

Όνοματεπώνυμο: Μοίρας Αλέξανδρος	Όνομα PC: LAPTOP-5A8R1JQR
Ομάδα: 3	Ημερομηνία: 16/3/2022

Εργαστηριακή Άσκηση 3

Τοπικά δίκτυα και μεταγωγείς LAN

Απαντήστε στα ερωτήματα στον χώρο που σας δίνεται παρακάτω και στην πίσω σελίδα εάν δεν επαρκεί. Το φυλλάδιο αυτό θα παραδοθεί στον επιβλέποντα.

1

1.1 `ifconfig em0 inet 192.168.1.1 netmask 255.255.255.0 (PC1)`

`ifconfig em0 inet 192.168.1.2 netmask 255.255.255.0 (PC2)`

1.2 `ifconfig em0 up`

`ifconfig em1 up`

1.3 `ping 192.168.1.2 (στο PC1)`

`ping 192.168.1.1 (στο PC2)`

Παρατηρούμε ότι δεν λαμβάνεται καμία απάντηση στα ICMP Echo request

1.4 Στη γέφυρα `tcpdump -i em0` (αντίστοιχα `em1`). Δεν παράγονται πακέτα icmp στα LAN1, LAN2 καθώς αφού τα VMs βρίσκονται σε διαφορετικά LANs που δεν συνδέονται μεταξύ τους και δεν υπάρχει default gateway σε αυτά, κανένα μηχάνημα δεν απαντά στα ARP Request για τη διεύθυνση του PC2 (αντίστοιχα του PC1) ώστε να παραχθούν ICMP πακέτα.

1.5 `ifconfig bridge0 create`

`ifconfig bridge0 addm em0 addm em1`

`ifconfig bridge0 up`

1.6 Ναι τώρα υπάρχει επικοινωνία

1.7 1 βήμα (φαίνεται να επικοινωνούν απευθείας).

1.8 Ότι τα μηχανήματα έχουν αντιστοιχίσει σωστά τις IP διευθύνσεις του άλλου μηχανήματος με τις MAC τους. (`arp -a`)

1.9 `tcpdump -vvnei em0`

`tcpdump -vvnei em1`

Πράγματι προωθεί πακέτα από τη φυσική διεύθυνση 08:00:27:ec:3d:57 (PC1) στην 08:00:27:32:93:02 (PC2) όσον αφορά τα echo requests και αντίστροφα για τα Echo replies.

1.10 Όχι δεν κάνουν κάποια αλλαγή.

1.11 Όχι δεν αλλάζει.

1.12 Όχι δεν υπάρχει γιατί οι γέφυρες είναι διαφανείς όσον αφορά τη χρήση τους, και το TTL μειώνεται πάνω σε αυτές ώστε να φανεί στην `traceroute`.

1.13 `tcpdump -l em1`

ping 192.168.1.2

ifconfig em0 inet 192.168.2.1 netmask 255.255.255.0

1.14 Η γέφυρα προωθεί τα πακέτα προς το PC2.

1.15 Το ping δεν είναι επιτυχές καθώς το PC2 παύει να απαντά στα echo requests του PC1 καθώς αυτά έχουν άλλη διεύθυνση προορισμού.

1.16 Όχι δεν μπορούμε.

1.17 ifconfig em2 up

ifconfig bridge0 addm em2

1.18 Ναι τώρα λαμβάνουμε.

1.19 Όχι δεν εμφανίζονται καθώς η γέφυρα έχει μάθει που βρίσκονται τα PC1, PC3 και προωθεί την κίνηση προς αυτά στην κατάλληλη διεπαφή της και δε χρησιμοποιεί την τεχνική της πλημμύρας.

1.20 Παρατηρούμε το arp request από το PC1 που αναζητεί τη φυσική διεύθυνση του PC3. Το παρατηρούμε καθώς τα ARP Requests είναι πακέτα εκπομπής που προωθούνται προς όλους τους κόμβους του δικτύου, οπότε η γέφυρα προωθεί το πακέτο και στο LAN1 και στο LAN2.

