

Επεξεργασία και Διαχείριση Δεδομένων σε Δίκτυα Αισθητήρων

2η εργαστηριακή αναφορά (LEACH)

Κωνσταντίνος Πεσλής AM:2014030074

Σοφοκλής Φιλάρετος Γαβριηλίδης AM:2014030062

Βοηθητικό Πρόγραμμα:

Η υλοποίηση του βοηθητικού προγράμματος, για την δημιουργία του topology.txt, έγινε στο αρχείο myTopology.py σε γλώσσα python .

Πρόγραμμα 2:

A) Τα κομμάτια του κώδικα που αφαιρέσαμε εμείς ήταν τα εξής:

- Όλα τα μηνύματα που εμφανιζόταν με την εντολή printf.
- Όλα τα interfaces που χρησιμοποιήθηκαν για serial port.
- Όλα τα interfaces που χρησιμοποιήθηκαν για τα leds.
- Στη receiveRoutingTask() , τον κώδικα που ειδοποιούσε κάθε κόμβος τον πατέρα του ότι έλαβε κάποιο μήνυμα.

Κώδικα που παραποιήσαμε:

- Στη RoutingMsgTimer.fired() αν ο κόμβος ήταν ο TOS_NODE_ID==0 καλούνταν ξανά η RoutingMsgTimer.fired(). Εμείς βάλαμε να καλείται η δικιά μας συνάρτηση, NotifyParentMsgTask, στην οποία ουσιαστικά εκτυπώνουμε τα AVERAGE και VARIANCE , ή ετοιμάζουμε την πληροφορία για να την στείλουμε στον πατέρα.

Β) Αρχικά ,προσθέσαμε στο event Boot() την αρχικοποίηση των μεταβλητών των κόμβων που χρειαζόμαστε για την υλοποίηση του leach ,όπως τις τιμές που χρειαζόμαστε για το average και variance , το κατάλληλο αρχικό parentID και τις boolean μεταβλητές για τους leaders .Επίσης καλούμε την συνάρτηση valueGenerator για να δώσουμε τιμή στους κόμβους(όπως στο tag)και την συνάρτηση leaderGenerator για να επιλέξουμε τους leaders σε κάθε γύρα με πιθανότητα 25%.

Έπειτα στην RoutingMsgTimer.fired() προσθέσαμε ,πριν το broadcast, συνθήκη για το αν ο κόμβος μας είναι leader και δεν υπήρξε leader στις προηγούμενες 3 γύρες.

Στην συνέχεια στην receiveRoutingTask() προσθέσαμε ένα έλεγχο ,ωστε να μην είναι leader ο κόμβος που θα πάρει το parentID από την RoutingMsgPacket και καλούμε την συνάρτηση NotifyParentMsgTimer που είναι η συνάρτηση που δημιουργήσαμε στο tag (και αλλάξαμε μερικώς για το leach) για την αποστολή των μηνυμάτων/τιμών για τον υπολογισμό των average και variance.Επίσης στην περίπτωση που ένας κόμβος λάβει παραπάνω από ένα routing message από leader , υλοποιήσαμε συνθήκη για την τυχαία επιλογή leader με πιθανότητα 20% για την αντικατάσταση του parentID.

Στο επόμενο βήμα μας , στην NotifyParentMsgTimer έχουμε αν είναι ο κόμβος 0 υπολογίζουμε το τελικό average και variance και αρχικοποιούμε τις τιμές για την έναρξη νέου γύρου και καλούμε την RoutingMsgTimer, αλλιώς οι υπόλοιποι κόμβοι εκχωρούν τις απαραίτητες τιμές στο μήνυμα που θα στείλουν,αφού πρώτα ελέγχουμε ώστε να έχουν το σωστό parentID .Αν είναι leader έχει parentID=0 ,ενώ άμα δεν έχει ακούσει την διαφήμιση κάποιου τότε γίνεται μερικώς leader και στέλνει απευθείας στον κόμβο 0 και αυτός.

Συμπληρωματικά, για την υλοποίηση του leach , στην receiveNotifyTask() αλλάξαμε το πώς κρατάμε σωστά τα κατάλληλα δεδομένα για τον υπολογισμό του average και του variance.Σε αυτό το σημείο αθροίζουμε τις απαραίτητες μεταβλητές του implementation του κόμβου με αυτές των μηνυμάτων που έλαβε .

Τέλος , στην NotifyAmSend.SendDone() υλοποιούμε την αρχικοποίηση των μεταβλητών των κόμβων για τον επόμενο γύρο και ανα 4 γύρους κάνουμε false τις boolean των leaders .

