



School of
Electrical &
Computer
Engineering

Επεξεργασία και Διαχείριση Δεδομένων σε Δίκτυα Αισθητήρων *ΠΛΗ 516*

Αναφορά δεύτερης φάσης εργασίας στο tinyOS
Επέκταση της λειτουργικότητας του TAG, και υλοποίηση του TiNA

14/12/2018

Γιακουμάκης Πάρις-Παύλος

Γκιώνης Νίκος

Εισαγωγή

Η δεύτερη φάση της άσκησης προέβλεπε την επέκταση της λειτουργικότητας του TAG που υλοποιήθηκε στην πρώτη φάση, ώστε να υπολογίζει τελικά τα επερωτήματα SUM, AVG, MAX, MIN, COUNT και VARIANCE. Επιπλέον ζητούνταν η υλοποίηση του TiNA, ώστε να υπολογίζει τα επερωτήματα SUM, MAX, MIN, COUNT. Το πρόγραμμα έπρεπε να επιλέγει τυχαία σε κάθε εκτέλεση, εάν θα τρέξει τον κώδικα του extended TAG ή του TiNA, αλλά και ποιο επερωτήμα θα υπολογίσει. Επίσης στην περίπτωση του extended TAG, επέλεγε τυχαία τον αριθμό των επερωτημάτων που πρέπει να υπολογιστούν ταυτόχρονα (μεταξύ ενός και δύο). Σημαντική παράμετρος ήταν η ελαχιστοποίηση της μεταδιδόμενης πληροφορίας, σε κάθε τύπου μετάδοση στο δίκτυο.

Διαδικασία Routing

Ο χρονισμός του Routing γίνεται με τον ίδιο τρόπο, όπως και στην πρώτη φάση. Σημαντική διαφορά είναι ότι, επειδή υπάρχουν τρεις εναλλακτικές περιπτώσεις εκτέλεσης (TiNA, Extended TAG με ένα επερωτήμα, extended TAG με 2 επερωτήματα), τα routing μηνύματα έπρεπε να φτιαχτούν κατάλληλα, ώστε να μη στέλνεται περιττή πληροφορία.

Στο event RoutingMsgTimer.fired, ο κόμβος μηδέν αποφασίζει αρχικά με κλίση μιας random συνάρτησης, εάν θα τρέξει TiNA ή extended TAG. Σχετικά με την random, προκειμένου να δίνει διαφορετικά νούμερα σε κάθε εκτέλεση, έπρεπε να αλλάξει το seed της, το οποίο ήταν προκαθορισμένο ως το TOS_NODE_ID. Για το σκοπό αυτό χρησιμοποιήθηκε ως seed πληροφορία από ένα αρχείο του λειτουργικού συστήματος[1] σε συνδυασμό με το TOS_NODE_ID, που εισήγαγε τυχαιότητα με βάση την ώρα του συστήματος.

Routing για Extended TAG με ένα επερωτήμα

Εάν ο κόμβος 0 αποφασίσει να τρέξει extended TAG, αμέσως μετά, επιλέγει τυχαία εάν θα υπολογίσει ένα ή δύο επερωτήματα. Η πιο απλή περίπτωση είναι με ένα επερωτήμα. Στην περίπτωση αυτή, καλεί εκ νέου μια συνάρτηση random που αποφασίζει ποιο επερωτήμα από τα έξι πιθανά θα εκτελέσει. Αφού αποφασιστούν τα παραπάνω, στη συνέχεια δημιουργείται το Routing μήνυμα προς αποστολή. Το μήνυμα στην περίπτωση αυτή (Routing3field) χρειαζόταν να έχει τρία μόνο πεδία. Ένα πεδίο που θα ενημερώνει για το εάν επιλέχθηκε TiNA ή extended TAG (mode), ένα πεδίο που θα ενημερώνει για το ποιο επερωτήμα αποφασίστηκε (select) και ένα πεδίο που θα ενημερώνει για το βάθος του αποστολέα (depth). Στη συνέχεια η διαδικασία αποστολής του routing μηνύματος συνεχίζεται, όπως και στην πρώτη φάση της εργασίας.

Routing για Extended TAG με δύο επερωτήματα και για TiNA

Οι άλλες περιπτώσεις, δηλαδή το TiNA και το extended TAG με δύο επερωτήματα, χρειάζονται την ίδια πληροφορία που αναλύθηκε παραπάνω για το extended TAG με ένα επερωτήμα, αλλά και ένα πεδίο επιπλέον. Συγκεκριμένα, για την περίπτωση του TiNA χρειάζεται να μεταδοθεί το tct, το οποίο επίσης επιλέγεται τυχαία, από 0 έως 50, ενώ για το extended TAG με δύο επερωτήματα, χρειάζεται να μεταδοθεί ο κωδικός του δεύτερου επερωτήματος που πρέπει να

υπολογιστεί. Συνεπώς, και για τις δύο αυτές περιπτώσεις χρησιμοποιείται κοινή δομή για το μήνυμα routing (Routing4field), το οποίο έχει ένα επιπλέον πεδίο (select2ortct). Εάν εκτελείται TiNA, τότε σε αυτό το πεδίο εισέρχεται το tct, ενώ αν εκτελείται extended TAG δύο επερωτημάτων, εισέρχεται ο κωδικός του δεύτερου επερωτήματος.

Επίσης, έγινε μία τροποποίηση στο receiveRoutingTask. Ο κάθε κόμβος, όταν λαμβάνει ένα routing μήνυμα, εάν δεν έχει ακόμη πατέρα, πρέπει αρχικά να θέσει ως πατέρα του τον αποστολέα, και στη συνέχεια να πάρει την υπόλοιπη πληροφορία από το μήνυμα. Για να γίνει αυτό, ελέγχει αρχικά με βάση το size του μηνύματος, να δει ποιον τύπο routingmsg έλαβε. Στην περίπτωση του Routing4field, λαμβάνει πρώτα σε μία global μεταβλητή το mode, και με βάση αυτό, αποκρυπτογραφεί εάν το πεδίο select2ortct αφορά σε τιμή tct ή στον κωδικό του δεύτερου επερωτήματος. Η περίπτωση του Routing3field είναι απλή, καθώς κάθε πεδίο αντιγράφεται στην αντίστοιχη global μεταβλητή του κόμβου. Να σημειωθεί επίσης ότι κάθε κόμβος έχει έναν πίνακα δύο θέσεων (select[]), στον οποίο κρατάει το επερώτημα προς υπολογισμό στην πρώτη θέση, ενώ στη δεύτερη θέση, εισάγεται το δεύτερο επερώτημα στην περίπτωση του extended TAG δύο επερωτημάτων. Αλλιώς η θέση αυτή μένει κενή. Τέλος, να αναφερθεί ότι, καθώς τα πεδία αυτά είναι απλοί ακέραιοι αριθμοί, τα πεδία των μηνυμάτων έχουν το μικρότερο δυνατό μέγεθος, δηλαδή nx_uint8_t. Το πεδίο mode, θα μπορούσε να είναι και Boolean μεταβλητή, ωστόσο δεν βρέθηκε τέτοιος τύπος δεδομένων (nx_bool).

Διαδικασία αποστολής μετρήσεων

Μετά την ολοκλήρωση του Routing, κάθε κόμβος έχει ενημερωθεί για το mode που καλείται να εκτελέσει και το επερώτημα που πρέπει να υπολογίσει. Αρχικά προκύπτει το event sendMeasTimer.fired, μέσα στο οποίο κάθε κόμβος λαμβάνει μία τυχαία μέτρηση, από 0 έως 50. Στη συνέχεια κάνει enqueue τη μέτρησή του αυτή, προκειμένου να την περάσει στο αντίστοιχο task, που θα κάνει την περαιτέρω επεξεργασία. Έτσι, στη συνέχεια δρομολογείται το sendMeasTask.

