



Εργαστήριο Δικτύων Υπολογιστών

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ **3**
ΤΟΠΙΚΑ ΔΙΚΤΥΑ ΚΑΙ ΜΕΤΑΓΩΓΕΙΣ LAN

Κουστένης Χρίστος | el20227 | 03/03/2024

Άσκηση 1: Γέφυρα - Διασύνδεση δύο LAN

1.1

ifconfig em0 192.168.1.1/24 (στο PC1)

ifconfig em0 192.168.1.2/24 (στο PC2)

(Προσθέτουμε στο `/etc/hosts` αρχείο του PC1 τη γραμμή :

`192.168.1.2 PC2 myPC2.local`

Προσθέτουμε στο `/etc/hosts` αρχείο του PC2 τη γραμμή :

`192.168.1.1 PC1 myPC1.local`)

1.2

Με τις εντολές

ifconfig em0 up

ifconfig em1 up

ενεργοποιούμε τις διεπαφές em0 και em1 αντίστοιχα.

1.3

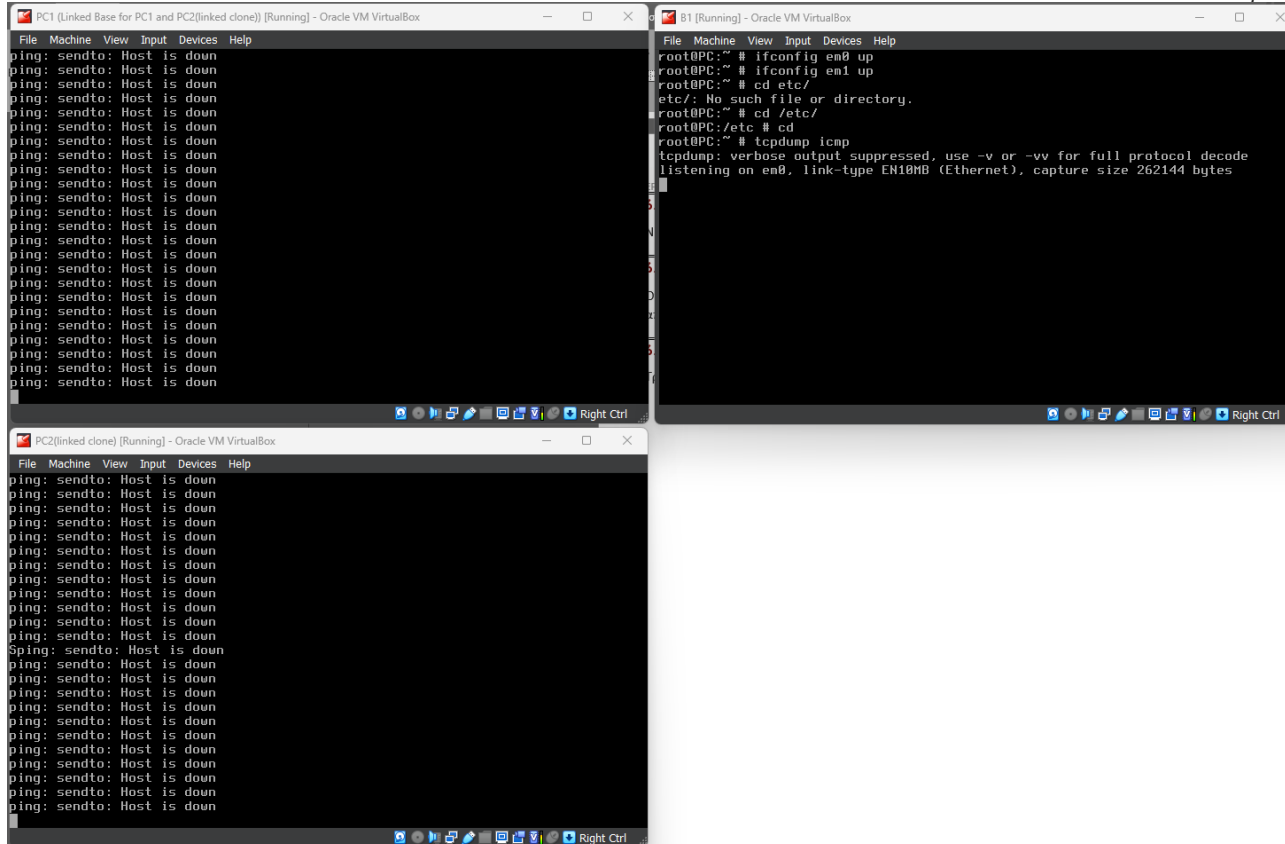
Εκτελώντας :

ping 192.168.1.2 (στο PC1)

ping 192.168.1.1 (στο PC2)

παρατηρούμε σε αμφότερα τα μηχανήματα πως τα πακέτα που έστειλαν απλά χάθηκαν.

1.4



Δεν παράγονται ICMP πακέτα όπως φαίνεται από την παραπάνω εικόνα των κονσόλων. Χωρίς τη γέφυρα δεν υπάρχει επικοινωνία μεταξύ τους και άρα το καθένα είναι απομονωμένο και ανήκει στο δικό του LAN (LAN1 και LAN2 αντίστοιχα).

1.5

ifconfig bridge create ---> Δημιουργία γέφυρας bridge0

ifconfig bridge0 addm em0 addm em1 up ---> Προσθήκη στη γέφυρα bridge0 των διεπαφών em0 και em1 και ενεργοποίηση της.

1.6

Ναι, υπάρχει.

1.7

Παρατηρούμε πως το πεδίο TTL έχει τιμή 64 που είναι και η default, επομένως είναι σαν το PC2 και το PC1 να συνδέονται άμεσα, πράγμα λογικό αφού η γέφυρα είναι « διαφανής » για το δίκτυο.

