

2η Σειρά Ασκήσεων HY-335α

Network layer

Παράδοση Παρασκευή 27/11/2015 23:55

Ευριπίδης Τζαμούσης (tzamasis@csd.uoc.gr)

Μαρία Πλακιά (plakia@csd.uoc.gr)

Ερώτηση 1 (5 μονάδες)

Ποια είναι η διαφορά μεταξύ των λειτουργιών της προώθησης και της δρομολόγησης του επιπέδου δικτύου;

Κατά την προώθηση, όταν ένα πακέτο φθάνει στην εισερχόμενη ζεύξη του δρομολογητή μεταφέρεται στην κατάλληλη εξερχόμενη ζεύξη, ενώ κατά την δρομολόγηση καθορίζεται η διαδρομή που θα ακολουθήσει ένα πακέτο για να φτάσει στον προορισμό του.

Ερώτηση 2 (5 μονάδες)

Συγκρίνετε τους αλγορίθμους δρομολόγησης κατάστασης ζεύξης (link-state) και απόστασης διανύσματος (distance-vector).

Οι αλγόριθμοι κατάστασης ζεύξης υπολογίζουν το μονοπάτι ελάχιστου κόστους μεταξύ πηγής και προορισμού. Απαραίτητη προϋπόθεση είναι κάθε δρομολογητής να έχει πλήρη γνώση του δικτύου, οπότε κάθε ένας δρομολογητής γνωρίζει το μονοπάτι ελάχιστου κόστους για να προσεγγίσει όλους τους υπόλοιπους δρομολογητές του δικτύου. Πληροφορίες σχετικές με την κατάσταση των ζεύξεων πλημυρίζουν το δίκτυο προκειμένου όλοι οι δρομολογητές να είναι ενήμερη για το δίκτυο.

Οι αλγόριθμοι απόστασης διανύσματος υπολογίζουν το μονοπάτι ελάχιστου κόστους μεταξύ πηγής και προορισμού με ένα επαναληπτικό και κατανεμημένο τρόπο. Πληροφορίες δρομολόγησης ανταλλάσσονται μόνο μεταξύ δύο γειτονικών κόμβων. Οπότε κάθε δρομολογητής έχει γνώση μόνο για τους δρομολογητές με τους οποίους συνδέεται άμεσα και δεν έχει καμία πληροφορία για τους υπόλοιπους δρομολογητές του δικτύου. Όταν συμβεί μία αλλαγή στο δίκτυο τότε ο κατάλληλος δρομολογητής θα πρέπει να ενημερώσει τους γειτονές του. Οπότε η υπολογιστική πολυπλοκότητα αυτών των αλγορίθμων είναι μικρότερη σε σύγκριση με τους αλγορίθμους του το ίδιο και ο όγκος μηνυμάτων που ανταλλάσσονται.

Ερώτηση 3 (5 μονάδες)

Εξηγήστε τι προβλήματα δρομολόγησης προκύπτουν λόγω του μεγάλου πλήθους συσκευών στο Διαδίκτυο που επικοινωνούν μεταξύ τους. Η έμφαση είναι στο πλήθος των συσκευών.

Καθώς ο αριθμός των δρομολογητών αυξάνεται, η καθυστέρηση που παρατηρείται για τον υπολογισμό, αποθήκευση και επικοινωνία των πληροφοριών δρομολόγησης γίνεται απαγορευτική. Η αποθήκευση πληροφοριών δρομολόγησης στους υπολογιστές υπηρεσίας θα απαιτούσε τεράστιες ποσότητες μνήμης. Η καθυστέρηση για εκπομπή ενημερώσεων κατάστασης ζεύξης μεταξύ όλων των δρομολογητών στο Διαδίκτυο δε θα άφηνε καθόλου εύρος ζώνης για αποστολή πακέτων δεδομένων. Ενώ οι αλγόριθμοι διανύσματος απόστασης δε θα δεν θα συνέκλιναν ποτέ λόγω του μεγάλου αριθμού των δρομολογητών.