Μέσα στο task, αρχικά λαμβάνεται με dequeue η μέτρηση του κόμβου. Στη συνέχεια ξεκινάει ο έλεγχος για το τι mode και επερωτήματα απαιτούνται. Στο σημείο αυτό είναι χρήσιμο να αναφερθούν οι δομές που δημιουργήθηκαν για την αποστολή μετρήσεων. Να σημειωθεί ότι το TiNA δεν χρειάζεται ειδικές δομές (διαφορετικού τύπου) σε σχέση με το extended TAG, καθώς η διαφοροποίησή τους έγκειται στο κατά πόσον θα περάσει ή όχι η νέα προς αποστολή τιμή, κατά tct%, την προηγούμενη τιμή που αποστάλθηκε. Επομένως, για τα επερωτήματα που είναι κοινά, τόσο το TiNA, όσο και το extended TAG εκμεταλλεύονται τις ίδιες δομές. Επίσης να αναφερθεί ότι, κάθε επερώτημα χρειάζεται το αντίστοιχο μέγεθος. Συγκεκριμένα, με πεδίο uint8_t εξυπηρετούνται αριθμοί μέχρι το 255, άρα οι μετρήσεις MIN, MAX, COUNT. Με πεδίο uint16_t εξυπηρετούνται μετρήσεις μέχρι 65535, άρα το SUM. Το SUM μεγιστοποιείται στην περίπτωση των 64 κόμβων, όταν κάθε ένας πάρει τη μέγιστη τιμή (50). Τότε είναι $64 \cdot 50 = 3200$. Τέλος, για το πεδίο SUMSQ, που απαιτείται για τον υπολογισμό του VARIANCE, η χειρότερη περίπτωση είναι κάθε κόμβος να παίρνει μέγιστη μέτρηση. Τότε, ο κάθε κόμβος συναθροίζει το

SUMSQ που έλαβε από τα παιδιά του, με την δική του μέτρηση στο τετράγωνο. Στην χειρότερη περίπτωση επομένως, προκύπτει $SUMSQ = 64 \cdot 50^2 = 160000$. Συνεπώς χρειάζεται ένα πεδίο `uint32_t` για την αποθήκευσή του, το οποίο αποθηκεύει μέχρι τον αριθμό 4.294.967.295.

Συνολικά υλοποιήθηκαν 7 δομές:

- **Δομή OneMeas8bit:** Περιέχει ένα πεδίο των 8 bit. Άρα εξυπηρετεί τα επερωτήματα MIN, MAX, COUNT.
- **Δομή OneMeas16bit:** Περιέχει ένα πεδίο των 16bit. Άρα εξυπηρετεί το επερώτημα SUM.
- **Δομή TwoMeas8bit:** Περιέχει δύο πεδία των 8bit. Άρα εξυπηρετεί τα επερωτήματα MAX, MIN, COUNT σε ζεύγη των δύο. Η δομή αυτή έχει μία ιδιαιτερότητα, και συνεπώς χρειάζεται να μεταδοθεί επιπλέον πληροφορία. Θα εξηγηθεί στη συνέχεια.
- **Δομή TwoMeasMixedbit:** Περιέχει ένα πεδίο των 16bit και ένα πεδίο των 8bit. Εξυπηρετεί το επερώτημα AVG, το ζεύγος AVG-COUNT, αλλά και τις περιπτώσεις όπου το SUM συνδυάζεται με οποιοδήποτε εκ των MIN, MAX, COUNT, AVG.
- **Δομή ThreeMeasMixedbit:** Περιέχει ένα πεδίο 16bit, και δύο πεδία των 8bit. Εξυπηρετεί τις περιπτώσεις AVG-MAX, AVG-MIN.
- **Δομή VarMeasSimple:** Περιέχει ένα πεδίο 32bit, ένα πεδίο 16bit και ένα πεδίο 8bit. Εξυπηρετεί όλες τις περιπτώσεις που ζητείται VARIANCE, εκτός από την περίπτωση που συνδυάζεται με MIN ή MAX.
- **Δομή VarMeasDouble:** Περιέχει ένα πεδίο 32bit, ένα πεδίο 16bit και δύο πεδία 8bit. Εξυπηρετεί τις περιπτώσεις VAR-MIN, VAR-MAX.

Επίσης δημιουργήθηκε μία συνάρτηση (`calculateQuery`) η οποία παίρνει σαν όρισμα τον κωδικό του επερωτήματος για το οποίο πρέπει να υπολογίσει ο κόμβος τη συναθροιστική τιμή του, καθώς και τη μέτρηση του κόμβου, την οποία συναθροίζει με τις τιμές των παιδιών του. Τις τιμές αυτές τις λαμβάνει διατρέχοντας τον πίνακα στον οποίο αποθηκεύει ο κόμβος τις τιμές κάθε παιδιού.

Ο κόμβος πρέπει να αποφασίσει αρχικά εντός του task, σε ποιο mode βρίσκεται. Στη συνέχεια, για να ξεχωρίσει τις περιπτώσεις extended TAG ενός επερωτήματος και extended TAG δύο επερωτημάτων, χρησιμοποιεί τη δεύτερη θέση του global πίνακα `select[]`, η οποία εάν είναι άδεια, σημαίνει ότι εκτελείται ένα επερώτημα. Η περίπτωση του TiNA μαζί με το extended TAG ενός επερωτήματος, είναι σχετικά απλή, καθώς αναλόγως του επερωτήματος, τίθεται ως payload του μηνύματος η κατάλληλη δομή, υπολογίζεται μέσω της συνάρτησης που προαναφέρθηκε η συναθροιστική τιμή, και ενημερώνονται τα πεδία του μηνύματος. Το TiNA ελέγχει επιπλέον με χρήση μίας μεταβλητής `previousQuery`, εάν η νέα συναθροιστική τιμή περνά κατά `tct%` την παλιά. Εάν ναι, τότε ανανεώνει τη μεταβλητή `previousQuery` και ετοιμάζει το μήνυμα. Αλλιώς δεν κάνει τίποτα. Να σημειωθεί ότι κατά τον πρώτο γύρο, όπου η μεταβλητή `previousQuery` δεν

έχει κάποιο νόημα, χρησιμοποιείται ως flag ο round counter, ο οποίος ακυρώνει πρακτικά τον έλεγχο του tct, μόνο στην περίπτωση που η τιμή του ισούται με 1.