1.8

PC1:

```
root@PC1:~# arp -a
? (192.168.1.1) at 08:00:27:3b:25:25 on em0 permanent [ethernet]
? (192.168.1.2) at 08:00:27:40:f2:96 on em0 expires in 454 seconds [ethernet]
```

```
root@PC1:~ # arp -a
PC1 (192.168.1.1) at 08:00:27:3b:25:25 on em0 expires in 301 seconds [ethernet]
? (192.168.1.2) at 08:00:27:40:f2:96 on em0 permanent [ethernet]
```

Έχουν γίνει οι κατάλληλες αντιστοιχίσεις IP με MAC διευθύνσεις, χωρίς να παρεμβάλλονται MAC της γέφυρας. Επίσης, η κάθε συσκευή δίπλα από τη δική της αντιστοίχιση IP-MAC έχει τη λέξη permanent(μόνιμη) που υποδηλώνει τη στατική ανάθεση διεύθυνσης IP στην παρούσα συσκευή.

1.9

Ανοίγοντας δύο κονσόλες για το B1 μηχανήμα ξεκινήσαμε 2 καταγραφές με χρήση της εντολής

```
tcpdump icmp -e -vvv
```

1.10

Όχι, δε φαίνεται κάποια αλλαγή μεταξύ των 2 καταγραφών.

1.11

Όχι.

1.12

Δεν υπάρχει καμία ένδειξη της γέφυρας, όπως φαίνεται παρακάτω. Για το δίκτυο, η γέφυρα είναι «διαφανής».

```
root@PC1:~ # traceroute PC2
traceroute to PC2 (192.168.1.2), 64 hops max, 40 byte packets
 1 PC2 (192.168.1.2) 0.651 ms 0.466 ms 0.443 ms
```

1.13

ping PC2 (στο PC1)

```
tcpdump -i em1 icmp (στο B1)
```

1.14

```
ifconfig em0 192.168.2.1/24 (στο PC2)
```

Ναι, η γέφυρα τα προωθεί στη θύρα που ήταν πριν το PC2 καθώς υπάρχει η καταγραφή στον πίνακα προώθησης.

1.15

Όχι, το ping δεν είναι επιτυχές, γιατί πλέον δεν υπάρχει μηχανήμα με IP 192.168.1.2 το οποίο μπορεί να απαντήσει.

1.16

Όχι, δεν μπορώ, γιατί η διεπαφή em2 του LAN3 όπου ανήκει το PC3 δεν έχει προστεθεί στο bridge0.

1.17

ifconfig em2 up ---> Ενεργοποίηση διεπαφής.

ifconfig bridge0 addm em2 ---> Προσθήκη διεπαφής στο bridge.

1.18

Ναι.

1.19

Όχι, δεν μεταδίδονται πακέτα στο LAN2 γιατί λόγω του προηγούμενου ring η γέφυρα έχει συμπληρώσει τον πίνακα προώθησης για τα PC1, PC3 και επομένως προωθεί τα πακέτα στη σωστή θύρα.

1.20

Αυτή τη φορά, καταγράφηκε το παρακάτω ARP request. Ο λόγος που έγινε αυτό, είναι πως αφού το PC1 δεν ήξερε την MAC διεύθυνση που αντιστοιχεί στην IP του PC3, το πακέτο προωθήθηκε και στο LAN2 αλλά και στο LAN3 (broadcast) προκειμένου να το απαντήσει ο υπολογιστής με IP αυτή του PC3 και να μάθουμε την MAC address του, ώστε να επιτευχθεί η δρομολόγηση του πακέτου.

```
03:05:36.247048 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.3  
tell 192.168.1.1, length 46
```

1.21

ifconfig bridge0

```
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    ether 58:9c:fc:10:ff:bf
    id 00:00:00:00:00:00 priority 32768 hellotime 2 fwhdelay 15
    maxage 20 holdcnt 6 proto rstp maxaddr 2000 timeout 1200
    root id 00:00:00:00:00:00 priority 32768 ifcost 0 port 0
    member: em2 flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPTP>
        ifmaxaddr 0 port 3 priority 128 path cost 20000
    member: em1 flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPTP>
        ifmaxaddr 0 port 2 priority 128 path cost 20000
    member: em0 flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPTP>
        ifmaxaddr 0 port 1 priority 128 path cost 20000
    groups: bridge
    nd6 options=9<PERFORMNUD,IFDISABLED>
root@PC:~ #
```

1.22

ifconfig bridge0 addr

```
root@PC:~ # ifconfig bridge0 addr
08:00:27:c6:ca:a4 Vlan1 em2 1070 flags=0<>
08:00:27:d8:f5:d9 Vlan1 em1 676 flags=0<>
08:00:27:3b:25:25 Vlan1 em0 1070 flags=0<>
```

1.23

08:00:27:3b:25:25 ---> PC1

08:00:27:d8:f5:d9 ---> PC2

08:00:27:c6:ca:a4 ---> PC3

1.24

ifconfig bridge0 flush (στο B1)

1.25

ifconfig bridge0 deletem em2 (στο B1)

1.26

ifconfig bridge0 destroy (στο B1)

1.27

ifconfig em0 delete (στα PC1, PC2 και PC3)

Άσκηση 2: Αυτο-εκπαίδευση γεφυρών

2.1

ifconfig em0 192.168.1.X ($X = \{1,2,3,4\}$)

2.2

Στο μηχάνημα B1, εκτελούμε διαδοχικά τις εντολές :

ifconfig em0 up

ifconfig em1 up

ifconfig bridge1 create

ifconfig bridge1 addm em0 addm em1 up

2.3

Στο μηχάνημα B2, εκτελούμε διαδοχικά τις εντολές :

ifconfig em0 up

ifconfig em1 up

ifconfig bridge2 create

ifconfig bridge2 addm em0 addm em1 up

2.4

Στο μηχάνημα B3, εκτελούμε διαδοχικά τις εντολές :

ifconfig em0 up

ifconfig em1 up

ifconfig bridge3 create

ifconfig bridge3 addm em0 addm em1 up

2.5

Εκτελούμε την εντολή « **ifconfig em0** » σε κάθε PC και βρίσκουμε τις MAC διευθύνσεις που αναγράφουμε παρακάτω. Επιπλέον, εκτελούμε « **arp -ad** ».