Ερώτηση 4 (5 μονάδες)

Γιατί η δρομολόγηση στο διαδίκτυο γίνεται με τη χρήση των IP διευθύνσεων και όχι των MAC διευθύνσεων;

Οι MAC addresses είναι μόνιμες: κατά την κατασκευή ενός network adapter, η MAC address γράφεται μόνιμα στην μνήμη ROM του. Επίσης, οι MAC addresses έχουν επίπεδη δομή. Αντίθετα, μια IP address ανατίθεται δυναμικά σε έναν adapter και αλλάζει κάθε φορά που η συσκευή μετακινείται και αλλάζει υποδίκτυο. Οι IP addresses έχουν

ιεραρχική δομή, που αποτελείται από ένα network part και ένα host part. Αν το addressing γινόταν μόνο στο MAC layer, θα χανόταν η ιεραρχική δομή των διευθύνσεων. Έτσι, για να είναι δυνατό να δρομολογηθούν τα πακέτα μέσω ενδιάμεσων κόμβων και να παραδοθούν στον παραλήπτη τους, θα έπρεπε κάθε ενδιάμεσος κόμβος να διατηρεί μια βάση δεδομένων με πληροφορίες δρομολόγησης προς όλους τους κόμβους του δικτύου. Επίσης, το prefix matching (ταίριασμα προθέματος) δεν θα ήταν δυνατό, με αποτέλεσμα η διαδικασία της δρομολόγησης να γινόταν ακόμα πιο αργή. Ενώ ένα ακόμη θέμα θα ήταν η δυσκολία επεκτασιμότητας του δικτύου.

Ερώτηση 5 (5 μονάδες)

Έχουν IP διευθύνσεις οι δρομολογητές; Αν ναι, πόσες;

Ναι έχουν μία IP διεύθυνση για κάθε διεπαφή.

Ερώτηση 6 (5 μονάδες)

Είναι αλήθεια ότι όλα τα αυτόνομα συστήματα χρησιμοποιούν τον ίδιο intra-AS αλγόριθμο δρομολόγησης;

Όχι, δεν χρησιμοποιούν όλα τα αυτόνομα συστήματα το ίδιο intra-AS αλγόριθμο δρομολόγησης. Είναι απαραίτητο οι δρομολογητές σε ένα αυτόνομο σύστημα να τρέχουν τον ίδιο intra-AS αλγόριθμο δρομολόγησης, αλλά τα διαφορετικά αυτόνομα συστήματα μπορούν να χρησιμοποιούν διαφορετικά intra-AS πρωτόκολλα.

Ερώτηση 7 (5 μονάδες)

Περιγράψτε πως το BGP χρησιμοποιεί το NEXT-HOP χαρακτηριστικό και πως το AS-PATH;

Το NEXT-HOP χαρακτηριστικό αναφέρει την IP διεύθυνση του επόμενου δρομολογητή σε ένα advertised path (εκτός του αυτόνομου συστήματος). Χρήση αυτού του χαρακτηριστικού γίνεται κατά την διαμόρφωση του πίνακα δρομολόγησης.

Το AS-PATH χρησιμοποιείται για την ανίχνευση και έτσι την αποτροπή της δημιουργίας «κύκλων» στο μονοπάτι, επίσης χρησιμοποιείται για την επιλογή ανάμεσα σε διαδρομές με το ίδιο πρόθεμα

Ερώτηση 8 (5 μονάδες)

MTU είναι η μέγιστη μονάδα μεταφοράς (maximum transfer unit) μίας ζεύξης και καθορίζει το μέγιστο μέγεθος που μπορεί να έχει ένα δεδομένογράμμο για να μπορέσει να σταλεί μέσω της συγκεκριμένης ζεύξης. Αν ένα IP δεδομένογράμμο έχει μέγεθος μεγαλύτερο από το MTU, τότε τεμαχίζεται σε μικρότερα τμήματα (fragmentation) και στη συνέχεια αποστέλλεται.

Ας θεωρήσουμε ότι εμείς θέλουμε να στείλουμε ένα IP datagram μεγέθους 3000 bytes μέσω μίας ζεύξης με MTU 500 bytes. Πόσα τεμάχια (fragments) παράγονται για αποστολή και τι τιμές θα έχουν στο πεδίο offset της κεφαλίδας τους;

MTU = 500 bytes, άρα κάθε IP datagram μπορεί να περιέχει δεδομένα μεγέθους $500 - 20 \text{ header bytes} = 480 \text{ bytes}$.

