Ατομική Διπλωματική Εργασία

ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ ΔΡΟΜΟΛΟΓΗΣΗΣ ΣΕ ΤΟΠΟΛΟΓΙΑ ΜΑΝΗΑΤΤΑΝ

Σάββας Χριστοφόρου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ιούνιος 2019

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάλυση Αλγόριθμων δρομολόγησης σε Τοπολογία Manhattan

Σάββας Χριστοφόρου

Επιβλέπων Καθηγητής

Ανδρέας Πιτσιλλίδης

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Ιούνιος 2019

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω ιδιαίτερα τον καθηγητή μου, κ. Αντρέα Πιτσιλλίδη για την καθοδήγηση του καθ' όλη τη διάρκεια της μελέτης, τις πολύτιμες συμβουλές του και την εμπιστοσύνη που μου έδειξε για την εκπόνηση της παρούσας Πτυχιακής Εργασίας.

Ακόμη θα ήθελα να ευχαριστήσω τον κ. Κωνσταντίνο Σκίτσα, ο οποίος με βοήθησε στην πιο εύκολη χρήση του simulator που χρησιμοποιήθηκε.

Τέλος δεν θα μπορούσα να μην ευχαριστήσω την οικογένεια μου για την ηθική τους συμπαράσταση.

Περίληψη

Στην παρούσα διπλωματική εργασία παρουσιάζονται μετρήσεις αποστολής πακέτων σε παραλλαγή της τοπολογία Manhattan από κάποιο input gateway σε κάποιο παραλήπτη. Στη συνέχεια ο παραλήπτης αποστέλλει acknowledgment σε κάποιο acknowledgment gateway.

Συγκρίνουμε αποτελέσματα αλλάζοντας κάθε φορά κάποιες παραμέτρους όπως το μέγεθος του πλέγματος, τον αλγόριθμο δρομολόγησης που χρησιμοποιήθηκε, το μοτίβο αποστολής πακέτων, καθώς και την θέση του acknowledgment gateway.

Για να εξάγουμε συμπεράσματα συγκρίνουμε τον αριθμό των hops, την ύπαρξη deadlocks καθώς και τον χρόνο προσομοίωσης. Επίσης βλέπουμε ποιοι κόμβοι χρησιμοποιούνται περισσότερο.

Περιεχόμενα

Κεφάλαιο **1 Εισαγωγή** **1**

1.1 Στόχος 1

1.2 Οργάνωση Κειμένου 1

Κεφάλαιο **2 Τοπολογία** **2**

2.1 Τοπολογία (ελεγκτές δικτύου, acknowledgment gateway και input gateway) 2

2.2 Διευθυνσιοδότηση κόμβων 4

2.3 Μορφή των πακέτων 4

2.4 Μοτίβα 5

2.4.1 Μοτίβο 1 5

2.4.2 Μοτίβο 2 6

2.4.3 Μοτίβο 3 7

2.4.4 Μοτίβο 4 7

2.4.5 Μοτίβο 5 8

2.4.6 Μοτίβο 6 8

2.4.7 Μοτίβο 7 9

2.4.8 Μοτίβο 8 10

Κεφάλαιο **3 Αλγόριθμοι** **12**

3.1 XY before 12

3.2 XY after 13

3.3 Χρήση simulator σε Anylogic 14

3.3.1 Simulator Input Data 15

3.3.2 Simulator Output Data 16

3.3.2.1 Apotelesmata.txt 16

3.3.2.2 Route.txt 17

3.3.2.3 Steps.txt 17

Κεφάλαιο **4 Πειραματικά αποτελέσματα** **19**

4.1 Μετρήσεις 19

Κεφάλαιο **5 Συμπεράσματα** **22**

5.1 Εισαγωγή 22
5.2 Number of hops 22
5.3 Μέγεθος Πλέγματος 23
5.4 Θέση acknowledgment gateway 23
5.5 Αλγόριθμος 23
5.6 Χρήση κόμβων 23
5.7 Μοτίβα αποστολής πακέτων 23
5.8 Ύπαρξη deadlock 24
5.9 Simulation time 27
5.10 Μελλοντική Εργασία 27
Βιβλιογραφία **29**

Κεφάλαιο 1

Εισαγωγή

1.1 Στόχος 1
1.2 Οργάνωση Κειμένου 1

1.1 Στόχος

Στόχος της παρούσας διπλωματικής εργασίας είναι η πειραματική αξιολόγηση αποστολής πακέτων από input gatewayσε κάποιο παραλήπτη. Στη συνέχεια ο παραλήπτης αποστέλλει acknowledgment σε κάποιο acknowledgment gateway.
Η αποστολή πακέτων πραγματοποιείται σε συγκεκριμένη παραλλαγή της τοπολογίας Manhattan.
Για την πιο πάνω αξιολόγησης θα δοκιμαστούν:

- διαφορετικοί αλγόριθμοι δρομολόγησης
- διαφορετική θέση της πύλης μηνυμάτων επιβεβαίωσης
- διαφορετικά μεγέθη πλέγματος
- διαφορετικά μοτίβα αποστολής πακέτων σε όλους τους κόμβους

1.2 Οργάνωση κειμένου

Στο κεφάλαιο 2 επεξηγούνται οι διάφοροι περιορισμοί και απαιτήσεις της τοπολογίας που χρησιμοποιήθηκε. Στο κεφάλαιο 3 επεξηγούνται οι δύο αλγόριθμοι που χρησιμοποιήθηκαν. Στο κεφάλαιο 4 γίνεται αναλυτική παρουσίαση των μετρήσεων που πήραμε για τα διάφορα σενάρια που μελετήσαμε. Στο κεφάλαιο 5 αξιολογούνται τα οι πιο πάνω μετρήσεις και εξάγονται τα συμπεράσματα.

ΚΕΦΑΛΑΙΟ 2

Τοπολογία

2.1 Τοπολογία (ελεγκτές δικτύου, acknowledgment gateway και input gateway) 2

2.2 Διευθυνσιοδότηση κόμβων 4

2.3 Μορφή των πακέτων 4

2.4 Μοτίβα 5

2.4.2 Μοτίβο 1 5

2.4.3 Μοτίβο 2 6

2.4.4 Μοτίβο 3 7

2.4.5 Μοτίβο 4 7

2.4.6 Μοτίβο 5 8

2.4.7 Μοτίβο 6 8

2.4.8 Μοτίβο 7 9

2.4.9 Μοτίβο 8 10

2.1 Τοπολογία (ελεγκτές δικτύου, acknowledgment gateway και input gateway)

Η τοπολογία που χρησιμοποιείται είναι παραλλαγή της τοπολογία Manhattan όπως παρουσιάζεται στο Σχήμα 2.2 με wrap arounds στα άκρα του πλέγματος. Η τοπολογία Manhattan είναι δομή πλέγματος. Η συγκεκριμένη παραλλαγή τους πλέγματος Manhattan που θα μελετήσουμε αποτελείται από κόμβους που έχουν 2 εισερχόμενες και 2 εξερχόμενες διασυνδέσεις.

