

ΓΕΩΡΓΑΚΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ 03120827

ΑΝΑΦΟΡΑ 3ΗΣ ΕΡΓΑΣΤΗΡΙΑΚΗΣ ΑΣΚΗΣΗΣ

**Ομάδα:** 1

**Λογισμικό:** Linux Ubuntu 22.04

**Όνομα PC:** glaptop

### **ΑΣΚΗΣΗ 1: Γέφυρα - Διασύνδεση δύο LAN**

**1.1** Θα χρησιμοποιήσω τις εντολές:

`ifconfig em0 inet 192.168.1.1`

`ifconfig em0 inet 192.168.1.2`

**1.2** Θα χρησιμοποιήσω τις εντολές:

`ifconfig em0 up`

`ifconfig em1 up`

**1.3** Τα PC1 και PC2 δεν επικοινωνούν, καθώς βρίσκονται σε διαφορετικά LAN

**1.4** Οχι, κάνοντας `tcpdump` στο B1 παρατηρώ ότι παράγονται μόνο πακέτα `arp`:  
`ARP, Request who has 192.168.1.2 tell 192.168.1.1`

**1.5** Θα χρησιμοποιήσω τις εντολές:

`ifconfig bridge0 create`

`ifconfig bridge0 addm em0 addm em1 up`

**1.6** Ναι, τώρα υπάρχει επικοινωνία

**1.7** Παρατηρώ `TTL=64`. Αρα τα PC1 και PC2 είναι 1 βήμα μακριά  
(η γέφυρα δεν μετρίεται ως βήμα, καθώς δεν μειώνει το `TTL`)

**1.8** Στους πίνακες `arp` αναγράφονται οι IP και οι MAC διευθύνσεις των PC1 και PC2 (συμφωνούν μεταξύ τους). Στο PC1 η εγγραφή για το PC2 έχει `expiration date`. Αντίστοιχα για το PC2

**1.9** Οντως, παρατηρώντας τις επικεφαλίδες με το `option -e` βλέπω ότι περιέχονται οι σωστές MAC διευθύνσεις και ότι οντως το B1 προωθεί τα πλαίσια `ethernet`

**1.10** Οχι, δεν κανει καποια αλλαγή

**1.11** Οχι, δεν κανει καποια αλλαγή

**1.12** Οχι δεν φαίνεται το B1, καθώς η `traceroute` βασίζεται στο `TTL` και το B1 δεν το επηρεάζει. Η εντολή επιτυγχάνει για `TTL=1`

**1.13** Θα χρησιμοποιήσω τις εντολές:

`ping 192.168.1.2 (στο PC1)`

`tcpdump -e -i em1 (στο B1)`

**1.14** Ναι, η γέφυρα συνεχίζει να προωθεί τα `ICMP` πακέτα στο PC2 μετά την αλλαγή διεύθυνσης, καθώς υπάρχει αυτή η διεύθυνση στον πίνακα της

(βλεπουμε δηλαδή τα μηνύματα ICMP requests). Ωστόσο αυτά τα πακέτα δεν λαμβάνονται απο κανενάν

- 1.15 Οχι, το ping δεν ήταν επιτυχές. Είχε 65.5% packet loss το οποίο συνέβη αφού αλλάξαμε την IP του PC2, αφού δεν υπήρχε καποιος στην 192.168.1.2 να απαντήσει
- 1.16 Οχι, δεν μπορώ ακόμα να κάνω ping απο το PC3 στο PC1
- 1.17 Θα χρησιμοποιήσω την εντολή:  
ifconfig bridge0 addm em2 up
- 1.18 Τώρα το ping πετυχαίνει
- 1.19 Οχι, δεν εμφανίζονται πακέτα ICMP στο LAN2. Αυτό είναι απολύτως λογικό καθώς η γέφυρα, έχοντας συμπληρώσει τον πίνακα προώθησης της, ξέρει που πρέπει να στείλει τα μηνύματα.
- 1.20 Σε αυτή την περίπτωση κατέγραφα ένα πακέτο σχετικό με το ping:  
ARP broadcast who has 192.168.1.1 tel 192.168.1.3  
(έκανα ping απο το PC3 στο PC1)
- 1.21 Με την εντολή: ifconfig bridge0
- 1.22 Με την εντολή: ifconfig bridge0 addr
- 1.23 Τα μηχανήματα που αντιστοιχούν στις εγγραφές του πίνακα προώθησης είναι τα PC1 και PC3
- 1.24 Θα χρησιμοποιήσω την εντολή: ifconfig bridge0 flush
- 1.25 Θα χρησιμοποιήσω την εντολή: ifconfig bridge0 deletem em2
- 1.26 Θα χρησιμοποιήσω την εντολή: ifconfig bridge0 destroy
- 1.27 Θα χρησιμοποιήσω την εντολή: ifconfig em0 remove

