

IRC Share

Πρόγραμμα ανταλλαγής αρχείων χωρίς την ύπαρξη εξυπηρετητών



**ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΚΡΗΤΗΣ**

Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων

Πτυχιακή εργασία από:

Παπαδόπουλος Φώτης

Οκτώβριος 2006 – Απρίλιος 2007

Υπό την επίβλεψη των καθηγητών:

Γεώργιος Παπαδουρακης

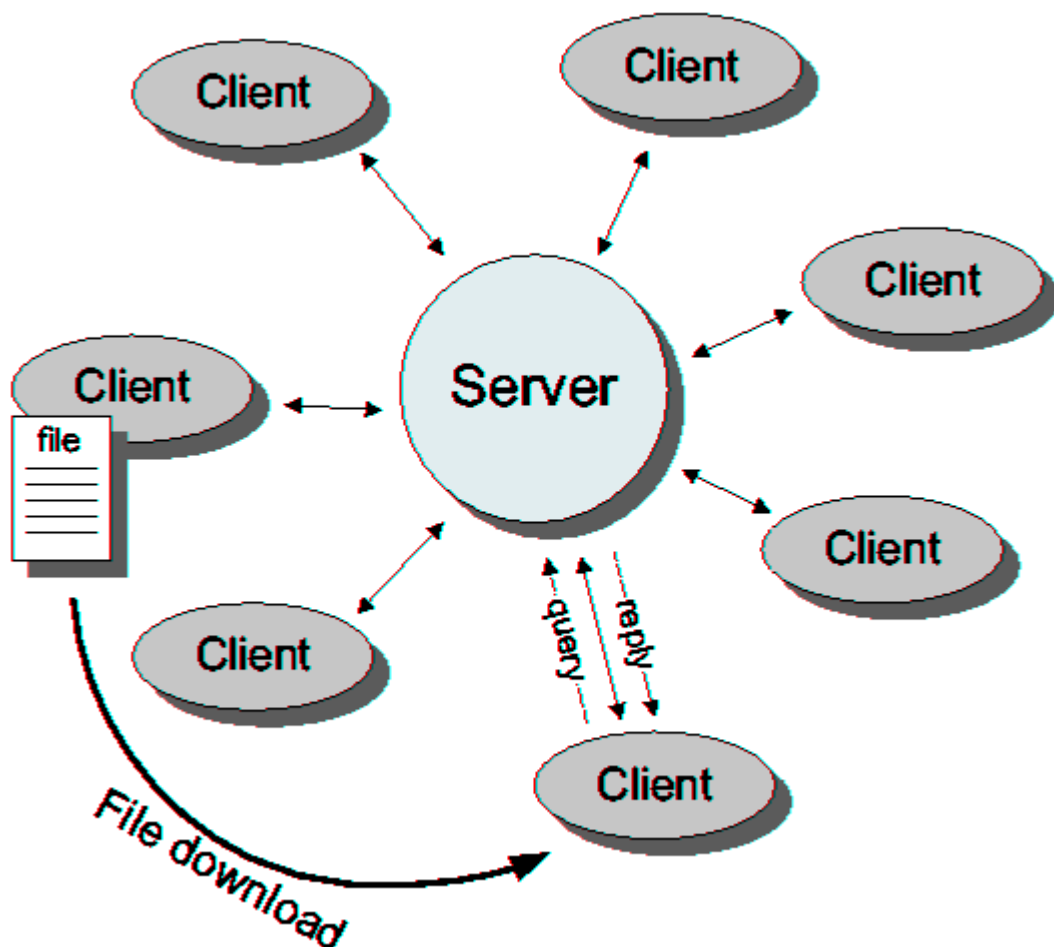
Andrew Ware

Περιεχόμενα

<i>Περιεχόμενα.....</i>	<i>2</i>
<i>Κεφάλαιο 1: Εισαγωγή.....</i>	<i>3</i>
<i>Κεφάλαιο 2: Σκοπός της πτοχιακής.....</i>	<i>6</i>
<i>Κεφάλαιο 3: Ανάλυση απαιτήσεων.....</i>	<i>7</i>
3.1: <i>Κατηγορίες χρηστών.....</i>	<i>7</i>
3.2: <i>Περιβάλλον ανάπτυξης.....</i>	<i>7</i>
3.3: <i>Κύρια τμήματα του συστήματος.....</i>	<i>7</i>
3.4: <i>Βασικές λειτουργίες.....</i>	<i>10</i>
3.4.1: <i>Σύνδεση σε δίκτυο IRC.....</i>	<i>10</i>
3.4.2: <i>Δημιουργία αξιόπιστου δικτύου – Ανάκτηση συνδέσεων.....</i>	<i>11</i>
3.4.3: <i>Δυνατότητα γρήγορης αναζήτησης.....</i>	<i>12</i>
3.4.4: <i>Μεταφορά αρχείων.....</i>	<i>14</i>
<i>Κεφάλαιο 4: Σχεδιασμός προγράμματος.....</i>	<i>16</i>
4.1: <i>Σχεδιασμός γραφικού περιβάλλοντος διεπαφής χρήστη.....</i>	<i>16</i>
4.2: <i>Σύνδεση σε δίκτυο IRC.....</i>	<i>21</i>
4.3: <i>Δίκτυο και αποκατάσταση συνδέσεων.....</i>	<i>24</i>
4.4: <i>Αναζήτηση.....</i>	<i>28</i>
4.5: <i>Μεταφορά αρχείων.....</i>	<i>31</i>
<i>Κεφάλαιο 5: Υλοποίηση συστήματος.....</i>	<i>36</i>
5.1: <i>Γενικά για την υλοποίηση.....</i>	<i>36</i>
5.2: <i>Εκκίνηση προγράμματος.....</i>	<i>36</i>
5.3: <i>Σύνδεση με IRC Server.....</i>	<i>39</i>
5.4: <i>Δημιουργία συνδέσεων.....</i>	<i>42</i>
5.5: <i>Ανάκτηση συνδέσεων</i>	<i>57</i>
5.6: <i>Αναζήτηση.....</i>	<i>63</i>
5.7: <i>Μεταφορά αρχείων.....</i>	<i>69</i>
<i>Κεφάλαιο 6: Μελλοντικές προσθήκες.....</i>	<i>76</i>
<i>Κεφάλαιο 7: Συμπεράσματα.....</i>	<i>80</i>
<i>Βιβλιογραφία.....</i>	<i>80</i>

Κεφάλαιο 1: Εισαγωγή

Η ανταλλαγή αρχείων έγινε γνωστή με το που εμφανίστηκε το Πρόγραμμα Napster το 1999. Τότε έμαθε ο κόσμος πόσο εύκολα μπορεί να ανταλλάξει οποιοδήποτε είδος αρχείου με τους υπόλοιπους χρήστες. Το δίκτυο αυτό στηριζόταν σε εξυπηρετητές οι οποίοι αποθήκευαν τα δεδομένα που αντάλλαζαν οι χρήστες. Το δίκτυο στην ακμή της λειτουργίας του μετρούσε περισσότερους από 30 εκατομμύρια χρήστες. Όταν άρχισε να γίνεται διάσημο μερικά συγκροτήματα κατέθεσαν μήνυση εναντίον του προγραμματιστή του που απαιτούσαν να απομακρυνθούν τα τραγούδια τους. Αυτό όμως δεν μπορούσε να γίνει και το Πρόγραμμα έγινε εμπορικό κάτω από μια εταιρία και μεταφέρονται αρχεία επί πληρωμή. Το Napster όμως έχει προωθήσει και πάρα πολλά συγκροτήματα τα οποία ήταν άγνωστα και τα βοήθησε.



Πελάτες συνδεδεμένοι σε δίκτυο Peer to Peer μέσω server

Στο ιντερνέτ αυτή τη στιγμή υπάρχουν πολλά προγράμματα και τύποι δικτύων που κάνουν αυτή τη δουλειά αλλά ο τρόπος που εφαρμόζεται σε αυτή την εργασία είναι ξεχωριστός. Είναι ξεχωριστός γιατί κάθε χρήστης όταν θέλει να συνδεθεί στο δίκτυο πρώτα μπαίνει σε ένα IRC δίκτυο και από εκεί βλέπει τον τελευταίο χρήστη όπου γίνεται η σύνδεση. Θα αναλυθεί παρακάτω λεπτομερώς ο τρόπος με τον οποίο λειτουργεί το Πρόγραμμα.

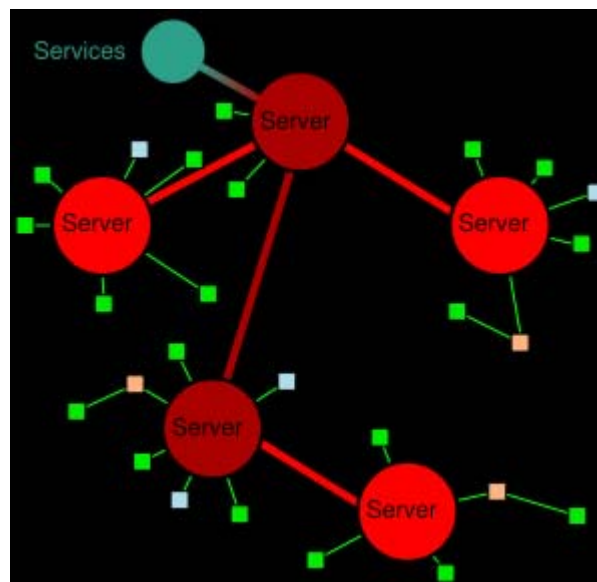
Το πιο γνωστό Πρόγραμμα είναι το eMule που βασίζεται στο δίκτυο eDonkey2000. Οι χρήστες πρώτα συνδέονται σε κάποιον ειδικό διακομιστή και μετά από εκεί κάνουν τις υπόλοιπες λειτουργίες. Το eMule υποστηρίζει ακόμα έναν τύπο δικτύου που είναι τελείως ανεξάρτητος από κάποιο είδος διακομιστή το οποίο λέγεται Kademia. Τώρα κάθε χρήστης είναι και διακομιστής αλλά για να συνδεθούν οι χρήστες μεταξύ τους πρέπει ο χρήστης που θέλει να συνδεθεί να ξέρει την τελευταία IP του χρήστη που συνδέθηκε τελευταίος στο δίκτυο.

Ένας άλλος πολύ δημοφιλής τρόπος είναι τα torrent. Τα torrent λειτουργούν ως εξής: Κάποιος χρήστης θέλει να κατεβάσει ένα Πρόγραμμα πχ το winamp. Υπάρχουν πολλές σελίδες μηχανές αναζήτησης μόνο για torrent. Το torrent το κατεβάζει ο χρήστης το οποίο είναι πολύ μικρό και έχει μέσα μόνο πληροφορίες για το διακομιστή που είναι υπεύθυνος για το αρχείο και ονόματα αρχείων. Ο χρήστης ανοίγει το torrent με ειδικό Πρόγραμμα(bitlord,azareus,...) και συνδέεται στον διακομιστή ο οποίος του δίνει πληροφορίες για το ποιους χρήστες έχουν το ίδιο αρχείο και το μοιράζονται. Έτσι ο χρήστης κατεβάζει το Πρόγραμμα από τους υπόλοιπους χρήστες αλλά ταυτόχρονα μοιράζει και αυτός ότι έχει κατεβάσει. Τα torrent είναι πολύ διαδεδομένα Αυτό το διάστημα. Το κύριο μειονέκτημα του είναι ότι χρειάζεται τους διακομιστές για να λειτουργήσει.

Υπάρχουν ακόμα πολλά προγράμματα στο δίκτυο αλλά αυτά είναι τα πιο γνωστά. Κάποια από τα υπόλοιπα είναι: Kazaa, LimeWire, Morpheus, WinMX

Το πρόγραμμα που παρουσιάζεται εδώ χρησιμοποιεί δίκτυα IRC για να δημιουργήσει το δίκτυο. Τα δίκτυα IRC πρωτοεμφανίστηκαν από το 1988 και τα αρχικά γράμματα σημαίνουν υπηρεσία συνδιάλεξης σε πραγματικό χρόνο μέσω Διαδικτύου. Η επικοινωνία γίνεται μέσω απλού κειμένου. Υπάρχουν κάποια κανάλια στο δίκτυο όπου οι χρήστες μπορούν να μπουν και να ανταλλάξουν μηνύματα. Σε αυτά τα δίκτυα στηρίζεται το πρόγραμμα για να συνδεθεί.

