

Όνοματεπώνυμο: Περόγαμβρος Γεώργιος		Όνομα PC:
Ομάδα: 2	Ημερομηνία: 23/3/2022	

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 Χρησιμοποίησα τις εντολές `ifconfig em0 192.168.1.1/24 (PC1)` και `ifconfig em0 192.168.1.2/24 (PC2)`.
- 1.2 Χρησιμοποίησα τις εντολές `ifconfig em0 up`, `ifconfig em1 up`.
- 1.3 Και στις δύο περιπτώσεις το `ping` απέτυχε με το μήνυμα `host is down`.
- 1.4 Κάνοντας `ping` από το PC1 στο PC2 και χρησιμοποιώντας `tcpdump` στο PC2 δεν καταγράφονται καθόλου πακέτα. Κάνοντας `tcpdump` στην `em0` του B1 βλέπω πως παράγονται `arp requests` από το PC1, αλλά κανένα ICMP πακέτο. Αντίστοιχα αποτελέσματα είχα και όταν έκανα `ping` από το PC2 στο PC1 και `tcpdump` στο PC1 και στην `em1` του B1. Καταλαβαίνουμε, λοιπόν, πως τα δύο δίκτυα LAN1 και LAN2 δεν επικοινωνούν μεταξύ τους.
- 1.5 Χρησιμοποίησα τις εντολές `ifconfig bridge0 create` και `ifconfig bridge0 addm em0 addm em1 up`.
- 1.6 Ναι, υπάρχει.
- 1.7 Το `ttl` δεν μειώνεται από την γέφυρα (η γέφυρα δεν επεξεργάζεται τις επικεφαλίδες `ip`) και άρα βάσει του `ttl` απέχουν μηδέν βήματα.
- 1.8 Παρατηρώ πως το PC1 γνωρίζει την MAC που αντιστοιχεί στην IP του PC2 και αντιστρόφως.
- 1.9 Χρησιμοποίησα τις εντολές `tcpdump -i em0 -e -v` και `tcpdump -i em1 -e -v`. Αφού η κίνηση των μηνυμάτων καταγράφεται τόσο στο LAN1 όσο και στο LAN2, τα μηνύματα πρέπει να προωθούνται μέσω της γέφυρας.
- 1.10 Όχι.
- 1.11 Όχι.
- 1.12 Δεν υπάρχει ένδειξη της ύπαρξης του B1, αφού οι γέφυρες δεν εμπλέκονται στο στρώμα ζεύξης δεδομένων και δεν μειώνουν το `ttl` των μηνυμάτων ώστε να είναι ανιχνεύσιμες από την `traceroute`.
- 1.13 Χρησιμοποίησα τις εντολές `ping 192.168.1.2` στο PC1 και την εντολή `tcpdump -i em1` στο B1.
- 1.14 Ναι, τα προωθεί.
- 1.15 Το `ping` σταματάει να είναι επιτυχές, αφού το PC1 εξακολουθεί να κάνει `ping` στην παλιά `ip` του PC2. Τα μηνύματα προωθούνται από τη γέφυρα, αφού αυτό γίνεται μέσω του πίνακα προώθησης, ο οποίος έχει MAC διευθύνσεις. Το PC2 όμως δεν απαντάει αφού έχει άλλη `ip`.

1.16 Όχι.

1.17 Χρησιμοποίησα τις εντολές `ifconfig em2 up` και `ifconfig bridge0 addm em2`.

1.18 Ναι, λαμβάνω.

1.19 Όχι, καθώς η κίνηση περιορίζεται μεταξύ των LAN1 και LAN3, αφού τα PC1 και PC3 γνωρίζουν ήδη το αντίστοιχο ζεύγος MAC-ip και η γέφυρα έχει αποθηκευμένο το ζεύγος της MAC του PC3 με την αντίστοιχη θύρα του LAN3.

