

## Εργαστηριακή Άσκηση 3

### Τοπικά Δίκτυα και Μεταγωγείς LAN

Όνοματεπώνυμο: Άννα Κουτσώνη	Όνομα PC: DESKTOP-90FT571
Ομάδα: 1	Ημερομηνία: 27/02/2024

#### Άσκηση 1

- 1.1. Τις ορίζουμε στατικά: `ifconfig em0 192.168.1.1/24` και `ifconfig em0 192.168.1.2/24`
- 1.2. `ifconfig em0 up`  
`ifconfig em1 up`
- 1.3. Λαμβάνουμε μήνυμα `host is down`. Τα δύο μηχανήματα δεν επικοινωνούν γιατί βρίσκονται σε διαφορετικά εσωτερικά δίκτυα LAN1, LAN2.
- 1.4. Εφόσον το `ping` από τον PC1 στον PC2 και αντίστροφα δεν είναι επιτυχές και δεν λαμβάνεται απάντηση δεν παράγονται ICMP πακέτα, αλλά μόνο ARP Requests τα οποία καταγράφονται από την `tcpdump`.
- 1.5. `ifconfig bridge0 create`  
`ifconfig bridge0 addm em0 addm em1 up`
- 1.6. Ναι υπάρχει επικοινωνία, το `ping` είναι επιτυχές.
- 1.7. TTL=64 που είναι η αρχική default τιμή άρα PC1 και PC2 συνδέονται άμεσα και η προσθήκη της γέφυρας δεν παρεμβάλλει κάποιο βήμα ενδιάμεσα.
- 1.8. `Arp -a`  
Παρατηρούμε ότι υπάρχουν καταχωρημένες μόνο οι διευθύνσεις των PC1, PC2 και όχι της γέφυρας, θεωρείται διαφανής.
- 1.9. `tcpdump -i em0 -vvv`  
`tcpdump -i em1 -vvv`  
και επιβεβαιώνουμε ότι γίνεται η προώθηση
- 1.10. Όχι δεν κάνει κάποια αλλαγή
- 1.11. Όχι δεν αλλάζει κάποιο πεδίο
- 1.12. Όχι δεν υπάρχει καμία ένδειξη ύπαρξης του B1, η γέφυρα θεωρείται διαφανής.
- 1.13. `tcpdump -i -em1`  
`ping 192.168.1.2`
- 1.14. Η γέφυρα συνεχίζει να προωθεί πακέτα προς την 192.168.1.2 αλλά αυτή η διεύθυνση δεν ανήκει πλέον στον PC2 άρα τα πακέτα δεν προωθούνται σε αυτόν.
- 1.15. Όχι δεν είναι επιτυχές, γιατί δεν υπάρχει μηχανήμα με την 192.168.1.2 για να απαντήσει.
- 1.16. Όχι δεν μπορούμε. Λαμβάνουμε μήνυμα `no route to host` και αυτό συμβαίνει διότι η διεπαφή `em2` δεν έχει προστεθεί στη γέφυρα `bridge0`.
- 1.17. `ifconfig em2 up`  
`ifconfig bridge0 addm em2 up`
- 1.18. Ναι λαβάνουμε

- 1.19. Όχι δεν εμφανίζονται. Αυτό συμβαίνει διότι λόγω του προηγούμενου ping η γέφυρα γνωρίζει σε ποια θύρα πρέπει να προωθήσει τα πακέτα και έτσι αυτά πηγαίνουν κατευθείαν από το LAN1 στο LAN3 χωρίς να περνάνε από το LAN2.
- 1.20. `arp -d -a`  
Καταγράφεται ένα ARP Request το οποίο προέρχεται από το PC1 και ρωτάει ποιος έχει την 192.168.1.3
- 1.21. `ifconfig bridge0`
- 1.22. `ifconfig bridge0 addr`
- 1.23. Στα μηχανήματα PC1,PC3
- 1.24. `ifconfig bridge0 flush`
- 1.25. `ifconfig bridge0 delete em2`
- 1.26. `ifconfig bridge0 destroy`
- 1.27. `ifconfig em0 delete` στα PC1,PC2,PC3

