

**Πολυτεχνείο Κρήτης  
Σχολή Ηλεκτρολόγων Μηχανικών  
και Μηχανικών Υπολογιστών**



**Επεξεργασία και Διαχείριση Δεδομένων σε  
Δίκτυα Αισθητήρων  
ΠΛΗ 511  
Εργασία στο TinyOS  
Αναφορά 2ου Προγράμματος**

Ιωαννίδης Χρήστος (2018030006)  
Παπαματθαϊάκη Ηλέκτρα-Δέσποινα (2018030106)

## 1. Εισαγωγή

Στα πλαίσια του δεύτερου μέρους της εργασίας στο TinyOS κληθήκαμε να προσθέσουμε κάποιες ακόμη λειτουργίες στο δίκτυο αισθητήρων που προσομοιώνουμε. Αναλυτικότερα, η λειτουργία που ζητείται προβλέπει τη σύνδεση των κόμβων που έχουν χάσει την επαφή με τον πατέρα τους. Η λειτουργικότητα σε High Level είναι να σταλεί ένα help μήνυμα από τον κάθε κόμβο που αντιμετωπίζει το πρόβλημα, ώστε οι γειτονικοί κόμβοι να στείλουν ξανά το routing μήνυμα. Στη συνέχεια, προσαρμόζεται το δέντρο που είχε δημιουργηθεί στο routing αναλόγως. Η υλοποίηση θα παρουσιαστεί λεπτομερώς παρακάτω.

## 2. Υλοποίηση Help Message

Με την χρήση ενός νέου timer, του **LoseParentTimer**, όταν περάσουν 500 δευτερόλεπτα από την έναρξη της προσομοίωσης, ορισμένοι κόμβοι αποσυνδέονται τεχνητά από τους γονείς τους με μια προκαθορισμένη πιθανότητα.

```
//first function to be called after boot is complete
event void RadioControl.startDone(error_t err)
{
.. CODE ..
    }else{
        call LoseParentTimer.startOneShot(500*1024);
    }
}

event void LoseParentTimer.fired() {
. . .
    loseConnectionProb=(uint8_t) (callRandomMeasurement.rand16()%100 );
    //LOSE CONNECTION WITH PROBABILITY 30%
    if(loseConnectionProb<30){
        dbg("help", "Lost parent\n");
        curdepth=-1; //LOST CONNECTION
        parentID=-1;

        atomic{
            mrpkt->help=(uint8_t)TOS_NODE_ID;
        }
. . .
        post sendHelpTask();
. . .
    }
}
```

Όταν ένας κόμβος χάνει τη σύνδεσή του με τον γονιό του, κάνει broadcast ένα μήνυμα για βοήθεια βοήθεια. Ο σκοπός είναι να προσπαθήσει να επανασυνδεθεί στο δέντρο. Η δομή του help message υλοποιήθηκε ως εξής:

```
typedef nx_struct HelpMsg
{
    nx_uint8_t help; //SENDER ID
} HelpMsg;
```

### 3. Υλοποίηση επαναποστολής Routing μηνύματος

Όταν ένας κόμβος λάβει ένα μήνυμα για βοήθεια, κάνει broadcast ένα μήνυμα που θα λάβουν όλοι οι γειτονικοί κόμβοι (μαζί με αυτον που ζήτησε βοήθεια) το οποίο περιλαμβάνει τα στοιχεία του προκειμένου να καταχωρηθεί σαν υποψήφιος γονέας. Ο κόμβος που ζήτησε βοήθεια με τη σειρά του περιμένει ένα χρονικό διάστημα μέχρι να του στείλουν και άλλοι υποψήφιοι και μετά αποφασίζει για τον νέο του γονιό.

```
typedef nx_struct NewRoutingMsg
{
    nx_uint16_t senderID;
    nx_uint8_t depth;
} NewRoutingMsg;
```

```
// processing of Help packet
task void receiveHelpTask()
{
    if(len == sizeof(HelpMsg))
    {
        HelpMsg * mpkt = (HelpMsg*) (call
HelpPacket.getPayload(&radioHelpRecPkt,len));

        //AVOID BEING YOUR PARENT'S PARENT CANDIDATE
        if( (mpkt->help) !=parentID) {
call RetransmissionTimer.startOneShot(1+(call randomMeasurement.rand16()%
20));
        }
    }
}
```

Τυχαιότητα έχει προστεθεί στην μετάδοση αυτών των μηνυμάτων προκειμένου να αποφευχθούν collisions.

Ο κόμβος που αρχικά ζήτησε βοήθεια, με το που λάβει την πρώτη απάντηση θα ξεκινήσει έναν timer που θα χτυπήσει για να τον ειδοποιήσει να σταματήσει να περιμένει, στο μεταξύ

λαμβάνει απαντήσεις και κρατάει στη μνήμη τα στοιχεία του κόμβου που ήταν πιο κοντά στο επίπεδο του παλαιού γονιού του, όταν χτυπήσει ο timer, ο κόμβος που είναι στην μνήμη εκείνη τη στιγμή γίνεται ο καινούριος γονιός του.

```
task void receiveRetransmissionTask()
{
.. CODE ..

    // processing of routing packet
    if(len == sizeof(NewRoutingMsg))
    {
.. CODE ..

        if(retransmissions_counter==0){
            call StopWaitingForNewCandidatesTimer.startOneShot(5*1024);
            retransmissions_counter=5;
        }
        if(mpkt->depth<candidate_curdepth){
            if(mpkt->depth>=old_curdepth-1){
                candidate_curdepth=mpkt->depth;
                candidate_parentID=mpkt->senderID;
            }
        }
    }
```

```
event void StopWaitingForNewCandidatesTimer.fired(){
parentID=candidate_parentID; //FINALIZE NEW PARENT
curdepth=candidate_curdepth+1;//SET NEW DEPTH

if(curdepth>old_curdepth){
//NOTIFY CHILDREN TO CHANGE DEPTH AS WELL
    call NotifyChildrenTimer.startOneShot(1+(call RandomMeasurement.rand16() % 20));
}
}
```