PC1 --> 08:00:27:3b:25:25

PC2 --> 08:00:27:d8:f5:d9

PC3 --> 08:00:27:c6:ca:a4

2.6

ifconfig bridgeX flush --> Καθαρισμός πίνακα προώθησης γέφυρας X = {1,2,3}

2.7

tcpdump -i em0 -e -vvv (σε όλα τα PC)

2.8

ifconfig bridgeX addr---> Πριν το ping έχουμε κενό πίνακας σε όλα τα bridges.

ifconfig bridgeX addr ---> Πίνακας προώθησης γέφυρας X={1,2,3}

bridge1 --> PC1, PC2

bridge2 --> PC1, PC2

bridge3 --> PC1

```
root@PC:~ # ifconfig bridge1 addr
08:00:27:d8:f5:d9 Vlan1 em1 333 flags=0<>
08:00:27:3b:25:25 Vlan1 em0 333 flags=0<>
```

```
root@PC:~ # ifconfig bridge2 addr
08:00:27:d8:f5:d9 Vlan1 em0 317 flags=0<>
08:00:27:3b:25:25 Vlan1 em0 317 flags=0<>
```

```
root@PC:~ # ifconfig bridge3 addr
08:00:27:3b:25:25 Vlan1 em0 298 flags=0<>
```

2.9

Η γέφυρα B1 έχει καταχωρήσει την MAC του PC1 στην em0 (LAN1) και του PC2 στην em1 (LNK1). Η B2 έχει καταχωρήσει τις MAC των PC1 και PC2 στην em0 (LNK1). Η B3 έχει καταχωρήσει μόνο την MAC του PC1 στην em0 (LNK2). Οι καταχωρήσεις έγιναν ως εξής:

- Το PC1 έστειλε ARP request για να μάθει την MAC του PC2, οπότε και το B1 καταχώρησε την MAC_{PC1} στην em0 (LAN1).
 - Από το B1, στάλθηκε στο LNK1 το ARP request, οπότε και το έλαβε ο στόχος PC2, αλλά και η B2, οπότε και το καταχώρησε στο em0 (LNK1).
 - Το B2 με τη σειρά του, το προώθησε στο LNK2, οπότε και έλαβε το PC2 αλλά και η B3, η οποία το καταχώρησε στο em0 (LNK2).
 - Στο μεταξύ, το PC2 εξέπεμψε την απάντηση του στο LNK1, την οποία και έλαβε η B1 και η B2, καταχωρώντας έτσι την MAC_{PC2} η μεν B1 στο em1 (LNK1) και η δε B2 στο em0 (LNK1).
-

2.10

Όχι, δεν υπήρξαν αλλαγές, διότι οι γέφυρες θα οδηγήσουν τα μηνύματα προς τον PC1 από το σωστό μονοπάτι και έτσι η B3 δεν θα μάθει ποτέ για τα στοιχεία του PC2.

2.11

bridge1 → PC1, PC2, PC4

bridge2 → PC1, PC2, PC4

bridge3 → PC1, PC2, PC4

Ο πίνακας προώθησης του B1 περιέχει εγγραφή για το PC4 γιατί όταν το ARP reply από το PC4 εκπέμπεται από το B2 στη θύρα em0 το λαμβάνει ο B1 και αν και δεν το προωθεί σε άλλη θύρα, το αποθηκεύει στο πίνακα προώθησης.

```

B1:00:27:3b:25:25 Vlan1 en0 1189 flags=0<
root0PC:~ # ifconfig bridge1 addr
00:00:27:01:c1:bc Vlan1 en1 1195 flags=0<
00:00:27:08:f5:d9 Vlan1 en1 1195 flags=0<
00:00:27:3b:25:25 Vlan1 en0 1098 flags=0<
root0PC:~ #

B2:00:27:3b:25:25 Vlan1 en0 1184 flags=0<
root0PC:~ # ifconfig bridge2 addr
00:00:27:01:c1:bc Vlan1 en1 1192 flags=0<
00:00:27:08:f5:d9 Vlan1 en0 1192 flags=0<
00:00:27:3b:25:25 Vlan1 en0 1095 flags=0<
root0PC:~ #

B3:00:27:3b:25:25 Vlan1 en0 1186 flags=0<
root0PC:~ # ifconfig bridge3 addr
00:00:27:01:c1:bc Vlan1 en1 1186 flags=0<
00:00:27:08:f5:d9 Vlan1 en0 1186 flags=0<
00:00:27:3b:25:25 Vlan1 en0 729 flags=0<
root0PC:~ #

```

Αναλυτικότερα

Πλέον όλες οι γέφυρες έχουν την MAC του PC4. Η καταγραφή αυτή υπάρχει στο B1, καθώς, προκειμένου το PC2 να στείλει το πακέτο στο PC4 έκανε αρχικά ένα ARP request. Όταν το PC4 το έλαβε, εξέπεμψε το ARP reply, το οποίο και ο B3 έκανε broadcast στο LNK2. Παραλαμβάνοντάς το από εκεί, επίσης broadcast στο LNK1 έκανε το B2 με τελικό αποτέλεσμα η καταγραφή MAC_{PC4} να υπάρχει και στο B1.

2.12

bridge1 → PC1, PC2, PC3, PC4

bridge2 → PC1, PC2, PC3, PC4

bridge3 → PC1, PC2, PC3, PC4

Το ARP request από το PC3 λαμβάνεται από τα B2, B3 και επειδή το B2 προωθεί το πακέτο στη θύρα του προς το LNK1 λαμβάνεται και από το B1. Έτσι, πλέον και οι 3 γέφυρες έχουν εγγραφές για όλα τα PC.