## Άσκηση 2

- 2.1. PC1: `ifconfig em0 192.168.1.1/24`  
PC2: `ifconfig em0 192.168.1.2/24`  
PC3: `ifconfig em0 192.168.1.3/24`  
PC4: `ifconfig em0 192.168.1.4/24`
- 2.2. `ifconfig bridge1 create`  
`ifconfig em0 up`  
`ifconfig em1 up`  
`ifconfig bridge1 addm em0 addm em1 up`
- 2.3. `ifconfig bridge2 create`  
`ifconfig em0 up`  
`ifconfig em1 up`  
`ifconfig bridge2 addm em0 addm em1 up`
- 2.4. `ifconfig bridge3 create`  
`ifconfig em0 up`  
`ifconfig em1 up`  
`ifconfig bridge3 addm em0 addm em1 up`
- 2.5. PC1: 08:00:27:a5:dc:a2  
PC2: 08:00:27:2d:53:b2  
PC3: 08:00:27:f0:46:62  
PC4: 08:00:27:74:09:25  
Αδειάζουμε τους πίνακες arp με `arp -d -a`
- 2.6. `ifconfig bridge1 flush`  
`ifconfig bridge2 flush`  
`ifconfig bridge3 flush`
- 2.7. `tcpdump`
- 2.8. `ifconfig bridge1 addr`  
B1: 08:00:27:2d:53:b2 Vlan1 em1 1170 flags=0<>  
08:00:27:a5:dc:a2 Vlan1 em0 1170 flags=0<>  
`ifconfig bridge2 addr`  
B2: 08:00:27:2d:53:b2 Vlan1 em0 1162 flags=0<>  
08:00:27:a5:dc:a2 Vlan1 em0 1162 flags=0<>

```
ifconfig bridge3 addr
```

```
B3: 08:00:27:a5:dc:a2 Vlan1 em0 1153 flags=0<>
```

2.9. Καταγράφηκαν τα εξής:

PC1: ARP Request

ARP Reply

ICMP Echo Request

ICMP Echo Reply

PC2: ARP Request

ARP Reply

ICMP Echo Request

ICMP Echo Reply

PC3: ARP Request

PC4: ARP Request

ARP Request: Ο PC1 ρωτάει να μάθει ποιος έχει την 192.168.1.2 και οι bridge1, bridge2, bridge3 μαθαίνουν ότι στην διεπαφή em0 βρίσκεται ο PC1

ARP Reply: Ο PC2 απαντά και οι bridge1 μαθαίνει ότι στην διεπαφή em1 βρίσκεται ο PC2, ενώ η bridge2 μαθαίνει ότι στην διεπαφή em0 βρίσκεται ο PC2  
ICMP Request

ICMP Request-ICMP Reply: Τα μηνύματα αυτά ανταλλάσσονται μεταξύ PC1,PC2 και διέρχονται μόνο από την bridge1, η οποία μετά τα ARP Request, ARP Reply ξέρει που να προωθήσει τα μηνύματα και έτσι η bridge3 δεν μαθαίνει κάτι παραπάνω.