Παράδειγμα σύνδεσης πελατών στο δίκτυο IRC



Κεφάλαιο 2: Σκοπός της πτυχιακής

Σκοπός αυτής της πτυχιακής είναι η δημιουργία ενός προγράμματος ανταλλαγής αρχείων αλλά χωρίς να απαιτείται κάποιος ειδικός διακομιστής. Το Πρόγραμμα πρέπει να υποστηρίζει: εύρεση και σύνδεση με τους υπόλοιπους χρήστες, σταθερότητα του δικτύου σε περίπτωση που κάποιος χρήστης αποσυνδεθεί από το δίκτυο απότομα, αναζήτηση αρχείων στους υπόλοιπους χρήστες και τέλος μεταφορά αυτών των αρχείων.

Το κύριο πλεονέκτημα αυτής της υλοποίησης είναι ότι δεν χρειάζεται κάποιος ειδικός διακομιστής παρά μόνο κάποιος απλός IRC server που υπάρχουν χιλιάδες. Το Πρόγραμμα δεν θα φορτίζει τον διακομιστή παρά μόνο ελάχιστα εξαιτίας της λειτουργίας σύνδεσης που χρησιμοποιείτε (ουράς). Έτσι κάθε πελάτης αναλαμβάνει και την δουλειά του εξυπηρετητή. Επίσης το κόστος λειτουργίας του είναι μηδενικό αφού δεν χρειάζεται κάποιον ειδικό διακομιστή.

Κεφάλαιο 3: Ανάλυση απαιτήσεων

3.1 Κατηγορίες χρηστών

Αυτό το πρόγραμμα απευθύνεται σε όλους τους χρήστες που επιθυμούν να ανταλλάξουν αρχεία και να μοιράσουν αυτά που ήδη έχουν χωρίς να λειτουργήσουν κάποιον server.

3.2 Περιβάλλον ανάπτυξης

Για την ανάπτυξη του κώδικα καθώς και της επαφής χρησιμοποίησα την γλώσσα προγραμματισμού Visual Basic 6 από την Microsoft. Αυτή η γλώσσα παρέχει όλα τα απαραίτητα εργαλεία για την δημιουργία του προγράμματος χωρίς κάποιο εξωτερικό βοήθημα. Δημιουργείτε κώδικας που είναι γρήγορος και τρέχει σε όλες τις πλατφόρμες windows.

3.3 Κύρια τμήματα του συστήματος

Το πρόγραμμα αποτελείται από μια φόρμα η οποία περιέχει όλα τα απαραίτητα στοιχεία για τον έλεγχο του προγράμματος από τον χρήστη. Χρησιμοποίησα τα κλασικά στοιχεία των windows ώστε να είναι πιο φιλική προς τον χρήστη. Τα στοιχεία που χρησιμοποιούνται είναι τα εξής:

Textboxes: Για την εισαγωγή πληροφοριών από τον χρήστη

Command buttons: Για την έναρξη κάποιας ενέργειας

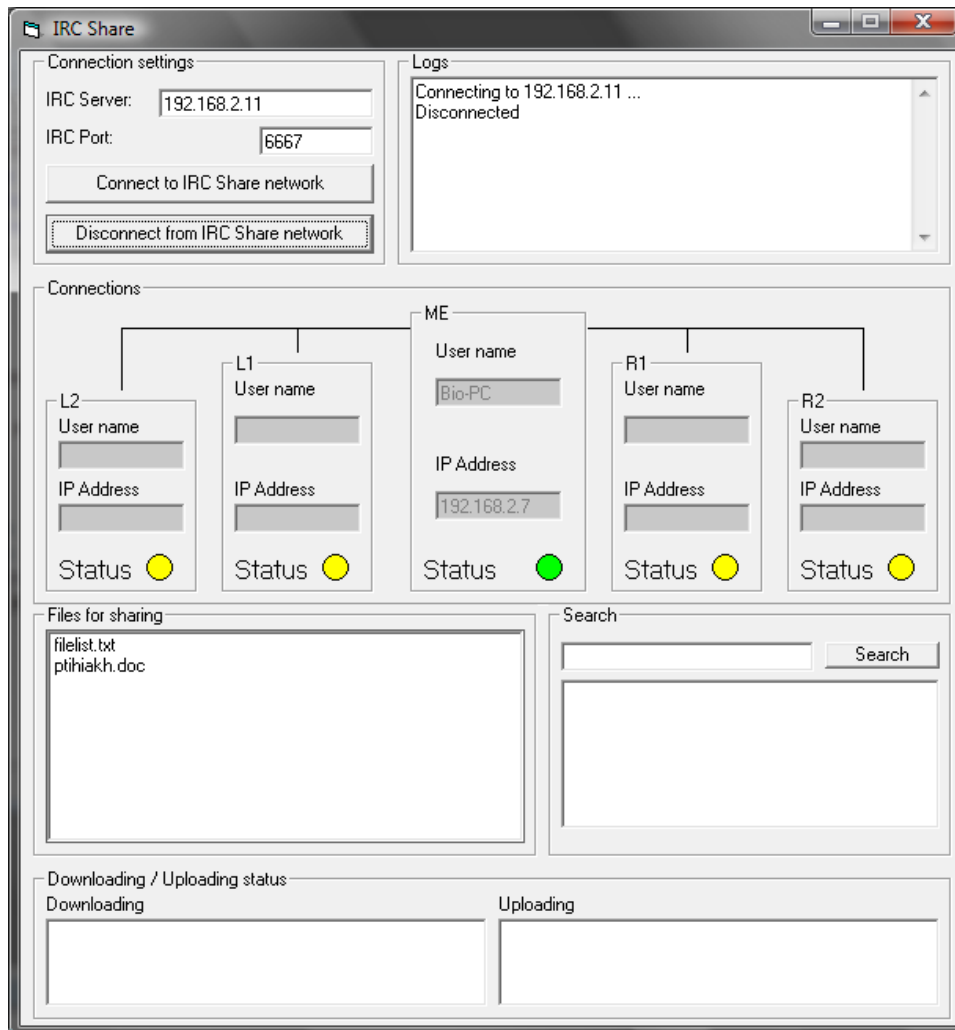
Labels: Για την πληροφόρηση του χρήστη

List boxes: Για την πληροφόρηση του χρήστη

Frames: Για την διάταξη των στοιχείων σε ομάδες

Shapes: Για την πληροφόρηση του χρήστη

Διεπαφή προγράμματος με χρήστη



Περιεχόμενα:

Connection settings

Σε αυτό το σημείο ο χρήστης επιλέγει την διεύθυνση του δικτύου IRC που θα συνδεθεί καθώς επίσης και την θύρα σύνδεσης σε αυτό. Επίσης του δίνονται οι επιλογές για σύνδεση και αποσύνδεση από αυτό.

Connections

Εδώ έχουμε 5 ξεχωριστά πλαίσια που ενημερώνουν τον χρήστη για τις συνδέσεις που έχει δημιουργήσει στο δίκτυο. Ενημερώνετε για τον όνομα του υπολογιστή και την διεύθυνση IP του χρήστη με τον οποίο συνδέθηκε.

Files for sharing

Αυτό το list box μας δείχνει πια αρχεία έχει κάθε χρήστης μέσα στο φάκελο του προγράμματος που ονομάζεται share. Αυτά τα αρχεία είναι που κάνει κοινόχρηστα κάθε χρήστης στους υπόλοιπους σε κάθε αναζήτηση.

Search

Εδώ έχουμε ένα text box από ο χρήστης τοποθετεί το αρχείο που θέλει να κάνει αναζήτηση στους υπόλοιπους χρήστες. Η αναζήτηση αρχίζει με το κουμπί search. Αν το αρχείο βρεθεί στους υπόλοιπους χρήστες τότε το αποτέλεσμα εμφανίζεται στο list box με την διεύθυνση χρήστη που το έχει. Οπότε με ένα διπλό κλικ αρχίζει η διαδικασία κατεβάσματος.

Downloading / Uploading status

Αποτελείτε από 2 list boxes και 2 labels. Το ένα list box αναφέρεται για το κατέβασμα των αρχείων και το άλλο για το ανέβασμα (για αυτά που στέλνει).

Εδώ φαίνονται πια αρχεία κατεβαίνουν και πια ανεβαίνουν καθώς επίσης και το τα πόσα bytes έχουν μεταφερθεί από όλα.

3.4 Βασικές λειτουργίες

3.4.1 Σύνδεση σε δίκτυο IRC

Το πρόγραμμα για να συνδεθεί και να δημιουργήσει το δίκτυο θα πρέπει πρώτα να συνδεθεί σε ένα δίκτυο IRC. Για να συνδεθεί πρέπει να χρησιμοποιήσει εντολές IRC.

Οι εντολές που χρησιμοποιούνται εδώ είναι οι εξής:

Nick: αλλάζει το nickname του χρήστη

Pong: στέλνει στον server μήνυμα απόκρισης

Privmsg: στέλνει μήνυμα σε κάποιον χρήστη

Quit: βγαίνει από το δίκτυο IRC

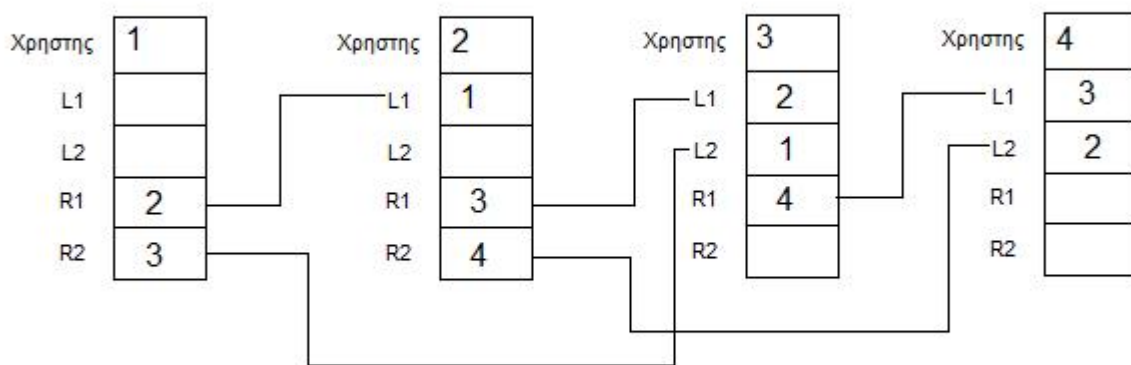
Κάθε νέος χρήστης που συνδέεται στο δίκτυο IRC πρέπει να έχει ξεχωριστό όνομα nickname. Σε περίπτωση που κάποιος έχει ίδιο όνομα τότε το σύστημα τον πληροφορεί και πρέπει να αλλάξει.

Αυτή την τεχνική χρησιμοποιεί και το πρόγραμμα μου.

3.4.2 Δημιουργία αξιόπιστου δικτύου – Ανάκτηση συνδέσεων

Το δίκτυο δημιουργείται με τους χρήστες και δεν βασίζεται σε κάποιον server για να υπάρχει. Κάθε χρήστης αναλαμβάνει και τον ρολό του server. Για να συνδεθεί ένας νέος χρήστης σε ένα υπάρχον δίκτυο πρέπει απλά να συνδεθεί στο IRC δίκτυο και μετά γίνονται αυτόματα οι απαραίτητες συνδέσεις για να μπει στο δίκτυο. Ο καθένας που είναι συνδεδεμένος στο δίκτυο πρέπει να γνωρίζει 2 χρήστες από αριστερά του και 2 χρήστες από δεξιά του. Χρειάζεται 2 για να μεγαλώσει η αξιοπιστία του δικτύου και να γίνονται πιο γρήγορα οι αναζητήσεις αρχείων. Σε περίπτωση που κάποιος χρήστης βγει από το δίκτυο αυτό δεν καταρρέει αφού είναι συνδεδεμένοι και οι υπόλοιποι χρήστες. Έτσι γίνεται αποκατάσταση των συνδέσεων. Ο τρόπος με τον οποίο γίνεται η σύνδεση και η αποκατάσταση θα εξηγηθεί παρακάτω.