Το πλέγμα έχει διαστάσεις X x Y και οι κατευθύνσεις ορίζονται ως εξής (σχήμα 2.1):

Τύπος 0: Έξοδος Πάνω και δεξιά

Είσοδος Κάτω και αριστερά

Τύπος 1: Έξοδος Κάτω και δεξιά

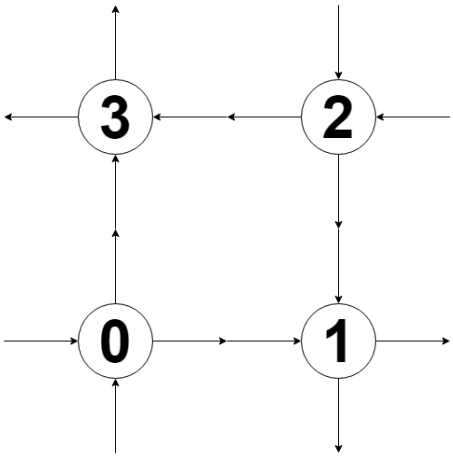
Είσοδος Πάνω και αριστερά

Τύπος 2: Έξοδος Κάτω και αριστερά

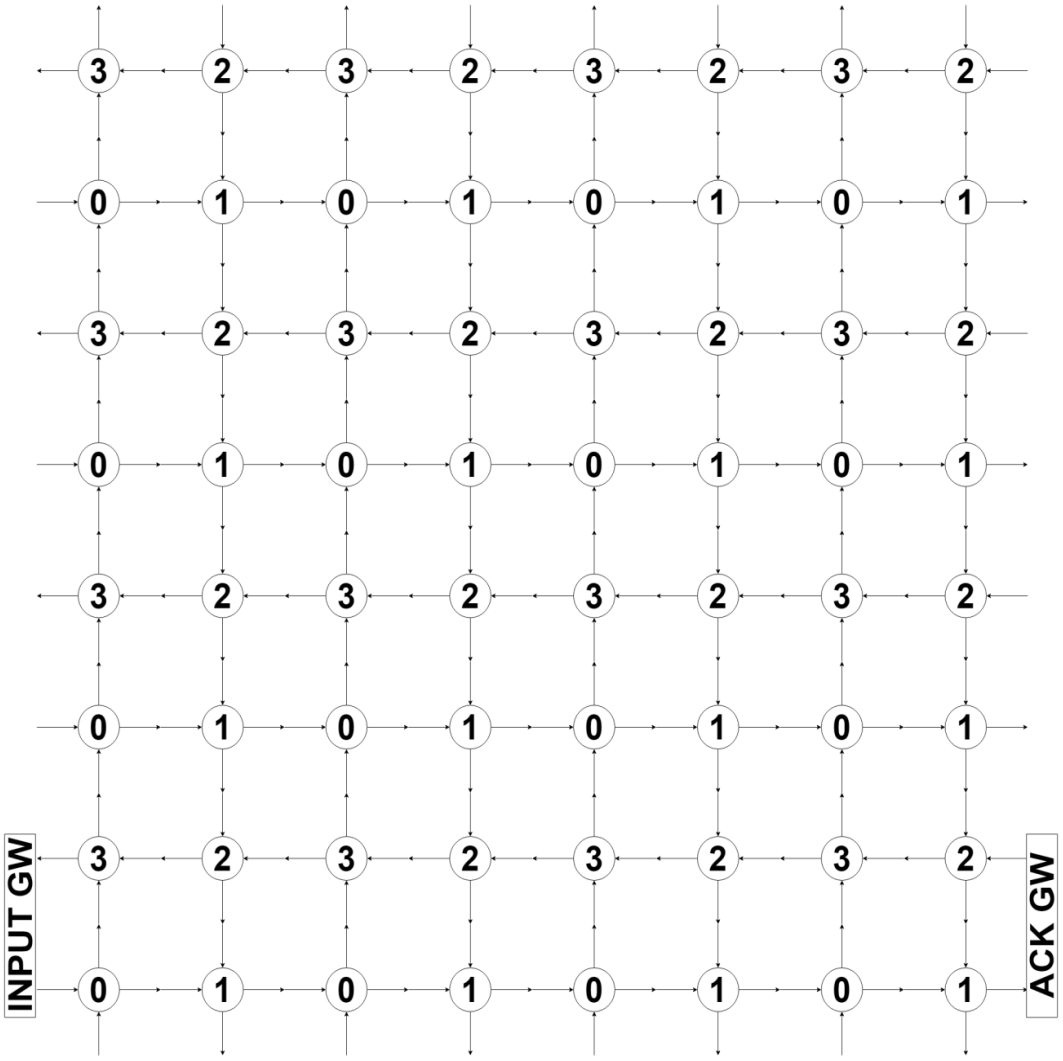
Είσοδος Πάνω και δεξιά

Τύπος 3: Έξοδος Πάνω και αριστερά

Είσοδος Κάτω και δεξιά

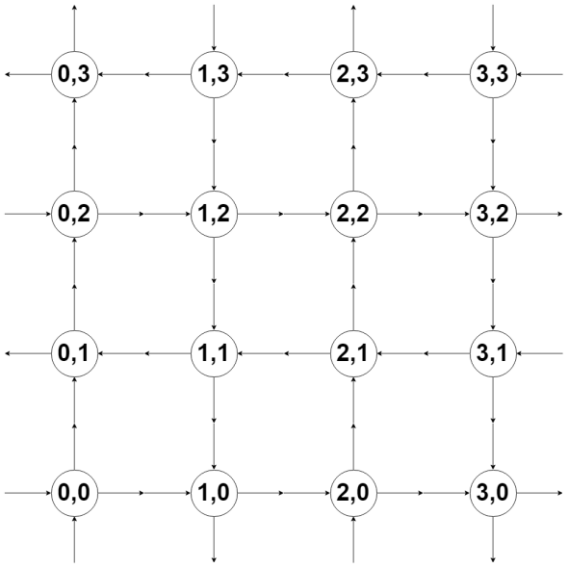


Στο δίκτυο μας υπάρχουν 2 gateways. 1 Input Gateway κάτω αριστερά και ένα acknowledgment gateway το οποίο κάποιες φορές θα είναι πάνω δεξιά κάποιες κάτω δεξιά (όπως φαίνεται στο σχήμα 2.2) και κάποιες φορές δεξιά στο κέντρο. Δοκιμάστηκε επίσης η περίπτωση να βρίσκεται στην κάτω αριστερή γωνία.



2.2 Διευθυνσιοδότηση κόμβων

Χρειάζεται να γίνει μοναδική διευθυνσιοδότηση των κόμβων για αυτό χρησιμοποιήθηκε η καρτεσιανές συντεταγμένες όπως φαίνεται στο σχήμα 2.3 Ο κόμβος 0,0 (κάτω αριστερά) είναι κόμβος τύπου 0.
Οι διευθύνσεις των κόμβων είναι μοναδικές και ο κάθε κόμβος ξέρει τη διεύθυνση του, και την διεύθυνση των τεσσάρων γειτονικών τους κόμβων.



Όταν ένας κόμβος παραλάβει κάποιο πακέτο ελέγχει την διεύθυνση προορισμού:

- Αν είναι ίδια με την δική του στέλνει acknowledgment
- Αν όχι το δρομολογεί με βάση τον αλγόριθμο δρομολόγησης που βρίσκεται στην κεφαλίδα του πακέτου.

