

LAB-03 (LANs & Switches)

Όνοματεπώνυμο : Νίκος Μπέλλος (el18183)	Όνομα PC : BELLOS-DELL-G3
Ομάδα : 3	Ημερομηνία : 30/03/2022

Άσκηση 1: Γέφυρα - Διασύνδεση δύο LAN

1. `ifconfig em0 inet 192.168.1.1` και `ifconfig em0 inet 192.168.1.2` αντίστοιχα
2. `ifconfig em0 up` και `ifconfig em1 up`
3. Παρατηρώ ότι δεν επικοινωνούν γιατί βρίσκονται σε διαφορετικά LANs
4. Δεν παράγονται πακέτα γιατί χωρίς τη γέφυρα δεν υπάρχει επικοινωνία μεταξύ τους και άρα το καθένα είναι απομονωμένο
5. `ifconfig bridge0 create`, `ifconfig bridge0 addm em0 addm em1 up` (τις εκτελούμε στο B1)
6. Ναι, τώρα επικοινωνούν !
7. Φαίνεται να απέχουν 1 βήμα, όπως και πριν, χωρίς την ύπαρξη της γέφυρας (η γέφυρα δεν περιλαμβάνεται στα βήματα του TTL)
8. Ο πίνακας ARP του αντίστοιχου VM περιέχει τη δική του εγγραφή και την διεύθυνση MAC του άλλου VM (λείπει η διεύθυνση MAC της γέφυρας)
9. Επιβεβαιώνουμε ότι το B1 προωθεί τα πακέτα. Χρησιμοποιήσαμε την `tcpdump -i em0/em1 -vvv`
10. Όχι, δεν αλλάζει τις διευθύνσεις τους
11. Όχι, δεν φαίνεται να αλλάζει κάτι
12. Όχι, δεν υπάρχει γιατί με `ttl=1` ο PC2 απαντάει (στο πρώτο βήμα του traceroute)
13. `ping 192.168.1.2`, `tcpdump -i em1 -vvv`
14. `ifconfig em0 inet 192.168.2.1` → στο PC2.
Ναι, η γέφυρα τα προωθεί στη θύρα που ήταν πριν το PC2 καθώς υπάρχει καταγραφή στο πίνακα προώθησης.
15. Όχι, το `ping` δεν είναι επιτυχές, γιατί πλέον δεν υπάρχει μηχανήμα με IP 192.168.1.2 το οποίο μπορεί να απαντήσει
16. Όχι, δεν μπορώ, γιατί η διεπαφή `em2` δεν έχει προστεθεί στο `bridge0`
17. `ifconfig em2 up`, `ifconfig bridge0 addm em2`
18. Ναι, λαμβάνω απάντηση
19. Όχι, δεν μεταδίδονται πακέτα στο LAN2 γιατί λόγω του προηγούμενου `ping` η γέφυρα έχει συμπληρώσει τον πίνακα προώθησης για τα PC1, PC3 και επομένως προωθεί τα πακέτα στη σωστή θύρα
20. Καταγράφεται ένα `arp request` το οποίο προέρχεται από το PC1 και ζητά ποιος έχει τη διεύθυνση του PC3 (192.168.1.3)
21. `ifconfig bridge0`
22. `ifconfig bridge0 addr`
23. Στα μηχανήματα PC1, PC3
24. `ifconfig bridge0 flush`
25. `ifconfig bridge0 deletem em2`
26. `ifconfig bridge0 destroy`
27. `ifconfig em0 192.168.1.x remove` / `ifconfig em0 192.168.1.x -alias`

