

Εργασία στο TinyOS

Βασική σελίδα με περιγραφή

http://tinyos.stanford.edu/tinyos-wiki/index.php/TinyOS_Tutorials

ΒΗΜΑ 1: ΕΓΚΑΤΑΣΤΑΣΗ

Στο

<https://drive.google.com/file/d/18CgwCw3XQMbafp2nrqmpaUWxQL4cNEzH/view?usp=sharing>

θα βρείτε ένα virtual machine (για Ubuntu 64bit λειτουργικά) όπου είναι εγκατεστημένο το TinyOS και ο βοηθητικός κώδικας. Εγώ το τρέχω από VirtualBox.

Ο χρήστης μου λέγεται tinyos και ο κωδικός είναι ο tinyosRULES.

ΠΡΟΣΟΧΗ: Σε καμία περίπτωση μην κάνετε αυτόματες ενημερώσεις λογισμικού στο virtual machine, καθώς αν κατεβάσετε νέες εκδόσεις του μεταγλωττιστή θα χρειαστείτε νέα εγκατάσταση (που δε θα σας κάνω ατομικά εγώ) του TinyOS. Εγώ έχω απενεργοποιήσει τις αυτόματες ενημερώσεις στο vdi που σας δίνω – μην τις ενεργοποιήσετε. Μπορείτε αν θέλετε να εγκαταστήσετε χειροκίνητα κάποιες ενημερώσεις.

Μπορείτε από τα settings να αλλάξετε (αν θέλετε) τις ρυθμίσεις της ανάλυσης της οθόνης (Display), σύμφωνα με αυτές του υπολογιστή σας.

Σημείωση: Η συγκεκριμένη έκδοση του Ubuntu έχει ως editors προγράμματα όπως τα gedit, emacs, nano, pico, vi, αλλά όχι άλλα όπως το sublime. Μπορείτε χειροκίνητα να εγκαταστήσετε κάποιο editor της επιλογής σας.

ΒΗΜΑ 2: ΒΟΗΘΗΤΙΚΟΣ ΚΩΔΙΚΑΣ

Ο βοηθητικός κώδικας βρίσκεται στο φάκελο `/home/tinyos/local/src/tinyos-2x/apps/tinyOS`.

Ο φάκελος έχει μέσα ένα αρχείο Makefile και ένα αρχείο README. Δοκιμάστε να κάνετε compile τον κώδικα με την εντολή:

make micaz sim

Τρέχετε το simulation με την εντολή: **python ./mySimulation.py**

ΒΗΜΑ 3: ΚΑΤΑΝΟΗΣΗ ΒΟΗΘΗΤΙΚΟΥ ΚΩΔΙΚΑ

1. ΤΟΠΟΛΟΓΙΑ

Τα αρχεία topology.txt και topology2.txt περιέχουν πληροφορίες για τη συνδεσιμότητα 2 κόμβων σε simulation mode. Εξετάζοντας το αρχείο topology.txt, γειτονικά ζευγάρια κόμβων είναι τα:

0 με 1

1 με 4

1 με 7

1 με 2

4 με 5

Όσα ζευγάρια δεν αναφέρονται στο topology.txt δε θα επικοινωνούν μεταξύ τους.

2. Αρχείο του Simulation

Τα αρχικά στάδια της εργασίας σας θα γίνουν σε simulation mode. Συνεπώς, είναι σημαντικό να κατανοήσετε (σε γενικές γραμμές) το αρχείο

mySimulation.py

το οποίο τρέχει την προσομοίωση.

Οι γραμμές:

```
from TOSSIM import *  
import sys ,os  
import random  
t=Tossim([])
```

δημιουργούν ένα αντικείμενο του Tossim.

Οι γραμμές:

```
for i in range(0,10):  
m=t.getNode(i) m.bootAtTime(10*t.ticksPerSecond() + i)
```

εκκινεί τους κόμβους (από 0..9) σε λίγο διαφορετικές στιγμές.

