ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΓΚΟΥΜΕ ΛΑΟΥΡΕΝΤΙΑΝ
ΑΜ: 031 18 014
ΕΞΑΜΗΝΟ: 8^{ο}
ΟΜΑΔΑ: 3

ΕΡΓΑΣΤΗΡΙΟ ΔΙΚΤΥΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 5: ΣΤΑΤΙΚΗ ΔΡΟΜΟΛΟΓΗΣΗ

Άσκηση 1: Δρομολόγηση σε ένα βήμα

1.1) Εκτελέσαμε τις εξής εντολές:

- PC1: “ifconfig em0 192.168.2.1/24”
- PC2: “ifconfig em0 192.168.2.2/24”
- R1: “ifconfig em0 192.168.1.1/24” και “ifconfig em1 192.168.2.1/24”

1.2) gateway_enable=“YES”

1.3) Εκτελούμε στο PC1 “route add -net 192.168.2.0/24 192.168.1.1”.

1.4) Εκτελούμε στο PC1 “netstat -rn”. Για το υποδίκτυο 192.168.2.0/24 βλέπουμε τις σημαίες UGS, οπότε σημαίνει πως η διαδρομή είναι ενεργή (U), ο προορισμός είναι πύλη, η οποία και θα αποφασίσει για την περαιτέρω προώθηση των πακέτων (G) και τέλος η διαδρομή ορίστηκε στατικά (S).

1.5) Ενώ στάλθηκε το ping, δε λαμβάνουμε απάντηση.

1.6) Στο LAN1 παρατηρούμε πως το PC1 στέλνει τα ICMP Echo requests του στη διεπαφή em0 του R1, ενώ στο LAN2 το R1 στέλνει μέσω της em1 τα αιτήματα στο PC2. Επομένως, ενώ ο PC2 λαμβάνει κανονικά τα requests του PC1, αδυνατεί να απαντήσει, καθώς δε ξέρει προς τα που πρέπει να προωθήσει τα replies.

1.7) Εκτελούμε στο PC2 την εντολή “route add -net 192.168.1.0/24 192.168.2.1”.

1.8) Ναι, πλέον επικοινωνούν κανονικά.

1.9) Ο πίνακας δρομολόγησης του R1, όπως φαίνεται έχει ήδη την απαραίτητη πληροφορία για δρομολόγηση στα LAN1 (192.168.1.0/24) και LAN2 (192.168.2.0/24), οπότε και δεν απαιτείται κάποια επιπλέον ρύθμιση.

```
root@R1:~ # netstat -r
Routing tables

Internet:
Destination        Gateway            Flags     Netif Expire
localhost          link#5             UH          lo0
192.168.1.0/24     link#1             U           em0
192.168.1.1        link#1             UHS         lo0
192.168.2.0/24     link#2             U           em1
192.168.2.1        link#2             UHS         lo0
```

Άσκηση 2: Proxy ARP

2.1) Εκτελούμε στο PC1 την εντολή “route del 192.168.2.0/24”.

2.2) Ξανά, στο PC1 “ifconfig em0 192.168.1.2/20”.

2.3) Το PC1 βρίσκεται στο υποδίκτυο 192.168.0.0. Εάν εφαρμόσουμε τη μάσκα του υποδικτύου του στις διευθύνσεις των PC2, PC3 βλέπουμε πως το PC1 τα αντιλαμβάνεται σα να ανήκουν στο ίδιο υποδίκτυο.

2.4) Από τα Ping αυτά λαμβάνουμε σφάλμα “ping: sendto: Host is down”.

2.5) Πλέον το ping είναι επιτυχές, καθώς ο δρομολογητής λειτουργεί ως proxy, επομένως απαντάει με τη δική του MAC στα ARP requests του PC1, δεδομένου ότι το PC2 βρίσκεται σε υποδίκτυο στο οποίο ο R1 ξέρει πώς να δρομολογήσει πακέτα για εκεί.

2.6) Αποτυγχάνει, καθώς για τον PC3, το PC1 φαίνεται πως είναι σε άλλο υποδίκτυο, οπότε δεδομένου ότι δεν έχει προκαθορισμένη πύλη ή κάποια πύλη για το υποδίκτυο του PC1, απλά απορρίπτει τα πακέτα που λαμβάνει χωρίς να απαντά.

2.7) Εκτελούμε στο PC3 την εντολή “route add -net 192.168.1.0/24 192.168.2.1”.

2.8) Εκτελούμε “arp -ad” σε όλα τα μηχανήματα.

2.9) Στο R1 εκτελούμε σε μία κονσόλα “tcpdump -ei em0” και σε μία δεύτερη “tcpdump -ei em1”.

2.10) Ξανακάνουμε ping από το PC1 στο PC3 και βλέπουμε πως το το R1 απαντάει με τη MAC του em0 του, παρόλο που το Ping έχει ως προορισμό το PC3.

2.11) Προς τη MAC 08:00:27:13:a7:15 (em0 του R1).

