



## **ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ  
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ  
[www.cslab.ece.ntua.gr](http://www.cslab.ece.ntua.gr)

### **ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ**

**9ο εξάμηνο ΗΜΜΥ, ακαδημαϊκό έτος 2017-18**

### **ΑΝΑΦΟΡΑ ΕΞΑΜΗΝΙΑΙΑΣ ΕΡΓΑΣΙΑΣ**

**ΟΜΑΔΑ: Distrib 17**

Γκούσκος Παναγιώτης A.M: 03113551 Email: [panagiotis.gou@hotmail.com](mailto:panagiotis.gou@hotmail.com)  
Δράγαζης Νικόλαος A.M: 03113162 Email: [ndragazis@outlook.com.gr](mailto:ndragazis@outlook.com.gr)  
Πέτρου Γεώργιος A.M: 03113145 Email: [el13145@central.ntua.gr](mailto:el13145@central.ntua.gr)

## 1. Σχεδιασμός Συστήματος

Το project έχει υλοποιηθεί σε γλώσσα προγραμματισμού Java και περιέχει τις κλάσεις Tracker, client, Group, Four, Message, Total\_Message και CustomComparator. Στη συνέχεια παρουσιάζονται λεπτομέρειες σχεδιασμού και υλοποίησης για τις βασικές κλάσεις και βασική περιγραφή των λειτουργιών τους:

- **Tracker:** Για το σχεδιασμό και υλοποίηση του tracker, αρχικά στο main thread του προγράμματος, ακούμε για νέες TCP συνδέσεις. Για κάθε TCP σύνδεση ξεκινάει ένα νέο thread που διαχειρίζεται τη συγκεκριμένη σύνδεση-socket. Οπότε, όταν ένας client συνδέεται μέσω TCP με τον tracker, ο tracker αναθέτει τη διαχείριση της σύνδεσης σε ένα νέο thread, το οποίο λαμβάνει ως είσοδο από τον client την εντολή που έχει στείλει (πχ. register) και πραγματοποιεί την κατάλληλη ενέργεια (πχ. καταγραφή των στοιχείων του client (σε κοινή δομή δεδομένων του tracker) κατά την εγγραφή του). Για τον συγχρονισμό των threads που διαχειρίζονται διαφορετικές συνδέσεις και πραγματοποιούν πρόσβαση σε κοινές δομές δεδομένων, κυρίως έχουμε χρησιμοποιήσει δομές δεδομένων όπως concurrent Hashmap, που επιτρέπουν παράλληλη ανάγνωση και συγχρονισμό κατά τις εγγραφές, καθώς επίσης, όπου δεν ήταν εφικτή η χρήση concurrent δομής δεδομένων, χρησιμοποιούμε lock και synchronize. Τέλος, ως προς τη διαχείριση των heartbeats από τον tracker, κάθε φορά που ένας client κάνει register, κρατάμε έναν timer για το συγκεκριμένο χρήστη. Αν λάβουμε heartbeat message εντός ενός χρονικού ορίου ακυρώνουμε τον συγκεκριμένο timer και τον επανεκκινούμε. Αν δεν λάβουμε heartbeat message στο αναμενόμενο χρονικό όριο ο timer εκτελεί μια συνάρτηση όπου στέλνει UDP μηνύματα στους υπόλοιπους clients (που συμμετέχουν τουλάχιστον σε ένα group με τον χρήστη που έχει σφάλμα) προκειμένου να τους ενημερώσει.
- **Client:** Ο client χρησιμοποιεί δύο threads. Το main thread αρχικοποιεί ένα UDP socket και ξεκινάει το δεύτερο thread, το οποίο μόνο ακούει για εισερχόμενα UDP μηνύματα. Στη συνέχεια, το main thread συνδέεται (μέσω TCP) με τον tracker για να κάνει register, και στη συνέχεια, αφού έχει γίνει επιτυχώς registered, ξεκινάει ένα timer που ανά κάποια sec στέλνει heartbeat message στον tracker. Αφού αρχικοποιήσει τον timer, ξεκινάει να διαβάζει τα μηνύματα εισόδου από το χρήστη, δηλαδή υλοποιεί το user interface. Το δεύτερο thread ακούει για εισερχόμενα UDP μηνύματα, οπότε αν εγγραφεί/διαγραφεί νέος χρήστης, ενημερώνεται για να κρατήσει/διαγράψει τα στοιχεία του, όπως επίσης αν έρθει ένα νέο μήνυμα από κάποιο χρήστη, το κάνει receive και το αποθηκεύει σε κατάλληλη ουρά για την υλοποίηση των πρωτοκόλλων (Fifo , Total Ordering + Fifo). Πάλι ο συγχρονισμός επιτυγχάνεται χρησιμοποιώντας concurrent Hashmap , PriorityBlockingQueue και