## **ΑΣΚΗΣΗ 2: Αυτο-εκπαίδευση γεφύρων**

- 2.1 Θα χρησιμοποιήσω την εντολή: ifconfig em0 inet 192.168.1.X
- 2.2 Θα χρησιμοποιήσω τις εντολές:  
ifconfig bridge1 create  
ifconfig bridge1 addm em0 addm em1 up
- 2.3 Θα χρησιμοποιήσω τις εντολές:  
ifconfig bridge2 create  
ifconfig bridge2 addm em0 addm em1 up
- 2.4 Θα χρησιμοποιήσω τις εντολές:  
ifconfig bridge3 create  
ifconfig bridge3 addm em0 addm em1 up
- 2.5 Οι MAC διευθύνσεις είναι οι εξής:  
PC1: 08:00:27:f1:5e:ab  
PC2: 08:00:27:ff:52:67  
PC3: 08:00:27:eb:93:c4  
PC4: 08:00:27:4e:b8:3a  
Για τον καθαρισμό των arp tables χρησιμοποιώ την arp -d -a
- 2.6 Θα χρησιμοποιήσω την εντολή: ifconfig bridgeX flush
- 2.7 Θα χρησιμοποιήσω την εντολή: tcpdump
- 2.8 Οι νέοι πίνακες προώθησης είναι:  
B1: 08:00:27:ff:52:67 em1 (δλδ το PC2)

08:00:27:f1:5e:ab em0 (δλδ το PC1)

B2: 08:00:27:ff:52:67 em0 (δλδ το PC2)

08:00:27:f1:5e:ab em0 (δλδ το PC1)

B3: 08:00:27:f1:5e:ab em0 (δλδ το PC1)

- 2.9** Αρχικά, το PC1 στέλνει ARP request για να μάθει την MAC του PC2. Αυτό το μήνυμα επαναστέλνεται απο όλες τις γέφυρες και έτσι όλες ξέρουν που είναι το PC1. Το PC2 απαντά με ARP reply το οποίο στέλνεται μόνο προς το PC1. Έτσι τα B1 και B2 που βρίσκονται στο ίδιο LNK1 με το PC2 μαθαίνουν για την MAC του. Το B2 δεν προώθησε αυτό το μήνυμα, επομένως το B3 δεν ξέρει για το PC2
- 2.10** Οχι, δεν υπήρξαν αλλαγές στους πίνακες προώθησης. Το PC2 είχε την MAC του PC1 και έτσι, η γέφυρα B2 δεν ξαναέστειλε πουθενά το μήνυμα και η B1 το έστειλε στο PC1 που ήδη είχε.
- 2.11** Τώρα, το B1 περιέχει την MAC του PC4 γιατί όταν το PC4 έστειλε το ARP reply που πήγαινε στο PC2, το B1 το πήρε (απο το LNK1) και το αποθήκευσε στον πίνακα, παρότι δεν προώθησε το μήνυμα
- 2.12** Ναι, υπάρχουν αλλαγές. Τώρα πλέον όλες οι γέφυρες περιέχουν εγγραφές και για τα 4 PC. Αυτό συνέβη γιατί, τα B2 και B3 αποθήκευσαν την MAC του PC3 απο το ARP request που έστειλε αυτό (μεσω του LNK2) και το B2 προώθησε το πακέτο στο B1.
- 2.13** Θα χρησιμοποιήσω τις εντολές:  
ping 192.168.1.2 (στο PC4)  
ping 192.168.1.2 (στο PC1)
- 2.14** Το ping πετυχαίνει καθώς τα PC2 και PC4 είναι στο ίδιο δίκτυο (LAN2)
- 2.15** Το ping σταματάει (60.7% packet loss) καθώς το B2 δεν μεταφέρει το μήνυμα απο το PC1 στο PC2
- 2.16** Μετά το ping απο το PC2 στο PC3, ο πίνακας του B2 ανανεώθηκε (για την θέση του PC2) και έτσι το ping απο το PC1 στο PC2 αρχίζει και πάλι
- 2.17** Θα έπρεπε να περιμένουμε μέχρι να λήξει η εγγραφή για το PC2 στον πίνακα προώθησης του B2

