



ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS

## ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ

Project\_Phase1

ΑΝΑΣΤΑΣΙΟΣ ΜΠΕΝΟΣ 3130141

ΕΥΘΥΜΙΟΣ ΤΣΟΠΕΛΑΣ 3130210

(Οδηγίες για Compile στο τέλος της αναφοράς.)

### Αναφορά Παραδοτέου

#### Περιβάλλον δοκιμών:

Στα πλαίσια της εργασίας υλοποιήσαμε ένα εικονικό περιβάλλον που συντονίζει τον διαμοιρασμό ηλεκτρικής ενέργειας μεταξύ χρηστών. Οι χρήστες μπορούν τόσο να ζητούν όσο και να προσφέρουν μονάδες ενέργειας από και προς τους υπόλοιπους χρήστες της κοινότητας αντίστοιχα και όπως περιγράφεται παρακάτω. Στόχος της κοινότητας είναι να υπάρχει αδιάλειπτη προσφορά μονάδων ενέργειας σε όλους τους χρήστες που έχουν ανάγκη και η ανταλλαγή των μονάδων ενέργειας να εκτελείται με δίκαιο τρόπο. Ως προς την υλοποίηση, αναπτύξαμε το λογισμικό του Server (SocketServer) και των energy users/χρηστών (SocketClient01) της υπηρεσίας. Κατά την εκκίνηση του προγράμματος, ο Server είναι σε θέση να ανταποκρίνεται σε αιτήματα των energy users για την αποστολή μονάδων ηλεκτρικής ενέργειας. Για τον σκοπό αυτό ο Server γνωρίζει την ποσότητα ενέργειας που διαθέτει ο κάθε energy user του δικτύου. Στην πραγματικότητα μέσω της μεθόδου SignIn λαμβάνει αιτήματα σύνδεσης από τον εκάστοτε user με τα στοιχεία του.

Από την άλλη πλευρά ο client μόλις δημιουργηθεί τρέχει την μέθοδο register στην οποία καλείται ο χρήστης να συμπληρώσει τα στοιχεία του και στη συνέχεια με την introduce\_myself στέλνει για πρώτη φορά τα προσωπικά του στοιχεία στον server. Από την στιγμή αυτή και έπειτα η κεντρική μέθοδος που χρησιμοποιείται είναι η circle of life μέσα στην οποία ο client “ζει” και πραγματοποιεί ενέργειες όπως να καταναλώνει μονάδες ενέργειας, να ελέγχει το απόθεμα του, να εκκινεί διαδικασίες ανατροφοδότησης και εν ολίγοις πραγματοποιεί επικοινωνίες είτε με άλλους clients είτε με τον server κ.α. Θέλοντας να εξηγήσουμε λεπτομερώς αυτή τη θεμελιώδη μέθοδο του client αρκεί να περιγράψουμε το γενικότερο σενάριο του κύκλου ζωής ενός energy user:

## ΣΕΝΑΡΙΟ

Όταν κάποιος client , μέσω της κατανάλωσης ενέργειας που επιτελεί, χάσει το 20% ή το 50% ή το 90% του αρχικού του αποθέματος, κάθε φορά στέλνει στον master server ένα αίτημα με το ποσό ενέργειας που ζητά (10% του αρχικού αποθέματος). Ο server με την σειρά του εκκινεί μια διαδικασία αναζήτησης του client που φέρει το μεγαλύτερο απόθεμα (μέσω concurrent hash λίστας που έχει δημιουργήσει κατά το Registration). Αν το αίτημα του client είναι μικρότερο από το 10% του αποθέματος του υποψήφιου supplier τότε το αίτημα γίνεται δεκτό (1. Single Supplier, βλ. Υπόδειγμα 1). Αν ο server δεν μπορεί να βρει Single Supplier τότε απαντάει στον αιτών client ~> fail . Σε αυτό το σημείο ο client αποφασίζει να ακολουθήσουν το Plan A, κατά το οποίο ο server καλείται να ψάξει για ένα ζεύγος suppliers το οποίο θα είναι ικανό να ικανοποιήσει αθροιστικά το αρχικό αίτημα του client. Η λογική που ακολουθείται είναι η εξής, έστω R η αιτούμενη ποσότητα ενέργειας του client .