locks όπου είναι απαραίτητο. Τέλος, να αναφέρουμε ότι το πρωτόκολλο με το οποίο γίνεται το ordering των multicast μηνυμάτων (Fifo , Total Ordering + Fifo) δίνεται ως command line argument (Fifo για Fifo και Total για Total Ordering + Fifo), όπως επίσης και το port number που ακούει για UDP μηνύματα ο Client.

- **Group:** Μια απλή κλάση που αναπαριστά ένα group. Κρατάει τις κατάλληλες πληροφορίες για τους χρήστες του group και παρέχει κατάλληλες μεθόδους για την τροποποίηση των πληροφοριών αυτών (πχ set\_new\_member). Η κλάση αυτή χρησιμοποιείται τόσο από τον client όσο και από τον tracker.
- **Four:** Η κλάση αυτή χρησιμοποιείται για την αναπαράσταση τετράδων και χρησιμοποιείται για να κρατάμε σαν τετράδες id, UserName, ip, port τις πληροφορίες για τους χρήστες (πχ ο tracker κρατάει όλους τους χρήστες σε ένα πίνακα τύπου four, δηλαδή ένα πίνακα τετράδων που κρατάει τις τέσσερις παραπάνω πληροφορίες για κάθε χρήστη).
- **Message:** Η κλάση αυτή χρησιμοποιείται μόνο από τον client και μόνο για την αναπαράσταση μηνυμάτων όταν χρησιμοποιούμε το πρωτόκολλο Fifo.
- **Total Message:** Η κλάση αυτή χρησιμοποιείται για την αναπαράσταση μηνυμάτων όταν χρησιμοποιούμε Total Order + Fifo διάταξη. Η κλάση Total\_Message κληρονομεί την κλάση Message και διαθέτει ορισμένα πεδία, καθώς οι πληροφορίες που πρέπει να κρατάμε σε σχέση με το Fifo πρωτόκολλο είναι περισσότερες.
- **CustomComparator:** Η κλάση αυτή χρησιμοποιείται στη priority queue όταν χρησιμοποιούμε Total Order + Fifo διάταξη, καθώς παρέχει μεθόδους σύγκρισης μηνυμάτων, ώστε να οριστεί η διάταξη στην priority queue.

Οι αλγόριθμοι που χρησιμοποιήσαμε για την υλοποίηση του ordering των μηνυμάτων είναι:

- Για Fifo Ordering χρησιμοποιήσαμε τον αλγόριθμο των διαφανειών χωρίς κάποια αλλαγή, δηλαδή κάθε χρήστης απλά κρατάει ένα vector για κάθε group που συμμετέχει, μεγέθους όσες είναι και οι διεργασίες που συμμετέχουν στο group. Κάθε φορά που λαμβάνει ένα μήνυμα, αναλόγως τον αύξοντα αριθμό του μηνύματος, είτε το αποθηκεύει είτε το παραδίδει.

