Εργαστήριο Δικτύων Υπολογιστών

Εργαστηριακή Άσκηση 5
Στατική δρομολόγηση

Ονοματεπώνυμο: Νικόλαος Παγώνας, el18175	**Όνομα PC:** nick-ubuntu
Ομάδα: 1 (Τρίτη 10:45)	**Ημερομηνία Εξέτασης:** Τρίτη 29/03/2022

Άσκηση 1: Δρομολόγηση σε ένα βήμα

1.1

Με τις εντολές:

```
PC1:
ifconfig em0 192.168.1.2/24

PC2:
ifconfig em0 192.168.2.2/24

R1:
ifconfig em0 192.168.1.1/24
ifconfig em1 192.168.2.1/24
```

1.2

Προσθέσαμε τη γραμμή:

```
gateway_enable="YES"
```

1.3

Προσθέτουμε τη ζητούμενη εγγραφή με την εντολή:

```
route add -net 192.168.2.0/24 192.168.1.1
```

1.4

Εκτελούμε `netstat -rn`. Η σημαία ”U” σημαίνει ότι η διαδρομή είναι ενεργή, η σημαία ”G” σημαίνει ότι ο προορισμός είναι πύλη, που θα αποφασίσει για το πώς θα προωθήσει τα πακέτα περαιτέρω, ενώ η σημαία ”S” σημαίνει ότι η διαδρομή έχει οριστεί στατικά.

1.5

Δοκιμάζουμε `ping 192.168.2.2` από το PC1 στο PC2 και παρατηρούμε ότι το PC1 δε λαμβάνει απάντηση.

1.6

Με `tcpdump -i em0` στα PC1 και PC2, παρατηρούμε ότι παράγονται πακέτα ICMP τόσο στο LAN1 όσο και στο LAN2. Παρολαυτά, στο PC2 δεν υπάρχει η κατάλληλη εγγραφή στον πίνακα δρομολόγησης ώστε να μπορεί να στείλει το PC2 ICMP reply στο PC1.

1.7

Προσθέτουμε τη ζητούμενη εγγραφή με την εντολή:

```
route add -net 192.168.1.0/24 192.168.2.1
```

1.8

Δοκιμάζουμε πάλι `ping 192.168.2.2` από το PC1 στο PC2. Πλέον υπάρχει επικοινωνία.

1.9

Δεν χρειάζεται να αλλάξουμε τον πίνακα δρομολόγησης του R1, διότι ο R1 διαθέτει διεπαφές και στα δύο υποδίκτυα (την `em0` στο `192.168.1.0/24` και την `em1` στο `192.168.2.0/24`), και άρα οι απαιτούμενες εγγραφές υπάρχουν ήδη στον πίνακα δρομολόγησής του.

Άσκηση 2: Proxy ARP

2.1

Καταργούμε τη στατική εγγραφή στο PC1 με την εντολή `route del 192.168.2.0/24`.

2.2

Αλλάζουμε το μήκος προθέματος στο PC1 με την εντολή `ifconfig em0 192.168.1.2/20`.

2.3

Από την προοπτική του PC1, τα PC2 και PC3 βρίσκονται στο ίδιο υποδίκτυο με αυτόν, αφού:

```
PC1 IP & PC1 subnet mask = 192.168.1.2 & 255.255.240.0 = 192.168.0.0
PC2 IP & PC1 subnet mask = 192.168.2.2 & 255.255.240.0 = 192.168.0.0
PC3 IP & PC1 subnet mask = 192.168.2.3 & 255.255.240.0 = 192.168.0.0
```

2.4

Από το PC1 κάνουμε `ping 192.168.2.2` και `ping 192.168.2.3`. Και από τα δύο ping λαμβάνουμε "Host is down".
Στον δρομολογητή ενεργοποιούμε το proxy ARP με την εντολή:

```
sysctl net.link.ether.inet.proxyall=1
```

2.5

Επαναλαμβάνουμε το `ping -c 1 192.168.2.2`. Τώρα το ping είναι επιτυχές. Αυτό συμβαίνει διότι ενεργοποιήσαμε τη λειτουργία proxy ARP στον R1, οπότε ο R1 απαντά στο ARP request του PC1 -που θέλει να μάθει τη φυσική διεύθυνση του PC2- με τη δική του διεύθυνση MAC, προωθεί το ICMP echo request στον PC2, απαντά πάλι με τη δική του διεύθυνση MAC στο ARP request του PC2 -που θέλει να μάθει τη φυσική διεύθυνση του PC1-, και τέλος προωθεί το ICMP echo reply του PC2 στο PC1.

2.6

Επαναλαμβάνουμε το `ping 192.168.2.3`. Το ping αποτυγχάνει, διότι δεν υπάρχει εγγραφή στον πίνακα δρομολόγησης του PC3 σχετική με το PC1, ώστε το PC3 να μπορεί να στείλει ICMP echo reply, όπως έγινε με το PC2.

2.7

Προσθέτουμε τη στατική εγγραφή στο PC3 με την εντολή:

```
route add 192.168.1.0/24 192.168.2.1
```

2.8

Καθαρίζουμε τους πίνακες ARP με την εντολή `arp -d -a`.

2.9

Ξεκινάμε τις ζητούμενες καταγραφές στο R1 με τις εντολές:

```
tcpdump -e -i em0
tcpdump -e -i em1
```

Επαναλαμβάνουμε το προηγούμενο ping από το PC1 στο PC3: `ping -c 1 192.168.2.3`.