### **ΑΣΚΗΣΗ 3: Καταιγίδα πλαισίων εκπομπής**

- 3.1** Θα χρησιμοποιήσω τις εντολές:  
ifconfig bridge1 create  
ifconfig bridge1 addm em0 addm em1 up
- 3.2** Θα χρησιμοποιήσω τις εντολές:  
ifconfig bridge2 create  
ifconfig bridge2 addm em0 addm em2 up
- 3.3** Οι MAC διευθύνσεις είναι οι εξής:  
PC1: 08:00:27:f1:5e:ab  
PC2: 08:00:27:ff:52:67  
PC3: 08:00:27:eb:93:c4  
Για τον καθαρισμό των arp tables χρησιμοποιώ την arp -d -a
- 3.4** Ναι, παρατηρώ arp request μήνυμα στο LAN1 το οποίο προκύπτει αφού το B1 προωθεί το μήνυμα που προώθησε το B2 που δημιούργησε το PC2

- κατα το ping για να μάθει την MAC του PC3
- 3.5** Θα χρησιμοποιήσω την εντολή: ping 192.168.1.1
- 3.6** Θα χρησιμοποιήσω τις εντολές:  
ifconfig bridge1 addm em2  
ifconfig bridge2 addm em1  
(Εφόσον οι διεπαφές ήταν ήδη up)
- 3.7** Θα χρησιμοποιήσω την εντολή: ifconfig bridgeX addr
- 3.8** Οι νέοι πίνακες προώθησης είναι:  
B1: 08:00:27:f1:5e:ab em0 (δλδ το PC1)  
08:00:27:eb:93:c4 em1 (δλδ το PC3)  
08:00:27:ff:52:67 em1 (δλδ το PC2)  
(Υπενθυμίζω ότι em1 = LNK1 και em0 = LAN1)  
B2: 08:00:27:f1:5e:ab em0 (δλδ το PC1)  
08:00:27:eb:93:c4 em2 (δλδ το PC3)  
08:00:27:ff:52:67 em2 (δλδ το PC2)  
(Υπενθυμίζω ότι em0 = LNK1 και em2 = LAN2)  
Επομένως οι MAC των PC1 και PC3 εμφανίζονται στις διεπαφές που σχετίζονται με το LNK1 μόνο
- 3.9** Θα χρησιμοποιήσω την εντολή: tcpdump -e
- 3.10** Οχι, το ping δεν είναι επιτυχές (παρότι το PC1 απαντά στο arp μήνυμα). Παρατηρώ έναν καταιγισμό arp request μηνυμάτων, ο οποίος προκύπτει από τις επαναπροωθήσεις των B1 και B2
- 3.11** Οι MAC διευθύνσεις των PC1 και PC3 εμφανίζονται στην διεπαφή em1 του B2 (δλδ στο LNK2). Αυτό συμβαίνει γιατί λόγω της ταυτόχρονης λειτουργίας των LNK1 και LNK2, το B2 λαμβάνει το arp request του PC3 από την em1 (ουσιαστικά επειδή υπάρχει κύκλος)
- 3.12** Κατα τον κατακλυσμό:  
ARP Request: Who has 192.168.1.1 tell 192.168.1.3  
ARP Reply: 192.168.1.1 is-at 08:00:27:f1:5e:ab
- 3.13** Η διεύθυνση πηγής σε αυτά τα μηνύματα είναι: 08:00:27:eb:93:c4 (PC3)
- 3.14** Λόγω της ταυτόχρονης λειτουργίας των LNK1 και LNK2, τα B1 και B2 επαναπροωθούν συνεχώς αυτά τα μηνύματα μεταξύ τους και άρα και σε όλο το υπόλοιπο δίκτυο
- 3.15** Λόγω της συνθήκης που περιέγραψα στο ερώτημα 3.11, το B2 στέλνει το ARP Reply εκεί που νομίζει ότι είναι το PC3. Επομένως το μήνυμα αυτό δεν φτάνει ποτέ πραγματικά στο PC3