Η πιο σύνθετη περίπτωση είναι αυτή του extended TAG με δύο επερωτήματα. Με παρόμοιο τρόπο, αναλόγως των επερωτημάτων που πρέπει να υπολογιστούν, τίθεται ως payload του μηνύματος η κατάλληλη δομή, υπολογίζονται οι συναθροιστικές τιμές με κλήσεις της συνάρτησης calculateQuery, και ενημερώνονται τα κατάλληλα πεδία του μηνύματος. Μία περίπτωση, όπως προαναφέρθηκε, εμφανίζει παραπάνω δυσκολίες. Αυτή είναι ο υπολογισμός των MAX, MIN, COUNT σε ζεύγη των δύο. Ο λόγος είναι ότι δε μπορούμε να γνωρίζουμε από πριν ποια επερωτήματα θα ζητηθούν εντός της συγκεκριμένης περίπτωσης. Για παράδειγμα, κατά τον υπολογισμό του AVG και του MAX/MIN, γνωρίζουμε από πριν ότι το ένα 16bit θα είναι σίγουρα το SUM, το ένα 8bit θα είναι σίγουρα το COUNT και το δεύτερο 8bit θα είναι είτε MAX είτε MIN. Οπότε κατά τη λήψη του μηνύματος μπορεί να αποκωδικοποιηθεί το εάν έχω MAX ή MIN, ελέγχοντας εάν σε κάποια εκ των δύο select θέσεων έχω MAX ή MIN αντίστοιχα. Στην περίπτωση των δύο 8bit, και τα δύο πεδία δεν είναι προκαθορισμένα. Ως εκ τούτου, είναι αναγκαία η αποστολή επιπλέον πληροφορίας, υπό την μορφή πίνακα δύο θέσεων. Στις θέσεις αυτές τοποθετείται, ανάλογα με τη συναθροιστική συνάρτηση που θα εμφανιστεί στο κάθε select, ο αντίστοιχος κωδικός του επερωτήματος που υπολογίζεται.

Τέλος, ανεξαρτήτως υποπερίπτωσης, ελέγχεται εάν είμαστε στο mode extended TAG ή εάν είμαστε στο TiNA και η νέα μέτρηση περνά το tct. Τότε το μήνυμα που ετοιμάστηκε παραπάνω, αποστέλλεται στον πατέρα. Σε αντίθετη περίπτωση, δηλαδή εάν στο TiNA δεν υπερβεί το tct η νέα μέτρηση, δεν στέλνεται μήνυμα.

Διαδικασία λήψης μετρήσεων

Όταν ένας κόμβος λάβει ένα μήνυμα μετρήσεων, προκύπτει το event MeasReceive.receive. Εκεί ο κόμβος-παραλήπτης, κάνει enqueue το μήνυμα που έλαβε, ώστε να το επεξεργαστεί στη συνέχεια το receiveMeasTask.

Το task αρχικά κάνει dequeue το μήνυμα. Στη συνέχεια, με ακριβώς τον ίδιο τρόπο με το sendMeasTask, εντοπίζει σε ποια περίπτωση βρίσκεται. Έπειτα, λαμβάνει σε τοπική μεταβλητή κατάλληλου τύπου το payload του μηνύματος και με βάση αυτή, ενημερώνει τα αντίστοιχα πεδία του πίνακα, στον οποίο κρατά τις τιμές των παιδιών του. Ο πίνακας αυτός αποτελείται από δομές τύπου ChildInfo, όπου για κάθε παιδί υπάρχει πεδίο για κάθε πιθανό επερωτήμα που μπορεί να προκύψει. Για την ιδιόμορφη περίπτωση των δύο 8bit τιμών, αξιοποιείται ο πίνακας που προαναφέρθηκε, ο οποίος αποστάλθηκε μαζί με το μήνυμα από το παιδί. Συγκεκριμένα, με βάση το περιεχόμενο αυτού του πίνακα, επιλέγεται σωστά το πεδίο του πίνακα για το παιδί-αποστολέα, που πρέπει να ενημερωθεί.

Παραδείγματα εκτελέσεων

Ακολουθως παρουσιάζονται screenshots και επεξηγήσεις σε ενδεικτικά παραδείγματα εκτέλεσης του προγράμματος. Αρχικά παρουσιάζεται η λειτουργία του TiNA, και στη συνέχεια του extended TAG με δύο επερωτήματα.

Παράδειγμα λειτουργίας TiNA

Με τοπολογία 3x3 και εμβέλεια κόμβων 1, εκτελείται ένα επερώτημα με βάση το TiNA. Όπως φαίνεται στην ακόλουθη εικόνα, αρχικά υπάρχει η ένδειξη ότι επιλέχθηκε ως mode το TiNA. Στη συνέχεια εμφανίζεται η επιλογή που έγινε για το επερώτημα προς υπολογισμό, όπου έτυχε ο κωδικός 1. Ο κωδικός αυτός αντιστοιχεί στο SUM. Επίσης εμφανίζεται το tct, το οποίο στη συγκεκριμένη περίπτωση είναι 27%. Τέλος εμφανίζονται τα αποτελέσματα του routing, όπου κάθε κόμβος εμφανίζει το βάθος, τον πατέρα του, το tct και τον κωδικό του επερωτήματος προς υπολογισμό, όπως τα έλαβε από το routing μήνυμα.

```
27 0:0:10.0000000000 DEBUG (0): curdepth = 0 , parentID= 0
28 0:0:10.0000000000 DEBUG (0):
29
30 MODE(extended -> 0, Tina -> 1 ): 1
31
32 0:0:10.0000000000 DEBUG (1): curdepth = 255 , parentID= 65535
33 0:0:10.0000000000 DEBUG (2): curdepth = 255 , parentID= 65535
34 0:0:10.0000000000 DEBUG (3): curdepth = 255 , parentID= 65535
35 0:0:10.0000000000 DEBUG (4): curdepth = 255 , parentID= 65535
36 0:0:10.0000000000 DEBUG (5): curdepth = 255 , parentID= 65535
37 0:0:10.0000000000 DEBUG (6): curdepth = 255 , parentID= 65535
38 0:0:10.0000000000 DEBUG (7): curdepth = 255 , parentID= 65535
39 0:0:10.0000000000 DEBUG (8): curdepth = 255 , parentID= 65535
40 0:0:10.0000000100 DEBUG (0): Radio initialized successfully!!!
41 0:0:10.0000000100 DEBUG (1): Radio initialized successfully!!!
42 0:0:10.0000000100 DEBUG (2): Radio initialized successfully!!!
43 0:0:10.0000000100 DEBUG (3): Radio initialized successfully!!!
44 0:0:10.0000000100 DEBUG (4): Radio initialized successfully!!!
45 0:0:10.0000000100 DEBUG (5): Radio initialized successfully!!!
46 0:0:10.0000000100 DEBUG (6): Radio initialized successfully!!!
47 0:0:10.0000000100 DEBUG (7): Radio initialized successfully!!!
48 0:0:10.0000000100 DEBUG (8): Radio initialized successfully!!!
49 0:0:10.2500000100 DEBUG (0): Query selected for TINA is: 1 with tct: 27
50 0:0:10.2500000100 DEBUG (0): #####
51 0:0:10.2500000100 DEBUG (0): ##### ROUND 1 #####
52 0:0:10.2500000100 DEBUG (0): #####
53 0:0:10.2566375640 DEBUG (3): New parent for NodeID= 3 : curdepth= 1 , parentID= 0
54 0:0:10.2566375640 DEBUG (3): tinaSelect= 1 , tct=27
55 0:0:10.2566375640 DEBUG (1): New parent for NodeID= 1 : curdepth= 1 , parentID= 0
56 0:0:10.2566375640 DEBUG (1): tinaSelect= 1 , tct=27
57 0:0:10.5115966760 DEBUG (6): New parent for NodeID= 6 : curdepth= 2 , parentID= 3
58 0:0:10.5115966760 DEBUG (6): tinaSelect= 1 , tct=27
59 0:0:10.5115966760 DEBUG (4): New parent for NodeID= 4 : curdepth= 2 , parentID= 3
60 0:0:10.5115966760 DEBUG (4): tinaSelect= 1 , tct=27
61 0:0:10.5146026400 DEBUG (2): New parent for NodeID= 2 : curdepth= 2 , parentID= 1
62 0:0:10.5146026400 DEBUG (2): tinaSelect= 1 , tct=27
63 0:0:10.7666931110 DEBUG (7): New parent for NodeID= 7 : curdepth= 3 , parentID= 6
64 0:0:10.7666931110 DEBUG (7): tinaSelect= 1 , tct=27
65 0:0:10.7669220080 DEBUG (5): New parent for NodeID= 5 : curdepth= 3 , parentID= 2
66 0:0:10.7669220080 DEBUG (5): tinaSelect= 1 , tct=27
67 0:0:11.0199432480 DEBUG (8): New parent for NodeID= 8 : curdepth= 4 , parentID= 5
68 0:0:11.0199432480 DEBUG (8): tinaSelect= 1 , tct=27
```