Οι εντολές

```
f=sys.stdout  
t.addChannel("Boot",f)  
t.addChannel("RoutingMsg",f)  
t.addChannel("NotifyParentMsg",f)  
t.addChannel("Radio",f) #t.addChannel("Serial",f)  
t.addChannel("SRTreeC",f)  
#t.addChannel("PacketQueueC",f)
```

καθορίζουν ποια dbg μηνύματα θα εκτυπώνονται στην οθόνη. Αν πχ το πρώτο όρισμα ενός dbg μηνύματος είναι Boot, το αντίστοιχο μήνυμα θα εκτυπωθεί, αφού έχει προστεθεί το κανάλι Boot. Το σύμβολο '#' είναι το σύμβολο για σχόλιο γραμμής.

Προσέξτε ότι οι γραμμές κώδικα που δημιουργούν το μοντέλο θορύβου απαιτούν αρκετή μνήμη για το VM σας. Αν σας σκάει το πρόγραμμα σε αυτό το σημείο, δοκιμάστε με λιγότερους (πχ 36) κόμβους.

3. Ουρά πακέτων

Τα αρχεία PacketQueue.nc και PacketQueueC.nc παρέχουν ένα interface και ένα module (αντίστοιχα) για μία ουρά πακέτων. Αυτές οι ουρές χρησιμοποιούνται συχνά στο βοηθητικό κώδικα που σας παρέχεται.

4. Κύριο module: SRTreeC.nc και configuration: SRTreeAppC.nc

Το κεντρικό αρχείο είναι το SRTreeAppC.nc, όπου γίνεται η καλωδίωση (wiring) όλων των components. Θα πρέπει να ξεχωρίσετε:

- a) το component MainC (για την εκκίνηση – θα υλοποιήσουμε το event Booted στο SRTreeC),
- b) το component LedsC για το χειρισμό των Leds του αισθητήρα,
- c) για αποστολή/λήψη μηνυμάτων στον ασύρματο, τα components ActiveMessageC, AMSenderC (2: ένα για routing μηνύματα, και ένα για μηνύματα προς τον πατέρα), και AMReceiverC (2: ένα για routing μηνύματα, και ένα για μηνύματα προς τον πατέρα),
- d) για αποστολή/λήψη μηνυμάτων στο serial port, τα components SerialActiveMessageC, SerialAMSenderC, και SerialAMReceiverC,
- e) Components για τις ουρές μηνυμάτων σχετικά με την αποστολή/λήψη των αντίστοιχων 2 τύπων μηνυμάτων,
- f) Διάφορους μετρητές (Timers) με ακρίβεια ms.

Αναζητήστε πως ο κόμβος 0 ξεκινάει ένα μετρητή για να στείλει το πρώτο routing μήνυμα. Παρατηρούμε ότι όταν χτυπάει ο μετρητής, κάνει στο τέλος post ένα task για την πραγματική αποστολή του μηνύματος.

Αντίστοιχα, στο αντίστοιχο task (receiveRoutingTask) λήψης του μηνύματος, ο κάθε κόμβος που δεν είχε πατέρα θέτει το επίπεδό του και τον πατέρα του, και στη συνέχεια στέλνει ένα μήνυμα NotifyParent στον πατέρα του, προτού προωθήσει το μήνυμα. Το συγκεκριμένο task είναι αρκετά σύνθετο, καθώς περιέχει και επιλογές για την περίπτωση που ενημερωθούμε για έναν καλύτερο πατέρα από αυτόν που έχουμε αρχικά επιλέξει.

ΕΡΓΑΣΙΑ

Η εργασία αποτελείται από κομμάτια, τα οποία βασίζονται το ένα στο άλλο:

- Ένα βοηθητικό πρόγραμμα που θα δημιουργεί τις τοπολογίες στις οποίες θα ελέγξετε το πρόγραμμά σας.
- Το πρώτο κομμάτι θα σας μάθει πώς να χρησιμοποιείτε μετρητή (ρολόι) και να στέλνετε/λαμβάνετε μηνύματα, υλοποιώντας μία συναθροιστική συνάρτηση **σύμφωνα με το TAG**.
- Στο δεύτερο κομμάτι θα πρέπει να υλοποιήσετε μια λειτουργία σύνδεσης των κόμβων που έχουν χάσει είτε το ερώτημα είτε την επαφή με τον πατέρα τους.
- Υπάρχει και ένα επιπλέον κομμάτι για τους μεταπτυχιακούς φοιτητές μόνο.

Πιο συγκεκριμένα, καλείστε να κάνετε τα ακόλουθα.