#### **ΑΣΚΗΣΗ 4: Συνάθροιση ζεύξεων**

- 4.1** Θα χρησιμοποιήσω τις εντολές:  
ifconfig bridgeX destroy (Για τα B1 και B2)  
ifconfig emX down (Για τις em0, em1 και em2 κάθε μηχανήματος)  
ifconfig bridgeX create (Για τα B1 και B2)
- 4.2** Θα χρησιμοποιήσω τις εντολές:  
ifconfig emX up (Για τις em0, em1 και em2 )  
ifconfig lag0 create

- 4.3** Θα χρησιμοποιήσω τις εντολές:  
ifconfig lagg0 up laggport em1  
ifconfig lagg0 up laggport em2
- 4.4** Οι εντολές είναι ίδιες
- 4.5** Θα χρησιμοποιήσω την εντολή:  
ifconfig bridge1 addm em0 addm lagg0 up
- 4.6** Θα χρησιμοποιήσω την εντολή:  
ifconfig bridge2 addm em2 addm lagg0 up
- 4.7** Ναι, εμφανίζεται ARP Request μήνυμα: Who has 192.168.1.3.  
Αυτό συμβαίνει καθώς το PC2 ψάχνει την MAC του PC3 και τα B1 και B2 επαναπροωθούν αυτό το μήνυμα (μέσω των lagg0)
- 4.8** Θα χρησιμοποιήσω την εντολή: tcpdump -e
- 4.9** Ναι, το ping είναι επιτυχές. Παρατηρώ 2 μηνύματα ARP στην καταγραφή (ένα ARP request και το αντίστοιχο ARP reply)
- 4.10** Τα πακέτα ICMP εμφανίζονται μόνο στην LNK1 ζευξη. Αυτό συμβαίνει λόγω του πρωτοκόλλου συνάθροισης που χρησιμοποιούμε το οποίο είναι το failover (default). Σε αυτό το πρωτόκολλο, χρησιμοποιείται η master ζευξη (κατα την σειρά που προστέθηκαν) και αν αυτή δεν λειτουργεί τότε χρησιμοποιείται η επόμενη
- 4.11** Το ping σταματάει όταν απενεργοποιώ την μια διεπαφή που ακουει το LNK1 (καθως έβγαλα την em1 του B1) αλλά όταν βγάλω και άλλη διεπαφή (em0 του B2) το ping συνεχίζει κανονικά (καθώς τώρα γίνεται μέσω του LNK2)
- 4.12** Το ping συνεχίζει και πάλι μέσω της LNK1 (καθώς σε αυτή είναι συνδεδεμένες οι master διεπαφές)