Αναπαράσταση συνδέσεων με 4 χρήστες



Εδώ φαίνεται πως ο χρήστης **1** έχει συνδεθεί με τον **2** και **3**. Ο **2** με τον **1**, **3** και **4**. Ο **3** με τον **1**, **2** και **4** και ο **4** με τον **2** και **3**.

Με αυτές τις διπλές συνδέσεις το δίκτυο γίνεται σταθερό και πιο γρήγορο.

3.4.3 Δυνατότητα γρήγορης αναζήτησης

Το πρόγραμμα παρέχει στον χρήστη αναζήτηση αρχείων η οποία γίνεται σε όλους τους υπολογιστές που είναι συνδεδεμένοι στο δίκτυο. Κάθε χρήστης που είναι μέσα στο δίκτυο έχει από ένα σύνολο αρχείων τα οποία φαίνονται και στους υπόλοιπους χρήστες σε κάθε αναζήτηση.

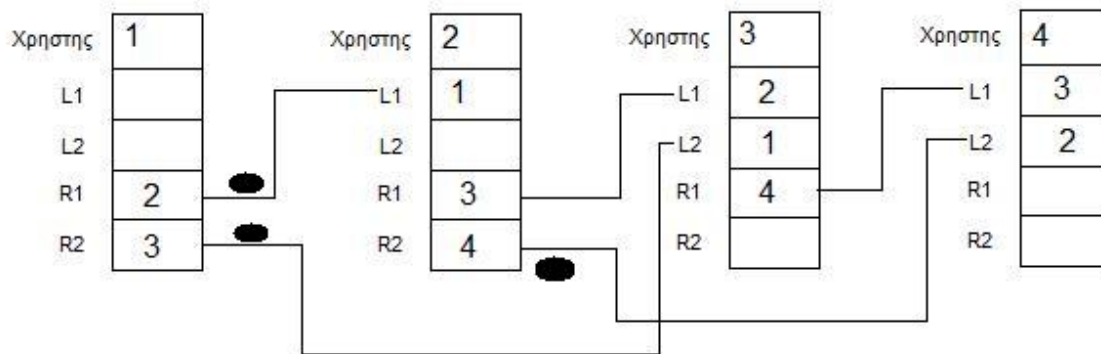
Για να γίνει αναζήτηση σε τέτοιου είδους δίκτυο θα πρέπει κάθε χρήστης να αναλάβει και τον ρόλο του εξυπηρετητή και αυτό γιατί δεν υπάρχει κάποιος κεντρικός υπολογιστής. Οπότε σε κάθε αναζήτηση κάθε χρήστης στο δίκτυο ψάχνει αν έχει το αρχείο που ζητείτε και προωθεί την αναζήτηση στον επόμενο χρήστη έως ότου ρωτηθούν όλοι στο δίκτυο.

Αυτή η αναζήτηση και προώθηση όμως έχει και ένα σημαντικό πρόβλημα. Τα δίκτυα που λειτουργούν με βάση κάποιον server προωθούν την αναζήτηση ταυτόχρονα σε πολλούς χρήστες με αποτέλεσμα να χρειάζεται μικρό χρονικό διάστημα για να τελειώσει.

Στο δικό μου δίκτυο όμως η φύση σύνδεσης μεταξύ των υπολογιστών καθώς και η έλλειψη server αναγκάζει την αναζήτηση να γίνεται καθυστερημένα. Το πρόβλημα δεν εμφανίζεται σε μικρά δίκτυα (500 υπολογιστές) αλλά σε μεγαλύτερα (πάνω από 1000) όπου εμφανίζεται καθυστέρηση σε κάθε αναζήτηση.

Αυτόν τον χρόνο κατάφερα να τον μειώσω κατά 4 φορές χάρις στην παράλληλη αναζήτηση. Κάθε χρήστης που θέλει να κάνει αναζήτηση στέλνει το ερώτημα του σε 4 χρήστες, 2 από αριστερά του και 2 από δεξιά. Αυτοί οι 4 χρήστες τώρα προωθούν το ερώτημα σε άλλους χρήστες. Τα ερωτήματα που αποστέλλονται δεν επιβαρύνουν το δίκτυο καθώς έχουν πολύ μικρό μέγεθος.

Παράδειγμα αναζήτησης



Εδώ βλέπουμε ένα απλό παράδειγμα αναζήτησης. Ο χρήστης που κάνει το ερώτημα είναι ο 1 και το στέλνει στους διαθέσιμους χρήστες που είναι ο 2 και 3. Το ερώτημα το παίρνει ο 2, ψάχνει αν έχει το αρχείο και προωθεί το ερώτημα στον χρήστη 4. Το ερώτημα το έχει δεχτεί επίσης και ο χρήστης 3 ο οποίος κάνει ακριβώς το ίδιο αλλά δεν το προωθεί πουθενά. Τέλος ψάχνει και ο χρήστης 4 από το ερώτημα που του προωθήθηκε από τον 2.

Όποιος χρήστης έχει το αρχείο προς εύρεση, ενημερώνει τον χρήστη 1 για την διάθεση του.

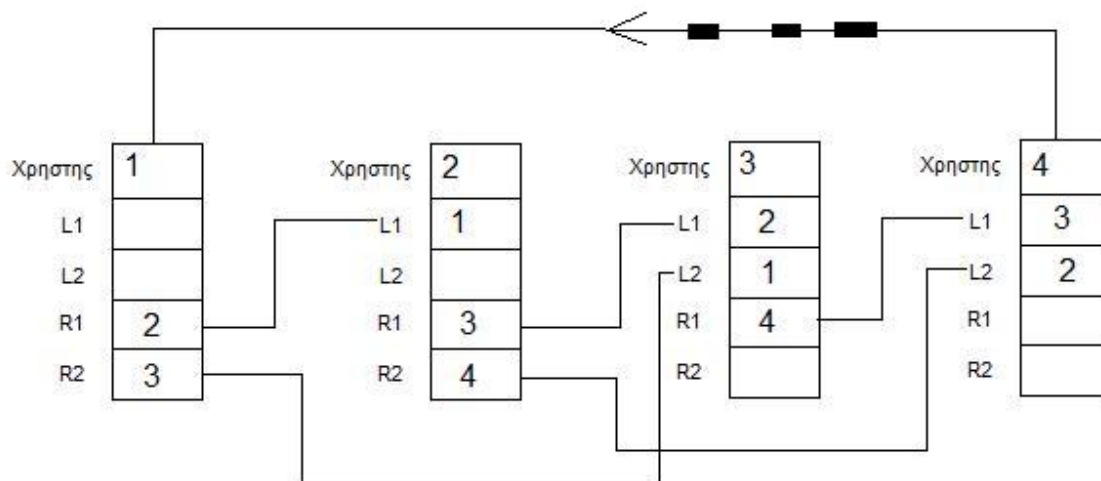
3.4.4 Μεταφορά αρχείων

Ο σκοπός κάθε δικτυακού προγράμματος τέτοιου είδους πρέπει να υποστηρίζει ασφαλώς την μεταφορά αρχείων. Αυτό δεν θα μπορούσε να λείπει από το πρόγραμμα που παρουσιάζω.

Η μεταφορά αρχείων γίνεται μέσω του πρωτοκόλλου TCP με απευθείας συνδέσεις.

Για να ξεκινήσει η διαδικασία μεταφοράς θα πρέπει πρώτα ο χρήστης να ξέρει ποιος έχει το αρχείο που θέλει. Αυτό γίνεται με την αναζήτηση στο προηγούμενο βήμα. Αφού λοιπόν ο χρήστης ξέρει ποιος έχει το αρχείο τότε το ζητά από τον αποστολέα. Ο αποστολέας του απαντάει με ένα OK και τότε αρχίζει η διαδικασία μεταφοράς έως ότου αποσταλούν όλα τα bytes.

Παρακάτω βλέπουμε ένα παράδειγμα με 4 χρήστες:



Εδώ ο χρήστης 1 έκανε το ερώτημα για κάποιο αρχείο. Ο 4 απάντησε ότι έχει στην διάθεση του αυτό το αρχείο και του έστειλε πληροφορίες για να αρχίσει η διαδικασία αποστολής εφόσον το επιλέξει ο 1. Ο 1 με το που επιλεγεί να το κατεβάσει, δημιουργεί μια απευθείας σύνδεση με τον χρήστη 4 και του ζητά το αρχείο. Ο χρήστης 4 του απατάει θετικά και ξεκινά η μεταφορά του αρχείου. Το αρχείο αποθηκεύεται σιγά σιγά στον σκληρό δίσκο του χρήστη 1 και γίνεται προσπελάσιμο όταν τελειώσει η μεταφορά. Το πρόγραμμα εκμεταλλεύεται όλη την διαθέσιμη χωρητικότητα του δικτύου οπότε το αρχείο αποστέλλεται με την μέγιστη δυνατή ταχύτητα.

Κεφάλαιο 4: Σχεδιασμός προγράμματος

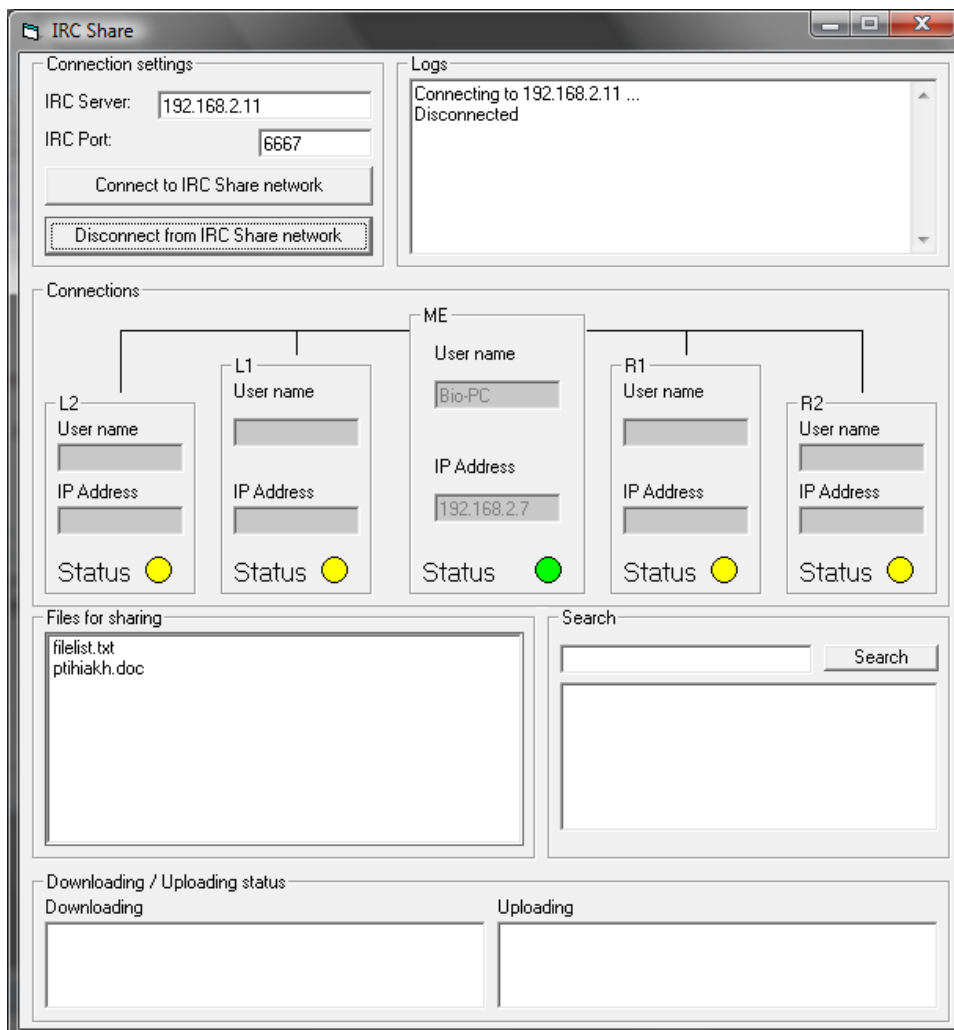
4.1 Σχεδιασμός γραφικού περιβάλλοντος διεπαφής χρήστη

Η σχεδίαση ολόκληρης της επαφής είναι πολύ απλή και δεν περιλαμβάνει κάποιο μενού. Αυτό έγινε για να είναι πιο εύκολο στον χρήστη και να παρουσιαστούν οι βασικότερες λειτουργίες του προγράμματος.

