2η ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ ΔΙΚΤΥΑ ΗΥ335α

ΚΩΝΣΤΑΝΤΙΝΟΣ ΣΤΟΛΑΚΗΣ 3220

ΔΗΜΗΤΡΗΣ ΚΑΡΝΙΚΗΣ 3215

ΑΛΕΞΑΝΔΡΟΣ ΜΑΡΑΝΤΟΣ 3329

ΕΡΩΤΗΣΕΙΣ

Ερώτηση 1)

Η λειτουργία της προώθησης (forwarding) αναφέρεται στα πακέτα που φτάνουν σε ένα σύστημα αλλά δε προωρίζονται για αυτό το σύστημα. Έτσι η λειτουργία αυτή έχει σκοπό να λάβει το πακέτο, να βρει τον παραλήπτη και να το στειλει σε αυτόν αντί να το καταστρέψει.

Η λειτουργία της δρομολόγησης (routing) επιλέγει ποια διεπαφή(interface) θα σταλεί ένα συγκεκριμένο πακέτο (είτε το πακέτο είναι από το διαδίκτυο είτε είναι τοπικά δημιουργημένο).

Ερώτηση 2)

Ο αλγόριθμος απόστασης διανύσματος(distance-vector) ονομάζεται έτσι από δύο παράγοντες

α)την απόσταση μέχρι τον προορισμό

β)την διαδρομή που θα ακολουθήσει για να φτάσει σε αυτόν.

Οι πληροφορίες ανταλλάζονται μόνο μεταξύ των γειτονικά συνδεδεμένων κόμβων. Αυτό έχει ως συνέπεια ένα router A να μαθαίνει μια πληροφορία από ένα router B χωρίς όμως να γνωρίζει ο A από που έλαβε την πληροφορία ο B

Ο αλγόριθμος δρομολόγησης κατάστασης (link-state) όμως απαιτεί να γνωρίζουν όλοι οι routers όλα τα δυνατά μονοπάτια από όλους τους υπόλοιπους router στο δίκτυο. Όλες οι πληροφορίες υπάρχουν σε όλα τα router με αποτέλεσμα όλα τα router να δουλεύουν στην ίδια βάση δεδομένων συγχρονισμένα. Με αυτό τον τρόπο κάθε router βρίσκει το μικρότερο δυνατό μονοπάτι με τον εαυτό του ως αρχή.

Ερώτηση 3)

Στο διαδικτυο ειναι συχνο φαινομενο ανα περιοδους καποιο δικτυο να υπερφορτωνεται με πολλους χρηστες και πολλες συσκευες που αλληλεπιδρουν μεταξυ τους.

Το προβλημα βρισκεται στο οτι ολοι αυτοι οι διαφορετικοι χρηστες «ανταγωνιζονται» για τους τους πορους των δρομολογητων, δηλαδη το ευρος ζωνης αλλα και τον απαθηκευτικο χωρο. Οταν λοιπον οι χρηστες και οι συσκευες αντιστοιχα ξεπερνουν καποιο οριο σε αριθμο, ειναι λογικο η δρομολογηση να ειναι εν μερει προβληματικη

Ερώτηση 4)

Ο λόγος που χρησιμοποιούμε IP αντί MAC διευθύνσεις στο διαδίκτυο είναι διότι οι IP διευθύνσεις διευκρινίζουν αν κάποια διεύθυνση ανήκει στο δίκτυο.

Οι MAC διευθύνσεις από την άλλη είναι μοναδική σε κάθε συσκευή (host) και δε επιτρέπουν στον router να εντοπίσει την τοποθεσία του host. Επίσης δεν έχουν όλες οι συσκευές MAC διευθύνσεις αφού είναι προνόμιο μόνο του πρωτοκόλλου ethernet

Ερώτηση 5)

Προφανώς οι routers χρησιμοποιούν IP διευθύνσεις και πρέπει να έχουν τουλάχιστον 2. Μία για τα δεδομένα που έρχονται προς το δίκτυο και μία για τα δεδομένα που φεύγουν από αυτό. Πέρα από τις 2 όμως, κάθε router μπορεί να έχει όσες διευθύνσεις θελήσει για κάθε διεπαφή που χρησιμοποεί αρκεί βέβαια να έχει τις απαραίτητες προδιαγραφές για να τις υποστηρίξει.