2.10

Παρατηρούμε ότι ο R1 απαντά στο ARP request του PC1 με τη δική του φυσική διεύθυνση, και συγκεκριμένα με τη φυσική διεύθυνση της διεπαφής του στο LAN1.

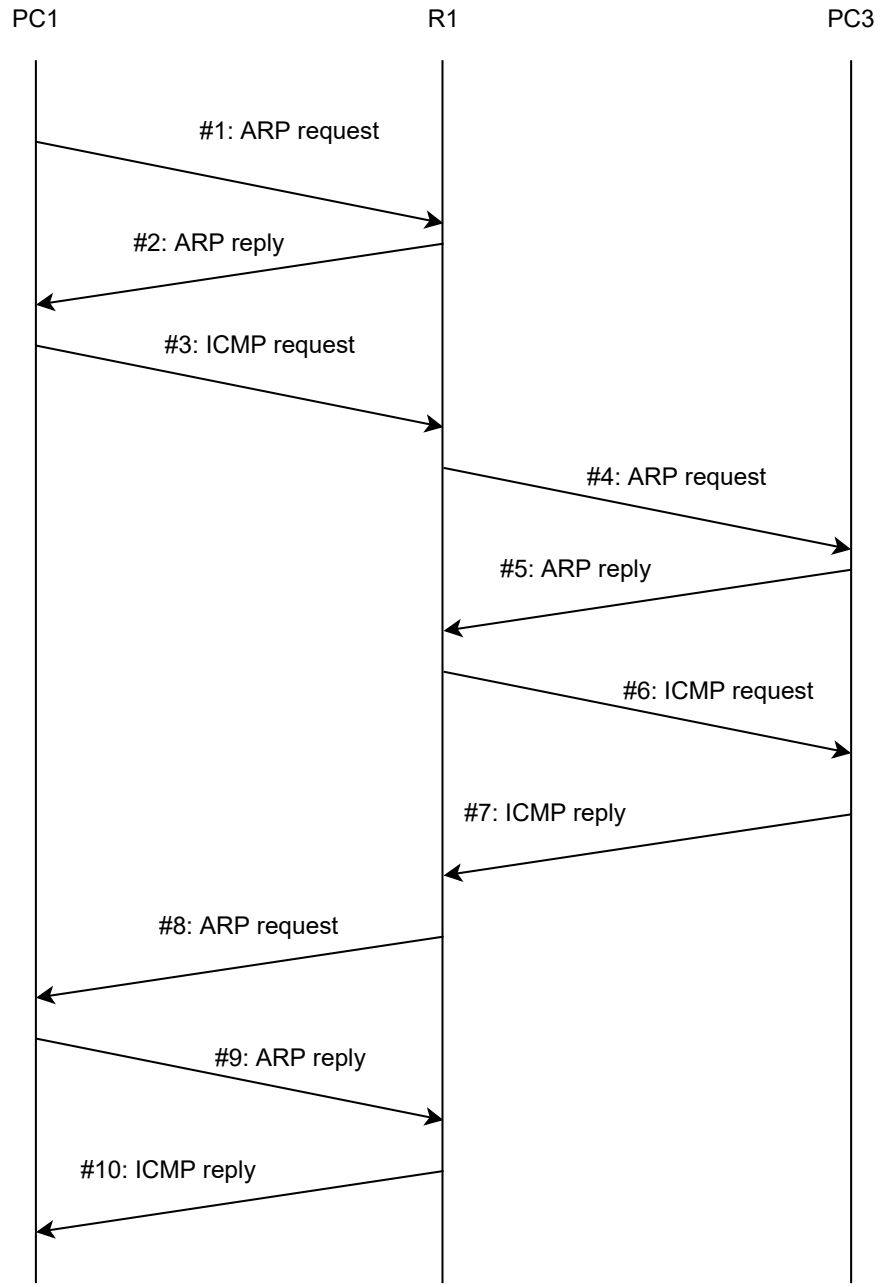
2.11

Ο PC1 στέλνει το ICMP request προς τη διεύθυνση `08:00:27:57:dd:d7`, δηλαδή τη φυσική διεύθυνση της διεπαφής του R1 στο LAN1.

2.12

Ο PC3 λαμβάνει το ICMP request από τη φυσική διεύθυνση της διεπαφής του R1 στο LAN2.

2.13



- #1: Επίλυση της φυσικής διεύθυνσης του PC3 για το PC1.
- #2: Proxy ARP, απάντηση στο ARP request #1. Ο R1 απαντά με τη δική του φυσική διεύθυνση αντί για τη διεύθυνση του PC3.
- #3: ICMP request από το PC1 με προορισμό το PC3.
- #4: Επίλυση της φυσικής διεύθυνσης του PC3 για το R1.

- #5: Απάντηση του ARP reply #4.
- #6: Προώθηση του ICMP request #3.
- #7: Απάντηση του ICMP request #6.
- #8: Επίλυση της φυσικής διεύθυνσης του PC1 για το R1. Να σημειωθεί ότι ο R1 δεν γνωρίζει τη φυσική διεύθυνση του PC1 λόγω του ARP request #1, αφού εκεί αναζητείται η φυσική διεύθυνση του PC3, και όχι του R1.
- #9: Απάντηση του ARP request #8.
- #10: Προώθηση του ICMP reply #7.

