

Ατομική Διπλωματική Εργασία

**ΒΕΛΤΙΣΤΟΤΗΤΑ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ΤΟΥ ΜΕΣΟΥ ΟΡΟΥ
ΓΙΑ ΤΟΝ ΣΥΓΧΡΟΝΙΣΜΟ ΡΟΛΟΓΙΩΝ ΣΕ ΔΙΚΤΥΑΚΕΣ
ΤΟΠΟΛΟΓΙΕΣ**

Άντρη Χατζηπαναγή

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2017

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΒΕΛΤΙΣΤΟΤΗΤΑ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ΤΟΥ ΜΕΣΟΥ ΟΡΟΥ
ΓΙΑ ΤΟΝ ΣΥΓΧΡΟΝΙΣΜΟ ΡΟΛΟΓΙΩΝ ΣΕ ΔΙΚΤΥΑΚΕΣ
ΤΟΠΟΛΟΓΙΕΣ

Αντρη Χατζηπαναγή

Επιβλέπων Καθηγητής
Μάριος Μαυρονικόλας

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

Μάιος 2017

Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας, σηματοδοτείται συνάμα το τέλος της πορείας μου στο Πανεπιστήμιο Κύπρου, ως προπτυχιακή φοιτήτρια του Τμήματος Πληροφορικής.

Θα ήθελα πρωτίστως να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κύριο Μάριο Μαυρονικόλα για την εμπιστοσύνη αλλά και για τη συνεχή καθοδήγηση του αφιερώνοντας πολύτιμο χρόνο κατά τη διάρκεια της πορείας μου, προς την πραγμάτευση της εργασίας αυτής. Θα ήθελα επίσης, να ευχαριστήσω την οικογένεια μου για την αμέριστη συμπαράσταση και βοήθεια που μου προσφέρουν όλα αυτά τα χρόνια.

Περίληψη

Ο συγχρονισμός των ρολογιών των επεξεργαστών είναι ένα από τα βασικά θέματα που ερευνά η επιστήμη της Πληροφορικής εφόσον ποικίλες είναι οι εφαρμογές σε ένα κατακευματισμένο σύστημα που στηρίζονται στον συγχρονισμό αυτό. Στην παρούσα διπλωματική εργασία, πρωτίστως θα αναλυθούν σημαντικές έννοιες και ορισμοί για το θέμα του συγχρονισμού ρολογιών των επεξεργαστών. Επιπρόσθετα, θα χρησιμοποιηθεί ο αλγόριθμος μέσου όρου ο οποίος θα δείξω ότι είναι βέλτιστος για τις τοπολογίες Δακτύλιος με χορδές, τοπολογία Σκάλα και τοπολογία Ανεμιστήρα, δίνοντας το αυστηρό κάτω φράγμα $\frac{\text{διάμετρος}}{2}$ σε σχέση με τις αβεβαιότητες που υπάρχουν στις καθυστερήσεις μηνυμάτων.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Ορισμός προβλήματος	1
	1.2 Στόχος	2
	1.3 Ανασκόπηση κεφαλαίων	3
Κεφάλαιο 2	Πλαίσιο Εργασίας.....	5
	2.1 Ορισμοί και έννοιες	5
	2.2 Αβεβαιότητα στην καθυστέρηση μηνυμάτων	6
	2.3 Αλγόριθμος μέσου όρου	8
Κεφάλαιο 3	Συγχρονισμός Ρολογιών Σε Δακτύλιο Με Χορδές.....	9
	3.1 Τοπολογία	9
	3.2 Ελάχιστη απόσταση	10
	3.3 Διάμετρος	14
	3.4 Αλγόριθμος	16
	3.5 Ανάλυση αλγορίθμου	17
Κεφάλαιο 4	Συγχρονισμός Ρολογιών Σε Τοπολογία Ανεμιστήρα.....	21
	4.1 Τοπολογία και γενικά χαρακτηριστικά	21
	4.2 Αλγόριθμος	22
	4.3 Ανάλυση αλγορίθμου	23
Κεφάλαιο 5	Συγχρονισμός Ρολογιών σε τοπολογία σκάλα.....	27
	5.1 Τοπολογία και γενικά χαρακτηριστικά	27
	5.2 Αλγόριθμος	28
	5.3 Ανάλυση αλγορίθμου	29
Κεφάλαιο 6	Συμπεράσματα	33

6.1	Αυστηρό φράγμα	33
6.2	Αποτελέσματα	34
6.3	Μελλοντική Εργασία	34

Βιβλιογραφία		35
---------------------	-------	-----------

Κεφάλαιο 1

Εισαγωγή

1.1 Ορισμός προβλήματος	1
1.2 Στόχος	2
1.3 Ανασκόπηση κεφαλαίων	3

1.1 Ορισμός προβλήματος

Ο συγχρονισμός του χρόνου είναι ένα από τα όλο και πιο κρίσιμα κομμάτια του τομέα της Πληροφορικής. Οι εταιρίες χρηματοοικονομικών εμπορικών συναλλαγών, τα ηλεκτρονικά παιχνίδια, συστήματα ηλεκτρικής ενέργειας και μεγάλες κατανεμημένες βάσεις δεδομένων είναι μερικά από τα παραδείγματα στα οποία ο συγχρονισμός ρολογιών όχι μόνο είναι σημαντικός αλλά είναι και απαραίτητος για την ομαλή λειτουργία τους.

Η πολυπλοκότητα του σύγχρονου συγχρονισμού των ρολογιών σε ένα κατανεμημένο σύστημα προέρχεται από την ακρίβεια την οποία χρειάζονται οι πλείστες εφαρμογές εφόσον σε κάποιες από αυτές τα ρολόγια θα πρέπει να είναι συγχρονισμένα με λοξότητα τουλάχιστον μερικά εκατοντάδες δισεκατομμύρια του δευτερολέπτου. Επιπρόσθετα, κάποια άλλα εμπόδια στον συγχρονισμό αυτό θεωρούνται τόσο οι καθυστερήσεις των μηνυμάτων όσο και η αξιοπιστία του δικτύου. Οι καθυστερήσεις των μηνυμάτων προέρχονται κυρίως από το γεγονός ότι η απόσταση η οποία θα διανύσει ένα μήνυμα από ένα επεξεργαστή σε ένα άλλο μπορεί να μην είναι ένα απλό καλώδιο αλλά αυτό το

μήνυμα να πρέπει να περάσει διαμέσου πολλών εκατομμυρίων, ενδιάμεσων επεξεργαστών.

Σε πολλές περιπτώσεις είναι αρκετό οι επεξεργαστές να έχουν συμφωνήσει τον ίδιο χρόνο. Αυτός ο χρόνος δεν είναι απαραίτητο να είναι απόλυτα ίδιος με την πραγματική, φυσική, ώρα, αλλά αυτό που μας ενδιαφέρει είναι η εσωτερική συνέπεια του χρόνου. Αυτή η κατάσταση ορίζει τα λογικά ρολόγια ενώ εάν θέλουμε να προσθέσουμε και τον περιορισμό να μην αποκλίνουν ως μία συγκεκριμένη τιμή από τον πραγματικό χρόνο τότε τα ρολόγια αυτά καλούνται φυσικά.

Πρόβλημα αποτελεί το γεγονός ότι τα λογικά ρολόγια των επεξεργαστών δεν είναι κάτω από τον έλεγχο των ίδιων των επεξεργαστών. Για αυτό τον λόγο, είναι απαραίτητη η χρησιμοποίηση μιας ειδικής κατάστασης την οποία θα μπορούν να επεξεργάζονται οι επεξεργαστές έτσι ώστε τα τελικά προσαρμοσμένα τους ρολόγια να είναι μία συνάρτηση της κατάστασης αυτής και των λογικών τους ρολογιών.

1.2 Στόχος

Με δεδομένα τα παραπάνω δεν εκπλήσσει το γεγονός ότι έχουν αφιερωθεί πολλά άρθρα και αλγόριθμοι για το σημαντικό αυτό κεφάλαιο του συγχρονισμού των ρολογιών. Στόχος είναι να βρεθεί βέλτιστος αλγόριθμος ο οποίος με δεδομένα την μορφή της τοπολογίας και τις διάφορες αβεβαιότητες στις καθυστερήσεις των μηνυμάτων να έχει ως αποτέλεσμα την μικρότερη, κατορθωτή λοξότητα ανάμεσα στα ρολόγια των επεξεργαστών του δικτύου. Πιο συγκεκριμένα, με δεδομένο ένα δίκτυο επικοινωνίας G , σε κάθε αποδεκτή εκτέλεση του αλγορίθμου υπάρχει ένας πραγματικός χρόνος t_f κατά τον οποίο ο αλγόριθμος τερματίζει και για όλους τους επεξεργαστές p_i και p_j και όλους τους χρόνους $t \geq t_f$ ισχύει ότι $|AC_i(t) - AC_j(t)| \leq \epsilon$, όπου $AC(t)$ είναι τα προσαρμοσμένα ρολόγια των επεξεργαστών τα οποία διαμορφώθηκαν από τον αλγόριθμο. Παράλληλα, η τιμή ϵ είναι η μικρότερη δυνατή και έτσι ορίζουμε ως $opt(G)$ την μετατροπή του δικτύου G στην οποία υπάρχουν ϵ -συγχρονισμένα ρολόγια.

Ο αλγόριθμος μέσου όρου, είναι ο απλούστερος αλγόριθμος ο οποίος έχει χρησιμοποιηθεί για τον συγχρονισμό των τοπικών ρολογιών των επεξεργαστών. Κύριο χαρακτηριστικό του αλγορίθμου αυτού είναι ότι όλοι οι επεξεργαστές του δικτύου εφόσον πρώτα λάβουν από όλους τους υπόλοιπους επεξεργαστές ένα μήνυμα το οποίο περιλαμβάνει την τιμή του ρολογιού τους στην συνέχεια καλούνται να προσαρμόσουν το ρολόι τους στον μέσο όρο των χρόνων αυτών σε σχέση βέβαια με την καλύτερη εκτίμηση για καθυστέρηση που θα βρεθεί σε κάθε τοπολογία.