2.10. Στους πίνακες προώθησης των B1,B2 δεν υπήρχαν αλλαγές. Ο πίνακας προώθησης του B3 έχει αλλάξει και μαθαίνει την MAC του PC2 διότι στάλθηκε νέο ARP Request (παρόλο που ο PC2 γνωρίζει ήδη την MAC του PC1 και δεν θα έπρεπε να σταλθεί νέο ARP Request και δεν αναμέναμε να γίνει)

2.11. ifconfig bridge1 addr

```
B1: 08:00:27:74:09:25 Vlan1 em1 1002 flags=0<>
```

```
08:00:27:2d:53:b2 Vlan1 em1 1002 flags=0<>
```

```
ifconfig bridge2 addr
```

```
B2: 08:00:27:74:09:25 Vlan1 em1 991 flags=0<>
```

```
08:00:27:2d:53:b2 Vlan1 em0 991 flags=0<>
```

```
ifconfig bridge3 addr
```

```
B3: 08:00:27:74:09:25 Vlan1 em1 982 flags=0<>
```

```
08:00:27:2d:53:b2 Vlan1 em0 982 flags=0<>
```

Όλοι οι πίνακες προώθησης περιέχουν την MAC του PC4, οι B2,B3 διότι τα ICMP request,reply ανταλλάσσονται μεταξύ του PC2 και του PC4 και ο B1 την περιέχει διότι το PC2 στέλνει ARP request στο οποίο απαντά το PC4 με ARP reply το οποίο περνάει από το LNK1.

2.12. ifconfig bridge1 addr

```
B1: 08:00:27:2d:53:b2 Vlan1 em1 1178 flags=0<>
```

```
08:00:27:f0:46:62 Vlan1 em1 1178 flags=0<>
```

```
ifconfig bridge2 addr
```

```
B2: 08:00:27:2d:53:b2 Vlan1 em0 1169 flags=0<>
```

```
08:00:27:f0:46:62 Vlan1 em1 1169 flags=0<>
```

```
ifconfig bridge3 addr
```

```
B3: 08:00:27:2d:53:b2 Vlan1 em0 1154 flags=0<>
```

```
08:00:27:f0:46:62 Vlan1 em0 1154 flags=0<>
```

Προστίθεται η MAC του PC3 στις B1,B2 λόγω του ARP Request και λόγω της προώθησης μέσω του B2 στο LNK1 μαθαίνει και καταγράφει την MAC του PC3 και η B1.

- 2.13. PC4: ping 192.168.1.2  
PC1: ping 192.168.1.2
- 2.14. Το ping συνεχίζει κανονικά να είναι επιτυχές αφού PC2,PC4 ανήκουν πλέον στο ίδιο εσωτερικό δίκτυο.
- 2.15. Το ping σταματάει, δεν είναι πλέον επιτυχές. Αυτό συμβαίνει διότι οι γέφυρες δεν είναι ενημερωμένες και δεν προωθούν σωστά στο LAN2 όπου βρίσκεται το PC2 πλέον και έτσι το πακέτο δεν φτάνει πότε.
- 2.16. Συνεχίζει κανονικά γιατί με το ping από το PC2 ανανεώθηκε ο πίνακας προώθησης του B2.
- 2.17. Μέχρι να λήξει η εγγραφή για τον PC2 από τον πίνακα προώθησης του B2. Με ifconfig bridge1 στο B1 βλέπουμε ότι ο χρόνος αυτός 1200sec.

### Άσκηση 3

- 3.1. ifconfig bridge1 create  
ifconfig em0 up  
ifconfig em1 up  
ifconfig bridge1 addm em0 addm em1 up
- 3.2. ifconfig bridge2 create  
ifconfig em0 up  
ifconfig em1 up  
ifconfig bridge2 addm em0 addm em1 up
- 3.3. PC1: 08:00:27:a5:dc:a2  
PC2: 08:00:27:2d:53:b2  
PC3: 08:00:27:f0:46:62  
Αδειάζουμε τους πίνακες arp με arp -d -a
- 3.4. Εμφανίζεται μόνο το πρώτο ARP request διότι οι πίνακες προώθησης όλων των γεφυρών είναι κενοί και το πακέτο στέλνεται σε όλα τα LANs.
- 3.5. Ping 192.168.1.1
- 3.6. ifconfig em2 up  
ifconfig bridge1 addm em2  
ifconfig em2 up  
ifconfig bridge2 addm em2
- 3.7. ifconfig bridge1 addr
- 3.8. B1: 08:00:27:a5:dc:a2 Vlan1 em0 1182 flags=0<>  
08:00:27:f0:46:62 Vlan1 em1 1182 flags=0<>  
ifconfig bridge2 addr  
B2: 08:00:27:a5:dc:a2 Vlan1 em0 1174 flags=0<>  
08:00:27:f0:46:62 Vlan1 em1 1174 flags=0<>  
Άρα B1: PC1→em0(LAN1)  
PC3→em1(LNK1)  
B2: PC1→em0(LNK1)  
PC3→em1(LAN2)