2.14

Ο πίνακας δρομολόγησης του PC1 (`netstat -rn`) έχει εγγραφές για τη loopback, το υποδίκτυο `192.168.0.0/20` και τη διεύθυνση `192.168.1.2`. Από αυτές η μόνη που μπορεί να φανεί χρήσιμη είναι αυτή του υποδικτύου `192.168.0.0/20`. Πρέπει λοιπόν το πρόθεμα δικτύου αυτής της εγγραφής να είναι τέτοιο ώστε να περιλαμβάνεται και το PC3 σε αυτό το υποδίκτυο.

```
PC3 IPv4:           11000000.10101000.00000010.00000011
PC1 subnet IPv4:    11000000.10101000.00000000.00000000
max prefix match == 22 bits
```

Επομένως το μέγιστο μήκος προθέματος για το PC1 είναι 22 bits.

2.15

Δίνουμε στο PC1 την εντολή `ifconfig em0 192.168.1.2/23`.

2.16

Στο PC1 εκτελούμε: `route add -net 192.168.2.0/24 -interface em0`

2.17

Στον πίνακα δρομολόγησης του PC1 εμφανίζεται:

```
Destination       Gateway              Netif
192.168.2.0/24    08:00:27:ca:53:74    em0
```

2.18

Τώρα το ping προς τον PC3 είναι επιτυχές. Αυτό συμβαίνει διότι ο PC1 βλέπει την εγγραφή `192.168.2.0/24` στον πίνακα δρομολόγησης και στέλνει το πακέτο στον R1, ο οποίος το προωθεί στον PC3, και ο PC3 βλέπει την εγγραφή `192.168.1.0` στον πίνακα δρομολόγησής του και μπορεί να απαντήσει. Για τα πακέτα ARP ισχύουν όσα έχουν αναφερθεί παραπάνω (proxy ARP κλπ).

2.19

Στον R1 ακυρώνουμε το proxy ARP με την εντολή: `sysctl net.link.ether.inet.proxyall=0`.

2.20

Στο PC1 εκτελούμε: `route change -net 192.168.2.0/24 192.168.1.1`.

2.21

Στο PC1 εκτελούμε: `ifconfig em0 192.168.1.2/24`.

2.22

Παρατηρούμε ότι η διαδρομή για το `192.168.2.0/24` δεν υπάρχει πλέον στον πίνακα δρομολόγησης του PC1. Την επαναπροσδιορίζουμε με την εντολή `route add -net 192.168.2.0/24 192.168.1.1`.

Άσκηση 3: Δρομολόγηση σε περισσότερα βήματα

3.1

Ορίζουμε διευθύνσεις IPv4 στον R1 με τις εντολές:

```
ifconfig em0 192.168.1.1/24
ifconfig em1 172.17.17.1/30
```

3.2

Ορίζουμε διευθύνσεις IPv4 στον R2 με τις εντολές:

```
ifconfig em0 172.17.17.2/30
ifconfig em1 192.168.2.1/24
```

3.3

Εκτελούμε `ping 192.168.2.2`. Εμφανίζεται μήνυμα ”Destination Host Unreachable”.

3.4

Με `tcpdump -e -i em0` και `tcpdump -e -i em1` στον R1 ελέγχουμε τα πακέτα που εκπέμπονται στα LAN1 και WAN1 αντίστοιχα. Στο LAN1 παρατηρούμε μηνύματα ”ICMP echo request” από το PC1 προς το R1, αλλά και μηνύματα ”ICMP host 192.168.2.2 unreachable” από το R1 στο PC1. Στο WAN1 δεν παρατηρούνται μηνύματα ICMP, αφού ο R1 δεν γνωρίζει πού να προωθήσει τα πακέτα που προορίζονται για το PC2 (δεν διαθέτει σχετική εγγραφή στον πίνακα δρομολόγησης).

3.5

Από το PC1 εκτελούμε `traceroute 192.168.2.2` και στην έξοδό της εντολής βλέπουμε 3 φορές το σύμβολο `!H`, που σημαίνει ότι ο PC1 έλαβε μήνυμα ”Destination host unreachable” από τον R1 (`192.168.1.1`).

3.6

Προσθέτουμε τη ζητούμενη εγγραφή μέσω της εντολής:

```
R1:
route add -net 192.168.2.0/24 172.17.17.2
```

3.7

Δοκιμάζουμε πάλι `ping 192.168.2.2` από το PC1 στο PC2. Δεν εμφανίζεται πλέον ”Destination Host Unreachable” αλλά πάλι δεν λαμβάνουμε απάντηση.

3.8

Με την εντολή `tcpdump -i em0` στο PC2 βλέπουμε ότι στο LAN2 παράγονται μηνύματα:

- ICMP echo request: Έχει προκύψει από την προώθηση του ICMP echo request του PC1 από τον R2.
- ICMP echo reply: Έχει προκύψει ως απάντηση στο ICMP echo request από τον PC2.
- ICMP host 192.168.1.2 unreachable: Έχει προκύψει από τον R2 και προορίζεται για τον PC2, επειδή ο R2 δεν γνωρίζει πού να προωθήσει το ICMP echo reply του PC2 που προορίζεται για τον PC1 (δεν έχει εγγραφή στον πίνακα δρομολόγησης σχετική με το PC1).

