

<b>Όνοματεπώνυμο:</b> Ιωάννης Γιαννούκος	<b>Όνομα PC:</b> John John
<b>Ομάδα:</b> 1	<b>Ημερομηνία:</b> 25/3/2023

## Εργαστηριακή Άσκηση 3 Τοπικά δίκτυα και μεταγωγείς LAN

### Άσκηση 1

1.1) Ορίζω τις διευθύνσεις των διεπαφών των PC1 και PC2 με τις παρακάτω εντολές αντίστοιχα:

```
ifconfig em0 inet 192.168.1.1
ifconfig em0 inet 192.168.1.2
```

1.2) Ενεργοποιώ τις διεπαφές του B1 με τις εξής εντολές:

```
ifconfig em0 up
ifconfig em1 up
```

1.3) Δοκιμάζω να κάνω ping από το PC1 στο PC2 και αντίστροφα, και παρατηρώ ότι δεν υπάρχει απάντηση από κανένα μηχάνημα. Αυτό είναι λογικό, επειδή βρίσκονται σε διαφορετικά LAN (PC1 → LAN1, PC2 → LAN2) (με δικτύωση Internal Network).

1.4) Χωρίς να σταματήσω να κάνω ping από τα δύο μηχανήματα PC1, PC2, παρατηρώ ότι το μηχάνημα B1 δεν αντιλαμβάνεται καμία κίνηση ICMP πακέτων στα δύο LANs, LAN1 και LAN2.

1.5) Για να δημιουργήσω και να ενεργοποιήσω μια ψευδο-συσκευή γέφυρα bridge0 στο B1 που να περιλαμβάνει τις δύο διεπαφές του, εκτελώ τις παρακάτω εντολές:

```
ifconfig bridge0 create
ifconfig bridge0 addm em0 addm em1 up
```

1.6) Ναι, τώρα τα δύο μηχανήματα PC1 PC2 μπορούν να επικοινωνήσουν με τη βοήθεια της γέφυρας bridge0.

1.7) Επειδή η τιμή TTL των ICMP πακέτων που στέλνονται με τα pings είναι 64, συμπεραίνω ότι τα μηχανήματα απέχουν 1 βήμα, δηλαδή η γέφυρα δεν προσμετράται στα hops των πακέτων.

1.8) Με την εντολή “**arp -a**” εκτυπώνω τον πίνακα arp των PCs και βλέπω ότι οι μόνες διευθύνσεις που είναι αποθηκευμένες είναι των PCs (και όχι της γέφυρας).

1.9) Εκτελώντας τις εντολές “**tcpdump -i em0 -vvn -l**” και “**tcpdump -i em1 -vvn -l**” σε δύο κονσόλες του μηχανήματος B1 (πατώντας Alt+F1 , Alt+F2), και κάνοντας ping από το PC1 στο PC2 και αντίστροφα, παρατηρώ ότι το B1 αντιλαμβάνεται την κίνηση μεταξύ των δύο LANs και ότι προωθεί τα πακέτα μεταξύ των.

1.10) Όχι, το B1 δεν αλλάζει τις διευθύνσεις MAC ή τις διευθύνσεις IPv4 των πακέτων που προωθεί.

1.11) Όχι, το B1 δεν φαίνεται να αλλάζει οποιοδήποτε πεδίο στα πακέτα που προωθεί μεταξύ των δύο LANs.

1.12) Εκτελώντας την εντολή “**traceroute -m 1 192.168.1.2**” από το PC1 (κάνοντας traceroute στο PC2) το PC2 απαντάει στο πακέτο που στέλνεται. Επομένως, αφού με τιμή TTL = 1 το πακέτο έφτασε στο PC2, αντιλαμβάνομαι ότι δεν υπάρχει καμία ανάμειξη του B1 στην διαδικασία. Διαφορετικά, το πακέτο δεν θα μπορούσε να φτάσει στο PC2, αφού θα καταναλωνόταν ένα βήμα για να φτάσει το πακέτο στην γέφυρα.

1.13) Κάνω ping από το PC1 στο PC2 και ξεκινώ μια καταγραφή από το B1 στο LAN2 με τις παρακάτω αντίστοιχες εντολές:

```
ping 192.168.1.2  
tcpdump -i em1 -vvn -l
```

1.14) Αλλάζοντας την διεύθυνση του PC2 σε 192.168.2.1, η γέφυρα προωθεί τα πακέτα που λαμβάνει από το PC1 στο LAN2, καθώς ο πίνακας προώθησης της γέφυρας δεν έχει ανανεωθεί ακόμα, και η καταγραφή για την διεύθυνση 192.168.1.2 υπάρχει ακόμα. (Την διεύθυνση του PC2 την άλλαξα με την εντολή “**ifconfig em0 inet 192.168.2.1**”).

1.15) Όχι. Παρόλο που η γέφυρα συνεχίζει να προωθεί τα πακέτα που λαμβάνει στο LAN2, δεν συμβαίνει καμία απάντηση, επομένως το ping δεν είναι επιτυχές.

Προσθέτω το μηχάνημα PC3 και θέτω την IPv4 διεύθυνσή του με την εξής εντολή: “**ifconfig em0 inet 192.168.1.3**”.

1.16) Όχι, δεν μπορώ να κάνω ping από το PC3 στο PC1, καθώς η διεπαφή em2 (αυτή που συνδέεται με το LAN3) δεν έχει προστεθεί στην γέφυρα bridge0.

1.17) Προσθέτω και ενεργοποιώ την διεπαφή em2 στην ψευδο-συσκευή bridge0 με τις εξής εντολές:

```
"ifconfig bridge0 addm em2"
```

```
"ifconfig em2 up"
```

1.18) Ναι, πλέον το ping από το PC3 στο PC1 επιτυγχάνει.