2.12) Από τη MAC 08:00:27:5a:86:2d (em1 του R1).

2.13) Παρατηρήσαμε τα παρακάτω πακέτα:

- Ο PC1 κάνει broadcast ARP request για να μάθει την MAC address της διεύθυνσης 192.168.2.3
- Το R1 απαντάει δίνοντάς του τη MAC της διεπαφής em0 ως MAC της 192.168.2.3, αφού το έχουμε δηλώσει ως proxy
- Το PC1 στέλνει στο em0 του R1 το ICMP echo request
- Το PC1, μέσω της em1 κάνει broadcast ένα ARP request με σκοπό να μάθει την MAC της 192.168.2.3
- Το PC3 απαντάει στο παραπάνω broadcast με τη MAC διεύθυνσή του

- Το R1 προωθεί στο PC3 το ICMP echo request μέσω της em1
- Το PC3 απαντάει στην em1 του R1 με ICMP echo reply, με τελικό αποδέκτη τη διεύθυνση 192.168.1.2
- Το R1 κάνει broadcast ένα ARP Request μέσω της em0, ώστε να μάθει την MAC της 192.168.1.2, μιας και δε φαίνεται να την αποθήκευσε από το προηγούμενο broadcast του PC1
- Το PC1 απαντάει με τη MAC διεύθυνσή του στο R1
- Το R1 προωθεί το ICMP echo reply στο PC1

2.14) Το ping θα επιτυγχάνει όσο το PC1 νομίζει πως το PC3 είναι στο ίδιο υποδίκτυο με αυτό. Επομένως, το μέγιστο μήκος προθέματος είναι 22, καθώς αν βάλουμε 23, το PC1 αντιλαμβάνεται πως το PC3 ανήκει στο 192.168.2.0/23, ενώ το ίδιο το PC1 ανήκει στο 192.168.0.0/23, άρα από τα 23 bits και μετά απαιτείται δρομολόγηση, για την οποία δεν έχουμε ορίσει κάποια πύλη στο PC1, οπότε και το Ping θα αποτυγχάνει. *(Σημείωση: Μέχρι τα 22 bits, το PC1 αντιλαμβάνεται αμφότερα μηχανήματα στο υποδίκτυο 192.168.0.0/22)*

2.15) Εκτελούμε στο PC1 την εντολή “ifconfig em0 192.168.1.2/23”.

2.16) Εκτελούμε στο PC1 την εντολή “route add -net 192.168.2.0/24 -interface em0”.

2.17) Με “netstat -r” βλέπουμε πως ως πύλη για το δίκτυο 192.168.2.0/24 εμφανίζεται η MAC address της διεπαφής em0.

2.18) Πλέον το Ping επιτυγχάνει, καθώς το ταίριασμα μεγαλύτερου προθέματος γίνεται με το υποδίκτυο 192.168.2.0/24, οπότε και το em0 κάνει τα κατάλληλα ARP requests, ώστε να στείλει τα ICMP πακέτα και λαμβάνει απαντήσεις από το proxy ARP, δηλαδή το R1, το οποίο απαντάει σα να ήταν το PC3.

2.19) Στο PC3 εκτελούμε “sysctl net.link.ether.inet.proxyall=0”.

2.20) Εκτελούμε “route change -net 192.168.2.0/24 192.168.1.1”.

2.21) Εκτελούμε στο PC1 “ifconfig em0 192.168.1.2/24”.

2.22) Η διαδρομή διαγράφηκε. Την επανορίζουμε με την εντολή “route add -net 192.168.2.0/24 192.168.1.1”.

Άσκηση 3: Δρομολόγηση σε περισσότερα βήματα

3.1) Εκτελούμε στο R1 “ifconfig em0 192.168.1.2/24” και “ifconfig em1 172.17.17.1/30”.

3.2) Εκτελούμε στο R2 “ifconfig em0 172.17.17.2/30” και “ifconfig em1 192.168.2.1/24”.

3.3) Λαμβάνουμε την εξής απάντηση με το λάθος “Destination Host Unreachable”.

```
root@PC1:~ # ping -c 1 192.168.2.2
PING 192.168.2.2 (192.168.2.2): 56 data bytes
36 bytes from 192.168.1.1: Destination Host Unreachable
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 0054 7c55   0 0000  40  01 79ff 192.168.1.2  192.168.2.2


--- 192.168.2.2 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

3.4) Στο LAN1 παρήχθησαν τα ICMP μηνύματα που φαίνονται παρακάτω: (ICMP echo request και ICMP host 192.168.2.2 unreachable)