Figure 1: TiNA Routing

Με βάση την παραπάνω εικόνα, προκύπτει ότι το δέντρο δρομολόγησης των μετρήσεων από τα φύλλα προς τη ρίζα είναι το εξής:

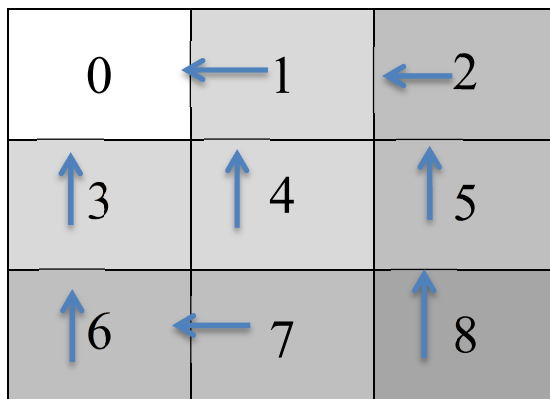


Table 1: TiNA - Δένδρο Δρομολόγησης

Η αποστολή μηνυμάτων, σύμφωνα με το δέντρο δρομολόγησης, αλλά και με το ακόλουθο screenshot γίνεται ως εξής: Αρχικά ξεκινάει ο κόμβος 8, ο οποίος βρίσκεται στο μεγαλύτερο βάθος. Μετρά 28, και το στέλνει στον κόμβο 5. Ο κόμβος 5 το λαμβάνει, και στη συνέχεια είναι η σειρά του να αποστείλει, αφού ανήκει στο αμέσως επόμενο επίπεδο. Μετρά 17, συναθροίζει με την τιμή που έλαβε από το παιδί του, και στέλνει στον κόμβο 2 τελικά τη μέτρηση 45. Στη συνέχεια στέλνει και ο κόμβος 7, που ανήκει στο ίδιο επίπεδο, την τιμή 14. Έπειτα δρομολογούνται οι κόμβοι με βάθος 2. Αρχικά ο κόμβος 2 λαμβάνει τη μέτρηση από τον κόμβο 5, μετράει ο ίδιος την τιμή 7, και τελικά προωθεί την τιμή 52. Όμοια ο κόμβος 6, λαμβάνει την τιμή από τον κόμβο 7, συναθροίζει με την δική του μέτρηση, και στέλνει την τιμή 32. Ο κόμβος 4, τέλος, μετράει 17 και στέλνει τη μέτρησή του. Στη συνέχεια δρομολογούνται οι κόμβοι βάθους 1, δηλαδή ο 1 και ο 3. Αυτοί λαμβάνουν τις τιμές των παιδιών τους, και αφού λάβουν και οι ίδιοι μέτρηση, συναθροίζουν και στέλνουν προς τη ρίζα τις τιμές 69 και 68 αντίστοιχα. Τέλος, η ρίζα λαμβάνει τις τιμές αυτές, και η δική της μέτρηση είναι 19. Ως εκ τούτου, το τελικό αποτέλεσμα που εκτυπώνει η ρίζα για το SUM είναι: $19+69+68 = 159$.

Με τον τρόπο αυτό ολοκληρώνεται ο πρώτος γύρος υπολογισμού του ερωτήματος SUM με TiNA. Όπως προαναφέρθηκε, στον πρώτο γύρο, όλοι οι κόμβοι στέλνουν υποχρεωτικά, καθώς δεν υπάρχει προηγούμενη μέτρηση για να γίνει σύγκριση, αξιοποιώντας το tct.


```

78 0:0:58.773437510 DEBUG (8): NodeID = 8 curdepth= 4
79 0:0:58.773437510 DEBUG (8): Starting Data transmission to parent!
80 0:0:58.773437510 DEBUG (8): measurement is: 28
81 0:0:58.773437520 DEBUG (8): Node's SUM is: 28
82 0:0:58.773437520 DEBUG (8): Measurement passes tct! Previous query: 0, Now sent: 28
83 0:0:58.781738253 DEBUG (5): Measurement received from: 8
84 0:0:58.781738263 DEBUG (5): Received from childID: 8 - sum:28
85 0:0:59.014648448 DEBUG (5): NodeID = 5 curdepth= 3
86 0:0:59.014648448 DEBUG (5): Starting Data transmission to parent!
87 0:0:59.014648448 DEBUG (5): measurement is: 17
88 0:0:59.014648458 DEBUG (5): ChildID: 8 has sum: 28
89 0:0:59.014648458 DEBUG (5): Node's SUM is: 45
90 0:0:59.014648458 DEBUG (5): Measurement passes tct! Previous query: 0, Now sent: 45
91 0:0:59.020507823 DEBUG (7): NodeID = 7 curdepth= 3
92 0:0:59.020507823 DEBUG (7): Starting Data transmission to parent!
93 0:0:59.020507823 DEBUG (7): measurement is: 14
94 0:0:59.020507833 DEBUG (7): Node's SUM is: 14
95 0:0:59.020507833 DEBUG (7): Measurement passes tct! Previous query: 0, Now sent: 14
96 0:0:59.023483245 DEBUG (2): Measurement received from: 5
97 0:0:59.023483255 DEBUG (2): Received from childID: 5 - sum:45
98 0:0:59.025009149 DEBUG (6): Measurement received from: 7
99 0:0:59.025009159 DEBUG (6): Received from childID: 7 - sum:14
100 0:0:59.255859385 DEBUG (2): NodeID = 2 curdepth= 2
101 0:0:59.255859385 DEBUG (2): Starting Data transmission to parent!
102 0:0:59.255859385 DEBUG (2): measurement is: 7
103 0:0:59.255859395 DEBUG (2): ChildID: 5 has sum: 45
104 0:0:59.255859395 DEBUG (2): Node's SUM is: 52
105 0:0:59.255859395 DEBUG (2): Measurement passes tct! Previous query: 0, Now sent: 52
106 0:0:59.261718760 DEBUG (4): NodeID = 4 curdepth= 2
107 0:0:59.261718760 DEBUG (4): Starting Data transmission to parent!
108 0:0:59.261718760 DEBUG (4): measurement is: 17
109 0:0:59.261718770 DEBUG (4): Node's SUM is: 17
110 0:0:59.261718770 DEBUG (4): Measurement passes tct! Previous query: 0, Now sent: 17
111 0:0:59.264404301 DEBUG (3): Measurement received from: 4
112 0:0:59.264404311 DEBUG (3): Received from childID: 4 - sum:17
113 0:0:59.266418415 DEBUG (1): Measurement received from: 2
114 0:0:59.266418425 DEBUG (1): Received from childID: 2 - sum:52
115 0:0:59.267578135 DEBUG (6): NodeID = 6 curdepth= 2
116 0:0:59.267578135 DEBUG (6): Starting Data transmission to parent!
117 0:0:59.267578135 DEBUG (6): measurement is: 18
118 0:0:59.267578145 DEBUG (6): ChildID: 7 has sum: 14
119 0:0:59.267578145 DEBUG (6): Node's SUM is: 32
120 0:0:59.267578145 DEBUG (6): Measurement passes tct! Previous query: 0, Now sent: 32
121 0:0:59.278320270 DEBUG (3): Measurement received from: 6
122 0:0:59.278320280 DEBUG (3): Received from childID: 6 - sum:32
123 0:0:59.502929697 DEBUG (1): NodeID = 1 curdepth= 1
124 0:0:59.502929697 DEBUG (1): Starting Data transmission to parent!
125 0:0:59.502929697 DEBUG (1): measurement is: 17
126 0:0:59.502929707 DEBUG (1): ChildID: 2 has sum: 52
127 0:0:59.502929707 DEBUG (1): Node's SUM is: 69
128 0:0:59.502929707 DEBUG (1): Measurement passes tct! Previous query: 0, Now sent: 69
129 0:0:59.506591795 DEBUG (0): Measurement received from: 1
130 0:0:59.506591805 DEBUG (0): Received from childID: 1 - sum:69
131 0:0:59.508789072 DEBUG (3): NodeID = 3 curdepth= 1
132 0:0:59.508789072 DEBUG (3): Starting Data transmission to parent!
133 0:0:59.508789072 DEBUG (3): measurement is: 19
134 0:0:59.508789082 DEBUG (3): ChildID: 4 has sum: 17
135 0:0:59.508789082 DEBUG (3): ChildID: 6 has sum: 32
136 0:0:59.508789082 DEBUG (3): Node's SUM is: 68
137 0:0:59.508789082 DEBUG (3): Measurement passes tct! Previous query: 0, Now sent: 68
138 0:0:59.511642459 DEBUG (0): Measurement received from: 3
139 0:0:59.511642469 DEBUG (0): Received from childID: 3 - sum:68
140 0:0:59.750000010 DEBUG (0): NodeID = 0 curdepth= 0
141 0:0:59.750000010 DEBUG (0): measurement is: 22
142 0:0:59.750000020 DEBUG (0): ChildID: 1 has sum: 69
143 0:0:59.750000020 DEBUG (0): ChildID: 3 has sum: 68
144 0:0:59.750000020 DEBUG (0): Node's SUM is: 159
145 0:0:59.750000020 DEBUG (0):
146 0:0:59.750000020 DEBUG (0): FINAL RESULTS: SUM: 159
147 0:0:59.750000020 DEBUG (0):