1.19) Όχι, κάνοντας ping από το PC1 στο PC3 και αντίστροφα, δεν καταγράφεται κίνηση πακέτων στο LAN2 (το βλέπω αυτό εκτελώντας την εντολή `"tcpdump -i em1 -vvn -l"` στο B1).

1.20) Έχοντας εκκαθαρίσει τον πίνακα arp των PC1 και PC3 και κάνοντας ping από το PC1 στο PC3, καταγράφεται ένα μόνο πακέτο στο LAN2 (κάνοντας από το B1 `"tcpdump -i em1 -vvn -l"`). Το πακέτο αυτό στέλνεται μόνο μία φορά, κατά την εκκίνηση του ping, με το οποίο το PC1 «ρωτά» τα υπόλοιπα μηχανήματα ποιος έχει την διεύθυνση 192.168.1.3 (και «ζητά» να σταλθεί απάντηση της ερώτησης στην διεύθυνση 192.168.1.1). (Με την εντολή `"arp -a -d"` μπορούμε να σβήσουμε όλες τις εγγραφές στους πίνακες arp των PC{1,3}).

1.21) Μπορώ να δω τις διεπαφές της γέφυρας bridge0 με την εξής εντολή:  
`"ifconfig bridge0"`

1.22) Μπορώ να δω τα περιεχόμενα του πίνακα προώθησης της γέφυρας bridge0 με την εξής εντολή: `"ifconfig bridge0 addr"`.

1.23) Οι διευθύνσεις MAC που περιέχονται στον πίνακα προώθησης της γέφυρας bridge0 αντιστοιχούν σε αυτές των μηχανημάτων PC1 και PC3.

1.24) Διαγράφω τα περιεχόμενα του πίνακα προώθησης της γέφυρας με την εντολή: `"ifconfig bridge0 flush"`.

1.25) Αφαιρώ από τη γέφυρα bridge0 τη διεπαφή της στο LAN3 με την εντολή `"ifconfig bridge0 deletem em2"`.

1.26) Καταστρέφω την γέφυρα bridge0 με την εντολή `"ifconfig bridge0 destroy"`.

1.27) Αφαιρώ τις διευθύνσεις IP από τις διεπαφές των PC{1,2,3} με τις εντολές: “ifconfig em0 192.168.1.i remove”. (i = 1,2,3)

## Άσκηση 2

2.1) Ορίζω διεύθυνση IPv4 στην διεπαφή του μηχανήματος PC<sub>i</sub>, i = 1,2,3,4, με τις εξής αντίστοιχες εντολές:

```
“ifconfig em0 inet 192.168.1.1”,  
“ifconfig em0 inet 192.168.1.2”,  
“ifconfig em0 inet 192.168.1.3”,  
“ifconfig em0 inet 192.168.1.4”
```

2.2) Δημιουργώ μια ψευδο-συσσκευή γέφυρα bridge1 στο B1 που να περιλαμβάνει τις διεπαφές στα LAN1 και LNK1 και ενεργοποιώ τις διεπαφές με τις εξής 4 εντολές:

```
ifconfig bridge1 create  
ifconfig em0 up  
ifconfig em1 up  
ifconfig bridge1 addm em0 addm em1 up
```

2.3) Δημιουργώ μια ψευδο-συσσκευή γέφυρα bridge2 στο B2 που να περιλαμβάνει τις διεπαφές στα LNK1 και LNK2 και ενεργοποιώ τις διεπαφές με τις εξής 4 εντολές:

```
ifconfig bridge2 create  
ifconfig em0 up  
ifconfig em1 up  
ifconfig bridge2 addm em0 addm em1 up
```

2.4) Δημιουργώ μια ψευδο-συσσκευή γέφυρα bridge3 στο B3 που να περιλαμβάνει τις διεπαφές στα LNK2 και LAN2 και ενεργοποιώ τις διεπαφές με τις εξής 4 εντολές:

```
ifconfig bridge3 create  
ifconfig em0 up; ifconfig em1 up  
ifconfig bridge3 addm em0 addm em1 up
```

2.5) Αδειάζω του πίνακες ARP των PC{1,2,3,4} με την εντολή: “arp -a -d”.

<i>Διευθύνσεις MAC των PC{1,2,3,4}</i>	
<b>PC1</b>	08:00:27:d4:9d:13
<b>PC2</b>	08:00:27:fe:57:4c
<b>PC3</b>	08:00:27:21:df:54
<b>PC4</b>	08:00:27:9e:66:2f

2.6) Διαγράφω τις εγγραφές του πίνακα προώθησης των B{1,2,3} με τις εξής αντίστοιχες εντολές: “ifconfig bridgeX flush”, X = 1,2,3.

2.7) Ξεκινώ καταγραφές με tcpdump στα PC{1,2,3,4} με την εξής εντολή: “tcpdump -i em0 -vvn -l”.

2.8) Βεβαιώνομαι ότι οι πίνακες προώθησης των ψευδο-συσκευών γεφυρών είναι κενοί εκτελώντας τις εξής 2 εντολές σε κάθε μηχανήμα B{1,2,3}: “ifconfig bridgeX flush”, “ifconfig bridgeX addr”, X = 1,2,3.

Ανοίγοντας νέο παράθυρο στο PC1 (Alt+F2) εκτελώ την εντολή “ping -c 1 192.168.1.2”.