Τέλος, με την επιλογή νέου γονιού, σε περίπτωση που το επίπεδο του κόμβου άλλαξε, ενημερώνει και τα παιδιά του να αλλάξουν επίπεδο (λεπτομέρειες παρακάτω).

## 4. Υλοποίηση αλλαγής χρονισμού

Όταν ένας κόμβος αλλάζει επίπεδο θα πρέπει να αλλάζει ανάλογα και ο χρονισμός με τον οποίο στέλνει δεδομένα. Η λειτουργία αυτή, υλοποιείται με την προσθήκη ενός καινούριου timer (**OffsetTimer**) ο οποίος θα λειτουργήσει σαν offset για το πόσο πιο μετά θα μεταφερθεί ο χρόνος που θα πρέπει να στείλει δεδομένα ο κόμβος. Η μεταφορά θα είναι ανάλογη με το πόσα επίπεδα κατέβηκε ο ίδιος ο κόμβος.

```
event void SendMeasurementTimer.fired() //TIME TO SEND DATA
{
  ..
  .. CODE..
  if(old_curdepth>curdepth){ //IF DEPTH CHANGED, SEND LATER
    //DELAY THE SEND TIMING TO ACCOMODATE FOR THE DEPTH CHANGE
    call OffsetTimer.startOneShot((curdepth-old_curdepth)*TIMER_FAST_SEND_MEASUREMENT_PERIOD);
  }else{//DEPTH IS THE SAME, SEND DATA AS USUAL
    //create a measurement message to send
    post RandNumMaker();
  }

}

event void OffsetTimer.fired(){
  post RandNumMaker(); //NOW SEND THE DATA
}
```

## 5. Ειδοποίηση παιδιών για αλλαγή επιπέδου γονέα

Για να ειδοποιηθούν τα παιδιά για την αλλαγή επιπέδου του γονέα τους, προκειμένου να αλλάξουν και αυτά επίπεδο, υλοποιήθηκε ένα καινούριο είδος μηνύματος, το **NotifyChildrenMsg**. Αυτό γίνεται broadcast από τον γονιό (περιέχει το ID του και το νέο depth του) και διαβάζεται απο τα παιδια του. Αφού το λάβουν και αλλάξουν το depth τους, παίρνουν την θέση του γονιού και αναλαμβάνουν να μεταβιβάσουν το μήνυμα παρακάτω στην ιεραρχία, μέχρις ότου, όλοι οι κόμβοι που συνδέονται με τον αρχικό που άλλαξε επίπεδο, να έχουν αλλάξει και αυτοι το επίπεδο τους αναλόγως.

Το μήνυμα που υλοποιήθηκε:

```
typedef nx_struct NotifyChildrenMsg
{
  nx_uint16_t senderID;
  nx_uint16_t new_depth;
} NotifyChildrenMsg;
```

```
//GOT NOTIFICATION THAT SOMEONE CHANGED DEPTH
task void receiveNotifyChildrenTask()    {
..
.. //CODE
..

    // processing of NotifyChildren packet
    if(len == sizeof(NotifyChildrenMsg))
    {
        NotifyChildrenMsg * mpkt = (NotifyChildrenMsg*) (call
NotifyChildrenPacket.getPayload(&radioNotifyChildrenRecPkt,len));

        if(mpkt->senderID==parentID){//IF THE SENDER IS MY PARENT
            curdepth=(uint8_t) (mpkt->new_depth+1); //MY NEW DEPTH= MY PARENT'S DEPTH +1

            //NOTIFY MY CHILDREN OF MY NEW DEPTH
            call NotifyChildrenTimer.startOneShot(1+(call RandomMeasurement.rand16() % 20));
        }
    }
}
```

Έχει προστεθεί τυχαιότητα στην μετάδοση αυτών των μηνυμάτων προκειμένου να αποφευχθούν collisions.

## 6. Testing

Προκειμένου να δούμε ότι λειτουργεί σωστά η **ειδοποίηση παιδιών για αλλαγή επιπέδου γονέα** αυξήσαμε τεχνητά το depth όλων των κόμβων που έστειλαν αρχικά help message, με στοχο να τους αναγκάσουμε να στείλουν μήνυμα αλλαγής βάθους στα παιδιά τους.

```
event void StopWaitingForNewCandidatesTimer.fired() {
    parentID=candidate_parentID;
    curdepth=candidate_curdepth+1;      //change +1 to +2 to test
```

```
event void StopWaitingForNewCandidatesTimer.fired() {
    parentID=candidate_parentID;
    curdepth=candidate_curdepth+2;      //change +1 to +2 to test
```

Για να μπορέσουμε να ελέγξουμε τον κώδικά μας, χρησιμοποιήσαμε πιθανότητα 30% αντί 2% καθώς ήταν πολύ πιο εύκολο να ελέγξουμε τα αποτελέσματα γιατί η συχνότητα που χανόταν η επαφή ήταν μεγαλύτερη.