Το πακέτο που θέλω να στείλω έχει μέγεθος 3000 bytes, άρα 2980 bytes δεδομένα.

Ο αριθμός των πακέτων που θα στείλω είναι $2980/480 = 6,2$ δηλαδή 7 πακέτα, με το 7ο πακέτο να μην γεμίζει πλήρως τα 480 data bytes που προσφέρονται.

Θα ισχύει:

- 1ο πακέτο έχει 0 ως 479 data bytes και offset = 0
- 2ο πακέτο έχει 480 ως 959 data bytes και offset = 480
- 3ο πακέτο έχει 960 ως 1439 data bytes και offset = 1440
- 4ο πακέτο έχει 1440 ως 1919 data bytes και offset = 1920
- 5ο πακέτο έχει 1920 ως 2399 data bytes και offset = 2400
- 6ο πακέτο έχει 2400 ως 2879 data bytes και offset = 2880
- 7ο πακέτο έχει 2880 ως 2980 data bytes και offset = 2980

Ερώτηση 9 (5 μονάδες)

Αν θέλουμε να έχουμε ένα υποδίκτυο (subnet) το οποίο θα επιτρέπει την ύπαρξη τουλάχιστον 8000 διευθύνσεων για κόμβους, ποια μάσκα διεύθυνσης (subnet mask) θα ήταν η καλύτερη επιλογή;

$2^{13} = 8192$, οπότε χρειαζόμαστε 13 bits για να εκφράσουμε 8000 διαφορετικές διευθύνσεις κόμβων.

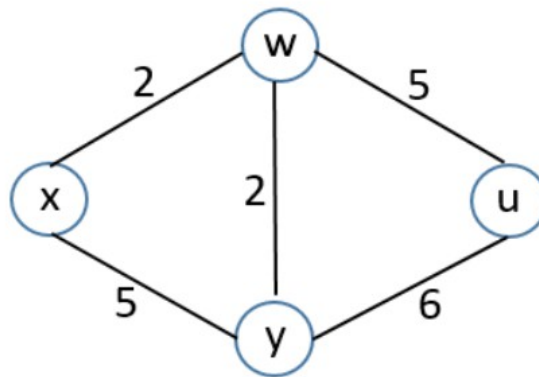
Επομένως η subnet mask του συγκεκριμένου υποδικτύου θα είναι

11111111.11111111.11100000.00000000₍₂₎ ή διαφορετικά

255.255.224.0₍₁₀₎

Ερώτηση 10 (5 μονάδες)

Θεωρήστε το δίκτυο που φαίνεται στην παρακάτω εικόνα και αποτελείται από τους δρομολογητές x, y, w και u. Χρησιμοποιείστε ως αλγόριθμο δρομολόγησης τον αλγόριθμο απόστασης διανυσμάτων (distance vector).



- a) Δώστε την απόσταση διανύσματος με βάση το κόστος του x για τους κόμβους w, y και u.
- $D_x(w) = c(x, w) = 2$
- $D_x(y) = c(x, w) + c(w, y) = 2 + 2 = 4$
- $D_x(u) = c(x, w) + c(w, u) = 2 + 5 = 7$
- b) Δώστε μία αλλαγή στο κόστος μίας ζεύξης, είτε για το $c(x, y)$, είτε για $c(x, w)$, τέτοια ώστε ο κόμβος x:
- i. Να πρέπει να ενημερώσει τους γείτονές του για ένα νέο μονοπάτι ελάχιστου κόστους προς τον u (εκτελώντας αλγόριθμο απόστασης διανύσματος).