3.9

Δοκιμάζουμε ξανά `traceroute 192.168.2.2` από το PC1 προς το PC2. Δεν παρατηρούμε μηνύματα ICMP echo request στο WAN1 (εκτελέσαμε `tcpdump -i em1` στο R1). Αντ’ αυτού, παρατηρήσαμε UDP datagrams, επειδή η traceroute στέλνει UDP datagrams by default.

3.10

Στο LAN2 (εκτελέσαμε `tcpdump -i em1` στο R2) παρατηρούνται UDP datagrams και μηνύματα ”ICMP 192.168.2.2 udp port 3344x unreachable”, όπου $x = \{2, 3, 4, 5\}$.

3.11

Σύμφωνα με την ιστοσελίδα, δεν παράγεται το error message ”ICMP host unreachable” ως απάντηση στο error message ”ICMP 192.168.2.2 udp port 3344x unreachable”, διότι εξ’ ορισμού δεν παράγονται ICMP μηνύματα λάθους ως απάντηση σε άλλα ICMP μηνύματα λάθους (κάτι τέτοιο θα δημιουργούσε έναν κατακλυσμό από ICMP μηνύματα λάθους).

3.12

Στο R2 εκτελούμε την εντολή `route add 192.168.1.0/24 172.17.17.1`.

3.13

Πλέον μπορούμε να κάνουμε `traceroute 192.168.2.2` από το PC1 στο PC2. Στο WAN1 (εκτελέσαμε `tcpdump -i em1` στο R1) παράγονται μηνύματα "ICMP time exceeded in-transit", επειδή η traceroute ως γνωστόν στέλνει UDP datagrams με σταδιακά αυξανόμενο TTL (οπότε κάποια αναπόφευκτα θα αποκτήσουν TTL=0 όσο προωθούνται προς τον προορισμό) και "ICMP 192.168.2.2 udp port 3344x unreachable" ($x = \{2, 3, 4\}$), αφού δεν υπάρχει κάποια εφαρμογή που "ακούει" στις συγκεκριμένες πόρτες στο PC2.

3.14

Κάνουμε `ping 172.17.17.1` από το PC2. Παρατηρούμε ότι εμφανίζεται το μήνυμα "No route to host".

3.15

Στο PC2 εκτελούμε `route del 192.168.1.0/24`.

3.16

Στο PC2 εκτελούμε `route add default 192.168.2.1`.

3.17

Στο PC2 εκτελούμε πάλι `ping 172.17.17.1`. Το ping είναι επιτυχές αυτή τη φορά.

3.18

Ο λόγος που το ping αρχικά δεν πετυχαίνει, ενώ στη συνέχεια πετυχαίνει, είναι επειδή την πρώτη φορά, στον πίνακα δρομολόγησης του PC2 δεν υπήρχε εγγραφή σχετική με τη διεύθυνση `172.17.17.1`. Τη δεύτερη φορά όμως, επειδή προστίθεται εγγραφή για την προεπιλεγμένη πύλη, το ICMP request στέλνεται στη διεπαφή του R2 στο LAN2 και αυτός το προωθεί με επιτυχία στον προορισμό.

Άσκηση 4: Ένα πιο πολύπλοκο δίκτυο με εναλλακτικές διαδρομές

4.1

Στο PC3 εκτελούμε:

```
ifconfig em0 up
ifconfig em0 192.168.2.3/24
```

4.2

Στο PC3 εκτελούμε:

```
route add -net 192.168.1.0/24 192.168.2.1
```

4.3

Για το R1 έχουμε:

```
em0: LAN1
ifconfig em0 192.168.1.1/24

em1: WAN1
ifconfig em1 172.17.17.1/30

em2: WAN2
ifconfig em2 172.17.17.5/30
```

4.4

Για το R2 έχουμε:

```
em0: WAN1
ifconfig em0 172.17.17.2/30

em1: WAN3
ifconfig em1 172.17.17.9/30

em2: LAN2
ifconfig em2 192.168.2.1/24
```

4.5

Για το R3 έχουμε:

```
em0: WAN2
ifconfig em0 172.17.17.6/30

em1: WAN3
ifconfig em1 172.17.17.10/30
```

4.6

Στον R1 εκτελούμε:

```
route add -net 192.168.2.0/24 172.17.17.2
```

4.7

Στον R2 εκτελούμε:

```
route add -net 192.168.1.0/24 172.17.17.1
```

4.8

Στον R3 εκτελούμε:

```
route add -net 192.168.1.0/24 172.17.17.5
route add -net 192.168.2.0/24 172.17.17.9
```

4.9

Στον R1 εκτελούμε:

```
route add -host 192.168.2.3 172.17.17.6
```

Εκτελούμε `netstat -rn`. Η σημαία που δηλώνει ότι πρόκειται για υπολογιστή είναι η "H".

4.10

Δοκιμάζουμε `traceroute 192.168.2.2` από το PC1 στο PC2. Βλέπουμε 3 βήματα.

4.11

Δοκιμάζουμε `ping -c 1 192.168.2.2` από το PC1 στο PC2. Το ICMP reply έχει TTL = 62, οπότε με βάση αυτή την τιμή βλέπουμε 3 βήματα, αφού:

- PC2: το πακέτο φεύγει με TTL = 64
- 1ος δρομολογητής, 1ο βήμα: μειώνει το TTL κατά 1, και το πακέτο φεύγει με TTL = 63
- 2ος δρομολογητής, 2ο βήμα: μειώνει το TTL κατά 1, και το πακέτο φεύγει με TTL = 62
- PC1, 3ο βήμα: το πακέτο φτάνει έχοντας TTL = 62.