2.13

ping 192.168.1.2

2.14

Το ping από το PC4 στο PC2 συνεχίζει κανονικά

2.15

Το ping σταματάει γιατί η γέφυρα B2 δεν προωθεί ποτέ το πακέτο στη θύρα em1 (λόγω του παλιού πίνακα προώθησης) και επομένως το icmp request δεν φτάνει ποτέ στο PC2.

2.16

Το ping αρχίζει και πάλι να λειτουργεί, διότι με το ping από το PC2 ο πίνακας προώθησης του B2 ανανεώθηκε.

Αυτό που συνέβη είναι πως το PC2 έστειλε πακέτο στο PC3, επομένως, στάλθηκε πακέτο στο B3, το οποίο επίσης μεταδόθηκε στο LNK2. Από εκεί, το B2 ενημέρωσε τον πίνακά του πως το PC2 βρίσκεται πλέον από τη θύρα em1 αντί

για την em0 που ήταν πριν. Επομένως, από το αμέσως επόμενο πακέτο που έστειλε το PC1, αυτό έφτασε στο B1, στάλθηκε στο LNK1, αλλά από εκεί όταν παρελήφθη από το B2 προωθήθηκε στο LNK2 για να φτάσει τελικά στο PC2.

2.17

Εκτελώντας « **ifconfig bridge1** » στο B1, βλέπουμε πως ο μέγιστος χρόνος που διατηρείται μια εγγραφή στον πίνακα προώθησης είναι 1200 seconds, άρα το πολύ μετά από 20 λεπτά θα μάθαινε μόνη της η γέφυρα για την αλλαγή.

Άρα θα έπρεπε να περιμένω μέχρι να λήξει η εγγραφή για το PC2 από το πίνακα προώθησης του B1.

Άσκηση 3 : Καταιγίδα πλαισίων εκπομπής

3.1

Έχοντας ήδη ενεργές τις διεπαφές em0, em1, εκτελούμε :

```
ifconfig bridge1 create
```

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

3.2

Έχοντας ήδη ενεργές τις διεπαφές em0, em1, εκτελούμε :

```
ifconfig bridge2 create
```

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

3.3

PC1 --> 08:00:27:3b:25:25

PC2 --> 08:00:27:d8:f5:d9

PC3 --> 08:00:27:c6:ca:a4

3.4

Παρατηρούμε πως καταγράφεται ARP κίνηση, όπου κάνοντας ping από το PC2 στο PC3 (και αντιστρόφως έπειτα από διαγραφή εκ νέου των arp tables), το PC2 ρωτάει για την MAC της IP_{PC3}. Αναλυτικότερα, δεδομένου ότι το PC2 δε γνώριζε τη MAC του PC3 για να στείλει αμέσως το πακέτο, έστειλε ένα ARP request για τη μάθει. Δεδομένου ότι το PC3 βρίσκεται στο ίδιο LAN με αυτό, έμαθε αμέσως την απάντηση οπότε και στάλθηκε το ICMP Echo μήνυμα. Ωστόσο, παράλληλα, προωθήθηκε το ARP Request μέσω του B2 στο LNK1 και από εκεί στο B1, το οποίο και προώθησε το request τέλος στο PC1 το κατέγραψε το request.

3.5

ping 192.168.1.1 (στο PC3)

3.6

Αφού ενεργοποιήσουμε την em2 στα B1, B2, τα κάνουμε add στην εκάστοτε γέφυρα με την εντολή
« **ifconfig bridgeX addm em2 up** » $X=\{1,2\}$

3.7

ifconfig bridgeX addr ($X=\{1,2\}$)--> Και οι δύο γέφυρες έχουν εγγραφές και για τα 3 PC

3.8

B1

PC1 ---> em0(LAN1)

PC3 ---> em1(LNK1)

B2

PC1 ---> em1(LNK1)

PC3 ---> em0(LAN2)

3.9

tcpdump -e

3.10

arp -ad

Το « **ping -c 1 192.168.1.1** » είναι ανεπιτυχές.

3.11

B1

PC1 ---> em0(LAN1)

PC3 ---> em1(LNK1)

B2

PC1 ---> em1(LNK1)

PC3 ---> em1(LNK1)

Εξήγηση

Το PC3 στέλνει το ARP request στο LAN2, το οποίο παραλαμβάνει το PC1 και το B2. Από εκεί, αυτό αποστέλλεται προς το LNK1 και το LNK2, επομένως, πλέον το B1 που τα παραλαμβάνει θεωρεί πως το PC3 είναι στο LNK1(em1) ή στο LNK2(em2) (όποιο παρέλαβε τελευταίο). Αφού προωθήσει το πακέτο στο LAN1(em0), το προωθεί και στο LNK1(em1 για το B1 και το B2) και LNK2(em2 για το B1 και το B2), αφού το έλαβε από το LNK2 ή από το LNK1. Επομένως, τα λαμβάνει

ο B2 και θεωρεί πλέον πως το PC3 είναι στο em1(LNK1) ή στο em2(LNK2), ανάλογα με το αν έλαβε τελευταίο πακέτο από το LNK1 ή το LNK2. Άρα, όταν έρθει η απάντηση από το PC1, θα εγκλωβιστεί αενάως μεταξύ των B1, B2 αφού το PC3 θα φαίνεται να είναι είτε στο LNK1 είτε στο LNK2.

3.12

ARP request: who has 192.168.1.1 ?

ARP reply: 192.168.1.1 is at 08:00:27:3b:25:25

3.13

MAC_{PC3} --> 08:00:27:c6:ca:a4

3.14

Λόγω του βρόχου που έχει δημιουργηθεί μεταξύ των B1, B2, το ARP Request, το οποίο και είναι broadcast, προωθείται συνέχεια εκτός του βρόχου, οπότε και λαμβάνεται από το PC2. Άρα εντοπίζεται σε αμφότερες τις καταγραφές

3.15

Διότι πλέον η MAC του PC3 είναι είτε στην em1/em2 (LNK1/LNK2) για το B1, είτε στην em1/em2 (LNK1/LNK2) για το B2, άρα δε φεύγει ποτέ εκτός βρόχου. Φεύγουν μόνο τα πακέτα που είναι για broadcast.