Άσκηση 2: Αυτο-εκπαίδευση γεφυρών

1. `ifconfig em0 inet 192.168.1.x`, $x = \{1,2,3,4\}$

2. `ifconfig bridge1 create`, `ifconfig em0 up`, `ifconfig em1 up`, `ifconfig bridge1 addm em0 addm em1 up`
3. `ifconfig bridge2 create`, `ifconfig em0 up`, `ifconfig em1 up`, `ifconfig bridge2 addm em0 addm em1 up`
4. `ifconfig bridge3 create`, `ifconfig em0 up`, `ifconfig em1 up`, `ifconfig bridge3 addm em0 addm em1 up`
5. **PC1** : 08:00:27:73:0C:EE
PC2 : 08:00:27:FD:48:D7
PC3 : 08:00:27:BE:32:1B
PC4 : 08:00:27:C5:C5:E9
 Άδειασμα με `arp -d -a`
6. `ifconfig bridgeX flush` → X = {1, 2, 3}
7. `tcpdump` → για κάθε PC
8. `ifconfig bridgeX flush` → Καθαρισμός πίνακα προώθησης γέφυρας X
`ifconfig bridgeX addr` → Πίνακας προώθησης γέφυρας X
 Bridge1 → PC1, PC2
 Bridge2 → PC1, PC2
 Bridge3 → PC1
9. Το PC1 θα ρωτήσει μέσω ARP request ποιος έχει τη διεύθυνση 192.168.1.2. Ο μόνος που θα απαντήσει με ARP reply θα είναι ο PC2 (Από το ARP request όλες οι γέφυρες θα προσθέσουν στο πίνακα προώθησης το PC1 και κατά το ARP reply μόνο οι γέφυρες B1 και B2 θα προσθέσουν στο πίνακα προώθησης το PC2. Πλέον και οι δύο γέφυρες B1, B2 έχουν συμπληρώσει σωστά τους πίνακές τους για τα PC1 και PC2 άρα τα ICMP request και reply θα ακολουθήσουν τη σωστή οδό από τον PC1 στο PC2
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
11. Bridge1 → PC1, PC2, PC3
 Bridge2 → PC1, PC2, PC3
 Bridge3 → PC1, PC2, PC3
 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 δεν το προωθεί σε άλλη θύρα, το αποθηκεύει στο πίνακα προώθησης
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 em0 λαμβάνεται και από το B1. Έτσι, πλέον και οι 3 γέφυρες έχουν εγγραφές για όλα τα PC
13. `ping 192.168.1.2`
14. Το ping λειτουργεί κανονικά γιατί το PC2 πλέον θεωρείται συνδεδεμένο με το PC4
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
16. Το ping αρχίζει και πάλι να λειτουργεί, διότι με το ping από το PC2 ο πίνακας προώθησης του B2 ανανεώθηκε.
17. Θα έπρεπε να περιμένω μέχρι να λήξει η εγγραφή για το PC2 από το πίνακα προώθησης.

Άσκηση 3: Καταιγίδα πλαισίων εκπομπής

1. `ifconfig bridge0 create`, `ifconfig bridge0 addm em0 addm em1 up`
2. `ifconfig bridge1 create`, `ifconfig bridge1 addm em0 addm em1 up`
3. **PC1** : 08:00:27:73:0C:EE
PC2 : 08:00:27:FD:48:D7
PC3 : 08:00:27:BE:32:1B
 Άδειασμα με `arp -d -a`

4. Εμφανίζεται το πρώτο ARP request του PC2 καθώς επειδή τα B1, B2 έχουν άδειους πίνακες προώθησης κάνουν flood το πακέτο αυτό.
5. `ping 192.168.1.1`
6. `ifconfig em2 up`, `ifconfig bridgeX addm em2`
7. `ifconfig bridgeX addr` → Και οι δύο γέφυρες έχουν εγγραφές και για τα 3 PC
8. B1 → PC1: em0, PC3: em1
B2 → PC1: em0, PC3: em1
9. `tcpdump`
10. Ναι, είναι επιτυχές και δημιουργεί αέναη κίνηση στο LAN1
11. PC1: em2
PC3: em0
12. **ARP request:** who has 192.168.1.1
ARP reply: 192.168.1.1 is at <MAC-PC1>
13. Γιατί η γέφυρα B2 κάθε φορά που δέχεται το ARP request το προωθεί στις υπόλοιπες 2 θύρες και έτσι το πακέτο αυτό μεταδίδεται συνέχεια μεταξύ των γεφυρών και στα άκρα τους
14. Λόγω συνεχούς flooding των γεφυρών. Τα ARP πακέτα λειτουργούν με το να στέλνονται σε όλες τις πόρτες
15. Γιατί στο πίνακα προώθησης του B2 έχει καταγραφεί (λόγω του κύκλου) ότι το PC3 βρίσκεται στη θύρα em2 → LNK2 και έτσι δεν στέλνει ποτέ το ARP reply στη σωστή θύρα