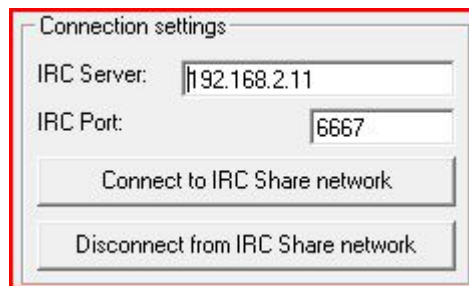
Η σχεδίαση της έγινε μέσα από την Visual Basic 6 μόνο με τα βασικά στοιχεία της εφαρμογής.

Το πρόγραμμα τρέχει από αναλύσεις οθόνης 1024 και πάνω γιατί αλλιώς δεν θα χωρέσει ολόκληρο στην οθόνη.

Το κεντρικό παράθυρο διαθέτει το στοιχείο για να τερματίζεται η εφαρμογή και να ελαχιστοποιείται.



Το πρώτο τμήμα του προγράμματος αποτελείται από ένα frame box ώστε να διαχωρίζει τα στοιχεία. Βρίσκεται στην κορυφή γιατί είναι το πρώτο που πρέπει να επιλέξει ο χρήστης. Μέσα στο frame box υπάρχει ένα text box ώστε να εισάγει ο χρήστης την διεύθυνση του δικτύου IRC που θέλει να συνδεθεί. Μπορεί να εισάγει είτε IP διεύθυνση είτε με γράμματα. Στο δεύτερο text box εισάγει τη θύρα στην οποία θα συνδεθεί το πρόγραμμα. (συνήθως 6667) Τα επόμενα 2 κουμπιά είναι για να επιλεγεί αν θέλει να συνδεθεί η να αποσυνδεθεί.



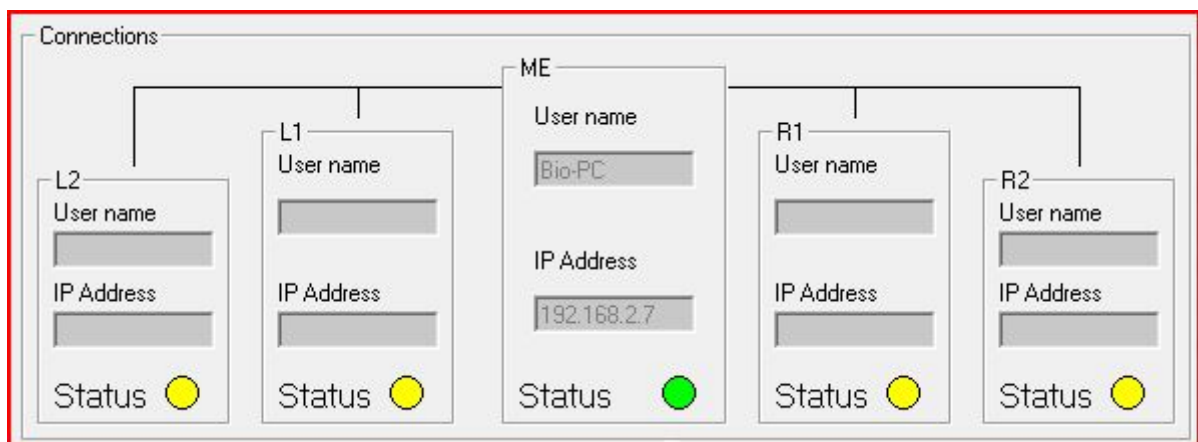
Στο δεύτερο τμήμα δεξιά αποτελείτε πάλι από ένα frame box ώστε να διαχωρίζεται διακριτά από τα υπόλοιπα στοιχεία και μέσα του περιέχει ένα text box. Αυτό είναι για να καταγράφει και να εμφανίζει στον χρήστη ότι συμβαίνει στο πρόγραμμα και τις συνδέσεις που δημιουργούνται. Κάθε φορά που γεμίζει γίνεται αυτόματη κύλιση προς τα κάτω ώστε να εμφανίζονται τα τελευταία.



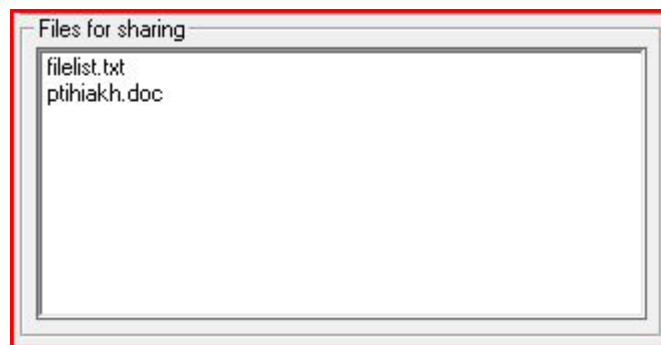
Στο τρίτο και κεντρικό μέρος της διεπαφής βλέπουμε τις συνδέσεις που πραγματοποιούνται με τους υπόλοιπους χρήστες. Αποτελείτε από 5 frame boxes που διαχωρίζουν τις συνδέσεις εικονικά. Κάθε frame box περιέχει:

- ένα text box το οποίο παίρνει το όνομα του χρήστη με τον οποίο έγινε η σύνδεση
- ένα text box το οποίο παίρνει την διεύθυνση IP του χρήστη με τον οποίο έγινε η σύνδεση
- ένα shape το οποίο αλλάζει χρώμα ανάλογα με την κατάσταση σύνδεσης (κίτρινο όταν βρίσκεται σε κατάσταση αναμονής, κόκκινο όταν είναι σε κατάσταση λάθους και πράσινο όταν έχει πραγματοποιηθεί σύνδεση με κάποιον χρήστη)

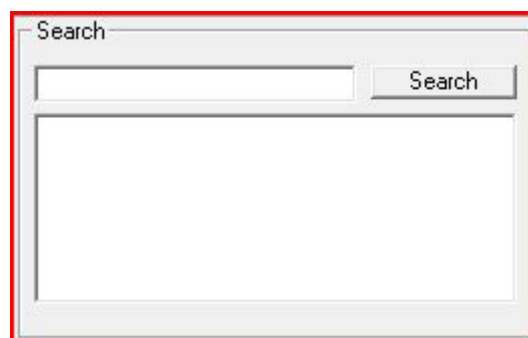
Το μεσαίο frame box μας πληροφορεί για την δική μας κατάσταση (το όνομα του υπολογιστή μας, την διεύθυνση IP που έχουμε στο δίκτυο και αν το socket βρίσκεται σε κατάσταση αναμονής)



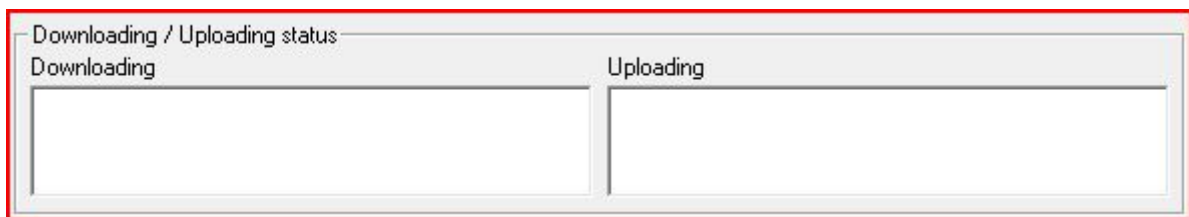
Σε αυτό το κομμάτι βλέπουμε ένα list box το οποίο εμφανίζει τα αρχεία που είναι διαθέσιμα στους άλλους χρήστες. Τα αρχεία που εμφανίζονται εδώ βρίσκονται μέσα στον φάκελο share του προγράμματος. Κάθε καινούριο αρχείο που μπαίνει στο φάκελο προστίθεται και στην λίστα αυτόματα.



Εδώ έχουμε 3 στοιχεία, ένα text box, ένα command button και ένα list box. Από εδώ ο χρήστης πραγματοποιεί την αναζήτηση θέτοντας το όνομα του αρχείου που επιθυμεί να κατεβάσει στο text box. Μετά πατεί το κουμπί search και τα αποτελέσματα εμφανίζονται στο list box το ένα κάτω από το άλλο. Με διπλό κλικ μέσα στην λίστα πάνω σε κάποιο αρχείο ενεργοποιείτε η διαδικασία μεταφοράς.



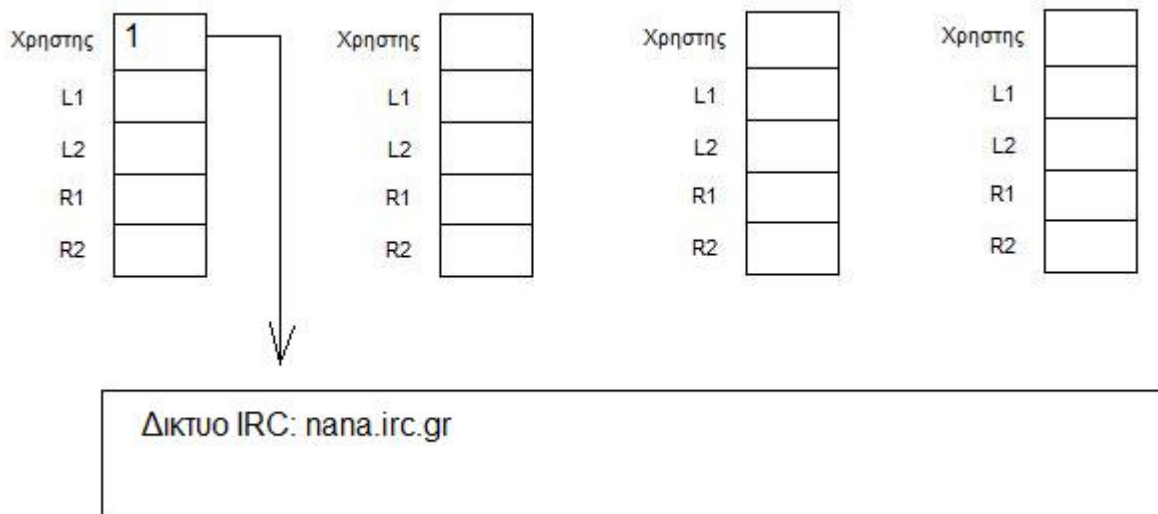
Σε αυτό το τελευταίο κομμάτι βλέπουμε ένα frame box που περιέχει όλα τα στοιχεία, ένα label downloading και ένα uploading, και 2 list boxes. Κάθε φορά που πραγματοποιείτε κάποια διαδικασία μεταφοράς αρχείου, αυτή προστίθεται εδώ. Το όνομα του αρχείου που αποστέλλεται ή παραλαμβάνεται προστίθεται στην λίστα. Τα bytes που αποστέλλονται ή παραλαμβάνονται εμφανίζονται στα ανάλογα labels (downloading – uploading) για να πληροφορούν τον χρήστη για την κατάσταση μεταφοράς.



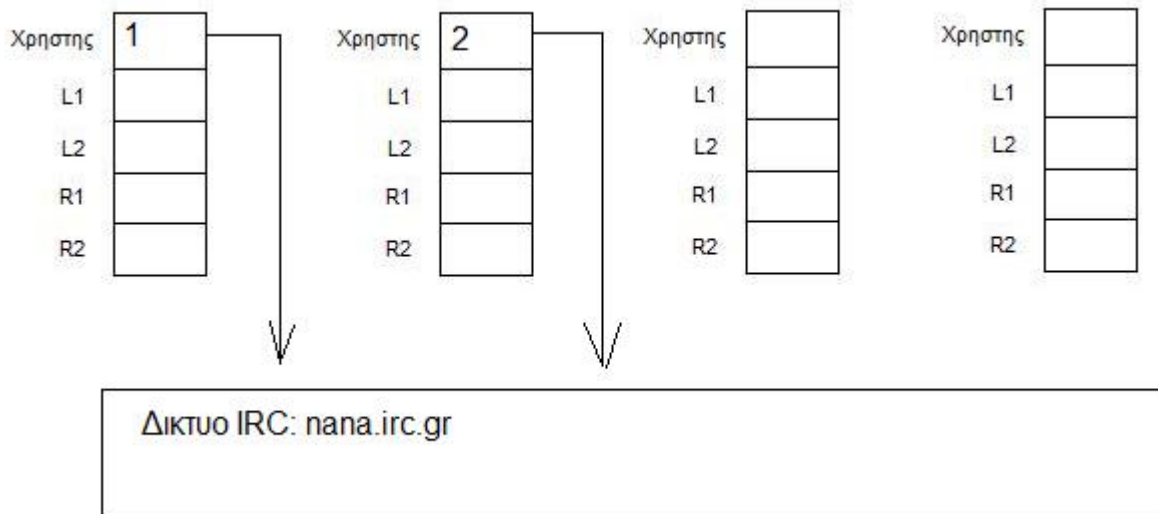
4.2 Σύνδεση σε δίκτυο IRC

Για να ξεκινήσει η διαδικασία δημιουργίας του δικτύου θα πρέπει πρώτα ο χρήστης να πραγματοποιήσει σύνδεση με ένα δίκτυο IRC. Αυτό το δίκτυο θα πρέπει να είναι κοινό για όλους τους χρήστες ώστε να ανήκουν στο ίδιο δίκτυο.