- Για Fifo + Total Ordering χρησιμοποιήσαμε μια παραλλαγή του αλγορίθμου ISIS. Ο αλγόριθμος ISIS εγγυάται Total Ordering αλλά όχι απαραίτητα και Fifo Ordering, οπότε η αλλαγή που πραγματοποιήσαμε ήταν ότι, όπως και στο Fifo ordering, κρατάγαμε πάλι ένα vector και κάθε φορά που ήταν έτοιμο για παράδοση ένα μήνυμα στη holdback queue, σύμφωνα με τον αλγόριθμο ISIS, απλά αντί να το παραδίδουμε, κοιτάγαμε αν έχει και το σωστό αύξοντα αριθμό, αλλιώς ψάχναμε στη holdback queue για το σωστό ως προς τον αύξοντα αριθμό μήνυμα, και αν αυτό ήταν έτοιμο για παράδοση το παραδίδαμε πρώτα, αλλιώς δεν παραδίδαμε κανένα μέχρι να γίνει έτοιμο προς παράδοση το μήνυμα με το σωστό αύξοντα αριθμό.

## **2. Τρόπος Εκτέλεσης**

Για να εκκινήσουμε τον Tracker εκτελούμε:

**java Distrib.Tracker**

(το πρόθεμα Distrib. Χρειάζεται αφού τα έχουμε ορίσει σε ένα πακέτο με το όνομα Distrib). Επιπλέον, ο Tracker ακούει για TCP συνδέσεις στην πόρτα 9002, οπότε δεν χρειάζεται να προσδιορίσουμε την πόρτα σαν command line argument.

Για τον κάθε Client:

**java Distrib.Client Port OrderingOption**

όπου:

- Port ο αριθμός της πόρτας που ακούει για UDP μηνύματα ο κάθε Client
- OrderingOption μπορεί να είναι είτε Fifo (για Fifo Ordering) είτε Total (για Total + Fifo Ordering).

## **3. Αποτελέσματα Μετρήσεων**

### **1. Απόδοση του συστήματος**

Για την πραγματοποίηση των μετρήσεων σχετικά με την απόδοση του συστήματος, στήσαμε ένα messenger με 5 clients στην υποδομή του εργαστηρίου και λάβαμε τις σχετικές μετρήσεις του χρόνου. Για να τρέξουμε τους clients παράλληλα, ορίσαμε άλλη μία παράμετρο στα command line arguments, που αντιστοιχεί στη χρονική στιγμή που θα αρχίσουν οι clients να στέλνουν τα μηνύματα. Οι clients, αφού εγγράφονται σε ένα γκρουπ, εκτελούν ενεργό αναμονή και διαβάζουν τη τρέχουσα ώρα από βιβλιοθήκη της java ώστε όταν φτάσει η χρονική στιγμή (που περνάμε σαν

argument) με ακρίβεια nanosec να αρχίσουν να στέλνουν. Για παράδειγμα, εκτελώντας **java Distrib.Client 9002 Fifo 15** σημαίνει ότι τη στιγμή 15 λεπτά (της τρέχουσας ώρας) και 0 msec και 0 nanosec ξεκινάει να στέλνει μηνύματα. Με αυτό το τρόπο καταφέραμε όλοι οι clients να συγχρονιστούν με ακρίβεια nanosec και να στείλουν την ίδια ώρα τα μηνύματα για να έχουμε ταυτόχρονη αποστολή των μηνυμάτων.

## **I. Fifo:**

- Throughput:
  - Client1: 50 messages /15632228 nsec = 3.199 messages/msec
  - Client2: 50 messages /23449697 nsec = 2.132 messages/msec
  - Client3: 50 messages /16428067 nsec = 3.044 messages/msec
  - Client4: 50 messages /28678741 nsec = 1.743 messages/msec
  - Client5: 50 messages /16439781 nsec = 3.041 messages/msec
- Συνολικό Throughput: 13.159 messages/msec
- Latency: 0.076 msec/message
- Το Κόστος σε συνολικό αριθμό μηνυμάτων που απαιτούνται για τη σωστή λειτουργία του συστήματος είναι όσα μηνύματα απαιτούνται για το πρωτόκολλο Fifo, δηλαδή N μηνύματα για κάθε γραμμή που στέλνει ένας Client, όπου N ο αριθμός των Clients. Στις παραπάνω μετρήσεις, κάθε client έστειλε 10 γραμμές/μηνύματα και N=5, οπότε κάθε Client στέλνει συνολικά 50 μηνύματα, άρα συνολικά όλοι οι Clients  $5 \times 50 = 250$  μηνύματα.