*An  $R/2 < 10\%$  του αποθέματος ενός random supplier*

*First Supplier Found*

*An  $R/2 < 10\%$  του αποθέματος ενός next supplier*

*Second Supplier Found*

Αν ο server μπορεί να βρεί ένα ζεύγος suppliers τότε αποστέλλει στον client τα ονόματα (username) τους.

Αν ο server δεν μπορεί να βρει ένα ζεύγος suppliers τότε αποστέλλει στον client fail, fail.

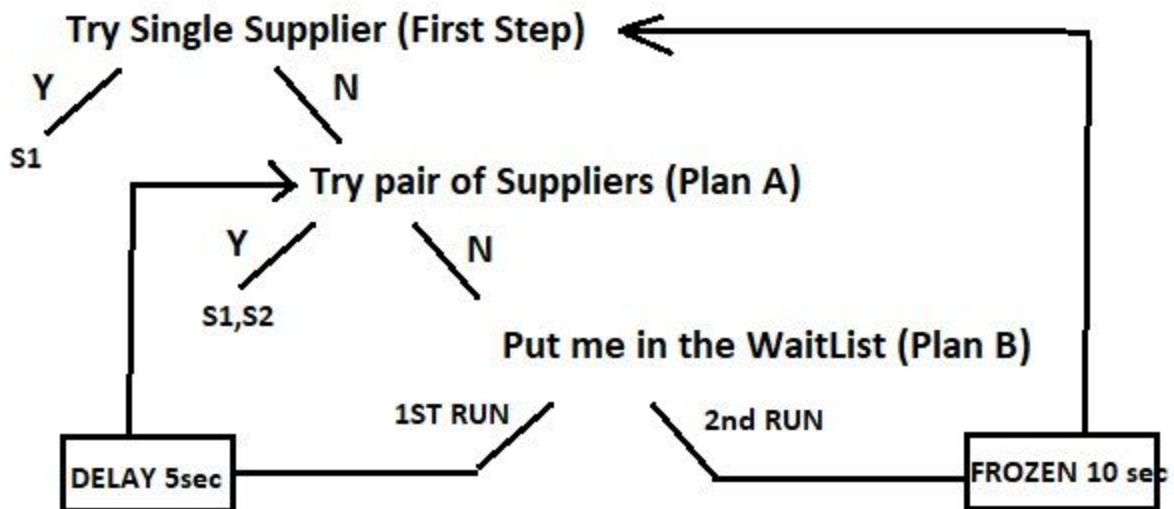
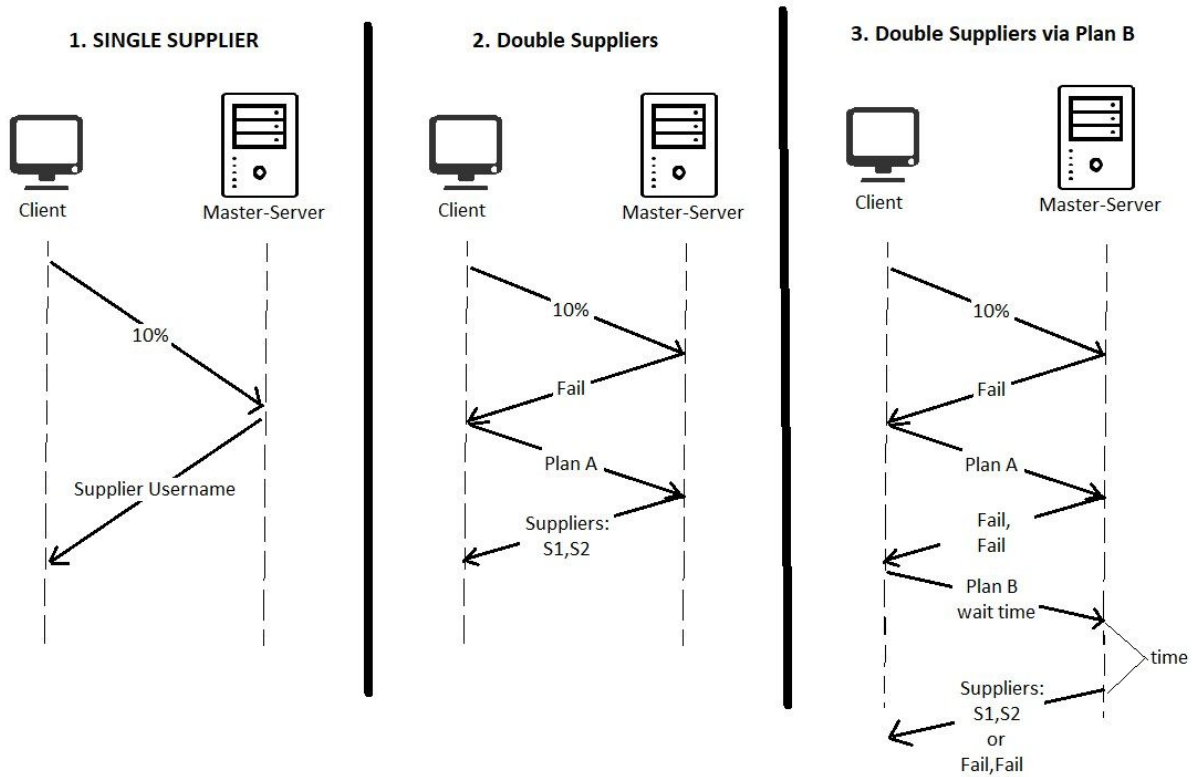
Αν ο client δεν λάβει τα 2 ονόματα τότε καταλήγουν στο plan B.