1.21 ifconfig bridge0

1.22 ifconfig bridge0 addr

1.23 Της em0 στο PC1, της em1 στο PC2 και της em2 στο PC3.

1.24 ifconfig bridge0 flush

1.25 ifconfig bridge0 deletem em2

1.26 ifconfig bridge0 destroy

1.27 ifconfig em0 delete

2

2.1 ifconfig em0 inet 192.168.1.1 netmask 255.255.255.0 (αντίστοιχα και για τους άλλους υπολογιστές)

2.2 ifconfig bridge1 create

ifconfig bridge1 addm em0 addm em1

ifconfig bridge1 up

2.3 ifconfig bridge2 create

ifconfig bridge2 addm em0 addm em1

ifconfig bridge2 up

2.4 ifconfig bridge3 create

ifconfig bridge3 addm em0 addm em1

ifconfig bridge3 up

2.5 PC1: 08:00:27:32:93:02

PC2: 08:00:27:ec:3d:57

PC3: 08:00:27:3b:cd:52

PC4: 08:00:27:3d:9d:22

2.6 ifconfig bridge1 flush. Αντιστοιχα 2,3

2.7 tcpdump

2.8 ifconfig bridge1(2,3) addr

B1:

```
root@PC:~ # ifconfig bridge1 addr
08:00:27:ec:3d:57 Vlan1 em1 1185 flags=0<>
08:00:27:32:93:02 Vlan1 em0 1185 flags=0<>
```

B2:

```
root@PC:~ # ifconfig bridge2 addr
08:00:27:ec:3d:57 Vlan1 em0 1092 flags=0<>
08:00:27:32:93:02 Vlan1 em0 1092 flags=0<>
```

B3:

```
root@PC:~ # ifconfig bridge3 addr
08:00:27:32:93:02 Vlan1 em0 1012 flags=0<>
```

2.9: Το arp request από το PC1 στο PC2 είναι Broadcast και για αυτό το παρατηρούμε σε όλες τις καταγραφές επομένως περνά από όλες τις γέφυρες οπότε η διεύθυνση του PC1 εγγράφεται στους πίνακες προώθησης όλων των γεφυρών. Το arp reply του PC2 στο PC1 είναι unicast και εφόσον οι γέφυρες B1, B2 γνωρίζουν πλέον σε ποια διεπαφή τους να προωθήσουν το πακέτο, η B2 θα το απορρίψει καθώς θα δει ότοι προέρχεται από εκεί που θα έπρεπε να προωθηθεί. Επομένως αυτό το πακέτο δε θα πάει ποτέ στο LNK2 ώστε να δημιουργηθεί εγγραφή για τη διεύθυνση του PC2 στον πίνακα δρομολόγησης B3, ενώ στις B1, B2 θα δημιουργηθεί. Ύστερα το ICMP echo θα προωθηθεί στο LNK1 από το B1 και δε θα προωθηθεί από το B2 όπως εξηγήθηκε παραπάνω και το ICMP reply πάλι δε θα προωθηθεί από το B2 αλλά από τη B1 στο LAN1.

2.10 Όχι δεν υπήρχε γιατί υπάρχει ήδη εγγραφή στο arp table του PC2 για το PC1, οπότε το PC2 δεν έκανε broadcast arp request για να μάθει τη θέση του η B3 και έστειλε κατευθείαν ICMP echo request που δεν προωθήθηκε από τη B2 η οποία γνωρίζει τη θέση του PC1, ώστε να μαθευτεί η θέση του PC2 από τη B3.