Ερώτηση 6)

Όχι, δε χρησιμοποιούν όλα τα αυτονομα συστήματα intra-as αλγόριθμο δρομολόγησης αλλά ομαδοποιούνται αυτά σε διαφορετικές περιπτώσεις

α) Δρομολογητές στο ίδιο AS τρέχουν το ίδιο πρωτόκολλο δρομολόγησης

“intra-AS” πρωτόκολλο δρομολόγησης

β) Δρομολογητές σε *διαφορετικά AS* μπορεί να τρέχουν διαφορετικά intra-AS πρωτόκολλα δρομολόγησης

Ερώτηση 7)

Το BGP πρωτοκολλο χρησιμοποιει την λειτουργια του next hop για να προσδιορισει ουσιαστικα την IP διευθυνση που πρεπει να χρησιμοποιηθει για τον συγκεκριμενο προορισμο (αναλογα την περιπτωση).

Γενικα το next-hop παρεχει περισσοτερη ευελιξια στην κατασκευη του δικτυου αφου επιτρεπει περισσοτερη παραμετροποιηση στο next hop attribute και επιπλεον επιτρεπει στο BGP να στελνει πληροφοριες ενημερωσης στο eBGP

Το AS-PATH attribute του BGP χρησιμοποιειται απο τον δεκτη για να καθορισει διαμεσου ποιου AS περασε η πληροφορια ενω το AS number του αποστολεα εμπεριεχεται στο AS-PATH attribute οταν η ενημερωση δρομολογησης περνα απο το AS

Ερώτηση 8)

IP datagram : 3000 bytes

MTU size : 500 bytes

header = 20 bytes άρα συνολικό datagram 3020 bytes

MTU size – 20 bytes (header) = 480 bytes

αριθμός πακέτων = (3000 / 480) = 6.4 = 7 πακέτα

Κάθε πακέτο έχει μέγεθος 500 bytes , δηλαδή 480 bytes για δεδομένα

Το τελευταίο πακέτο έχει μέγεθος 20 bytes + 20 bytes (header) = 40 bytes

offset size = 480 bytes / 8 = 60 bytes

Δηλαδή το 1ο πακέτο θα έχει 0 στο πεδίο offset

Το 2ο 60 bytes

Το 3ο 120 bytes

Ερώτηση 9)

Η κατάλληλη μάσκα διεύθυνσης θα ήταν κλάσσης B και θα ήταν η εξής :

Subnet : 6

Mask : 255.255.224.0

Hosts : 8.190

Που θα επαρκούσαν να καλύψουν 8000 hosts

Ερώτηση 10)

α)

Destination	x	y	w	u
x	0	4	2	7
y	4	0	2	6
w	2	2	0	5
u	7	6	5	0

Ερώτηση 11)

a)

Αρχικα το eBGP χρησιμοποιει 2 διαφορετικα AS (αυτονομα συστηματα) ενω το iBGP το ιδιο AS.

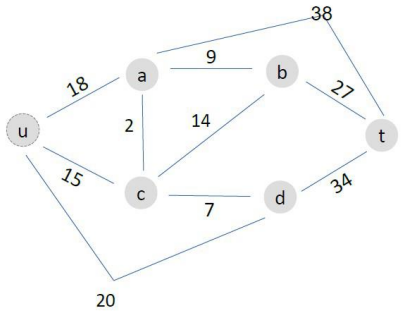
Επιπλεον, το eBGP μεταδιδει στο BGP, iBGP το «δρομολογιο» που «εμαθε» απο την δρομολογηση, ενω αντιθετα το iBGP δεν το κανει αυτο.

Οι peers του eBGP παιρνουν by default TTL = 1 πραγμα που σημαινει οτι οι γειτονες ειναι αμεσα συνδεδεμενοι, πραγμα που δεν συναντάται στο iBGP

Στο eBGP τα routes εχουν διαχειριστικη αποσταση 20 ενω αντιστοιχα στο iBGP 200.

ΑΣΚΗΣΕΙΣ

ΑΣΚΗΣΗ 1



a)

Αλγοριθμος Dijsktra

Step	N'	D(a),p(a)	D(b),p(b)	D(c),p(c)	D(d),p(d)	D(t),p(t)
0	u	18,u	∞	15,u	20,u	∞
1	uc	18,u	29,c		20,u	∞
2	uca		27,a		20,u	∞
3	ucad		27,a			54,d
4	ucadb					54,d
5	ucadbt					

b)

i) Προορισμος ο κομβος b: ελαχιστο κοστος θα ηταν το ucab , 15+2+9 = 26

ii) Προορισμος το d: ελαχιστος κοστος θα ηταν το ud, 20

iii) Προορισμος το t: ελαχιστο κοστος ηταν το udt, 20+34 = 54

ΑΣΚΗΣΗ 2

a)

Ενας ISP δινει το subnet: 149.53.82.0/24

Η εταιρια θελει να το χωρισει σε 20 διαφορετικα subnets
Ποιες θα ειναι αυτες, και ποιο το ευρος τους.