ΒΟΗΘΗΤΙΚΟ ΠΡΟΓΡΑΜΜΑ (Προθεσμία: 08/12/2023)

Για να μπορέσετε να ελέγξετε τη λειτουργία των προγραμμάτων σας, απαιτείται η δημιουργία αρχείων τοπολογίας, που θα περιλαμβάνουν πολλούς αισθητήρες, με «αυτόματο» τρόπο. Στο βοηθητικό πρόγραμμα αυτό καλείστε να δημιουργήσετε (σε όποια γλώσσα προγραμματισμού θέλετε) ένα πρόγραμμα που να:

- Παίρνει ως παραμέτρους 1 ακέραιο (θα αναφερόμαστε σε αυτόν ως διάμετρος D) και 1 αριθμό κινητής υποδιαστολής (θα αναφερόμαστε σε αυτόν ως εμβέλεια).
- Θα δημιουργεί $D \times D$ κόμβους, με αναγνωριστικά από 0 έως D^2-1 , τοποθετημένους σε ένα grid μεγέθους $D \times D$. Ο κόμβος j θα ανήκει στη γραμμή j/D και στη στήλη $j \% D$.
- Θεωρώντας ότι οι οριζόντιες και κάθετες αποστάσεις των κόμβων στο grid είναι ίσες με 1, είναι εύκολο για έναν οποιοδήποτε κόμβο να βρείτε όλους τους κόμβους που βρίσκονται σε απόσταση μικρότερη ή ίση με την εμβέλειά του (δεύτερη παράμετρος του προγράμματος). Πχ, αν η εμβέλεια είναι 1.5, τότε ένας κεντρικός κόμβος έχει 8 γείτονες (σε σχηματισμό αστεριού γύρω από αυτόν, δηλαδή πάνω/κάτω/αριστερά/δεξιά και στις διαγωνίους).
- Αν για κάθε κόμβο βρείτε τους γείτονές του, τότε μπορείτε αυτή την πληροφορία να τη χρησιμοποιήσετε για να δημιουργήσετε ένα αρχείο τοπολογίας, με παρόμοια μορφή με αυτή που σας δίνεται στο αρχείο topology.txt.

ΠΡΟΓΡΑΜΜΑ 1 (Προθεσμία: 08/12/2023)

Ο σταθμός βάσης επιλέγει να εκτελέσει group by queries, ορίζοντας τυχαία τον τρόπο ομαδοποίησης των κόμβων. Πιο συγκεκριμένα:

- Η συναθροιστική συνάρτηση που θα πρέπει να υπολογίζεται ανά ομάδα θα είναι πάντα η AVG (μέσος όρος).
- Ο σταθμός βάσης, πριν στείλει το ερώτημα στο δίκτυο θα επιλέγει τυχαία έναν ακέραιο αριθμό RANDOM_NUM από το 1 έως και το 3 (δηλαδή, 1, 2 ή 3).
- Κάθε κόμβος με αναγνωριστικό id θα ανήκει στο group $id \% RANDOM_NUM$. Συνεπώς, αν $RANDOM_NUM=1$, όλοι οι κόμβοι θα ανήκουν στο ίδιο group.

Η λειτουργία να τερματίζεται μετά από 900 δευτερόλεπτα. Ο σταθμός βάσης (κόμβος 0) θα πρέπει να τυπώνει το τελικό αποτέλεσμα σε κάθε εποχή, η οποία διαρκεί 60 δευτερόλεπτα.

Κάθε αισθητήρας θα πρέπει στην πρώτη εποχή να παράξει μία τυχαία ακέραια τιμή στο διάστημα [1..100] ως μέτρησή του. Σε κάθε επόμενη εποχή, η μέτρηση ενός αισθητήρα θα επιλέγεται τυχαία, χωρίς όμως να απέχει παραπάνω από 20% από την προηγούμενη μέτρησή του, και πάντα στο διάστημα [1..100]. Ο κάθε αισθητήρας θα πρέπει να προωθεί στον πατέρα του κατάλληλη πληροφορία ώστε να μπορεί να υπολογιστεί τελικά η ζητούμενη συναθροιστική συνάρτηση σε όλο το δίκτυο.