Γ) Αρχικά είχαμε dbg μηνύματα και κλήσεις μόνο για να υλοποιήσουμε σωστά το routing με βάση την leach.Στην συνέχεια , προσθέσαμε dbg σε όλους τους ελέγχους που χρειάστηκαν για το 2ο πρόγραμμα για τους leader ,καθώς και στις αλλαγές receiveNotifyMsg για να δούμε αν στέλνονται σωστά οι τιμές .Στο τέλος , αφήσαμε dbg μόνο για την εμφάνιση των γύρων και ζητουμένων.

Δ)

```
tinyos@tinyos-VirtualBox: ~/Desktop/Leach
23 19 -50
24 19 -50
24 23 -50
24 18 -50
0:2:40.200195342 DEBUG (0):
#####
0:2:40.200195342 DEBUG (0): ##### ROUND 0 #####
0:2:40.200195342 DEBUG (0): #####
0:2:40.200195342 DEBUG (0): sum 1
0:2:40.200195342 DEBUG (0): Average ts:1.000000
0:2:40.200195342 DEBUG (0): Variance ts:0.000000
0:2:40.200195342 DEBUG (0): ROUND ts:0
0:2:40.200195342 DEBUG (0): children ts:1
0:5:10.200195342 DEBUG (0):
#####
0:5:10.200195342 DEBUG (0): ##### ROUND 1 #####
0:5:10.200195342 DEBUG (0): #####
0:5:10.200195342 DEBUG (0): sum 5
0:5:10.200195342 DEBUG (0): Average ts:5.000000
0:5:10.200195342 DEBUG (0): Variance ts:0.000000
0:5:10.200195342 DEBUG (0): ROUND ts:1
0:5:10.200195342 DEBUG (0): children ts:1
0:7:40.200195342 DEBUG (0):
#####
0:7:40.200195342 DEBUG (0): ##### ROUND 2 #####
0:7:40.200195342 DEBUG (0): #####
0:7:40.200195342 DEBUG (0): sum 3
0:7:40.200195342 DEBUG (0): Average ts:3.000000
0:7:40.200195342 DEBUG (0): Variance ts:0.000000
0:7:40.200195342 DEBUG (0): ROUND ts:2
0:7:40.200195342 DEBUG (0): children ts:1
0:10:10.200195342 DEBUG (0):
#####
0:10:10.200195342 DEBUG (0): ##### ROUND 3 #####
0:10:10.200195342 DEBUG (0): #####
0:10:10.200195342 DEBUG (0): sum 9
0:10:10.200195342 DEBUG (0): Average ts:9.000000
0:10:10.200195342 DEBUG (0): Variance ts:0.000000
0:10:10.200195342 DEBUG (0): ROUND ts:3
0:10:10.200195342 DEBUG (0): children ts:1
0:12:40.200195342 DEBUG (0):
#####
0:12:40.200195342 DEBUG (0): ##### ROUND 4 #####
0:12:40.200195342 DEBUG (0): #####
0:12:40.200195342 DEBUG (0): sum 22
0:12:40.200195342 DEBUG (0): Average ts:22.000000
0:12:40.200195342 DEBUG (0): Variance ts:0.000000
0:12:40.200195342 DEBUG (0): ROUND ts:4
0:12:40.200195342 DEBUG (0): children ts:1
0:15:10.200195342 DEBUG (0):
#####
0:15:10.200195342 DEBUG (0): ##### ROUND 5 #####
0:15:10.200195342 DEBUG (0): #####
0:15:10.200195342 DEBUG (0): sum 21
```

Το παραπάνω παράδειγμα είναι από τρέξιμο του προγράμματος για 25 κόμβους σύμφωνα με το αρχείο τοπολογίας που παράχθηκε από το βοηθητικό μας πρόγραμμα . Δυστυχώς όπως φαίνεται στην εικόνα το πρόγραμμα δεν τρέχει σωστά (στέλνει μόνο του τελικού κόμβου) , καθώς στην προσπάθειά μας για την βελτίωση του προγράμματος μας όταν έβγαζε αποτελέσματα για μισούς ή λιγότερους κόμβους μας χαλάσαμε τον συγχρονισμό των κλήσεων και δεν προλάβαμε να το διορθώσουμε ,ούτε κρατήσαμε το version που έτρεχε καλύτερα ,αλλά όχι πλήρως σωστά.

Ε)Ολόκληρη η εργασία εκπονήθηκε ομαδικώς.