```

Figure 2: TiNA Round 1

Στη συνέχεια εμφανίζεται ένα screenshot από το γύρο 2. Ο κόμβος 2, υπολογίζει τιμή προς αποστολή στον δεύτερο γύρο τον αριθμό 44. Όμως, η προηγούμενη τιμή που είχε αποστείλει είναι η 52. Ελέγχοντας το tct προκύπτει ότι $52 * tct = 52 * 27\% = 14.04$, και $54 - 14.04 = 40$. Επομένως, η νέα μέτρηση βρίσκεται εντός του tct ορίου της προηγούμενης μέτρησης, και για το λόγο αυτό η τιμή αυτή δε στέλνεται.

```
166 0:1:59.020507823 DEBUG (7): NodeID = 7 curdepth= 3
167 0:1:59.020507823 DEBUG (7): Starting Data transmission to parent!
168 0:1:59.020507823 DEBUG (7): measurement is: 41
169 0:1:59.020507833 DEBUG (7): Node's SUM is: 41
170 0:1:59.020507833 DEBUG (7): Measurement passes tct! Previous query: 14, Now sent: 41
171 0:1:59.029846157 DEBUG (6): Measurement received from: 7
172 0:1:59.029846167 DEBUG (6): Received from childID: 7 - sum:41
173 0:1:59.255859385 DEBUG (2): NodeID = 2 curdepth= 2
174 0:1:59.255859385 DEBUG (2): Starting Data transmission to parent!
175 0:1:59.255859385 DEBUG (2): measurement is: 24
176 0:1:59.255859395 DEBUG (2): ChildID: 5 has sum: 20
177 0:1:59.255859395 DEBUG (2): Node's SUM is: 44
178 0:1:59.255859395 DEBUG (2): Measurement doesn't pass tct! Previous query: 52, Now measured: 44
179 0:1:59.255859395 DEBUG (2): Doesn't send because of tct
```

Figure 3: TiNA Round 2

Επίσης, για να ολοκληρωθεί η εικόνα της λειτουργίας του TiNA, παρουσιάζεται ένα screenshot από τον τέταρτο γύρο, κατά τον οποίο η τιμή προς αποστολή του κόμβου 2 ξεπερνά το tct. Συγκεκριμένα, ο κόμβος υπολογίζει τιμή 95. Η τιμή με την οποία συγκρίνεται το 95, όπως φαίνεται στην εικόνα, είναι η τελευταία τιμή που έστειλε ο κόμβος 2, δηλαδή το 52. Προφανώς το 95 περνά το tct, οπότε ο κόμβος 2 προωθεί σε αυτή την περίπτωση τη νέα τιμή στον πατέρα του.

```
315 0:3:59.255859385 DEBUG (2): Starting Data transmission to parent!
316 0:3:59.255859385 DEBUG (2): measurement is: 6
317 0:3:59.255859395 DEBUG (2): ChildID: 5 has sum: 89
318 0:3:59.255859395 DEBUG (2): Node's SUM is: 95
319 0:3:59.255859395 DEBUG (2): Measurement passes tct! Previous query: 52, Now sent: 95
```