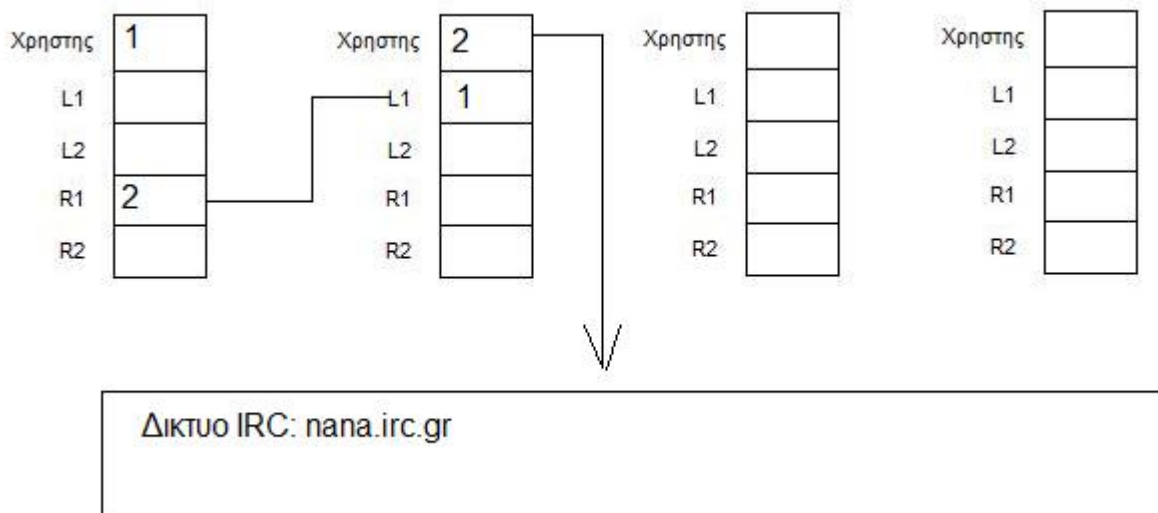
Στο παρακάτω σχέδιο βλέπουμε πως γίνεται σχηματικά μια σύνδεση στο IRC



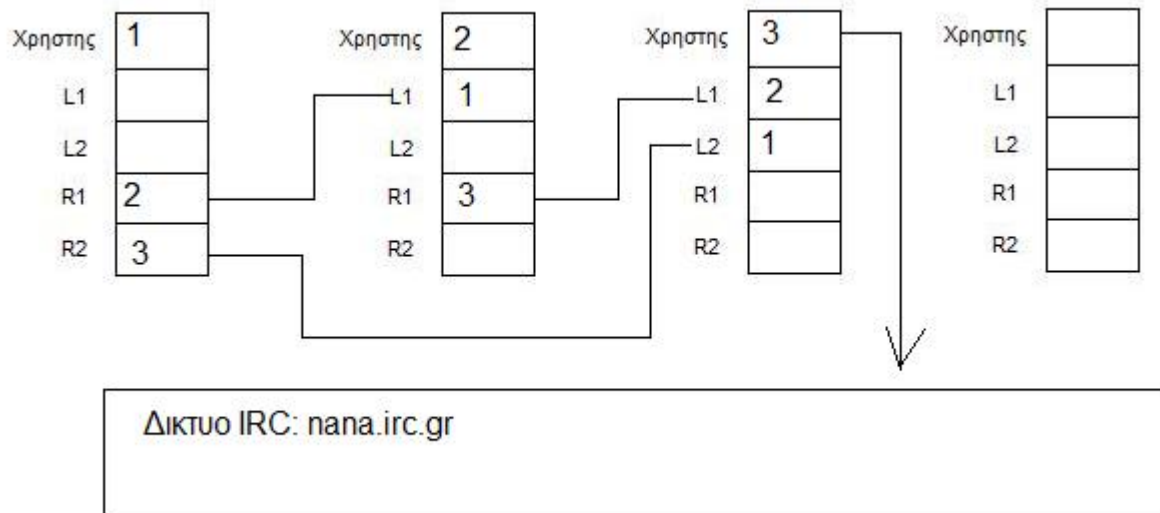
Εδώ ο χρήστης 1 πραγματοποιεί μια σύνδεση με το IRC nana.irc.gr και περιμένει τον επόμενο χρήστη να συνδεθεί.



Παραπάνω βλέπουμε πως έχει συνδεθεί και ο χρήστης 2 στο ίδιο δίκτυο. Ο 2 χρήστης βλέπει ότι είναι ήδη συνδεδεμένος ο χρήστης 1 και επικοινωνούν μεταξύ τους μέσω εντολών IRC.



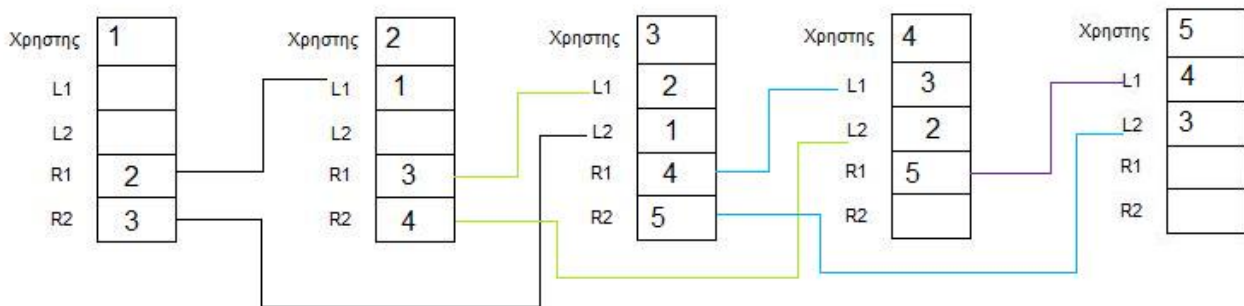
Οι χρήστες 1 και 2 έχουν ανταλλάξει διευθύνσεις και συνδέθηκαν μεταξύ τους. Μετά από αυτό το βήμα ο χρήστης 1 βγαίνει από το δίκτυο IRC και την θέση του παίρνει ο 2.



Σε αυτό το βήμα συνδέθηκε ο χρήστης 3 στο δίκτυο IRC, αντάλλαξε διευθύνσεις με τον χρήστη 2 και έγιναν οι κατάλληλες συνδέσεις. Τώρα μόνο ο χρήστης 3 είναι μέσα στο δίκτυο IRC και περιμένει τον επόμενο.

4.3 Δίκτυο και αποκατάσταση συνδέσεων

Κάθε χρήστης στο δίκτυο δημιουργεί 4 συνδέσεις με τους υπόλοιπους χρήστες. Αυτό γίνεται ώστε να χτιστεί ένα σταθερό και αποτελεσματικό δίκτυο μεταξύ των χρηστών. Θα μπορούσε κάλλιστα να δημιουργεί περισσότερες συνδέσεις αλλά τότε θα καθυστερούσε σημαντικά το δίκτυο. Όλες οι συνδέσεις γίνονται με το πρωτόκολλο TCP και χρησιμοποιείται το στοιχείο Winsock της Visual Basic 6. Ένα άλλο σημαντικό στοιχείο που έχει το πρόγραμμα είναι η αποκατάσταση συνδέσεων. Χωρίς αυτήν το δίκτυο θα κατέρρεε έστω και με έναν υπολογιστή εκτός. Παρακάτω θα δούμε σχηματικά πως λειτουργούν και τα δύο.



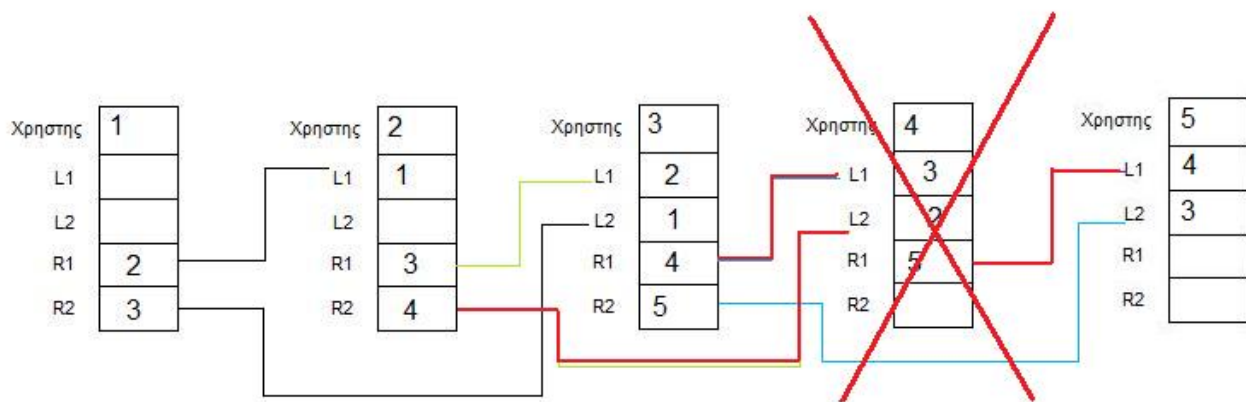
Σε αυτό το σημείο βλέπουμε 5 χρήστες συνδεδεμένους μεταξύ τους. Τα σήματα L1,L2,R1,R2 σημαίνουν τα εξής:

- L1 Η σύνδεση που υπάρχει με τον χρήστη μια θέση αριστερά του
- L2 Η σύνδεση που υπάρχει με τον χρήστη δύο θέσεις αριστερά του
- R1 Η σύνδεση που υπάρχει με τον χρήστη μια θέση δεξιά του
- R2 Η σύνδεση που υπάρχει με τον χρήστη δύο θέσεις δεξιά του

Όλες οι συνδέσεις δημιουργούνται από τα δεξιά προς τα αριστερά.

Πρώτα συνδέθηκε ο χρήστης 2 με τον 1, μετά ο 3 με τον 2 και τον 1, μετά ο 4 με τον 3 και τον 2 και τέλος ο 5 με τον 4 και 3.