```
root@R1:~ # tcpdump -i em0
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em0, link-type EN10MB (Ethernet), capture size 262144 bytes
22:39:10.825631 ARP, Request who-has 192.168.1.1 tell 192.168.1.2, length 46
22:39:10.825907 ARP, Reply 192.168.1.1 is-at 08:00:27:13:a7:15 (oui Unknown), le
ngth 28
22:39:10.826722 IP 192.168.1.2 > 192.168.2.2: ICMP echo request, id 24580, seq 0
, length 64
22:39:10.826814 IP 192.168.1.1 > 192.168.1.2: ICMP host 192.168.2.2 unreachable,
 length 36
```

Παρατηρούμε πως στο WAN1 δε καταγράφεται κανένα πακέτο. Ο λόγος είναι πως στον πίνακα δρομολόγησης του R1 δεν υπάρχει default gateway, αλλά ούτε και εγγραφή προς το υποδίκτυο 192.168.2.0/24., με αποτέλεσμα να επιστρέφεται το μήνυμα λάθους που είδαμε.

3.5) Λαμβάνουμε τα παρακάτω κάνοντας “traceroute 192.168.2.2”:

```
root@PC1:~ # traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  2.315 ms  1.041 ms  0.905 ms
 2  192.168.1.1 (192.168.1.1)  0.971 ms !H  1.053 ms !H  0.897 ms !H
```

Εκτελώντας “man traceroute” βλέπουμε πως το “!H” αναφέρεται στο Host Unreachable.

3.6) Εκτελούμε στο R1 την εντολή “route add -net 192.168.2.0/24 172.17.17.2”.

3.7) Πλέον, το ping του PC1 στο PC2 επιτυγχάνει ως Request, ωστόσο δε λαμβάνουμε πίσω το Reply, καθώς όταν στέλνει το Reply το PC2, αυτό αδυνατεί να προωθηθεί από το R2, επομένως στέλνεται ένα “ICMP host 192.168.1.2 unreachable” από το R2 στο PC2.

3.8) Όπως είδαμε, παρατηρούμε τα παρακάτω:

- IP 192.168.1.2 > 192.168.2.2: ICMP echo request, το οποίο είναι το πακέτο του PC1 που το R2 προωθεί στο PC2
- IP 192.168.2.2 > 192.168.1.2: ICMP echo reply, το οποίο είναι το πακέτο που το PC2 στέλνει στο R2 με τελικό προορισμό το PC1
- IP 192.168.2.1 > 192.168.2.2: ICMP host 192.168.1.2 unreachable, το οποίο είναι η απάντηση που το R1 στέλνει στο PC2, ενημερώνοντας το πως δε μπορεί να προωθήσει το προηγούμενο reply

3.9) Στο WAN1 δε παρατηρούμε πακέτα ICMP, παρά μόνο UDP, με αποστολέα το 192.168.1.2.33968 και παραλήπτη τη διεύθυνση 192.168.2.2 με διαφορετική κάθε φορά θύρα προορισμού προκειμένου ο κόμβος-παραλήπτης να μην επεξεργαστεί τα UDP packets. Καταγράφουμε τα εν λόγω πακέτα, καθώς αυτά αποστέλλονται μέσω του tracer με ένα μικρό TTL μέχρι να ληφθεί απάντηση ICMP time exceeded.

3.10) Στο LAN2 βλέπουμε να προωθούνται τα ανωτέρω UDP πακέτα από το R2 στο PC2, ενώ επιπλέον βλέπουμε ως απάντηση από το PC2 (192.168.2.2) στο R2 (192.168.1.2) μηνύματα ICMP 192.168.2.2 udp port XXXXX unreachable, όπου XXXXX η εκάστοτε θύρα προορισμού.

```
19:30:28.175499 IP 192.168.1.2.33968 > 192.168.2.2.33525: UDP, length 12
19:30:28.176061 IP 192.168.2.2 > 192.168.1.2: ICMP 192.168.2.2 udp port 33525 un
reachable, length 36
19:30:33.192702 IP 192.168.1.2.33968 > 192.168.2.2.33526: UDP, length 12
19:30:33.192916 IP 192.168.2.2 > 192.168.1.2: ICMP 192.168.2.2 udp port 33526 un
reachable, length 36
```

3.11) Είδαμε πως το PC2 αποκρίνεται στο R2 λέγοντας πως ήταν unreachable το port του μηνύματος που έλαβε. Δε παράγονται ICMP destination unreachable μηνύματα ως απόκριση στα ICMP που παράγει το PC2, καθώς σε αυτή την περίπτωση θα προκαλούνταν loop στο σύστημα.

3.12) Εκτελούμε στο PC2 “route add -net 192.168.1.0/24 172.17.17.1”.

3.13) Πλέον μπορούμε να κάνουμε κανονικά traceroute. Στο WAN2 παράγονται μηνύματα τύπου ICMP time exceeded in-transit (172.17.17.2 > 192.168.1.2), ενώ επιπλέον καταγράφονται μηνύματα ICMP 192.168.2.2 udp port 33443 unreachable (192.168.2.2 > 192.168.1.2). Τα ICMP time exceeded μηνύματα που παρήχθησαν οφείλονται στο γεγονός ότι στη διεπαφή 172.17.17.2 του R2 μηδενίστηκε το TTL (TTL = 2) της δεύτερης τριάδας πακέτων που απεστάλησαν από το traceroute του PC1.