- 3.9. tcpdump -e
- 3.10.       arp -d -a  
          ping -c 1 192.168.1.1  
          To ping αποτυγχάνει
- 3.11.       Αποσυνδέουμε το καλώδιο από τις κάρτες δικτύου των B1,B2 στην ζεύξη LNK2. Με ifconfig bridge2 addr βλέπουμε ότι οι MAC των PC1,PC3 εμφανίζονται και οι δύο στην διεπαφή em0.
- 3.12.       ARP request: Who has 192.168.1.1 tell 192.168.1.3  
          ARP reply: 192.168.1.1 is at 08:00:27:a5:dc:a2
- 3.13.       Είναι η 08:00:27:f0:46:62 , δηλαδή η MAC του PC3
- 3.14.       Διότι το μήνυμα είναι broadcast και γίνεται flooding μεταξύ των γεφυρών.
- 3.15.       Γιατί η γέφυρα B2 έχει καταγράψει στον πίνακα προώθησης της ότι το PC3 βρίσκεται στην διεπαφή em2 και τα πακέτα δεν φτάνουν ποτέ στο LAN2, κάνουν κύκλους στον βρόχο που έχει δημιουργηθεί μεταξύ LNK1,LNK2.

#### Άσκηση 4

- 4.1 ifconfig bridge1 destroy  
    ifconfig em0 down  
    ifconfig em1 down  
    ifconfig em2 down  
    ifconfig bridge1 create
- ifconfig bridge2 destroy  
    ifconfig em0 down  
    ifconfig em1 down  
    ifconfig em2 down  
    ifconfig bridge2 create
- 4.2 ifconfig em0 up  
    ifconfig em1 up  
    ifconfig em2 up  
    ifconfig lagg0 create
- 4.3 ifconfig lagg0 up laggport em1 laggport em2
- 4.4 ifconfig em0 up  
    ifconfig em1 up  
    ifconfig em2 up  
    ifconfig lagg0 create  
    ifconfig lagg0 up laggport em0 laggport em2
- 4.5 ifconfig bridge1 addm em0 addm lagg0 up
- 4.6 ifconfig bridge2 addm em1 addm lagg0 up
- 4.7 Εμφανίζεται ένα ARP request με το οποίο ο PC2 προσπαθεί να μάθει την MAC του PC3. Το πακέτο φτάνει στο LAN1 μέσω της B1 η οποία το έχει λάβει μετά από broadcast της B2 που έλαβε το πακέτο από τον PC2 μέσω του LAN2.
- 4.8 tcpdump
- 4.9 arp -a -d  
    ping -c 1 192.168.1.1

Το ping είναι επιτυχές και παρατηρούμε στην καταγραφή ARP request, ARP reply, ICMP echo request, ICMP echo reply.

- 4.10 Εμφανίζεται κίνηση, πακέτα ICMP μόνο στη ζεύξη LNK1. Αυτό συμβαίνει λόγω του default πρωτοκόλλου failover το οποίο ορίζει τη μια ζεύξη ως master και την άλλη ως backup.
- 4.11 Το ping συνεχίζει να επιτυγχάνει αλλά η κίνηση τώρα παρατηρείται στο LNK2 αφού η master ζεύξη LNK1 είναι πλέον unavailable χρησιμοποιείται η backup LNK2.
- 4.12 Η κίνηση μεταφέρεται και πάλι στην master ζεύξη που είναι και πάλι διαθέσιμη LNK1.