2.11 Γιατί οι απαντήσεις του PC4 στο PC2, τόσο το arp reply όσο και το ICMP echo reply πηγαίνοντας στο PC2 φτάνουν στο LNK1, οπότε η B1 αποθηκεύει τη διεύθυνση MAC του PC4

2.12 Ναι τώρα προστέθηκε η MAC διεύθυνση του PC3 σε όλους του πίνακες δρομολόγησης καθώς έγινε ARP request από το PC3 (broadcast) και επίσης προστέθηκε η MAC διεύθυνση του PC2 στη B3 καθώς τα arp και icmp echo replies από το PC2 έφτασαν στο LNK2

2.13 ping 192.168.1.2 (από το PC4)

ping 192.168.1.2 (από το PC1)

2.14 Τίποτα συνεχίζει κανονικά.

2.15 Σταματά να λειτουργεί (να δέχεται απαντήσεις) καθώς η γέφυρα B2 δε θα προωθούσε τα πακέτα προς το PC2 στο LNK2 οπότε δε θα έφταναν ποτέ στο PC2 που έχει μετακινηθεί.

2.16 Ξανααρχίζει να λειτουργεί δηλαδή να δέχεται απαντήσεις. Αυτό συμβαίνει επειδή το icmp echo request από το PC2 φτάνει στο LNK2 για να καταλήξει στο PC3, οπότε η B2 μαθαίνει τη νέα διεύθυνση του PC2 και αρχίζει να προωθεί στο LNK2 τα πακέτα που προορίζονται για αυτό από το PC1 αντί να τα απορρίπτει.

2.17 Θα έπρεπε να περιμένουμε 1200 δευτερόλεπτα από την τελευταία εγγραφή της διεύθυνσης του PC2 στην B2 ώστε να επαναφερθούν η εγγραφή για το PC2 στον πίνακα δρομολόγησής της.

```
timeout 1200 (ifconfig bridge0)
```

3

3.1 ifconfig bridge0 create

```
ifconfig bridge0 addm em0 addm em1
```

```
ifconfig bridge0 up
```

```
ifconfig em0 up
```

```
ifconfig em1 up
```

3.2 ifconfig bridge1 create

```
ifconfig bridge0 addm em0 addm em2
```

```
ifconfig bridge1 up
```

```
ifconfig em0 up
```

```
ifconfig em2 up
```

3.3 PC1: 08:00:27:32:93:02

PC2: 08:00:27:ec:3d:57

PC3: 08:00:27:3b:cd:52

3.4 Εμφανίζεται το arp request που κάνει broadcast το PC2 για να βρει τη MAC διεύθυνση του PC3. Αφού πρόκειται για πακέτο broadcast η κάθε γέφυρα θα το προωθήσει προς όλες τις διεπαφές της πέραν αυτής από όπου προήλθε, οπότε αυτό θα καταλήξει στο PC1. Αφού το πακέτο θα διέλθει και από τις δύο γέφυρες και οι δύο θα μάθουν τη θέση του PC2 (σε ποια διεπαφή τους βρίσκεται). Ύστερα το PC3 θα απαντήσει με arp reply και η B2 γνωρίζοντας που είναι το PC2 δε θα το προωθήσει στο LNK1 και πλέον γνωρίζει και τη θέση του PC3. Πλέον η ανταλλαγή icmp echo request και reply θα γίνεται χωρίς να εμφανίζεται κίνηση στο LAN1.

3.5 ping 192.168.1.1

3.6 ifconfig bridge0 addm em2

```
ifconfig em2 up
```

```
ifconfig bridge1 addm em1
```

```
ifconfig em1 up
```

3.7

```

root@PC:~ # ifconfig bridge0 addr
08:00:27:32:93:02 Vlan1 em0 1182 flags=0<>
08:00:27:3b:cd:52 Vlan1 em1 1182 flags=0<>
08:00:27:ec:3d:57 Vlan1 em1 113 flags=0<>
root@PC:~ # ifconfig bridge1 addr
08:00:27:32:93:02 Vlan1 em0 993 flags=0<>
08:00:27:3b:cd:52 Vlan1 em2 993 flags=0<>
08:00:27:ec:3d:57 Vlan1 em2 0 flags=0<>

```

3.8 B1: PC1: em0 (LAN1) PC3: em1(LNK1)

B2: PC1: em0 (LNK1) PC3: em2(LAN2)

3.9 tcpdump

3.10 arp -ad

Όχι το Ping είναι ανεπιτυχές.