Η λειτουργία του κώδικά σας ΠΡΕΠΕΙ να γίνεται **ΣΥΜΦΩΝΑ ΜΕ ΤΟ TAG**, τροποποιώντας κατάλληλα πράγματα που υπάρχουν στο βοηθητικό κώδικα που σας δίνω, όταν κάτι δε συμβαδίζει με τη λογική του TAG, ιδίως κατά τη διαδικασία του routing.

Σας ζητείται να υλοποιήσετε τη βελτιωμένη λογική του TAG, όπως την περιγράψαμε στο μάθημα. Προσέξτε ότι ανάλογα με την τιμή του RANDOM NUM, καθώς και ανάλογα με το πόσα groups βρίσκονται στο υποδέντρο του, αλλάζει ο όγκος της πληροφορίας που θα χρειαστεί να μεταδίδει ο κάθε κόμβος.

Το συγκεκριμένο πρόγραμμα ΔΕΝ απαιτεί συγχρονισμό των κόμβων με βάση το TAG και δεν απαιτεί να ανοιγοκλείνετε τον ασύρματο. Θα πρέπει να προσπαθήσετε (μπορεί να μην την καταφέρετε τέλεια, αλλά έστω μερικώς θα πρέπει να γίνει) να πετύχετε όμως τη λογική του TAG (πρώτα μεταδίδουν τα παιδιά, μετά οι γονείς) χρησιμοποιώντας το επίπεδο του κάθε κόμβου στο δέντρο. Σκεφτείτε ότι η λειτουργία που θα κάνει ο κάθε κόμβος θα είναι επαναλαμβανόμενη και θα εξαρτάται από ένα μετρητή. Αν το σκεφτείτε καλά, πρέπει να ρυθμίσετε τότε θα χτυπήσει πρώτη φορά ο μετρητής αυτός.

Επίσης, θα πρέπει ο χρόνος στον οποίο θα λειτουργεί ένας κόμβος να ελαχιστοποιηθεί, καθώς ζητούμενο είναι οι κόμβοι να κοιμούνται για όσο το δυνατόν μεγαλύτερο διάστημα.

Το πρόγραμμά σας θα πρέπει να δουλεύει για οποιοδήποτε αρχείο topology.txt. Η μοναδική υπόθεση που μπορείτε να κάνετε (αν χρειαστεί) είναι για το μέγιστο αριθμό παιδιών κάθε κόμβου (πχ, 16, ανάλογα με το πώς δημιουργήσατε την τοπολογία σας), ή το μέγιστο αριθμό κόμβων στο δίκτυο (πχ, 100).

ΠΡΟΓΡΑΜΜΑ 2 (Προθεσμία: 22/12/2023)

Σας ζητείται να υλοποιήσετε μια λειτουργία σύνδεσης των κόμβων που έχουν χάσει είτε το ερώτημα είτε την επαφή με τον πατέρα τους. Πιο συγκεκριμένα:

- Μετά από 500 δευτερόλεπτα ο κάθε κόμβος δεν έχει λάβει το routing μήνυμα θα πρέπει να στείλει ένα help μήνυμα, ζητώντας έτσι από γειτονικούς κόμβους να του μεταδώσουν ξανά το routing μήνυμα.
- Το ίδιο θα κάνει, την ίδια χρονική στιγμή, και κάθε κόμβος (εκτός από τον κόμβο 0) που είχε λάβει το routing μήνυμα, αλλά με πιθανότητα 2%. Αυτό το σενάριο προσομειώνει την πιθανότητα απώλειας σύνδεσης με τον πατέρα του κόμβου. Σημειώστε ότι δεν απαγορεύεται, αν λάβει απάντηση στο help μήνυμα από τον προηγούμενο πατέρα του, ένας κόμβος να μην αλλάξει πατέρα μετά από αυτή τη διαδικασία.
- Θα πρέπει να περιμένετε μερικά ms για να επιλέξετε νέο πατέρα στην περίπτωση των help μηνυμάτων, καθώς το ιδανικό είναι να μην αλλάξει το επίπεδο του κόμβου, ώστε να μη χρειαστεί επαναπρογραμματισμός του χρονισμού του υποδέντρου του. Όπως αναλύσαμε στο μάθημα, αν αλλάξει το επίπεδο ενός κόμβου, αλλάζει ο χρονισμός του, και αυτό ίσως οδηγήσει κόμβους στο υποδέντρο του να αλλάξουν επίσης επίπεδο.