<i>Περιεχόμενα πινάκων προώθησης B{1,2,3}</i>	
<b>Πιν. Προώθ. B1</b>	08:00:27:d4:9d:13 ( <b>PC1</b> ), 08:00:27:fe:57:4c ( <b>PC2</b> )
<b>Πιν. Προώθ. B2</b>	08:00:27:d4:9d:13 ( <b>PC1</b> ), 08:00:27:fe:57:4c ( <b>PC2</b> )
<b>Πιν. Προώθ. B3</b>	08:00:27:d4:9d:13 ( <b>PC1</b> )

2.9) Αρχικά, το PC1 θα στείλει πακέτο ARP request ρωτώντας ποιος έχει την διεύθυνση 192.168.1.2. Το μόνο μηχανήμα που θα απαντήσει στέλνοντας πακέτο ARP reply είναι το PC2. Έτσι, έπειτα από την παραπάνω διαδικασία, οι γέφυρες bridge1, bridge2, bridge3 θα έχουν αποθηκεύσει στον πίνακα προώθησης την διεύθυνση του PC1 και οι γέφυρες bridge1, bridge2 θα έχουν αποθηκεύσει στον πίνακα προώθησης την διεύθυνση του PC2.

Στη συνέχεια, τα πακέτα ICMP echo request/reply προωθήθηκαν κατευθείαν στα μηχανήματα PC1, PC2, αφού οι πίνακες προώθησης των γεφυρών των B1, B2 ήταν πλέον ενημερωμένοι.

(Παρατηρώ ότι όλες οι γέφυρες έχουν αποθηκεύσει στους πίνακες προώθησής τους και τις διευθύνσεις των PC3, PC4. Ωστόσο, αυτό δεν πρέπει να συμβαίνει και υπάρχει κάποιο λάθος στις ρυθμίσεις του Virtual Box).

2.10) Όχι, δεν υπήρξαν αλλαγές στους πίνακες προώθησης των γεφυρών των B1, B2, αφού οι διευθύνσεις των PC1, PC2 είναι ήδη αποθηκευμένοι από την διαδικασία του ερωτήματος (2.9).

2.11) Κάνω ping από το PC2 στο PC4 με την εντολή “ping -c 1 192.168.1.4”. Παρατηρώ ότι ο πίνακας της γέφυρας του B1 περιέχει την διεύθυνση του PC4. Αυτό συμβαίνει επειδή η απάντηση ICMP echo reply που στάλθηκε από το PC4 με προορισμό το PC2, όταν στάλθηκε στο LAN

“LNK1” διαβάστηκε από την διεπαφή em1 της γέφυρας του B1. Παρόλο, που η γέφυρα αυτή δεν το προώθησε στο LAN1, αποθήκευσε την διεύθυνση του PC4 στον πίνακα προώθησής του.

<i>Περιεχόμενα πινάκων προώθησης B{1,2,3}</i>	
<b>Πιν. Προώθ. B1</b>	08:00:27:d4:9d:13 ( <b>PC1</b> ), 08:00:27:fe:57:4c ( <b>PC2</b> ), 08:00:27:9e:66:2f ( <b>PC4</b> )
<b>Πιν. Προώθ. B2</b>	08:00:27:d4:9d:13 ( <b>PC1</b> ), 08:00:27:fe:57:4c ( <b>PC2</b> ), 08:00:27:9e:66:2f ( <b>PC4</b> )
<b>Πιν. Προώθ. B3</b>	08:00:27:d4:9d:13 ( <b>PC1</b> ), 08:00:27:fe:57:4c ( <b>PC2</b> ), 08:00:27:9e:66:2f ( <b>PC4</b> )

2.12) Κάνω ping από το PC3 στο PC2 με την εντολή “ping -c 1 192.168.1.2”.

<i>Περιεχόμενα πινάκων προώθησης B{1,2,3}</i>	
<b>Πιν. Προώθ. B1</b>	08:00:27:d4:9d:13 ( <b>PC1</b> ), 08:00:27:fe:57:4c ( <b>PC2</b> ), 08:00:27:21:df:54 ( <b>PC3</b> ), 08:00:27:9e:66:2f ( <b>PC4</b> )
<b>Πιν. Προώθ. B2</b>	08:00:27:d4:9d:13 ( <b>PC1</b> ), 08:00:27:fe:57:4c ( <b>PC2</b> ), 08:00:27:21:df:54 ( <b>PC3</b> ), 08:00:27:9e:66:2f ( <b>PC4</b> )
<b>Πιν. Προώθ. B3</b>	08:00:27:d4:9d:13 ( <b>PC1</b> ), 08:00:27:fe:57:4c ( <b>PC2</b> ), 08:00:27:21:df:54 ( <b>PC3</b> ), 08:00:27:9e:66:2f ( <b>PC4</b> )

Επειδή αρχικά το PC3 δεν γνωρίζει την διεύθυνση του PC2, στέλνει πακέτο ARP request για να την μάθει. Το πακέτο αυτό «ακούγεται» από τις γέφυρες των B2, B3, αυτή του B2 το προωθεί στο LNK1 και το ανιχνεύει και η γέφυρα του B1. Με αυτόν τον τρόπο, όλες οι γέφυρες θα έχουν αποθηκευμένοι πλέον στους πίνακες προώθησής τους την διεύθυνση του PC3.

2.13) Κάνω ping από τα PC1, PC4 στο PC2 και αφήνω να «τρέχουν» με την εντολή “ping 192.168.1.2”.

2.14) Αφού μετακινήσω το PC2 από το LNK1 στο LAN2, όπου βρίσκεται το PC4, το ping από το PC4 στο PC2 συνεχίζει να επιτυγχάνει, αφού τα δύο μηχανήματα βρίσκονται πλέον στο ίδιο τοπικό δίκτυο.