4.12

Δοκιμάζουμε `traceroute 192.168.2.3` από το PC1 στο PC3. Βλέπουμε 4 βήματα.

4.13

Δοκιμάζουμε `ping -c 1 192.168.2.3` από το PC1 στο PC3. Το ICMP reply έχει TTL = 62, οπότε με βάση την τιμή αυτή βλέπουμε 3 βήματα. Η αιτιολόγηση είναι ίδια με αυτή του ερωτήματος 4.11.

4.14

Με βάση την έξοδο της εντολής traceroute, το ICMP echo request προς το PC3 ακολουθεί τη διαδρομή:

```
PC1 --> R1 --> R3 --> R2 --> PC3
```

4.15

Το ICMP echo reply προς το PC1 ακολουθεί τη διαδρομή:

```
PC3 --> R2 --> R1 --> PC1
```

Η διαφορά των δύο διαδρομών έγκειται στο ότι στην περίπτωση του ερωτήματος 4.14, ο R1 έχει δύο εγγραφές σχετικές με το PC3 στον πίνακα δρομολόγησης, μία που αναφέρεται στη διεύθυνση του PC3 και μόνο, και μία που αναφέρεται στο LAN2. Από τις δύο, ο R1 επιλέγει το ταίριασμα μέγιστου προθέματος, δηλαδή την εγγραφή που αναφέρεται στη διεύθυνση του PC3, και με βάση αυτή την εγγραφή προωθεί το πακέτο στον R3. Αντίθετα, στον πίνακα δρομολόγησης του R2 βρίσκεται η στατική εγγραφή που προσθέσαμε προηγουμένως, ώστε ο R2 να προωθεί πακέτα για το LAN1 μέσω του R1, η οποία είναι και η μόνη έγκυρη εγγραφή, οπότε είναι και αυτή που επιλέγεται.

4.16

Απενεργοποιούμε την διεπαφή του R1 στο WAN1 και ξεκινάμε τη ζητούμενη καταγραφή στο R2 με την εντολή `tcpdump -i em2`.

4.17

Δοκιμάζουμε `traceroute 192.168.2.2` από το PC1 στο PC2 και αφήνουμε να ολοκληρωθούν τουλάχιστον 3 βήματα. Δεν παρατηρούμε να φτάνουν ή να παράγονται πακέτα UDP στο PC2.

4.18

Δοκιμάζουμε `traceroute 192.168.2.3` από το PC1 στο PC3 και αφήνουμε να ολοκληρωθούν τουλάχιστον 4 βήματα. Παρατηρούμε ότι φτάνουν και παράγονται πακέτα UDP στο PC3.

4.19

Η ζητούμενη σύνταξη της εντολής `route` είναι:

```
R1:
route change -net 192.168.2.0/24 172.17.17.6

R2:
route change -net 192.168.1.0/24 172.17.17.10
```

Με χρήση της traceroute επιβεβαιώνουμε ότι υπάρχει επικοινωνία μετά την αλλαγή.

4.20

Στον R1 εκτελούμε `route show 192.168.2.2` και `route show 192.168.2.3`. Παρατηρούμε ότι στην πρώτη περίπτωση το πεδίο "destination" αναφέρεται στο υποδίκτυο LAN2 -γι' αυτόν τον λόγο υπάρχει και η επιπλέον πληροφορία για τη μάσκα υποδικτύου-, ενώ στην δεύτερη περίπτωση το πεδίο "destination" αναφέρεται συγκεκριμένα στον host PC3 -γι' αυτόν τον λόγο υπάρχει και η επιπλέον σημαία "HOST".

4.21

Επιλέγεται η εγγραφή που αναφέρεται στον host `192.168.2.3` (πεδίο destination: `192.168.2.3`), επειδή αυτή η εγγραφή προσφέρει το μέγιστο ταίριασμα προθέματος.

Άσκηση 5: Βρόχοι κατά τη δρομολόγηση

5.1

Τροποποιούμε τη ζητούμενη εγγραφή στον R3 με την εντολή:

```
route change -net 192.168.2.0/24 172.17.17.5
```

5.2

Εκτελούμε `ping -c 1 192.168.2.2` από το PC1 στο PC2. Το ping δεν είναι επιτυχές, καθώς εμφανίζεται μήνυμα "Time to live exceeded".

5.3

Το μήνυμα λάθους προέρχεται από τη διεπαφή του R3 στο WAN2, με διεύθυνση `172.17.17.6`.

5.4

Πρέπει να καταγράψουμε στον R1 και στο δίκτυο WAN2, επειδή εκεί έχει δημιουργηθεί ο βρόχος λόγω της στατικής εγγραφής που τροποποιήσαμε στο ερώτημα 5.1.

5.5

Πρέπει να εφαρμόσουμε το φίλτρο `"icmp[icmptype] == icmp-echo"`.

5.6

Ξεκινάμε στον R1 την καταγραφή που περιγράφηκε παραπάνω με την εντολή:

```
tcpdump -e -i em2 "icmp[icmptype] == icmp-echo"
```

Παρατηρούμε ότι λήφθηκαν 64 πακέτα, ενώ καταγράφηκαν 63 πακέτα. Το ένα πακέτο που λήφθηκε αλλά δεν καταγράφηκε είναι τύπου ICMP time exceeded in-transit. Έτσι έχουμε ότι εμφανίσθηκαν στο WAN2 63 πακέτα ICMP echo request. Τέλος, από το PC1 παράχθηκε μόνο ένα πακέτο ICMP echo request, όπως εξάλλου ορίσαμε και στην εντολή ping με την παράμετρο `-c 1`.