- Σωστή συμπεριφορά του κώδικα για **25 κόμβους**:

```
0:8:45.000000010 DEBUG (3): Lost parent
0:8:45.000000010 DEBUG (5): Lost parent
0:8:45.000000010 DEBUG (7): Lost parent
0:8:45.000000010 DEBUG (9): Lost parent
0:8:45.000000011 DEBUG (19): Lost parent
0:8:45.000000012 DEBUG (21): Lost parent
0:8:45.000000012 DEBUG (24): Lost parent
0:8:45.006088253 DEBUG (23): reveived help message from node:19
0:8:45.006088253 DEBUG (23): broadcasted rerouting message succesfully
0:8:45.006088253 DEBUG (13): reveived help message from node:19
0:8:45.006088253 DEBUG (13): broadcasted rerouting message succesfully
0:8:45.006088253 DEBUG (8): reveived help message from node:19
0:8:45.006088253 DEBUG (8): broadcasted rerouting message succesfully
0:8:45.006088253 DEBUG (22): reveived help message from node:19
0:8:45.006088253 DEBUG (22): broadcasted rerouting message succesfully
0:8:45.006088253 DEBUG (12): reveived help message from node:19
0:8:45.006088253 DEBUG (12): broadcasted rerouting message succesfully
0:8:45.007858262 DEBUG (18): reveived help message from node:21
0:8:45.007858262 DEBUG (18): broadcasted rerouting message succesfully
0:8:45.007858262 DEBUG (13): reveived help message from node:21
0:8:45.007858262 DEBUG (13): broadcasted rerouting message succesfully
0:8:45.007858262 DEBUG (17): reveived help message from node:21
0:8:45.007858262 DEBUG (17): broadcasted rerouting message succesfully
0:8:45.007858262 DEBUG (12): reveived help message from node:21
0:8:45.007858262 DEBUG (12): broadcasted rerouting message succesfully
0:8:45.007858262 DEBUG (15): reveived help message from node:21
0:8:45.007858262 DEBUG (15): broadcasted rerouting message succesfully
0:8:45.007858262 DEBUG (10): reveived help message from node:21
0:8:45.007858262 DEBUG (10): broadcasted rerouting message succesfully
0:8:45.009216287 DEBUG (18): reveived help message from node:24
0:8:45.009216287 DEBUG (18): broadcasted rerouting message succesfully
0:8:45.009216287 DEBUG (13): reveived help message from node:24
0:8:45.009216287 DEBUG (13): broadcasted rerouting message succesfully
0:8:45.009216287 DEBUG (17): reveived help message from node:24
```

```
0:8:45.009216287 DEBUG (17): broadcasted rerouting message succesfully
0:8:45.009262061 DEBUG (11): received help message from node:3
0:8:45.009262061 DEBUG (11): broadcasted rerouting message succesfully
0:8:45.009262061 DEBUG (6): received help message from node:3
0:8:45.009262061 DEBUG (6): broadcasted rerouting message succesfully
0:8:45.009262061 DEBUG (14): received help message from node:3
0:8:45.009262061 DEBUG (14): broadcasted rerouting message succesfully
0:8:45.010955777 DEBUG (10): received help message from node:7
0:8:45.010955777 DEBUG (15): received help message from node:7
0:8:45.010955777 DEBUG (15): broadcasted rerouting message succesfully
0:8:45.010955777 DEBUG (0): received help message from node:7
0:8:45.010955777 DEBUG (0): broadcasted rerouting message succesfully
0:8:45.010955777 DEBUG (14): received help message from node:7
0:8:45.010955777 DEBUG (4): received help message from node:7
0:8:45.010955777 DEBUG (4): broadcasted rerouting message succesfully
0:8:45.013778694 DEBUG (7): received rerouting message from parent candidate
0:8:45.013778694 DEBUG (7): candidate parent ID= 15 , depth= 2
0:8:45.013778694 DEBUG (7): initiate wait timer
0:8:45.013778694 DEBUG (7): candidate parent accepted ID= 15 , depth= 2
0:8:45.013778694 DEBUG (21): received rerouting message from parent candidate
0:8:45.013778694 DEBUG (21): candidate parent ID= 15 , depth= 2
0:8:45.013778694 DEBUG (21): initiate wait timer
0:8:45.013778694 DEBUG (21): candidate parent accepted ID= 15 , depth= 2
0:8:45.013793963 DEBUG (3): received rerouting message from parent candidate
0:8:45.013793963 DEBUG (3): candidate parent ID= 6 , depth= 1
0:8:45.013793963 DEBUG (3): initiate wait timer
0:8:45.013793963 DEBUG (3): candidate parent accepted ID= 6 , depth= 1
0:8:45.015304581 DEBUG (9): received rerouting message from parent candidate
0:8:45.015304581 DEBUG (9): candidate parent ID= 12 , depth= 1
0:8:45.015304581 DEBUG (9): initiate wait timer
0:8:45.015304581 DEBUG (9): candidate parent accepted ID= 12 , depth= 1
0:8:45.015304581 DEBUG (5): received rerouting message from parent candidate
0:8:45.015304581 DEBUG (5): candidate parent ID= 12 , depth= 1
0:8:45.015304581 DEBUG (5): initiate wait timer
0:8:45.015304581 DEBUG (5): candidate parent accepted ID= 12 , depth= 1
0:8:45.015304581 DEBUG (21): received rerouting message from parent candidate
0:8:45.015304581 DEBUG (21): candidate parent ID= 12 , depth= 1
0:8:45.015304581 DEBUG (21): candidate parent accepted ID= 12 , depth= 1
0:8:45.015304581 DEBUG (3): received rerouting message from parent candidate