Πρόκειται για μία ιδιαίτερη περίπτωση κατά την οποία ο Client δηλώνει στον server ότι επιθυμεί το αίτημα ανατροφοδότησης του, να αποθηκευτεί και να επανεξεταστεί έπειτα από κάποιο συγκεκριμένο χρονικό διάστημα . Με το πέρας αυτού του διαστήματος ο Server εφαρμόζει για άλλη μία φορά το Plan A . Εν τέλει, είτε θα βρεί ένα ζεύγος διαθέσιμων προμηθευτών είτε θα αποστείλει τελική απόρριψη στον Client.

Αν ο Client λάβει μια τέτοια απόρριψη , χαρακτηρίζεται “σε κρίση” και σταματά την κατανάλωση ενέργειας για δέκα δευτερόλεπτα. Έπειτα είναι σε θέση να επαναλάβει εξ αρχής την όλη διαδικασία

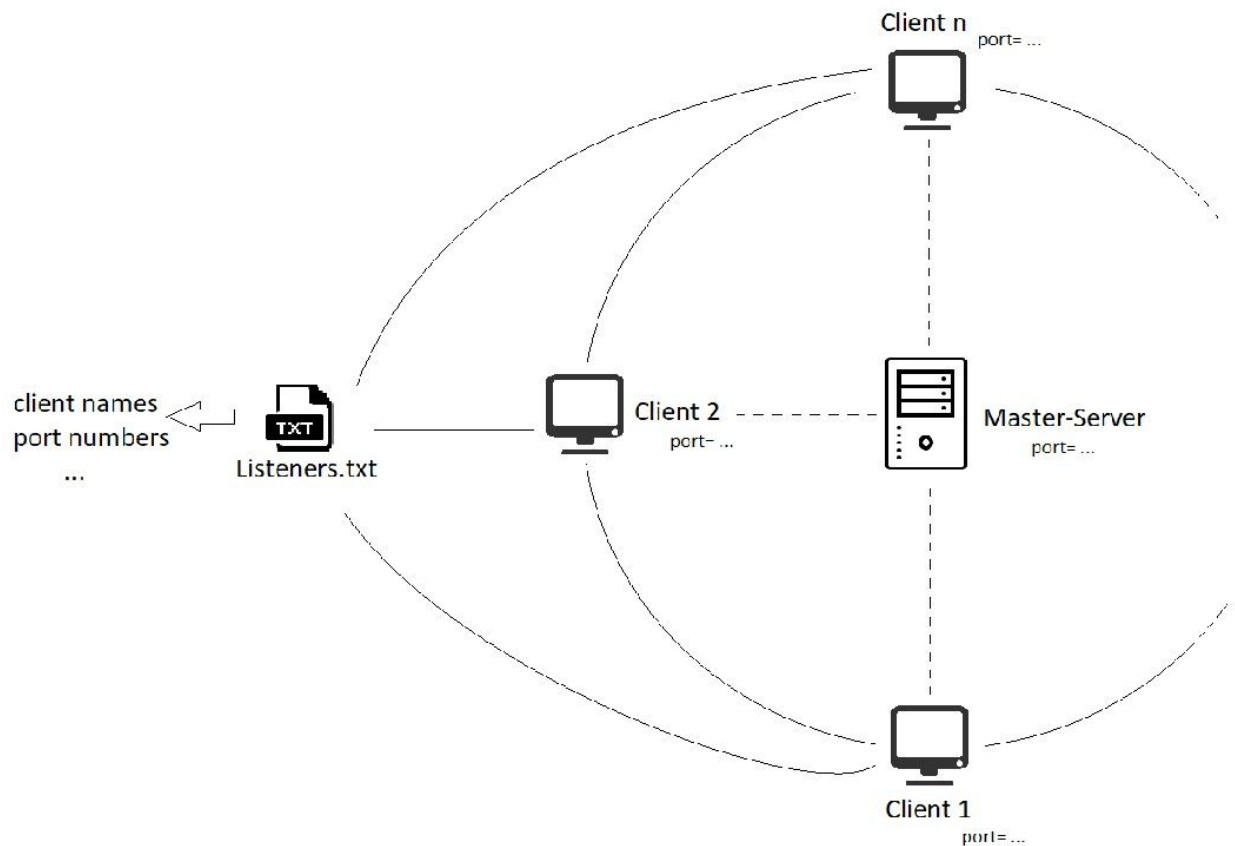
Σχηματικά έχουμε:

### Υπόδειγμα 1



Όταν ο εκάστοτε client έχει στην κατοχή του τα Username των/του Supplier τότε πραγματοποιεί αναζήτηση στο αρχείο Listeners.txt , με σκοπό να ενημερωθεί για την port στην οποία ακούει/ακούν ο supplier/suppliers. Να σημειωθεί ότι όταν ένας client εκκινείται καταγράφει σε αυτό το κοινόχρηστο αρχείο το Username και το port στο οποίο θα ακούει για πιθανά αιτήματα τροφοδοσίας απο άλλους clients. Εφόσον