3.11 PC1: διεπαφή em1(LNK2)

PC3: διεπαφή em0(LNK1)

εμφανίζεται το PC3 σε αυτή τη διεπαφή γιατί η B1 προωθεί το arp request (broadcast) που λαμβάνει (που έχει τη MAC του PC3) σε όλες τις διεπαφές της πέραν αυτής από όπου το έλαβε άρα και στο LNK1 ή στο LNK2 όπου θα δημιουργηθεί βρόχος καθώς το arp request θα προωθείται συνεχώς από τη μία γέφυρα στην άλλη. Το PC1 εμφανίζεται στο LNK2 καθώς η B1 λόγω των arp request θα θεωρούσε ότι το PC3 βρίσκεται στο LNK2 οπότε θα προώθησε το arp reply εκεί και η B2 θα αντιστοίχισε αυτή τη διεπαφή της στο PC1. Πλέον και το arp reply θα έχει κολλήσει στον βρόχο.

3.12 Γίνεται η ερώτηση ποιος έχει την IP 192.168.1.1 από τον 192.168.1.3 και δίνεται η απάντηση ότι ο 192.168.1.1 βρίσκεται στη διεύθυνση 08:00:27:32:93:02.

```

23:28:22.155563 ARP, Request who-has 192.168.1.1 tell 192.168.1.3, length 46
23:28:22.155586 ARP, Reply 192.168.1.1 is-at 08:00:27:32:93:02 (oui Unknown), length 28

```

3.13 Γιατί, όπως εξηγήθηκε παραπάνω, η B2 θα προωθήσει το arp request και στο LNK1 και στο LNK2, οπότε μετά η B1, που θα το προωθήσει σε όλες τις θύρες πέραν αυτής από όπου το έλαβε, θα καταλήξει να το προωθήσει πίσω στη B2 μέσω των LNK1, LNK2, η οποία πάλι προωθώντας το σε όλες της διεπαφές της πέραν αυτής που το έλαβε, θα το προωθήσει στο LAN2. Αυτός ο κύκλος θα συνεχιστεί αενάως.

3.14 Τα arp request επαναλαμβάνονται και στην καταγραφή του PC1 για εντελώς αντίστοιχο λόγο με αυτόν που επαναλαμβάνονται στην καταγραφή του PC2 που εξηγήθηκε παραπάνω.

3.15 Γιατί λόγω του βρόχου μεταξύ B1, B2 η B2 δε θεωρεί πλέον, όπως παρατηρήσαμε και παραπάνω, ότι το PC3 βρίσκεται στη διεπαφή προς το LAN2 οπότε δεν προωθεί εκεί τα unicast arp reply πακέτα.

4

4.1 ifconfig bridge0 destroy

ifconfig bridge0 create

ifconfig em0 down

ifconfig em1 down

ifconfig em2 down
αντίστοιχα για B2

4.2 ifconfig em0 up

ifconfig em1 up

ifconfig em2 up

ifconfig lagg0 create

4.3 ifconfig lagg0 up laggport em1 laggport em2

ifconfig lagg0 up

4.4 Εντελώς αντίστοιχα με τα προηγούμενα βήματα.

4.5 ifconfig bridge0 addm em0 addm lagg0

ifconfig bridge0 up

4.6 ifconfig bridge1 addm em2 addm lagg0

ifconfig bridge1 up

4.7 Εμφανίζεται μόνο το arp request που είναι Broadcast, ύστερα η B2 έχει μάθει που βρίσκεται το PC2 και προωθεί μόνο στην αντίστοιχη διεπαφή της το arp reply και πλέον έχει μάθει που βρίσκεται και το PC3 οπότε η κίνηση Icmp echo request-reply Που θα ακολουθήσει θα γίνεται εντός του LAN2.