1.20 Η διαγραφή γίνεται με την εντολή `arp -da`. Εμφανίζεται ένα `arp request` στην καταγραφή από τον PC1 που προσπαθεί να μάθει την mac του PC3. Αφού το `arp` είναι broadcast η γέφυρα το προωθεί και στο LAN2.

1.21 Με την εντολή `ifconfig bridge0`.

1.22 Με την εντολή `ifconfig bridge0 addr`.

1.23 Αντιστοιχούν στα μηχανήματα PC1, PC2, PC3.

1.24 Χρησιμοποίησα την εντολή `ifconfig bridge0 flush`.

1.25 Χρησιμοποίησα την εντολή `ifconfig bridge0 delete em2`.

1.26 Χρησιμοποίησα την εντολή `ifconfig bridge0 destroy`.

1.27 Χρησιμοποίησα την εντολή `ifconfig em0 delete` στα τρία μηχανήματα.

2

2.1 Χρησιμοποίησα την εντολή `ifconfig em0 ip` όπου `ip` η αντίστοιχη διεύθυνση του κάθε PC.

2.2 Χρησιμοποίησα τις εντολές `ifconfig em0 up`, `ifconfig em1 up`, `ifconfig bridge1 create`, `ifconfig bridge1 addm em0 addm em1 up`.

2.3 Χρησιμοποίησα τις εντολές `ifconfig em0 up`, `ifconfig em1 up`, `ifconfig bridge2 create`, `ifconfig bridge2 addm em0 addm em1 up`.

2.4 Χρησιμοποίησα τις εντολές `ifconfig em0 up`, `ifconfig em1 up`, `ifconfig bridge3 create`, `ifconfig bridge3 addm em0 addm em1 up`.

2.5 Έχουμε τα εξής:

PC1: 08:00:27:d1:72:44

PC2: 08:00:27:ef:b2:a1

PC3: 08:00:27:e4:eb:1e

PC4: 08:00:27:dc:a7:da

Για το άδειασμα των πινάκων χρησιμοποίησα την εντολή `arp -da`.

2.6 Χρησιμοποίησα την εντολή `ifconfig bridgex flush` όπου `x` ο αριθμός της εκάστοτε γέφυρας.

2.7 `tcpdump`

2.8

Bridge1: 08:00:27:ef:b2:a1 Vlan1 em1 1196 flags=0<>, 08:00:27:d1:72:44 Vlan1 em0 1196 flags=0<>

Bridge2: 08:00:27:ef:b2:a1 Vlan1 em0 1192 flags=0<>, 08:00:27:ef:72:44 Vlan1 em0 1192 flags=0<>

Bridge3: 08:00:27:ef:72:44 Vlan1 em0 1188 flags=0<>

2.9 Αρχικά, ο PC1 έκανε broadcast το arp request για να μάθει την mac του PC2, το οποίο πέρασε και από τις 3 γέφυρες και έτσι και οι 3 ξέρουν σε ποιο LAN βρίσκεται ο PC1 με mac 08:00:27:ef:72:44 (μέσω της διεπαφής em0 και για τις 3 γέφυρες). Τόσο το arp reply του PC2 όσο και τα ακόλουθα ICMP echo request και ICMP reply δεν γίνονται broadcast, οπότε δεν περνάνε από τη γέφυρα B3, παρά μόνο από τις B1 και B2. Έτσι, μόνο αυτές οι δύο ενημερώνονται για το LAN του PC2 με mac 08:00:27:ef:b2:a1, που αντιστοιχεί στις θύρες em1 για τη B1 και em0 για τη B2.

2.10 Όχι, δεν υπήρξαν αφού στάλθηκαν απλά τα μηνύματα ICMP echo request και echo reply μεταξύ των 2 υπολογιστών.

2.11 Λόγω του ότι το B1, βρίσκεται στο LNK1, όπως και το PC2, ακούει τόσο το arp reply όσο και το icmp reply από τον PC4 και έτσι ενημερώνεται για τη mac του.