Figure 4: TiNA Round 3

Παράδειγμα λειτουργίας Extended TAG

Στο παρακάτω παράδειγμα υπολογίζονται με extended TAG δύο ερωτήματα ταυτόχρονα. Η τοπολογία που δημιουργήθηκε ήταν 4x4 με εμβέλεια 1.5. Όπως φαίνεται στην ακόλουθη εικόνα, ο κόμβος μηδέν αποφασίζει ότι τα δύο ερωτήματα που θα υπολογιστούν είναι τα 6 και 5, που αντιστοιχούν στα VAR και AVG. Η περίπτωση αυτή είναι χαρακτηριστική, καθώς συμπεριλαμβάνει τον υπολογισμό της νέας συναθροιστικής συναρτησης VARIANCE. Μέσω των routing μηνυμάτων, όπως φαίνεται, διαδίδεται η πληροφορία των ερωτήσεων που καλείται να υπολογίσει το δίκτυο, σε όλους τους κόμβους.

```
123 0:0:10.250000010 DEBUG (0): We have 2 queries!
124 0:0:10.250000010 DEBUG (0): Query 1 is: 6
125 0:0:10.250000010 DEBUG (0): Query 2 is: 5
126 0:0:10.250000010 DEBUG (0): #####
127 0:0:10.250000010 DEBUG (0): ##### ROUND 1 #####
128 0:0:10.250000010 DEBUG (0): #####
129 0:0:10.254745485 DEBUG (5): New parent for NodeID= 5 : curdepth= 1 , parentID= 0
130 0:0:10.254745485 DEBUG (5): ExtendSelect1= 6, ExtendSelect2= 5
131 0:0:10.254745485 DEBUG (4): New parent for NodeID= 4 : curdepth= 1 , parentID= 0
132 0:0:10.254745485 DEBUG (4): ExtendSelect1= 6, ExtendSelect2= 5
133 0:0:10.254745485 DEBUG (1): New parent for NodeID= 1 : curdepth= 1 , parentID= 0
134 0:0:10.254745485 DEBUG (1): ExtendSelect1= 6, ExtendSelect2= 5
135 0:0:10.506744398 DEBUG (8): New parent for NodeID= 8 : curdepth= 2 , parentID= 5
136 0:0:10.506744398 DEBUG (8): ExtendSelect1= 6, ExtendSelect2= 5
137 0:0:10.506744398 DEBUG (6): New parent for NodeID= 6 : curdepth= 2 , parentID= 5
138 0:0:10.506744398 DEBUG (6): ExtendSelect1= 6, ExtendSelect2= 5
139 0:0:10.506744398 DEBUG (10): New parent for NodeID= 10 : curdepth= 2 , parentID= 5
140 0:0:10.506744398 DEBUG (10): ExtendSelect1= 6, ExtendSelect2= 5
141 0:0:10.506744398 DEBUG (2): New parent for NodeID= 2 : curdepth= 2 , parentID= 5
142 0:0:10.506744398 DEBUG (2): ExtendSelect1= 6, ExtendSelect2= 5
143 0:0:10.506744398 DEBUG (9): New parent for NodeID= 9 : curdepth= 2 , parentID= 5
144 0:0:10.506744398 DEBUG (9): ExtendSelect1= 6, ExtendSelect2= 5
145 0:0:10.757095360 DEBUG (7): New parent for NodeID= 7 : curdepth= 3 , parentID= 6
146 0:0:10.757095360 DEBUG (7): ExtendSelect1= 6, ExtendSelect2= 5
147 0:0:10.757095360 DEBUG (11): New parent for NodeID= 11 : curdepth= 3 , parentID= 6
148 0:0:10.757095360 DEBUG (11): ExtendSelect1= 6, ExtendSelect2= 5
149 0:0:10.757095360 DEBUG (3): New parent for NodeID= 3 : curdepth= 3 , parentID= 6
150 0:0:10.757095360 DEBUG (3): ExtendSelect1= 6, ExtendSelect2= 5
151 0:0:10.760025031 DEBUG (12): New parent for NodeID= 12 : curdepth= 3 , parentID= 9
152 0:0:10.760025031 DEBUG (12): ExtendSelect1= 6, ExtendSelect2= 5
153 0:0:10.760025031 DEBUG (14): New parent for NodeID= 14 : curdepth= 3 , parentID= 9
154 0:0:10.760025031 DEBUG (14): ExtendSelect1= 6, ExtendSelect2= 5
155 0:0:10.760025031 DEBUG (13): New parent for NodeID= 13 : curdepth= 3 , parentID= 9
156 0:0:10.760025031 DEBUG (13): ExtendSelect1= 6, ExtendSelect2= 5
157 0:0:10.765853854 DEBUG (15): New parent for NodeID= 15 : curdepth= 3 , parentID= 10
158 0:0:10.765853854 DEBUG (15): ExtendSelect1= 6, ExtendSelect2= 5
```

Figure 5: Extended TAG Routing

Με βάση την παραπάνω εικόνα, προκύπτει ότι το δέντρο δρομολόγησης των μετρήσεων από τα φύλλα προς τη ρίζα είναι το εξής:

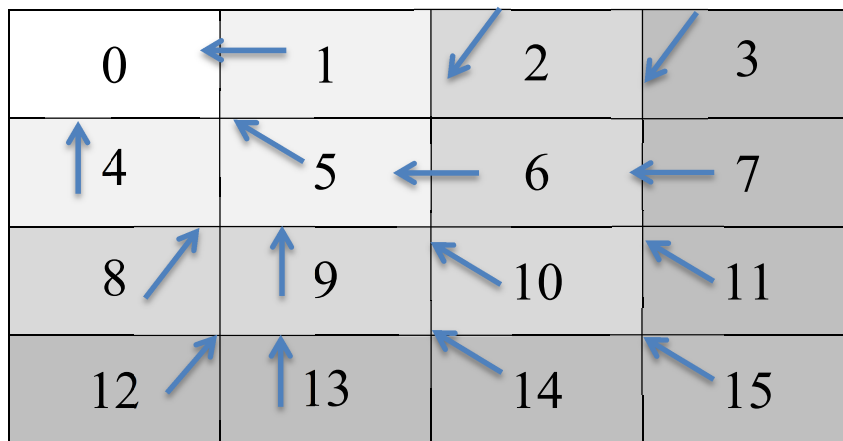


Table 2: Extended TAG - Δένδρο Δρομολόγησης

Έπειτα, στο ακόλουθο screenshot, παρουσιάζεται η διαδικασία με την οποία οι μετρήσεις λαμβάνονται και συναθροίζονται σε κάθε κόμβο, από τα φύλλα προς τη ρίζα. Αρχικά στέλνουν οι κόμβοι του τελευταίου επιπέδου με βάθος 3. Πρώτα στέλνει ο κόμβος 3, ο οποίος μετρά 46. Με βάση τη μέτρηση αυτή υπολογίζει το SUM και το SUMSQ, ενώ προωθεί και το COUNT, το οποίο εφόσον είναι φύλλο, θα είναι ίσο με 1. Ομοίως στέλνουν και οι υπόλοιποι κόμβοι του επιπέδου αυτού, δηλαδή κατά σειρά οι 7, 11, 12, 13, 14, 15. Οι κόμβοι του επόμενου επιπέδου, με βάθος 2, λαμβάνουν αρχικά τις τιμές των κόμβων βάθους 3.


```

175 0:0:59.008789072 DEBUG (3): NodeID = 3 curdepth= 3
176 0:0:59.008789072 DEBUG (3): Starting Data transmission to parent!
177 0:0:59.008789072 DEBUG (3): measurement is: 46
178 0:0:59.008789082 DEBUG (3): Node's SUMSQ is: 2116
179 0:0:59.008789082 DEBUG (3): Node's SUM is: 46
180 0:0:59.008789082 DEBUG (3): Node's COUNT is: 1
181 0:0:59.015792826 DEBUG (6): Measurement received from: 3
182 0:0:59.015792836 DEBUG (6): Received from childID: 3 - SUMSQ: 2116, SUM: 46, COUNT: 1
183 0:0:59.020507823 DEBUG (7): NodeID = 7 curdepth= 3
184 0:0:59.020507823 DEBUG (7): Starting Data transmission to parent!
185 0:0:59.020507823 DEBUG (7): measurement is: 42
186 0:0:59.020507833 DEBUG (7): Node's SUMSQ is: 1764
187 0:0:59.020507833 DEBUG (7): Node's SUM is: 42
188 0:0:59.020507833 DEBUG (7): Node's COUNT is: 1
189 0:0:59.030136072 DEBUG (6): Measurement received from: 7
190 0:0:59.030136082 DEBUG (6): Received from childID: 7 - SUMSQ: 1764, SUM: 42, COUNT: 1
191 0:0:59.032226573 DEBUG (11): NodeID = 11 curdepth= 3
192 0:0:59.032226573 DEBUG (11): Starting Data transmission to parent!
193 0:0:59.032226573 DEBUG (11): measurement is: 21
194 0:0:59.032226583 DEBUG (11): Node's SUMSQ is: 441
195 0:0:59.032226583 DEBUG (11): Node's SUM is: 21
196 0:0:59.032226583 DEBUG (11): Node's COUNT is: 1
197 0:0:59.035156261 DEBUG (12): NodeID = 12 curdepth= 3
198 0:0:59.035156261 DEBUG (12): Starting Data transmission to parent!
199 0:0:59.035156261 DEBUG (12): measurement is: 15
200 0:0:59.035156271 DEBUG (12): Node's SUMSQ is: 225
201 0:0:59.035156271 DEBUG (12): Node's SUM is: 15
202 0:0:59.035156271 DEBUG (12): Node's COUNT is: 1
203 0:0:59.038085948 DEBUG (13): NodeID = 13 curdepth= 3
204 0:0:59.038085948 DEBUG (13): Starting Data transmission to parent!
205 0:0:59.038085948 DEBUG (13): measurement is: 29
206 0:0:59.038085958 DEBUG (13): Node's SUMSQ is: 841
207 0:0:59.038085958 DEBUG (13): Node's SUM is: 29
208 0:0:59.038085958 DEBUG (13): Node's COUNT is: 1
209 0:0:59.041015636 DEBUG (14): NodeID = 14 curdepth= 3
210 0:0:59.041015636 DEBUG (14): Starting Data transmission to parent!
211 0:0:59.041015636 DEBUG (14): measurement is: 35
212 0:0:59.041015646 DEBUG (14): Node's SUMSQ is: 1225
213 0:0:59.041015646 DEBUG (14): Node's SUM is: 35
214 0:0:59.041015646 DEBUG (14): Node's COUNT is: 1
215 0:0:59.041183457 DEBUG (9): Measurement received from: 12
216 0:0:59.041183467 DEBUG (9): Received from childID: 12 - SUMSQ: 225, SUM: 15, COUNT: 1
217 0:0:59.041717495 DEBUG (6): Measurement received from: 11
218 0:0:59.041717505 DEBUG (6): Received from childID: 11 - SUMSQ: 441, SUM: 21, COUNT: 1
219 0:0:59.043869006 DEBUG (9): Measurement received from: 13
220 0:0:59.043869016 DEBUG (9): Received from childID: 13 - SUMSQ: 841, SUM: 29, COUNT: 1
221 0:0:59.043945324 DEBUG (15): NodeID = 15 curdepth= 3
222 0:0:59.043945324 DEBUG (15): Starting Data transmission to parent!
223 0:0:59.043945324 DEBUG (15): measurement is: 10
224 0:0:59.043945334 DEBUG (15): Node's SUMSQ is: 100
225 0:0:59.043945334 DEBUG (15): Node's SUM is: 10
226 0:0:59.043945334 DEBUG (15): Node's COUNT is: 1
227 0:0:59.048461909 DEBUG (10): Measurement received from: 15
228 0:0:59.048461919 DEBUG (10): Received from childID: 15 - SUMSQ: 100, SUM: 10, COUNT: 1
229 0:0:59.050064055 DEBUG (9): Measurement received from: 14
230 0:0:59.050064065 DEBUG (9): Received from childID: 14 - SUMSQ: 1225, SUM: 35, COUNT: 1