Άσκηση 4: Συνάθροιση ζεύξεων

1. `ifconfig bridgeX destroy`, `ifconfig emX down`, `ifconfig bridgeX create`
2. `ifconfig emX up`, `ifconfig lagg0 create`
3. `ifconfig lagg0 up laggport em1 laggport em2`
4. Ίδιες εντολές με 2, 3
5. `ifconfig bridge0 addm em0 addm lagg0 up`
6. `ifconfig bridge1 addm em1 addm lagg0 up`
7. Ναι, εμφανίζεται κανονικά κίνηση (ARP request)
8. `tcpdump` → PC1
9. Ναι, είναι επιτυχές, στάλθηκαν κανονικά ARP request, reply και ICMP request, reply
10. Εμφανίζεται κίνηση μόνο στο LNK1, καθώς η ζεύξη LNK2 δεν είναι active. Αυτό οφείλεται στο πρωτόκολλο failover που είναι default και ορίζει μόνο τη μία από τις δύο ζεύξεις ως master και την άλλη ως backup
11. Παρατηρούμε ότι η κίνηση μεταφέρθηκε από το LNK1 στο LNK2
12. Η κίνηση επέστρεξε και πάλι στη ζεύξη LNK1

Άσκηση 5: Αποφυγή βρόχων

1. `ifconfig lagg0 destroy`, `ifconfig bridgeX destroy`
2. `ifconfig bridge1 create`, `ifconfig bridge1 addm em0 addm em1 addm em2 up`
3. `ifconfig bridge2 create`, `ifconfig bridge2 addm em0 addm em1 addm em2 up`
4. `ifconfig bridge1 stp em0 stp em1 stp em2`
5. `ifconfig bridge2 stp em0 stp em1 stp em2`
6. B1 → 08:00:27:1A:BB:D7
B2 → 08:00:27:0B:10:33

7. Και οι δύο γέφυρες έχουν ίδια προτεραιότητα (default), άρα γέφυρα ρίζα θα είναι η B2 που έχει τη μικρότερη MAC
8. Και οι τρεις θύρες της γέφυρας ρίζας έχουν το ρόλο των designated ports, άρα θα διαδίδεται πληροφορία σε όλες
9. Για το B1 ριζική θύρα είναι αυτή στο LNK1
10. Η άλλη θύρα (LNK2) έχει το ρόλο της alternate θύρας
11. Η θύρα αυτή είναι designated
12. `tcpdump -i em1 -e -vvv` → εκπέμπονται κάθε 2 δευτερόλεπτα (hello-time)
13. Χρησιμοποιείται ενθυλάκωση IEEE 802.3
14. Η διεύθυνση πηγής είναι η Bridge ID, ενώ η διεύθυνση προορισμού είναι η 01:80:C2:00:00:00
15. Ανήκει στη θύρα em0 που συνδέεται στο LNK1
16. Multicast (λόγω των μηδενικών στο τέλος)
17. Root ID : 08:00:27:0b:10:33
Bridge ID : 08:00:27:0b:10:33.8001
root path cost : 0
18. Bridge ID : 08:00:27:0b:10:33.8003 → το priority βρίσκεται στο τέλος της του ID (είναι το 8003)
Εκπέμπεται από την θύρα em2 η οποία είναι συνδεδεμένη στο LNK2
19. Όχι, δεν παρατηρούμε κίνηση στα LNK1, LNK2
20. Παρατηρούμε μόνο στη θύρα em0 που είναι designated (και αυτή είναι συνδεδεμένη με το LAN1)
21. Root ID : 08:00:27:0B:10:33 (LNK1)
Bridge ID : 08:00:27:1A:BB:D7
Root path cost : 20000
22. Ναι, είναι επιτυχές
23. Περνάνε περίπου 6 δευτερόλεπτα. Αυτό γίνεται διότι στο RSTP χρειάζονται περίπου 3x2 seconds για να ανταποκριθεί σε κάποια τοπολογική αλλαγή του δικτύου
24. Όχι, δεν υπάρχει διακοπή