ενημερωθεί, προχωρά στην επίτευξη επικοινωνίας με τον/τους προμηθευτή/προμηθευτές του και εν συνεχεία στην λήψη των μονάδων. Αυτό μας κάνει σαφές ότι πρόκειται για μια peer-to-peer εφαρμογή κατα την οποία οι energy users δεν επικοινωνούν μόνο με τον master-server αλλά και μεταξύ τους (βλ. Υπόδειγμα 2).

Υπόδειγμα 2



## **Screenshots:**

### **Client-side:**

#### **Single-supplier**

```
Life: 74.0
Life: 73.0
Life: 72.0
Life: 71.0
Life: 70.0
Life: 69.0
[ system ] LIFE STATUS = 80%
[#code 69] Message send to Server : In need of Energy.
[#code 69] Message received from Server : Client-3 will be my Single Supplier.
I found Client-3 in Listeners file with port = 8940
[#code 36] Attempting to connect with Supplier Client-3 via port 8940
I found Client-3 in Listeners file with port = 8940
[#code 36] Socket Connection Established with my Supplier: Client-3
[#code 36] Successfully received 8.5 kwhrs from my Single Supplier. New Life: 76.5
[#code 10] Message send to Server : Updated Life.
[#code 10] Message send to Server : Updated Life.
Life: 76.5
Life: 75.5
Life: 74.5
```