2.15) Αφού μετακινήσω το PC2 από το LNK1 στο LAN2, όπου δεν βρίσκεται το PC1, το ping από το PC1 στο PC2 αποτυγχάνει. Αυτό συμβαίνει επειδή οι γέφυρες προώθησης δεν είναι ενημερωμένες με την νέα θέση του PC2, και έτσι η γέφυρα του B3 δεν προωθεί το ICMP πακέτο από το LNK2 στο LAN2, εκεί δηλαδή που βρίσκεται πλέον το PC2.

2.16) Κάνω ping από το PC2 στο PC3 με την εντολή “`ping -c 1 192.168.1.3`”. Παρατηρώ τώρα ότι το ping από το PC1 πλέον επιτυγχάνει. Αυτό συμβαίνει επειδή οι πίνακες προώθησης των γεφυρών ενημερώθηκαν με το ping που έγινε από το PC2 στο PC3.

2.17) Εάν δεν έκανα ping από το PC2 στο PC3, η τοποθεσία του PC2 θα έμενε ανενημέρωτη μέχρι οι πίνακες προώθησης να «έληγαν», δηλαδή στην προκειμένη περίπτωση που χρησιμοποιούμε λειτουργικό FreeBSD, θα έπρεπε να περάσουν 20 λεπτά.

### Άσκηση 3

3.1) Δημιουργώ και ενεργοποιώ την γέφυρα bridge1 στο B1 που περιλαμβάνει τις διεπαφές του στα LAN1 και LNK1, ενεργοποιώντας τις με τις εξής εντολές:

“`ifconfig bridge1 create`”

“`ifconfig bridge1 addm em0 addm em1 up`”

3.2) Δημιουργώ και ενεργοποιώ την γέφυρα bridge2 στο B2 που περιλαμβάνει τις διεπαφές του στα LAN2 και LNK1, ενεργοποιώντας τις με τις εξής εντολές:

“`ifconfig bridge2 create`”

“`ifconfig bridge2 addm em0 addm em2 up`”

3.3) Αδειάζω τους πίνακες ARP των PC{1,2,3} με την εντολή “`arp -a -d`”.

<i>Διευθύνσεις MAC των PC{1,2,3}</i>	
<b>PC1</b>	08:00:27:d4:9d:13
<b>PC2</b>	08:00:27:fe:57:4c
<b>PC3</b>	08:00:27:21:df:54

3.4) Ναι, κάνοντας ping από το PC2 στο PC3 το PC1 αντιλαμβάνεται κίνηση, ειδικότερα λαμβάνει το πακέτο ARP request που στέλνει το PC2 για να μάθει την διεύθυνση του PC3. Σε αυτό το σημείο, όλες οι γέφυρες έχουν κενούς πίνακες προώθησης, επομένως το πακέτο λογικώς στέλνεται σε όλα τα LANs.

3.5) Ξεκινώ ένα ping από το PC3 στο PC1 και το αφήνω να «τρέχει» με την εντολή “ping 192.168.1.1”.

3.6) Προσθέτω στις γέφυρες bridge1, bridge2 τις διεπαφές τους στο LNK2 και τις ενεργοποιώ με τις εντολές, αντίστοιχα:  
“ifconfig em2 up” “ifconfig bridge1 addm em2”,  
“ifconfig em1 up” “ifconfig bridge2 addm em1”.

3.7) Αφού σταματήσω το ping από το PC3 στο PC1, βλέπω ότι το περιεχόμενο των πινάκων προώθησης των δύο γεφυρών bridge1, bridge2 περιέχουν τις διευθύνσεις των PC{1,2,3}. Αυτό είναι λογικό, επειδή στις προηγούμενες 2 διαδικασίες έστειλαν πακέτα και τα 3 αυτά μηχανήματα. (Το περιεχόμενο των πινάκων προώθησης εμφανίζεται με την εντολή “ifconfig bridgeX addr”, X = 1,2).

3.8) Οι διευθύνσεις των PC{1,3} εμφανίζονται στις διεπαφές των B{1,2} ως εξής:  
B1 → PC1: em0, PC3: em0  
B2 → PC1: em0, PC3: em2

3.9) Ξεκινώ καταγραφές στα PC{1,2} με την εντολή “tcpdump -vvn -l”.

3.10) Καθαρίζω τον πίνακα ARP του PC3 με την εντολή “arp -a -d”. Κάνοντας ping από το PC3 στο PC1 με την εντολή “ping -c 1 192.168.1.1” βλέπω ότι το ping είναι επιτυχές.

3.11) Τα PC{1,3} εμφανίζονται στις διεπαφές em0, em2 του B2, αντίστοιχα.

3.12) Στον κατακλυσμό που γίνεται από ARP πακέτα, η ερώτηση που γίνεται στέλνεται για να μαθευτεί η διεύθυνση του PC1 και η απάντηση που δίδεται στέλνεται για να απαντηθεί η διεύθυνση του PC1.

3.13) Στην καταγραφή στο PC2 παρατηρώ έναν κατακλυσμό από ARP requests. Η διεύθυνση MAC πηγής τους είναι αυτή του PC3.



3.14) Τα πακέτα ARP request επαναλαμβάνονται συνεχώς στις δύο καταγραφές, επειδή δημιουργήσαμε flooding μεταξύ των δύο γεφυρών.

3.15) Τα πακέτα ARP reply δεν προωθούνται στο LAN2. Αυτό συμβαίνει επειδή η γέφυρα B2 έχει αποθηκευμένη στον πίνακα προώθησής της την διεύθυνση του PC3.

#### **Άσκηση 4**

4.1) Καταστρέφω τις γέφυρες bridge{1,2} με την εντολή “**ifconfig bridgeX destroy**”, X = 1,2.