Άσκηση 4 : Συνάθροιση ζεύξεων

4.1

ifconfig bridgeX destroy

ifconfig emX down

ifconfig bridgeX create

$X = \{1,2,3\}$

4.2

ifconfig emX up

ifconfig lagg create

4.3

ifconfig lagg0 up laggport em1 laggport em2

4.4

Ίδιες εντολές με 4.2, 4.3

4.5

ifconfig bridge0 addm em0 addm lagg0 up (στο B1)

4.6

ifconfig bridge1 addm em0 addm lagg0 up (στο B2)

4.7

Κάνοντας ping από το PC2 στο PC3, βλέπουμε στην καταγραφή του PC1 το ARP Request στο οποίο το PC3 ρωτάει για την MAC του PC2. Αυτό που συνέβη είναι πως, δεδομένου ότι είχαμε καθαρίσει όλους τους ARP πίνακες, όταν πήγαμε να στείλουμε πακέτα από το PC2 στο PC3, δεδομένου ότι ο πρώτος δε γνώριζε τη MAC του 2^{ου} έκανε broadcast στο LAN2 ένα ARP request για να τη μάθει. Από εκεί, το B2 την έκανε επίσης broadcast προς κάθε άλλη θύρα, εν προκειμένω προς το lagg1, από όπου και το παρέλαβε στο δικό του lagg0 το B1. Τέλος, το B1 το έκανε broadcast στο LAN1, από όπου και παρελήφθη από το PC1.

4.8

tcpdump -i em0 (στο PC1)

4.9

Ναι, είναι επιτυχές, στάλθηκαν κανονικά ARP request, reply και ICMP request, reply.

```
root@PC1:~ # tcpdump -i em0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
20:34:36.824681 ARP, Request who-has 192.168.1.1 tell PC3, length 46
20:34:36.824714 ARP, Reply 192.168.1.1 is-at 08:00:27:3b:25:25 (oui Unknown), length 28
20:34:36.827140 IP PC3 > 192.168.1.1: ICMP echo request, id 44547, seq 0, length 64
20:34:36.827212 IP 192.168.1.1 > PC3: ICMP echo reply, id 44547, seq 0, length 64
```

4.10

Εκτελούμε « **tcpdump -i em1** » στο B1 και « **tcpdump -i em2** » στο B2. Παρατηρούμε πως τα πακέτα εμφανίζονται στο LNK1 (em1 του B1, em1 του B2).

Εξήγηση

Το προεπιλεγμένο πρωτόκολλο συνάθροισης είναι το failover και μελετώντας την τεκμηρίωσή του, βλέπουμε πως η κίνηση μεταφέρεται μέσα από το master port, το οποίο εν προκειμένω είναι το em1 στο B1 και το em1 στο B2, καθώς αυτές ήταν οι πρώτες διεπαφές που προστέθηκαν στις συσκευές συνάθροισης lagg0. Παρακάτω, βλέπουμε τα Master Ports των B1 και B2 για να επαληθεύσουμε τα παραπάνω.

The image shows two terminal windows from Oracle VM VirtualBox. The left window is titled 'B2 [Running] - Oracle VM VirtualBox' and shows the configuration of the 'lagg0' interface. It includes commands like 'ifconfig bridge2 addm em0 addm lagg0 up', 'tcpdump -i em2', and 'ifconfig lagg0'. The right window is titled 'B1 [Linked Base for B1 and B3] [Running] - Oracle VM VirtualBox' and shows network traffic capture on the 'lagg0' interface, displaying ICMP echo requests and replies between 192.168.1.1 and 192.168.1.2.

4.11

Πλέον, η κίνηση καταγράφεται στο LNK2 (στο em2 του B2 που έτρεχε πριν), για τον ίδιο λόγο που αναφέραμε παραπάνω, το πρωτόκολλο της συνάθροισης που είναι ορισμένο στο failover, το οποίο και μας λέει πως όταν συμβεί κάποια βλάβη σε master port, γίνεται master port η διεπαφή που προστέθηκε αμέσως μετά αυτής που ήταν προηγουμένως master. Βλέπουμε παρακάτω τις καταστάσεις των ports:

The image shows two terminal windows. The left window is titled 'B2 [Running] - Oracle VM VirtualBox' and shows the configuration of the 'lagg0' interface, including 'ifconfig lagg0' and 'ifconfig lagg0'. The right window is titled 'B1 [Linked Base for B1 and B3] [Running] - Oracle VM VirtualBox' and shows the configuration of the 'lagg0' interface, including 'ifconfig lagg0' and 'ifconfig lagg0'.

4.12

Η κίνηση επέστρεξε και πάλι στη ζεύξη LNK1.

Άσκηση 5 : Αποφυγή βρόχων

5.1

`ifconfig lagg0 destroy`

`ifconfig bridgeX destroy ,X={1,2}`

`ifconfig emX down ,X={0,1,2}`

5.2

`ifconfig em0 up`

```
ifconfig em1 up
```

```
ifconfig em2 up
```

```
ifconfig bridge1 create
```

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

5.3

```
ifconfig em0 up
```

```
ifconfig em1 up
```

```
ifconfig em2 up
```

```
ifconfig bridge2 create
```

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

5.4

```
ifconfig bridge1 stp em0 stp em1 stp em2
```

5.5

```
ifconfig bridge2 stp em0 stp em1 stp em2
```

5.6

```
B1 BridgeID ---> 08:00:27:5c:3c:5a
```

```
B2 BridgeID ---> 08:00:27:46:91:96
```

