

# ΕΡΓΑΣΤΗΡΙΟ ΔΙΚΤΥΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

## 3<sup>η</sup> ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ΠΑΝΑΓΙΩΤΗΣ ΠΑΠΑΔΕΑΣ

A.M el18039

### ΑΣΚΗΣΗ 1

1.1 Στατικά: `ifconfig em0 192.168.1.1/24`, `ifconfig em0 192.168.1.2/24`

1.2 `ifconfig em0 up`, `ifconfig em1 up`

1.3 δεν στέλνονται τα πακέτα

1.4 δεν παράγονται πακέτα icmp αλλά μόνο arp requests τα οποία δεν λαμβάνουν απάντηση εφόσον ανήκουν σε διαφορετικά δίκτυα οι δύο υπολογιστές

1.5 `ifconfig bridge create`

`ifconfig bridge0 addm em0 addm em1 up`

1.6 Τώρα το PC1 επικοινωνεί με το PC2

1.7 `ttl=64`, δεν άλλαξε η τιμή του άρα δεν έκανε hop καθώς δεν υπάρχει κάποιος δρομολογητής στη διαδρομή του

1.8 Οι MAC των PC1 και PC2 έχουν ίδιες τις 3 πρώτες δυάδες (08:00:27)

1.9 `traceroute -l > lan1`, `traceroute -i em1 -l > lan2`

και παρατηρούμε τα πακέτα icmp που στέλνονται από το PC1 στο PC2

1.10 διατηρεί τις Ipv4 των vms

1.11 Όχι

1.12 Δεν υπάρχει ένδειξη του B1 καθώς η λειτουργία της γέφυρας είναι διαφανής και οι υπολογιστές έχουν την εντύπωση ότι ανήκουν στο ίδιο δίκτυο

1.13 `ping 10.0.1.2`

1.14 Η γέφυρα συνεχίζει να προωθεί τα πακέτα στο PC2

1.15 Το `ping` δεν είναι επιτυχές καθώς δεν λαμβάνει πακέτα reply ο PC1 από τον PC2 επειδή δεν γνωρίζει πλέον την διεύθυνση του και τα στέλνει στην προηγούμενη διεύθυνση του PC2 που είναι καταχωρημένη στον πίνακα arp του.

1.16 Όχι

1.17 `ifconfig bridge0 addm em0 addm em1 addm em2 up`

1.18 Ναι

1.19 Δεν εμφανίζονται καθώς τα πακέτα πηγαίνουν από το LAN1 κατευθείαν στο LAN3 χωρίς να περνάνε από το LAN2.

1.20 Λαμβάνουμε ένα πακέτο arp request που αναζητάει τη διεύθυνση του PC3 και στέλνεται broadcast

1.21 ifconfig bridge0

1.22 ifconfig bridge0 addr

1.23 Στα PC1 PC2 PC3

1.24 ifconfig bridge0 flush

1.25 ifconfig bridge0 delete

1.26 ifconfig bridge0 destroy

1.27 ifconfig em0 delete

## **ΑΣΚΗΣΗ 2**

2.1 ifconfig em0 192.168.1.{1,2,3,4}/24

2.2 ifconfig bridge1 create

ifconfig bridge1 addm em0 addm em1 up

2.3 ifconfig bridge2 create

ifconfig bridge2 addm em0 addm em1 up

2.4 ifconfig bridge3 create

ifconfig bridge3 addm em0 addm em1 up

2.5

PC1: 08:00:27:cd:d4:0e

PC2: 08:00:27:36:e1:e6

PC3: 08:00:27:80:fb:2d

PC4: 08:00:27:dc:63:3f

arp -d -a

2.6 ifconfig bridge{1,2,3} flush

2.7 tcpdump -vi em0

2.8 ifconfig bridge{1,2,3} addr

B1:

08:00:27:36:e1:e6 Vlan1 em1 1162 flags=0<>

08:00:27:cd:d4:0e Vlan1 em0 1162 flags=0<>

B2:

08:00:27:36:e1:e6 Vlan1 em0 1177 flags=0<>

08:00:27:cd:d4:0e Vlan1 em0 1177 flags=0<>

B3:

08:00:27:cd:d4:0e Vlan1 em0 1192 flags=0<>

2.9 broadcast arp request από το pc1 όπότε όλα τα bridges μαθαίνουν τη διεύθυνσή του arp reply από το pc2 στοχευμένη προς τη διεύθυνση του pc1 το οποίο περνάει από το LNK1 και καταχωρείται από το B1 και το B2.

2.10 Δεν υπήρχε αλλαγή στους πίνακες των γεφυρών καθώς το PC2 έχει καταχωρημένη τη διεύθυνση του PC1 στον πίνακα arp του και έστειλε απευθείας το μήνυμα icmp σε εκείνον ακολουθώντας την διαδρομή που είχε ακολουθήσει και προηγουμένως. (Η μοναδική αλλαγή παρατηρείται στις θύρες).