2.12 Και στους 3 πίνακες προστέθηκε η mac του PC3. Αυτό συνέβη καθώς, όπως και στο προηγούμενο ερώτημα, το request του PC3 ακούστηκε από όλες τις γέφυρες, καθώς πέρασε και από το LNK2 και από το LNK1.

2.13 ping 192.168.1.2

2.14 Το ping συνεχίζει επιτυχώς.

2.15 Το ping του PC1 σταματάει να είναι επιτυχές, καθώς η B1 συνεχίζει να προσπαθεί να προωθήσει τα ICMP echo request στην παλιά διεύθυνση του PC2 στο LNK1, οπότε ο PC2 δεν τα λαμβάνει ποτέ. Η διεύθυνση του PC2 θα ενημερωθεί όταν αυτός παράξει κάποιο πλαίσιο που θα φτάσει στο LNK1 ή στο LAN1. Όσον αφορά το ping από το PC4 αυτό είναι επιτυχές αφού πλέον και τα δύο PC βρίσκονται στο ίδιο τοπικό δίκτυο LAN2 και έτσι τα μηνύματα παραδίδονται άμεσα.

2.16 Λόγω του πλαισίου από το PC2 στο LNK2 ο B2 ενημερώθηκε για τη νέα θέση του. Έτσι, τα πλαίσια του PC1 που προωθεί ο B1 στο LNK1 τα λαμβάνει ο B2 και τα προωθεί σωστά προς το LNK2 και τέλος ο B3 προς το LAN2.

2.17 Θα έπρεπε να περιμένουμε μερικά λεπτά μέχρι να διαγραφεί η mac του PC2 από τον πίνακα προώθησης του B2 για να ενημερωθεί αυτός για τη σωστή διεύθυνση μέσω flooding.

3

3.1 ifconfig bridge0 create, ifconfig bridge 0 addm em0 addm em1 up, ifconfig em0 up, ifconfig em1 up

3.2 ifconfig bridge1 create, ifconfig bridge 1 addm em0 addm em2 up, ifconfig em0 up, ifconfig em2 up

3.3 PC1: 08:00:27:d1:72:44

PC2: 08:00:27:ef:b2:a1

PC3: 08:00:27:e4:eb:1e

Χρησιμοποίησα την εντολή arp -da για το άδειασμα.

3.4 Παρατηρείται μόνο ένα arp request που γίνεται broadcast αφού αδειάσαμε τους πίνακες arp. Τα υπόλοιπα μηνύματα δεν φαίνονται στο δίκτυο του PC1 αφού τα PC2 και PC3 είναι στο ίδιο δίκτυο.

3.5 Χρησιμοποίησα την εντολή ping 192.168.1.1

3.6 `ifconfig bridge0 addm em2, ifconfig em2 up, ifconfig bridge1 addm em1, ifconfig em1 up.`

3.7 `ifconfig bridge0 addr, ifconfig bridge1 addr`

3.8 Για το B1 έχουμε : PC1 → em0, PC3 → em1 και για το B2 έχουμε: PC1 → em0, PC3 → em2

3.9 Χρησιμοποίησα την εντολή `tcpdump`.

3.10 Χρησιμοποίησα τις εντολές `arp -da, ping 192.168.1.1`. Το ping δεν ήταν επιτυχές.

3.11 Και οι δύο υπολογιστές φαίνονται να είναι στην διεπαφή em1 βάσει του πίνακα προώθησης. Αυτό συμβαίνει διότι το `arp broadcast` που στέλνει το PC3 προωθείται στο B2 μέσω του LAN2 και αφού προωθηθεί στο B1, αυτό το προωθεί τόσο στο LNK2, όσο και στο LAN1, έτσι το request του PC3 ξαναφθάνει στο B2 με αποτέλεσμα το B2 να θεωρεί ως διεπαφή του PC3 την em1. Επίσης, το reply του PC1 προωθείται στο LNK2 κι έτσι το B2 θεωρεί διεπαφή του PC1 επίσης την em1. Αυτό το reply δεν προωθείται ποτέ στο LAN2, αφού το B2 έχει τη διεύθυνση του PC3 στη διεπαφή em1 και όχι στην διεπαφή em2.