0:8:45.015304581 DEBUG (3): candidate parent ID= 12 , depth= 1
0:8:45.015304581 DEBUG (19): received rerouting message from parent candidate
0:8:45.015304581 DEBUG (19): candidate parent ID= 12 , depth= 1
0:8:45.015304581 DEBUG (19): initiate wait timer
0:8:45.015304581 DEBUG (19): candidate parent accepted ID= 12 , depth= 1
0:8:45.015304581 DEBUG (24): received rerouting message from parent candidate
0:8:45.015304581 DEBUG (24): candidate parent ID= 12 , depth= 1
0:8:45.015304581 DEBUG (24): initiate wait timer
0:8:45.015304581 DEBUG (24): candidate parent accepted ID= 12 , depth= 1
0:8:45.016815192 DEBUG (3): received rerouting message from parent candidate
0:8:45.016815192 DEBUG (3): candidate parent ID= 11 , depth= 1
0:8:45.016815192 DEBUG (5): received rerouting message from parent candidate
0:8:45.016815192 DEBUG (5): candidate parent ID= 11 , depth= 1
0:8:45.016815192 DEBUG (7): received rerouting message from parent candidate
0:8:45.016815192 DEBUG (7): candidate parent ID= 11 , depth= 1
0:8:45.016815192 DEBUG (7): candidate parent accepted ID= 11 , depth= 1
0:8:45.018508920 DEBUG (24): received rerouting message from parent candidate
0:8:45.018508920 DEBUG (24): candidate parent ID= 17 , depth= 2
0:8:45.018508920 DEBUG (9): received rerouting message from parent candidate
0:8:45.018508920 DEBUG (9): candidate parent ID= 17 , depth= 2
0:8:45.018508920 DEBUG (21): received rerouting message from parent candidate
```



```
0:8:45.018508920 DEBUG (21): candidate parent ID= 17 , depth= 2
0:8:45.018508920 DEBUG (5): received rerouting message from parent candidate
0:8:45.018508920 DEBUG (5): candidate parent ID= 17 , depth= 2
0:8:45.021423332 DEBUG (19): received rerouting message from parent candidate
0:8:45.021423332 DEBUG (19): candidate parent ID= 22 , depth= 2
0:8:45.024276720 DEBUG (19): received rerouting message from parent candidate
0:8:45.024276720 DEBUG (19): candidate parent ID= 23 , depth= 2
0:8:45.026885986 DEBUG (9): received rerouting message from parent candidate
0:8:45.026885986 DEBUG (9): candidate parent ID= 13 , depth= 2
0:8:45.026885986 DEBUG (19): received rerouting message from parent candidate
0:8:45.026885986 DEBUG (19): candidate parent ID= 13 , depth= 2
0:8:45.026885986 DEBUG (24): received rerouting message from parent candidate
0:8:45.026885986 DEBUG (24): candidate parent ID= 13 , depth= 2
0:8:45.028121928 DEBUG (19): received rerouting message from parent candidate
0:8:45.028121928 DEBUG (19): candidate parent ID= 8 , depth= 2
0:8:45.030120836 DEBUG (9): received rerouting message from parent candidate
0:8:45.030120836 DEBUG (9): candidate parent ID= 18 , depth= 2
0:8:45.030120836 DEBUG (24): received rerouting message from parent candidate
0:8:45.030120836 DEBUG (24): candidate parent ID= 18 , depth= 2
0:8:45.031845092 DEBUG (7): received rerouting message from parent candidate
0:8:45.031845092 DEBUG (7): candidate parent ID= 14 , depth= 2
0:8:45.031845092 DEBUG (3): received rerouting message from parent candidate
0:8:45.031845092 DEBUG (3): candidate parent ID= 14 , depth= 2
0:8:50.013671885 DEBUG (3): candidate parent finalized ID= 6 , depth= 2,old depth=2
0:8:50.013671885 DEBUG (3): candidate parent finalized ID= 6 , depth= 2,old depth=2
0:8:50.013671885 DEBUG (7): candidate parent finalized ID= 11 , depth= 2,old depth=1
0:8:50.013671885 DEBUG (7): candidate parent finalized ID= 11 , depth= 2,old depth=1
0:8:50.013671885 DEBUG (7): NOTIFYING CHILDREN
0:8:50.013671887 DEBUG (21): candidate parent finalized ID= 12 , depth= 2,old depth=2
0:8:50.013671887 DEBUG (21): candidate parent finalized ID= 12 , depth= 2,old depth=2
0:8:50.014648448 DEBUG (5): candidate parent finalized ID= 12 , depth= 2,old depth=2
0:8:50.014648448 DEBUG (5): candidate parent finalized ID= 12 , depth= 2,old depth=2
0:8:50.014648448 DEBUG (9): candidate parent finalized ID= 12 , depth= 2,old depth=2
0:8:50.014648448 DEBUG (9): candidate parent finalized ID= 12 , depth= 2,old depth=2
0:8:50.014648449 DEBUG (19): candidate parent finalized ID= 12 , depth= 2,old depth=2
0:8:50.014648449 DEBUG (19): candidate parent finalized ID= 12 , depth= 2,old depth=2
0:8:50.014648449 DEBUG (24): candidate parent finalized ID= 12 , depth= 2,old depth=2
0:8:50.014648449 DEBUG (24): candidate parent finalized ID= 12 , depth= 2,old depth=2
0:8:50.022460948 DEBUG (7): Sending NotifyChildrenMsg...
0:8:50.027114871 DEBUG (13): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:3
0:8:50.027114871 DEBUG (14): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:3
0:8:50.027114871 DEBUG (10): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:3
0:8:50.029296886 DEBUG (13): Sending NotifyChildrenMsg...
0:8:50.038085948 DEBUG (10): Sending NotifyChildrenMsg...
0:8:50.043945323 DEBUG (14): Sending NotifyChildrenMsg...
```