ΛΥΣΗ

149.53.82.0 -> 10010101.00110101.01001000.00000000

Εφοσον απο τον ISP ειναι δεσμευμενα ειναι τα πρωτα 24 bits (οπως οριζει το subnet), δεσμευμενα bits (που δεν επιτρεπεται να αλλαξουν δηλαδη) ειναι τα : 10010101.00110101.01001000 (24 πρωτα).

Αλλα, για να τα χωρισει η εταιρια σε 20 ξεχωριστα subnets θα δεσμευτουν αλλα 5 bits ωστε να χωρανε οι 20 διαφορετικες περιπτωσεις (2^4 = 16, 2^5 = 32)

Αρα δεσμευονται (απο την εταιρια αυτη τη φορα) και τα πρωτα 5 bits απο τα εναπομειναντα 8 (αδεσμευτα απο τον ISP)

Αρα usable bits απομενουν 3. 2^3 = 8

Ετσι λοιπον: **οι διευθυνσεις δικτυου θα ειναι οι:**	**και το ευρος καθε host address:**	
1) 10010101.00110101.01010010.**00000**000	149.53.82.0/29	address range 0 to 7
2) 10010101.00110101.01010010.**00001**000	149.53.82.8/29	address range 8 to 15
3) 10010101.00110101.01010010.**00010**000	149.53.82.16/29	address range 16 to 23
4) 10010101.00110101.01010010.**00011**000	149.53.82.24/29	address range 24 to 31
5) 10010101.00110101.01010010.**00100**000	149.53.82.32/29	address range 32 to 39
6) 10010101.00110101.01010010.**00101**000	149.53.82.40/29	address range 40 to 47
7) 10010101.00110101.01010010.**00110**000	149.53.82.48/29	address range 48 to 55
8) 10010101.00110101.01010010.**00111**000	149.53.82.56/29	address range 56 to 63
9) 10010101.00110101.01010010.**01000**000	149.53.82.64/29	address range 64 to 71
10) 10010101.00110101.01010010.**01001**000	149.53.82.72/29	address range 72 to 79
11) 10010101.00110101.01010010.**01010**000	149.53.82.80/29	address range 80 to 87
12) 10010101.00110101.01010010.**01011**000	149.53.82.88/29	address range 88 to 95
13) 10010101.00110101.01010010.**01100**000	149.53.82.96/29	address range 96 to 103
14) 10010101.00110101.01010010.**01101**000	149.53.82.104/29	address rng. 104 to 111
15) 10010101.00110101.01010010.**01110**000	149.53.82.112/29	address rng. 112 to 119
16) 10010101.00110101.01010010.**01111**000	149.53.82.120/29	address rng. 120 to 127
17) 10010101.00110101.01010010.**10000**000	149.53.82.128/29	address rng. 128 to 135
18) 10010101.00110101.01010010.**10001**000	149.53.82.136/29	address rng. 136 to 143
19) 10010101.00110101.01010010.**10010**000	149.53.82.144/29	address rng. 144 to 151
20) 10010101.00110101.01010010.**10011**000	149.53.82.152/29	address rng. 152 to 159

b)

Ειναι διαθεσιμο το μπλοκ διευθυνσεων 149.53.32.0/19
Το σπαμε σε 4 μικροτερα subnets 1 σε καθε φιλο

ΛΥΣΗ

Για να χωρισουμε το παρον subnet σε 4 ξεχωριστα subnets χρειαζομαστε 2 επιπλεον δεσμευμενα bits, αφου 2^2 = 4.