4.8 tcpdump

4.9 Ναι το ping είναι επιτυχές και παρατηρήσαμε πακέτα ARP στην καταγραφή.

```
root@PC1:~ # tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
00:52:38.612588 ARP, Request who-has 192.168.1.1 tell 192.168.1.3, length 46
00:52:38.612683 ARP, Reply 192.168.1.1 is-at 08:00:27:32:93:02 (oui Unknown), length 28
00:52:38.619120 IP 192.168.1.3 > 192.168.1.1: ICMP echo request, id 8196, seq 0, length 64
00:52:38.619135 IP 192.168.1.1 > 192.168.1.3: ICMP echo reply, id 8196, seq 0, length 64
```

4.10 tcpdump -i em1 (B1), tcpdump -i em1 (B2)

Τα πακέτα εμφανίζονται στη ζεύξη LNK1 καθώς το πρωτόκολλο συνάθροισης είναι το προεπιλεγμένο δηλαδή failover σύμφωνα με το οποίο όλη η κίνηση γίνεται μέσω της master port, δηλαδή ενός συγκεκριμένου link που συνδέεται σε αυτή τη διεπαφή, εκτός και αν αυτό πάψει να είναι διαθέσιμο οπότε η κίνηση γίνεται από την αμέσως επόμενη διαθέσιμη θύρα του aggregated link.

4.11 Παρατηρούμε στην έξοδο της εντολής ping ότι συνεχίζεται η λήψη απαντήσεων icmp echo reply με τα πακέτα τώρα να μεταφέρονται από τη ζεύξη LNK2.

4.12 Τώρα τα πακέτα ICMP μεταφέρονται πάλι από την LNK1 με την εντολή ping να συνεχίζει να λαμβάνει απαντήσεις κανονικά.

5

5.1 ifconfig bridge0 destroy

ifconfig lagg0 destroy

ifconfig em0 down

ifconfig em1 down

ifconfig em2 down

5.2 ifconfig bridge1 create

ifconfig bridge1 addm em0 addm em1 addm em2

ifconfig bridge1 up

ifconfig em0 up

ifconfig em1 up

ifconfig em2 up

5.3 ifconfig bridge2 create

ifconfig bridge2 addm em0 addm em1 addm em2

ifconfig bridge2 up

ifconfig em0 up

ifconfig em1 up

ifconfig em2 up

5.4 `ifconfig bridge1 stp em0 stp em1 stp em2`

5.5 `ifconfig bridge2 stp em0 stp em1 stp em2`

5.6 32768.08:00:27:2e:65:32

32768.08:00:27:56:a7:76

```
root@PC:~ # ifconfig bridge2
bridge2: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
ether 02:76:0f:f3:c5:02
nd6 options=9<PERFORMNUD,IFDISABLED>
groups: bridge
id 08:00:27:56:a7:76 priority 32768 hellotime 2 fwddelay 15
maxage 20 holdcnt 6 proto rstp maxaddr 2000 timeout 1200
root id 08:00:27:2e:65:32 priority 32768 ifcost 20000 port 1
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 alternate state discarding
member: em0 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
ifmaxaddr 0 port 1 priority 128 path cost 20000 proto rstp
role root state forwarding
```

5.7 Η γέφυρα ρίζα του επικαλύπτοντος δέντρου είναι η bridge1 καθώς οι δύο γέφυρες έχουν ίσες προτεραιότητες και η bridge1 έχει μικρότερη MAC address.