```
root@PC1:~ # traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  1.936 ms  1.385 ms  1.154 ms
 2  172.17.17.2 (172.17.17.2)  3.004 ms  1.929 ms  2.136 ms
 3  192.168.2.2 (192.168.2.2)  3.921 ms  3.431 ms  3.174 ms
```

3.14) Λαμβάνουμε ως απάντηση “no route to host”, πράγμα που οφείλεται στο γεγονός ότι ο πίνακας δρομολόγησης του PC2 δε περιλαμβάνει εγγραφές ούτε για το υποδίκτυο της διεύθυνσης 172.17.17.1, αλλά ούτε και έχει default gateway, ώστε το πακέτο να δρομολογηθεί από εκεί.

```
root@PC2:~ # ping -c 1 172.17.17.1
PING 172.17.17.1 (172.17.17.1): 56 data bytes
ping: sendto: No route to host

--- 172.17.17.1 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

3.15) Στο PC2 εκτελούμε την εντολή “route del 192.168.1.0/24”.

```
root@PC2:~ # netstat -r
Routing tables

Internet:
Destination        Gateway            Flags     Netif Expire
localhost          link#2             UH          lo0
192.168.1.0/24     192.168.2.1        UGS         em0
192.168.2.0/24     link#1             U           em0
192.168.2.2        link#1             UHS         lo0
```

3.16) Στο PC2 εκτελούμε την εντολή “route add default 192.168.2.1”.

3.17) Πλέον το ping επιτυγχάνει κανονικά.

3.18) Όπως είπαμε, στο πρώτο ping το PC2 αδυνατούσε να στείλει το πακέτο του, καθώς δεν είχε κάποια εγγραφή για το πού έπρεπε να το στείλει. Αφού ορίσαμε προκαθορισμένη πύλη, πλέον το πακέτο που στέλνει το PC2 δρομολογείται σε αυτή, και από εκεί, εφόσον υπάρχει εγγραφή για το υποδίκτυο του τελικού προορισμού πηγαίνει σε αυτόν.

Άσκηση 4: Ένα πιο πολύπλοκο δίκτυο με εναλλακτικές διαδρομές

4.1) Εκτελούμε στο PC3 “ifconfig em0 192.168.2.3/24”.

4.2) Εκτελούμε στο PC3 “route add -net 192.168.1.0/24 192.168.2.1”.

4.3) Οι κάρτες em0, em1 και em2 του R1 θα πρέπει να βρίσκονται στα εσωτερικά δίκτυα LAN1, WAN1 και WAN2 αντίστοιχα. Ορίζουμε τις διευθύνσεις με τις εντολές “ifconfig em0 192.168.1.1/24”, “ifconfig em1 172.17.17.1/30” και “ifconfig em2 172.17.17.5/30”.

4.4) Οι κάρτες em0, em1 και em2 του R2 θα πρέπει να βρίσκονται στα εσωτερικά δίκτυα WAN1, LAN2 και WAN3 αντίστοιχα. Ορίζουμε τις διευθύνσεις με τις εντολές “ifconfig 172.17.17.2/30”, “ifconfig em1 192.168.2.1/24” και “ifconfig em2 172.17.17.9/30”.

4.5) Οι κάρτες em0 και em1 του R3 θα πρέπει να βρίσκονται στα εσωτερικά δίκτυα WAN2 και WAN3 αντίστοιχα. Ορίζουμε τις διευθύνσεις με τις εντολές “ifconfig em0 172.17.17.6/30” και “ifconfig em1 172.17.17.10/30”.

4.6) Εκτελούμε στο R1 “route add -net 192.168.2.0/24 172.17.17.2”.

4.7) Εκτελούμε στο R2 “route add -net 192.168.1.0/24 172.17.17.1”.

4.8) Εκτελούμε στο R3 “route add -net 192.168.1.0/24 172.17.17.5” και “route add -net 192.168.2.0/24 172.17.17.9”.

4.9) Εκτελούμε στο R1 “route add -host 192.168.2.3 172.17.17.6” και βλέπουμε στα αποτελέσματα της εντολής “netstat -r” τις σημαίες UGHS για τη συγκεκριμένη εγγραφή, εκ των οποίων η H δηλώνει πως αναφέρεται σε συγκεκριμένο host.

4.10) Βλέπουμε συνολικά 3 βήματα, το 3^{ο} εκ των οποίων γίνεται στο destination (PC2).

```
root@PC1:~ # traceroute 192.168.2.2
traceroute to 192.168.2.2 (192.168.2.2), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  14.287 ms  1.370 ms  1.058 ms
 2  172.17.17.2 (172.17.17.2)  2.846 ms  1.926 ms  2.279 ms
 3  192.168.2.2 (192.168.2.2)  4.271 ms  3.447 ms  2.773 ms
```

4.11) Βλέπουμε πως το TTL που λάβαμε ως απάντηση έχει τιμή 62, επομένως μεσολάβησαν 2 ενδιάμεσοι κόμβοι, όπως και πριν.