Στόχος λοιπόν αυτής της εργασίας είναι η απόδειξη ότι για τις τοπολογίες Δακτύλιος με χορδές, Ανεμιστήρα και Σκάλα ο αλγόριθμος μέσου όρου είναι βέλτιστος με το κάτω φράγμα συγχρονισμού που είναι $\frac{\text{διάμετρος}}{2}$ σε σχέση με τις αβεβαιότητες καθυστερήσεις μηνυμάτων στους συνδέσμους να είναι αυστηρό.

1.3 Ανασκόπηση κεφαλαίων

Στο πρώτο κεφάλαιο έχω ορίσει το πρόβλημα του συγχρονισμού ρολογιών στο οποίο βασίζεται η εργασία αυτή. Στην συνέχεια παρουσίασα τον βασικό στόχο ο οποίος είναι η κατασκευή βέλτιστων, κατανεμημένων αλγορίθμων και τέλος έκανα ανασκόπηση των κεφαλαίων όπου θα παρουσιαστούν στην συνέχεια αυτής της εργασίας.

Στο δεύτερο κεφάλαιο, αρχικά αναλύονται οι ορισμοί και οι έννοιες που θα χρησιμοποιηθούν και στα επόμενα κεφάλαια του βιβλίου. Στην συνέχεια παρουσιάζεται η έννοια της αβεβαιότητας στις καθυστερήσεις των μηνυμάτων εφόσον είναι ένα κύριο κριτήριο για τον συγχρονισμό που θέλουμε να επιτευχθεί. Επιπρόσθετα αναλύεται ο αλγόριθμος μέσου όρου (averaging algorithm) τον οποίο έχω χρησιμοποιήσει και ορίζονται τόσο το άνω όσο και το κάτω φράγμα του.

Στο τρίτο κεφάλαιο, αρχικά παρουσιάζονται τα χαρακτηριστικά της τοπολογίας δακτυλίου με χορδές (chordal ring topology) και στο τέλος παρουσιάζεται ο αλγόριθμος για συγχρονισμό ρολογιών της τοπολογίας αυτής όπως και η ανάλυση του.

Στο τέταρτο κεφάλαιο, παρουσιάζεται η τοπολογία ανεμιστήρας (fan topology). Στην συνέχεια εφαρμόζεται και σε αυτή την τοπολογία ο αλγόριθμος μέσου όρου και η ανάλυση του.

Στο πέμπτο κεφάλαιο, παρουσιάζονται τα χαρακτηριστικά μίας ακόμη τοπολογίας, την σκάλα (ladder topology). Και σε αυτό το κεφάλαιο θα εφαρμοστεί ο ίδιος αλγόριθμος Και θα δοθεί η ανάλυση του.

Στο έκτο κεφάλαιο, συνοψίζονται τα αποτελέσματα και από τις τρεις τοπολογίες και παρουσιάζεται ένα γενικό αποτέλεσμα για το συγχρονισμό ρολογιών με την χρησιμοποίηση του κατανεμημένου αλγορίθμου μέσου όρου.

Κεφάλαιο 2

Πλαίσιο Εργασίας

2.1 Ορισμοί και Έννοιες	5
2.2 Αβεβαιότητα στην καθυστέρηση μηνυμάτων	6
2.3 Αλγόριθμος μέσου όρου	8

2.1 Ορισμοί και Έννοιες

Οι ορισμοί και οι έννοιες που ακολουθούν είναι βασισμένοι στο βιβλίο [1] και στα άρθρα [2], [3], [4] και [5].

Θεωρούμε ένα σύνολο από n επεξεργαστές (με δείκτες από p_0 ως p_{n-1}), οι οποίοι συνδέονται μέσω point-to-point συνδέσεις. Η τοπολογία παρουσιάζεται με ένα μη-κατευθυνόμενο γράφημα $G=(V,E)$ όπου οι κόμβοι είναι οι επεξεργαστές και οι ακμές είναι οι συνδέσεις μεταξύ τους. Κάθε σύνδεσμος (link) μεταξύ p_i και p_j παρέχει αξιόπιστα μηνύματα από το p_i στο p_j ενώ παράλληλα ενδέχεται η ύπαρξη αυθαίρετων καθυστερήσεων οι οποίες συμβολίζονται ως d . Επιπρόσθετα, κάθε σύνδεσμος έχει μία αβεβαιότητα σχετικά με την καθυστέρηση των μηνυμάτων η οποία συμβολίζεται με u .

Θεωρούμε ότι ο πραγματικός χρόνος (real time) ή τουλάχιστον η προσέγγιση του είναι διαθέσιμος στους επεξεργαστές. Ο μηχανισμός με τον οποίο ο χρόνος αυτός γίνεται διαθέσιμος στους επεξεργαστές είναι το hardware clock, μία αύξουσα συνάρτηση HC_i από μη-αρνητικούς πραγματικούς αριθμούς σε μη-αρνητικούς πραγματικούς αριθμούς. Συγκεκριμένα, όταν ένας επεξεργαστής p_i εκτελεί ένα υπολογιστικό βήμα κατά τον πραγματικό χρόνο t , η συνάρτηση HC_i είναι διαθέσιμη ως μέρος της εισόδου της συνάρτησης μετάβασης (transition) του επεξεργαστή p_i μεταφέροντας της τον τωρινό πραγματικό χρόνο.

Υποθέτουμε ότι κάθε p_i επεξεργαστής έχει μια ειδική κατάσταση οντότητας adj_i η οποία χρησιμοποιείται για να ρυθμίσει την τιμή του προσαρμοσμένου του ρολογιού.

Μετά την εκτέλεση του αλγορίθμου για συγχρονισμό ρολογιών, οι επεξεργαστές λειτουργούν πλέον τα προσαρμοσμένα τους ρολόγια που συμβολίζονται ως AC_i και έτσι μπορούμε τώρα να ορίσουμε με ακρίβεια το πρόβλημα του συγχρονισμού ρολογιών. Ορίζουμε το προσαρμοσμένο ρολόι του επεξεργαστή $AC_i(t)$ να είναι ίσο με $HC_i(t) + adj_i(t)$

2.2 Αβεβαιότητα στην καθυστέρηση μηνυμάτων

Η αβεβαιότητα u ενός συνδέσμου ορίζεται ως την διαφορά της ελάχιστης από την μέγιστη καθυστέρηση για να σταλθεί ένα μήνυμα από τον επεξεργαστή του ενός άκρου του συνδέσμου μέχρι να παραληφθεί από τον επεξεργαστή στο άλλο άκρο και η τιμή της είναι σίγουρα $u \leq d$, όπου d είναι η μέγιστη καθυστέρηση του μηνύματος. Πιο συγκεκριμένα και όπως αναφέρεται στο [2], υπάρχουν δύο συναρτήσεις $L(i,j)$ και $H(i,j)$. Η $L(i,j)$ δίνει την ελάχιστη καθυστέρηση ενώ το $H(i,j)$ δίνει την μέγιστη καθυστέρηση ενός μηνύματος ανάμεσα στον επεξεργαστή p_i και p_j . Συνεπώς η αβεβαιότητα στην καθυστέρηση του μηνύματος είναι ίση με $H(i,j) - L(i,j)$.

Στην παρούσα εργασία, θα χρησιμοποιήσω τον ορισμό της αβεβαιότητας όπως παρουσιάζεται στο άρθρο [1]. Όπως αναφέρεται στο άρθρο αυτό, η καθυστέρηση του μηνύματος είναι η ώρα ανάμεσα στο υπολογιστικό βήμα όταν στέλνεται το μήνυμα και το υπολογιστικό βήμα όταν ο παραλήπτης επεξεργάζεται το μήνυμα. Όταν ο επεξεργαστής p_i στείλει ένα μήνυμα m στον επεξεργαστή p_j την πραγματική ώρα t , τότε αυτό θα πρέπει να παραληφθεί όχι αργότερα από το χρόνο $t + d$ και όχι νωρίτερα από το χρόνο $t + d - u$. Η πρώτη ιδέα ήταν όταν ο επεξεργαστής παραλήπτης παραλάβει το μήνυμα από τον επεξεργαστή αποστολέα να θέσει το προσαρμοσμένο του ρολόι να είναι $T + (d - u)$, όπου T είναι η τιμή του μηνύματος. Η καλύτερη απόδοση του αλγορίθμου, βασισμένη σε αυτή την ιδέα είναι όταν το μήνυμα έχει πράγματι καθυστέρηση $d - u$ και έτσι η λοξότητα ανάμεσα στα προσαρμοσμένα τους ρολόγια θα ήταν 0. Από την άλλη η χειρότερη περίπτωση στην απόδοση του αλγορίθμου θα ήταν

όταν το μήνυμα είχε καθυστέρηση d , διότι η υπόθεση ήταν ότι το μήνυμα θα καθυστερούσε μόνο $d - u$ μονάδες χρόνου και αυτό κάνει την λοξότητα να είναι σε αυτή την περίπτωση d . Μία άλλη ιδέα θα ήταν ο επεξεργαστής παραλήπτης να θέσει το προσαρμοσμένο του ρολόι να είναι $T + d$, όπως αυτό δεν λύνει το πρόβλημα εφόσον τώρα η λοξότητα είναι και πάλι d όταν η πραγματική του καθυστέρηση είναι $d - u$. Όπως είναι φανερό από τις δύο παραπάνω ιδέες, η δυσκολία του συγχρονισμού ρολογιών είναι η διαφορά ανάμεσα στην προσέγγιση της καθυστέρησης και την πραγματική καθυστέρηση του μηνύματος. Τελικά, το θέμα της προσέγγισης της καθυστέρησης κλείνει θέτοντας την να είναι $d - \frac{u}{2}$ και αποδεικνύεται ότι είναι η καλύτερη προσέγγιση που μπορεί να επιλεγεί κάνοντας την λοξότητα του συγχρονισμού να είναι $\frac{u}{2}$. Η απόδειξη του είναι απλή. Έστω ότι η καθυστέρηση του μηνύματος είναι x τότε $d - u \leq x \leq d$ και έτσι $|x - (d - \frac{u}{2})| \leq \frac{u}{2}$. Επίσης, σύμφωνα με το [1] η ίδια λοξότητα επιτυγχάνεται είτε η καθυστέρηση και η αβεβαιότητα είναι άγνωστες είτε είναι γνωστές στον αλγόριθμο.