Το τελικό αποτέλεσμα:

```
0:16:24.016601603 DEBUG (6): ...
0:16:24.016601603 DEBUG (6): ...
0:16:24.016601603 DEBUG (6): ...
0:16:24.016601603 DEBUG (6): ...
0:16:24.016601603 DEBUG (6): ...
0:16:24.016601603 DEBUG (6): SENDING MESSAGE OF LENGHT:4
0:16:24.016601603 DEBUG (6): Message succesfully sent
0:16:24.022277849 DEBUG (0): ReceiveMeasurementTask(): len=4
0:16:24.022277849 DEBUG (0): received group0
0:16:24.332031271 DEBUG (11): children belong to 1 different groups
0:16:24.332031291 DEBUG (11): Node results :count= 13 , aggregate= 907 , node depth: 1 , group:0 , parent ID:0
0:16:24.332031291 DEBUG (11): Node results :count1= 0 , aggregate1= 0 , node depth: 1 , group:0 , parent ID:0
0:16:24.332031291 DEBUG (11): Node results :count2= 0 , aggregate2= 0 , node depth: 1 , group:0 , parent ID:0

0:16:24.332031291 DEBUG (11): ...
0:16:24.332031291 DEBUG (11): ...
0:16:24.332031291 DEBUG (11): ...
0:16:24.332031291 DEBUG (11): ...
0:16:24.332031291 DEBUG (11): ...
0:16:24.332031291 DEBUG (11): SENDING MESSAGE OF LENGHT:4
0:16:24.332031291 DEBUG (11): Message succesfully sent
0:16:24.337966935 DEBUG (0): ReceiveMeasurementTask(): len=4
0:16:24.337966935 DEBUG (0): received group0
Amazon 2520 DEBUG (0): children belong to 1 different groups
0:16:25.039062540 DEBUG (0): Node results :count= 25 , aggregate= 1680 , node depth: 0 , group:0 , parent ID:0
0:16:25.039062540 DEBUG (0): Node results :count1= 0 , aggregate1= 0 , node depth: 0 , group:0 , parent ID:0
0:16:25.039062540 DEBUG (0): Node results :count2= 0 , aggregate2= 0 , node depth: 0 , group:0 , parent ID:0

0:16:25.039062540 DEBUG (0): FINAL RESULT group 1: 67
0:16:25.039062540 DEBUG (0): FINAL RESULT group 2: 0
0:16:25.039062540 DEBUG (0): FINAL RESULT group 3: 0
0:16:25.039062540 DEBUG (0): #####
0:16:25.039062540 DEBUG (0): ##### Epoch 15 #####
0:16:25.039062540 DEBUG (0): #####
0:16:25.039062540 DEBUG (0): ...
```