Στην παραλλαγή που μελετούμε το δίκτυο έχει ζυγό αριθμό κόμβων σε κάθε διάσταση και ο αριθμός των κόμβων στις 2 διαστάσεις είναι ίσος

2.3 Μορφή των πακέτων

Χρησιμοποιήθηκε η πιο κάτω δομή πακέτων όπως φαίνεται στο σχήμα (2.4)

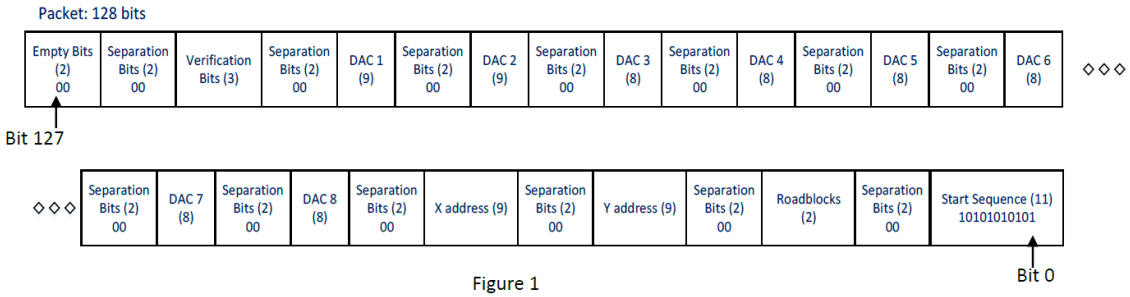


Figure 1

Τα πακέτα θα έχουν μέγεθος 128 bits και περιλαμβάνουν 4 βασικά μέρη:

1. Η κεφαλίδα του πακέτου που χωρίζεται σε 3 μέρη: Την διεύθυνση X του παραλήπτη και την διεύθυνση Y του παραλήπτη
2. Η ακολουθία (start sequence) που υποδηλώνει την αρχή ενός πακέτου από :
3. Configuration data
4. Acknowledgment verification data.

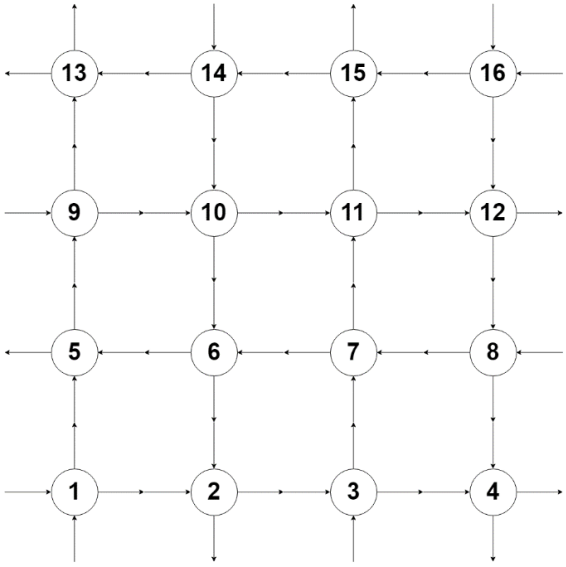
Στην ουσία για την παρούσα εργασία μας ενδιαφέρει η κεφαλίδα και τα bits για το acknowledgment

2.4 Μοτίβα

Μοτίβο ονομάζουμε την διαφορετική σειρά αποστολής πακέτων σε κάθε κόμβο. Σε κάθε σενάριο αποστέλλουμε πακέτα σε όλους τους κόμβους. Στην παρούσα διπλωματική θα εξετάσουμε οχτώ μοτίβα, τα οποία περιγράφονται πιο κάτω

2.4.1 Μοτίβο 1

Ξεκινούμε από τον κάτω αριστερά κόμβο και πάμε προς τον πάνω δεξιά προχωρώντας πρώτα στην κατεύθυνση χ και μετά στην ψ.

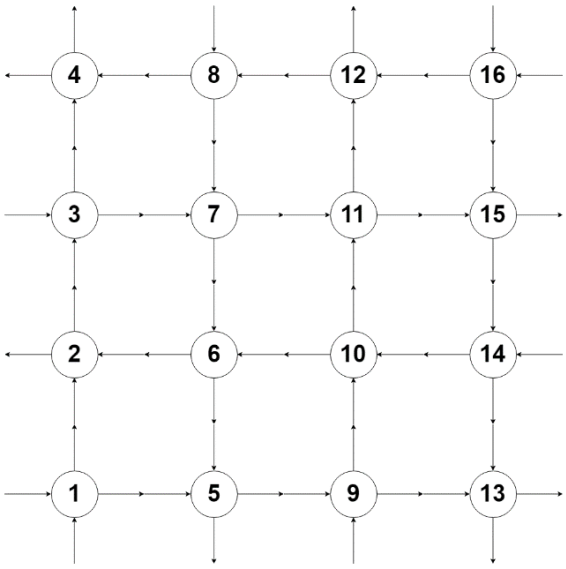


p.x=((int)packetCreated)%size;

p.y=((int)packetCreated)/size;

2.4.2 Μοτίβο 2

Ξεκινούμε από τον κάτω αριστερά κόμβο και πάμε προς τον πάνω δεξιά προχωρώντας πρώτα στην κατεύθυνση ψ και μετά στην χ.

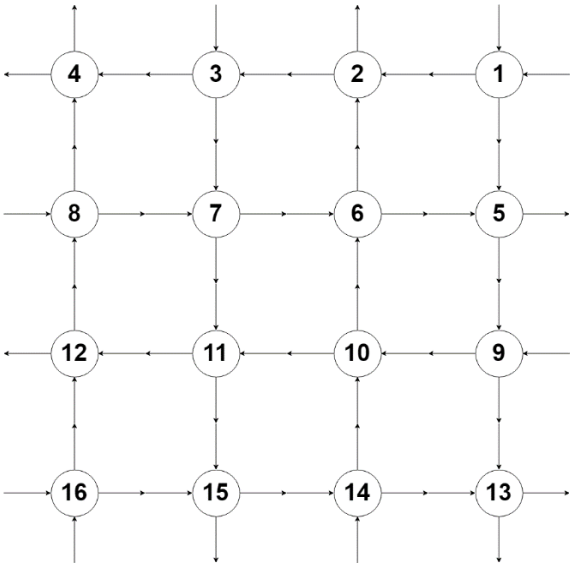


p.x=((int)packetCreated)/size;

p.y=((int)packetCreated)%size;

2.4.3 Μοτίβο 3

Ξεκινούμε από τον πάνω δεξιά κόμβο και πάμε προς τον κάτω αριστερά προχωρώντας πρώτα στην κατεύθυνση χ και μετά στην ψ.

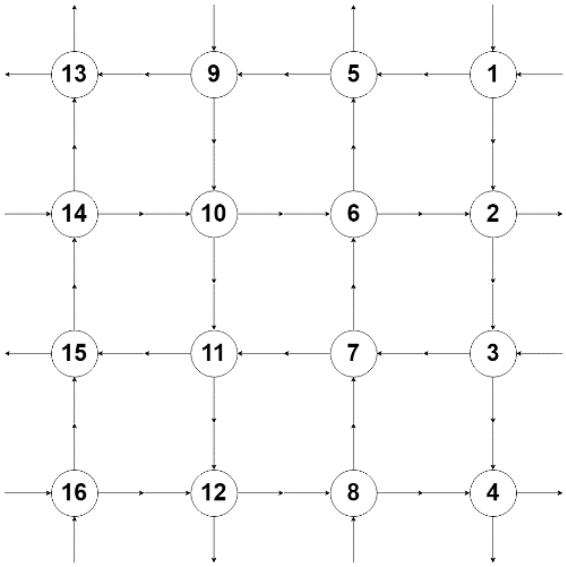


p.x=size-1-((int)packetCreated)%size;

p.y=size-1-((int)packetCreated)/size;

2.4.4 Μοτίβο 4

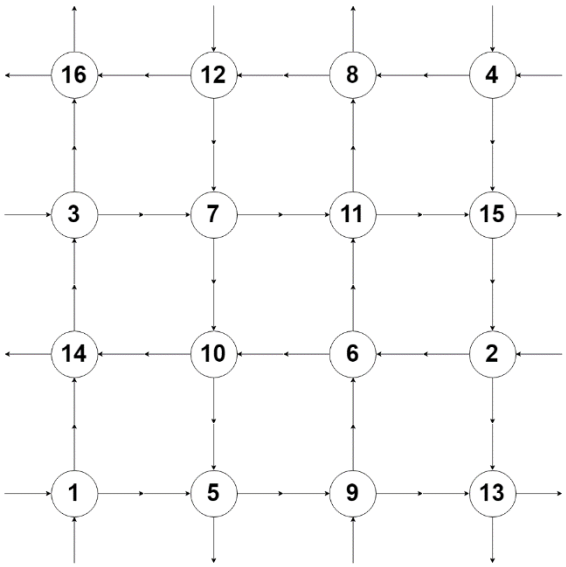
Ξεκινούμε από τον πάνω δεξιά κόμβο και πάμε προς τον κάτω αριστερά προχωρώντας πρώτα στην κατεύθυνση ψ και μετά στην χ.



```
p.x=size-1-((int)packetCreated)/size;
p.y=size-1-((int)packetCreated)%size;
```