#### **Double supplier**

```
Life: 14.0
Life: 13.0
Life: 12.0
[ system ] LIFE STATUS = 10%
[#code 69] Message send to Server : In need of Energy.
[#code 69] Message received from Server : failto find a Single Supplier.
[#code 69] Message send to Server : Plan A
    |_[#code 69][Plan A] Description: Request a Pair of Suppliers.
        [#code 69][Plan A] Message received from Server : Success.
        [#code 69][Plan A] Supplier 1: Client-1 , Supplier 2: Client-2
I found Client-1 in Listeners file with port = 5969
[#code 36] Socket Connection Established with my Supplier: Client-1
[#code 36] Successfully received 5.25 kwhrs from my Single Supplier. New Life: 16.25
[#code 10] Message send to Server : Updated Life.
[#code 10] Message send to Server : Updated Life.
I found Client-2 in Listeners file with port = 8042
[#code 36] Socket Connection Established with my Supplier: Client-2
[#code 36] Successfully received 5.25 kwhrs from my Single Supplier. New Life: 21.5
[#code 10] Message send to Server : Updated Life.
[#code 10] Message send to Server : Updated Life.
Life: 21.5
Life: 20.5
Life: 19.5
```

## Server-side:

### Single supplier

```
~Registration Completed~
key: Client-1 value: 85.0
key: Client-3 value: 105.0
key: Client-2 value: 90.0

[#code 69][Request] New Request: Client Client-1 in need of 8.5 kwhrs.
[#code 69][Process/] Searching for a Supplier.
[#code 69][ .... ] The Client with the most kwhrs is Client-3 => 105.0
[#code 69][Process] This User (Client-3) will become a Single Supplier.
[#code 69][Response] Single Supplier Found: Client-3. Information, send to requester client.
=====

[#code 10][Update/] Client Client-1 sent his new life.
[#code 10]***** UPDATED-INFO*****
[#code 10] key: Client-1 value: 76.5
[#code 10] key: Client-3 value: 105.0
[#code 10] key: Client-2 value: 90.0
[#code 10][Update] Successfull Update.
```

### Double supplier

```
[#code 10][Update/] Client Client-3 sent his new life.
[#code 10]***** UPDATED-INFO*****
[#code 10] key: Client-1 value: 67.0
[#code 10] key: Client-3 value: 49.5
[#code 10] key: Client-2 value: 81.5
[#code 10][Update] Successfull Update.

[#code 69][Request] New Request: Client Client-2 in need of 9.0 kwhrs.
[#code 69][Process/] Searching for a Supplier.
[#code 69][ .... ] The Client with the most kwhrs is Client-2 => 81.5
[#code 69][Process] This User (Client-2) is unable to become a Single Supplier.
[#code 69][Response] Unable to find a Single Supplier for this client. On hold for new instruction.
    |___ [#code 69][Request] Client has decided to try plan: A
        |___ [#code 69][Plan A] Description: Client requested a Pair of Suppliers.
            |___ [#code 69][Plan A][Process/] Searching for the two Suppliers.
                |___ [#code 69][Plan A][ .... ] Found 1st Contender Supplier: Client-1 donates 4.5 kwhrs to Client-2.
                    |___ [#code 69][Plan A][ .... ] Found 2nd Contender Supplier: Client-3 donates 4.5 kwhrs to Client-2.
                        |___ [#code 69][Plan A][Process] Pair of Suppliers successfully found.
                            |___ [#code 69][Response] Sending both Suppliers information to requester client.
```

## Οδηγίες Compile:

Αλλαγή της IP στις γραμμες:

SocketClient01: 22, 145, 154, 181,190, 216

- SocketClient01 ~~>Allow Parallel Run
- Run: SocketServer
- Run: SocketClient (1st Time: for 1st Energy User)
- Run: SocketClient (2nd Time: for 2nd Energy User)
- ...
- Run: SocketClient (n Time: for n Energy User)