4.12) Αντίστοιχα, βλέπουμε 4 συνολικά βήματα, εκ των οποίων το τελευταία αφορά ξανά το destination.

```
root@PC1:~ # traceroute 192.168.2.3
traceroute to 192.168.2.3 (192.168.2.3), 64 hops max, 40 byte packets
 1  192.168.1.1 (192.168.1.1)  1.067 ms  1.161 ms  0.980 ms
 2  172.17.17.6 (172.17.17.6)  3.077 ms  2.116 ms  2.188 ms
 3  172.17.17.2 (172.17.17.2)  3.908 ms  2.984 ms  2.507 ms
 4  192.168.2.3 (192.168.2.3)  4.400 ms  3.900 ms  42.569 ms
```

4.13) Λαμβάνουμε απάντηση με TTL = 62, επομένως, βλέπουμε 2 ενδιάμεσα βήματα.

4.14) Το ICMP echo request ακολουθεί τη διαδρομή PC1 → R1 → R3 → R2 → PC3.

4.15) Αντιθέτως, το ICMP echo reply ακολουθεί τη διαδρομή PC3 → R2 → R1 → PC1, δεδομένου πως είχαμε ορίσει στατική εγγραφή στον R2 ώστε να προωθούνται πακέτα προς το LAN1 μέσω του R1, ενώ είχαμε ορίσει επίσης στατική εγγραφή στο R1, έτσι ώστε πακέτα προς το PC3 να διέρχονται από το R3.

4.16) Εκτελούμε στο R2 “tcpdump -i em1”.

4.17) Όχι, δε καταγράφουμε τίποτα, επομένως ούτε έφτασε κάτι, ούτε παρήχθησαν UDP στο PC2.

4.18) Καταγράφουμε τα παρακάτω, επομένως φτάνουν μεν τα πακέτα από το PC1 στο PC3 και στη συνέχεια παράγονται πακέτα “ICMP 192.168.2.3 udp port XXXXX unreachable”.

```
root@PC:~ # tcpdump -i em1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on em1, link-type EN10MB (Ethernet), capture size 262144 bytes
23:55:59.853346 IP 192.168.1.2.34631 > 192.168.2.3.33444: UDP, length 12
23:55:59.853884 IP 192.168.2.3 > 192.168.1.2: ICMP 192.168.2.3 udp port 33444 un
reachable, length 36
23:56:04.870734 IP 192.168.1.2.34631 > 192.168.2.3.33445: UDP, length 12
23:56:04.871401 IP 192.168.2.3 > 192.168.1.2: ICMP 192.168.2.3 udp port 33445 un
reachable, length 36
23:56:09.882878 IP 192.168.1.2.34631 > 192.168.2.3.33446: UDP, length 12
23:56:09.883432 IP 192.168.2.3 > 192.168.1.2: ICMP 192.168.2.3 udp port 33446 un
reachable, length 36
23:56:14.898899 IP 192.168.1.2.34631 > 192.168.2.3.33447: UDP, length 12
23:56:14.899422 IP 192.168.2.3 > 192.168.1.2: ICMP 192.168.2.3 udp port 33447 un
reachable, length 36
```

4.19) Εκτελούμε “route change -net 192.168.2.0/24 172.17.17.6” και “route change -net 192.168.1.0/24 172.17.17.10” στα R1 και R2 αντίστοιχα. Κάνοντας από το PC1 “traceroute 192.168.2.2” και “traceroute 192.168.2.3” βλέπουμε πως επικοινωνούν κανονικά.

4.20) Εκτελούμε στο R1 “route show 192.168.2.2” και “route show 192.168.2.3” οπότε και βλέπουμε τα παρακάτω:

```
root@R1:~ # route show 192.168.2.2
   route to: 192.168.2.2
destination: 192.168.2.0
       mask: 255.255.255.0
    gateway: 172.17.17.6
        fib: 0
  interface: em2
      flags: <UP,GATEWAY,DONE,STATIC>
 recvpipe  sendpipe  ssthresh  rtt,msec    mtu        weight    expire
       0         0         0         0      1500         1         0
```

```
root@R1:~ # route show 192.168.2.3
   route to: 192.168.2.3
destination: 192.168.2.3
    gateway: 172.17.17.6
        fib: 0
  interface: em2
      flags: <UP,GATEWAY,HOST,DONE,STATIC>
 recvpipe  sendpipe  ssthresh  rtt,msec    mtu        weight    expire
       0         0         0         0      1500         1         0
```

Η διαφορά που παρατηρούμε είναι πως, για το PC2 το destination είναι το subnet 192.168.2.0/24, ενώ για το PC3 το destination είναι η ίδια η IP του (192.168.2.3), μιας και είχαμε ορίσει προηγουμένως στατική εγγραφή προς αυτό από το R1 μέσω του R3.