Στηριζόμενη στα παραπάνω, εφόσον η καθυστέρηση του μηνύματος πάνω σε ένα σύνδεσμο είναι $d - \frac{u}{2}$, θα αναλύσω την περίπτωση όπου δύο επεξεργαστές θέλουν να ανταλλάξουν μηνύματα αλλά δεν είναι απευθείας συνδεδεμένοι με έναν μόνο σύνδεσμο. Σε αυτή την περίπτωση θα λάβει σημαντικό ρόλο η ελάχιστη απόσταση (δηλαδή ο ελάχιστος αριθμός ακμών) ανάμεσα στους δύο επεξεργαστές. Εφόσον για να διανύσει έναν σύνδεσμο ένα μήνυμα υπάρχει καθυστέρηση $d - \frac{u}{2}$, τότε η ελάχιστη καθυστέρηση που μπορεί να υπάρχει ανάμεσα σε δύο επεξεργαστές οι οποίοι δεν είναι απευθείας ενωμένοι είναι $(d - \frac{u}{2}) \times \text{ελάχιστη απόσταση}$.

Η παραπάνω καθυστέρηση θα χρησιμοποιηθεί στο αλγόριθμο μέσου όρου που θα κατασκευάσω στο μετέπειτα στάδιο της εργασίας αυτής, εφόσον οι τοπολογίες οι οποίες θα αναλυθούν δεν είναι πλήρης συνδεδεμένες και έτσι τα μηνύματα δεν θα περνούν από ένα μόνο σύνδεσμο.

2.3 Αλγόριθμος μέσου όρου

Πολλοί είναι οι αλγόριθμοι οι οποίοι κατασκευάστηκαν με την πάροδο των χρόνων για να βοηθήσουν στον βέλτιστο συγχρονισμό των ρολογιών των επεξεργαστών. Για ένα κεντρικό σύστημα φτιάχτηκαν για παράδειγμα οι αλγόριθμοι Cristian [4] και ο αλγόριθμος Berkeley [5]. Από την άλλη για ένα κατακεντρωμένο σύστημα το πρόβλημα συγχρονισμού ρολογιών γίνεται πιο περίπλοκο εφόσον οι επεξεργαστές είναι κατακεντρωμένοι σε ολόκληρο το δίκτυο και δεν υπάρχει ένας κεντρικός διακομιστής που θα “υπαγορεύσει” τον χρόνο του συστήματος.

Στην εργασία αυτή χρησιμοποίησα μία παραλλαγή του αλγορίθμου Berkeley όπως παρουσιάζεται στο [5] έτσι ώστε να μπορεί να χρησιμοποιηθεί σε κατακεντρωμένα συστήματα. Πιο συγκεκριμένα, στον αλγόριθμο Berkeley “εκλέγεται” ένας κόμβος ως αρχηγός ο οποίος ζητά από τους υπόλοιπους κόμβους της τοπολογίας την τιμή των ρολογιών τους. Στην συνέχεια, παίρνει όλες τις τιμές και υπολογίζει τον μέσο τους όρο και τέλος τους επιστρέφει την μέση τιμή της ώρας που υπολόγισε. Στις τοπολογίες κατακεντρωμένου συστήματος που θα ακολουθήσουν, κατά την εκτέλεση του αλγορίθμου, όλοι οι επεξεργαστές της τοπολογίας πρέπει να γίνουν “αρχηγοί” πριν το τέλος του αλγορίθμου. Κάθε επεξεργαστής του δικτύου παραλαμβάνει τις τιμές της ώρας των ρολογιών όλων των υπόλοιπων επεξεργαστών μέσω μηνυμάτων. Ο αλγόριθμος αυτός, ονομάζεται αλγόριθμος μέσου όρου εφόσον μόλις ένας επεξεργαστής παραλάβει όλες τις τιμές των ρολογιών των υπόλοιπων επεξεργαστών και εφόσον υπολογίσει την διαφορά των τιμών αυτών από την δική του τιμή ρολογιού σε σχέση βέβαια με την εκτίμηση καθυστέρησης όπου θα κάνει, υπολογίζει τον μέσο όρο των τιμών αυτών και εφαρμόζει το προσαρμοσμένο του ρολόι στην μέση αυτή τιμή.

Κεφάλαιο 3

Συγχρονισμός Ρολογιών Σε Δακτύλιο Με Χορδές

3.1 Τοπολογία	9
3.2 Ελάχιστη Απόσταση	10
3.3 Διάμετρος	14
3.4 Αλγόριθμος	16
3.5 Ανάλυση Αλγορίθμου	17

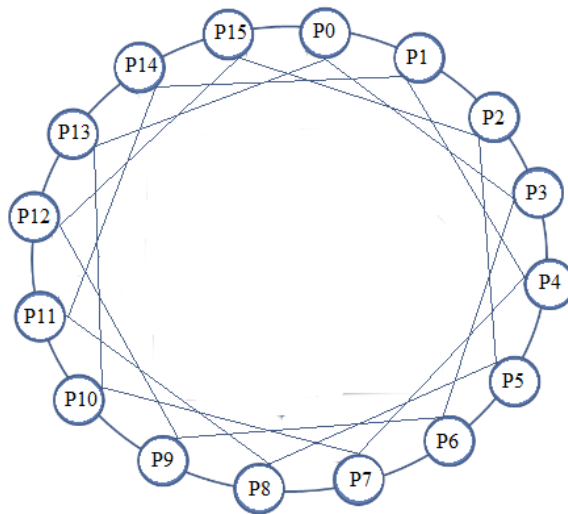
3.1 Τοπολογία

Η τοπολογία η οποία θα αναλυθεί στο κεφάλαιο αυτό είναι ο δακτύλιος με χορδές. Η όψη της τοπολογίας αυτής έχει ως βάση την τοπολογία του απλού δακτυλίου με επιπρόσθετους δύο συνδέσμους από κάθε επεξεργαστή.

Πιο συγκεκριμένα, κάθε επεξεργαστής p_i του δικτύου θα έχει τέσσερις γείτονες, δηλαδή τέσσερις επεξεργαστές με τους οποίους θα ενώνεται άμεσα με έναν μόνο σύνδεσμο μήκους 1. Οι δύο από τους τέσσερις γείτονες είναι αυτοί οι οποίοι θα ήταν και στον απλό δακτύλιο, ένας εκ των δεξιών του p_{i+1} και ένας εκ τ' αριστερών του p_{i-1} . Οι υπόλοιποι δύο γείτονες του p_{i+k} και p_{i-k} είναι εκείνοι με τους οποίους ενώνεται με μία χορδή από αυτόν. Ένα παράδειγμα του σχήματος της τοπολογίας παρουσιάζεται στο σχήμα 3.1.

Σημαντικό παράγοντα στην τοπολογία αυτή έχει η σταθερά k , της οποίας η τιμή ορίζει πόσοι ενδιάμεσοι σύνδεσμοι υπάρχουν ανάμεσα σε δύο επεξεργαστές οι οποίοι ενώνονται με μία χορδή. Με άλλα λόγια υπάρχει η δυνατότητα όταν ένας επεξεργαστής p_i θέλει να αποστείλει ένα μήνυμα στον επεξεργαστή p_j , αντί το μήνυμα να περάσει από k ενδιάμεσους συνδέσμους να προχωρήσει κατευθείαν από ένα μόνο σύνδεσμο, την χορδή την οποία τους ενώνει, μειώνοντας έτσι σημαντικά τον χρόνο αποστολής.

Τι γίνεται όμως εάν δύο επεξεργαστές δεν είναι απευθείας ενωμένοι με μία κοινή χορδή και θέλουν να επικοινωνήσουν; Θα αναλυθεί πιο κάτω ένας τύπος εύρεσης της ελάχιστης απόστασης ανάμεσα σε δύο οποιουσδήποτε δύο επεξεργαστές ο οποίος αξιοποιεί τόσο τις χορδές όσο και τους απλούς συνδέσμους.