## **ΑΣΚΗΣΗ 5: Αποφυγή βρόχων**

- 5.1** Θα χρησιμοποιήσω τις εντολές:  
ifconfig bridgeX destroy (Για τα B1 και B2)  
ifconfig lagg0 destroy  
ifconfig emX down (Για τις em0, em1 και em2 κάθε μηχανήματος)
- 5.2** Θα χρησιμοποιήσω τις εντολές:  
ifconfig bridge1 create  
ifconfig emX up (Για τις em0, em1 και em2 )  
ifconfig bridge1 addm em0 addm em1 addm em2 up
- 5.3** Θα χρησιμοποιήσω τις εντολές:  
ifconfig bridge1 create  
ifconfig emX up (Για τις em0, em1 και em2 )  
ifconfig bridge1 addm em0 addm em1 addm em2 up
- 5.4** Θα χρησιμοποιήσω την εντολή:  
ifconfig bridge1 stp em0 stp em1 stp em2
- 5.5** Θα χρησιμοποιήσω την εντολή:  
ifconfig bridge2 stp em0 stp em1 stp em2
- 5.6** Θα χρησιμοποιήσω την εντολή: ifconfig bridgeX | grep id  
B1: 08:00:27:21:df:0d

B2: 08:00:27:d8:53:bc

**5.7** Η γέφυρα ρίζα του επικαλύπτοντος δέντρου είναι η B1  
(μικρότερο BridgeID καθώς έχουν ίδια προτεραιότητα)  
(επίσης φαίνεται και απο το root id)

**5.8** Γέφυρα ρίζα:

B1: em0: role = designated / state = forwarding  
em1: role = designated / state = forwarding  
em2: role = designated / state = forwarding

**5.9** Η ριζική θύρα του B2 είναι η em0:

em0: role = root / state = forwarding (δλδ με το LNK1)

**5.10** Η άλλη διεπαφή (δλδ με το LNK2) έχει τα εξής χαρακτηριστικά:

em1: role = alternate / state = discarding

**5.11** Η τρίτη διεπαφή του B2 (δλδ με το LAN2) έχει τα εξής χαρακτηριστικά:

em2: role = designated / state = forwarding

**5.12** Απο το hello-time βλέπουμε οτι εκπόμενται BPDUs κάθε 2 δευτερόλεπτα

Χρησιμοποίησα την tcpdump -e -v -i em0 (στο B1)

**5.13** Χρησιμοποιείται ενθυλάκωση IEEE 802.3

**5.14** Για το BPDUs ισχύει οτι:

MAC source: 08:00:27:21:df:0d (BridgeID)

MAC destination: 01:80:c2:00:00:00

**5.15** Η MAC πηγής των BPDUs ανήκει στην διεπαφή em0 (LAN1)

**5.16** Η διεύθυνση προορισμού είναι multicast διεύθυνση

**5.17** Στα πλαίσια που κατέγραψα:

root ID: 8000.08:00:27:21:df:0d

bridge ID: 8000.08:00:27:21:df:0d.8001

root path cost: 0

**5.18** Τώρα τα πλαίσια έχουν:

root ID: 8000.08:00:27:21:df:0d

bridge ID: 8000.08:00:27:21:df:0d.8002

root path cost: 0

Τα 2 αυτά μηνύματα έχουν ίδια προτεραιότητα, επομένως αυτή είναι τα πρώτα 2 bytes (δηλαδή το 8000)

**5.19** Το δεύτερο μέρος της BridgeID είναι η MAC της θύρας

**5.20** Η τιμή του portID προκύπτει απο το port identifier και το port ID

**5.21** Οχι, δεν παρατηρώ παραγωγή κίνησης απο την άλλη γέφυρα  
(Η τοπολογία δεν έχει αλλάξει)

**5.22** Παρατηρώ παραγωγή κίνησης BPDUs στην θύρα em2 του B2 (που είναι συνδεδεμένο στο LAN2).

**5.23** Σε αυτά τα BPDUs ισχύει οτι:

root ID: 8000.08:00:27:21:df:0d

bridge ID: 8000.08:00:27:d8:53:bc.8003

root path cost: 2000

**5.24** Ναι, το ping πετυχαίνει

**5.25** Παρατηρώντας το ping βλέπω οτι χρειάζεται περίπου 7 δευτερόλεπτα για να αποκατασταθεί η σύνδεση (καθώς χάθηκαν 7 πακέτα τα οποία