2.4.5 Μοτίβο 5

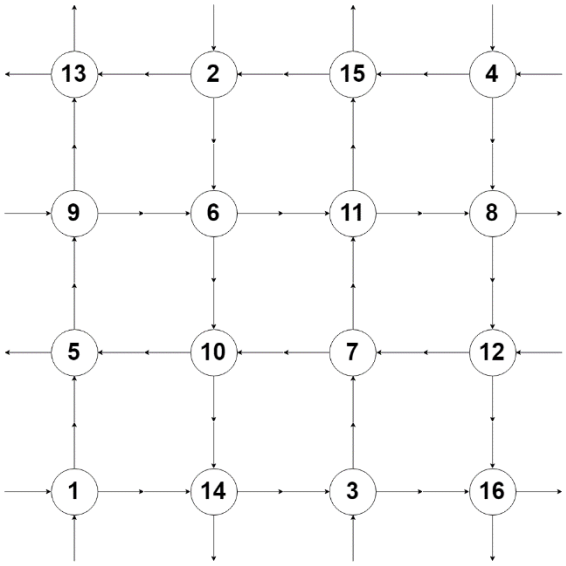
Ξεκινούμε από τον κάτω αριστερά κόμβο και πάμε προς τον πάνω αριστερά προχωρώντας πρώτα στην κατεύθυνση ψ και μετά στην χ. Στην κατεύθυνση χ πηγαίνουμε από έξω προς τα μέσα εναλλάξ και όταν φτάσουμε στο κέντρο από μέσα προς τα έξω, πάλι εναλλάξ.




```
if(packetCreated%2==0)
        p.x=((int)packetCreated)/size;
else
        p.x=size-1-((int)packetCreated)/size;
p.y=((int)packetCreated)%size;
```

2.4.6 Μοτίβο 6

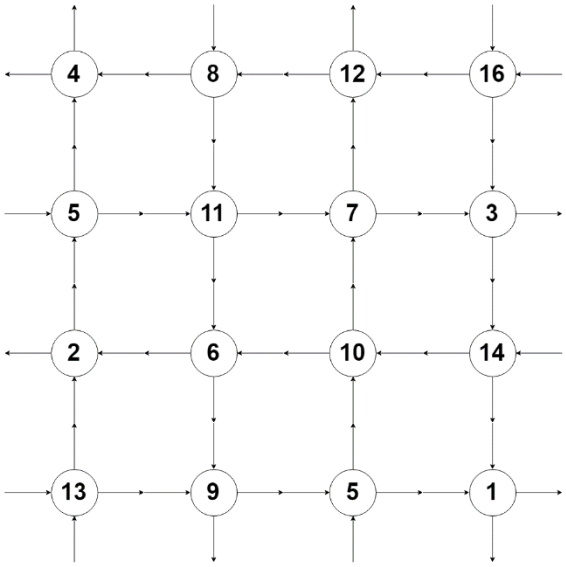
Ξεκινούμε από τον κάτω αριστερά κόμβο και πάμε προς τον κάτω δεξιά προχωρώντας πρώτα στην κατεύθυνση χ και μετά στην ψ. Στην κατεύθυνση ψ πηγαίνουμε από έξω προς τα μέσα εναλλάξ και όταν φτάσουμε στο κέντρο από μέσα προς τα έξω, πάλι εναλλάξ.




```
if(packetCreated%2==0)
        p.y=((int)packetCreated)/size;
else
        p.y=size-1-((int)packetCreated)/size;
p.x=((int)packetCreated)%size;
```

2.4.7 Μοτίβο 7

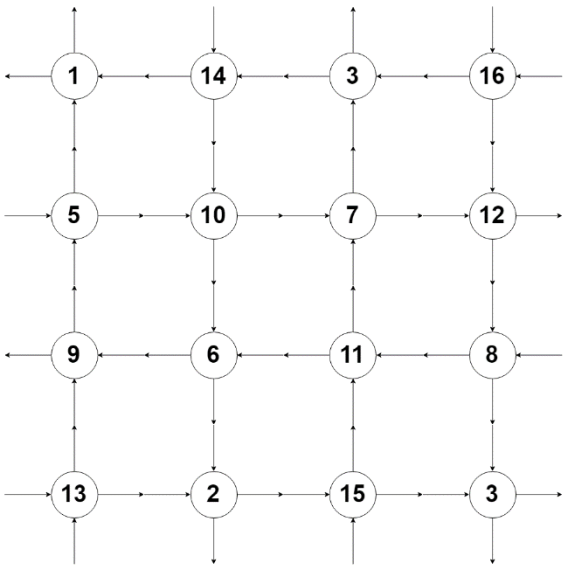
Ξεκινούμε από τον πάνω δεξιά κόμβο και πάμε προς τον πάνω αριστερά προχωρώντας πρώτα στην κατεύθυνση ψ και μετά στην χ. Στην κατεύθυνση χ πηγαίνουμε από έξω προς τα μέσα εναλλάξ και όταν φτάσουμε στο κέντρο από μέσα προς τα έξω, πάλι εναλλάξ.




```
if(packetCreated%2==1)
        p.x=((int)packetCreated)/size;
else
        p.x=size-1-((int)packetCreated)/size;
p.y=((int)packetCreated)%size;
```

2.4.8 Μοτίβο 8

Ξεκινούμε από τον πάνω αριστερά κόμβο και πάμε προς τον πάνω δεξιά προχωρώντας πρώτα στην κατεύθυνση χ και μετά στην ψ. Στην κατεύθυνση ψ πηγαίνουμε από έξω προς τα μέσα εναλλάξ και όταν φτάσουμε στο κέντρο από μέσα προς τα έξω, πάλι εναλλάξ.




```
if(packetCreated%2==1)
        p.y=((int)packetCreated)/size;
else
        p.y=size-1-((int)packetCreated)/size;
p.x=((int)packetCreated)%size;
```

ΚΕΦΑΛΑΙΟ 3

Αλγόριθμοι

3.1 XY before 12

3.2 XY after 13

3.3 Χρήση simulator σε Anylogic 14

3.3.1 Simulator Input Data 15

3.3.2 Simulator Output Data 16

3.3.2.1 Apotelesmata.txt 16

3.3.2.2 Route.txt 17

3.3.2.3 Steps.txt 17

3.1 XY before

Ο αλγόριθμος XY before κινείται πρώτα στον άξονα X και μετά στον άξονα Y. Όταν δεν επιτρέπεται η πρόσβαση στον κόμβο πηγαίνοντας ευθεία πάει από τον προηγούμενο συγκρίνει την τρέχουσα διεύθυνση με την διεύθυνση προορισμού που είναι αποθηκευμένη στην κεφαλίδα του πακέτου. Όταν είναι ίσα παραδίδεται το πακέτο. Αν όχι συγκρίνεται το Dx me to Cx, δηλαδή συγκρίνουμε πρώτα οριζόντια. Ο πιο κάτω αλγόριθμος αποδεδειγμένα δεν προκαλεί deadlock

Ψευδοκώδικάς:

```
if(node.x/2<p.x/2)
        sendRight(p, size,node);
else if(node.x > p.x && node.y + 1 >= p.y)
        sendLeft(p, size,node);
else if(node.y<p.y)
        sendUp(p, size,node);
else if(node.x<p.x){
        sendRight(p, size,node);
else
        sendDown(p, size,node);
```

3.2 XY after

Η φιλοσοφία αυτού του αλγόριθμου είναι ίδια με την φιλοσοφία του προηγούμενου. Η διαφορά τους είναι όταν είναι μονόδρομος από το μονοπάτι του κόμβου, πάει από το επόμενο και όχι από το προηγούμενο.