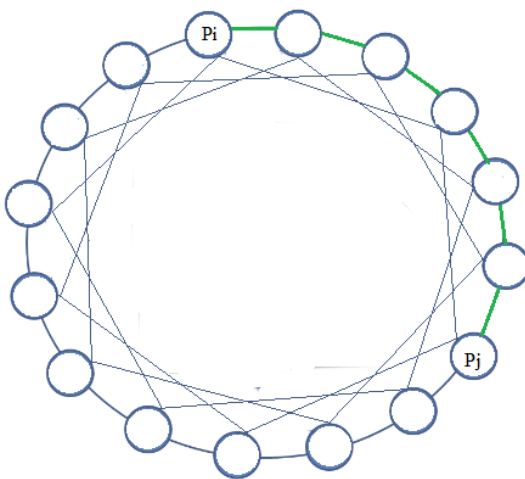
3.1 Δακτύλιος με χορδές, $n=16$, $\kappa=3$

3.2 Ελάχιστη Απόσταση

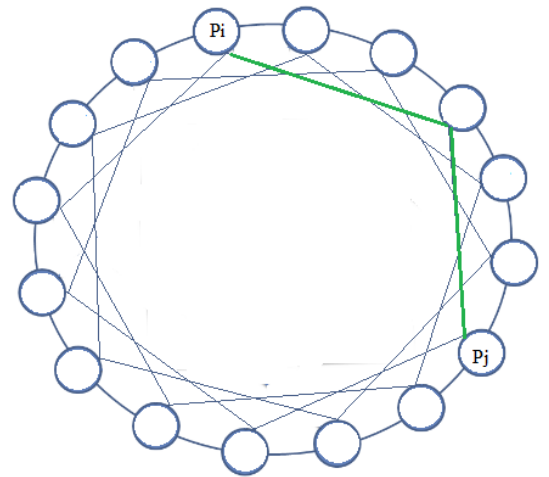
Στο μαθηματικό τομέα της θεωρίας των γράφων, η απόσταση μεταξύ δύο κορυφών σε ένα γράφημα είναι ο αριθμός των συνδέσμων που υπάρχουν στην συντομότερη διαδρομή που τους συνδέει. Σημαντικό είναι ότι μπορεί να υπάρχουν περισσότερες από μία συντομότερες διαδρομές μεταξύ δύο κορυφών. Στα κατακευκτωμένα συστήματα, οι κόμβοι της τοπολογίας επικοινωνούν μεταξύ τους μέσω ανταλλαγής μηνυμάτων. Εφόσον επιδιώκουμε συγχρονισμό της τοπολογίας, οι τυχόν καθυστερήσεις των μηνυμάτων δεν είναι επιθυμητές για τον υπολογισμό μας και έτσι για να μειώσουμε αυτές τις καθυστερήσεις, τα μηνύματα θα πρέπει να χρησιμοποιούν την ελάχιστη απόσταση για να φτάσουν στον προορισμό τους.

Στο σημείο αυτό, θα υποθέσω ότι η απόσταση μεταξύ δύο επεξεργαστών p_i και p_j χρησιμοποιώντας μόνο τις ακμές που βρίσκονται στην περίμετρο του δακτυλίου είναι θ . Πιο αναλυτικά εάν αυτοί οι δύο επεξεργαστές βρίσκονταν σε ένα απλό δακτύλιο χωρίς χορδές, η ελάχιστη απόσταση τους θα ήταν ίση με θ . Για να μειωθεί η απόσταση αυτή στην συγκεκριμένη τοπολογία αξιοποιούνται οι χορδές που υπάρχουν, οι οποίες όταν χρησιμοποιούνται παραλείπουν πάντα κ ακμές της περιμέτρου του δακτυλίου.

Συμπερασματικά και όπως μπορείτε να παρατηρήσετε στα σχήματα 3.2 και 3.3 η ελάχιστη απόσταση μεταξύ των επεξεργαστών p_i και p_j μπορεί να μειωθεί από θ σε $\frac{\theta}{\kappa}$.



3.2 Απόσταση χωρίς της χρήση χορδών,
distance = $\theta = 6$



3.3 Απόσταση με χρήση χορδών $\kappa = 3$,
distance = $\frac{\theta}{\kappa} = 2$

Αν θέλουμε όμως να αναλύσουμε με ακρίβεια την ελάχιστη απόσταση δύο επεξεργαστών στην τοπολογία αυτή, θα πρέπει να εξετάσουμε και το ενδεχόμενο όπου η απόσταση μεταξύ τους δεν μπορεί να υποδιαιρεθεί ακριβώς με την σταθερά κ . Πιο συγκεκριμένα, η πράξη $\frac{\theta}{\kappa}$ να δίνει υπόλοιπο αφήνοντας να γίνει εύκολα αντιληπτό ότι θα πρέπει αναγκάστηκε να χρησιμοποιηθεί ένας συνδυασμός χορδών και συνδέσμων της

περιμέτρου για να κτιστεί η ελάχιστη απόσταση μεταξύ των δύο επεξεργαστών. Από όλα τα παραπάνω γίνεται φανερό ότι, η ελάχιστη απόσταση χωρίζεται σε δύο περιπτώσεις.

Στην πρώτη περίπτωση, οι χορδές αξιοποιούνται για διαρροή του μηνύματος από τον επεξεργαστή αποστολέα ως κάποιο επεξεργαστή πριν από τον επεξεργαστή παραλήπτη και στην συνέχεια να περνά από κάποιους επιπρόσθετους συνδέσμους μέχρι να φτάσει στον προορισμό του. Αυτοί οι επιπρόσθετοι σύνδεσμοι θα είναι σίγουρα λιγότεροι από την σταθερά κ , διότι εάν ήταν περισσότεροι θα είχε αξιοποιηθεί η χορδή η οποία έχει μήκος 1. Η απόσταση σε αυτή την περίπτωση είναι $\lfloor \frac{\theta}{\kappa} \rfloor +$ επιπρόσθετους συνδέσμους.

Στην δεύτερη περίπτωση, οι χορδές αξιοποιούνται έτσι ώστε το μήνυμα να “ταξιδέψει” από τον επεξεργαστή αποστολέα και να καταλήξει σε κάποιο επεξεργαστή ο οποίος είναι μετά τον επεξεργαστή παραλήπτη. Στην συνέχεια το μήνυμα θα μετακινηθεί επιπρόσθετους συνδέσμους προς την αντίθετη φορά στην περίμετρο του δακτυλίου. Οι σύνδεσμοι αυτοί για τον ίδιο λόγο όπως και παραπάνω εξακολουθούν να είναι λιγότεροι από την σταθερά κ . Η απόσταση σε αυτή την περίπτωση είναι $\lceil \frac{\theta}{\kappa} \rceil +$ επιπρόσθετους συνδέσμους.

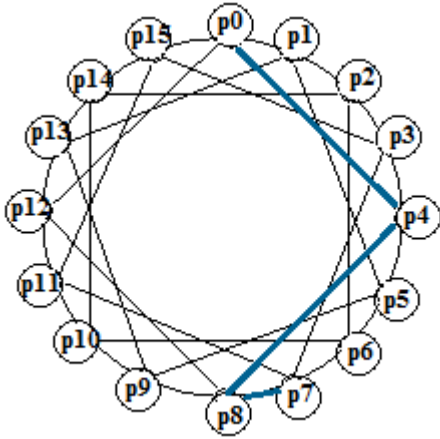
Οι επιπρόσθετοι σύνδεσμοι όπου χρειάζονται για να φτάσει το μήνυμα στον επεξεργαστή παραλήπτη μπορούν να υπολογισθούν με μία απλή πράξη. Λόγω του ότι οι σύνδεσμοι αυτοί βρίσκονται στην περίμετρο του δακτυλίου αρκεί απλά να βρεθεί η διαφορά των συνδέσμων της περιμέτρου που αντιστοιχούν στις χορδές οι οποίες αξιοποιήθηκαν από τους συνδέσμους τις περιμέτρου από τον επεξεργαστή αποστολέα στον επεξεργαστή παραλήπτη. Με αυτόν τον τρόπο, η ελάχιστη απόσταση στην πρώτη περίπτωση είναι

$$\lfloor \frac{\theta}{\kappa} \rfloor + \left| \theta - \lfloor \frac{\theta}{\kappa} \rfloor * \kappa \right| \quad \text{ενώ στην δεύτερη περίπτωση είναι} \quad \lceil \frac{\theta}{\kappa} \rceil + \left| \theta - \lceil \frac{\theta}{\kappa} \rceil * \kappa \right|.$$

Ποια όμως από τις δύο περιπτώσεις είναι η ιδανικότερη να βρεθεί την ελάχιστη απόσταση;

Όπως έχω παρατηρήσει, σε κάποιες περιπτώσεις είναι καλύτερο να χρησιμοποιηθεί η πρώτη και σε κάποιες η δεύτερη περίπτωση και αυτό οφείλεται σε διάφορους παράγοντες οι οποίοι αλλάζουν από τοπολογία σε τοπολογία όπως είναι η τιμή της σταθεράς κ , το εάν το κ είναι

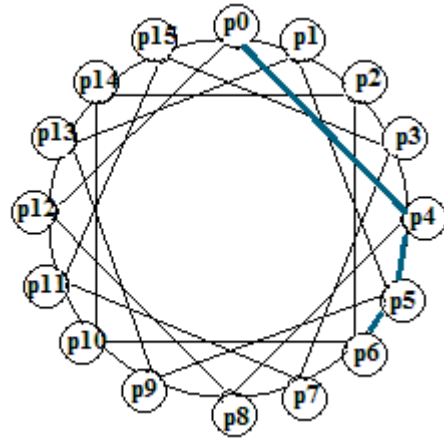
περιττός ή άρτιος αριθμός όπως και η σχέση του με το θ , για παράδειγμα όμοιος δακτύλιος με χορδές παρουσιάζεται στα σχήματα 3.4 και 3.5 αλλά ο τύπος ελάχιστης απόστασης που επιλέγεται για κάθε ζεύγος επεξεργαστών διαφέρει (η ελάχιστη διαδρομή φαίνεται στα σχήματα).