στέλνονται ανα 1 sec). Η τιμή αυτή είναι φυσιολογική καθώς χρειάζεται χρόνος για να ανταποκριθεί το δέντρο σε μια τοπολογική αλλαγή (και επίσης στο RSTP κάθε μήνυμα στέλνεται ανα 2 δευτερόλεπτα)

**5.26** Ναι και πάλι υπάρχει διακοπή (αλλά σημαντικά μικρότερη, καθώς χάθηκαν μόνο 2 πακέτα)

## **ΑΣΚΗΣΗ 6: Ένα πιο πολύπλοκο δίκτυο με εναλλακτικές διαδρομές**

**6.1** Θα χρησιμοποιήσω τις εντολές:

`ifconfig bridge1 addm em3`

`ifconfig bridge1 stp em3`

**6.2** Θα χρησιμοποιήσω τις εντολές:

`ifconfig bridge2 addm em3`

`ifconfig bridge2 stp em3`

**6.3** Θα χρησιμοποιήσω τις εντολές:

`ifconfig bridge3 create`

`ifconfig bridge3 addm em0 addm em1 addm em2`

`ifconfig bridge3 stp em0 stp em1 stp em2`

**6.4** Θα χρησιμοποιήσω την εντολή: `ifconfig bridgeX flush` (Για τα B1 και B2)

Ναι, το ping είναι επιτυχές

**6.5** Θα χρησιμοποιήσω την εντολή: `ifconfig bridge1 priority 0`

**6.6** Και για τις 3 ζεύξεις το path cost είναι 20000. Αυτό προκύπτει απο το γεγονός ότι bandwidth = 1Gbps

**6.7** Για να δω αυτά τα πλαίσια θα χρησιμοποιήσω την `tcpdump` στο B3.

Απο το B1 το πλαίσιο γράφει: root path cost = 0

Απο το B2 το πλαίσιο γράφει: root path cost = 20000

Με βάση αυτά τα δυο καλείται να αποφασίσει ποια θα διαλέξει

**6.8** Η ριζική θύρα είναι η em0 (στο LNK3) λόγω του μικρότερου root path cost

**6.9** Για την θύρα που δεν είναι ριζική ισχύει οτι:

role = designated / state = forwarding

Η αντίστοιχη στο B2 είναι:

role = alternate / state = discarding

**6.10** Εδώ ισχύει οτι root path cost = 40000

(Χρησιμοποίησα την εντολή `tcpdump -i em2 -v` στο B3)

**6.11** Θα χρησιμοποιήσω την εντολή: `ping 192.168.1.3`

**6.12** Θα χρησιμοποιήσω την εντολή:

`ifconfig bridge3 ifpathcost em0 60000`

Χρησιμοποιώ την τιμή 60000 γιατί θέλω μεγαλύτερη τιμή απο το 40000 (που είναι η πραγματική απόσταση)

**6.13** Πέρασαν περίπου 2 δευτερόλεπτα (Χάθηκαν 2 πακέτα κατα το ping)

**6.14** Μετά απο αυτη την αλλαγή ισχύει οτι:

B2: em2 (LNK4): role = designated / state = forwarding

B3: em0 (LNK3): role = alternate / state = discarding

**6.15** Οχι, δεν υπήρξε

**6.16** Ναι, υπάρχει διαφορά (σχετικά με το root path cost)

**6.17** Χρειάστηκαν περίπου 8 δευτερόλεπτα (σύμφωνα με το ping)

**6.18** Χρειάστηκαν περίπου 4 δευτερόλεπτα (σύμφωνα με το ping)

**6.19** Πλέον για το B3 ισχύει ότι:

em2: role = designated / state = forwarding

em3: role = backup / state = discarding

(Υπενθύμιση: Πρόσθεσα την em3)