2.11 Το PC4 απαντάει με icmp reply στο PC2 το οποίο βρίσκεται στο LNK1 που αντιστοιχεί στη δεξιά διεπαφή του B1. Συνεπώς ενημερώνεται και ο πίνακας της γέφυρας του B1.

2.12 Πλέον οι γέφυρες περιλαμβάνουν στον πίνακά τους και τη διεύθυνση του PC3 λόγω του broadcast arp request που έστειλε αρχικά για να βρει τη διεύθυνση του PC2.

2.13 ping

2.14 Το ping από το PC4 συνεχίζει να μεταφέρει τα πακέτα με επιτυχία καθώς βρίσκονται πλέον στο ίδιο εσωτερικό δίκτυο

2.15 Τα πακέτα από το ping του PC1 δεν μεταφέρονται πλέον με επιτυχία καθώς εκεί που στέλνονται δεν βρίσκεται πλέον το PC2.

2.16 Τα bridges ενημερώθηκαν με τη νέα τοποθεσία του PC2 μετά το ping προς το PC3, γι' αυτό και τώρα το ping του PC1 προς το PC2 είναι επιτυχές.

2.17 Θα έπρεπε να περιμένουμε αρκετό χρόνο μέχρι να διαγραφεί η καταχώριση του PC2 από τον πίνακα arp του PC1.

### ΑΣΚΗΣΗ 3

3.1 ifconfig bridge0 create

ifconfig bridge0 addm em0 addm em1 up

ifconfig em{0,1} up

3.2 ifconfig bridge1 create

ifconfig bridge1 addm em0 addm em1 up

ifconfig em{0,1} up

3.3 PC1: 08:00:27:cd:d4:0e

PC2: 08:00:27:36:e1:e6

PC3: 08:00:27:80:fb:2d

arp -da

3.4 Εμφανίζεται το arp request για τη διεύθυνση του PC2.

3.5 ping 192.168.1.1

3.6 ifconfig bridge0 addm em2

ifconfig em2 up

ifconfig bridge1 addm em2

ifconfig em2 up

3.7 ifconfig bridge0 addr

ifconfig bridge1 addr

3.8 bridge0 -> PC1:em0, PC3:em1

3.9 bridge1 -> PC1:em0, PC3:em1

3.10 Αποτυγχάνει

3.11 Το PC1 εμφανίζεται στη διεπαφή em0 και το PC3 εμφανίζεται στην διεπαφή em1, προς την ίδια κατεύθυνση με το PC1, λόγω του κατακλυσμού των πακέτων που ανακάτεψαν τον πίνακα δρομολόγησης της γέφυρας.

3.12 Γίνεται ερώτηση για τη διεύθυνση του PC1 και δίνεται η απάντηση από το pc1 για τη MAC διεύθυνση του.

3.13 Στο PC2 φτάνουν μόνο τα requests που έχουν δημιουργηθεί από τον καταιγισμό λόγω του διπλού link και δεν φτάνουν οι απαντήσεις του PC1.

3.14 Καθώς στέλνονται με broadcast προς όλες τις κατευθύνσεις.

3.15 Λόγω της δρομολόγησης στη γέφυρα τα reply κάνουν κύκλους μεταξύ LNK1 και LNK2 και δε φτάνουν ποτέ στο LAN2.

#### **ΑΣΚΗΣΗ 4**

4.1 ifconfig bridge0 destroy

ifconfig emX down

4.2 ifconfig emX up

ifconfig lagg0 create

4.3 ifconfig lagg0 up laggport em1 laggport em2

4.4 ifconfig lagg0 up laggport em0 laggprot em2

4.5 ifconfig bridge0 addm em0 addm lagg0 up

4.6 ifconfig bridge1 addm lagg0 addm em1 up

4.7 Μόνο το arp request εμφανίζεται

4.8 tcpdump -v

4.9 Επιτυχές

4.10 Λόγω του πρωτοκόλλου failover η κίνηση στέλνεται στο πρώτο interface μέχρι να κάνει fail όπου θα χρησιμοποιηθεί το επόμενο. Γι' αυτό και η κίνηση εμφανίζεται στο LNK1 που αντιστοιχεί στη διεπαφή em1 για το bridge1.

4.11 Τα πακέτα πλέον μεταφέρονται μέσω του LNK2.

4.12 Τα πακέτα ξαναμεταφέρονται μέσω του LNK1.