## Άσκηση 5

- 5.1 `ifconfig bridge1 destroy`  
`ifconfig lagg0 destroy`  
`ifconfig em0 down`  
`ifconfig em1 down`  
`ifconfig em2 down`
- `ifconfig bridge2 destroy`  
`ifconfig lagg0 destroy`  
`ifconfig em0 down`  
`ifconfig em1 down`  
`ifconfig em2 down`
- 5.2 `ifconfig bridge1 create`  
`ifconfig em0 up`  
`ifconfig em1 up`  
`ifconfig em2 up`  
`ifconfig bridge1 addm em0 addm em1 addm em2 up`
- 5.3 `ifconfig bridge2 create`  
`ifconfig em0 up`  
`ifconfig em1 up`  
`ifconfig em2 up`  
`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 `ifconfig bridge1`  
`B1: 08:00:27:82:96:27`  
`ifconfig bridge2`  
`B2: 08:00:27:8a:be:01`
- 5.7 `root id: 08:00:27:82:96:27`  
Ίδια με την id της B1 άρα η bridge1 είναι η γέφυρα ρίζα του επικαλύπτοντος δέντρου
- 5.8 `em0: role designated state forwarding`  
`em1: role designated state forwarding`  
`em2: role designated state forwarding`
- 5.9 Η em1 δηλαδή η LNK1 που έχει role root

- 5.10 em2: role alternate state discarding
- 5.11 Γέφυρα B2, διεπαφή στο LAN2 em0: role designated state forwarding
- 5.12 Στην B1 tcpdump -i em0 -e -vvv . Κάθε 2 sec
- 5.13 IEEE 802.3
- 5.14 MAC πηγής: 08:00:27:f0:d0:74  
MAC προορισμού: 01:80:c2:00:00:00
- 5.15 Στην em0 που αντιστοιχεί στο LAN1 και γίνεται η καταγραφή
- 5.16 multicast διότι το LSB του πρώτου byte είναι 1
- 5.17 root id: 8000.08:00:27:82:96:27  
bridge id: 8000.08:00:27:82:96:27.8001  
root path cost: 0
- 5.18 tcpdump -i em1 -e -vvv  
bridge id: 8000.08:00:27:82:96:27.8002  
Τα τελευταία τέσσερα ψηφία δηλαδή το πρώτο μέρος της bridge id είναι η προτεραιότητα.
- 5.19 Το δεύτερο μέρος είναι η bridge id της γέφυρας ρίζας
- 5.20 Προκύπτει ως άθροισμα της προτεραιότητας και της θύρας της διεπαφής em1
- 5.21 Όχι δεν παρατηρούμε
- 5.22 Στην em1
- 5.23 root id: 8000.08:00:27:82:96:27  
bridge id: 8000.08:00:27:82:96:27.8002  
root path cost: 0
- 5.24 Ναι είναι επιτυχές
- 5.25 Χρειάζονται περίπου 8 sec , τιμή κοντά στην αναμενόμενη που είναι  $3 \times \text{hello time} = 3 \times 2 = 6 \text{ sec}$ . Τα δύο έξτρα δευτερόλεπτα οφείλονται ίσως σε καθυστέρηση λήψης και απόκρισης του ping
- 5.26 Όχι δεν υπάρχει

## Άσκηση 6

- 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 em0 up  
ifconfig em1 up  
ifconfig em2 up  
ifconfig bridge3 addm em0 addm em1 addm em2 up  
ifconfig bridge3 stp em0 stp em1 stp em2
- 6.4 ifconfig bridge1 flush  
ifconfig bridge2 flush  
ifconfig bridge3 flush  
Το ping είναι επιτυχές