Για αυτό τον λόγο προκύπτει και deadlock

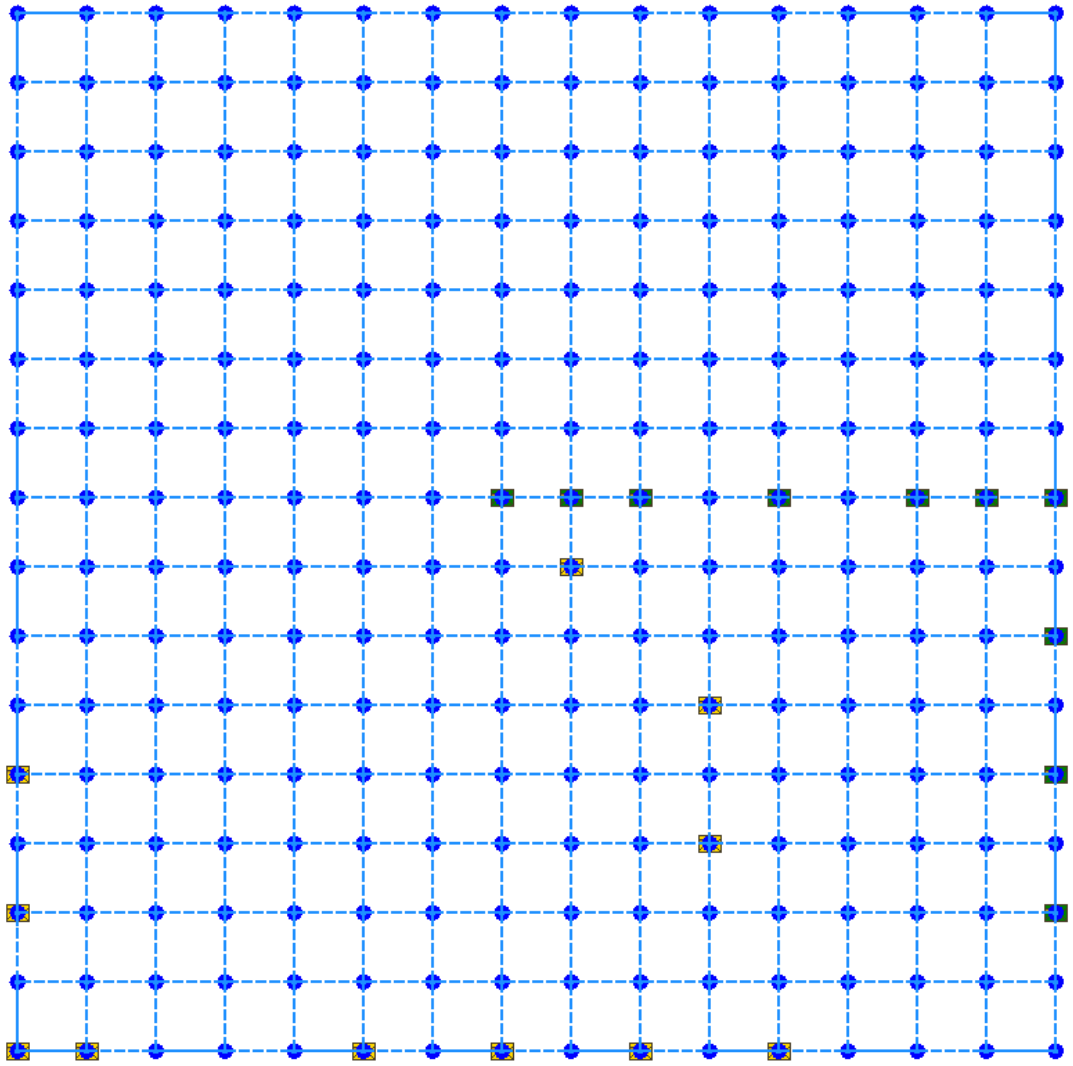
Ψευδοκώδικάς:

```
if (enough==false){
        if (node.x<p.x){
                if (p.x==23 && node.x==2) {
                        if (node.y<p.y)
                                sendUp(p, size,node);
                        else
                                sendRight(p, size,node);
                }
                else
                        sendRight(p, size,node);
        }
        else if (node.x==p.x+1){
                if (node.y<p.y)
                        sendUp(p, size,node);
                else
                        sendLeft(p, size,node);
        }
        else if (node.x>p.x+1)
                sendLeft(p, size,node);
        else if (node.x==p.x){
                if (node.y<p.y)
                        sendUp(p, size,node);
                else if (node.y==p.y)
```

```
                    sendRight(p, size,node);
            else if (node.y>p.y)
                    if (t==0 || t==2)
                            sendRight(p, size,node);
                    else sendDown(p, size,node);
        }
}
```

3.3 Χρήση Simulator Anylogic

Στο σχήμα 3, φαίνεται η προσομοίωση αποστολής πακέτων. Στο συγκεκριμένο σενάριο το μέγεθος του πλέγματος είναι 16, ο αλγόριθμος που χρησιμοποιείται είναι ο XY Before με μοτίβο 1 και το acknowledgment gateway βρίσκεται κάτω δεξιά. Με ανοικτό κίτρινο χρώμα είναι τα πακέτα που κατευθύνονται προς τον κόμβο παραλήπτη, ενώ με σκούρο κίτρινο χρώμα, τα μηνύματα acknowledgment.



3.3.1 Simulator Input Data

Όπως αναφέραμε και πιο πάνω στα διάφορα σενάρια που δοκιμάζουμε, αλλάζουμε κάποιες συγκεκριμένες παραμέτρους, όπως:

- αλγόριθμος δρομολόγησης (algorithm)
- θέση της πύλης μηνυμάτων επιβεβαίωσης (ackx και acky)
- μέγεθος του πλέγματος (size)
- μοτίβο αποστολής πακέτων σε όλους τους κόμβους (Reordering)

ενώ κάποιες άλλες παράμετροι παραμένουν σταθερές:

- θέση input gateway (GWy και GWx)

3.3.2 Simulator Output Data

Όπως επίσης έχουμε προαναφέρει, θέλουμε να εξάγουμε συμπεράσματα για τον αριθμό των hops, την ύπαρξη deadlocks, τον χρόνο προσομοίωσης, καθώς και ποιοι κόμβοι χρησιμοποιούνται περισσότερο. Για τον πιο πάνω λογο δημιουργούμε 3 αρχεία κατά την προσομίωση που το καθένα από αυτά περιλαμβάνει σημαντικές πληροφορίες:

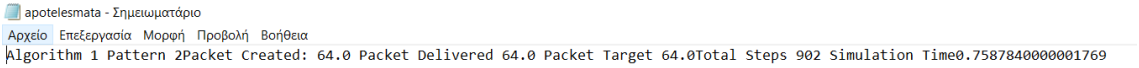
- Apotelesmata.txt
- Route.txt
- Steps.txt

3.3.2.1 Apotelesmata.txt

Το πρώτο αρχείο που δημιουργείται παρέχει πληροφορίες για:

- τον αλγόριθμο που χρησιμοποιήθηκε
- το μοτίβο που χρησιμοποιήθηκε
- τον αριθμό των πακέτων που αποστάλθηκαν
- τον αριθμό των πακέτων που παραλήφθηκαν επιτυχώς

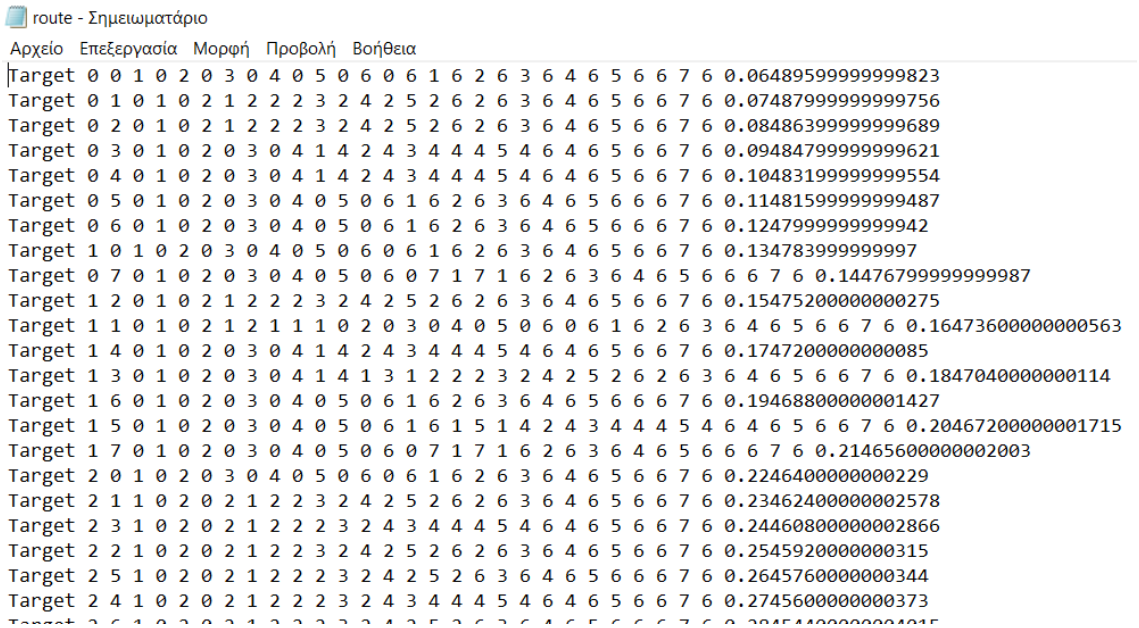
- τον συνολικό αριθμό των hops
- και το συνολικό χρόνο προσομοίωσης



3.3.2.2 Route.txt

Το δεύτερο αρχείο που δημιουργείται παρέχει πληροφορίες:

- την διευθυνση του κόμβου παραλήπτη
- τη διαδρομή που ακολουθηθηκε και
- τη στιγμή (simulation time) που παραλήφθηκε επιτυχώς το πακέτο



3.3.2.3 Steps.txt

Το τρίτο αρχείο που δημιουργείται παρέχει πληροφορίες για την χρήση του κάθε κόμβου ξεχωριστά:

steps - Σημειωματάριο

Αρχείο Επεξεργασία Μορφή Προβολή Βοήθεια

```
Node 0 0 Packets served : 0
Node 0 1 Packets served : 51
Node 0 2 Packets served : 51
Node 0 3 Packets served : 38
Node 0 4 Packets served : 38
Node 0 5 Packets served : 25
Node 0 6 Packets served : 25
Node 0 7 Packets served : 1
Node 1 0 Packets served : 14
Node 1 1 Packets served : 1
Node 1 2 Packets served : 14
Node 1 3 Packets served : 1
Node 1 4 Packets served : 14
Node 1 5 Packets served : 1
Node 1 6 Packets served : 26
Node 1 7 Packets served : 2
Node 2 0 Packets served : 14
```

ΚΕΦΑΛΑΙΟ 4

Μετρήσεις

4.1 Αποτελέσματα μετρήσεων 19

4.1 Αποτελέσματα μετρήσεων

A	P	S	X	Y	DEAD	PACKET	HOPS	SIMULATION TIME
1	1	8	7	0	NO	64	944	0,7288320000001680000000
1	1	8	7	6	NO	64	902	0,7488000000001740000000
1	1	24	23	0	NO	576	27024	6,0802560000104800000000
1	1	24	23	12	NO	576	23868	6,0303360000103200000000
1	1	24	23	22	NO	576	26518	6,1551360000107100000000
1	1	64	63	32	NO	4096	456608	41,6782080005246000000000
1	2	8	7	0	NO	64	944	0,7537920000001750000000
1	2	8	7	6	NO	64	902	0,7587840000001760000000
1	2	24	23	0	NO	576	27024	6,1052160000105500000000
1	2	24	23	12	NO	576	23868	6,1351680000106500000000
1	2	24	23	22	NO	576	26518	6,1950720000108400000000
1	3	8	7	0	NO	64	944	0,7288320000001680000000
1	3	8	7	6	NO	64	902	0,7438080000001720000000
1	3	24	23	0	NO	576	27024	6,0702720000104400000000
1	3	24	23	12	NO	576	23868	6,0552960000104000000000
1	3	24	23	22	NO	576	26518	6,2549760000110200000000
1	4	8	7	0	NO	64	944	0,7238400000001660000000
1	4	8	7	6	NO	64	902	0,7338240000001690000000
1	4	24	23	0	NO	576	27532	6,0952320000105200000000
1	4	24	23	12	NO	576	23868	5,9354880000100200000000
1	4	24	23	22	NO	576	26518	6,0053760000102400000000
1	4	32	31	0	NO	1024	64448	10,67289599996400000000
1	5	24	23	12	NO	576	23868	5,7957120000095800000000
1	5	32	31	0	NO	1024	64448	10,283519999716000000000
1	5	32	31	16	NO	1024	56784	10,278527999717000000000
1	5	64	63	32	NO	4096	456608	40,9693440005063000000000
1	6	64	63	32	NO	4096	456608	41,2838400005144000000000
1	7	32	31	16	NO	1024	56784	10,258559999721000000000
1	8	32	31	16	NO	1024	56784	1,000000000000000000000
2	1	8	7	0	YES			
2	1	8	7	6	NO	64	1118	0,8037120000001890000000
2	1	24	23	0	YES			
2	1	24	23	12	YES			
2	1	24	23	22	YES			