Απενεργοποιώ τις διεπαφές με την εντολή “**ifconfig emX down**”, X = 0,1,2.

Ξαναδημιουργώ τις γέφυρες με την εντολή “**ifconfig bridgeX create**”, X = 1,2.

4.2) Ενεργοποιώ όλες τις κάρτες δικτύου στο B1 με την εντολή “**ifconfig emX up**”, X = 0,1,2.

Δημιουργώ μια ψευδο-συσσκευή συνάθροισης lag0 με την εντολή “**ifconfig lag0 create**”.

4.3) Εντάσσω στην ψευδο-συσσκευή lag0 τις διεπαφές της B1 στα LNK1, LNK2 και την ενεργοποιώ με την εντολή “**ifconfig lag0 up lag0port em1 lag0port em2**”.

4.4) Εκτελώ τις ίδιες εντολές με τα υποερωτήματα (4.2), (4.3).

4.5) Εντάσσω στη γέφυρα που δημιούργησα στο B1 τη διεπαφή στο LAN1 και την ψευδο-συσσκευή lag0 και ενεργοποιώ την γέφυρα με την εντολή “**ifconfig bridge1 addm em0 addm lag0 up**”.

4.6) Επαναλαμβάνω την παραπάνω διαδικασία με την εντολή “**ifconfig bridge2 addm em2 addm lag0 up**”.

4.7) Ναι, εμφανίζεται κίνηση στο LAN1, δηλαδή ανιχνεύεται το πακέτο ARP request που στέλνει το PC2 στο PC3 στην αρχή του ping.

4.8) Ξεκινώ μια καταγραφή στο PC1 με την εντολή “**ping -vvv -l**”.

4.9) Εκκαθαρίζω τον πίνακα ARP του PC3 με την εντολή “**arp -a -d**” και κάνω ping στο PC1 με την εντολή “**ping -c 1 192.168.1.1**”.

Παρατηρώ ότι στέλνονται κανονικά τα πακέτα ARP και ICMP από το PC1 στο PC3.

4.10) Ξεκινώ καταγραφές στην διεπαφή του B1 στο LNK1 και στην διεπαφή του B2 στο LNK2 με την εντολή “`tcpdump -em1 -vnn -l`”. Κάνοντας ring από το PC2 στο PC1 παρατηρώ ότι η κίνηση γίνεται από το δίκτυο LNK1. Αυτό οφείλεται στο γεγονός ότι από προεπιλογή εφαρμόζεται το πρωτόκολλο συνάθροισης “failover”, με το οποίο από τις δύο ζεύξεις η μία ορίζεται ως master και η άλλη ως backup.

4.11) Αφού «αποσυνδέσω» τα καλώδια από τις κάρτες δικτύου των B1, B2 στην ζεύξη LNK1 (η ζεύξη master), παρατηρώ ότι η κίνηση τώρα διέρχεται από την ζεύξη LNK2. Αυτό είναι αναμενόμενο, αφού εξ αρχής δημιουργήσαμε την συνδεσμολογία αυτή για να μην «πέσει» η σύνδεση σε περίπτωση βλάβης της ζεύξης master (στην προκειμένη περίπτωση η αποσύνδεση των καλωδίων).

4.12) Ναι, πλέον η κίνηση διέρχεται και πάλι από την ζεύξη master, δηλαδή την LNK1.

## Άσκηση 5

5.1) Καταστρέφω τις γέφυρες και τις ψευδο-συσκευές συνάθροισης και απενεργοποιώ τις κάρτες δικτύου στα B{1,2} με τις εξής εντολές:

```
ifconfig bridgeX destroy
ifconfig lagg0 destroy
ifconfig emX down
```

5.2) Δημιουργώ την γέφυρα bridge1 στο B1 που περιλαμβάνει τις διεπαφές του B1 στα LAN1, LNK1, LNK2 και την ενεργοποιώ, καθώς και όλες τις διεπαφές με τις εξής εντολές:

```
ifconfig emX up
ifconfig bridge1 create
ifconfig bridge1 addm em0 addm em1 addm em2 up
```

5.3) Επαναλαμβάνω την διαδικασία του (5.2) για την γέφυρα bridge2 με τις εντολές:

```
ifconfig emX up
ifconfig bridge2 create
ifconfig bridge2 addm em0 addm em1 addm em2 up
```

5.4) Ενεργοποιώ το STP σε όλες τις διεπαφές που μετέχουν στην bridge1 με την εντολή `ifconfig bridge1 stp em0 stp em1 stp em2`.

5.5) Ενεργοποιώ το STP σε όλες τις διεπαφές που μετέχουν στην bridge2 με την εντολή `ifconfig bridge1 stp em0 stp em1 stp em2`.

5.6) Εκτελώντας την εντολή “`ifconfig bridgeX`” (X = 1,2) βλέπω τις τιμές του πεδίου id, οι οποίες είναι:

bridge1: 08:00:27:01:0d:26

bridge2: 08:00:27:52:a8:6b

5.7) Από την εκτέλεση της παραπάνω εντολής, φαίνεται ότι στο πεδίο “root id” υπάρχει το id της γέφυρας bridge1.

5.8) Δεν υπάρχει κάποια διάκριση στις τρεις θύρες της γέφυρας ρίζας (bridge1), καθώς όλες έχουν τον ρόλο των designated ports και βρίσκονται στην κατάσταση forwarding.

5.9) Η ριζική θύρα της γέφυρας bridge2 είναι αυτή που συνδέεται στην ζεύξη LNK1 (em0).

5.10) Η θύρα που δεν είναι ριζική και συνδέεται στην ζεύξη LNK2 έχει ρόλο alternate και βρίσκεται στην κατάσταση discarding.