5.8 B1: (γέφυρα ρίζα)

Διεπαφή	Ρόλος	Κατάσταση
em0 (προς LAN1)	designated	forwarding
em1 (προς LNK1)	designated	forwarding
em2 (προς LNK2)	designated	forwarding

B2: (για επόμενα ερωτήματα)

Διεπαφή	Ρόλος	Κατάσταση
em0 (προς LNK1)	root	forwarding
em1 (προς LNK2)	alternate	discarding
em2 (προς LAN2)	designated	forwarding

5.9 Ριζική θύρα είναι η em0 (προς LNK1)

5.10 Ο ρόλος της είναι **alternate** (εναλλακτική διαδρομή προς τη ρίζα) και η κατάστασή της είναι **discarding** (δεν αποστέλει πλαίσια-δεδομένα χρηστών).

5.11 Ο ρόλος της διεπαφής στο LAN2 της B2 είναι **designated** και η κατάστασή της είναι **forwarding**

5.12 Κάθε 2 δευτερόλεπτα.

5.13 IEEE 802.3

5.14 MAC πηγής: 08:00:27:3b:27:74 (διεύθυνση MAC της διεπαφής της B1 που στέλνει το BPDU)

MAC προορισμού: 01:80:c2:00:00:00 (multicast address)

5.15 Στην **em1** που είναι η διεπαφή προς το **LNK1**.

5.16 multicast

5.17 root ID: 8000.08:00:27:2e:65:32 ($8000_{16}=32768_{10}$)

bridge ID: 8000.08:00:27:2e:65:32.8002

root path cost: 0

5.18 Τα πρώτα 4 ψηφία της **Bridge ID** είναι η προτεραιότητα με τιμή 8000_{16} για τα πλαίσια **BPDU** που παράγονται σε αμφότερες τις διεπαφές, ενώ τα τελευταία 4 δεκαεξαδικά ψηφία αποτελούν το **ID** της θύρας από την οποία εκπέμπεται το πλαίσιο **BPDU** με τιμές 8002 για τη θύρα στο **LNK1** και τιμή 8003 για τη θύρα στο **LNK2**.

5.19 Όχι δεν παρατηρούμε.

5.20 Στη θύρα προς το **LAN2 (em2)**

5.21 root ID: 8000.08:00:27:2e:65:32 ($8000_{16}=32768_{10}$)

bridge ID: 8000.08:00:27:56:a7:76.8003

root path cost: 20000

5.22 Ναι είναι επιτυχές.

5.23 Περίπου 6-7 δευτερόλεπτα περνούν μέχρι να αποκατασταθεί η επικοινωνία, τιμή αναμενόμενη αφού το πρωτόκολλο **RSTP** τυπικά μπορεί να ανταποκριθεί σε αλλαγές στην τοπολογία σε χρόνο περίπου $3 \cdot \text{hello times}$ ο οποίος είναι 6 seconds για hello time 2 seconds που έχουμε όπως παρατηρήσαμε παραπάνω.

5.24 Όχι δεν υπάρχει καμία διακοπή στην επικοινωνία μετά την επανασύνδεση του καλωδίου.

6

6.1 ifconfig bridge1 addm em3

ifconfig em3 up

ifconfig bridge1 stp em3

6.2 ifconfig bridge2 addm em3

ifconfig em3 up

ifconfig bridge2 stp em3

6.3 ifconfig bridge3 create

ifconfig bridge3 addm em0 addm em1 addm em2

ifconfig bridge3 up

ifconfig em0 up

ifconfig em1 up

ifconfig em2 up

ifconfig bridge3 stp em0 stp em1 stp em2

6.4 ifconfig bridge1 flush

ifconfig bridge2 flush

ifconfig bridge3 flush

Ναι το ping είναι επιτυχές.

6.5 ifconfig bridge1 priority 0

6.6 Pathcosts:

LNK1: 20000

LNK2: 20000

LNK4: 2000000 (θα ήταν down η διεπαφή όταν κάναμε stp).