- 6.5 `ifconfig bridge1 priority 0`
- 6.6 Και για τα τρία είναι  $\text{path cost} = 20000$  . Προκύπτει από τον τύπο  $\frac{20000 \text{Tbps}}{\text{bandwidth} = 20 \text{Tbps} / 1 \text{ Gbps}} = 20000$
- 6.7 Για την B1 κάνουμε `tcpdump -i em1 -e -vnn` και βρίσκουμε `root path cost` ίσο με 0  
 διότι η `bridge1` είναι `root`  
 Για την B2 κάνουμε `tcpdump -i em2 -e -vnn` και βρίσκουμε `root path cost` ίσο με 20000 και προκύπτει όπως αναφέρθηκε πριν
- 6.8 Είναι η LNK3 με το χαμηλότερο `root path cost`
- 6.9 Στο B3 η LNK4 έχει `role alternate state discarding`  
 Στο B2 η LNK4 αντίστοιχα έχει `role designated state forwarding`
- 6.10 `root path cost=20000`
- 6.11 `ping 192.168.1.3`
- 6.12 `ifconfig bridge3 ifpathcost em1 50000`  
 Επιλέγουμε την τιμή 50000 διότι το συνολικό κόστος για την διεπαφή LNK4 είναι 40000 και άρα θέλουμε κάτι μεγαλύτερο για την LNK3 ώστε η LNK4 να γίνει ριζική
- 6.13 6 sec
- 6.14 LNK3: `role alternate state discarding`  
 LNK4: `role designated state forwarding`
- 6.15 Όχι δεν υπήρξε κάποια αλλαγή
- 6.16 Το `root path cost` είναι 40000
- 6.17 6 sec
- 6.18 Δεν υπήρξε διακοπή
- 6.19 `Ifconfig em3 up`  
`Ifconfig bridge3 addm em3`  
`Ifconfig bridge3 stp em3`  
 1<sup>η</sup> θύρα: `role designated state forwarding`  
 2<sup>η</sup> θύρα: `role backup state discarding`
- 6.20 `ifconfig bridge3 ifpathcost em1 20000`  
 Το επαναφέρουμε στο 20000

## Άσκηση 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 em0.5 create up`  
`ifconfig em0.6 create up`
- 7.3 `ifconfig em1.6 create up`  
`ifconfig em3.5 create up`
- 7.4 `ifconfig em0.6 create vlan 6 vlandev em0 inet 192.168.6.2/24`
- 7.5 `ifconfig em0.6 create up`  
`ifconfig em1.6 create up`
- 7.6 `ifconfig em0.5 create vlan 5 vlandev em0 inet 192.168.5.3/24`
- 7.7 `ifconfig em0.5 create up`  
`ifconfig em1.5 create up`
- 7.8 Ναι μπωρώ
- 7.9 `ifconfig bridge1 -stp em0`



- 7.10      `tcpdump -i em0 -e -vvv -x`
- 7.11      `arp -a -d`  
          `ping -c 1 192.168.1.1`  
          ethertype ARP (0x0806)  
          ethertype IPv4 (0x0800)
- 7.12      Είναι μεγαλύτερα κατά 4 bytes διότι προστίθεται το Vlan tag που είναι 802.1Q
- 7.13      Η τιμή του ethertype είναι 0x8100 και στη συνέχεια υπάρχει σε κάθε πακέτο η επικεφαλίδα ethertype IPv4 ή ethertype ARP
- 7.14      Μετά το πεδίο length
- 7.15      `tcpdump -i em0.5 -e -vvv -x`
- 7.16      ethertype ARP (0x0806)  
          ethertype IPv4 (0x0800)  
          Όχι δεν υπάρχει πεδίο για τον Vlan
- 7.17      `ifconfig bridge1 stp em0`  
          `tcpdump -i em0 -e -vvv -x`
- 7.18      Όχι δεν είναι ίδιου τύπου, στη θέση του ethertype υπάρχει το 802.3
- 7.19      `tcpdump not stp`