```

Figure 6: Extended TAG Round 1 (1)

Στη συνέχεια, οι κόμβοι βάθους 2, συναθροίζουν τις μετρήσεις τους με αυτές των παιδιών τους, εφόσον έχουν, και προωθούν τις τελικές τιμές τους στους κόμβους βάθους 1.

```

231 0:0:59.255859385 DEBUG (2): NodeID = 2 curdepth= 2
232 0:0:59.255859385 DEBUG (2): Starting Data transmission to parent!
233 0:0:59.255859385 DEBUG (2): measurement is: 14
234 0:0:59.255859395 DEBUG (2): Node's SUMSQ is: 196
235 0:0:59.255859395 DEBUG (2): Node's SUM is: 14
236 0:0:59.255859395 DEBUG (2): Node's COUNT is: 1
237 0:0:59.259140015 DEBUG (5): Measurement received from: 2
238 0:0:59.259140025 DEBUG (5): Received from childID: 2 - SUMSQ: 196, SUM: 14, COUNT: 1
239 0:0:59.267578135 DEBUG (6): NodeID = 6 curdepth= 2
240 0:0:59.267578135 DEBUG (6): Starting Data transmission to parent!
241 0:0:59.267578135 DEBUG (6): measurement is: 20
242 0:0:59.267578145 DEBUG (6): ChildID: 3 has sumsq: 2116
243 0:0:59.267578145 DEBUG (6): ChildID: 7 has sumsq: 1764
244 0:0:59.267578145 DEBUG (6): ChildID: 11 has sumsq: 441
245 0:0:59.267578145 DEBUG (6): Node's SUMSQ is: 4721
246 0:0:59.267578145 DEBUG (6): ChildID: 3 has sum: 46
247 0:0:59.267578145 DEBUG (6): ChildID: 7 has sum: 42
248 0:0:59.267578145 DEBUG (6): ChildID: 11 has sum: 21
249 0:0:59.267578145 DEBUG (6): Node's SUM is: 129
250 0:0:59.267578145 DEBUG (6): ChildID: 3 has count: 1
251 0:0:59.267578145 DEBUG (6): ChildID: 7 has count: 1
252 0:0:59.267578145 DEBUG (6): ChildID: 11 has count: 1
253 0:0:59.267578145 DEBUG (6): Node's COUNT is: 4
254 0:0:59.269378672 DEBUG (5): Measurement received from: 6
255 0:0:59.269378682 DEBUG (5): Received from childID: 6 - SUMSQ: 4721, SUM: 129, COUNT: 4
256 0:0:59.273437510 DEBUG (8): NodeID = 8 curdepth= 2
257 0:0:59.273437510 DEBUG (8): Starting Data transmission to parent!
258 0:0:59.273437510 DEBUG (8): measurement is: 2
259 0:0:59.273437520 DEBUG (8): Node's SUMSQ is: 4
260 0:0:59.273437520 DEBUG (8): Node's SUM is: 2
261 0:0:59.273437520 DEBUG (8): Node's COUNT is: 1
262 0:0:59.276367198 DEBUG (9): NodeID = 9 curdepth= 2
263 0:0:59.276367198 DEBUG (9): Starting Data transmission to parent!
264 0:0:59.276367198 DEBUG (9): measurement is: 1
265 0:0:59.276367208 DEBUG (9): ChildID: 12 has sumsq: 225
266 0:0:59.276367208 DEBUG (9): ChildID: 13 has sumsq: 841
267 0:0:59.276367208 DEBUG (9): ChildID: 14 has sumsq: 1225
268 0:0:59.276367208 DEBUG (9): Node's SUMSQ is: 2292
269 0:0:59.276367208 DEBUG (9): ChildID: 12 has sum: 15
270 0:0:59.276367208 DEBUG (9): ChildID: 13 has sum: 29
271 0:0:59.276367208 DEBUG (9): ChildID: 14 has sum: 35
272 0:0:59.276367208 DEBUG (9): Node's SUM is: 80
273 0:0:59.276367208 DEBUG (9): ChildID: 12 has count: 1
274 0:0:59.276367208 DEBUG (9): ChildID: 13 has count: 1
275 0:0:59.276367208 DEBUG (9): ChildID: 14 has count: 1
276 0:0:59.276367208 DEBUG (9): Node's COUNT is: 4
277 0:0:59.279235826 DEBUG (5): Measurement received from: 8
278 0:0:59.279235836 DEBUG (5): Received from childID: 8 - SUMSQ: 4, SUM: 2, COUNT: 1
279 0:0:59.279296886 DEBUG (10): NodeID = 10 curdepth= 2
280 0:0:59.279296886 DEBUG (10): Starting Data transmission to parent!
281 0:0:59.279296886 DEBUG (10): measurement is: 29
282 0:0:59.279296896 DEBUG (10): ChildID: 15 has sumsq: 100
283 0:0:59.279296896 DEBUG (10): Node's SUMSQ is: 941
284 0:0:59.279296896 DEBUG (10): ChildID: 15 has sum: 10
285 0:0:59.279296896 DEBUG (10): Node's SUM is: 39
286 0:0:59.279296896 DEBUG (10): ChildID: 15 has count: 1
287 0:0:59.279296896 DEBUG (10): Node's COUNT is: 2