5.7

Η B2. Και οι δύο γέφυρες έχουν ίδια προτεραιότητα (default), όμως η γέφυρα ρίζα θα είναι η B2 που έχει τη μικρότερη MAC.

```
root@PC:~ # ifconfig bridge2
bridge2: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
ether 58:9c:fc:00:2a:0b
id 08:00:27:46:91:96 priority 32768 hellotime 2 fwddelay 15
maxage 20 holdcnt 6 proto rstp maxaddr 2000 timeout 1200
root id 08:00:27:46:91:96 priority 32768 ifcost 0 port 0
member: em2 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
ifmaxaddr 0 port 3 priority 128 path cost 20000 proto rstp
role designated state forwarding
member: em1 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
ifmaxaddr 0 port 2 priority 128 path cost 20000 proto rstp
role designated state forwarding
member: em0 flags=1e7<LEARNING,DISCOVER,STP,EDGE,AUTOEDGE,PTP,AUTOPTP>
ifmaxaddr 0 port 1 priority 128 path cost 20000 proto rstp
role designated state forwarding
groups: bridge
nd6 options=9<PERFORMNUD,IFDISABLED>
root@PC:~ #
```

5.8

Οι καταστάσεις και οι ρόλοι των διεπαφών της γέφυρας ρίζας περιγράφονται παρακάτω:

- em0:

State → Forwarding

Role → Designated

- em1:

State → Forwarding

Role → Designated

- em2:

State → Forwarding

Role → Designated

Επομένως, όλες οι θύρες είναι πλήρως λειτουργικές και προωθούν προς τμήματα LAN μακριά από τη ρίζα.

5.9

Η em1, δηλαδή η διεπαφή LNK1 είναι η ριζική θύρα.

Role : root

State : discarding

```
bridge1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
 ether 58:9c:fc:10:ff:f7
 id 08:00:27:5c:3c:5a priority 32768 hellotime 2 fwhdelay 15
 maxage 20 holdcnt 6 proto rstp maxaddr 2000 timeout 1200
 root id 08:00:27:46:91:96 priority 32768 ifcost 20000 port 2
 member: em2 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
         ifmaxaddr 0 port 3 priority 128 path cost 20000 proto rstp
         role alternate state discarding
 member: em1 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
         ifmaxaddr 0 port 2 priority 128 path cost 20000 proto rstp
         role root state forwarding
 member: em0 flags=1e7<LEARNING,DISCOVER,STP,EDGE,AUTOEDGE,PTP,AUTOPTP>
         ifmaxaddr 0 port 1 priority 128 path cost 20000 proto rstp
         role designated state forwarding
 groups: bridge
 nd6 options=9<PERFORMNUD,IFDISABLED>
root@PC:~ #
```

5.10

LNK2(em2)

Role : alternate

Είναι εναλλακτική της ριζικής θύρας για τη διαδρομή προς τη γέφυρα ρίζα και δεν αποστέλλει πλαίσια.

5.11

LAN1(em0)

Role : designated

State : forwarding

Συνεπώς είναι πλήρως λειτουργική και προωθεί προς τμήμα LAN (προς το LAN1 συγκεκριμένα).

5.12

`tcpdump -i em1 -e -vvv (στο B2) --->` εκπέμπονται κάθε 2 δευτερόλεπτα (hello-time)

5.13

Χρησιμοποιείται ενθυλάκωση IEEE 802.3.

5.14

MAC Source : 08:00:27:93:60:c1 (em1 LNK1)

MAC Destination : 01:80:c2:00:00:00

```
02:13:59.312275 08:00:27:93:60:c1 (oui Unknown) > 01:80:c2:00:00:00 (oui Unknown), 802.3, length 39: LLC, dsap STP (0x42) Individual, ssap STP (0x42) Command, ctrl 0x03: STP 802.1w, Rapid STP, Flags [Learn, Forward], bridge-id 8000.08:00:27:46:91:96.8002, length 36
    message-age 0.00s, max-age 20.00s, hello-time 2.00s, forwarding-delay 15.00s
    root-id 8000.08:00:27:46:91:96, root-pathcost 0, port-role Designated
```

5.15

Ανήκει στη θύρα em1 που συνδέεται στο LNK1.

Σημείωση : Επειδή η ριζική θύρα στην περίπτωση μου ήταν η bridge2 έκανα καταγραφή στη διεπαφή προς την κατεύθυνση της LAN1 που είναι η LNK1 αφού πρώτα από εκεί θα περάσουν τα πακέτα που φτάνουν στα LAN1.

5.16

Είναι multicast (01:80:c2:00:00:00), καθώς η σειρά με την οποία θα διαβαστεί η διεύθυνση είναι πρώτα το LSB του πρώτου byte, άρα το 1.

5.17

bridge ID : 8000.08:00:27:46:91:96.8002

root ID : 8000.08:00:27:46:91:96

root-pathcost : 0

5.18

`tcpdump -i em0 -e -vvv (στο B2)`

Από την Bridge ID (**8000**.08:00:27:46:91:96.8002) προτεραιότητα είναι το πρώτο μέρος με τιμή 8000, το οποίο είναι σε δεκαεξαδική μορφή και μεταφράζεται σε 32.768 σε δεκαδική.

5.19

8000.**08:00:27:46:91:96**.8002

Το δεύτερο μέρος είναι η MAC διεύθυνση της γέφυρας ρίζας.

5.20

8000.08:00:27:46:91:96.**8002**

Εκτελώντας `ifconfig bridge2` βλέπουμε πως η `em1` έχει port 2, επομένως το 8002 προκύπτει ως άθροισμα της προτεραιότητας με τον αριθμό της θύρας.