149.53.32.0 -> 10010101.00110101.00100000.00000000

Δεσμευμενα ειναι τα 19 πρωτα, δηλαδη (τα bold) : **10010101.00110101.001** 00000**.**00000000

i)

Οι διευθυνσεις του καθε subnet θα ειναι λοιπον:

1) 10010101.00110101.001**00**000.00000000
2) 10010101.00110101.001**01**000.00000000
3) 10010101.00110101.001**10**000.00000000
4) 10010101.00110101.001**11**000.00000000

ii)

Αν λοιπον δωσουμε σε καθε φιλο μια τετοια διευθυνση, θα εχει δεσμευμενα 21 bits και 11 ελευθερα

Αυτος ο φιλος θελει να δεσμευσει για δικους του λογους αλλα 3 bits και να εχει διευθυνσεις τις μορφης: /24.
Αυτα τα 3 επιπλεον bits θα του δωσουν δυνατοτητα να εχει 8 (2^3) subnet μηκους /24 ωστε να τα κατανημει οπως αυτος θελει.

ΑΣΚΗΣΗ 3

ΛΥΣΗ

Αρχικα τα μετατρεπω ολα σε binary		interface
00000000.00000000.00000000.00000000	0.0.0.0/0	0
10010101.00110101.10000000.00000000	149.53.128.0/17	1
10010101.00110101.10000000.00000000	149.53.128.0/19	2
10010101.00110101.10100000.00000000	149.53.160.0/19	3
10010101.00110101.11000000.00000000	149.53.192.0/19	4

10010101.00110101.1 ειναι το prefix για το 149.53.128.0/17
10010101.00110101.100 ειναι το prefix για το 149.53.128.0/19
10010101.00110101.101 ειναι το prefix για το 149.53.160.0/19
10010101.00110101.110 ειναι το prefix για το 149.53.192.0/19

Στη συνεχεια μετατρεπω τις δοθεισες IP διευθυνσεις σε binary επισης:

a) 149.53.214.50 -> 10010101.00110101.11010110.00110010

b) 149.53.168.36 -> 10010101.00110101.10101000.00100100
c) 149.53.155.40 -> 10010101.00110101.10011011.00101000
d) 149.53.199.111 -> 10010101.00110101.11000111.01101111
e) 149.53.208.42 -> 10010101.00110101.11010000.00101010
f) 149.53.224.200 -> 10010101.00110101.11100000.11001000
g) 149.53.127.11 -> 10010101.00110101.01111111.00001011
h) 149.53.179.20 -> 10010101.00110101..10110011.00010100

Ετσι θα βρω τις σωστες ζευξεις συμφωνα με το longest prefix matching

a) 149.53.214.50 με το interface 4 καθως εκει βρεθηκε το μακρυτερο match (10010101.00110101.110)
b) 149.53.168.36 με το interface 3 καθως εκει βρεθηκε το μακρυτερο match (10010101.00110101.101)
c) 149.53.155.40 με το interface 2 καθως εκει βρεθηκε το μακρυτερο match (10010101.00110101.100)
d) 149.53.199.111 με το interface 4 καθως εκει βρεθηκε το μακρυτερο match (10010101.00110101.110)
e) 149.53.208.42 με το interface 4 καθως εκει βρεθηκε το μακρυτερο match (10010101.00110101.110)
f) 149.53.224.200 με το interface 4 καθως εκει βρεθηκε το μακρυτερο match (10010101.00110101.11)
g) 149.53.127.11 με το interface 0
h) 149.53.179.20 με το interface 3 καθως εκει βρεθηκε το μακρυτερο match (10010101.00110101.101)

ΑΣΚΗΣΗ 4

Διαδικασία:

1) Θέτω το πεδίο header checksum = 0.
2) Μετατρέπω σε binary τις δεκαεξαδικές τιμές που έλαβα.
3) Προσθέτω όλες τις τιμές.
4) Εφαρμόζω 1's complement στο αποτέλεσμα
5) Ελέγχω αν το αποτέλεσμα είναι ίσο με το αρχικό header checksum.

4500 → 0100 0101 0000 0000

003C → 0000 0000 0011 1100

1C46 → 0001 1100 0100 0110

4000 → 0100 0000 0000 0000

4006 → 0100 0000 0000 0110

B1E6 → 1011 0001 1110 0110 // header checksum

AC10 → 1010 1100 0001 0000

0A63 → 0000 1010 0110 0011

0A0B → 0000 1010 0000 1011

1)

0100 0101 0000 0000 (4500)

+ 0100 0101 0000 0000 (003C)

= 0100 0101 0011 1100 (453C)

2)

0100 0101 0011 1100 (453C)

+ 0001 1100 0100 0110 (1C46)

= 0110 0001 1000 0010 (6182)

3)

0110 0001 1000 0010 (6182)

+ 0100 0000 0000 0000 (4000)

= 1010 0001 1000 0010 (A182)