Είναι φανερό, ότι ο κώδικας έχει την λειτουργία που ζητήθηκε.

- Fringe case run με 9 κόμβους:

```
0:8:29.000000010 DEBUG (2): Lost parent
0:8:29.000000010 DEBUG (3): Lost parent
0:8:29.000000010 DEBUG (4): Lost parent
0:8:29.000000010 DEBUG (5): Lost parent

0:8:29.005462644 DEBUG (1): reveived help message from node:3
0:8:29.005462644 DEBUG (1): broadcasted rerouting message succesfully
0:8:29.005462644 DEBUG (8): reveived help message from node:3
0:8:29.005462644 DEBUG (8): broadcasted rerouting message succesfully
0:8:29.009628270 DEBUG (7): reveived help message from node:5
0:8:29.009628270 DEBUG (7): broadcasted rerouting message succesfully
0:8:29.009628270 DEBUG (1): reveived help message from node:5
0:8:29.009628270 DEBUG (6): reveived help message from node:5
0:8:29.009628270 DEBUG (0): reveived help message from node:5
0:8:29.009628270 DEBUG (0): broadcasted rerouting message succesfully
0:8:29.011871299 DEBUG (8): reveived help message from node:4
0:8:29.011871299 DEBUG (6): reveived help message from node:4
0:8:29.011871299 DEBUG (6): broadcasted rerouting message succesfully
0:8:29.011871299 DEBUG (0): reveived help message from node:4
0:8:29.011871299 DEBUG (0): broadcasted rerouting message succesfully
0:8:29.014923101 DEBUG (5): received rerouting message from parent candidate
0:8:29.014923101 DEBUG (5): candidate parent ID= 0 , depth= 0
0:8:29.014923101 DEBUG (5): initiate wait timer
0:8:29.014923101 DEBUG (5): candidate parent accepted ID= 0 , depth= 0
0:8:29.014923101 DEBUG (4): received rerouting message from parent candidate
0:8:29.014923101 DEBUG (4): candidate parent ID= 0 , depth= 0
0:8:29.014923101 DEBUG (4): initiate wait timer
0:8:29.014923101 DEBUG (4): candidate parent accepted ID= 0 , depth= 0
0:8:29.018188458 DEBUG (4): received rerouting message from parent candidate
0:8:29.018188458 DEBUG (4): candidate parent ID= 8 , depth= 2
0:8:29.018188458 DEBUG (3): received rerouting message from parent candidate
```

```
0:8:29.018188458 DEBUG (3): candidate parent ID= 8 , depth= 2
0:8:29.018188458 DEBUG (3): initiate wait timer
0:8:29.018188458 DEBUG (3): candidate parent accepted ID= 8 , depth= 2
0:8:29.023162827 DEBUG (5): received rerouting message from parent candidate
0:8:29.023162827 DEBUG (5): candidate parent ID= 0 , depth= 0
0:8:29.023162827 DEBUG (4): received rerouting message from parent candidate
0:8:29.023162827 DEBUG (4): candidate parent ID= 0 , depth= 0
0:8:29.025970435 DEBUG (2): received rerouting message from parent candidate
0:8:29.025970435 DEBUG (2): candidate parent ID= 7 , depth= 1
0:8:29.025970435 DEBUG (2): initiate wait timer
0:8:29.025970435 DEBUG (2): candidate parent accepted ID= 7 , depth= 1
0:8:29.025970435 DEBUG (3): received rerouting message from parent candidate
0:8:29.025970435 DEBUG (3): candidate parent ID= 7 , depth= 1
0:8:29.025970435 DEBUG (3): candidate parent accepted ID= 7 , depth= 1
0:8:29.029510492 DEBUG (5): received rerouting message from parent candidate
0:8:29.029510492 DEBUG (5): candidate parent ID= 1 , depth= 2
0:8:29.029510492 DEBUG (3): received rerouting message from parent candidate
0:8:29.029510492 DEBUG (3): candidate parent ID= 1 , depth= 2
0:8:29.035690289 DEBUG (5): received rerouting message from parent candidate
0:8:29.035690289 DEBUG (5): candidate parent ID= 6 , depth= 2
0:8:29.035690289 DEBUG (4): received rerouting message from parent candidate
0:8:29.035690289 DEBUG (4): candidate parent ID= 6 , depth= 2
0:8:34.014648447 DEBUG (4): candidate parent finalized ID= 0 , depth= 2,old depth=1
0:8:34.014648447 DEBUG (4): candidate parent finalized ID= 0 , depth= 2,old depth=1
0:8:34.014648447 DEBUG (4): NOTIFYING CHILDREN
0:8:34.014648448 DEBUG (5): candidate parent finalized ID= 0 , depth= 2,old depth=1
0:8:34.014648448 DEBUG (5): candidate parent finalized ID= 0 , depth= 2,old depth=1
0:8:34.014648448 DEBUG (5): NOTIFYING CHILDREN
0:8:34.017578135 DEBUG (3): candidate parent finalized ID= 7 , depth= 3,old depth=2
0:8:34.017578135 DEBUG (3): candidate parent finalized ID= 7 , depth= 3,old depth=2
0:8:34.017578135 DEBUG (3): NOTIFYING CHILDREN
0:8:34.019531260 DEBUG (3): Sending NotifyChildrenMsg...
0:8:34.019531260 DEBUG (4): Sending NotifyChildrenMsg...
0:8:34.023437510 DEBUG (5): Sending NotifyChildrenMsg...
0:8:34.025390635 DEBUG (2): candidate parent finalized ID= 7 , depth= 3,old depth=2
0:8:34.025390635 DEBUG (2): candidate parent finalized ID= 7 , depth= 3,old depth=2
0:8:34.025390635 DEBUG (2): NOTIFYING CHILDREN
0:8:34.029342624 DEBUG (8): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:4, old depth:2 new depth:3
0:8:34.032333352 DEBUG (1): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:5, old depth:2 new depth:3
0:8:34.032333352 DEBUG (6): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:5, old depth:2 new depth:3
```