5.7

Ξεκινάμε τη ζητούμενη καταγραφή στο R1 με την εντολή `tcpdump -l -i em0 | tee capture1`. Αντίστοιχα στο R3 με την εντολή `tcpdump -l -i em0 | tee capture2`.

5.8

Εκτελούμε `traceroute -I -q 1 192.168.2.2` από το PC1 στο PC2. Εμφανίζονται 64 βήματα μέχρι να ολοκληρωθεί η εκτέλεση της εντολής. Η διαδρομή που καταγράφουμε είναι η:

```
PC1 --> R1 --> R3 --> R1 --> R3 --> ... --> R1 --> R3
```

δηλαδή ο βρόχος που έχει σχηματιστεί.

5.9

Σταματάμε τις καταγραφές. Με την εντολή `grep "ICMP echo request" capture1 | wc -l` στο R1 βρίσκουμε ότι στάλθηκαν 64 πακέτα ICMP echo request από το PC1.

5.10

Με την εντολή `grep "ICMP echo request" capture2 | wc -l` στο R3 βρίσκουμε ότι εμφανίστηκαν 2016 πακέτα ICMP echo request στο WAN2.

Η εξήγηση είναι η εξής:

Έστω ότι από το PC1 ξεκινάει ένα πακέτο με TTL $= T$. Αυτό θα σταλεί στον R1, και εκεί θα αποκτήσει TTL $= T - 1$. Ύστερα, θα μπει στον βρόχο R1 $\rightarrow$ R3 $\rightarrow$ R1..., μέχρι το TTL του να μηδενιστεί, οπότε και θα απορριφθεί από τον αντίστοιχο δρομολογητή. Επομένως, ένα πακέτο με TTL $= T$, θα εμφανιστεί συνολικά $T - 1$ φορές στο WAN2. Επειδή η εντολή traceroute στέλνει ένα πακέτο ανά βήμα (παράμετρος `-q 1`), και σε κάθε βήμα το TTL αυξάνεται κατά 1, ο συνολικός αριθμός πακέτων ICMP echo request που θα εμφανιστούν στο WAN2 είναι:

$$1 + 2 + 3 + ... + 63 = \frac{63 \cdot (63 + 1)}{2} = 2016 \text{ πακέτα.}$$

5.11

Με την εντολή `grep "ICMP time exceeded in-transit" capture2 | wc -l` βρίσκουμε ότι εμφανίστηκαν 32 πακέτα ICMP time exceeded στο WAN2. Αυτό συμβαίνει διότι από τα 64 πακέτα ICMP echo request που παράχθηκαν συνολικά από το PC1, τα μισά θα απορριφθούν από τον R1 (αυτά με αρχικό TTL $= 1, 3, 5, 7, ..., 63$), οπότε θα σταλεί μήνυμα ICMP time exceeded από τον R1 στο PC1 μέσω του LAN1, ενώ τα άλλα μισά θα απορριφθούν από τον R2 (αυτά με αρχικό TTL $= 2, 4, 6, 8, ..., 64$), οπότε θα σταλεί μήνυμα ICMP time exceeded από τον R2 στον R1 μέσω του WAN2. Επομένως, τα 32 μηνύματα ICMP time exceeded που καταγράφηκαν αφορούν τα μηνύματα ICMP echo request με αρχικό TTL άρτιο.

5.12

Μπορούμε να αποφύγουμε την αποθήκευση σε αρχείο ως εξής:

```
Question 5.9:
R1: tcpdump -i em0 "icmp[icmptype] == icmp-echo"

Question 5.10:
R3: tcpdump -i em0 "icmp[icmptype] == icmp-echo"

Question 5.11:
R3: tcpdump -i em0 "icmp[icmptype] == icmp-timexceed"
```

Σε κάθε περίπτωση η απάντηση που ψάχνουμε δίνεται αφού τερματίσουμε την εντολή tcpdump με Ctrl-C, στη γραμμή `"xxxx packets captured"`.

5.13

Τα πακέτα ICMP echo request που παράχθηκαν από την εντολή ping έχουν μήκος 64 bytes και αρχικό TTL = 64, ενώ τα πακέτα ICMP echo request που παράχθηκαν από την εντολή traceroute έχουν μήκος 28 bytes και αρχικό TTL που κυμαίνεται από 1 έως 64.

5.14

Τα πακέτα ICMP echo request δεν κυκλοφορούν αενάως στο δίκτυο, διότι κάθε φορά που το πακέτο περνά από έναν δρομολογητή, αυτός μειώνει το TTL του κατά 1, και όταν το TTL γίνει μηδέν, τότε ο δρομολογητής απορρίπτει το πακέτο και στέλνει μήνυμα ICMP time exceeded.