## **II. Total:**

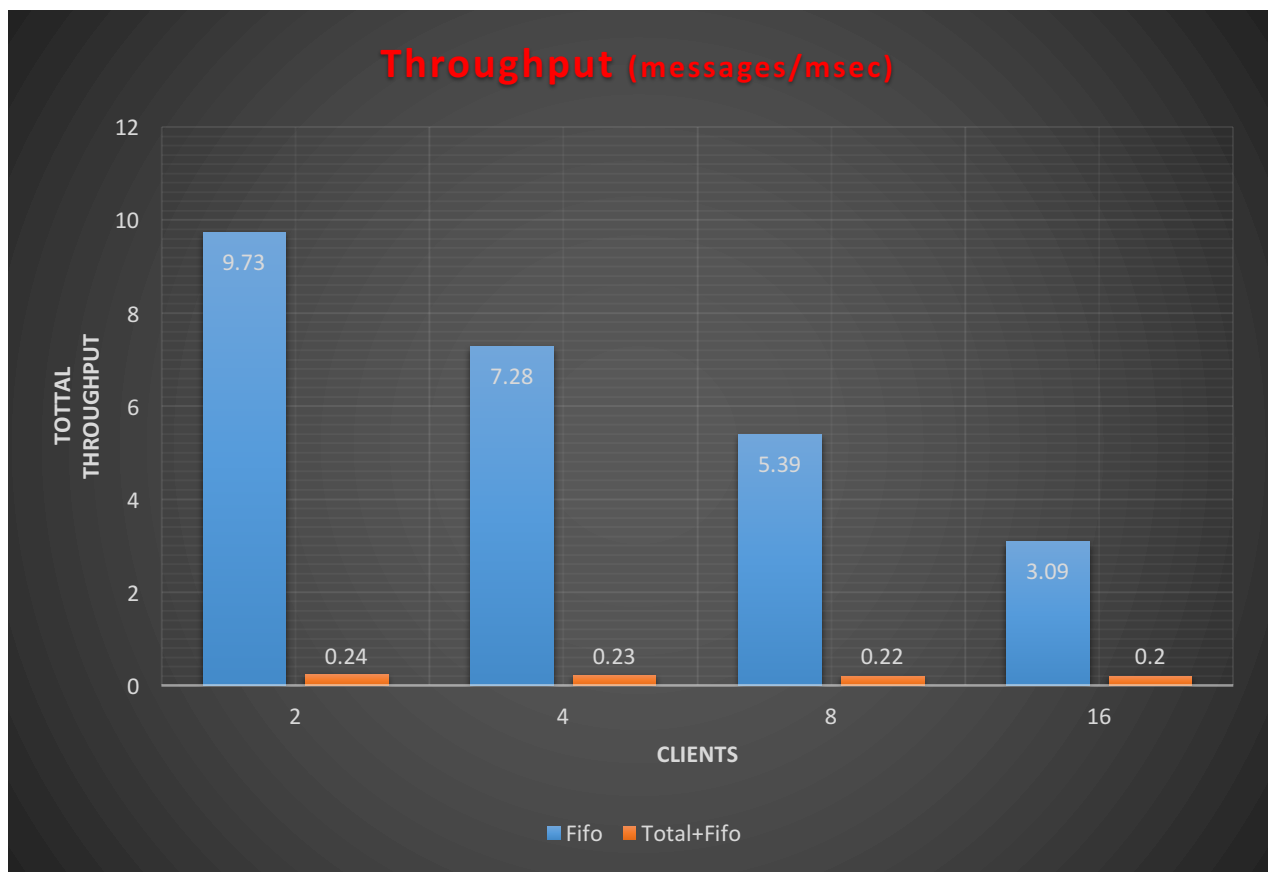
- Throughput:
  - Client1: 50 messages /216820486 nsec = 0.231 messages/msec
  - Client2: 50 messages /216658700 nsec = 0.231 messages/msec
  - Client3: 50 messages /216813857 nsec = 0.231 messages/msec
  - Client4: 50 messages /217292483 nsec = 0.230 messages/msec
  - Client5: 50 messages /217146899 nsec = 0.230 messages/msec
- Συνολικό Throughput: 1.153 messages/msec
- Latency: 0.8673 msec/message
- Το Κόστος σε συνολικό αριθμό μηνυμάτων που απαιτούνται για τη σωστή λειτουργία του συστήματος είναι όσα μηνύματα απαιτούνται για το πρωτόκολλο Total Order(ISIS algorithm), δηλαδή  $3 \times N$  μηνύματα για κάθε γραμμή που στέλνει ένας Client, όπου N ο αριθμός των Clients. Στις παραπάνω μετρήσεις, κάθε client