3.4 Δακτύλιος με χορδές $\kappa=4$, $\theta=7$

$$\lfloor \frac{\theta}{\kappa} \rfloor + | \theta - \lfloor \frac{\theta}{\kappa} \rfloor * \kappa | = 4$$

$$\lceil \frac{\theta}{\kappa} \rceil + | \theta - \lceil \frac{\theta}{\kappa} \rceil * \kappa | = 3$$



3.5 Δακτύλιος με χορδές $\kappa=4$, $\theta=6$

$$\lfloor \frac{\theta}{\kappa} \rfloor + | \theta - \lfloor \frac{\theta}{\kappa} \rfloor * \kappa | = 3$$

$$\lceil \frac{\theta}{\kappa} \rceil + | \theta - \lceil \frac{\theta}{\kappa} \rceil * \kappa | = 4$$

Συμπερασματικά, ο τύπος της ελάχιστης απόστασης στον δακτύλιο με χορδές είναι ο εξής:

$$\min ((\lfloor \frac{\theta}{\kappa} \rfloor + | \theta - \lfloor \frac{\theta}{\kappa} \rfloor * \kappa |) , (\lceil \frac{\theta}{\kappa} \rceil + | \theta - \lceil \frac{\theta}{\kappa} \rceil * \kappa |))$$

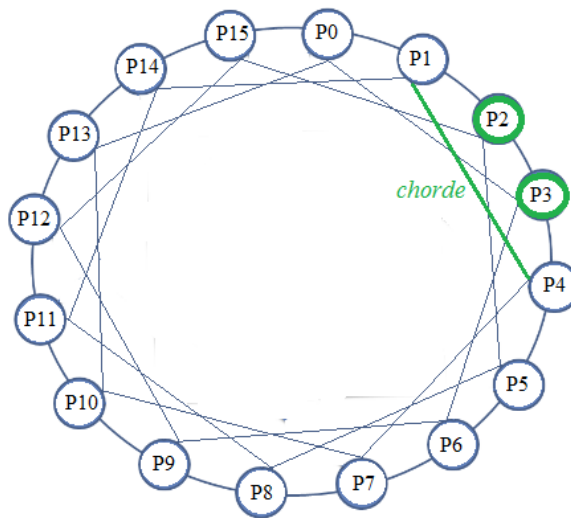
3.3 Διάμετρος

Η διάμετρος μίας τοπολογίας ορίζεται ως η μέγιστη από τις ελάχιστες αποστάσεις της. Με πιο απλά λόγια για κάθε ζεύγος επεξεργαστών εάν βρεθεί η ελάχιστη τους απόσταση, τότε η διάμετρος είναι η μέγιστη από αυτές τις αποστάσεις. Στηριζόμενοι στο υποκεφάλαιο 3.2 θα μπορούσαμε να πούμε ότι εφόσον η εύρεση της ελάχιστης απόστασης ανάμεσα σε δύο επεξεργαστές της τοπολογίας είναι σημαντική για μικρότερη λοξότητα ανάμεσα στα ρολόγια των επεξεργαστών έτσι και η διάμετρος είναι απαραίτητη για να βρούμε το άνω φράγμα, δηλαδή στην χειρότερη περίπτωση χρησιμοποιώντας τον αλγόριθμο που θα ακολουθήσει στο επόμενο υποκεφάλαιο ποιος συγχρονισμός ρολογιών θα αποκτηθεί.

Σε αυτό το σημείο θα υπολογίσουμε την διάμετρο στην τοπολογία Δακτύλιος με χορδές η οποία εξαρτάται κυρίως από το εάν η σταθερά κ είναι περιττός ή άρτιος αριθμός. Η διάμετρος στην συγκεκριμένη τοπολογία αποτελείται από την μέγιστη αξιοποίηση τόσο των χορδών όσο και των συνδέσμων στην περιφέρεια του δακτυλίου.

Η μέγιστη αξιοποίηση των χορδών της τοπολογίας γίνεται όταν ένας επεξεργαστής θέλει να προωθήσει μήνυμα σε απόσταση μισής περιφέρειας του δακτυλίου από το σημείο όπου βρίσκεται δηλαδή $\frac{n}{2}$. Πιο συγκεκριμένα με την χρησιμοποίησή χορδών το μήνυμα διανύει $\frac{\frac{n}{2}}{\kappa} = \frac{n}{2\kappa}$ και για σκοπούς στρογγυλοποίησης των βημάτων εφόσον αυτά επιβάλλεται να είναι θετικός ακέραιος αριθμός καταλήγω ότι ο αριθμός αυτός έχει την τιμή $\frac{n}{2\kappa} - (\frac{n}{2} \bmod \kappa) \quad (1)$

Η μέγιστη από τις ελάχιστες αποστάσεις όμως αποτελεί μία απόσταση μηνύματος η οποία περιλαμβάνει και επιπρόσθετα βήματα στους συνδέσμους στην περιφέρεια του δακτυλίου, χωρίς την αξιοποίηση χορδών. Ο αριθμός των συνδέσμων αυτών, θα εξαρτηθούν από το εάν η σταθερά κ είναι άρτιος η περιττός αριθμός. Προφανώς, εφόσον μια χορδή αντικαθιστά κ συνδέσμους, προσπερνά ακριβώς $\kappa-1$ επεξεργαστές στην περιφέρεια όπως φαίνεται στο σχήμα 3.6.



3.6 Δακτύλιος με χορδές $\kappa=3$, επεξεργαστές στην περιφέρεια ανά χορδή=2

- Εάν το κ είναι περιττός αριθμός
 - $\Rightarrow$ Υπάρχει άρτιος αριθμός επεξεργαστών $\kappa-1$ στην περιφέρεια ανά χορδή
 - $\Rightarrow$ Το μήνυμα αρκεί να προωθηθεί προς αντίθετη κατεύθυνση, στους μισούς από τους ενδιάμεσους επεξεργαστές στην περιφέρεια άρα επιπρόσθετα $\frac{\kappa-1}{2}$ βήματα. **(2α)**
- Εάν το κ είναι άρτιος αριθμός
 - $\Rightarrow$ Υπάρχει περιττός αριθμός επεξεργαστών στην περιφέρεια ανά χορδή
 - $\Rightarrow$ Το μήνυμα αρκεί να προωθηθεί προς αντίθετη κατεύθυνση, σε ένα ακριβώς λιγότερο από τους μισούς από τους ενδιάμεσους επεξεργαστές στην περιφέρεια άρα επιπρόσθετα $\frac{\kappa}{2}-1$ βήματα. Εάν το μήνυμα είχε προωθηθεί σε περισσότερους επεξεργαστές _στην περιφέρεια τότε θα ήταν αχρείαστη η χρήση της χορδής.
(2β)

Διάμετρος για κ περιττό αριθμό

$$\text{Από } (\alpha) \text{ και } (2\alpha) : \text{Diam} = \frac{n}{2k} - \left(\frac{n}{2} \bmod k\right) + \left(\frac{k-1}{2}\right)$$

$$\Leftrightarrow \text{Diam} = \frac{n+k^2-k}{2k} - n \bmod 2k$$

Διάμετρος για κ άρτιο αριθμό

$$\text{Από } (\alpha) \text{ και } (2\beta) : \text{Diam} = \frac{n}{2k} - \left(\frac{n}{2} \bmod k\right) + \left(\frac{k}{2} - 1\right)$$

$$\Leftrightarrow \text{Diam} = \frac{n+k^2-2k}{2k} - n \bmod 2k$$

3.4 Αλγόριθμος

Σε αυτό το υποκεφάλαιο θα παρουσιαστεί ο αλγόριθμος μέσου όρου για τον συγχρονισμό ρολογιών της τοπολογίας Δακτύλιος με χορδές. Πιο συγκεκριμένα είναι ένας αλγόριθμος μέσου όρου (averaging algorithm) , στον οποίο όλοι οι επεξεργαστές προσθέτουν στην τοπική τους μνήμη την ώρα όλων των υπόλοιπων επεξεργαστών του δικτύου. Η επικοινωνία γίνεται μέσω μηνυμάτων τα οποία περνούν από το ελάχιστο μονοπάτι έτσι ώστε να μειωθούν οι τυχόν καθυστερήσεις.

Πριν την εκτέλεση του αλγορίθμου, κάθε επεξεργαστής υπολογίζει τοπικά την ελάχιστη απόσταση μεταξύ του ιδίου και τους υπόλοιπους επεξεργαστές του δακτυλίου με χορδές, χρησιμοποιώντας τον τύπο ελάχιστης απόστασης του υποκεφαλαίου 3.2

Αλγόριθμος

- 1- Αρχικοποίηση $\text{diff}[i] = 0$, για κάθε επεξεργαστή.
- 2- Κάθε επεξεργαστής αρχικοποιά και μία μεταβλητή η οποία λειτουργά ως μετρητής $\text{count} = 0$.
- 3- Στην συνέχεια ολόκληρο το δίκτυο στέλνει την τιμή του ρολογιού του στους υπόλοιπους επεξεργαστές του δικτύου διαμέσου της ελάχιστης διαδρομής.

Η δρομολόγηση του μηνύματος μέσω της ελάχιστης διαδρομής επιλέγεται ως εξής:

- I. $X = \text{min_distance}$
 - II. Εάν η ελάχιστη απόσταση είναι $X \geq \kappa$, τότε μετακινείται διαμέσου της χορδής και μειώνει κατά μία μονάδα την τιμή της τοπικής μεταβλητής min_distance .
 - III. Επαναλαμβάνεται το πιο πάνω μέχρις ότου η ελάχιστη απόσταση να είναι $X < \kappa$, τότε μετακινείται από επεξεργαστή-σε επεξεργαστή με βήματα ίσα με την νέα τιμή της ελάχιστης απόστασης.
- 4- Όταν οποιοδήποτε μήνυμα ενός επεξεργαστή p_j φτάσει στον επεξεργαστή p_i τότε ο πίνακας diff διαμορφώνεται ως εξής :
- $$\text{Diff}_i[j] = T + (d - \frac{u}{2}) * \text{min_distance} - \text{HC}_i.$$
- Και ο μετρητής count αυξάνεται κατά μία μονάδα.
- 5- Όταν ο μετρητής count γίνει ίσος με $n-1$ δηλαδή, όταν παραληφθούν και τα $n-1$ μηνύματα και ο πίνακας diff είναι πλήρης τότε ο επεξεργαστής μπορεί να υπολογίσει τον μέσο όρο των τιμών που έχει στον τοπικό του πίνακα diff ως εξής:
- $$\text{Adj} = \frac{1}{n} * \sum_{k=0}^{n-1} \text{Diff}_i[k]$$