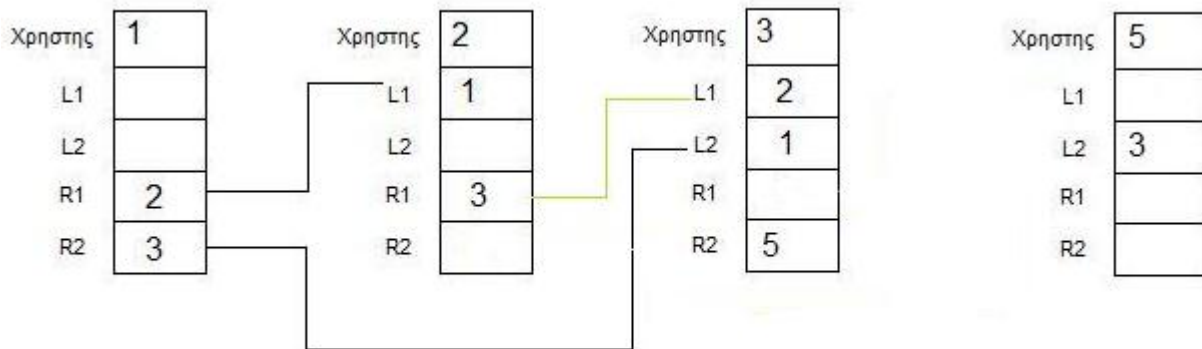
Αποκατάσταση σύνδεσης



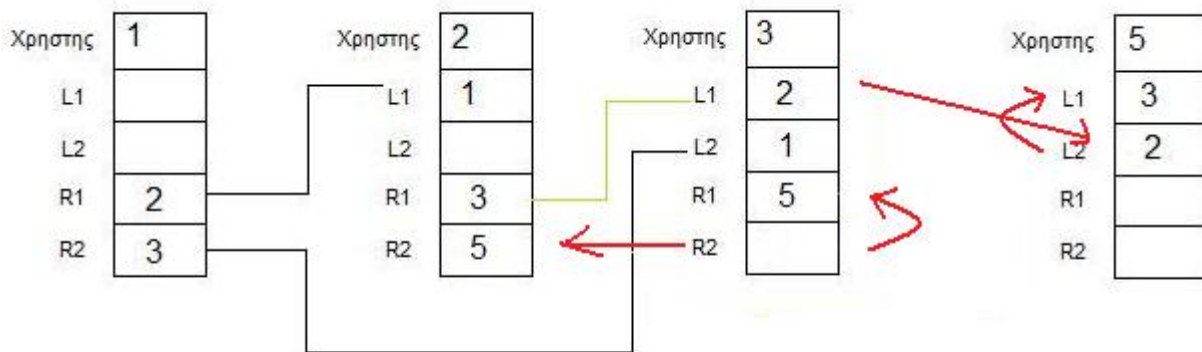
Εδώ ο χρήστης 4 έφυγε από το δίκτυο απρόσμενα.

Όλες οι συνδέσεις που εξαρτιόταν από αυτόν έχουν τερματιστεί. (είναι με κόκκινο)

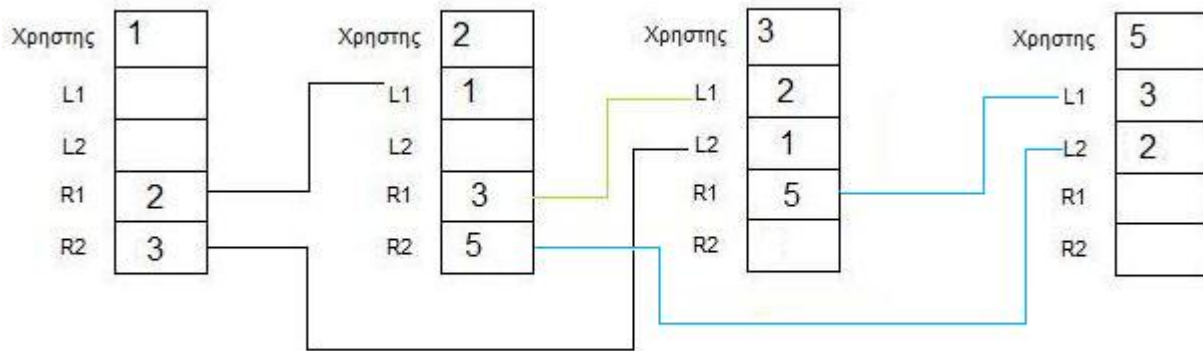
Επειδή το δίκτυο είναι τύπου αλυσίδας θα πρέπει να ξανά ενωθεί. Για αυτόν το λόγο υπάρχουν οι διπλές συνδέσεις. Θα δούμε στο επόμενο σχήμα πώς γίνεται η αποκατάσταση.



Ο χρήστης 4 έχει φύγει τελείως από το δίκτυο. Έτσι διαγράφηκαν και οι συνδέσεις από τους εξαρτώμενους χρήστες (2,3,5).



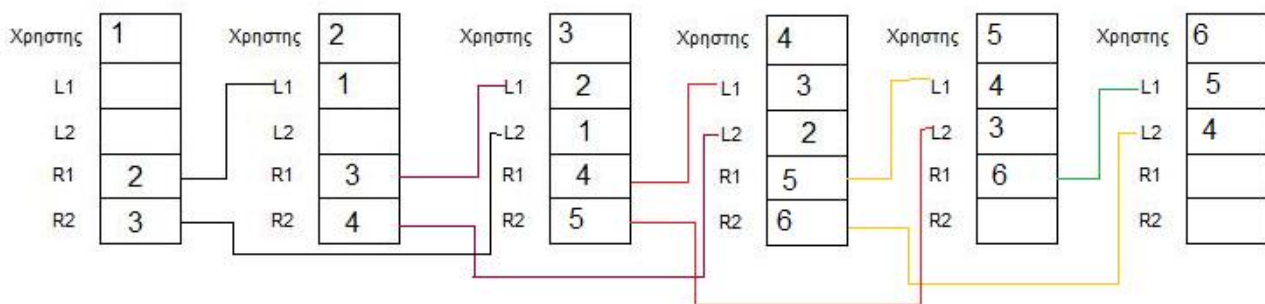
Τώρα με τα κόκκινα βελάκια βλέπουμε πως αναδιατάσσονται οι συνδέσεις στους χρήστες 2,3 και 5. Ο χρήστης 2 είχε χάσει το R2 του οπότε του στέλνει ο 3 το R2 του που είναι ο 5. Ο χρήστης 3 είχε χάσει το R1 του. Μετακινεί μια θέση πιο κάτω το R2 και γίνεται R1. Επίσης στέλνει το L1 του στον χρήστη 5 για να το βάλει στην θέση L2 αφού πρώτα ο 5 μετακινήσει το L2 του στο L1.



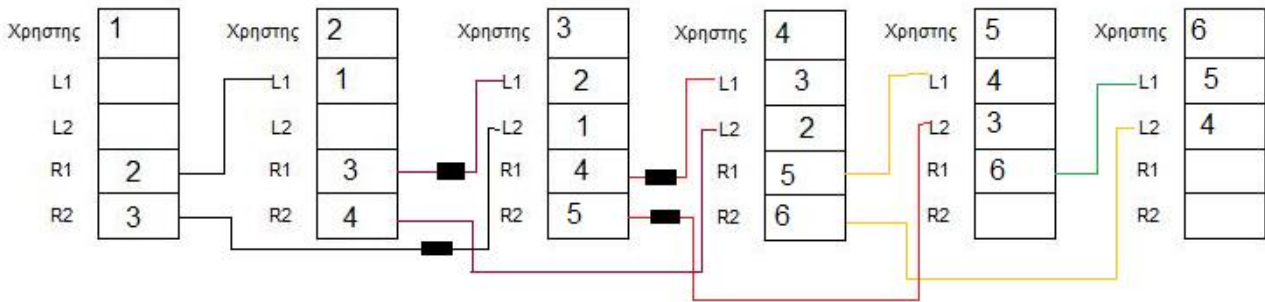
Έτσι θα είναι το δίκτυο μας μετά από τις αποκαταστάσεις συνδέσεων. Το δίκτυο λειτουργεί πάλι κανονικά σαν να μην υπήρχε ποτέ ο χρήστης 4.

4.4 Αναζήτηση

Το πρόγραμμα διαθέτει δυνατότητα αναζήτησης στο δίκτυο. Η αναζήτηση πρέπει να πραγματοποιείται γρήγορα ώστε σε μεγάλα δίκτυα να μην παρατηρείτε καθυστέρηση. Στα παρακάτω σχέδια θα δούμε πως πραγματοποιείται μια αναζήτηση σε ένα δίκτυο με 6 υπολογιστές.



Σε αυτό το δίκτυο έχουμε 6 χρήστες. Την αναζήτηση θα την ξεκινήσει ο χρήστης 3.

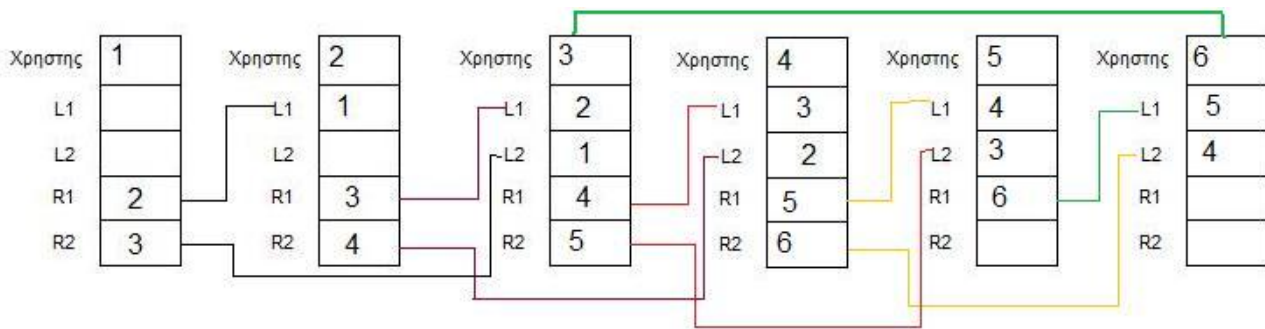


Ο χρήστης 3 στέλνει σε όσους γνωρίζει πακέτα με την αναζήτηση που δημιουργεί. Δηλαδή αυτοί οι χρήστες είναι οι 1,2,3,4.

Ο χρήστης 2 παραλαμβάνει το πακέτο και ψάχνει στα τοπικά του αρχεία για την συγκεκριμένη αναζήτηση. Ας υποθέσουμε πως το αρχείο που επιθυμεί ο χρήστης 3 το έχει μόνο ο 6.

Οι χρήστες 1,4 και 5 παραλαμβάνουν επίσης το πακέτο με την αναζήτηση και ψάχνουν και αυτοί στα τοπικά τους αρχεία όπως και ο χρήστης 2 παραπάνω. Εδώ όμως ο 4 έχει στην λίστα του και τον χρήστη 6 οπότε του προωθεί το πακέτο αναζήτησης. Ο 6 αφού παραλάβει το πακέτο διαπιστώνει ότι έχει το αρχείο προς αναζήτηση.

Σε αυτήν την περίπτωση ο 6 δημιουργεί μια απευθείας σύνδεση με τον χρήστη 3 που ξεκίνησε την αναζήτηση για να τον ενημερώσει για την διαθεσιμότητα του αρχείου.

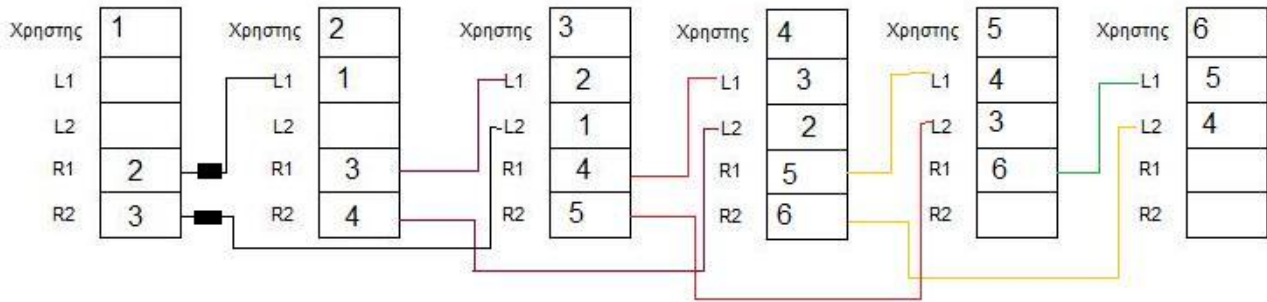


Ο χρήστης 6 έχει δημιουργήσει μια απευθείας σύνδεση με τον 3 (πράσινη γραμμή).