Άσκηση 6: Χωρισμός σε υποδίκτυα

6.1

LAN1, 120 υπολογιστές, πρέπει:

$$2^n - 2 > 120 \implies n = 7 \implies \text{Μήκος προθέματος LAN1} = 32 - 7 = 25.$$

Λαμβάνουμε υπόψιν μας το γεγονός ότι το μπλοκ διευθύνσεων 172.17.17.129-172.17.17.138 είναι δεσμευμένο από τους δρομολογητές.

```
129 dec = 1000 0001 bin
138 dec = 1000 1010 bin
```

βλέπουμε ότι μπορούμε να διαθέσουμε το μπλοκ 172.17.17.0-172.17.17.127, επομένως η διεύθυνση υποδικτύου του LAN1 είναι **172.17.17.0/25**.

6.2

LAN2, 60 υπολογιστές, πρέπει:

$$2^n - 2 > 60 \implies n = 6 \implies \text{Μήκος προθέματος LAN2} = 32 - 6 = 26.$$

βλέπουμε ότι μπορούμε να διαθέσουμε το μπλοκ 172.17.17.192-172.17.17.255, επομένως η διεύθυνση υποδικτύου του LAN2 είναι **172.17.17.192/26**.

6.3

LAN3, 30 υπολογιστές, πρέπει:

$$2^n - 2 > 30 \implies n = 5 \implies \text{Μήκος προθέματος LAN3} = 32 - 5 = 27.$$

βλέπουμε ότι μπορούμε να διαθέσουμε το μπλοκ 172.17.17.160-172.17.17.191, επομένως η διεύθυνση υποδικτύου του LAN3 είναι **172.17.17.160/27**.

6.4

Στο PC1 εκτελούμε:

```
ifconfig em0 172.17.17.1/25
```

Ενώ στον R1 εκτελούμε:

```
ifconfig em0 172.17.17.126/25
```

6.5

Στο PC4 εκτελούμε:

```
ifconfig em0 172.17.17.161/27
```

Ενώ στον R3 εκτελούμε:

```
ifconfig em0 172.17.17.190/27
```

6.6

Στον R2 εκτελούμε:

```
ifconfig em2 172.17.17.193/26
```

Στο PC2 εκτελούμε:

```
ifconfig em0 172.17.17.253/26
```

Ενώ στο PC3 εκτελούμε:

```
ifconfig em0 172.17.17.254/26
```

6.7

Εκτελούμε τις εντολές:

```
PC1: route add default 172.17.17.126
PC2: route add default 172.17.17.193
PC3: route add default 172.17.17.193
PC4: route add default 172.17.17.190
```

6.8

Στον R1 εκτελούμε:

```
route add -net 172.17.17.192/26 172.17.17.130
route add -net 172.17.17.160/27 172.17.17.130
```

6.9

Στον R2 εκτελούμε:

```
route add -net 172.17.17.0/25 172.17.17.137
route add -net 172.17.17.160/27 172.17.17.137
```

6.10

Στον R3 εκτελούμε:

```
route add -net 172.17.17.0/25 172.17.17.133
route add -net 172.17.17.192/26 172.17.17.133
```

6.11

Για να επιβεβαιώσουμε ότι υπάρχει επικοινωνία ανάμεσα σε όλα τα LAN, εκτελούμε:

```
PC1: ping -c 1 172.17.17.253
PC2: ping -c 1 172.17.17.161
PC3: ping -c 1 172.17.17.1
```

Και τα τρία ping είναι επιτυχή.

Άσκηση 7: Ταυτόσημες διευθύνσεις IP

Επιβεβαιώνουμε ότι τα PC1, PC2 και PC3 επικοινωνούν μεταξύ τους εκτελώντας:

```
PC1 <-> PC2: ping -c 1 172.17.17.253
PC1 <-> PC3: ping -c 1 172.17.17.254
PC2 <-> PC3: ping -c 1 172.17.17.254
```

7.1

Με `ifconfig em0` στα PC2,3 βρίσκουμε ότι:

```
PC2 MAC address == 08:00:27:10:0c:23
PC3 MAC address == 08:00:27:be:32:ef
```

7.2

Στο PC2 εκτελούμε `ifconfig em0 172.17.17.254/26`.

7.3

Εμφανίστηκε στο PC2 η ένδειξη:

```
PC kernel: arp: 08:00:27:be:32:ef is using my IP address 172.17.17.254 on em0!
```

7.4

Ναι, εμφανίστηκε στο PC3 η ένδειξη:

```
PC kernel: arp: 08:00:27:10:0c:23 is using my IP address 172.17.17.254 on em0!
```

7.5

Ναι, με `ifconfig em0` στο PC2 βλέπουμε ότι η IP διεύθυνση έχει οριστεί κανονικά. Το νόημα των μηνυμάτων λάθους είναι απλώς να ενημερώσει τους χρήστες των δύο μηχανημάτων ότι μοιράζονται την ίδια διεύθυνση IPv4.

7.6

Όχι, ο R2 δεν παραμένει ως προεπιλεγμένη πύλη στο PC2 (`netstat -rn`), επειδή εκτελέσαμε `ifconfig` και αλλάξαμε την IP διεύθυνση του PC2. Όταν γίνεται αυτό, η προεπιλεγμένη πύλη διαγράφεται από τον πίνακα δρομολόγησης.

7.7

Στο PC2 ορίζουμε και πάλι ως προεπιλεγμένη πύλη τον δρομολογητή R2 με την εντολή:

```
route add default 172.17.17.193
```

7.8

Καθαρίζουμε τους πίνακες ARP στα PC2, PC3 και R2 με την εντολή `arp -d -a`.