έστειλε 10 γραμμές/μηνύματα και  $N=5$ , οπότε κάθε Client στέλνει συνολικά 150 μηνύματα, άρα συνολικά όλοι οι Clients  $5 \cdot 150 = 750$  μηνύματα.

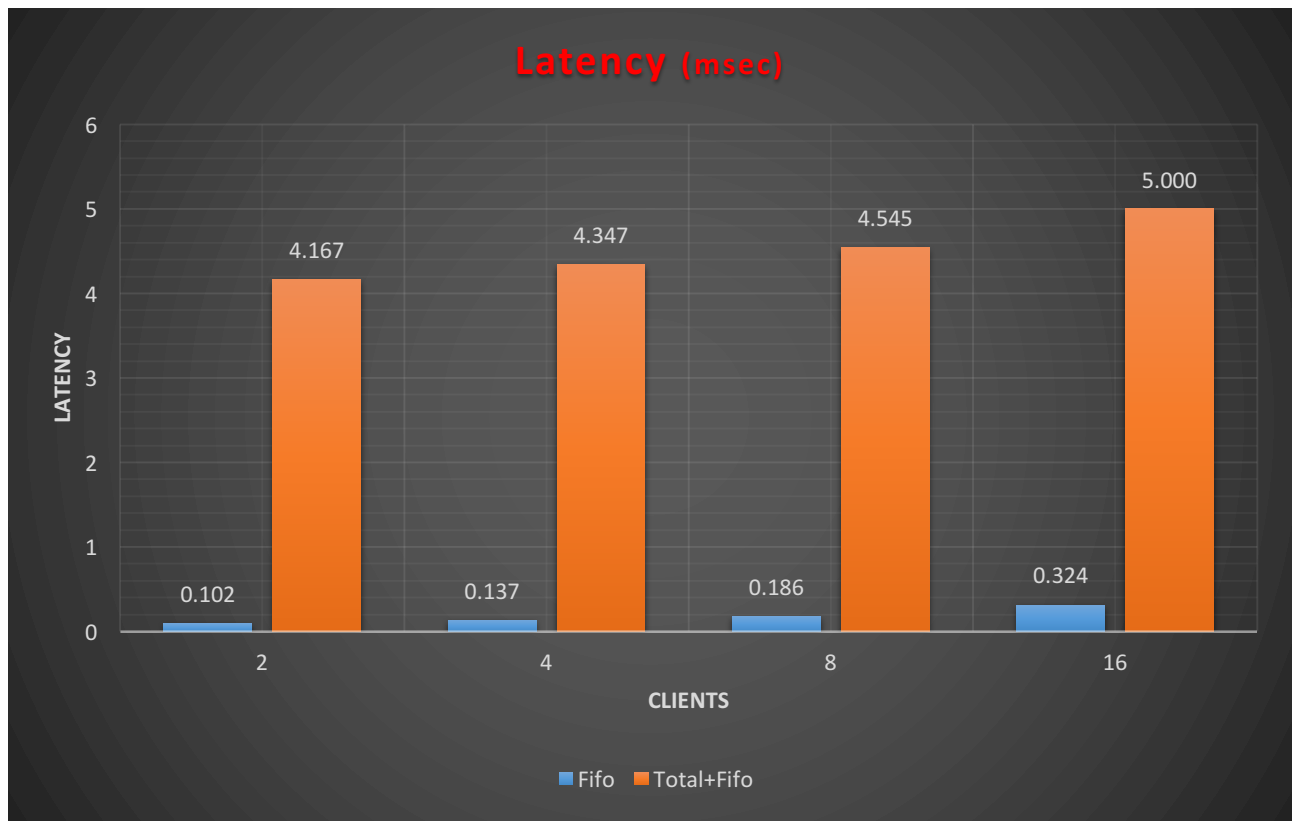
## **2. Κλιμακωσιμότητα του συστήματος**