5.21

Ναι, παρατηρώ.

```
root@PC:~# tcpdump -i em1 -e -vvv
tcpdump: listening on em1, link-type EN10MB (Ethernet), capture size 262144 bytes
03:29:13.539491 08:00:27:93:60:c1 (oui Unknown) > 01:80:c2:00:00:00 (oui Unknown), 802.3, length 39: LLC, dsap STP (0x42) Individual, ssap STP (0x42) Command, control 0x03: STP 802.1w, Rapid STP, Flags [Learn, Forward], bridge-id 8000.08:00:27:46:91:96.8002, length 43
    message-age 0.00s, max-age 20.00s, hello-time 2.00s, forwarding-delay 15.00s
    root-id 8000.08:00:27:46:91:96, root-pathcost 0, port-role Designated
03:29:15.554494 08:00:27:93:60:c1 (oui Unknown) > 01:80:c2:00:00:00 (oui Unknown), 802.3, length 39: LLC, dsap STP (0x42) Individual, ssap STP (0x42) Command, control 0x03: STP 802.1w, Rapid STP, Flags [Learn, Forward], bridge-id 8000.08:00:27:46:91:96.8002, length 43
    message-age 0.00s, max-age 20.00s, hello-time 2.00s, forwarding-delay 15.00s
    root-id 8000.08:00:27:46:91:96, root-pathcost 0, port-role Designated
```

```

root@PC:~# tcpdump -i em2 -vvve
tcpdump: listening on em2, link-type EN10MB (Ethernet), capture size 262144 bytes
03:28:40.930077 08:00:27:90:45:95 (oui Unknown) > 01:80:c2:00:00:00 (oui Unknown), 802.3, length 39: LLC, dsap STP (0x42) Individual, ssap STP (0x42) Command, ctrl 0x03: STP 802.1w, Rapid STP, Flags [Learn, Forward], bridge-id 8000.08:00:27:46:91:96.8003, length 43
    message-age 0.00s, max-age 20.00s, hello-time 2.00s, forwarding-delay 15.00s
    root-id 8000.08:00:27:46:91:96, root-pathcost 0, port-role Designated
03:28:42.993665 08:00:27:90:45:95 (oui Unknown) > 01:80:c2:00:00:00 (oui Unknown), 802.3, length 39: LLC, dsap STP (0x42) Individual, ssap STP (0x42) Command, ctrl 0x03: STP 802.1w, Rapid STP, Flags [Learn, Forward], bridge-id 8000.08:00:27:46:91:96.8003, length 43
    message-age 0.00s, max-age 20.00s, hello-time 2.00s, forwarding-delay 15.00s
    root-id 8000.08:00:27:46:91:96, root-pathcost 0, port-role Designated
03:28:44.993539 08:00:27:90:45:95 (oui Unknown) > 01:80:c2:00:00:00 (oui Unknown), 802.3, length 39: LLC, dsap STP (0x42) Individual, ssap STP (0x42) Command, ctrl 0x03: STP 802.1w, Rapid STP, Flags [Learn, Forward], bridge-id 8000.08:00:27:46:91:96.8003, length 43
    message-age 0.00s, max-age 20.00s, hello-time 2.00s, forwarding-delay 15.00s
    root-id 8000.08:00:27:46:91:96, root-pathcost 0, port-role Designated

```

5.22

Από τις θύρες 1, 2 και 3 των διεπαφών em0(LAN1), em1(LNK1) και em2(LNK2) αντίστοιχα.

5.23

Root ID : **8000.08:00:27:46:91:96** (em0-LAN1, em1-LNK1, em2-LNK2)

Bridge ID : **8000.08:00:27:5c:3c:5a.8001** (em0-LAN1), **8000.08:00:27:46:91:96.8002** (em1-LNK1), **8000.08:00:27:46:91:96.8003** (em2-LNK2)

Root path cost : 0 (em1-LNK1, em2-LNK2) , 20000 (em0-LAN1)

5.24

ping 192.168.1.2

Ναι, είναι επιτυχές.

5.25

Περνάνε περίπου 6 δευτερόλεπτα. Αυτό γίνεται διότι στο RSTP χρειάζονται περίπου 3x2 seconds για να ανταποκριθεί σε κάποια τοπολογική αλλαγή του δικτύου.

5.26

Όχι, δεν υπάρχει διακοπή.

Άσκηση 6: Ένα πιο πολύπλοκο δίκτυο με εναλλακτικές διαδρομές

Ευρετήριο διεπαφών

B1

em0: LAN1

em2: LNK2

em3: LNK3

B2

em0: LAN2

em1: LNK1

em2: LNK2

em3: LNK4

B3

em0: LAN3

em1: LNK3

em2: LNK4

6.1

ifconfig em3 up

ifconfig bridge1 addm em3

ifconfig bridge1 stp em3

6.2

ifconfig em3 up

ifconfig bridge2 addm em3

ifconfig bridge2 stp em3

6.3

ifconfig bridge3 create

ifconfig bridge3 addm em0 addm em1 addm em2 up

ifconfig bridge3 stp em0 stp em1 stp em2

6.4

ifconfig bridgeX flush , X={1,2,3}

Ναι είναι επιτυχές.

6.5

ifconfig bridge1 priority 0

6.6

ifconfig bridge2

Τα path costs είναι όλα 20000. Επομένως από τον τύπο $\frac{20\text{Tbit/s}}{\text{bandwidth}}$ προκύπτει σωστά για bandwidth = 1Gbps.

6.7

Τα root path cost που λαμβάνει στα πλαίσια BPDU από τις γέφυρες 1 και 2 είναι αντίστοιχα 0 και 20.000 (τα διαβάζουμε με « **tcpdump -i em1 -vvve** » και « **tcpdump -i em2 -vvve** » αντίστοιχα). Η πρώτη τιμή είναι 0 επειδή η bridge1 είναι η ρίζα, επομένως δεν υπάρχει κόστος από αυτή μέχρι τη γέφυρα ρίζα, ενώ η δεύτερη είναι 20000 επειδή το Bandwidth της κάρτας δικτύου της ριζικής θύρας LNK1/LNK2 της γέφυρας bridge2 είναι 1Gbps, αφού για να φτάσει ένα πακέτο στην ρίζα από την bridge2 θα χρειαστεί να διανύσει και την απόσταση του LNK1/LNK2.

6.8

Εκτελώντας « **ifconfig bridge3** » βλέπουμε πως ριζική θύρα είναι η em1 (LNK3) και αυτό διότι από εκεί πάει άμεσα στο B1, επομένως έχει το μικρότερο δυνατό κόστος (δεδομένου ότι όλες οι διαδρομές έχουν κόστος 20000).