5.11) Η θύρα της γέφυρας bridge2 που συνδέεται στην ζεύξη LAN2 έχει ρόλο designated port και βρίσκεται στην κατάσταση forwarding.

5.12) Ξεκινώ μία καταγραφή από την διεπαφή της bridge1 που συνδέεται στο LAN1 (em0) και παρατηρώ ότι εκπέμπονται BPDUs κάθε 2 seconds.

5.13) Χρησιμοποιείται ενθυλάκωση IEEE 802.3.

5.14) Η διεύθυνση MAC πηγής είναι η διεύθυνση MAC της γέφυρας bridge1 (root) (08:00:27:01:0d:26) και η διεύθυνση MAC προορισμού είναι η διεύθυνση 01:80:c2:00:00:00.

5.15) Η διεύθυνση 08:00:27:01:0d:26 ανήκει στην διεπαφή της γέφυρας ρίζας που συνδέεται στην ζεύξη LNK1.

5.16) Εφόσον η διεύθυνση προορισμού έχει απλώς μηδενικά στο τέλος της, συμπεραίνω ότι το είδος της είναι multicast (πολλαπλή διανομή).

5.17) Στα πλαίσια που καταγράφηκαν οι τιμές των πεδίων root id, bridge id και rootpathcost είναι οι εξής:

root id: 08:00:27:01:0d:26

bridge id: 08:00:27:01:0d:26.8001

root path cost: 0

5.18) Ξεκινώ μία καταγραφή αντίστοιχης αυτής του (5.12) με την εντολή “tcpdump -i em1 -e -vvn -l”. Τα τελευταία 4 ψηφία του Bridge ID αντιστοιχούν στην προτεραιότητα. Βλέπουμε ότι στα πακέτα αυτά υπάρχει επίθεμα “8002”.

5.19) Το 2<sup>ο</sup> μέρος της Bridge ID αντιστοιχεί στην θύρα από την οποία εκπέμπεται το πλαίσιο BPDU, και το 3<sup>ο</sup> μέρος αντιστοιχεί στην προτεραιότητα.

5.20) Όχι, δεν παρατηρείται κίνηση στις ζεύξεις LNK{1,2} από την γέφυρα bridge2.

5.21) Κίνηση BPDU παρατηρώ στην θύρα της γέφυρας bridge2 που συνδέεται στο LAN2 (em2).

5.22) Στα BPDU που καταγράφηκαν από την γέφυρα bridge2 στην διεπαφή em2 φαίνονται οι εξής τιμές στα αντίστοιχα πεδία:

Root id: 8000. 08:00:27:01:0d:26

Bridge id: 8000.08:00:27:52:a8:6b.8003

Root path cost: 20000

5.23) Ναι, το ping που κάνω από το PC1 στο PC2 είναι επιτυχές.

5.24) Για να αποκατασταθεί η επικοινωνία μεταξύ των PC{1,2} χρειάζονται περίπου 6 δευτερόλεπτα (δηλαδή 3 x 2). Αυτό είναι αναμενόμενο, καθώς τόσοι είναι ο χρόνος που χρειάζεται το RSTP για να προσαρμοστεί σε οποιαδήποτε αλλαγή στο δίκτυο.

5.25) Όχι, τώρα δεν υπάρχει διακοπή.

## **Άσκηση 6**

(Για τα παρακάτω υποερωτήματα σιγουρεύομαι ότι σε όλες τις γέφυρες έχουν προστεθεί όλες οι διεπαφές, οι οποίες είναι ενεργοποιημένες και με STP και στα σωστά LAN, και ότι όλες οι γέφυρες είναι ενεργοποιημένες).

6.1) Ενεργοποιώ την διεπαφή του B1 στο LNK3 και την προσθέτω στη bridge1 με τις εντολές

```
"ifconfig em3 up",  
"ifconfig bridge1 create",  
"ifconfig bridge1 addm em3",  
"ifconfig bridge1 stp em3",  
"ifconfig bridge1 up"
```

6.2) Ενεργοποιώ την διεπαφή του B2 στο LNK4 και την προσθέτω στη bridge1 με τις εντολές

```
"ifconfig em3 up",  
"ifconfig bridge2 create",  
"ifconfig bridge2 addm em3",  
"ifconfig bridge2 stp em3",  
"ifconfig bridge2 up"
```

6.3) Δημιουργώ την bridge3 στο B3 με την εντολή "ifconfig bridge3 create". Ενεργοποιώ τις διεπαφές του B3 στα LAN3, LNK3, LNK4 και τις προσθέτω στην γέφυρα bridge3 με τις εντολές

```
"ifconfig emX up", (X = 0,1,2)  
"ifconfig bridge3 addm em0 addm em1 addm em2 up",  
"ifconfig bridge3 stp em0 stp em1 stp em2 up",  
"ifconfig bridge3 up".
```

6.4) Διαγράφω τους πίνακες προώθησης όλων των γεφυρών με την εντολή "ifconfig bridgeX flush" (X = 1,2,3). Έπειτα κάνω ping από το PC1 στα PC{2,3} και βλέπω ότι είναι επιτυχή.

6.5) Ορίζω την bridge1 ως ριζική γέφυρα με την εντολή "ifconfig bridge1 priority 0".

6.6) Οι ζεύξεις LNK{1,2,4} στο B2 έχουν τιμή 20000 στο πεδίο "path cost".

6.7) Η γέφυρα bridge3 λαμβάνει κόστος 0 από την γέφυρα bridge1 (λογικό αφού η bridge1 είναι η ρίζα) και κόστος 20000 από την γέφυρα bridge2, αφού για να φτάσει ένα πακέτο στην ρίζα από την bridge2 θα χρειαστεί να διανύσει και την «απόσταση» του LNK1.