Έστω ότι ο κόμβος x προσπαθεί να προσεγγίσει τον κόμβο u μέσω του κόμβου w, οπότε το κόστος είναι $D_x(u) = c(x, w) + c(w, u) = 2 + 5 = 7$, ενώ μέσω του y το κόστος μονοπατιού θα είναι $D_x(u) = c(x, y) + c(y, u) = 5 + 6 = 11$. Μία αλλαγή στο κόστος του $c(x, w)$ $\delta > 6$, κάνει πλέον μονοπάτι ελάχιστου κόστους αυτό που περνάει από τον y και όχι αυτό που περνάει από το w. Άρα ο x θα πρέπει να ενημερώσει τους γείτονές του με την αλλαγή που έγινε για το νέο μονοπάτι ελάχιστου κόστους. Να σημειωθεί ότι οποιαδήποτε αλλαγή στο $c(x, w)$ $\delta \leq 6$ δεν θα αναγκάσει τον x να ενημερώσει τους γείτονές του, καθώς η εκτέλεση αλγορίθμου απόστασης διανύσματος δε θα αναδείξει νέο μονοπάτι ελάχιστου κόστους.

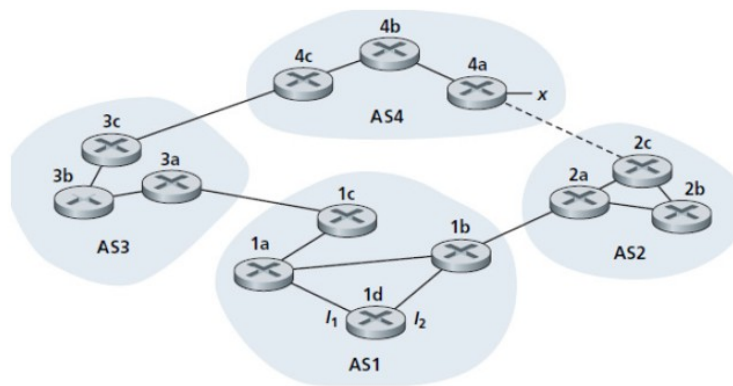
- ii. Να μην πρέπει να ενημερώσει τους γείτονες του για ένα νέο μονοπάτι.

Έστω ότι αλλάζει το κόστος του $c(x, y)$. Το μονοπάτι του x για να προσεγγίσει τον u μέσω του y θα έχει συνολικό κόστος $D_x(u) = c(x, y) + c(y, u) = c(x, y) + 6$, ακόμη και αν το $c(x, y)$ γίνει ίσο με 1 το κόστος μονοπατιού μέσω του w είναι $D_x(u) = c(x, w) + c(w, u) = 2 + 5 = 7$. Οπότε με αυτή την αλλαγή δεν υπάρχει λόγος ο x να ενημερώσει τους γείτονές του.

Ερώτηση 11 (5 μονάδες)

Ένα πρωτόκολλο δρομολόγησης ενδοαυτόνομου συστήματος χρησιμοποιείται για να καθορίσει πως γίνεται η δρομολόγηση μέσα σε ένα αυτόνομο σύστημα (AS). Δύο γνωστά πρωτόκολλα είναι το πρωτόκολλο δρομολόγησης πληροφοριών / routing information protocol (RIP) και το πρωτόκολλο ανοιχτή πρώτη η βραχύτερη διαδρομή / open shortest path first (OSPF). Από την άλλη, το πρωτόκολλο BGP χρησιμοποιείται για την δρομολόγηση μεταξύ διαφορετικών αυτόνομων συστημάτων.

Θεωρήστε το δίκτυο που φαίνεται στην παρακάτω εικόνα. Έστω ότι τα αυτόνομα συστήματα AS2 και AS3 χρησιμοποιούν το ίδιο πρωτόκολλο δρομολόγησης ενδο-ΑΣ (intra-AS), το OSPF, ενώ τα AS1 και AS4 χρησιμοποιούν το RIP. Υποθέστε ότι το eBGP και το iBGP χρησιμοποιούνται ως πρωτόκολλα δρομολόγησης δια-αυτόνομου συστήματος (inter-AS).