3.5 Ανάλυση Αλγορίθμου

$$|AC_i(t) - AC_j(t)| = |HC_i(t) + \frac{1}{n} \sum_{k=0}^{n-1} \text{diff}_i[k] - HC_j(t) - \frac{1}{n} \sum_{k=0}^{n-1} \text{diff}_j[k]|$$

$$\Leftrightarrow \frac{1}{n} (|HC_i(t) - HC_j(t) + diff_i[i] - diff_j[i] + HC_i(t) - HC_j(t) + diff_i[j] - diff_j[j]| + \sum_{k=0, k \neq i, j}^{n-1} (HC_i(t) - HC_j(t) + diff_i[k] - diff_j[k]))$$

$$\Leftrightarrow \frac{1}{n} (\underbrace{|HC_j(t) - HC_i(t) + diff_j[i]|}_{A} + \underbrace{|HC_i(t) - HC_j(t) + diff_i[j]|}_{B} +$$

$$\underbrace{\sum_{k=0, k \neq i, j}^{n-1} |HC_i(t) - HC_j(t) + diff_i[k] - diff_j[k]|}_{C})$$

$$\begin{aligned} A. & |HC_j(t) - HC_i(t) + diff_j[i]| \\ &= HC_j(t) - HC_i(t) + HC_i(t) - HC_j(t) + \frac{u}{2} * \text{min_distance}_{ij} \\ &= \frac{u}{2} * \text{min_distance}_{ij} \end{aligned}$$

$$\begin{aligned} B. & |HC_i(t) - HC_j(t) + diff_i[j]| \\ &= HC_i(t) - HC_j(t) + HC_j(t) - HC_i(t) + \frac{u}{2} * \text{min_distance}_{ij} \\ &= \frac{u}{2} * \text{min_distance}_{ij} \end{aligned}$$

C. Οι υπόλοιποι n-2 όροι :

$$\sum_{k=0, k \neq i, j}^{n-1} |HC_i(t) - HC_j(t) + diff_i[k] - diff_j[k]|$$

$$\begin{aligned} \text{Κάθε όρος : } & HC_i(t) - HC_j(t) + diff_i[k] - diff_j[k] \\ &= HC_i(t) - HC_j(t) + HC_k(t) - HC_i(t) + (err_{ki} * \text{min_distance}_{ki}) - \\ & \quad (HC_k(t) - HC_i(t) - (err_{kj} * \text{min_distance}_{kj})) \end{aligned}$$

Όλοι οι όροι απαλείφονται και απομένει μόνο :

$$\text{err}_{ki} * \text{min_distance}_{ki} + \text{err}_{kj} * \text{min_distance}_{kj}$$

Γνωρίζουμε όμως ότι $\text{min_distance}_{ki} + \text{min_distance}_{kj}$ ισούται ακριβώς με την min_distance_{ij} εφόσον εάν το άθροισμα τους ήταν μικρότερο τότε θα υπήρχε πιο σύντομη διαδρομή ανάμεσα στον p_i και p_j και αυτό θα οδηγούσε σε άτοπο .

$$\Rightarrow \frac{u}{2} * \text{min_distance}_{ki} + \frac{u}{2} * \text{min_distance}_{kj}$$

$$\Rightarrow \frac{u}{2} * (\text{min_distance}_{ki} + \text{min_distance}_{kj})$$

$$\Rightarrow \frac{u}{2} * \text{min_distance}_{ij}$$

$$\Rightarrow \text{Άρα συνολικά από τους } n-2 \text{ όρους έχουμε : } (n-2) * \frac{u}{2} * \text{min_distance}_{ij}$$

Από A , B και C έχουμε :

$$\frac{u}{2} * \text{min_distance}_{ij} + \frac{u}{2} * \text{min_distance}_{ij} + (n-2) * \frac{u}{2} * \text{min_distance}_{ij}$$

$$\Leftrightarrow \frac{u}{2} * \text{min_distance}_{ij} (1 + 1 + n-2)$$

$$\Leftrightarrow \frac{u * \text{min_distance}_{ij} * n}{2}$$

Άρα τέλος, κατασκευάζεται ένα αυστηρό φράγμα. Η διάμετρος της τοπολογίας θα πρέπει να αντικαταστήσει την ελάχιστη απόσταση και έτσι είναι πλέον φανερό ότι το αυστηρό αυτό φράγμα εξαρτάται από το εάν η σταθερά κ είναι περιττός ή άρτιος αριθμός.

$$\frac{1}{n} \left(\frac{u * \min_distance_{ij} * n}{2} \right)$$

$$\Rightarrow \frac{u * \min_distance_{ij}}{2}$$

Για κ περιττός αριθμός :

$$\frac{u * \left(\frac{n + k^2 - k}{2k} - n \bmod 2k \right)}{2}$$

$$\Rightarrow u * \left(\frac{n + k^2 - k}{4k} - n \bmod 4k \right)$$

Για κ άρτιος αριθμός :

$$\frac{u * \left(\frac{n + k^2 - 2k}{2k} - n \bmod 2k \right)}{2}$$

$$\Rightarrow u * \left(\frac{n + k^2 - 2k}{4k} - n \bmod 4k \right)$$

$$u \cdot \left(\frac{n}{4k} + \frac{1}{4} - (n \bmod 4k) \right) - \begin{cases} \frac{1}{4}, & \kappa \text{ περιττός} \\ \frac{1}{2}, & \kappa \text{ άρτιος} \end{cases}$$

Συμπερασματικά, για κ περιττό αριθμό έχουμε ως αποτέλεσμα το πιο μεγάλο, αυστηρό φράγμα του αλγόριθμου μέσου όρου στην τοπολογία δακτύλιος με χορδές.

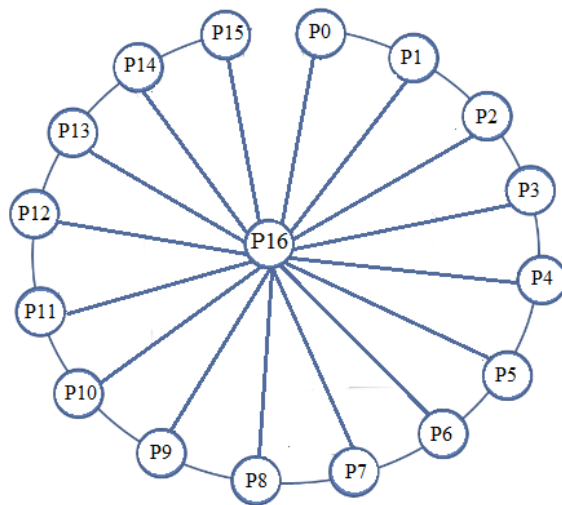
Κεφάλαιο 4

Συγχρονισμός Ρολογιών Σε Τοπολογία Ανεμιστήρα

4.1 Τοπολογία και γενικά χαρακτηριστικά	21
4.2 Αλγόριθμος	22
4.3 Ανάλυση Αλγορίθμου	23

4.1 Τοπολογία και γενικά χαρακτηριστικά

Στο κεφάλαιο αυτό θα παρουσιαστεί η τοπολογία ανεμιστήρα (fan topology), μια τοπολογία η οποία μοιάζει μερικώς με την τοπολογία του δακτυλίου με την διαφορά ότι ένας από τους επεξεργαστές βρίσκεται στο κέντρο των υπολοίπων και επιπρόσθετα ένας σύνδεσμος αφαιρείται από την περιφέρεια του κύκλου. Αναλυτικότερα, η τοπολογία μπορεί να παρουσιαστεί σαν ένα γράφημα με n επεξεργαστές με δείκτες. Οι επεξεργαστές με δείκτες από p_0 ως p_{n-2} ενώνονται κυκλικά, όμως ένας από τους συνδέσμους στην περιφέρεια παραλείπεται. Επιπρόσθετα, ο τελευταίος επεξεργαστής με δείκτη p_{n-1} τοποθετείται στο κέντρο και συνδέεται με ακμή με τους υπόλοιπους $n-1$ επεξεργαστές. Η τοπολογία ανεμιστήρα είναι όπως φαίνεται στο σχήμα 4.1.



4.1 Τοπολογία Ανεμιστήρας με μήκος $n = 17$

Οι αποστάσεις μεταξύ των επεξεργαστών είναι είτε ένας σύνδεσμος εάν οι δύο επεξεργαστές είναι γείτονες είτε δύο σύνδεσμοι εάν δεν είναι γείτονες. Ο τρόπος που είναι κτισμένη η συγκεκριμένη τοπολογία βοηθά απομακρυσμένους επεξεργαστές να απέχουν μόνο δύο συνδέσμους στέλνοντας το μήνυμα διαμέσου του κεντρικού επεξεργαστή. Έτσι είναι φανερό ότι η μέγιστη από τις ελάχιστες αποστάσεις, δηλαδή η διάμετρος της τοπολογίας είναι 2 σύνδεσμοι.