3.12 Arp Request: Request who-has 192.168.1.1 tell 192.168.1.3

Arp Reply: Reply 192.168.1.1 is-at 08:00:27:d1:72:44

3.13 Όπως αναφέραμε και στο 3.11, το `arp request` έχει “παγιδευτεί” λόγω του κύκλου LNK1-LNK2, και επαναπροωθείται συνέχεια σε όλο το δίκτυο, λόγω broadcast. Τα `arp reply` δεν παρατηρούνται επειδή το B2 προωθεί τα μηνύματα για το PC3 προς το LNK2.

3.14 Επειδή τα `arp request` προωθούνται με broadcast το B2 τα προωθεί στα LNK1,LNK2,LAN2 και το B1 τα προωθεί στα LNK1,LNK2,LAN1, οπότε τα B1 και B2 λαμβάνουν συνέχεια το ίδιο `arp request` το οποίο και επαναπροωθούν.

3.15 Τα `arp reply` δεν παρατηρούνται επειδή το B2 προωθεί τα μηνύματα για το PC3 προς το LNK2, αφού σύμφωνα με τον πίνακα προώθησης της bridge2 τα μηνύματα του PC3 προωθούνται μέσω της διεπαφής em1 και όχι της em2 που είναι η διεπαφή του LAN2.

4

4.1 `ifconfig bridge0 destroy, ifconfig em0 down, ifconfig em1 down, ifconfig em2 down, ifconfig bridge1 create` στο B1

`ifconfig bridge1 destroy, ifconfig em0 down, ifconfig em1 down, ifconfig em2 down, ifconfig bridge2 create` στο B2

4.2 `ifconfig em0 up, ifconfig em1 up, ifconfig em2 up, ifconfig lagg0 create`

- 4.3 `ifconfig lagg0 up laggport em1, ifconfig lagg0 up laggport em2.`
- 4.4 `ifconfig em0 up, ifconfig em1 up, ifconfig em2 up, ifconfig lagg0 create, ifconfig lagg0 up laggport em0, ifconfig lagg0 up laggport em1`
- 4.5 `ifconfig bridge1 addm em0 addm lagg0 up.`
- 4.6 `ifconfig bridge2 addm lagg0 addm em2 up.`
- 4.7 Τα PC2 και PC3 βρίσκονται στο ίδιο LAN, οπότε στο LAN1 εμφανίζεται μόνο το arp request που έγινε broadcast από το PC2.
- 4.8 `tcpdump`
- 4.9 Το ping είναι επιτυχές. Παρατηρήθηκαν ένα arp request και ένα arp reply.
- 4.10 Παρατηρούνται ICMP μόνο στο LNK1. Αυτό συμβαίνει επειδή το προεπιλεγμένο πρωτόκολλο failover προωθεί την κίνηση μέσω της masterport, η οποία είναι η θύρα η οποία είναι η θύρα που προσθέσαμε πρώτη στο link aggregation.
- 4.11 Τα πακέτα μεταφέρονται πλέον μέσω του LNK2.
- 4.12 Ναι, η κίνηση ξαναγίνεται μέσω του LNK1, αφού σύμφωνα με το πρωτόκολλο η em1 είναι η masterport.