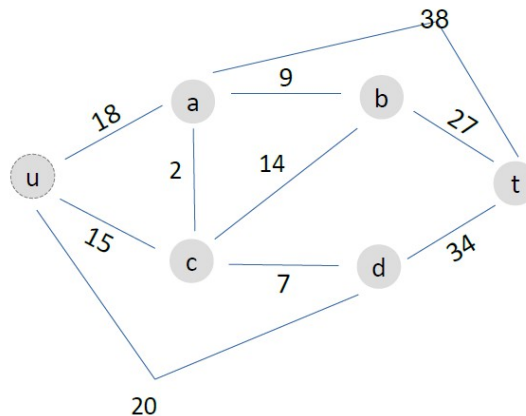
- a) Ποιά είναι η βασική διαφορά ανάμεσα σε eBGP και το iBGP;

Το eBGP είναι ένα πρωτόκολλο δρομολόγησης ανάμεσα σε διαφορετικά αυτόνομα συστήματα. Ενώ το iBGP χρησιμοποιείται για κατανομή πληροφοριών δρομολόγησης σε δρομολογητές μέσα στο αυτόνομο σύστημα, οι οποίες αφορούν άλλα αυτόνομα συστήματα.

- b) Απαντήστε από ποιό πρωτόκολλο δρομολόγησης δια-ΑΣ μαθαίνει για το πρόθεμα x:
- i. Ο δρομολογητής 3c **eBGP**
 - ii. Ο δρομολογητής 3a **iBGP**
 - iii. Ο δρομολογητής 1c **eBGP**
 - iv. Ο δρομολογητής 1d **iBGP**

Άσκηση 1 (15 μονάδες)

Θεωρήστε το δίκτυο που απεικονίζεται παρακάτω με u τον κόμβο προέλευσης.



- a) Υπολογίστε τις διαδρομές ελάχιστου κόστους από τον u προς όλους τους πιθανούς κόμβους, εκτελώντας τον αλγόριθμο κατάστασης ζεύξης (αλγόριθμος του Dijkstra). Για την εκτέλεση του αλγορίθμου και την εμφάνιση των αποτελεσμάτων κάθε βήματος χρησιμοποιήστε κατάλληλο πίνακα.

N'	D(c),p(c)	D(a),p(a)	D(d),p(d)	D(b),p(b)	D(t),p(t)
u	15, u	18, u	20, u	∞	∞
uc	-	17,c	20, u (22,c)	29,c	∞
uca	-	-	20, u	26, a	55, a
ucad	-	-	-	26, a	54, d
ucadb	-	-	-	-	53, b
ucadbt	-	-	-	-	-

- b) Ποια θα είναι η διαδρομή και το κόστος της διαδρομής ελάχιστου κόστους που υπολογίσατε στο α), θεωρώντας ως κόμβο προορισμού

- τον κόμβο b $u \rightarrow c \rightarrow a \rightarrow b$ (κόστος = 26)
- τον κόμβο d $u \rightarrow d$ (κόστος = 20)
- τον κόμβο t $u \rightarrow c \rightarrow a \rightarrow b \rightarrow t$ (κόστος = 53)

- c) Έχοντας εκτελέσει τον αλγόριθμο, είμαστε σε θέση να συμπληρώσουμε τον πίνακα προώθησης του κόμβο u. Για κάθε πιθανό προορισμό, γράψτε την ζεύξη στην οποία θα προωθείται το αντίστοιχο πακέτο.

Θεωρούμε ότι η ζεύξη προς το κόμβο α αντιστοιχεί στον interface 0, προς τον c το interface 1 και προς τον κόμβο d στο interface 2.

Destination	Interface
a	0
b	1
c	1
d	2
t	1

Άσκηση 2 (15 μονάδες)

- a) Ένας Internet Service Provider (ISP) δίνει σε μία εταιρεία το μπλοκ διευθύνσεων (subnet) 149.53.82.0/24. Η εταιρεία χρειάζεται να χωρίσει το δίκτυο σε 20 ξεχωριστά subnets. Ποιες θα είναι οι 20 διευθύνσεις δικτύου (network addresses) και ποιο το εύρος της κάθε μίας σε διευθύνσεις κόμβων (host addresses);

$$2^4 = 16 < 20$$

$2^5 = 32 > 20$ οπότε χρειάζομαι 5 bits για να εκφράσω 20 διαφορετικές διευθύνσεις δικτύου.