4.2 Αλγόριθμος

Χρησιμοποίηση αλγορίθμου μέσου όρου στην τοπολογία ανεμιστήρας (fan topology)

Αλγόριθμος

- 1- Αρχικοποίηση $\text{diff} [i] = 0$, για κάθε επεξεργαστή.
- 2- Κάθε επεξεργαστής αρχικοποιά και μία μεταβλητή η οποία λειτουργά ως μετρητής $\text{count} = 0$.
- 3- Στην συνέχεια ολόκληρο το δίκτυο στέλνει την τιμή του ρολογιού του στους υπόλοιπους επεξεργαστές του δικτύου διαμέσου της ελάχιστης διαδρομής.
- 4- Όταν οποιοδήποτε μήνυμα ενός επεξεργαστή p_i φτάσει στον επεξεργαστή p_i τότε ο πίνακας diff διαμορφώνεται ως εξής :

$$\text{Diff}_i [j] = T + (d - \frac{u}{2}) * \text{min_distance} - \text{HC}_i .$$

Και ο μετρητής count αυξάνεται κατά μία μονάδα.
- 5- Όταν ο μετρητής count γίνει ίσος με $n-1$ δηλαδή, όταν παραληφθούν και τα $n-1$ μηνύματα και ο πίνακας diff είναι πλήρης τότε ο επεξεργαστής μπορεί να υπολογίσει τον μέσο όρο των τιμών που έχει στον τοπικό του πίνακα diff ως εξής:

$$\text{Adj} = \frac{1}{n} * \sum_{k=0}^{n-1} \text{Diff}_i [k]$$

4.3 Ανάλυση Αλγορίθμου

$$| \text{AC}_i(t) - \text{AC}_j(t) | = | \text{HC}_i(t) + \frac{1}{n} \sum_{k=0}^{n-1} \text{diff}_i[k] - \text{HC}_j(t) - \frac{1}{n} \sum_{k=0}^{n-1} \text{diff}_j[k] |$$

$$\Leftrightarrow \frac{1}{n} (| \text{HC}_i(t) - \text{HC}_j(t) + \text{diff}_i[i] - \text{diff}_j[i] + \text{HC}_i(t) - \text{HC}_j(t) + \text{diff}_i[j] - \text{diff}_j[j] | + \sum_{k=0, k \neq i, j}^{n-1} (\text{HC}_i(t) - \text{HC}_j(t) + \text{diff}_i[k] - \text{diff}_j[k]))$$

$$\Leftrightarrow \frac{1}{n} (\underbrace{| \text{HC}_j(t) - \text{HC}_i(t) + \text{diff}_j[i] |}_{\text{A}} + \underbrace{| \text{HC}_i(t) - \text{HC}_j(t) + \text{diff}_i[j] |}_{\text{B}} +$$

$$\underbrace{\sum_{k=0, k \neq i, j}^{n-1} | HC_i(t) - HC_j(t) + diff_i[k] - diff_j[k] | }_C$$

$$\begin{aligned} A. & | HC_j(t) - HC_i(t) + diff_j[i] | \\ &= HC_j(t) - HC_i(t) + HC_i(t) - HC_j(t) + \frac{u}{2} * \min_distance_{ij} \\ &= \frac{u}{2} * \min_distance_{ij} \end{aligned}$$

$$\begin{aligned} B. & | HC_i(t) - HC_j(t) + diff_i[j] | \\ &= HC_i(t) - HC_j(t) + HC_j(t) - HC_i(t) + \frac{u}{2} * \min_distance_{ij} \\ &= \frac{u}{2} * \min_distance_{ij} \end{aligned}$$

C. Οι υπόλοιποι n-2 όροι :

$$\sum_{k=0, k \neq i, j}^{n-1} | HC_i(t) - HC_j(t) + diff_i[k] - diff_j[k] |$$

$$\begin{aligned} \text{Κάθε όρος : } & HC_i(t) - HC_j(t) + diff_i[k] - diff_j[k] \\ &= HC_i(t) - HC_j(t) + HC_k(t) - HC_i(t) + (err_{ki} * \min_distance_{ki}) - \\ & \quad (HC_k(t) - HC_i(t) - (err_{kj} * \min_distance_{kj})) \end{aligned}$$

Όλοι οι όροι απαλείφονται και απομένει μόνο :

$$err_{ki} * \min_distance_{ki} + err_{kj} * \min_distance_{kj}$$

Γνωρίζουμε όμως ότι $\min_distance_{ki} + \min_distance_{kj}$ ισούται ακριβώς με την $\min_distance_{ij}$ εφόσον εάν το άθροισμα τους ήταν μικρότερο τότε θα υπήρχε πιο σύντομη διαδρομή ανάμεσα στον p_i και p_j και αυτό θα οδηγούσε σε άτοπο .

$$\Rightarrow \frac{u}{2} * \min_distance_{ki} + \frac{u}{2} * \min_distance_{kj}$$

$$\Rightarrow \frac{u}{2} * (\text{min_distance}_{ki} + \text{min_distance}_{ki})$$

$$\Rightarrow \frac{u}{2} * \text{min_distance}_{ij}$$

$$\Rightarrow \text{Άρα συνολικά από τους } n-2 \text{ όρους έχουμε : } (n-2) * \frac{u}{2} * \text{min_distance}_{ij}$$

Από A , B και C έχουμε :

$$\frac{u}{2} * \text{min_distance}_{ij} + \frac{u}{2} * \text{min_distance}_{ij} + (n-2) * \frac{u}{2} * \text{min_distance}_{ij}$$

$$\Leftrightarrow \frac{u}{2} * \text{min_distance}_{ij} (1 + 1 + n-2)$$

$$\Leftrightarrow \frac{u * \text{min_distance}_{ij} * n}{2}$$

$$\Leftrightarrow \frac{1}{n} (\frac{u * \text{min_distance}_{ij} * n}{2})$$

$$\Leftrightarrow \frac{u * \text{min_distance}_{ij}}{2}$$

Άρα τέλος, κατασκευάζεται ένα αυστηρό φράγμα. Η διάμετρος της τοπολογίας θα πρέπει να αντικαταστήσει την ελάχιστη απόσταση.

$$\Rightarrow \frac{u*2}{2}$$

$\Rightarrow u$

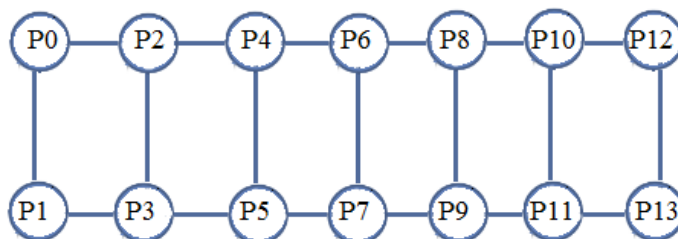
Κεφάλαιο 5

Συγχρονισμός Ρολογιών Σε Τοπολογία Σκάλα

5.1 Τοπολογία και γενικά χαρακτηριστικά	27
5.2 Αλγόριθμος	28
5.3 Ανάλυση Αλγορίθμου	29

5.1 Τοπολογία και γενικά χαρακτηριστικά

Στο κεφάλαιο που θα ακολουθήσει, θα παρουσιαστεί η τοπολογία σκάλα (ladder). Η κατασκευή της έχει ως εξής : Θεωρούμε τα δύο μονοπάτια από επεξεργαστές $Path_1 = P_0 - P_2 - \dots - P_{n-2}$ και $Path_2 = P_1 - P_3 - \dots - P_{n-1}$ να είναι ανεξάρτητα μεταξύ τους. Στην συνέχεια προσθέτουμε $\frac{n}{2}$ συνδέσμους οι οποίοι συνδέουν τους επεξεργαστές των δύο μονοπατιών ως εξής $edge (P_i , P_{i+1})$, $i = 0,1,\dots,n-1$. Στην τοπολογία αυτή υπάρχουν $n + (\frac{n}{2} - 2)$ σύνδεσμοι, όπου n είναι ο αριθμός των επεξεργαστών που υπάρχουν σε αυτό το δίκτυο. Παράλληλα, τέσσερις από τους επεξεργαστές της τοπολογίας έχουν δύο γείτονες, ενώ οι υπόλοιποι $n - 4$ επεξεργαστές έχουν 3 γείτονες. Παράδειγμα της τοπολογίας υπάρχει στο σχήμα 5.1



5.1 Τοπολογία σκάλα με μήκος $n = 14$

Για τον υπολογισμό τόσο της ελάχιστης απόστασης ανάμεσα σε δύο επεξεργαστές όσο και την διάμετρο της τοπολογίας σκάλα, θα χρησιμοποιήσω τους δείκτες των επεξεργαστών όπου όπως όρισα παραπάνω το μονοπάτι $Path_1$ αποτελείται από επεξεργαστές με άρτιο δείκτη ενώ το μονοπάτι $Path_2$ αποτελείται από επεξεργαστές με περιττό δείκτη.

Η ελάχιστη απόσταση ανάμεσα σε δύο επεξεργαστές εξαρτάται από το εάν ο επεξεργαστής αποστολέας και επεξεργαστής παραλήπτης έχουν ή όχι και οι δύο άρτιο ή και οι δύο περιττό δείκτη. Εάν έχουν και οι δύο άρτιο ή εάν έχουν και οι δύο περιττό δείκτη τότε η ελάχιστη τους απόσταση είναι οι ενδιάμεσοι τους σύνδεσμοι στο ανάλογο μονοπάτι. Εάν από την άλλη ο ένας από τους δύο έχει άρτιο δείκτη και ο άλλος περιττό τότε θα προστεθεί στην απόσταση που τους χωρίζει ακόμη ένας επιπλέον σύνδεσμος έτσι ώστε το μήνυμα να φτάσει στο “απέναντι μονοπάτι”.

Η διάμετρος του δικτύου είναι ίση με $\frac{n}{2}$ εφόσον η μέγιστη από τις ελάχιστες αποστάσεις που δίνετε να “ταξιδέψει” ένα μήνυμα είναι ένας σύνδεσμος για να περάσει στο αντίθετο μονοπάτι και οι υπόλοιποι $\frac{n}{2} - 1$ σύνδεσμοι για να φτάσει στο πιο μακρινό από αυτόν επεξεργαστή παραλήπτη.

5.2 Αλγόριθμος

Ο αλγόριθμος που θα ακολουθήσει και για την τελευταία αυτή τοπολογία είναι ο αλγόριθμος μέσου όρου στον οποίο και σε αυτήν την περίπτωση κατά τον τερματισμό του οι επεξεργαστές θα θέσουν το προσαρμοσμένο τους ρολόι στον μέσο όρο των τιμών που στάλθηκαν από τους υπόλοιπους επεξεργαστές της τοπολογίας. Πριν την εκτέλεση του αλγορίθμου, κάθε επεξεργαστής υπολογίζει τοπικά την ελάχιστη απόσταση μεταξύ του ιδίου και τους υπόλοιπους επεξεργαστές του συγκεκριμένου δικτύου.