5

- 5.1 Χρησιμοποίησα τις εντολές `ifconfig bridge1 destroy, ifconfig em0 down, ifconfig em1 down, ifconfig em2 down, ifconfig lagg0 create` στο B1
`ifconfig bridge2 destroy, ifconfig em0 down, ifconfig em1 down, ifconfig em2 down, ifconfig lagg0 destroy` στο B2
- 5.2 Χρησιμοποίησα τις εντολές `ifconfig bridge1 create, ifconfig bridge1 addm em0 addm em1 addm em2 up, ifconfig em0 up, ifconfig em1 up, ifconfig em2 up.`
- 5.3 Χρησιμοποίησα τις εντολές `ifconfig bridge2 create, ifconfig bridge2 addm em0 addm em1 addm em2 up, ifconfig em0 up, ifconfig em1 up, ifconfig em2 up.`
- 5.4 Χρησιμοποίησα την εντολή `ifconfig bridge1 stp em0 stp em1 stp em2.`
- 5.5 Χρησιμοποίησα την εντολή `ifconfig bridge2 stp em0 stp em1 stp em2.`
- 5.6 Bridge1: 08:00:27:1d:de:eb
Bridge2: 08:00:27:0e:78:23
- Τα βρήκα με την εντολή `ifconfig bridge1, ifconfig bridge2`
- 5.7 Η γέφυρα ρίζα είναι η bridge2, αφού το root id είναι ίδιο με το bridge id της.
- 5.8 Και οι τρεις διεπαφές της root bridge έχουν ρόλο designated και κατάσταση forwarding.
- 5.9 Η ριζική θύρα είναι η LNK1.
- 5.10 Ο ρόλος της είναι alternate και η κατάσταση discarding.
- 5.11 Ο ρόλος της είναι designated και η κατάσταση forwarding.
- 5.12 Χρησιμοποίησα την εντολή `tcpdump -i em0 -vvv -e` στην B2. Εκπέμπονται BPDUs κάθε 2 δευτερόλεπτα.

5.13 Είναι IEEE 802.3

5.14 Η διεύθυνση mac πηγής είναι η 08:00:27:0e:78:23, ενώ η προορισμού είναι η 01:80:c2:00:00:00

5.15 Στη LNK1.

5.16 Είναι multicast επειδή το least significant bit του most significant byte είναι 1.

5.17 rootId: 8000. 08:00:27:0e:78:23, bridgeId: 8000. 08:00:27:0e:78:23.8001, rootPathCost: 0.

5.18 BridgeID: 8000. 08:00:27:0e:78:23.8002. Το 8000 είναι η προτεραιότητα σε δεκαεξαδικό ($0x8000 = 32768$) και το 8002 είναι το id της θύρας ($0x80 = 128$ $0x02 = 2$ προτεραιότητα θύρας και θύρα αντίστοιχα).

5.19 Καταγράφονται πακέτα BDPU τα οποία δεν παράγει η ίδια, αλλά είναι τα παραγόμενα από τη γέφυρα ρίζα.

5.20 Παράγονται στην em0 που είναι η διεπαφή του LAN1.

5.21 rootId: 8000. 08:00:27:0e:78:23, bridgeId: 8000. 08:00:27:1d:de:eb.8001, rootPathCost: 20000

5.22 Χρησιμοποίησα την εντολή ping 192.168.1.2. Ναι, είναι επιτυχές.

5.23 Πέρασαν περίπου 7 δευτερόλεπτα. Αυτό είναι λογικό, αφού στο RSTP ο χρόνος ανίχνευσης σφάλματος είναι $3 * hellotime$, δηλαδή στην προκειμένη $3 * 2 = 6$ δευτερόλεπτα.

5.24 Όχι.

6

6.1 Χρησιμοποίησα τις εντολές ifconfig bridge1 addm em3, ifconfig bridge1 stp em3, ifconfig em3 up.

6.2 Χρησιμοποίησα τις εντολές ifconfig bridge2 addm em3, ifconfig bridge2 stp em3, ifconfig em3 up.

6.3 Χρησιμοποίησα τις εντολές ifconfig bridge3 create, ifconfig bridge3 addm em0 addm em1 addm em2 up, ifconfig bridge2 stp em0 stp em1 stp em2, ifconfig em0 up, ifconfig em1 up, ifconfig em2 up

6.4 Χρησιμοποίησα τις εντολές ifconfig bridge1 flush, ifconfig bridge2 flush, ifconfig bridge3 flush, ping 192.168.1.2, ping 192.168.1.3. Και τα δύο ping ήταν επιτυχή.