```

Figure 7: Extended TAG Round 1 (2)

Ακολουθώς, οι κόμβοι του πρώτου επιπέδου, δηλαδή οι 1, 4 και 5, λαμβάνουν τις μετρήσεις των κόμβων του δεύτερου επιπέδου, και συναθροίζοντας στέλνουν τις τελικές τιμές τους στη ρίζα. Η ρίζα εν τέλει λαμβάνει από τον κόμβο 1 τις τιμές $SUMSQ_1 = 361$, $SUM_1 = 19$, $COUNT_1 = 1$, από τον κόμβο 4 λαμβάνει τις τιμές $SUMSQ_4 = 25$, $SUM_4 = 5$, $COUNT_4 = 1$, και από τον κόμβο 5, ο οποίος τυχαίνει να έχει στο υποδέντρο του όλους τους υπόλοιπους κόμβους, λαμβάνει τις τιμές $SUMSQ_5 = 9835$, $SUM_5 = 305$, $COUNT_5 = 13$.


```

288 0:0:59.281723012 DEBUG (5): Measurement received from: 9
289 0:0:59.281723022 DEBUG (5): Received from childID: 9 - SUMSQ: 2292, SUM: 80, COUNT: 4
290 0:0:59.283905023 DEBUG (5): Measurement received from: 10
291 0:0:59.283905033 DEBUG (5): Received from childID: 10 - SUMSQ: 941, SUM: 39, COUNT: 2
292 0:0:59.502929697 DEBUG (1): NodeID = 1 curdepth= 1
293 0:0:59.502929697 DEBUG (1): Starting Data transmission to parent!
294 0:0:59.502929697 DEBUG (1): measurement is: 19
295 0:0:59.502929707 DEBUG (1): Node's SUMSQ is: 361
296 0:0:59.502929707 DEBUG (1): Node's SUM is: 19
297 0:0:59.502929707 DEBUG (1): Node's COUNT is: 1
298 0:0:59.509567242 DEBUG (0): Measurement received from: 1
299 0:0:59.509567252 DEBUG (0): Received from childID: 1 - SUMSQ: 361, SUM: 19, COUNT: 1
300 0:0:59.511718760 DEBUG (4): NodeID = 4 curdepth= 1
301 0:0:59.511718760 DEBUG (4): Starting Data transmission to parent!
302 0:0:59.511718760 DEBUG (4): measurement is: 5
303 0:0:59.511718770 DEBUG (4): Node's SUMSQ is: 25
304 0:0:59.511718770 DEBUG (4): Node's SUM is: 5
305 0:0:59.511718770 DEBUG (4): Node's COUNT is: 1
306 0:0:59.514648448 DEBUG (5): NodeID = 5 curdepth= 1
307 0:0:59.514648448 DEBUG (5): Starting Data transmission to parent!
308 0:0:59.514648448 DEBUG (5): measurement is: 41
309 0:0:59.514648458 DEBUG (5): ChildID: 2 has sumsq: 196
310 0:0:59.514648458 DEBUG (5): ChildID: 6 has sumsq: 4721
311 0:0:59.514648458 DEBUG (5): ChildID: 8 has sumsq: 4
312 0:0:59.514648458 DEBUG (5): ChildID: 9 has sumsq: 2292
313 0:0:59.514648458 DEBUG (5): ChildID: 10 has sumsq: 941
314 0:0:59.514648458 DEBUG (5): Node's SUMSQ is: 9835
315 0:0:59.514648458 DEBUG (5): ChildID: 2 has sum: 14
316 0:0:59.514648458 DEBUG (5): ChildID: 6 has sum: 129
317 0:0:59.514648458 DEBUG (5): ChildID: 8 has sum: 2
318 0:0:59.514648458 DEBUG (5): ChildID: 9 has sum: 80
319 0:0:59.514648458 DEBUG (5): ChildID: 10 has sum: 39
320 0:0:59.514648458 DEBUG (5): Node's SUM is: 305
321 0:0:59.514648458 DEBUG (5): ChildID: 2 has count: 1
322 0:0:59.514648458 DEBUG (5): ChildID: 6 has count: 4
323 0:0:59.514648458 DEBUG (5): ChildID: 8 has count: 1
324 0:0:59.514648458 DEBUG (5): ChildID: 9 has count: 4
325 0:0:59.514648458 DEBUG (5): ChildID: 10 has count: 2
326 0:0:59.514648458 DEBUG (5): Node's COUNT is: 13
327 0:0:59.518508891 DEBUG (0): Measurement received from: 4
328 0:0:59.518508901 DEBUG (0): Received from childID: 4 - SUMSQ: 25, SUM: 5, COUNT: 1
329 0:0:59.520233142 DEBUG (0): Measurement received from: 5
330 0:0:59.520233152 DEBUG (0): Received from childID: 5 - SUMSQ: 9835, SUM: 305, COUNT: 13
331 0:0:59.750000010 DEBUG (0): NodeID = 0 curdepth= 0
332 0:0:59.750000010 DEBUG (0): measurement is: 3
333 0:0:59.750000020 DEBUG (0): ChildID: 1 has sumsq: 361
334 0:0:59.750000020 DEBUG (0): ChildID: 4 has sumsq: 25
335 0:0:59.750000020 DEBUG (0): ChildID: 5 has sumsq: 9835
336 0:0:59.750000020 DEBUG (0): Node's SUMSQ is: 10230
337 0:0:59.750000020 DEBUG (0): ChildID: 1 has sum: 19
338 0:0:59.750000020 DEBUG (0): ChildID: 4 has sum: 5
339 0:0:59.750000020 DEBUG (0): ChildID: 5 has sum: 305
340 0:0:59.750000020 DEBUG (0): Node's SUM is: 332
341 0:0:59.750000020 DEBUG (0): ChildID: 1 has count: 1
342 0:0:59.750000020 DEBUG (0): ChildID: 4 has count: 1
343 0:0:59.750000020 DEBUG (0): ChildID: 5 has count: 13
344 0:0:59.750000020 DEBUG (0): Node's COUNT is: 16
345 0:0:59.750000020 DEBUG (0):
346 0:0:59.750000020 DEBUG (0): FINAL RESULTS: VAR: 208.81, AVG: 20.75

```

Figure 8: Extended TAG Round 1 (3)

Η ρίζα εκτυπώνει τελικά τα αποτελέσματα:

$$\text{VAR} = [(\text{SUMSQ}_1 + \text{SUMSQ}_4 + \text{SUMSQ}_5 + \text{MEAS}_0^2) / \text{COUNT}] - [(\text{SUM}_1 + \text{SUM}_4 + \text{SUM}_5 + \text{MEAS}_0) / \text{COUNT}]^2 = [(361 + 25 + 9835 + 9) / 16] - [(19 + 5 + 305 + 3) / 16]^2 = 639.375 - 430.5625 = 208.81$$

$$\text{AVG} = (\text{SUM}_1 + \text{SUM}_4 + \text{SUM}_5 + \text{MEAS}_0) / \text{COUNT} = (19 + 5 + 305 + 3) / 16 = 20.75$$

Βιβλιογραφικές Αναφορές

[1] Random number generation, with seed from Ubuntu file:

<http://mail.millennium.berkeley.edu/pipermail/tinyos-help/2014-July/058471.html>