Αλγόριθμος

- 1- Αρχικοποίηση $\text{diff} [i] = 0$, για κάθε επεξεργαστή.
- 2- Κάθε επεξεργαστής αρχικοποιά και μία μεταβλητή η οποία λειτουργά ως μετρητής $\text{count} = 0$.
- 3- Στην συνέχεια ολόκληρο το δίκτυο στέλνει την τιμή του ρολογιού του στους υπόλοιπους επεξεργαστές του δικτύου διαμέσου της ελάχιστης διαδρομής.
- 4- Όταν οποιοδήποτε μήνυμα ενός επεξεργαστή p_j φτάσει στον επεξεργαστή p_i τότε ο πίνακας diff διαμορφώνεται ως εξής :

$$\text{Diff}_i [j] = T + (d - \frac{u}{2}) * \text{min_distance} - \text{HC}_i .$$

Και ο μετρητής count αυξάνεται κατά μία μονάδα.

- 5- Όταν ο μετρητής count γίνει ίσος με $n-1$ δηλαδή, όταν παραληφθούν και τα $n-1$ μηνύματα και ο πίνακας diff είναι πλήρης τότε ο επεξεργαστής μπορεί να υπολογίσει τον μέσο όρο των τιμών που έχει στον τοπικό του πίνακα diff ως εξής:

$$\text{Adj} = \frac{1}{n} * \sum_{k=0}^{n-1} \text{Diff}_i [k]$$

5.3 Ανάλυση Αλγορίθμου

Σε αντίθεση με την τοπολογία του δακτυλίου με χορδές ο συγχρονισμός ρολογιών στην παρούσα τοπολογία δεν εξαρτάται από σταθερές ή από το εάν το πλήθος των επεξεργαστών είναι άρτιο ή περιττό διότι προφανώς πάντα το μήκος στην συγκεκριμένη τοπολογία είναι άρτιο. Συνάγεται το συμπέρασμα ότι η διάμετρος, η οποία όπως προανέφερα και στις προηγούμενες τοπολογίες είναι σημαντικό κομμάτι του συγχρονισμού ρολογιών, είναι ίση ακριβώς με $\frac{n}{2}$.

$$| \text{AC}_i(t) - \text{AC}_j(t) | = | \text{HC}_i(t) + \frac{1}{n} \sum_{k=0}^{n-1} \text{diff}_i[k] - \text{HC}_j(t) - \frac{1}{n} \sum_{k=0}^{n-1} \text{diff}_j[k] |$$

$$\Leftrightarrow \frac{1}{n} (| HC_i(t) - HC_j(t) + diff_i[i] - diff_j[i] + HC_i(t) - HC_j(t) + diff_i[j] - diff_j[j] + \sum_{k=0, k \neq i, j}^{n-1} (HC_i(t) - HC_j(t) + diff_i[k] - diff_j[k]))$$

$$\Leftrightarrow \frac{1}{n} (\underbrace{| HC_j(t) - HC_i(t) + diff_j[i] |}_{A} + \underbrace{| HC_i(t) - HC_j(t) + diff_i[j] |}_{B} +$$

$$\underbrace{\sum_{k=0, k \neq i, j}^{n-1} | HC_i(t) - HC_j(t) + diff_i[k] - diff_j[k] | }_{C})$$

$$\begin{aligned} A. & | HC_j(t) - HC_i(t) + diff_j[i] | \\ &= HC_j(t) - HC_i(t) + HC_i(t) - HC_j(t) + \frac{u}{2} * \text{min_distance}_{ij} \\ &= \frac{u}{2} * \text{min_distance}_{ij} \end{aligned}$$

$$\begin{aligned} B. & | HC_i(t) - HC_j(t) + diff_i[j] | \\ &= HC_i(t) - HC_j(t) + HC_j(t) - HC_i(t) + \frac{u}{2} * \text{min_distance}_{ij} \\ &= \frac{u}{2} * \text{min_distance}_{ij} \end{aligned}$$

C. Οι υπόλοιποι n-2 όροι :

$$\sum_{k=0, k \neq i, j}^{n-1} | HC_i(t) - HC_j(t) + diff_i[k] - diff_j[k] |$$

$$\begin{aligned} \text{Κάθε όρος : } & HC_i(t) - HC_j(t) + diff_i[k] - diff_j[k] \\ &= HC_i(t) - HC_j(t) + HC_k(t) - HC_i(t) + (err_{ki} * \text{min_distance}_{ki}) - \\ & \quad (HC_k(t) - HC_i(t) - (err_{kj} * \text{min_distance}_{kj})) \end{aligned}$$

Όλοι οι όροι απαλείφονται και απομένει μόνο :

$$\text{err}_{\kappa i} * \text{min_distance}_{\kappa i} + \text{err}_{\kappa j} * \text{min_distance}_{\kappa j}$$

Γνωρίζουμε όμως ότι $\text{min_distance}_{\kappa i} + \text{min_distance}_{\kappa j}$ ισούται ακριβώς με την min_distance_{ij} εφόσον εάν το άθροισμα τους ήταν μικρότερο τότε θα υπήρχε πιο σύντομη διαδρομή ανάμεσα στον p_i και p_j και αυτό θα οδηγούσε σε άτοπο .

$$\Rightarrow \frac{u}{2} * \text{min_distance}_{\kappa i} + \frac{u}{2} * \text{min_distance}_{\kappa j}$$

$$\Rightarrow \frac{u}{2} * (\text{min_distance}_{\kappa i} + \text{min_distance}_{\kappa j})$$

$$\Rightarrow \frac{u}{2} * \text{min_distance}_{ij}$$

$$\Rightarrow \text{Άρα συνολικά από τους } n-2 \text{ όρους έχουμε : } (n-2) * \frac{u}{2} * \text{min_distance}_{ij}$$

Από A , B και C έχουμε :

$$\frac{u}{2} * \text{min_distance}_{ij} + \frac{u}{2} * \text{min_distance}_{ij} + (n-2) * \frac{u}{2} * \text{min_distance}_{ij}$$

$$\Leftrightarrow \frac{u}{2} * \text{min_distance}_{ij} (1 + 1 + n-2)$$

$$\Leftrightarrow \frac{u * \text{min_distance}_{ij} * n}{2}$$

Άρα τέλος, κατασκευάζεται ένα αυστηρό φράγμα. Η διάμετρος της τοπολογίας θα πρέπει να αντικαταστήσει την ελάχιστη απόσταση.

$$\frac{1}{n} \left(\frac{u * \min_distance_{ij} * n}{2} \right)$$

$$\Rightarrow \frac{u * \min_distance_{ij}}{2}$$

$$\Rightarrow \frac{u * (\frac{n}{2})}{2}$$

$$\Rightarrow u * \frac{n}{4}$$

Κεφάλαιο 6

Συμπεράσματα

6.1 Αυστηρό φράγμα	39
6.2 Αποτελέσματα	40
6.3 Μελλοντική Εργασία	41

6.1 Αυστηρό φράγμα

Οι Lundelius και Lynch έδειξαν ότι τα ρολόγια των n επεξεργαστών σε ένα πλήρως συνδεδεμένο δίκτυο με την ίδια αβεβαιότητα u σε κάθε ακμή δεν μπορούν να συγχρονιστούν καλύτερα από $u * (1 - \frac{1}{n})$.

Μεταγενέστερα οι Halpern et al. επέκτειναν αυτό το έργο για να εξετάσουν αυθαίρετες τοπολογίες στις οποίες κάθε κατευθυνόμενη ακμή μπορεί να έχει διαφορετική αβεβαιότητα. Ίδρυσαν μία κλειστή μορφή έκφρασης για βέλτιστο συγχρονισμό σε ένα δέντρο και ένα τρίγωνο. Όσον αφορά το δέντρο, το φράγμα είναι ίσο με $\frac{1}{2}$ διαμέτρου του δέντρου σε σχέση με τις αβεβαιότητες στις συνδέσεις.

6.2 Αποτελέσματα

Κλείνοντας, θα παρουσιάσω όλα τα αποτελέσματα του αλγορίθμου μέσου όρου ο οποίος είναι βέλτιστος και στις τρεις τοπολογίες όπως φαίνονται στον πίνακα 6.1 . Γενική παρατήρηση είναι ότι όλα τα αυστηρά αυτά φράγματα είναι $\frac{u \cdot \text{διάμετρος}}{2}$, όπου u η αβεβαιότητα της καθυστέρησης των μηνυμάτων στους συνδέσμους.

Τοπολογία	Αυστηρό Φράγμα
Δακτύλιος με χορδές	$u * (\frac{n + k^2 - k}{4k} - n \bmod 4k)$
Ανεμιστήρα	u
Σκάλα	$u * \frac{n}{4}$

6.1 Τοπολογία – Αυστηρό Φράγμα

6.3 Μελλοντική Εργασία

Η παρούσα εργασία θα μπορούσε να επεκταθεί βρίσκοντας και άλλες τοπολογίες δικτύων στις οποίες ο αλγόριθμος μέσου όρου είναι βέλτιστος με αυστηρό φράγμα συγχρονισμού ρολογιών $\frac{u \cdot \text{διάμετρος}}{2}$. Η διάμετρος των τοπολογιών απαιτείται να υπολογισθεί έτσι ώστε να βρεθεί ακριβώς η λοξότητα ανάμεσα στα προσαρμοσμένα ρολόγια των επεξεργασιών των δικτύων με το τέλος της εκτέλεσης του βέλτιστου αυτού αλγορίθμου.

Βιβλιογραφία

- [1] Hagit Attiya, J. W. (April 2004). *Distributed Computing: Fundamentals, Simulations, and Advanced Topics* (Second ed.). Canada: A. John Wiley & Sons.
- [2] Saâd Biaz, J. L. (2001). Closed form bounds for clock synchronization under simple uncertainty assumptions. *Information Processing Letters*, 80(3), 151-157.
- [3] Ante Ćustić, R. Z. (2016). The Quadratic Minimum Spanning Tree Problem and its Variations. 1-13.
- [4] Cristian, F. (1989), "Probabilistic clock synchronization", *Distributed Computing*, Springer, 3 : 146–158
- [5] Gusella, R.; Zatti, S. (1989), "The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD", *Software Engineering, IEEE Transactions on*, IEEE, 15 (7): 847–853