2	1	32	31	0	YES			
2	1	32	31	16	YES			
2	1	32	31	30	NO	1024	79660	10,9873919999578000000000
2	2	8	7	0	YES			
2	2	8	7	6	NO	64	1118	0,8885760000002140000000
2	2	24	23	0	NO	576	27532	7,2783360000142300000000
2	2	24	23	12	YES			
2	2	24	23	22	YES			
2	2	32	31	0	YES			
2	2	32	31	16	YES			
2	2	32	31	30	NO	1024	79660	13,1688959999149000000000
2	3	8	7	0	YES			
2	3	8	7	6	NO	64	1118	0,7987200000001880000000
2	3	24	23	0	NO	576	27532	6,1052160000105500000000
2	3	24	23	12	NO	576	25996	5,9454720000100500000000
2	3	24	23	22	NO	576	32306	6,0652800000104300000000
2	3	32	31	0	YES			
2	3	32	31	16	YES			1,0000000000000000000000
2	3	32	31	30	NO	1024	79660	10,8875519999597000000000
2	4	8	7	0	NO	64	1004	0,7288320000001680000000
2	4	8	7	6	NO	64	1118	0,7987200000001880000000
2	4	24	23	0	NO	576	27532	6,0952320000105200000000
2	4	24	23	12	NO	576	25996	6,0602880000104100000000
2	4	24	23	22	NO	576	32306	6,2949120000111500000000
2	4	32	31	0	NO	1024	65770	10,6828799999638000000000
2	4	32	31	16	NO	1024	62426	10,6828799999638000000000
2	4	32	31	30	NO	1024	79660	10,9624319999583000000000
2	5	8	7	0	YES			
2	5	24	23	0	NO	576	27532	6,0552960000104000000000
2	5	24	23	12	NO	576	25996	5,9854080000101800000000
2	5	24	23	22	YES			
2	5	32	31	0	YES			
2	5	32	31	16	YES			
2	5	32	31	30	NO	1024	79660	10,9175039999592000000000
2	6	8	7	0	YES			
2	6	24	23	0	NO	576	27532	6,0253440000103000000000
2	6	24	23	12	YES			
2	6	24	23	22	YES			
2	6	32	31	0	YES			
2	6	32	31	16	YES			
2	6	32	31	30	YES			
2	7	8	7	0	YES			
2	7	24	23	0	NO	576	27532	6,0602880000104100000000
2	7	24	23	12	YES			
2	7	24	23	22	YES			
2	7	32	31	0	YES			

2	7	32	31	16	YES			
2	7	32	31	30	NO	1024	79660	10,9224959999591000000000
2	8	8	7	0	YES			
2	8	24	23	0	NO	576	27532	6,0952320000105200000000
2	8	24	23	12	NO	576	25996	6,0103680000102600000000
2	8	24	23	22	YES			
2	8	32	31	0	YES			
2	8	32	31	16	YES			
2	8	32	31	30	NO	1024	79660	11,0223359999571000000000
2	8	32	31	30	NO	1024	79660	11,0223359999714000000000

ΚΕΦΑΛΑΙΟ 5

Συμπεράσματα

5.1 Εισαγωγή 22

5.2 Number of hops 22

5.3 Μέγεθος πλέγματος 23

5.4 Θέση acknowledgment gateway 23

5.5 Αλγόριθμος 23

5.6 Χρήση κόμβων 23

5.7 Μοτίβα αποστολής πακέτων 23

5.8 Ύπαρξη deadlock 24

5.9 Simulation time 27

5.10 Μελλοντική Εργασία 27

5.1 Εισαγωγή

Στα συμπεράσματα θα δούμε ξεχωριστά τις μετρικές που αναφέραμε πιο πάνω, ότι μας ενδιαφέρουν, και ανάλογα με τα αποτελέσματα που είδαμε στο προηγούμενο κεφάλαιο θα εξάγουμε κάποια συμπεράσματα. Θα εξεταστεί επίσης το ενδεχόμενο συνδυασμός διαφορετικών παραμέτρων να μας βοηθά να εξάγουμε κάποια συμπεράσματα.

5.2 Number of hops

- Θέση acknowledgment gateway
    - Με αλγόριθμο XY before πιο καλά αποτελέσματα acky=size-2
    - Με αλγόριθμο XY after πιο καλά αποτελέσματα acky=0
- Μέγεθος πλέγματος
    - Αναλογική Σχέση
- Μοτίβα αποστολής πακέτων

    - Μόνο στην περίπτωση A1P4S24X23Y0 βλέπουμε διαφορετικά αποτελέσματα
- Αλγόριθμος
    - Πιο καλά αποτελέσματα XY before

5.3 Μέγεθος πλέγματος

Δεν συμπεραίνουμε κάτι για πιο μεγάλο size της τοπολογίας εκτός από τον χρόνο που χρειάζεται και τον αριθμό των hops.

5.4 Θέση acknowledgment gateway

Από τις μετρήσεις φαίνεται ότι είναι καλύτερα όταν βρίσκεται στην μέση του πλέγματος και στην συνέχεια όταν βρίσκεται στην κάτω δεξιά γωνία. Στην περίπτωση που βρίσκεται στην κάτω αριστερή γωνία δημιουργείται deadlock με όλα τα μοτίβα.

5.5 Αλγόριθμος

Ο XY before έχει καλύτερα αποτελέσματα από τον XY after, τόσο σε θέμα χρόνου, όσο και σε θέμα ύπαρξης deadlock. Ο XY before αποδεδειγμένα δεν μπαίνει ποτέ σε deadlock

5.6 Χρήση κόμβων

Οι κόμβοι που είναι κοντά στα gateways είναι οι πιο χρησιμοποιημένοι. Επίσης οι κόμβοι που έχουν διεύθυνση τόσο στο χ-άξονα όσο και στον ψ-άξονα μονό αριθμό δεν χρησιμοποιούνται σχεδόν καθόλου.

5.7 Μοτίβα αποστολής πακέτων

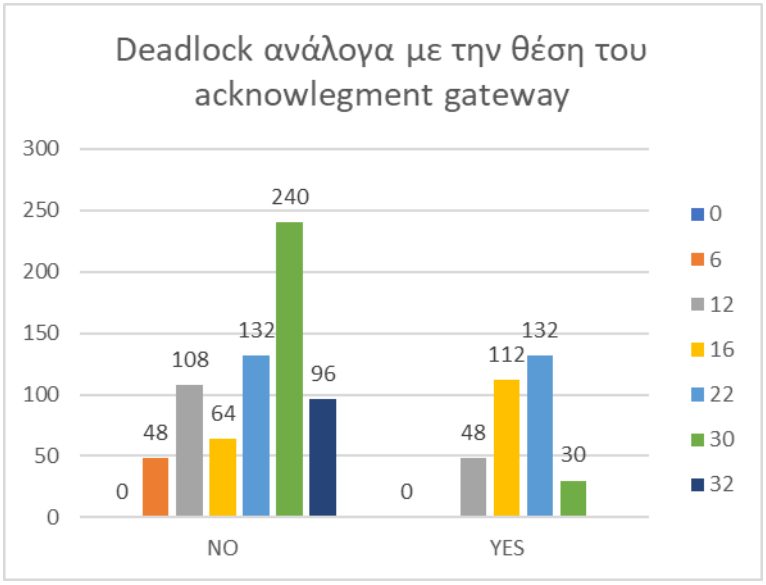
- Το μοτίβο 4 έχει τα καλύτερα αποτελέσματα

- Έχει γρήγορο χρόνο διεκπεραίωσης
- Φαίνεται να αποφεύγει τα deadlocks

- Και το μοτίβο 5 παρουσιάζει αρκετά καλά αποτελέσματα
- Κακά αποτελέσματα παρουσιάζουν τα μοτίβα 8 και 2
- Η θέση του ack gateway εξαρτάται από τον αλγόριθμο για καλύτερα αποτελέσματα