**6.20** Θα χρησιμοποιήσω την εντολή:

ifconfig bridge3 ifpathcost em0 40000

Μπορώ να διαλέξω οποιαδήποτε τιμή μικρότερη ή ίση του 40000

## **ΑΣΚΗΣΗ 7: Εικονικά τοπικά δίκτυα (VLAN)**

**7.1** Θα χρησιμοποιήσω τις εντολές:

ifconfig em0.5 create vlan 5 vlandev em0 inet 192.168.5.1

ifconfig em0.6 create vlan 6 vlandev em0 inet 192.168.6.1

**7.2** Θα χρησιμοποιήσω τις εντολές:

ifconfig em0.5 create vlan 5 vlandev em0 (για το LAN1)

ifconfig em0.6 create vlan 6 vlandev em0 (για το LAN1)

**7.3** Θα χρησιμοποιήσω τις εντολές:

ifconfig em1.6 create vlan 6 vlandev em1 (για το LNK1)

ifconfig em3.5 create vlan 5 vlandev em3 (για το LNK3)

**7.4** Θα χρησιμοποιήσω την εντολή:

ifconfig em0.6 create vlan 6 vlandev em0 inet 192.168.6.2

**7.5** Θα χρησιμοποιήσω τις εντολές:

ifconfig em0.6 create vlan 6 vlandev em0 (για το LNK1)

ifconfig em3.6 create vlan 6 vlandev em3 (για το LAN2)

(Σημείωση: Σε όλα τα ερωτήματα, για την ενεργοποίηση των διεπαφών χρησιμοποιώ την εντολή ifconfig emX up)

**7.6** Θα χρησιμοποιήσω την εντολή:

ifconfig em0.5 create vlan 5 vlandev em0 inet 192.168.5.3

**7.7** Θα χρησιμοποιήσω τις εντολές:

ifconfig em0.5 create vlan 5 vlandev em0 (για το LNK3)

ifconfig em2.5 create vlan 5 vlandev em2 (για το LAN3)

**7.8** Ναι, τα ping είναι πετυχημένα

**7.9** Θα χρησιμοποιήσω την εντολή:

ifconfig bridge1 -stp em0

**7.10** Θα χρησιμοποιήσω την εντολή:

tcpdump -v -e -x

**7.11** Παρατηρώ ότι:

ARP: Ethertype = 0x0806

IPv4: Ethertype = 0x0800

**7.12** Τα πλαίσια ethernet διαφέρουν σε μέγεθος

(46 αντι για 42 bytes λόγω του VLAN tag που έχει μέγεθος 4 bytes)

**7.13** Τώρα και στις δυο περιπτώσεις (ARP και IPv4) το Ethertype είναι 0x8100

(0x8100 = 801.2Q). Τα ARP ξεχωρίζουν από τα IPv4 μηνύματα, από τα επόμενα 4 byte καθώς ο τύπος του αρχικού πλαισίου Ethernet ακολουθεί αμέσως μετά την ετικέτα 802.1Q



**7.14** Η πληροφορία VLAN εμφανίζεται αμέσως μετά την διεύθυνση πηγής

**7.15** Θα χρησιμοποιήσω την εντολή:

```
tcpdump -i em0.5 -e -v -x
```

**7.16** Τώρα ισχύει ότι:

Ethertype = 0x0806 (ARP) / Ethertype = 0x0800 (IPv4)

Δεν υπάρχει κάποιο πεδίο σχετικό με το VLAN

**7.17** Θα χρησιμοποιήσω τις εντολές:

```
ifconfig bridge1 stp em0
```

```
tcpdump -i em0 -e -v -x
```

**7.18** Όχι τα πλαίσια τώρα έχουν μεγαλύτερο μήκος. Επίσης εκεί που πριν παρατηρούσαμε το Ethertype (δηλαδή μετά το source MAC address) τώρα παρατηρούμε απλά την τιμή 802.3

**7.19** Θα χρησιμοποιούσα το φίλτρο not stp. Η εντολή δηλαδή θα ήταν:

```
tcpdump not stp -i em0.5 -e -v -x
```