Το υποδίκτυο που έχω στη διάθεση μου και πρέπει να χωρίσω σε 20 μικρότερα subnets έχει διεύθυνση

149.53.82.0/24 ή αλλιώς

10010101.00110101.01010010.00000000 με τα bold bits να είναι τα δεσμευμένα.

Θα χρειαστεί να δεσμεύσω 5 bits από τα πιο σημαντικά των ελεύθερων. Κι έτσι οι διευθύνσεις των υποδικτύων που θέλω είναι:

10010101.00110101.01010010.00000000 → 149.53.82.0

10010101.00110101.01010010.00001000 → 149.53.82.8

10010101.00110101.01010010.00010000 → 149.53.82.16

...

...

10010101.00110101.01010010.10001000 → 149.53.82.136
10010101.00110101.01010010.10010000 → 149.53.82.144
10010101.00110101.01010010.10011000 → 149.53.82.152

Κάθε διεύθυνση επιτρέπει 8 διευθύνσεις κόμβων.

Παρατηρούμε ότι μένουν αχρησιμοποίητες οι διευθύνσεις δικτύου από 149.53.82.160 ως

149.53.82.248. Γι' αυτό το λόγο είναι καλό να χωρίζουμε το δίκτυο σε μεγέθη που είναι δυνάμεις του 2 (2,4,8,16 κλπ) ώστε να αξιοποιούνται όλες οι δυνατές διευθύνσεις.

- b) Θεωρήστε ότι έχουμε στη διάθεση μας το μπλοκ διευθύνσεων 149.53.32.0/19 και αποφασίζουμε να το χωρίσουμε σε 4 μικρότερα μπλοκ διευθύνσεων, κάθε ένα από τα οποία δίνουμε και σε ένα φίλο μας.

149.53.32.0/19 → **10010101.00110101.00100000.00000000** η διεύθυνση υποδικτύου

$2^2 = 4$ άρα θα χρειαστεί να δεσμεύσουμε 2 πιο σημαντικά bits από αυτά που είναι ελεύθερα.

- i. Ποια θα είναι η διεύθυνση του κάθε subnet;

1° subnet = **10010101.00110101.00100000.00000000** → 149.53.32.0/21

2° subnet = **10010101.00110101.00101000.00000000** → 149.53.40.0/21

3° subnet = **10010101.00110101.00110000.00000000** → 149.53.48.0/21

4° subnet = **10010101.00110101.00111000.00000000** → 149.53.56.0/21

- ii. Πόσα subnets μήκους /24 έχει στη διάθεση του ο κάθε φίλος μας;

Κάθε ένας φίλος μας έχει μία διεύθυνση υποδικτύου με δεσμευμένα τα 21 πιο σημαντικά bits. Αν

χωρίσουμε την κάθε διεύθυνση υποδικτύου /21 σε υποδίκτυα /24, τότε για κάθε διεύθυνση /21 θα έχουμε 8 subnets /24.

Δικαιολογήστε τις απαντήσεις σας.

Άσκηση 3 (7 μονάδες)

Ας υποθέσουμε ότι έχουμε έναν router με το παρακάτω forwarding table

Destination	Interface	Binary
0.0.0.0/0	0	00000000.00000000.00000000.00000000
149.53.128.0/17	1	10010101.00110101.10000000.00000000
149.53.128.0/19	2	10010101.00110101.10000000.00000000
149.53.160.0/19	3	10010101.00110101.10100000.00000000
149.53.192.0/19	4	10010101.00110101.11000000.00000000

Πακέτα φτάνουν στο router και πρέπει να προωθηθούν σε μία από τις 5 ζεύξεις. Γνωρίζουμε ότι ο router προωθεί τα πακέτα στη ζεύξη που σχετίζεται με το ταίριασμα μεγαλύτερου προθέματος της διεύθυνσης προορισμού (longest prefix matching). Δηλαδή βρίσκει την μεγαλύτερη καταχώρηση που ταιριάζει μέσα στον πίνακα. Για κάθε μία από τις παρακάτω διευθύνσεις προορισμού, βρείτε την ζεύξη στην οποία θα προωθείται το αντίστοιχο πακέτο.