Το τελικό αποτέλεσμα:

```
0:15:7.929687540 DEBUG (7): ...
0:15:7.929687540 DEBUG (7): ...
0:15:7.929687540 DEBUG (7): ...
0:15:7.929687540 DEBUG (7): ...
0:15:7.929687540 DEBUG (7): ...
0:15:7.929687540 DEBUG (7): SENDING MESSAGE OF LENGHT:10
0:15:7.929687540 DEBUG (7): Message succesfully sent
0:15:7.931579629 DEBUG (0): ReceiveMeasurementTask(): len=10
0:15:7.931579629 DEBUG (0): received group2
0:15:8.315429707 DEBUG (4): children belong to 2 different groups
0:15:8.315429727 DEBUG (4): Node results :count= 0 , aggregate= 0 , node depth: 2 , group:1 , parent ID:0
0:15:8.315429727 DEBUG (4): Node results :count1= 1 , aggregate1= 42 , node depth: 2 , group:1 , parent ID:0
0:15:8.315429727 DEBUG (4): Node results :count2= 1 , aggregate2= 77 , node depth: 2 , group:1 , parent ID:0
0:15:8.315429727 DEBUG (4): ...
0:15:8.315429727 DEBUG (4): ...
0:15:8.315429727 DEBUG (4): ...
0:15:8.315429727 DEBUG (4): ...
0:15:8.315429727 DEBUG (4): ...
0:15:8.315429727 DEBUG (4): SENDING MESSAGE OF LENGHT:7
0:15:8.315429727 DEBUG (4): Message succesfully sent
0:15:8.318008458 DEBUG (0): ReceiveMeasurementTask(): len=7
0:15:8.318008458 DEBUG (0): received group1
0:15:9.109375020 DEBUG (0): children belong to 3 different groups
0:15:9.109375040 DEBUG (0): Node results :count= 3 , aggregate= 203 , node depth: 0 , group:0 , parent ID:0
0:15:9.109375040 DEBUG (0): Node results :count1= 3 , aggregate1= 162 , node depth: 0 , group:0 , parent ID:0
0:15:9.109375040 DEBUG (0): Node results :count2= 3 , aggregate2= 229 , node depth: 0 , group:0 , parent ID:0
System Settings
0:15:9.109375040 DEBUG (0): FINAL RESULT group 1: 67
0:15:9.109375040 DEBUG (0): FINAL RESULT group 2: 54
0:15:9.109375040 DEBUG (0): FINAL RESULT group 3: 76
0:15:9.109375040 DEBUG (0): #####
0:15:9.109375040 DEBUG (0): ##### Epoch 14 #####
0:15:9.109375040 DEBUG (0): #####
0:15:9.109375040 DEBUG (0): #####
```

## 7. Bugs που εντοπίστηκαν

Ένα bug το οποίο εντοπίσαμε στον κώδικά μας είναι ότι μερικές φορές όταν ένας κόμβος προσπαθεί να επανασυνδεθεί με τον πατέρα του, δημιουργείται ένα loop, και βρίσκονται περιπτώσεις που δύο κόμβοι είναι και παιδιά και πατέρες ο ένας του άλλου την ίδια στιγμή. Το πρόβλημα αντιμετωπίστηκε σε περιπτώσεις με διαφορά βάθους ίση με 1 με κάποιους απλούς ελέγχους, αλλά προκύπτει και σε loop περισσότερων κόμβων. Η επίλυση αυτού του bug είναι περίπλοκη γι'αυτό και προσπεράστηκε.

```
if ( (mpkt->help) !=parentID) { //DONT TRY TO BECOME THE PARENT OF YOUR
PARENT
```