6.8) Η ριζική θύρα της γέφυρας bridge3 είναι αυτή που συνδέεται στο LNK3 (em1). Αυτό δικαιολογείται στο γεγονός ότι αν χρησιμοποιηθεί η ζεύξη στο LNK3 ελαχιστοποιείται η απόσταση μέχρι την γέφυρα ρίζα.

6.9) Η θύρα στο LNK4 (em2) του B3 έχει ρόλο alternative και βρίσκεται σε κατάσταση forwarding. Η αντίστοιχη θύρα της γέφυρας του B2 στο LNK4 (em3) έχει ρόλο alternate και βρίσκεται σε κατάσταση discarding. (Τα προηγούμενα φαίνονται με την εντολή “`ifconfig bridgeX`”).

6.10) Το root path cost στα πλαίσια BPDU που παράγει η bridge3 στο LAN3 είναι 20000.

6.11) Ξεκινώ ένα ping από το PC1 στο PC3 και το αφήνω να «τρέχει» με την εντολή “`ping 192.168.1.3`”.

6.12) Θέτω το κόστος της διεπαφής της bridge3 στο LNK3 (em1) στην τιμή 45000, ώστε η θύρα στο LNK4 (em2) να γίνει ριζική. Για αυτό εκτελώ την εντολή “`ifconfig bridge3 ifpathcost em1 45000`”.

6.13) Για να αποκατασταθεί η επικοινωνία πέρασαν περίπου 2 δευτερόλεπτα (αναμενόμενο, αφού ο χρόνος hellotime είναι 2).

6.14) Η θύρα της bridge3 στο LNK3 (em1) έχει ρόλο alternate και βρίσκεται σε κατάσταση discarding. Η θύρα της bridge2 στο LNK4 (em3) έχει ρόλο designated και βρίσκεται σε κατάσταση forwarding.

6.15) Όχι, δεν φαίνεται να υπήρξε κάποια διαφορά στις τιμές των παραμέτρων των BPDU που λαμβάνει η bridge3.

6.16) Ναι, φαίνεται ότι η τιμή του root path cost της bridge3 έχει πλέον την τιμή 40000.

6.17) Αφού «αποσύνδεσα» το καλώδιο της θύρας στο LNK4 (em3) της bridge2, χρειάστηκαν περίπου 6 δευτερόλεπτα ( $3 \times \text{hellotime} = 3 \times 2$ ) μέχρι να αποκατασταθεί ξανά η επικοινωνία.

6.18) Αφού επανασυνέδεσα το καλώδιο που αποσυνέδεσα στο προηγούμενο ερώτημα χρειάστηκαν περίπου 2 δευτερόλεπτα ( $\text{hellotime} = 2$ ) έως ότου αποκατασταθεί ξανά η επικοινωνία.

6.19) Συνδέω την 4<sup>η</sup> διεπαφή του B3 στο LAN3, την ενεργοποιώ, την προσθέτω στην bridge3 και ενεργοποιώ για αυτήν το STP με τις παρακάτω εντολές:

```
“ifconfig em3 up”,
```

```
“ifconfig bridge3 addm em3”
```

```
“ifconfig bridge3 stp em3”
```

Παρατηρώ ότι η διεπαφή που μόλις προσέθεσα έχει ρόλο backup και βρίσκεται σε κατάσταση discarding, ενώ η διεπαφή που υπήρχε από πριν έχει ρόλο designated και βρίσκεται σε κατάσταση forwarding.

6.20) Αποσυνδέω την 4<sup>η</sup> διεπαφή του B3 και θέτω το κόστος της διεπαφής της bridge3 στο LNK3 ίσο με 20000, ώστε αυτή να γίνει η ριζική θύρα.

## Άσκηση 7

7.1) Στο PC1 δημιουργώ με βάση την em0 δύο νέες διεπαφές: η μία ανήκει στο VLAN 5 με διεύθυνση 192.168.5.1 και η άλλη στο VLAN 6 με διεύθυνση 192.168.6.1. Τις δημιουργώ με τις εξής εντολές:

1<sup>η</sup>: `“ifconfig em0.5 create;  
ifconfig em0.5 inet 192.168.5.1 netmask 255.255.255.0;  
ifconfig em0.5 vlan 5 vlandev em0”`  
2<sup>η</sup>: `“ifconfig em0.6 create;  
ifconfig em0.6 inet 192.168.6.1 netmask 255.255.255.0;  
ifconfig em0.6 vlan 6 vlandev em0”`

7.2) Στο B1 δημιουργώ δύο νέες διεπαφές με βάση αυτήν στο LAN1 (em0), μία στο VLAN 5 και μία στο VLAN 6, με τις εξής εντολές:

1<sup>η</sup>: `“ifconfig em0.5 create; ifconfig em0.5 vlan 5 vlandev em0”`  
2<sup>η</sup>: `“ifconfig em0.6 create; ifconfig em0.6 vlan 6 vlandev em0”`

7.3) Στο B1 δημιουργώ μία νέα διεπαφή με βάση αυτήν στο LNK1 (em1) στο VLAN 6 με την εντολή

`“ifconfig em1.6 create; ifconfig em1.6 vlan 6 vlandev em1”`.  
Επίσης, δημιουργώ μία νέα διεπαφή με βάση αυτήν στο LNK3 (em3) στο VLAN 5 με την εντολή  
`“ifconfig em3.5 create; ifconfig em3.5 vlan 5 vlandev em3”`.