## **ΑΣΚΗΣΗ 5**

5.1 ifconfig bridgeX destroy

ifconfig lagg0 destroy

ifconfig emX down

5.2 ifconfig bridge1 create

ifconfig bridge1 addm em0 addm em1 addm em2 up

ifconfig emX up

5.3 ifconfig bridge2 create

ifconfig bridge2 addm em0 addm em1 addm em2 up

ifconfig emX up

5.4 ifconfig bridge1 stp em0 stp em1 stp em2

5.5 ifconfig bridge2 stp em0 stp em1 stp em2

5.6 B1->bridge id= 08:00:27:56:04:09

B2->bridge id= 08:00:27:21:8a:9d

5.7 B1->root id= 08:00:27:56:04:09

B2->root id= 08:00:27:21:8a:9d

5.8 B2 είναι η γέφυρα ρίζα.

em0 role: designated, state: forwarding

em1 role: designated, state: forwarding

em2 role: designated, state: forwarding

5.9 η διεπαφή em1 (LNK1)

5.10 em2 (LNK2) role: alternate, state: discarding

5.11 em0 (LAN1) role: designated, state: forwarding

5.12 tcpdump -vei em0

Κάθε 2 seconds εκπέμπονται BPDUs

5.13 IEEE 802.3

5.14 MAC src: 08:00:27:3f:28:c5

MAC dst: 01:80:c2:00:00:00

5.15 Στη διεπαφή em0 (LNK1)

5.16 multicast (τελευταίο bit πρώτου byte είναι 1)

5.17 root ID: 8000.08:00:27:21:8a:9d

bridge id: 8000.08:00:27:21:8a:9d.8001

root path cost: 0

5.18 bridge ID: 8000.08:00:27:21:8a:9d.8003 (προτεραιότητα 8003 > 8001, άρα 8001 μεγαλύτερη προτεραιότητα)

5.19 Ναι

5.20 Στη θύρα με ρόλο designated

5.21 LNK1:

root id: 8000.08:00:27:21:8a:9d

bridge id: 8000.08:00:27:56:04:09.8001

root path cost: 0

LNK2:

root id: 8000.08:00:27:21:8a:9d

bridge id: 8000.08:00:27:21:8a:9d.8003

root path cost: 0

5.22 Επιτυγχάνει

5.23 χάνονται 8 πακέτα με 1.5 sec το καθένα άρα περίπου μετά από 12 secs αποκαθίσταται η επικοινωνία

Η αναμενόμενη καθυστέρηση για νέο port είναι περίπου 15 seconds άρα είναι δικαιολογημένη η καθυστέρηση.

5.24 Η επικοινωνία συνεχίζεται χωρίς διακοπή

## **ΑΣΚΗΣΗ 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 up

ifconfig emX up

ifconfig bridge3 stp em0 stp em1 stp em2

6.4 ifconfig bridgeX flush

To ping επιτυγχάνει

6.5 ifconfig bridge1 priority 0

6.6 LNK1: 0

LNK2: 0

LNK4: 20000

ταχύτητα κάρτας δικτύου: 1000000bits/sec (1Gbps)

Προκύπτει από τη διαίρεση 20Tbps/1Gbps = 20000

6.7 από B1: 0

από B2: 20000

6.8 η θύρα 1 στο LNK3 όπου path cost=0

6.9 role: alternate, state: discarding

6.10 20000

6.11 ping 192.168.1.3

6.12 ifconfig bridge3 ifpathcost em0 41000

6.13 Σε 4 δευτερόλεπτα περίπου

6.14 role alternate, state discarding

6.15 Όχι

6.16 Ναι, πλέον το root path cost που παράγει η bridge3 στο LAN3 είναι 40000

6.17 Περίπου 7 δευτερόλεπτα

6.18 Περίπου 6 δευτερόλεπτα, αντίστοιχος χρόνος με πριν

## ΑΣΚΗΣΗ 7

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

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

7.4 Ναι

7.5 Όχι, καθώς οι διεπαφές τους βρίσκονται σε διαφορετικά εικονικά υποδίκτυα VLAN.

7.6 Όχι, καθώς οι διεπαφές τους βρίσκονται σε διαφορετικά εικονικά υποδίκτυα VLAN.

7.7 `ifconfig bridge1 -stp em0`

7.8 `tcpdump -xe`

7.9 `arp -da`

ARP: 0x0806

IPv4: 0x0800

7.10 Έχουν ethertype 802.1Q

7.11 0x8100

Ξεχωρίζουν επειδή αμέσως μετά το vlan tag υπάρχει και πεδίο με τον αρχικό τύπο Ethernet (IPv4 ή ARP)

7.12 Η πληροφορία για το VLAN βρίσκεται στην αρχή του πεδίου Type αμέσως μετά τις MAC διευθύνσεις πηγής και προορισμού και πριν το αρχικό type των πακέτων.

7.13 `tcpdump -vxei em0.5`

7.14 ARP: 0x0806

IPv4: 0x0800

Δεν υπάρχει πεδίο σχετικό με VLAN

7.15 `ifconfig bridge1 stp em0`

`tcpdump -vxei em0`

7.16 Όχι, τα BPDU είναι τύπου 802.3 αντί για 802.1Q που ήταν ο τύπος των προηγούμενων πακέτων.

7.17 `tcpdump -vexi em0 'not stp'`