4.21) Μεταξύ των 2 τελευταίων εγγραφών, επιλέγεται η τελευταία, καθώς έχουμε ταίριασμα μήκους 32 bits.

```
root@R1:~ # netstat -r
Routing tables

Internet:
Destination        Gateway            Flags     Netif Expire
localhost          link#5             UH          lo0
172.17.17.0/30     link#2             U           em1
172.17.17.1        link#2             UHS         lo0
172.17.17.4/30     link#3             U           em2
172.17.17.5        link#3             UHS         lo0
192.168.1.0/24     link#1             U           em0
192.168.1.1        link#1             UHS         lo0
192.168.2.0/24     172.17.17.6        UGS         em2
192.168.2.3        172.17.17.6        UGHS        em2
```

Άσκηση 5: Βρόχοι κατά τη δρομολόγηση

5.1) Εκτελούμε στο R3 “route change -net 192.168.1.0/24 172.17.17.9”.

5.2) Εκτελούμε στο PC1 “ping -c 1 192.168.2.2”, από όπου και λαμβάνουμε σφάλμα “Time to live exceeded”.

```
root@PC1:~ # ping -c 1 192.168.2.2
PING 192.168.2.2 (192.168.2.2): 56 data bytes
36 bytes from 172.17.17.6: Time to live exceeded
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 0054 7dd6   0 0000  01  01 b77e 192.168.1.2  192.168.2.2


--- 192.168.2.2 ping statistics ---
1 packets transmitted, 0 packets received, 100.0% packet loss
```

5.3) Από τη διεπαφή em0 του R3, με διεύθυνση 172.17.17.6. (Αν, ωστόσο, βάζαμε ως TTL μια περιττή τιμή λ.χ. ping -m 65, τότε θα παίρναμε απάντηση από τη διεπαφή 192.168.1.1).

5.4) Στο δίκτυο WAN2 είτε μέσω της em0 του R3 είτε μέσω της em2 του R1.

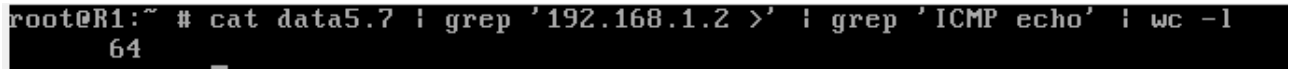
5.5) Το “tcpdump ‘icmp[icmptype] == icmp-echo’ ”.

5.6) Εκτελώντας στο PC1 “tcpdump ‘icmp[icmptype]==icmp-echo’ ”, στο R1 “tcpdump -ei em2 ‘icmp[icmptype]==icmp-echo’ ” και στη συνέχεια “ping -c 1 192.168.2.2” από το PC1, βλέπουμε πως από το PC1 παρήχθησε μόνο 1 ICMP echo request, ενώ στο WAN2 καταγράφηκαν 63 (packets captured). (64 packets received by filter).

5.7) Εκτελούμε στο R1 “tcpdump -i em0 > data5.7”, ενώ στο R3 “tcpdump -i em2 > data5.7”.

5.8) Εμφανίζονται 64 βήματα, ενώ η διαδρομή που καταγράφεται είναι: PC1 → R1 (192.168.1.1) → R3 (172.17.17.6) → R1 (192.168.1.1) → R3 (172.17.17.6) → R1 (192.168.1.1) → R3 (172.17.17.6) … → R1 (192.168.1.1) → R3 (172.17.17.6).

5.9) Εκτελούμε στην καταγραφή του R1 (για το LAN1) “cat data5.7 | grep ‘192.168.1.2 >’ | grep ‘ICMP echo’ | wc -l” οπότε και παίρνουμε αποτέλεσμα 64.



5.10) Εκτελούμε στο R3 “cat data5.7 | grep ‘ICMP echo request’ | wc -l” και λαμβάνουμε ως αποτέλεσμα τον αριθμό 2016. Στο πρώτο ICMP echo request του PC1, αναμένουμε πως δε καταγράφηκε τίποτα στο WAN2, καθώς το TTL ήταν 1, οπότε και απάντησε το R1 αμέσως. Για τα υπόλοιπα ICMP echo requests του PC1:

- 2^{O} ICMP echo request: Καταγράφεται 1 στο WAN2
- 3^{O} ICMP echo request: Καταγράφονται 2 στο WAN2
- 4^{O} ICMP echo request: Καταγράφονται 3 στο WAN2
- ν-οστό ICMP echo request: Καταγράφονται ν-1 στο WAN2

Συνεπώς, εφόσον έχουμε συνολικά 64 requests, ψάχνουμε το άθροισμα: 0 + 1 + 2 + 3 + … + 63 = $\frac{63*64}{2}$ = 2016, όσα και τα πακέτα που καταγράψαμε.