Μετά την επίλυση παρέμεινε το bug:

```
0:8:34.010742197 DEBUG (1): candidate parent finalized ID= 8 , depth= 2,old depth=2
0:8:34.017578135 DEBUG (7): candidate parent finalized ID= 2 , depth= 4,old depth=1
0:8:34.017578135 DEBUG (7): NOTIFYING CHILDREN
0:8:34.019531260 DEBUG (7): NOTIFY CHILDREN TIMER FIRED
0:8:34.027923565 DEBUG (3): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:5
0:8:34.029296885 DEBUG (3): NOTIFY CHILDREN TIMER FIRED
0:8:34.039092990 DEBUG (2): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:3, old depth:3 new depth:6
0:8:34.041015635 DEBUG (2): NOTIFY CHILDREN TIMER FIRED
0:8:34.044097912 DEBUG (7): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:2, old depth:1 new depth:7
0:8:34.045898448 DEBUG (7): NOTIFY CHILDREN TIMER FIRED
0:8:34.050430302 DEBUG (3): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:8
0:8:34.051757822 DEBUG (3): NOTIFY CHILDREN TIMER FIRED
0:8:34.053802508 DEBUG (2): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:3, old depth:3 new depth:9
0:8:34.055664072 DEBUG (2): NOTIFY CHILDREN TIMER FIRED
0:8:34.059814459 DEBUG (7): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:2, old depth:1 new depth:10
0:8:34.061523448 DEBUG (7): NOTIFY CHILDREN TIMER FIRED
```

```
0:8:34.064804088 DEBUG (3): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:11
0:8:34.066406260 DEBUG (3): NOTIFY CHILDREN TIMER FIRED
0:8:34.070297248 DEBUG (2): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:3, old depth:3 new depth:12
0:8:34.071289072 DEBUG (2): NOTIFY CHILDREN TIMER FIRED
0:8:34.079788188 DEBUG (7): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:2, old depth:1 new depth:13
0:8:34.081054698 DEBUG (7): NOTIFY CHILDREN TIMER FIRED
0:8:34.087615959 DEBUG (3): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:14
0:8:34.088867197 DEBUG (3): NOTIFY CHILDREN TIMER FIRED
0:8:34.094604489 DEBUG (2): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:3, old depth:3 new depth:15
0:8:34.095703135 DEBUG (2): NOTIFY CHILDREN TIMER FIRED
0:8:34.098495497 DEBUG (7): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:2, old depth:1 new depth:16
0:8:34.099609385 DEBUG (7): NOTIFY CHILDREN TIMER FIRED
0:8:34.105285642 DEBUG (3): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:17
0:8:34.106445322 DEBUG (3): NOTIFY CHILDREN TIMER FIRED
0:8:34.112167355 DEBUG (2): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:3, old depth:3 new depth:18
0:8:34.113281260 DEBUG (2): NOTIFY CHILDREN TIMER FIRED
0:8:34.124160733 DEBUG (7): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:2, old depth:1 new depth:19
0:8:34.125976573 DEBUG (7): NOTIFY CHILDREN TIMER FIRED
0:8:34.130828859 DEBUG (3): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:20
0:8:34.131835947 DEBUG (3): NOTIFY CHILDREN TIMER FIRED
0:8:34.141601535 DEBUG (2): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:3, old depth:3 new depth:21
0:8:34.142578135 DEBUG (2): NOTIFY CHILDREN TIMER FIRED
0:8:34.148025511 DEBUG (7): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:2, old depth:1 new depth:22
0:8:34.149414073 DEBUG (7): NOTIFY CHILDREN TIMER FIRED
0:8:34.151535052 DEBUG (3): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:23
0:8:34.153320322 DEBUG (3): NOTIFY CHILDREN TIMER FIRED
0:8:34.161834697 DEBUG (2): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:3, old depth:3 new depth:24
0:8:34.163085947 DEBUG (2): NOTIFY CHILDREN TIMER FIRED
0:8:34.165664688 DEBUG (7): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:2, old depth:1 new depth:25
0:8:34.166992198 DEBUG (7): NOTIFY CHILDREN TIMER FIRED
0:8:34.176483129 DEBUG (3): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:26
0:8:34.177734385 DEBUG (3): NOTIFY CHILDREN TIMER FIRED
0:8:34.181808478 DEBUG (2): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:3, old depth:3 new depth:27
0:8:34.183593760 DEBUG (2): NOTIFY CHILDREN TIMER FIRED
0:8:34.189743036 DEBUG (7): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:2, old depth:1 new depth:28
0:8:34.191406260 DEBUG (7): NOTIFY CHILDREN TIMER FIRED
0:8:34.193344135 DEBUG (3): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:29
0:8:34.194335947 DEBUG (3): NOTIFY CHILDREN TIMER FIRED
0:8:34.202697735 DEBUG (2): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:3, old depth:3 new depth:30
0:8:34.204101572 DEBUG (2): NOTIFY CHILDREN TIMER FIRED
0:8:34.210098261 DEBUG (7): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:2, old depth:1 new depth:31
0:8:34.211914073 DEBUG (7): NOTIFY CHILDREN TIMER FIRED
0:8:34.214477554 DEBUG (3): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:32
0:8:34.215820322 DEBUG (3): NOTIFY CHILDREN TIMER FIRED
0:8:34.223892195 DEBUG (2): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:3, old depth:3 new depth:33
0:8:34.225585947 DEBUG (2): NOTIFY CHILDREN TIMER FIRED
0:8:34.236389127 DEBUG (7): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:2, old depth:1 new depth:34
0:8:34.238281260 DEBUG (7): NOTIFY CHILDREN TIMER FIRED
0:8:34.241577159 DEBUG (3): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:35
0:8:34.243164072 DEBUG (3): NOTIFY CHILDREN TIMER FIRED
0:8:34.253753630 DEBUG (2): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:3, old depth:3 new depth:36
0:8:34.254882822 DEBUG (2): NOTIFY CHILDREN TIMER FIRED
0:8:34.264724704 DEBUG (7): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:2, old depth:1 new depth:37
0:8:34.266601573 DEBUG (7): NOTIFY CHILDREN TIMER FIRED
0:8:34.271286014 DEBUG (3): RECEIVED CHANGE DEPTH MESSAGE FROM PARENT:7, old depth:2 new depth:38
0:8:34.272460947 DEBUG (3): NOTIFY CHILDREN TIMER FIRED
```

## 6. Τρόπος εργασίας

- Υλοποίηση Help Message: Ιωαννίδης Χρήστος-Παπαματθαίακη Ηλέκτρα-Δέσποινα
- Υλοποίηση επαναποστολής Routing μηνύματος: Ιωαννίδης Χρήστος-Παπαματθαίακη Ηλέκτρα-Δέσποινα
- Υλοποίηση αλλαγής χρονισμού: Ιωαννίδης Χρήστος
- Ειδοποίηση παιδιών για αλλαγή επιπέδου γονέα: Ιωαννίδης Χρήστος
- Report: Ιωαννίδης Χρήστος-Παπαματθαίακη Ηλέκτρα-Δέσποινα

Όπως και στο πρώτο μέρος της εργασίας αυτής, συνεχίσαμε να χρησιμοποιούμε το Github για τους ίδιους λόγους που είχαμε αναφέρει, δηλαδή την διευκόλυνση που προσφέρει στην συγγραφή του κώδικα εξ αποστάσεως και για την βελτίωση του workflow. Αυτή τη φορά παρατηρήσαμε ότι ήταν πιο βολικό το να βρισκόμαστε από κοντά και να γράφουμε κώδικα στον ίδιο υπολογιστή, χωρίς αυτό να σημαίνει ότι δεν δουλέψαμε και ανεξάρτητα. Μπορείτε να δείτε παρακάτω ένα στιγμιότυπο από την νέα εργασία στον ιστότοπο (και αυτό το repository μέχρι να ολοκληρωθεί η εργασία, είναι private). Τέλος, όπως και καθόλη την διάρκεια της εργασίας εργαστήκαμε και οι δύο ενεργά, συζητώντας τα βήματα που θα ακολουθήσουμε πριν την υλοποίηση τους και ανταλλάσσοντας πληροφορίες σχετικά με την πρόοδο που επιτυγχάναμε κάθε φορά.