7.9

Ξεκινάμε τη ζητούμενη καταγραφή στον R2 με την εντολή `tcpdump -i em2 "arp"`.

7.10

Ξεκινάμε τις ζητούμενες καταγραφές εκτελώντας την εντολή `tcpdump -n -i em0 "tcp"` στα PC2 και PC3.

7.11

Από το PC1 εκτελούμε `ssh lab@172.17.17.254`. Εμφανίζεται ένδειξη λάθους:

```
ssh_exchange_identification: read: Connection reset by peer
```

7.12

Σταματάμε τις καταγραφές και επαναλαμβάνουμε το `ssh lab@172.17.17.254`. Αυτή τη φορά η προσπάθεια είναι επιτυχής.

7.13

Με την εντολή `arp -a` στον R2 καταγράφουμε:

```
PC2: 172.17.17.254 at 08:00:27:10:0c:23 on em2
```

Δεν υπάρχει εγγραφή για το PC3.

7.14

Στην καταγραφή πακέτων arp, το PC3 απάντησε πρώτο στο ARP Request του R2, ενώ το PC2 απάντησε δεύτερο.

7.15

Η διεύθυνση MAC που περιέχει ο πίνακας ARP του R2 ανήκει στο PC2.

7.16

Την δεύτερη φορά συνδεθήκαμε στο PC2, επειδή το PC2 απάντησε δεύτερο και έτσι η πιο πρόσφατη εγγραφή στον πίνακα ARP έδειχνε ότι η κοινή IPv4 διεύθυνση αντιστοιχεί στην φυσική διεύθυνση του PC2.

7.17

Άλλοι τρόποι με τους οποίους μπορούμε να καταλάβουμε σε ποιο μηχάνημα έχουμε συνδεθεί, πέρα από τον τρόπο που χρησιμοποιήσαμε παραπάνω (δηλαδή μέσω της καταγραφής των πακέτων arp):

- Μπορούμε να εκτελέσουμε `ifconfig em0` και να δούμε τη διεύθυνση MAC που αναγράφεται στην έξοδο της εντολής.
- Μπορούμε να εκτελέσουμε την εντολή `w` στα PC2, PC3. Στο ένα από τα δύο θα υπάρχει μόνο ο χρήστης `root`, ενώ στο άλλο θα φαίνεται ότι έχει συνδεθεί και ένας χρήστης `lab`.
- Μππορούμε να εκτελέσουμε την εντολή `netstat | grep ssh` στα PC2, PC3. Στο ένα από τα δύο θα υπάρχει μια σύνδεση ssh που θα περιλαμβάνει τη διεύθυνση IPv4 του PC1 και τη διεύθυνση IPv4 του μηχανήματος που έχει γίνει η σύνδεση.

7.18

Στον R2 λήφθηκε πρώτα το ARP reply από το PC3 και ύστερα από το PC2. Στις δύο καταγραφές TCP τεμαχίων παρατηρούμε ότι τα τεμάχια έχουν τις εξής σημαίες:

- Καταγραφή PC3:

  - SYN: από το PC1 στο PC3, με σκοπό να ξεκινήσει η τριπλή χειραψία
  - SYN, ACK: η απάντηση του PC3 προς το PC1, το δεύτερο τεμάχιο της χειραψίας
  - SYN, ACK: επανάληψη του παραπάνω
  - SYN, ACK: επανάληψη του παραπάνω
  - SYN, ACK: επανάληψη του παραπάνω, τελικά εγκαταλείπεται η προσπάθεια, αφού δεν λαμβάνεται ποτέ το πακέτο με σημαία ACK (τρίτο τεμάχιο της χειραψίας)

- Καταγραφή PC2 (πλέον ο R2 έχει λάβει ARP reply από το PC2, και έτσι τα επόμενα TCP τεμάχια με προορισμό τη διεύθυνση 172.17.17.254 στέλνονται στο PC2, και γι' αυτό δημιουργείται η "σύγχυση" που προκαλεί την αποτυχία του πρώτου SSH):
  - ACK: από το PC1 στο PC2, είναι το τρίτο τεμάχιο της χειραψίας που δεν πάει ποτέ στο PC3, που έστειλε το δεύτερο τεμάχιο της χειραψίας
  - RST: από το PC2 στο PC1, αφού ο PC2 δεν έχει εγκατεστημένη κάποια σύνδεση με το PC1 και έτσι στέλνει τεμάχιο με σκοπό τον τερματισμό της παραπάνω επικοινωνίας.
  - PSH, ACK: από το PC1 στο PC3, δείχνει ότι τα δεδομένα πρέπει να προωθηθούν στην εφαρμογή
  - RST: από το PC2 στο PC1 με σκοπό τον τερματισμό της σύνδεσης
  - RST: από το PC1 στο PC2 με σκοπό τον τερματισμό της σύνδεσης
  - RST: από το PC1 στο PC2 με σκοπό τον τερματισμό της σύνδεσης
  - RST: από το PC1 στο PC2 με σκοπό τον τερματισμό της σύνδεσης

Αυτή η "σύγχυση" επιλύεται τη δεύτερη φορά που συνδεόμαστε με SSH, αφού πλέον υπάρχει η εγγραφή στον πίνακα ARP του R2 που αντιστοιχεί την IPv4 διεύθυνση 172.17.17.254 με την MAC διεύθυνση του PC2, και η σύνδεση με SSH μεταξύ PC1 και PC2 είναι επιτυχής.