Στο RSTP η προκαθορισμένη τιμή για το κόστος μίας ζεύξης είναι τα 20Tbps/bandwidth και η ταχύτητα των ζεύξεων είναι 1 Gbps άρα το κόστος κάθε ζεύξης ανέρχεται σε 20000 (αν είναι down είναι 10Mbps)

6.7

Από τη B1: root path cost: 0

Από τη B2: root path cost: 20000

6.8 Η θύρα στο LNK3 γιατί εξετάζοντας το root ID στα πλαίσια BPDU που λαμβάνει από τις άλλες δύο γέφυρες, προκύπτουν ίσα οπότε επιλέγεται η διαδρομή μέσω της γέφυρας που οδηγεί στο μικρότερο κόστος διαδρομής προς τη ρίζα που είναι η B1 βάσει του προηγούμενου ερωτήματος και αφού η B3 συνδέεται με ισοδύναμες ζεύξεις με τα B1, B2.

6.9 Ο ρόλος της άλλης θύρας είναι designated και η κατάστασή της είναι forwarding.

6.10 20000

6.11 ping 192.168.1.3

6.12 ifconfig bridge3 ifpathcost em0 400001. Επιλέξαμε την τιμή 40001 διότι το κόστος του μονοπατιού προς τη B1 μέσω της LNK4 είναι $20000+20000=40000$ και πρέπει να είναι μικρότερο του κόστους του μονοπατιού μέσω της LNK3, ώστε να γίνει ριζική θύρα η διεπαφή της στο LNK4 όπως εξηγήθηκε παραπάνω.

6.13 Περίπου 5 δευτερόλεπτα.

6.14 Ο ρόλος της διεπαφής της Bridge3 στο LNK3 είναι alternate και η κατάστασή της είναι discarding.

6.15 Όχι δεν υπήρξε.

6.16 Ναι υπήρξε, τώρα το root path cost των BPDUs που παράγει η bridge3 είναι 40000 αντί για 20000 που ήταν πριν

6.17 Περίπου 9 δευτερόλεπτα.

6.18 Δεν διακόπηκε καθόλου η επικοινωνία ή επαναφέρθηκε πολύ γρήγορα και δεν έγινε αισθητή η διακοπή.

7

7.1 ifconfig em0.1 create vlan 5 vlandev em0 inet 192.168.5.1/24

ifconfig em0.2 create vlan 6 vlandev em0 inet 192.168.6.1/24

7.2 ifconfig em0.1 create vlan 6 vlandev em0 inet 192.168.6.2/24

7.3 ifconfig em0.1 create vlan 5 vlandev em0 inet 192.168.5.1/24

7.4 Ναι μπορούμε γιατί το PC1 ανήκει στα VLAN που ανήκουν οι υπολογιστές με αυτές τις δύο διευθύνσεις.

7.5 Όχι δεν μπορούμε καθώς το PC2 δε διαθέτει διεπαφή που να ανήκει στο VLAN 5, που ανήκει η διεπαφή του PC3 με διεύθυνση 192.168.5.3.

7.6 Όχι δεν μπορούμε καθώς το PC3 δε διαθέτει διεπαφή που να ανήκει στο VLAN 6, που ανήκει η διεπαφή του PC2 με διεύθυνση 192.168.6.2.