Άσκηση 6: Ένα πιο πολύπλοκο δίκτυο με εναλλακτικές διαδρομές

1. `ifconfig em3 up`, `ifconfig bridge1 addm em3`, `ifconfig bridge1 stp em3`
2. `ifconfig em3 up`, `ifconfig bridge2 addm em3`, `ifconfig bridge2 stp em3`
3. `ifconfig bridge3 create`, `ifconfig bridge3 addm em0 addm em1 addm em2 up`, `ifconfig bridge3 stp em0 stp em1 stp em2`
4. `ifconfig bridgeX flush` → Ναι είναι επιτυχές
5. `ifconfig bridge1 priority 0`
6. Τα path costs είναι όλα 20000. Επομένως από το τύπο $\frac{20 \text{ Tbit/s}}{\text{bandwidth}}$ προκύπτει σωστά για bandwidth = 1Gbps
7. Από τη γέφυρα B1 λαμβάνει κόστος 0, ενώ από τη B2 κόστος 20000 (αυτό σημαίνει πως αν επιλέξει τη θύρα LNK3 θα πληρώσει κόστος 0+cost[LNK3] ενώ αν επιλέξει την LNK4 θα πληρώσει κόστος 20000+cost[LNK4])
8. Είναι ριζική η em0 → LNK3 γιατί ελαχιστοποιεί το κόστος για τη γέφυρα ρίζα
9. Η άλλη θύρα έχει ρόλο alternative θύρας
10. Root-path cost = 20000
11. `ping 192.168.1.3`
12. `ifconfig bridge3 ifpathcost em0 50000` → Η διεπαφή για το LNK4 κοστίζει συνολικά 40000, άρα θέλουμε κάτι μεγαλύτερο από αυτό.
13. Περνάνε περίπου 2 δευτερόλεπτα (hello-time)
14. Είναι πλέον alternate port

15. Όχι, φαίνεται οι παράμετροι να μην έχουν αλλάξει
16. Ναι, πλέον η bridge3 φαίνεται να έχει root-path cost ίσο με 40000
17. Περίπου 6-8 δευτερόλεπτα (3 hello-time)
18. Η επικοινωνία δεν διακόπτεται

Άσκηση 7: Εικονικά τοπικά δίκτυα (VLAN)

1. `ifconfig em0.5 create vlan 5 vlandev em0 inet 192.168.5.1/24` (αντίστοιχα για το vlan 6)
2. `ifconfig em0.6 create vlan 6 vlandev em0 inet 192.168.6.2/24`
3. `ifconfig em0.5 create vlan 5 vlandev em0 inet 192.168.5.3/24`
4. Ναι, μπορώ
5. Όχι, γιατί δεν βρίσκονται σε κοινό VLAN
6. Όχι, για τον ίδιο λόγο
7. `ifconfig bridge1 -stp em0`
8. `tcpdump -i em0 -e -vvv -x`
9. Ethertype ARP → 0806
Ethertype IPv4 → 0800
10. Έχουν μεγαλύτερο μέγεθος γιατί περιέχουν και το vlan tag (4 bytes)
11. Ethertype → 802.1Q (8100)
12. 4 bytes μετά το πεδίο length
13. `tcpdump -i em0.5 -e -vvv -x`
14. Ethertype ARP → 0806
Ethertype IPv4 → 0800
Όχι, δεν υπάρχει κάποιο σχετικό πεδίο
15. `ifconfig bridge1 tcp em0`, `tcp -i em0 -e -vvv`
16. Στη θέση του Ethertype υπάρχει το 802.3
17. `tcpdump not stp`