Στη συνέχεια στήνουμε έναν messenger με 2, 4, 8 και 16 clients. Αφού όλοι εγγραφούν στο σύστημα και εισέλθουν στο ίδιο group, ο Client1 διαβάζει το αρχείο many\_messages.txt και θα στέλνει ένα μήνυμα ανά γραμμή στο group. Παρακάτω παρουσιάζουμε σε γράφημα τις μετρικές του προηγούμενου πειράματος (άξονας y) σε σχέση με τον αριθμό των clients (άξονας x) για Fifo και Fifo+total ordering.

- **Throughput (messages/msec):**



- **Latency(msec):**



#### 4. Παραδείγματα Εκτέλεσης

Στη συνέχεια παρουσιάζονται στιγμιότυπα εκτέλεσης για δύο clients:

- Για Fifo Ordering:

##### Client 1:

```
***** - Client Application - *****
## Please Insert a Valid User Name ##
## Registering User ##
1
## User Registered !! ##
[A] > 1 A 9004 127.0.0.1
in Group: D User B send Message 1b 0 Deep Thought computerA] >
in Group: D User B send Message 2b We want you to tell us...The Answer
in Group: D User B send Message 3b The Answer to what?
in Group: D User B send Message 4b Life!
in Group: D User B send Message 5b The Universe!
in Group: D User B send Message 6b Everything!
in Group: D User A send Message 1a Don't panic
in Group: D User A send Message 2a Forty-two, said Deep Thought, with infinite majesty and calm.
in Group: D User A send Message 3a For a moment, nothing happened. Then, after a second or so, nothing continued to happen.
in Group: D User B send Message 7b Tricky
in Group: D User A send Message 4a It is a mistake to think you can solve any major problems just with potatoes.
in Group: D User B send Message 8b But can you do it?
in Group: D User A send Message 5a Nothing travels faster than the speed of light with the possible exception of bad news
in Group: D User B send Message 9b Yes
in Group: D User A send Message 6a In the beginning the Universe was created.
in Group: D User B send Message 10b There is an answer?
in Group: D User A send Message 7a This has made a lot of people very angry.
in Group: D User A send Message 8a Time is an illusion.
in Group: D User A send Message 9a Lunchtime doubly so.
in Group: D User A send Message 10a I'd far rather be happy than right any day.
```

##### Client 2:

```

***** - Client Application - *****

## Pleas Insert a Valid User Name ##
## Registering User ##
2
## User Registered !! ##
[B] > 1 A 9004 127.0.0.1 2 B 9005 127.0.0.1
in Group: D User B send Message 1b 0 Deep Thought computerB] >
in Group: D User B send Message 2b We want you to tell us...The Answer
in Group: D User B send Message 3b The Answer to what?
in Group: D User B send Message 4b Life!
in Group: D User B send Message 5b The Universe!
in Group: D User B send Message 6b Everything!
in Group: D User A send Message 1a Don't panic
in Group: D User A send Message 2a Forty-two, said Deep Thought, with infinite majesty and calm.
in Group: D User A send Message 3a For a moment, nothing happened. Then, after a second or so, nothing continued to happen.
in Group: D User B send Message 7b Tricky
in Group: D User A send Message 4a It is a mistake to think you can solve any major problems just with potatoes.
in Group: D User B send Message 8b But can you do it?
in Group: D User A send Message 5a Nothing travels faster than the speed of light with the possible exception of bad news
in Group: D User B send Message 9b Yes
in Group: D User B send Message 10b There is an answer?
in Group: D User A send Message 6a In the beginning the Universe was created.
in Group: D User A send Message 7a This has made a lot of people very angry.
in Group: D User A send Message 8a Time is an illusion.
in Group: D User A send Message 9a Lunchtime doubly so.
in Group: D User A send Message 10a I'd far rather be happy than right any day.

