

Τμήμα Πληροφορικής και Τηλεπικοινωνιών Ασύρματα Δίκτυα Αισθητήρων (DI367)

Περίοδος Ιουλίου 2018

Όνομα: Αικατερίνα
Επίθετο: Μάστοράκη
A.M. : 1115201400100

Περιγραφή

Το αρχείο στο οποίο έγινε η διεξαγωγή της άσκησης ονομάζεται PEAS.java και είναι το midlet που κάνουμε deploy στα virtual nodes του Sunspot μέσω του προσομοιωτή solarium. Άκολουθησαν τα βήματα για την χρήση της πλατφόρμας που υπήρχαν στα tutorials του sunspot στις διαφάνειες του eclass και ήρθα αντιμέτωπη με το εξής πρόβλημα:

1. Δημιούργησα το .jar file μέσω της εντολής `ant jar-app` ώστε να φτιαχτεί το ant script που χρειάζεται το virtual node για να κάνει deploy το midlet bundle
2. Έφτιαξα το manifest file(MANIFEST.MF) και πρόσθεσα το package στο οποίο συμπεριλάμβανε το PEAS.java
`MIDlet-1: PEAS, , net.java.dev.netbeansspot.PEAS.PEAS`
3. Έκανα αντιγραφή των template των παραδειγμάτων project επειδή δεν μπορούσα να προσθέσω την J2ME πλατφόρμα, αφού δεν την αναγνώριζε, με αποτέλεσμα να μην μπορώ να φτιάξω από την αρχή δικό μου project με την J2ME embedded form που ζητάγε η άσκηση.
4. Δυστυχώς, στο run του midlet στο output console μου έβγαζε exception στα thread του κάτι το οποίο με εμπόδιζε να τεστάρω τα αποτελέσματα που το πρόγραμμά μου θα έπρεπε να βγάζει.

Υλοποίηση

Ξεκινώντας, γίνονται τα απαραίτητα import που χρειάζονται για την εκτέλεση του προγράμματος. Η βασική μας κλάση, η οποία επεκτείνει το ήδη υπάρχον midlet είναι η PEAS, η οποία περιλαμβάνει τις εξής μεταβλητές:

- **Node_state** που δείχνει την κατάσταση στην οποία βρίσκεται ο κόμβος, δηλαδή αν είναι ενεργός, αν κοιμάται ή αν στέλνει probes
- **I(λ)** είναι ο ρυθμός με τον οποίο στέλνονται τα probes
- **B_radius** είναι η ακτίνα επικοινωνίας, δηλαδή η απόσταση που καθορίζει αν κάποιος κόμβος μπορεί να ακούσει τα μηνύματα που στέλνει κάποιος άλλος κόμβος
- **MAC** που καθορίζει την mac address του κάθε κόμβου
- **x,y** οι συντεταγμένες του κάθε κόμβου που δημιουργούνται τυχαία στο διάστημα 0,100
- **Energy** που είναι η ενέργεια που έχει κάθε κόμβος στην αρχή της λειτουργίας του
- **S_time** που είναι ο χρόνος ύπνου του κάθε κόμβου

Συνεχίζοντας υπάρχει η συνάρτηση `ln` που υπολογίζει τον νεπέριο λογάριθμο με βάση μια συνάρτηση την οποία βρήκα από το internet.

Η βασική υλοποίηση της άσκησης έχει γίνει με την χρήση 2 threads:

- Broadcast_Thread
- Report_Thread

Το **Report_Thread** είναι το νήμα μέσω του οποίου γίνονται όλες οι εκτυπώσεις που αφορούν την κατάσταση του κόμβου. Το νήμα αυτό τρέχει χωρίς καμία διακοπή και ανά τακτά χρονικά διαστήματα τυπώνει στην κονσόλα εξόδου του προγράμματος, πληροφορίες για τον κόμβο όπως την κατάσταση στην οποία βρίσκεται καθώς και την MAC διεύθυνσή του (δεν μπόρεσα να τυπώσω το timestamp γιατί είχα θέμα με τον ορισμό την αντίστοιχης κλάσης)

Το **Broadcast_Thread** είναι το νήμα στο οποίο γίνεται όλη η λειτουργία του αλγορίθμου PEAS. Αρχικά ελέγχεται η κατάσταση στην οποία βρίσκεται ο κόμβος. Αν ο κόμβος βρίσκεται σε κατάσταση probing και αν η ενέργεια και η απόσταση το επιτρέπουν τότε στέλνεται το broadcast μήνυμα μέσω ενός datagram. Το μήνυμα αυτό περιέχει τις συντεταγμένες του κόμβου και την MAC διεύθυνσή του. Θεωρητικά όταν ένας κόμβος λάμβανε το μήνυμα αυτό θα έπαιρνε τις συντεταγμένες του κόμβου που το έστειλε και θα του απάντούσε με ένα άμεσο μήνυμα προς αυτόν κάτι το οποίο δεν έχει υλοποιηθεί εδώ καθώς αντιμετωπίσα πρόβλημα με την συναρτησή receive. Τέλος υπολογίζεται το νέο sleeping time μέσω του τύπου που μας δώθηκε και μειώνεται η ενέργεια του κόμβου.

Δυστυχώς δεν είχα αρκετό χρόνο να ολοκληρώσω την άσκηση καθώς έχασά πολλές μέρες στο να ανακαλύψω πως ακριβώς έπρεπε να γίνει η όλη υλοποίηση.