5.11) Εκτελώντας στο R3 “cat data5.7 | grep ‘ICMP time exceeded’ | wc -l ” παίρνουμε ως αποτέλεσμα 32. Εξετάζοντας το αποτέλεσμα, σκεφτόμαστε αντίστοιχα με πριν. Το πρώτο request θα λάβει ως απάντηση από ICMP time exceeded από το R1, δεδομένου ότι έχει TTL = 1, επομένως και δε θα καταγραφεί στο WAN2. Το δεύτερο ωστόσο, με TLL = 2, θα λάβει τέτοια απάντηση από το R3, επομένως θα καταγραφεί στο WAN2. Συνεχίζοντας αυτή τη συλλογιστική, βλέπουμε πως θα καταγραφούν μόνο ICMP time exceeded από πακέτα που είχαν ζυγό TTL έως και 64, άρα για TTL = 2, 4, 6, 8, … 64, συνεπώς 32 πακέτα.

5.12) Εφαρμόζοντας κατάλληλο φίλτρο (“tcpdump ‘icmp[icmptype]==8’ ” για ICMP Echo ή “tcpdump ‘icmp[icmptype]==11’ ” για ICMP time exceeded) και εκτελώντας στο τέλος Ctrl+C βλέπουμε πόσα πακέτα καταγράψαμε στο τερματικό μας.

5.13) Τα πακέτα που στέλνουμε με Ping έχουν προκαθορισμένο αριθμό TTL (64 by default στο σύστημά μας), ενώ αυτά που στέλνουμε με traceroute έχουν διαρκώς αυξανόμενο TTL που ξεκινάει από το 1 μέχρι την απαιτούμενη τιμή για να φτάσει στον προορισμό του.

5.14) Λόγω του μηχανισμού του TTL, ο οποίος βοηθάει να αποφευχθεί τέτοιου είδους συμφόρηση από πακέτα που έχουν κολλήσει σε loops, καθώς όταν μηδενίζεται μετά από κάθε hop, το πακέτο χάνεται.

Άσκηση 6: Χωρισμός σε υποδίκτυα

6.1) Από το μπλοκ 172.17.17.0/24 μπορούμε να φτιάξουμε 2 υποδίκτυα που να χωράνε 126 hosts, τα 172.17.17.0/25 και 172.17.17.128/25. Ωστόσο, παρατηρούμε πως το υποδίκτυο 172.17.17.128/30 είναι δεσμευμένο από τα WANs. Επομένως, αναθέτουμε στο **LAN1 το 172.17.17.0/25** προκειμένου να αποφύγουμε ενδεχόμενες μελλοντικές συγκρούσεις στα ταιριάσματα μήκους. Μας απομένει το 172.17.17.128/25.

6.2) Από το προηγούμενο μπλοκ μας απομένει το 172.17.17.128/25, το οποίο διαθέτει 2 υποδίκτυα χωρητικότητας 62 κόμβων έκαστο, το 172.17.17.192/26 και το 172.17.17.128/26. Για τους λόγους που αναφέραμε και προηγουμένως, αναθέτουμε στο **LAN2 το 172.17.17.192/26**.

6.3) Αντίστοιχα, το 172.17.17.128/26 που μας έμεινε μπορεί να σπάσει σε 172.17.17.128/27 και 172.17.17.160/27 με 30 υπολογιστές έκαστο. Αναθέτουμε το **172.17.17.160/27 στο LAN3**

6.4) Εκτελούμε “ifconfig em0 **172.17.17.1/25**” στο **PC1** και “ifconfig em0 **172.17.17.126/25**” στο **R1**.

6.5) Εκτελούμε “ifconfig em0 **172.17.17.161/27**” στο **PC4** και “ifconfig em0 **172.17.17.190/27**” στο **R3**.

6.6) Εκτελούμε “ifconfig em1 **172.17.17.193/26**” στο **R2**, “ifconfig em0 **172.17.17.253/26**” στο **PC2** και “ifconfig em0 **172.17.17.254/26**” στο **PC3**.

6.7) Εκτελούμε:

- 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) Εκτελούμε στο R1 “route add -net 172.17.17.192/26 172.17.17.130” και “route add -net 172.17.17.160/27 172.17.17.130”.

6.9) Εκτελούμε στο R2 “route add -net 172.17.17.0/25 172.17.17.137” και “route add -net 172.17.17.160/27 172.17.17.137”.
6.10) Εκτελούμε στο R3 “route add -net 172.17.17.0/25 172.17.17.133” και “route add -net 172.17.17.192/26 172.17.17.133”.

6.11) Εκτελούμε τα ζητούμενα ping και όλα επιτυγχάνουν.

Άσκηση 7: Ταυτόσημες διευθύνσεις IP

7.1) Κατασκευάζουμε τον παρακάτω πίνακα με όλες τις πληροφορίες για τις em0 κάθε PC.

Υπολογιστής	IP	MAC
PC1	172.17.17.1	08:00:27:a7:4d:d1
PC2	172.17.17.253	08:00:27:d5:2f:28
PC3	172.17.17.254	08:00:27:20:c6:95
PC4	172.17.7.161	08:00:27:e2:93:3a

7.2) Εκτελούμε στο PC2 “ifconfig em0 172.17.17.254”.