```

Όπως παρατηρούμε, τα μηνύματα ακολουθούν Fifo Ordering αλλά όχι Total Ordering (πχ. τα μηνύματα 10b, 6a έρχονται με αντίστροφη σειρά για κάθε Client).

- Για Total + Fifo Ordering:

#### Client 1:

```

***** - Client Application - *****

## Pleas Insert a Valid User Name ##
## Registering User ##
2
## User Registered !! ##
[B] > 1 A 9004 127.0.0.1 2 B 9005 127.0.0.1
in Group: D User B send Message 1b 0 Deep Thought computer
in Group: D User A send Message 1a Don't panic
in Group: D User B send Message 2b We want you to tell us...The Answer
in Group: D User A send Message 2a Forty-two, said Deep Thought, with infinite majesty and calm.
in Group: D User A send Message 3a For a moment, nothing happened. Then, after a second or so, nothing continued to happen.
in Group: D User B send Message 3b The Answer to what?
in Group: D User B send Message 4b Life!
in Group: D User A send Message 4a It is a mistake to think you can solve any major problems just with potatoes.
in Group: D User A send Message 5a Nothing travels faster than the speed of light with the possible exception of bad news
in Group: D User B send Message 5b The Universe!
in Group: D User A send Message 6a In the beginning the Universe was created.
in Group: D User B send Message 6b Everything!
in Group: D User A send Message 7a This has made a lot of people very angry.
in Group: D User B send Message 7b Tricky
in Group: D User A send Message 8a Time is an illusion.
in Group: D User B send Message 8b But can you do it?
in Group: D User B send Message 9b Yes
in Group: D User A send Message 9a Lunchtime doubly so.
in Group: D User A send Message 10a I'd far rather be happy than right any day.
in Group: D User B send Message 10b There is an answer?

```

#### Client 2:



```

***** - Client Application - *****

## Pleas Insert a Valid User Name ##
## Registering User ##
1
## User Registered !! ##
[A] > 1 A 9004 127.0.0.1
in Group: D User B send Message 1b 0 Deep Thought computer
in Group: D User A send Message 1a Don't panic
in Group: D User B send Message 2b We want you to tell us...The Answer
in Group: D User A send Message 2a Forty-two, said Deep Thought, with infinite majesty and calm.
in Group: D User A send Message 3a For a moment, nothing happened. Then, after a second or so, nothing continued to happen.
in Group: D User B send Message 3b The Answer to what?
in Group: D User B send Message 4b Life!
in Group: D User A send Message 4a It is a mistake to think you can solve any major problems just with potatoes.
in Group: D User A send Message 5a Nothing travels faster than the speed of light with the possible exception of bad news
in Group: D User B send Message 5b The Universe!
in Group: D User A send Message 6a In the beginning the Universe was created.
in Group: D User B send Message 6b Everything!
in Group: D User A send Message 7a This has made a lot of people very angry.
in Group: D User B send Message 7b Tricky
in Group: D User A send Message 8a Time is an illusion.
in Group: D User B send Message 8b But can you do it?
in Group: D User B send Message 9b Yes
in Group: D User A send Message 9a Lunchtime doubly so.
in Group: D User A send Message 10a I'd far rather be happy than right any day.
in Group: D User B send Message 10b There is an answer?

```

Όπως παρατηρούμε, τα μηνύματα φτάνουν στους Client1,2 με Total και Fifo Ordering.