Ο router θα προωθήσει τα πακέτα στη ζεύξη η οποία αντιστοιχεί στη διεύθυνση όπου γίνεται το μεγαλύτερο ταίριασμα προθέματος (longest prefix matching). Μετατρέψτε τις διευθύνσεις στη δυαδική τους μορφή, δεσμεύστε τον αριθμό των πιο σημαντικών bits που ορίζει η κάθε διεύθυνση και ελέγξτε ποιά διεύθυνση destination θα επιλεγεί σε κάθε περίπτωση από τις παρακάτω.

Παρατήρηση: Η διεύθυνση 0.0.0.0/0 ταιριάζει με όλες τις διευθύνσεις από τη στιγμή που δεν απαιτεί να ταιριάζει κάποιο prefix. Θεωρήστε, λοιπόν, ότι κάνει match με όλες τις παρακάτω διευθύνσεις.

- (a) 149.53.214.50 \rightarrow **10010101.00110101.11010110.00110010**₍₂₎
matches **10010101.00110101.10000000.00000000** (interface 1)
matches **10010101.00110101.11000000.00000000** (interface 4)
 Οπότε επιλέγεται το interface 4.
- (b) 149.53.168.36 \rightarrow **10010101.00110101.10101000.00100100**₍₂₎
matches **10010101.00110101.10000000.00000000** (interface 1)
matches **10010101.00110101.10100000.00000000** (interface 3)
 Οπότε επιλέγεται το interface 3.
- (c) 149.53.155.40 \rightarrow **10010101.00110101.10011011.00101000**₍₂₎
matches **10010101.00110101.10000000.00000000** (interface 1)
matches **10010101.00110101.10000000.00000000** (interface 2)
 Οπότε επιλέγεται το interface 2.
- (d) 149.53.199.111 \rightarrow **10010101.00110101.11000111.01101111**₍₂₎
matches **10010101.00110101.10000000.00000000** (interface 1)
matches **10010101.00110101.11000000.00000000** (interface 4)
 Οπότε επιλέγεται το interface 4.
- (e) 149.53.208.42 \rightarrow **10010101.00110101.11010000.00101010**₍₂₎
matches **10010101.00110101.10000000.00000000** (interface 1)
matches **10010101.00110101.11000000.00000000** (interface 4)
 Οπότε επιλέγεται το interface 4.
- (f) 149.53.224.200 \rightarrow **10010101.00110101.11100000.11001000**₍₂₎
matches **10010101.00110101.10000000.00000000** (interface 1)
 Οπότε επιλέγεται το interface 1.
- (g) 149.53.127.11 \rightarrow **10010101.00110101.01111111.00001011**₍₂₎
 Κανένα prefix της διεύθυνσης δεν ταιριάζει με κάποια διεύθυνση destination. Οπότε επιλέγεται το interface 0, αφού δεν απαιτεί να κάνει match κάποιο bit.
- (h) 149.53.179.20 \rightarrow **10010101.00110101.10110011.00010100**₍₂₎
matches **10010101.00110101.10000000.00000000** (interface 1)
matches **10010101.00110101.10100000.00000000** (interface 3)
 Οπότε επιλέγεται το interface 3.

Άσκηση 4 (8 μονάδες)

Το άθροισμα ελέγχου κεφαλίδας βοηθά έναν δρομολογητή να ανιχνεύσει σφάλματα bit στην κεφαλίδα του λαμβανόμενου IP δεδομενογράμματος. Θεωρούμε ότι ένα IP δεδομενόγραμμα έχει τους headers που εμφανίζονται παρακάτω.

0	4	8	16	19	31
Version	Header Length	Service Type	Total Length		
Identification			Flags	Fragment Offset	
TTL		Protocol	Header Checksum		
Source IP Addr					
Destination IP Addr					

Λαμβάνουμε ένα IP datagram και διαβάζοντας τους headers του, παίρνουμε τις εξής τιμές στο δεκαεξαδικό.

0	16	31
4 5 0 0 0 0 3 C		
1 C 4 6 4 0 0 0		
4 0 0 6 B 1 E 6		
A C 1 0 0 A 6 3		
A C 1 0 0 A 0 B		

Υπολογίστε το άθροισμα ελέγχου κεφαλίδας του λαμβανόμενου IP datagram και αποφασίστε αν θα θεωρήσετε το λαμβανόμενο πακέτο σωστό ή λάθος. Δικαιολογήστε την απάντησή σας.