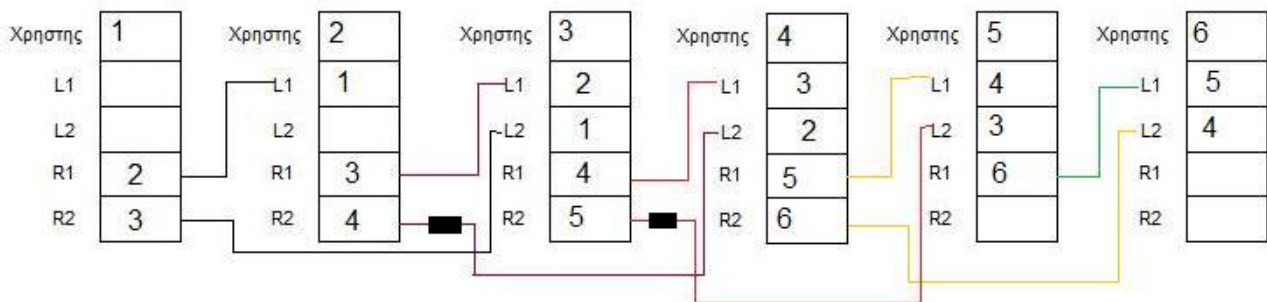
Με αυτή την σύνδεση ο 6 ενημερώνει τον 3 για την διαθεσιμότητα του αρχείου. Μετά την ενημέρωση αυτή η σύνδεση διακόπτεται και το κανάλι απελευθερώνεται έως ότου ο 3 επιθυμήσει το αρχείο.

Θα πρέπει να σημειωθεί εδώ πως κάθε χρήστης που παραλαμβάνει το πακέτο αναζήτησης το προωθεί στους δεύτερους χρήστες που έχει στην λίστα του(στον R2 και L2).

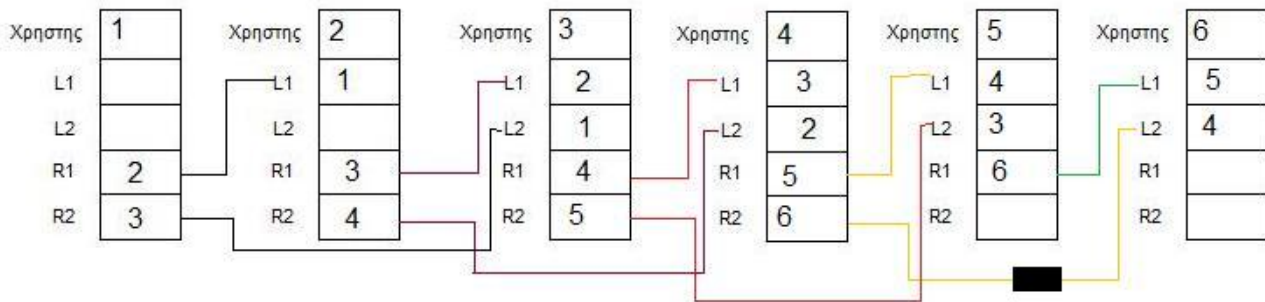
Στο παρακάτω παράδειγμα θα δούμε πως γίνεται η προώθηση των πακέτων.



Σε αυτό το παράδειγμα ο χρήστης 1 ξεκινάει την αναζήτηση και την στέλνει στους διαθέσιμους χρήστες που είναι ο 2 και 3. Η αναζήτηση θα προωθηθεί σε όλους τους χρήστες που είναι συνδεδεμένοι στο δίκτυο άσχετα με το αν βρεθεί το αρχείο σε αρχική φάση.



Οι χρήστες 2 και 3 προωθούν την αναζήτηση στους δεύτερους συνδεδεμένους που είναι οι 4 και 5 αντίστοιχα. Βλέπουμε εδώ ότι η αναζήτηση σε αυτήν τη φάση προχωράει παράλληλα με δύο πακέτα και όχι με 4 γιατί αριστερά από τον χρήστη 1 δεν υπάρχουν άλλοι. (είναι ο δημιουργός του δικτύου)



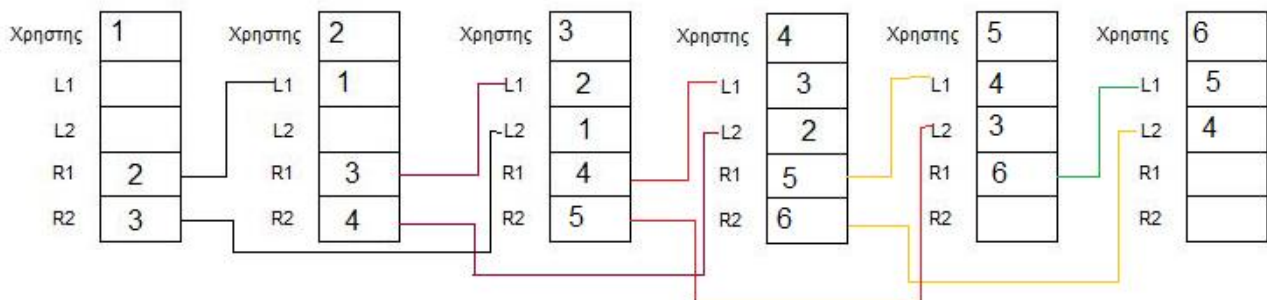
Ο 4 προωθεί το πακέτο αναζήτησης στον 6. Αυτή θα είναι και η τελευταία προώθηση που θα γίνει στο δίκτυο, μιας και ο 6 είναι ο τελευταίος χρήστης.

Σε αυτήν την αναζήτηση όπως και στην προηγούμενη, όποιος χρήστης βρει το αρχείο, δημιουργεί μια απευθείας σύνδεση με τον χρήστη που την δημιούργησε για να τον ενημερώσει για την διαθεσιμότητα του.

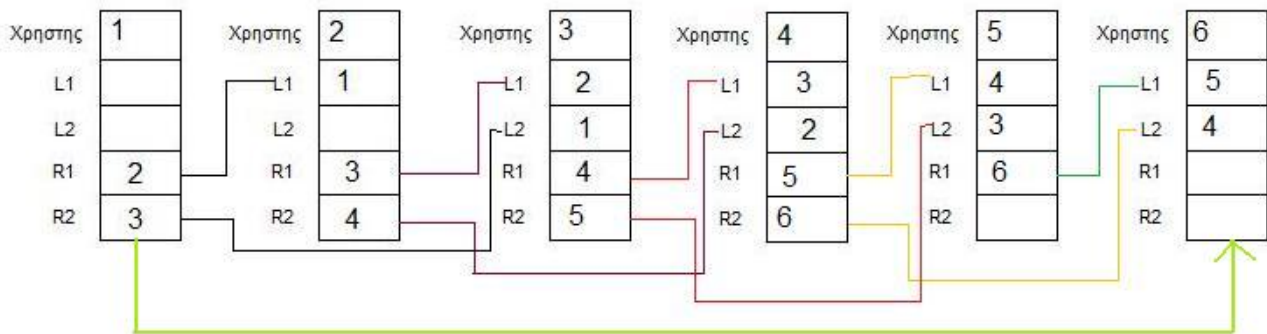
4.5 Μεταφορά αρχείων

Ο χρήστης που πραγματοποίησε την αναζήτηση, εάν έχει αποτελέσματα επιστροφής από τους υπόλοιπους, έχει την δυνατότητα για την μεταφορά αυτού του αρχείου στον υπολογιστή του.

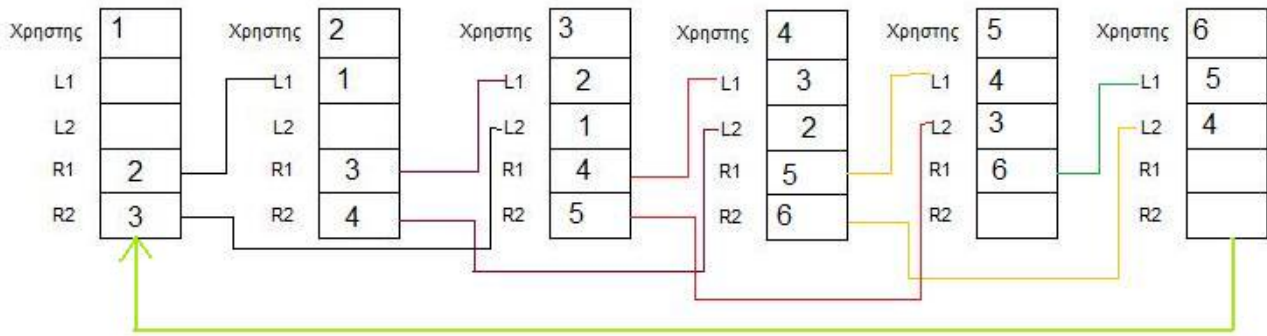
Αυτό πραγματοποιείτε με τον εξής τρόπο:



Θα υποθέσουμε ότι ο χρήστης 1 από το προηγούμενο παράδειγμα επιθυμεί να μεταφέρει το αρχείο που αναζήτησε από τον χρήστη 6 (αφού μόνο αυτός δήλωσε ότι το έχει).



Στην αρχή ο 1 δημιουργεί μια απευθείας σύνδεση με τον 6 αφού ξέρει από την αναζήτηση ότι έχει το αρχείο. Με αυτήν την σύνδεση ζητάει το αρχείο από τον 6 και του στέλνει πληροφορίες για το ποιο αρχείο θέλει να κατεβάσει. Από την μεριά του ο 6 θα του απαντήσει θετικά ή αρνητικά.



Στην περίπτωση που ο 6 είναι διαθέσιμος θα απαντήσει θετικά και θα του στείλει το αρχείο. Το αρχείο θα μεταφέρεται με ρυθμό ίσο με την διαθέσιμη χωρητικότητα του δικτύου. Αυτό όμως δεν θα προκαλέσει πρόβλημα στο δίκτυο για τις υπόλοιπες δραστηριότητες του χρήστη, γιατί όλες οι συνδέσεις πραγματοποιούνται μέσω του πρωτοκόλλου TCP το οποίο διανέμει με τον καλύτερο δυνατό τρόπο την διαθεσιμότητα της σύνδεσης.

Η σύνδεση τερματίζεται μετά την ολοκλήρωση της μεταφοράς του αρχείου και τα στοιχεία της απευθείας σύνδεσης και των δύο χρηστών επανέρχονται σε κατάσταση ακρόασης.

Κεφάλαιο 5: Υλοποίηση Συστήματος

5.1 Γενικά για την υλοποίηση

Το πρόγραμμα υλοποιήθηκε με την χρήση της γλώσσας προγραμματισμού Visual Basic 6 η οποία περιέχει debugger για τον εύκολο εντοπισμό σφαλμάτων.

Στη συνέχεια αυτού του κεφαλαίου γίνεται αναλυτική αναφορά σε όλες τις λειτουργίες της εφαρμογής που αναπτύχθηκε, παρουσιάζονται στιγμιότυπα (screenshots) από την τελική εφαρμογή και δίνονται ορισμένα παραδείγματα χρήσης.

5.2 Εκκίνηση προγράμματος

Κάθε φορά που ξεκινάει το πρόγραμμα να τρέχει εκτελεί ορισμένες εντολές.

Η πρώτη που εκτελείτε είναι για να δει αν το πρόγραμμα εκτελείτε ήδη. Σε τέτοια περίπτωση ενημερώνει τον χρήστη κατάλληλα και τερματίζεται η νέα εφαρμογή.

Μετά από αυτό σειρά έχει ο κατάλογος των αρχείων να ενημερωθεί κατάλληλα με τα τρέχων αρχεία.

Τελευταία διαδικασία που πραγματοποιείτε είναι να τεθούν σε κατάσταση αναμονής συνδέσεων όλα τα στοιχεία του προγράμματος.

Σε αυτή την φάση το πρόγραμμα είναι έτοιμο να πραγματοποιήσει και να δεχτεί συνδέσεις.