7.7 ifconfig bridge1 -stp em0

7.8 tcpdump -vnxvxei em0

7.9 Ethertype:

ARP: 0x0806

IPv4: 0x0800

7.10 Τώρα η επικεφαλίδα Ethernet περιέχει και ετικέτα VLAN Tag των 4 Bytes που βρίσκεται στη θέση του πεδίου τύπος του Ethernet (Μετά τις διευθύνσεις προορισμού και πηγής). Ο τύπος του πλαισίου ακολουθεί.

```
11:49:33.388442 08:00:27:32:93:02 (oui Unknown) > 08:00:27:ec:3d:57 (oui Unknown), ethertype ARP (0x0806), length 42: Ethernet (len 6), IPv4 (len 4), Reply 192.168.1.1 is-at 08:00:27:32:93:02 (oui Unknown), length 28
    0x0000: 0800 27ec 3d57 0800 2732 9302 0806 0001
    0x0010: 0800 0604 0002 0800 2732 9302 c0a8 0101
    0x0020: 0800 27ec 3d57 c0a8 0102
11:49:33.390019 08:00:27:ec:3d:57 (oui Unknown) > 08:00:27:32:93:02 (oui Unknown), ethertype IPv4 (0x0800), length 98: (tos 0x0, ttl 64, id 31939, offset 0, flags [none], proto ICMP (1), length 84)
    192.168.1.2 > 192.168.1.1: ICMP echo request, id 1283, seq 0, length 64
    0x0000: 0800 2732 9302 0800 27ec 3d57 0800 4500
    0x0010: 0054 7cc3 0000 4001 7a92 c0a8 0102 c0a8
    0x0020: 0101 0800 6a6d 0503 0000 6231 cecd 0002
    0x0030: 6c8b 0809 0a0b 0c0d 0e0f 1011 1213 1415
    0x0040: 1617 1819 1a1b 1c1d 1e1f 2021 2223 2425
    0x0050: 2627 2829 2a2b 2c2d 2e2f 3031 3233 3435
```

7.11 Η τιμή του πεδίου Ethertype τώρα είναι 8100 (Tag Protocol ID- 2 πρώτα Bytes - του VLAN Tag 802.1Q). Τώρα τα πακέτα ARP ξεχωρίζουν από τα IPv4 από το πεδίο τύπος που ακολουθεί της ετικέτας VLAN Tag. (κόκκινα βέλη)

```

11:54:30.677842 08:00:27:32:93:02 (oui Unknown) > 08:00:27:ec:3d:57 (oui Unknown), ethertype 802.1Q (0x8100), length 46: vlan 6, p 0, ethertype ARP, Ethernet (len 6), IPv4 (len 4), Reply 192.168.6.1 is-at 08:00:27:32:93:02 (oui Unknown), length 28
    0x0000: 0800 27ec 3d57 0800 2732 9302 8100 0006
    0x0010: 0806 0001 0800 0604 0002 0800 2732 9302
    0x0020: c0a8 0601 0800 27ec 3d57 c0a8 0602
11:54:30.684176 08:00:27:ec:3d:57 (oui Unknown) > 08:00:27:32:93:02 (oui Unknown), ethertype 802.1Q (0x8100), length 102: vlan 6, p 0, ethertype IPv4, (tos 0x0, ttl 64, id 31940, offset 0, flags [none], proto ICMP (1), length 84)
    192.168.6.2 > 192.168.6.1: ICMP echo request, id 2819, seq 0, length 64
    0x0000: 0800 2732 9302 0800 27ec 3d57 8100 0006
    0x0010: 0800 4500 0054 7cc4 0000 4001 7091 c0a8
    0x0020: 0602 c0a8 0601 0800 f00e 0b03 0000 6231
    0x0030: cff6 0006 dfbc 0809 0a0b 0c0d 0e0f 1011
    0x0040: 1213 1415 1617 1819 1a1b 1c1d 1e1f 2021
    0x0050: 2223 2425 2627 2829 2a2b 2c2d 2e2f 3031
    0x0060: 3233 3435 3637

```

7.12 Στα δύο τελευταία Bytes της ετικέτας VLAN Tag (μπλε βελάκι στο πάνω σχήμα)

7.13 tcpdump -vnnxxei em0.5

7.14 Ethertype:

ARP: 0x0806

Πακέτα που μεταφέρουν μηνύματα ICMP: 0x0800

Όχι δεν υπάρχει πεδίο σχετικό με το VLAN (έχει αφαιρεθεί από τη διεπαφή em0 πριν προωθηθεί στην em0.5)

7.15 ifconfig bridge1 stp em0

tcpdump -vnnxxei em0

7.16 Όχι δεν είναι ίδιου τύπου. Είναι πλαίσια IEEE 802.3 και στη θέση του πεδίου Ethertype υπάρχει το μήκος του πακέτου.

7.17 Θα χρησιμοποιούσαμε το φίλτρο 'ether proto not \stp'