6.5 Χρησιμοποίησα την εντολή ifconfig bridge1 priority 0.

6.6 Τα κόστη είναι για το LNK1: 20000 για το LNK2: 20000 για το LNK4: 20000. Έχουμε ότι $20000 = 1\text{Gigabit}$

6.7 Από τη B1 είναι 0 και από τη B2 είναι 20000. Οι τιμές είναι η απόσταση από τη root γέφυρα.

6.8 Η LNK3, αφού έχει το μικρότερο κόστος.

6.9 Είναι alternate με state discarding.

6.10 Είναι 20000

6.11 Χρησιμοποίησα την εντολή ping 192.168.1.3

6.12 Χρησιμοποίησα την εντολή ifconfig bridge3 ifpathcost em0 40001, γιατί το κόστος από τη

LNK4 είναι $20000+20000=40000$ και για να την προτιμήσει πρέπει το κόστος από τη LNK3 να είναι χαμηλότερο.

6.13 Χρειάστηκαν περίπου 3 δευτερόλεπτα.

6.14 Ο ρόλος είναι alternate και η κατάσταση discarding.

6.15 Όχι, δεν άλλαξαν.

6.16 Ναι, υπήρξε αλλαγή. Έγινε 40000.

6.17 Χρειάστηκαν περίπου 9 δευτερόλεπτα.

6.18 Χρειάστηκαν περίπου 3 δευτερόλεπτα.

7

7.1 Χρησιμοποίησα τις εντολές `ifconfig em0.5 create vlan 5 vlandev em0 inet 192.168.5.1/24` και `ifconfig em0.6 create vlan 6 vlandev em0 inet 192.168.6.1/24`

7.2 Χρησιμοποίησα την εντολή `ifconfig em0.6 create vlan 6 vlandev em0 inet 192.168.6.2/24`

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

7.4 Ναι, μπορούμε.

7.5 Όχι, καθώς η 192.168.5.3 ανήκει στο VLAN5 και το PC2 δεν έχει διεπαφή στο VLAN5.

7.6 Όχι, καθώς η 192.168.6.2 ανήκει στο VLAN6 και το PC3 δεν έχει διεπαφή στο VLAN6.

7.7 Χρησιμοποίησα την εντολή `ifconfig bridge1 -stp em0`.

7.8 Χρησιμοποίησα την εντολή `tcpdump -i em0 -e -xx`.

7.9 Χρησιμοποίησα τις εντολές `arp -da`, `ping -c 1 192.168.1.1`

Για τα arp είναι ethertype: arp (0x0806) και για τα icmp είναι ethertype: ipv4 (0x0800)

7.10 Έχουν ένα επιπλέον tag μετά τη διεύθυνση προορισμού το VLAN tag και πριν τον τύπο του πλαισίου ethernet, μήκους 4 byte, το οποίο περιέχει πληροφορίες για το VLAN.

7.11 Το ethertype έχει τιμή και στα arp και στα ICMP 802.1Q. Μπορούμε όμως να τα ξεχωρίσουμε αφού μετά το VLAN tag ακολουθεί ο τύπος του αρχικού πλαισίου ethernet. (Όπως και στην 7.9)

7.12 Βρίσκεται αμέσως μετά την διεύθυνση mac προορισμού και το πεδίο τύπου του ethernet.

7.13 Χρησιμοποίησα την εντολή `tcpdump -i em0.5 -e -vvv -xx`.

7.14 Χρησιμοποίησα τις εντολές `arp -da` `ping 192.168.5.1`

Έχω ethertype 0x0806 για τα arp και 0x0800 για τα ipv4. Δεν υπάρχει πεδίο για το VLAN.

7.15 Χρησιμοποίησα τις εντολές `ifconfig bridge1 stp em0` και `tcpdump -i em0 -e -vvv -xx`.

7.16 Όχι, δεν είναι ίδια. Στη θέση του ethertype υπάρχει το 802.3.

7.17 Μπορούμε να χρησιμοποιήσουμε το φίλτρο “not stp”.