ΠΡΟΓΡΑΜΜΑ 3 (Προθεσμία: 22/12/2023)

Το Πρόγραμμα 3 αφορά μόνο ομάδες με μεταπτυχιακούς φοιτητές. Ομάδες προπτυχιακών φοιτητών που θα το υλοποιήσουν με δική τους πρωτοβουλία θα πάρουν μέγιστο μπόνους (ανάλογα με το πόσο σωστά θα υλοποιήσουν το συγκεκριμένο πρόγραμμα) βαθμού του

πρότζεκτ ίσο με 15%.

Σας ζητείται να υλοποιήσετε ένα μηχανισμό όπου σε κάποια τυχαία εποχή ο σταθμός βάσης αποφασίζει να τροποποιήσει το RANDOM_NUM και να στείλει ξανά το ερώτημα στο δίκτυο με τη νέα τιμή του. Προσέξτε ότι οι κόμβοι ΔΕΝ είναι απαραίτητο να αλλάξουν τον πατέρα τους σε αυτή την περίπτωση (εκτός και αν δεν είχαν λάβει το αρχικό routing μήνυμα).

Θα λειτουργεί σωστά το πρόγραμμά σας αν κάποιος κόμβος δε λάβει το νέο ερώτημα με το διαφορετικό RANDOM_NUM;

ΤΙ ΘΑ ΠΑΡΑΔΩΣΕΤΕ (σε 2 φάσεις παράδοσης)

1. Κώδικα με σχόλια

2. Μία αναφορά (1 σε κάθε φάση παράδοσης) που θα περιγράφει ΑΝΑΛΥΤΙΚΑ

- a. Τι διορθώσατε στο αρχείο που σας δόθηκε και γιατί. Βρήκατε κομμάτια που δεν εκτελούνται ποτέ και τα αφαιρέσατε; Αν ναι, ποια; Βρήκατε κομμάτια που δε λειτουργούσαν σύμφωνα με τις αρχές του TAG; Αν ναι, ποια;
- b. Σημαντικά κομμάτια κώδικα για κάθε ερώτημα και επεξήγηση.
- c. Εξήγηση για τα περιεχόμενα κάθε μηνύματος που στέλνετε – σε τι σας χρησιμεύει το κάθε πεδίο;
- d. Παραδείγματα με πίνακες που θα έχουν συγκεκριμένο δέντρο, δεδομένα κόμβων και το τι υπολόγισε στη ρίζα του δέντρου το πρόγραμμά σας.
- e. Ποιος φοιτητής υλοποίησε ποιο κομμάτι. Η δήλωση αυτή πρέπει να είναι απολύτως ακριβής. Σε περίπτωση αναληθών δηλώσεων θα υπάρχει βαθμολογική ποινή σε όλα τα μέλη της ομάδας.

Η αναφορά βαθμολογείται και είναι σημαντικό να είναι αναλυτική και να περιγράφει την πληροφορία που αναφέρθηκε παραπάνω. Ελλιπείς ή ανύπαρκτες αναφορές θα οδηγήσουν σε απώλεια βαθμών, άσχετα με το πόσο καλά τρέχει ο κώδικας.

Επισημαίνεται ότι στη βαθμολογία σας θα μετρήσει ΣΗΜΑΝΤΙΚΑ και το αν έχετε ελαχιστοποιήσει τη μεταδιδόμενη πληροφορία (αριθμό μηνυμάτων και αριθμό bytes σε κάθε μήνυμα), καθώς και το χρόνο που ακούει/λαμβάνει δεδομένα ένας κόμβος. Αυτό πρέπει να γίνει στον κώδικα σας. Δεν αρκεί να τρέχει το πρόγραμμά σας, αλλά και να τρέχει σωστά και βέλτιστα. Το να πείτε στην προφορική εξέταση του πρότζεκτ σας «Ε, ναι, θα μπορούσα να μην το κάνω αυτό» δε μετράει – θα πρέπει να έχετε κάνει τις βελτιστοποιήσεις σας στον κώδικα.

Σημείωση: Στο 2^ο παραδοτέο πρέπει να συμπεριλάβετε και το βοηθητικό πρόγραμμα του 1^{ου} παραδοτέου.