7.3) Λαμβάνουμε το παρακάτω μήνυμα σφάλματος:

```
root@PC2:~ # Apr  5 23:45:29 PC2 kernel: arp: 08:00:27:20:c6:95 is using my IP a
ddress 172.17.17.254 on em0!
```

7.4) Ναι, αντίστοιχο μήνυμα περί χρήσης της IP του από το PC2:

```
root@PC3:~ # Apr  5 23:49:26 PC3 kernel: arp: 08:00:27:d5:2f:28 is using my IP a
ddress 172.17.17.254 on em0!
```

7.5) Ναι, ορίστηκε κανονικά. Τα μηνύματα λάθους εμφανίζονται προκειμένου να προβούμε σε αλλαγές, ώστε να λύσουμε το ζήτημα των ίδιων IP σε ένα υποδίκτυο, ώστε να αποφύγουμε μελλοντικά προβλήματα με αποστολές από/σε αυτή τη διεύθυνση.

7.6) Όχι, καθώς αλλάξαμε την IP του, επομένως και διαγράφηκε το default gateway.

7.7) Εκτελούμε στο PC2 “route add default 172.17.17.193”.

7.8) Εκτελούμε στα PC2, PC3 και R3 “arp -ad”.

7.9) Εκτελούμε στο R2 “tcpdump -i em1 arp”.

7.10) Εκτελούμε στα PC2 και PC3 “tcpdump -n tcp”.

7.11) Εμφανίζονται γενικά μηνύματα “ssh: connect to host 172.17.17.254 port 22: Operation timed out”, ωστόσο παρατηρήθηκε και το “ssh_exchange_identification: read: Operation timed out”.

```
root@PC1:~ # ssh lab@172.17.17.254
ssh_exchange_identification: read: Operation timed out
root@PC1:~ # ssh lab@172.17.17.254
ssh: connect to host 172.17.17.254 port 22: Operation timed out
```

Κάνοντας επανεκκίνηση τα PC1 και PC3, λαμβάνουμε το παρακάτω.

```
root@PC1:~ # ssh lab@172.17.17.254
The authenticity of host '172.17.17.254 (172.17.17.254)' can't be established.
ECDSA key fingerprint is SHA256:E0pEsoVLpSwVQ5nRap7h3+XSu2Y1swJvkpFXxxikHwU.
No matching host key fingerprint found in DNS.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.17.17.254' (ECDSA) to the list of known hosts.
Password for lab@PC3:
Password for lab@PC3:
Password for lab@PC3:
Permission denied (publickey,keyboard-interactive).
```

7.12) Πλέον μας παραπέμπει κατευθείαν στο prompt.

```
root@PC1:~ # ssh lab@172.17.17.254
ssh: connect to host 172.17.17.254 port 22: Connection refused
root@PC1:~ # ssh lab@172.17.17.254
Password for lab@PC3:
```

7.13) Δεν υπάρχει διαθέσιμη εγγραφή για το PC2, καταγράφεται μόνο το PC3 στη διεύθυνση 172.17.17.254.

7.14) Παρατηρούμε πως απάντησε πρώτα το PC2.

```
23:59:51.276356 ARP, Request who-has 172.17.17.254 tell 172.17.17.193, length 28
23:59:51.276514 ARP, Reply 172.17.17.254 is-at 08:00:27:d5:2f:28 (oui Unknown),
length 46
23:59:51.276590 ARP, Reply 172.17.17.254 is-at 08:00:27:20:c6:95 (oui Unknown),
length 46
```

7.15) Ανήκει στο PC3, όπως είπαμε.

7.16) Στο PC3.

7.17) Είτε μέσω του πίνακα ARP του R2, είτε μέσω του prompt που εμφανίζεται όταν κάνουμε ssh σύνδεση (**lab@PC3**), είτε με nestat -a, το οποίο και θα εμφανίσει στη συσκευή που κάνουμε ssh μια νέα σύνδεση ssh.

7.18) Την πρώτη φορά που πήγε να γίνει η σύνδεση, το R2 έκανε broadcast ARP request για να μάθει τη MAC της 172.17.17.254. Όπως είδαμε, έλαβε πρώτα απάντηση από το PC2, οπότε και η σύνδεση ξεκίνησε με εκείνο πρώτα, ωστόσο μετά το πρώτο SYN, το PC2 απάντησε με το flag reset, καθώς κατάλαβε πως κάτι δε πάει καλά στη σύνδεση, πράγμα που προκλήθηκε από το γεγονός πως αμέσως μετά την απάντηση του PC2 στο R2, το PC3 απάντησε με τη δικιά του MAC, η οποία και καταχωρήθηκε ως τελευταία στο R2 με αποτέλεσμα να προκαλέσει conflict και να διακοπεί η τριπλή χειραψία. Επομένως, όταν ξανακάναμε ssh, το R2 είχε καταχωρημένη την MAC του PC3 ως αυτή που αντιστοιχεί στην 172.17.17.254, οπότε και δεν είχαμε conflicts.