4)

1010 0001 1000 0010 (A182)

+ 0100 0000 0000 0110 (4006)

= 1110 0001 1000 1000 (E188)

5)

1110 0001 1000 1000 (E188)

+ 1010 1100 0001 0000 (AC10)

= 1 1000 1101 1001 1000 (18D98)

= 1000 1101 1001 1000 (8D98)

6)

1000 1101 1001 1000 (8D98)

+ 0000 1010 0110 0011 (0A63)

= 1001 0111 1111 1100 (97FC)

7)

1001 0111 1111 1100 (97FC)

+ 1010 1100 0001 0000 (AC10)

= 1 0100 0100 0000 1100 (1440C)

= 0100 0100 0000 1101 (440D)

8)

0100 0100 0000 1101 (440D)

+ 0000 1010 0000 1011 (0A0B)

= 0100 1110 0001 1000 (4E18) // αποτέλεσμα

1's complement

0100 1110 0001 1000 (4E18)

1011 0001 1110 0111 (B1E7) Διαφορετικό από 1011 0001 1110 0110 (B1E6)

'Αρα:

Το λαμβανόμενο πακέτο είναι ΛΑΘΟΣ.

ΑΣΚΗΣΗ 5

```
C:\Users\alexandros>netstat -r
===========================================================================
Interface List
  7...00 90 f5 e2 99 bd ......Realtek PCIe GBE Family Controller
  4...64 5a 04 03 c0 fc ......Microsoft Wi-Fi Direct Virtual Adapter
  6...64 5a 04 03 c0 fc ......Realtek RTL8723AE Wireless LAN 802.11n PCI-E NIC
  1...........................Software Loopback Interface 1
  5...00 00 00 00 00 00 00 e0 Microsoft Teredo Tunneling Adapter
  9...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter #2
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      192.168.1.1     192.168.1.70     25
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    306
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    306
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    306
      192.168.1.0    255.255.255.0         On-link      192.168.1.70    281
     192.168.1.70  255.255.255.255         On-link      192.168.1.70    281
    192.168.1.255  255.255.255.255         On-link      192.168.1.70    281
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    306
        224.0.0.0        240.0.0.0         On-link      192.168.1.70    281
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    306
  255.255.255.255  255.255.255.255         On-link      192.168.1.70    281
===========================================================================
Persistent Routes:
  None

IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
  5    306 ::/0                     On-link
  1    306 ::1/128                  On-link
  5    306 2001::/32                On-link
  5    306 2001:0:5ef5:79fb:340e:1c79:b1a8:7491/128
                                    On-link
  6    281 fe80::/64                On-link
  5    306 fe80::/64                On-link
  6    281 fe80::11df:8c09:a85:af9f/128
                                    On-link
  5    306 fe80::340e:1c79:b1a8:7491/128
                                    On-link
  1    306 ff00::/8                 On-link
  6    281 ff00::/8                 On-link
  5    306 ff00::/8                 On-link
===========================================================================
Persistent Routes:
  None

C:\Users\alexandros>netstat -r
```

Αρχικα παρατηρουμε οτι το output της εντολης αυτης δινει τους routing tables για IPv4 και IPv6.

Στο IPv4 Route Table:

Το **destination** αναφερεται στο destination host, ειτε subnet address, ειτε network address, ειτε default route. (Το default route ειναι το 0.0.0.0)

Το **Netmask** ειναι ο μηχανισμος που μας βοηθα να προσδιορισουμε ποια ζευξη πρεπει να γινει αναλογα τα subnet.

Το **Gateway** ειναι η IP διευθυνση του επομενου router στον οποιο το πακετο πρεπει να σταλει.
Το On-Link σε αυτη τη στηλη σημαινει οτι ειναι αμεσα προσβασιμο

Το **Interface** αφορα το interface που δειχνει ποιο LAN ή demand-dial interface θα χρησιμοποιηθει για την προσεγγιση του επομενου router

Το **Metric** αφορα το σχετικο κοστος χρησης της συγκεκριμενης διαδρομης. Συνηθως ειναι ο αριθμος των routers που απαιτουνται για να φτασει η πληροφορια-πακετο στον τελικο προορισμο. Αν ελεγχουμε ιδιες διαδρομες καθε φορα, δηλαδη με τον ιδιο προορισμο, προφανως καλυτερο ειναι το μικροτερο metric.

Στο IPv6 Route Table ισχυουν περιπου τα ιδια με παραπανω