5.8 Ύπαρξη deadlock

- Θέση acknowledgment gateway

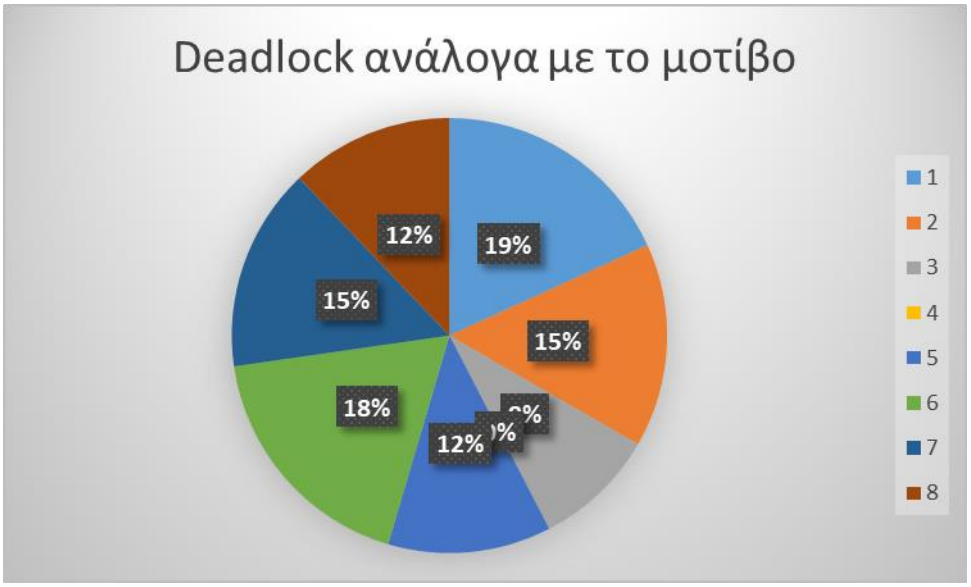


- 45,45% όταν το acky=0
- 21,21% όταν το acky=size-2 και
- 33,33% όταν το acky= size/2

- Μέγεθος πλέγματος

- 21,21% για size 8
- 33,33% για size 24 και,
- 45,45% για size 32

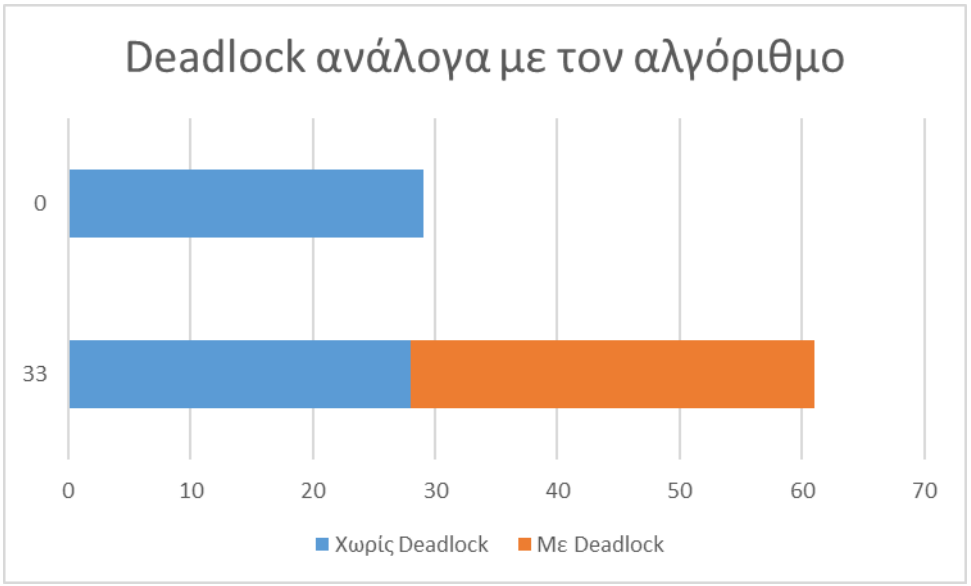
- Μοτίβα αποστολής πακέτων



- 18,18% Μοτίβο 1
- 15,15% Μοτίβο 2
- 09,09% Μοτίβο 3
- 00,00% Μοτίβο 4
- 12,12% Μοτίβο 5
- 18,18% Μοτίβο 6
- 15,15% Μοτίβο 7

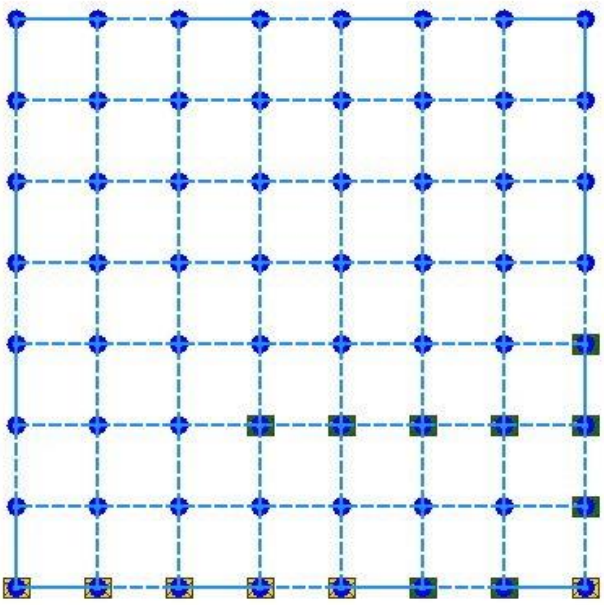
- 12,12% Μοτίβο 8

- Αλγόριθμος



- Ο αλγόριθμος XY before δεν προκαλεί deadlock
- 54,10% με αλγόριθμο XY after

Στο πιο κάτω σχήμα παρουσιάζεται ύπαρξη deadlock στο σενάριο με size=8, algorithm=2, reordering=1, ackx=0, acky=0.



Στο πιο πάνω σχήμα έχουν δημιουργηθεί τα πρώτα 27 πακέτα (το τελευταίο έχει προορισμό τον κόμβο 2,3) και δημιουργήθηκε deadlock. Όπως φαίνεται και στο σχήμα, 9 πακέτα έφτασαν στον κόμβο παραλήπτη και κατευθύνονται προς τον acknowledgment gateway, ενώ 6 πακέτα κατευθύνονται στον κόμβο παραλήπτη. Από τα πιο πάνω συμπεραίνουμε ότι 12 πακέτα έχουν παραδοθεί κανονικά. Η δημιουργία deadlock οφείλεται στο ότι τα πακέτα που βρίσκονται στο 6,0 και 7,1 κατευθύνονται προς το acknowledgment gateway (7,0) και εμποδίζονται από το πακέτο που βρίσκεται στο 7,0 που ακόμα δεν έχει παραδοθεί.

Γενικά για τη δημιουργία deadlock ρόλο παίζουν:

- Ο αλγόριθμος
- Η θέση του acknowledgment gateway και
- Το μοτίβο

5.9 Simulation time

- Θέση acknowledgment gateway
    - Κάνει την πιο λίγη διαφορά
- Μέγεθος πλέγματος
    - Αναλογική σχέση
- Μοτίβα αποστολής πακέτων
    - Σειρά κατάταξης
        - Μοτίβο 4
        - Μοτίβο 3
        - Μοτίβο 1
        - Μοτίβο 1
- Αλγόριθμος
    - Πάρα πολύ κοντινά αποτελέσματα

5.10 Μελλοντική εργασία

Στο μέλλον μπορεί να γίνει μελέτη για διαφορετικούς αλγόριθμους και διαφορετικά μοτίβα αποστολής πακέτων σε όλους τους κόμβους

Βιβλιογραφία

[1] Taqwa Saeed, Constantinos Skitsas, Dimitrios Kouzapas, Marios Lestas, Vassos Soteriou, Anna Philippou, Sergi Abadal, Christos Liaskos, Loukas Petrou, Julius Georgiou, and Andreas Pitsillides, Fault Adaptive Routing in Metasurface Controller Networks (Invited Paper),2018.

[2] Ηλίας Πηλακούτας, Ανάπτυξη, υλοποίηση και πειραματική αξιολόγηση αλγόριθμου δρομολόγησης με ανοχή σε σφάλματα, σε τοπολογία Manhattan, 2018.

[3] Design of the HyperSurface Controller Network.

[4] Visorsurf: A hardware platform for software-driven functional metasurfaces. http://www.visorsurf.eu/.

[5] Panagiotis Kouvaros, Dimitrios Kouzapas, Anna Philippou, Julius Georgiou, Loukas Petrou and Andreas Pitsillides, Formal Verification of a Programmable Hypersurface,2018