή της ζεύξης που εκτελέσαμε προηγουμένως.

5.14

Αποσυνδέουμε το ζητούμενο καλώδιο. Εκτελούμε `do show ip route` και διαπιστώνουμε ότι έγινε σωστά η μετάβαση στην εναλλακτική διαδρομή μέσω του R3 (`172.17.17.10`).

5.15

Στον PC1 εκτελούμε `traceroute 192.168.2.2`, και βλέπουμε ότι στα βήματα 2 και 3 της διαδρομής λαμβάνουμε απαντήσεις από τον R3 (`172.17.17.6`) και τον R2 (`172.17.17.9`) αντίστοιχα, επιβεβαιώνοντας όσα συζητήθηκαν παραπάνω.

5.16

Στον PC2 εκτελούμε `ssh lab@192.168.1.2` και ύστερα επανασυνδέουμε τα δύο καλώδια που αποσυνδέσαμε προηγουμένως. Παρατηρούμε ότι η σύνδεση SSH δεν χάνεται.

5.17

Εκτελώντας `do show ip route` στους R1, R2, διαπιστώνουμε ότι πλέον προτιμώνται οι αρχικές διαδρομές, δηλαδή:

- LAN1 → LAN2: R1 → R2
- LAN2 → LAN1: R2 → R1

Για να το εξακριβώσουμε, εκτελούμε `traceroute 192.168.2.2` από το PC1 ή `traceroute 192.168.1.2` από το PC2.

Άσκηση 6: Διευθύνσεις διαχείρισης (loopback)

6.1

Εκτελούμε μέσω cli:

```
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

Εκτελούμε "`ping 172.22.22.x`" $(x = 1, 2, 3)$ από τα PC1 και PC2. Μόνο τα ping από PC1 προς `172.22.22.1` και PC2 προς `172.22.22.2` επιτυγχάνουν. Τα υπόλοιπα ping όπως είναι αναμενόμενο αποτυγχάνουν, αφού δεν υπάρχουν αντίστοιχες εγγραφές στους πίνακες δρομολόγησης των R1,2,3.

6.3

Στον R1 μέσω cli:

```
ip route 172.22.22.2/32 172.17.17.2
ip route 172.22.22.3/32 172.17.17.6
```

6.4

Στον R2 μέσω cli:

```
ip route 172.22.22.1/32 172.17.17.1
ip route 172.22.22.3/32 172.17.17.10
```

6.5

Στον R3 μέσω cli:

```
ip route 172.22.22.1/32 172.17.17.5
ip route 172.22.22.2/32 172.17.17.9
```

6.6

Ξαναεκτελούμε `ping 172.22.22.x` από τα PC1,PC2. Αυτή τη φορά όλα τα ping επιτυγχάνουν.

6.7

Εκτελούμε:

```
PC1: tcpdump -i em0
PC2: tcpdump -i em0
```

Αν εκτελέσουμε `ping 192.168.1.2` και `ping 192.168.2.2` από την κονσόλα του R3:

```
ICMP source address for ping R3 --> PC1 == 172.17.17.6
ICMP source address for ping R3 --> PC2 == 172.17.17.10
```

6.8

Θα εκτελέσουμε `ping -S 172.22.22.3 192.168.1.2` και `ping -S 172.22.22.3 192.168.2.2`

6.9

Η δυσκολία έγκειται στο ότι κάθε φορά που δημιουργείται ένα πρόβλημα δρομολόγησης στα PC, πρέπει να ελέγξουμε όλον τον πίνακα δρομολόγησης για να διαπιστώσουμε μήπως το πρόβλημα οφείλεται στην απουσία κάποιας στατικής εγγραφής (που ενδεχομένως ξεχάσαμε να ορίσουμε ή διαγράφηκε με τον επαναπροσδιορισμό της διεύθυνσης IP του PC). Αντίθετα, αν έχουμε ορίσει default διαδρομή, αρκεί να κάνουμε ping σε αυτή για να καταλάβουμε αν το πρόβλημα δημιουργείται εξαιτίας του PC ή κάποιου άλλου κόμβου.

6.10

Όλα τα ping θα ήταν επιτυχή, εκτός από PC1 → loopback R2 και PC2 → loopback R1, διότι παρόλο που θα μπορούσαν τα πακέτα να προωθηθούν μέσω του R3, δεν έχει οριστεί διαδρομή στον πίνακα δρομολόγησης των R1,2 για τις loopback διευθύνσεις αυτές μέσω του R3.

6.11

Στον R1 μέσω cli:

```
ip route 172.22.22.2/32 172.17.17.6 2
ip route 172.22.22.3/32 172.17.17.2 2
```

6.12

Στον R2 μέσω cli:

```
ip route 172.22.22.1/32 172.17.17.10 2
ip route 172.22.22.3/32 172.17.17.1 2
```

6.13

Στον R3 μέσω cli:

```
ip route 172.22.22.1/32 172.17.17.9 2
ip route 172.22.22.2/32 172.17.17.5 2
```

6.14

Στον R1 μέσω cli:

```
do show ip route
```

Επιλεγμένη διαδρομή είναι αυτή μέσω του R2 `172.17.17.2`.

6.15

Προσομοιώνουμε τη βλάβη στο WAN1 και παρατηρούμε ότι οι διαδρομές που διέρχονται μέσω του WAN1 έχουν δηλωθεί ως `inactive`.

6.16

Αυτή τη φορά δεν εμφανίζονται inactive διαδρομές, αφού δεν έχουμε κάνει enable το `link-detect` στις διεπαφές του WAN2.

Άσκηση 7: Ένα εταιρικό δίκτυο

Υλοποίηση συνδεσμολογίας

Αντιστοίχιση Network Adapters σε LAN

- PC1:
  - Network Adapter 1 (em0): LAN1
- PC2:
  - Network Adapter 1 (em0): LAN2
- R1:
  - Network Adapter 1 (em0): LAN1
  - Network Adapter 2 (em1): WAN1
  - Network Adapter 3 (em2): WAN3
- R2:
  - Network Adapter 1 (em0): LAN2
  - Network Adapter 2 (em1): WAN2
  - Network Adapter 3 (em2): WAN4
- C1:
  - Network Adapter 1 (em0): CORE
  - Network Adapter 2 (em1): WAN1
  - Network Adapter 3 (em2): WAN2
- C2:
  - Network Adapter 1 (em0): CORE
  - Network Adapter 2 (em1): WAN3
  - Network Adapter 3 (em2): WAN4

Ορισμός ονομάτων και διευθύνσεων IP μέσω cli στους δρομολογητές

- R1:

```
cli
configure terminal

hostname R1

interface em0
ip address 192.168.1.1/24

interface em1
ip address 10.0.1.1/30

interface em2
ip address 10.0.1.5/30

interface lo0
ip address 172.22.1.1/32
```

- R2:

```
cli
configure terminal

hostname R2

interface em0
ip address 192.168.2.1/24

interface em1
ip address 10.0.2.1/30

interface em2
ip address 10.0.2.5/30

interface lo0
ip address 172.22.2.1/32
```

- C1:

```
cli
configure terminal

hostname C1

interface em0
```

```
ip address 10.0.0.1/30

interface em1
ip address 10.0.1.2/30

interface em2
ip address 10.0.2.2/30

interface lo0
ip address 172.22.1.2/32
```

- C2:

```
cli
configure terminal

hostname C2

interface em0
ip address 10.0.0.2/30

interface em1
ip address 10.0.1.6/30

interface em2
ip address 10.0.2.6/30

interface lo0
ip address 172.22.2.2/32
```

Ενεργοποίηση link-detect σε όλες τις διεπαφές WAN

- R1:

```
interface em1
link-detect

interface em2
link-detect
```

- R2:

```
interface em1
link-detect

interface em2
link-detect
```

- C1:

```
interface em1
link-detect

interface em2
link-detect
```

- C2:

```
interface em1
link-detect

interface em2
link-detect
```

Ορισμός διευθύνσεων IP και προεπιλεγμένης πύλης στα PC

- PC1:

```
ifconfig em0 192.168.1.2/24
route add default 192.168.1.1
```

- PC2:

```
ifconfig em0 192.168.2.2/24
route add default 192.168.2.1
```

7.1

Στον C1, μέσω cli:

```
ip route 192.168.1.0/24 10.0.1.1
ip route 192.168.1.0/24 10.0.0.2  2
ip route 192.168.2.0/24 10.0.2.1
ip route 192.168.2.0/24 10.0.0.2  2
```

7.2

Στον C2, μέσω cli:

```
ip route 192.168.1.0/24 10.0.1.5
ip route 192.168.1.0/24 10.0.0.1  2
ip route 192.168.2.0/24 10.0.2.5
ip route 192.168.2.0/24 10.0.0.1  2
```

7.3

Στον R1, μέσω cli:

```
ip route 192.168.2.0/24 10.0.1.2
ip route 192.168.2.0/24 10.0.1.6  2
```

7.4

Στον R2, μέσω cli:

```
ip route 192.168.1.0/24 10.0.2.2
ip route 192.168.1.0/24 10.0.2.6  2
```

7.5

Στο PC1 εκτελούμε `ping -c 1 192.168.2.2`. Το ping είναι επιτυχές.

7.6

Αποσυνδέουμε τη ζεύξη WAN2 και εκτελούμε `ping -c 1 192.168.2.2` το οποίο είναι επιτυχές, άρα το PC1 εξακολουθεί να επικοινωνεί με το PC2.

7.7

- PC1 → PC2:
  - PC1 → R1→ C1 → C2 → R2 → PC2
- PC2 → PC1:
  - PC2 → R2 → C2 → R1 → PC1

7.8

Στο PC1 εκτελούμε `traceroute 192.168.2.2`. Οι διευθύνσεις IP της διαδρομής είναι:

```
1   192.168.1.1 (R1)
2   10.0.1.2 (C1)
3   10.0.1.6 (C2)
4   10.0.2.5 (R2)
5   192.168.2.2 (PC2)
```

Παρατηρούμε ότι στην περίπτωση του C2, αντί να εμφανιστεί η διεύθυνση `10.0.0.2` από την οποία διέρχονται τα IP πακέτα του PC1, εμφανίζεται η `10.0.1.6`. Αυτό συμβαίνει διότι η traceroute καταγράφει τις διεπαφές που απαντάνε στον PC1 με μηνύματα ”Time to live exceeded”. Επειδή λοιπόν η απάντηση του C2 δρομολογείται από άλλη διεπαφή απ᾽ ό,τι τα πακέτα του PC1 -εξαιτίας της προτιμώμενης διαδρομής που έχει διαχειρστική απόσταση 1-, έχουμε αυτή τη συμπεριφορά.

7.9

Στον PC2 εκτελούμε `traceroute 192.168.1.2`. Οι διευθύνσεις IP της διαδρομής είναι:

```
1   192.168.2.1 (R2)
2   10.0.2.6 (C2)
3   10.0.1.1 (R1)
4   192.168.1.2 (PC1)
```

Παρατηρούμε ότι στην περίπτωση του R1, δεν εμφανίζεται η διεύθυνση `10.0.1.5`, αλλά η `10.0.1.1`, καθώς η απάντηση του R1 δρομολογείται μέσω του C1, επειδή η διαδρομή αυτή έχει μικρότερη απόσταση, και άρα μεγαλύτερη προτεραιότητα.

7.10

Προσομοιώνουμε βλάβη και στο WAN3, και στο PC1 εκτελούμε `traceroute 192.168.2.2`. Η έξοδος της εντολής είναι:

```
1   192.168.1.1 (R1)
2   10.0.1.2 (C1)
3   10.0.0.2 (C2)
4   10.0.2.5 (R2)
5   192.168.2.2 (PC2)
```

Ακολουθείται λοιπόν η διαδρομή: R1 $\rightarrow$ C1 $\rightarrow$ C2 $\rightarrow$ R2 $\rightarrow$ PC2.

7.11

Το ping θα ακολουθήσει τη διαδρομή:

PC1 $\rightarrow$ R1 $\rightarrow$ C1 $\rightarrow$ C2 $\rightarrow$ C1 $\rightarrow$ C2 $\rightarrow$ ... (βρόχος C1-C2)

Αυτό συμβαίνει επειδή ο C1 δεν μπορεί να προωθήσει το πακέτο μέσω του WAN2, και ο C2 δεν μπορεί να προωθήσει το πακέτο μέσω του WAN4. Έτσι, το πακέτο απλά θα επανεκπεμφθεί πολλές φορές μέχρι να μηδενιστεί το TTL του, και δεν θα φτάσει ποτέ στον προορισμό. Αντίθετα, ο PC1 θα λάβει μήνυμα "Time to live exceeded".

7.12

Το σημαντικότερο μειονέκτημα μιας τέτοιας τοπολογίας εταιρικού δικτύου είναι το διαχειριστικό overhead για τυχόν μελλοντικές αλλαγές της τοπολογίας, όπως για παράδειγμα η προσθήκη ενός επιπλέον δρομολογητή (έστω R3). Σε μια τέτοια περίπτωση, θα χρειαζόταν να ορίσουμε χειροκίνητα τον πίνακα προώθησης του R3, τυχόν διαδρομές από άλλα μηχανήματα μέσω αυτού, να ελέγξουμε μήπως υπάρχουν καλύτερες διαδρομές από τις ήδη υπάρχουσες λόγω της παρουσίας του R3, να ορίσουμε εναλλακτικές διαδρομές κλπ. Όπως είναι φανερό, κάτι τέτοιο κλιμακώνεται πολύ δύσκολα, ειδικά σε πραγματικές συνθήκες, όπου τα δίκτυα είναι πολύ μεγαλύτερα.