Η τιμή του Header Checksum είναι, σύμφωνα με τις εικόνες 5.1 και 5.2, $B1E6_{(16)}$, ή αλλιώς $1011\ 0001\ 1110\ 0110_{(2)}$

Ας υπολογίσουμε τώρα το άθροισμα ελέγχου κεφαλίδας του πακέτου που λήφθηκε.

$$\begin{aligned}
 &4500_{(16)} \rightarrow 0100010100000000_{(2)} \\
 + &003C_{(16)} \rightarrow 0000000000111100_{(2)} \\
 = &453C_{(16)} \rightarrow 0100010100111100_{(2)}
 \end{aligned}$$

$$\begin{aligned}
 &453C_{(16)} \rightarrow 0100010100111100_{(2)} \\
 + &1C46_{(16)} \rightarrow 0001110001000110_{(2)} \\
 = &6182_{(16)} \rightarrow 0110000110000010_{(2)}
 \end{aligned}$$

$$\begin{aligned}
 &6182_{(16)} \rightarrow 0110000110000010_{(2)} \\
 + &4000_{(16)} \rightarrow 0100000000000000_{(2)} \\
 = &A182_{(16)} \rightarrow 1010000110000010_{(2)}
 \end{aligned}$$

$$\begin{aligned}
 &A182_{(16)} \rightarrow 1010000110000010_{(2)} \\
 + &4006_{(16)} \rightarrow 0100000000000110_{(2)} \\
 = &E188_{(16)} \rightarrow 1110000110001000_{(2)}
 \end{aligned}$$


```

E188(16) → 1110000110001000(2)
+ AC10(16) → 1010110000010000(2)
=18D98(16) → 11000110110011000(2) (υπερχείλιση πέραν των 16-bits)

```

```

8D98(16) → 1000110110011000(2)
+ 0001(16) → 0000000000000001(2)
= 8D99(16) → 1000110110011001(2)
(Προσθέτουμε το επιπλέον bit στο τρέχον 16-μπιτο αποτέλεσμα, για να διατηρήσουμε το
αποτέλεσμα 16-μπιτο αριθμό και συνεχίζουμε κανονικά)

```

```

8D99(16) → 1000110110011001(2)
+ 0A63(16) → 0000101001100011(2)
= 97FC(16) → 1001011111111100(2)

```

```

97FC(16) → 1001011111111100(2)
+ AC10(16) → 1010110000010000(2)
=1440C(16) → 10100010000001100(2) (υπερχείλιση πέραν των 16-bits)

```

```

440C(16) → 0100010000001100(2)
+ 0001(16) → 0000000000000001(2)
= 440D(16) → 0100010000001101(2)

```

```

440D(16) → 0100010000001101(2)
+ 0A0B(16) → 0000101000001011(2)
= 4E18(16) → 0100111000011000(2)

```

Τώρα θα κάνω το συμπλήρωμα ως προς 1 (one's complement) του αθροίσματος που υπολόγισα, δηλαδή θα αντιστρέψω τα bits και θα έχω **1011 0001 1110 0111**₍₂₎.

Όμως, το checksum του πακέτου που λάβαμε είχε υπολογιστεί κατά την αποστολή να έχει την τιμή **1011 0001 1110 0110**₍₂₎

Κάνοντας το bitwise operation XOR μεταξύ των δύο τιμών, θα έχω αποτέλεσμα

```

    1011 0001 1110 0111(2)
XOR 1011 0001 1110 0110(2)

```

```

0000 0000 0000 0001(2)

```

το οποίο σημαίνει ότι οι 2 bits διαφέρουν, κι επομένως θεωρούμε το πακέτο που λάβαμε ως λανθασμένο.

Bonus Άσκηση (5 μονάδες)

Εκτελέστε την εντολή `netstat -r` στον υπολογιστή σας και αφού μελετήσετε το output της εντολής, περιγράψτε το στην αναφορά σας, στην οποία πρέπει να περιλαμβάνεται και ένα screenshot του output.