6.9

Όσον αφορά τη θύρα στο LNK4 για το B3, ο ρόλος της είναι designated και η κατάστασή της forwarding (αν και θα αναμέναμε role alternate state discarding), ωστόσο στο B2 η θύρα στο LNK4 έχει ρόλο alternate και κατάσταση discarding.

6.10

Εκτελώντας « **tcpdump -vvvei em0** » στο PC3 βλέπουμε πως καταγράφονται BPDU πλαίσια με root path cost ίσο με 20000.

6.11

ping 192.168.1.3

6.12

Το κόστος μέσω της LNK4 από το B3 προς τη γέφυρα ρίζα αναμένεται πως είναι 20000 (κόστος LNK4) + 20000 (κόστος LNK1) = 40000. Άρα θα θέσουμε μια τιμή μεγαλύτερη από 40000 στο υπάρχον κόστος της (20000). Εκτελούμε, επομένως, την εντολή « **ifconfig bridge3 ifpathcost em1 40001** », και παρατηρούμε (« **ifconfig bridge3** ») πως ριζική πλέον θύρα στο B3 είναι η em2 (LNK4) .

6.13

Περνάνε περίπου 2 δευτερόλεπτα (hello-time).

6.14

ifconfig bridge3

ifconfig bridge2

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

Δεν παρατηρείται κάποια διαφορά στις παραμέτρους που λαμβάνει η bridge3 (« **tcpdump -vvve -i em0** » και « **tcpdump -vvve -i em1** »).

6.16

Όσον αφορά τα παραγόμενα BPDUs από τη γέφυρα bridge3, πλέον το root path cost έχει τιμή 40000.

6.17

Περίπου 6-8 δευτερόλεπτα (3 hello-time).

6.18

Αφού επανασυνέδεσα το καλώδιο που αποσυνέδεσα στο προηγούμενο ερώτημα χρειάστηκαν περίπου 2 δευτερόλεπτα (hellotime = 2 sec) έως ότου αποκατασταθεί ξανά η επικοινωνία.

6.19

Συνδέω την 4η 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

```
ifconfig bridge3 ifpathcost em1 20000
```

Διαλέξαμε την αρχική της τιμή αφού στο LNK3 βρίσκεται και η B1 οπότε θα επιλεγεί ως ριζική, καθώς το κόστος μέσω της LNK4 από το B3 προς τη γέφυρα ρίζα είναι 40000, ενώ μέσω της LNK3 είναι πλέον 20000.

```
root@PC:~ # ifconfig bridge3 ifpathcost em1 20000
root@PC:~ # ifconfig bridge3
bridge3: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    ether 58:9c:fc:10:ff:da
    id 08:00:27:05:5c:b2 priority 32768 hellotime 2 fiddelay 15
    maxage 20 holdcnt 6 proto rstp maxaddr 2000 timeout 1200
    root id 08:00:27:5c:3c:5a priority 0 ifcost 20000 port 2
    member: em3 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
        ifmaxaddr 0 port 4 priority 128 path cost 20000 proto rstp
        role disabled state discarding
    member: em2 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
        ifmaxaddr 0 port 3 priority 128 path cost 20000 proto rstp
        role designated state discarding
    member: em1 flags=5c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
        ifmaxaddr 0 port 2 priority 128 path cost 20000 proto rstp
        role root state learning
    member: em0 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
        ifmaxaddr 0 port 1 priority 128 path cost 20000 proto rstp
        role designated state forwarding
    groups: bridge
    nd6 options=9<PERFORMNUD,IFDISABLED>
root@PC:~ #
```

Άσκηση 7: Εικονικά τοπικά δίκτυα (VLAN)

7.1

```
ifconfig em0.5 create vlan 5 vlandev em0 inet 192.168.5.1/24
```

```
ifconfig em0.6 create vlan 6 vlandev em0 inet 192.168.6.1/24
```

7.2

```
ifconfig em0.5 create vlan 5 vlandev em0 up
```

```
ifconfig em0.6 create vlan 6 vlandev em0 up
```

7.3

```
ifconfig em1.6 create vlan 6 vlandev em1 up
```

```
ifconfig em3.5 create vlan 5 vlandev em3 up
```

7.4

```
ifconfig em0.6 create vlan 6 vlandev em0 inet 192.168.6.2/24 up
```

7.5

```
ifconfig em0.6 create vlan 6 vlandev em0 up
```

```
ifconfig em1.6 create vlan 6 vlandev em1 up
```

7.6

```
ifconfig em0.5 create vlan 5 vlandev em0 inet 192.168.5.3/24 up
```

7.7

```
ifconfig em0.5 create vlan 5 vlandev em0 up
```

```
ifconfig em1.5 create vlan 5 vlandev em1 up
```

7.8

```
ping 192.168.6.2
```

```
ping 192.168.5.3
```

Ναι μπορούμε.

7.9

```
ifconfig bridge1 -stp em0
```

7.10

```
tcpdump -i em0 -e -vvv -x -l | tee cap1
```

7.11

Καθαρίζω τον πίνακα ARP στο PC2 με την εντολή « **arp -ad** » 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

```
less cap1
```

Έχουν μεγαλύτερο μέγεθος γιατί περιέχουν και το vlan tag (4 bytes).

7.13

Πλέον η τιμή του ethertype είναι 0x8100, ωστόσο παρατηρούμε πως είναι ενθυλακωμένος και ο τύπος που είχαμε δει πιο πριν (ethertype ARP) και (ethertype IPv4) αμέσως μετά.

7.14

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

7.15

```
tcpdump -i em0 -e -vvv -x -l | tee file15.txt
```

7.16

Στο PC3 καθαρίζω τον πίνακα ARP με την εντολή « **arp -ad** » 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 file17.txt) όπως πριν, για την διεπαφή em0 αυτή τη φορά με την εντολή “tcpdump -i em0 -e -vnn -x -l | tee file17.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

tcpdump not stp (+άλλα φίλτρα που ζητήθηκαν).