7.4) Στο PC2 δημιουργώ με βάση την em0 μία νέα διεπαφή στο VLAN 6 με διεύθυνση 192.168.6.2/24 με τις εξής εντολές:

`“ifconfig em0.6 create;  
ifconfig em0.6 inet 192.168.6.2 netmask 255.255.255.0;  
ifconfig em0.6 vlan 6 vlandev em0”`

7.5) Στο B2 δημιουργώ με βάση τις διεπαφές του B2 στα LAN2 (em0), LNK1 (em1) νέες διεπαφές για το VLAN 6 με τις εξής εντολές:

1<sup>η</sup>: `“ifconfig em0.6 create; ifconfig em0.6 vlan 6 vlandev em0”`  
2<sup>η</sup>: `“ifconfig em1.6 create; ifconfig em1.6 vlan 6 vlandev em0”`

7.6) Στο PC3 δημιουργώ νέα διεπαφή στο VLAN 5 με διεύθυνση 192.168.5.3/24 με τις εξής εντολές:

```
“ifconfig em0.5 create;  
ifconfig em0.5 inet 192.168.5.3 netmask 255.255.255.0;  
ifconfig em0.5 vlan 5 vlandev em0”
```

7.7) Στο B3 δημιουργώ νέες διεπαφές για το VLAN 5 στα LAN3 (em0), LNK3 (em1) με τις εξής εντολές:

1<sup>η</sup>: “ifconfig em0.5 create; ifconfig em0.5 vlan 5 vlandev em0”

2<sup>η</sup>: “ifconfig em1.5 create; ifconfig em1.5 vlan 5 vlandev em1”

7.8) Ναι, τα pings από το PC1 στις διευθύνσεις 192.168.6.2 και 192.168.5.3 είναι επιτυχή (τα εκτελώ με τις εντολές “ping 192.168.6.2” και “ping 192.168.5.3”).

7.9) Αφαιρώ από το επικαλύπτον δένδρο τη διεπαφή της γέφυρας bridge1 στο LAN1 με την εντολή “ifconfig bridge1 -stp em0”.

7.10) Ξεκινώ μια λεπτομερή καταγραφή στη διεπαφή em0 ώστε να εμφανίζονται οι επικεφαλίδες Ethernet, καθώς και το περιεχόμενο των πλαισίων σε δεκαεξαδική μορφή (αποθηκεύοντάς την στο αρχείο file1.txt) με την εντολή “tcpdump -i em0 -e -vvn -x -l | tee file1.txt”.

7.11) Καθαρίζω τον πίνακα ARP στο PC2 με την εντολή “arp -a -d” και εκτελώ την εντολή “ping -c 1 192.168.1.1”. Παρατηρώ ότι η τιμή του πεδίου Ethertype των πλαισίων για τα πακέτα ARP είναι 0x0806, ενώ των πλαισίων IPv4 είναι 0x0800.

7.12) Εκτελώ την εντολή “ping -c 1 192.168.6.1” από το PC2. Παρατηρώ ότι τα πλαίσια Ethernet που παράγονται τώρα είναι μεγαλύτερα κατά 4 bytes. Αυτό δικαιολογείται από το γεγονός ότι η VLAN tag δεσμεύει 4 bytes από το πακέτο.

7.13) Η τιμή του πεδίου Ethertype των πακέτων αυτών είναι 0x8100. Τα πακέτα ARP περιέχουν επικεφαλίδα “ethertype arp” και τα πακέτα IPv4 περιέχουν επικεφαλίδα “ethertype IPv4”.

7.14) Η πληροφορία για το VLAN εμφανίζεται ακριβώς μετά το πεδίο που αναφέρει το μήκος length.

7.15) Ξεκινώ μια λεπτομερή καταγραφή στη διεπαφή em0 στο VLAN 5 ώστε να εμφανίζονται οι επικεφαλίδες Ethernet, καθώς και το περιεχόμενο



των πλαισίων σε δεκαεξαδική μορφή (αποθηκεύοντάς την στο αρχείο file2.txt) με την εντολή

```
“tcpdump -i em0.5 -e -vvv -x -l | tee file2.txt”.
```

7.16) Στο PC3 καθαρίζω τον πίνακα ARP με την εντολή “arp -a -d” και εκτελώ την εντολή “ping -c 1 192.168.5.1”. Το πεδίο Ethertype στα πλαίσια ARP έχει την τιμή 0x0806 και στα πακέτα που μεταφέρουν τα μηνύματα ICMP έχει την τιμή 0x0800. Επίσης, δεν φαίνεται στην καταγραφή κάποιο πεδίο σχετικό με το VLAN.

7.17) Στη γέφυρα bridge1 προσθέτω στο επικαλύπτον δένδρο τη διεπαφή του B1 στο LAN1 με την εντολή “ifconfig bridge1 stp em0”. Έπειτα, ξεκινώ μια λεπτομερή καταγραφή στο PC1 (αποθηκεύοντάς την στο αρχείο file3.txt) όπως πριν, για την διεπαφή em0 αυτή τη φορά με την εντολή “tcpdump -i em0 -e -vvv -x -l | tee file3.txt”.

7.18) Αυτά τα πλαίσια Ethernet που μεταφέρουν BPDU δεν είναι του ίδιου τύπου. Αυτό το συμπεραίνω από το γεγονός ότι στη θέση του πεδίου Ethertype υπάρχει η τιμή 802.3.

7.19) Αν δεν είχα αφαιρέσει τη διεπαφή του B1 στο LAN1 από το επικαλύπτον δένδρο, για να μην έκανα σύλληψη πλαισίων BPDU στις προηγούμενες καταγραφές στο PC1 θα εφάρμοζα το εξής φίλτρο: “tcpdump not stp ...”.