Το στοιχείο L1 περιμένει για σύνδεση στην πόρτα 4001

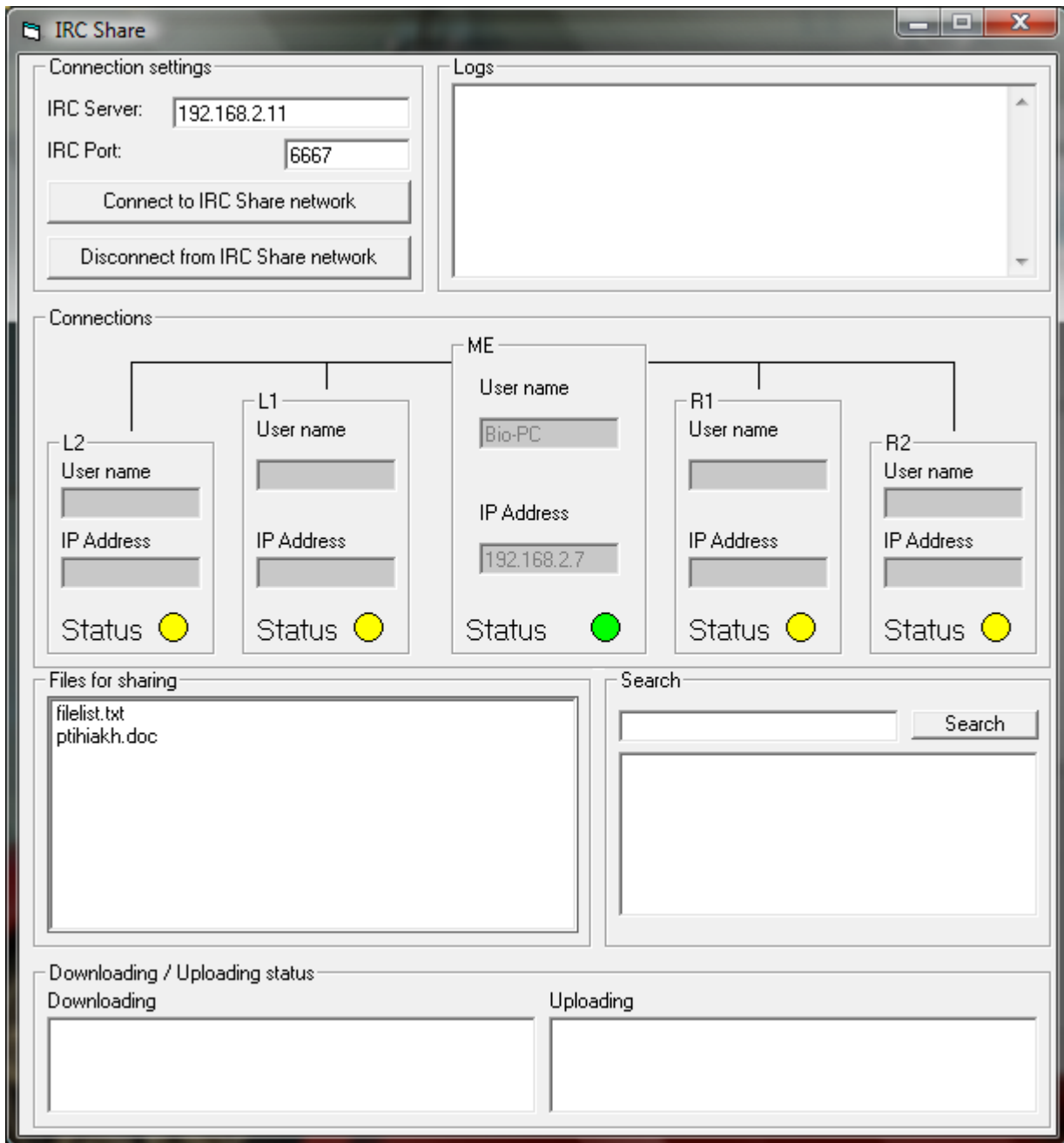
Το στοιχείο L2 περιμένει για σύνδεση στην πόρτα 4002

Το στοιχείο R1 περιμένει για σύνδεση στην πόρτα 4003

Το στοιχείο R2 περιμένει για σύνδεση στην πόρτα 4004

Το στοιχείο που είναι υπεύθυνο για τις συνδέσεις αποτελεσμάτων ακούει στην πόρτα 4005

Σαν τελευταίο βήμα είναι η μορφοποίηση των στοιχείων που εμφανίζονται στον χρήστη ώστε να καταλάβει ότι το πρόγραμμα είναι έτοιμο για λειτουργία.



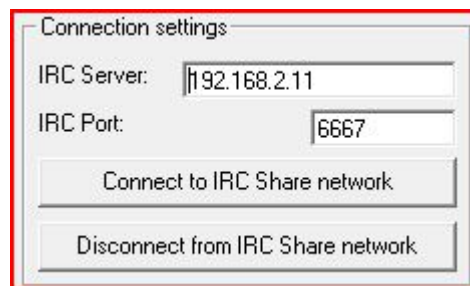
Εδώ βλέπουμε ένα screenshot του προγράμματος κατά την εκκίνηση.

Βλέπουμε ότι η λίστα με τα αρχεία έχει ενημερωθεί κατάλληλα.

Τα στοιχεία των συνδέσεων φαίνονται με την σειρά τους όπου το κίτρινο χρώμα δηλώνει ότι βρίσκονται σε κατάσταση αναμονής ενώ το πράσινο ότι είναι σε σύνδεση. Το μεσαίο στοιχείο είναι πράσινο για να μας δείξει ότι όλα τα στοιχεία του προγράμματος λειτουργούν.

5.3 Σύνδεση με IRC Server

Από την στιγμή που το πρόγραμμα έχει ξεκινήσει είναι έτοιμο να δημιουργήσει σύνδεση με τον IRC Server. Αυτό γίνεται με το κουμπί επιλογής Connect to IRC Network.



Όταν ο χρήστης πατήσει το κουμπί τότε το πρόγραμμα δημιουργεί μια απλή σύνδεση με τον IRC Server και στην θύρα που έχουμε δηλώσει. Στην περίπτωση μας ο Server θα είναι τοπικός και η θύρα η 6667 που είναι standard για σχεδόν όλα τα δίκτυα που υπάρχουν.

Όταν πραγματοποιηθεί η σύνδεση με τον Server το πρόγραμμα του στέλνει το USER ircshare1 και nickname ircshare1. Αυτές οι δύο εντολές είναι απαραίτητες για να δημιουργηθεί η σύνδεση με το IRC. Έτσι λέμε στο IRC server ότι ο χρήστης που συνδέθηκε τώρα θα έχει το όνομα ircshare1. Θα δούμε στα επόμενα βήματα τον λόγο που θέλουμε αυτό το όνομα.

Τώρα εμείς στείλαμε τα απαραίτητα στοιχεία για την σύνδεση και περιμένουμε από τον Server να απαντήσει κατάλληλα.

Αν όλα πήγαν καλά τότε ο server θα στείλει διάφορες πληροφορίες από τις οποίες λίγες μα ενδιαφέρουν.

Σε περίπτωση που στείλει την λέξη PING τότε θα πρέπει να απαντήσουμε άμεσα με την λέξη PONG ώστε να διατηρείτε η σύνδεση.

Σε οποιαδήποτε άλλη περίπτωση τα δεδομένα που περνούμε από τον server θα περάσουν από το φίλτρο του προγράμματος που λέγεται ircCom.

Η ircCom είναι μια διαδικασία η οποία παίρνει όρισμα και λειτουργεί ανάλογα και θα την δούμε στην επόμενη σελίδα αναλυτικά.

Θα πρέπει να σημειώσω ότι όλα τα δεδομένα που παραλαμβάνονται από τον server εκτυπώνονται στο log listbox.

Διαδικασία ircCom

Είναι η πιο σημαντική διαδικασία για το IRC και είναι υπεύθυνη για τις πρώτες συνδέσεις που δημιουργούνται στο δίκτυο. Παίρνει σαν όρισμα τις απαντήσεις που δέχεται από το δίκτυο IRC. Μέσα της περιέχει τρεις if then.

Η πρώτη if εξετάζει το ενδεχόμενο αν είναι η εντολή “in use” που σημαίνει ότι μέσα στο δίκτυο IRC υπάρχει ήδη ένας χρήστης με το ίδιο όνομα.

Σε αυτήν την περίπτωση το πρόγραμμα στέλνει στον server την εντολή να αλλάξει nickname σε ircshare2 ώστε να μπορούν να συνυπάρχουν. Μετά από την αλλαγή του ονόματος ο χρήστης ircshare2 στέλνει στον ircshare1 την εντολή “getout” μέσω μηνυμάτων IRC.

Η δεύτερη if εξετάζει αν η εντολή είναι “getout” (από την προηγούμενη if). Αυτή η if εκτελείτε στον χρήστη με το όνομα ircshare1 γιατί την έστειλε ο ircshare2.

Μόλις διαπιστωθεί ότι είναι αυτή η εντολή τότε ο χρήστης ircshare1 στέλνει στον ircshare2 ένα μήνυμα και την IP διεύθυνση του και βγαίνει τελείως από το δίκτυο IRC με την εντολή QUIT.

Η Τρίτη και τελευταία if εξετάζει αν η εντολή είναι IP. Τέτοιου είδους εντολή στέλνεται μόνο από τον χρήστη ircshare1 και άρα εκτελείτε στον ircshare2. Με το που διαπιστωθεί ότι πρόκειται για τέτοιου είδους εντολή, φιλτράρεται το περιεχόμενο της και το πρόγραμμα αποθηκεύει την διεύθυνση που περιέχεται μέσα στην εντολή. Έπειτα χρησιμοποιεί αυτή τη διεύθυνση που πήρε και δημιουργεί μια σύνδεση από το στοιχείο L1 στην θύρα 4003. Αυτό σημαίνει ότι θα δημιουργήσει σύνδεση με το πρώτα αριστερό άκρο του.

Έχοντας πλέον δημιουργήσει σύνδεση με τον χρήστη ircshare1 και αφού προηγουμένως είδαμε ότι ο ircshare1 έχει βγει από το δίκτυο IRC, το τελευταίο βήμα που απομένει να γίνει είναι να αλλάξει το όνομα του σε ircshare1 ώστε να πάρει την θέση του προηγούμενου χρήστη.

Με λίγα λόγια βλέπουμε ότι στην αρχή ήταν μέσα στο δίκτυο IRC μόνο ο χρήστης ircshare1 και περίμενε να συνδεθεί στο ίδιο δίκτυο ο επόμενος χρήστης. Μετά την σύνδεση του δεύτερου χρήστη, είδαμε ότι έδιωξε τον πρώτο αφού ανταλλάζανε πρώτα διευθύνσεις και έγιναν οι απαραίτητες συνδέσεις και μετέπειτα πήρε και το όνομα του περιμένοντας τώρα αυτός για νέους χρήστες.

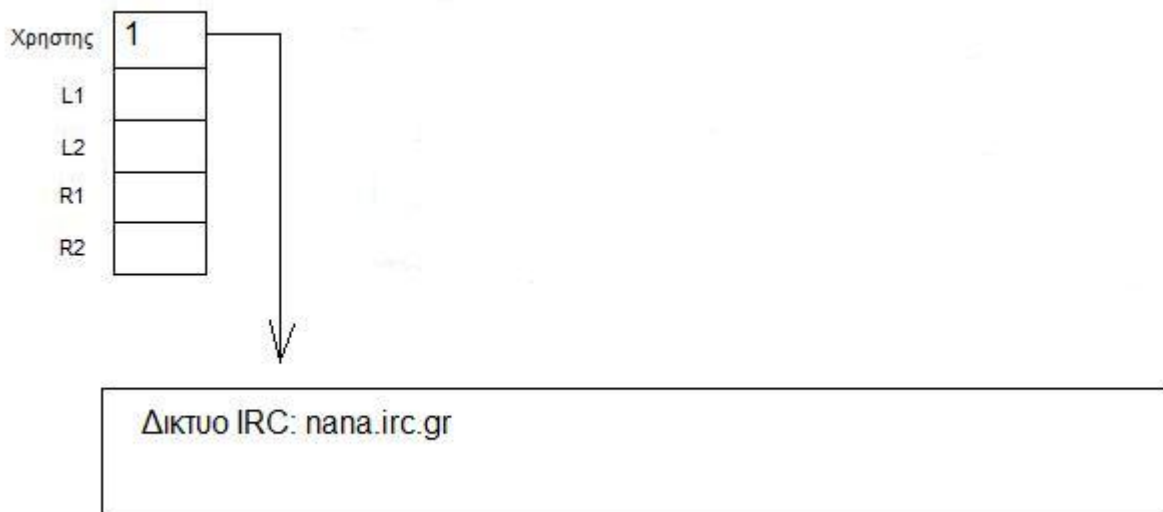
Θα εξηγηθεί παρακάτω με παραδείγματα και σχήματα αναλυτικά η διαδικασία.

5.4 Δημιουργία συνδέσεων

Ο καλύτερος τρόπος για να εξηγηθεί η διαδικασία συνδέσεων είναι με ένα μεγάλο παράδειγμα βήμα-βήμα.

Θα συνδεθούν στο σύστημα πέντε χρήστες.

Πρώτος χρήστης

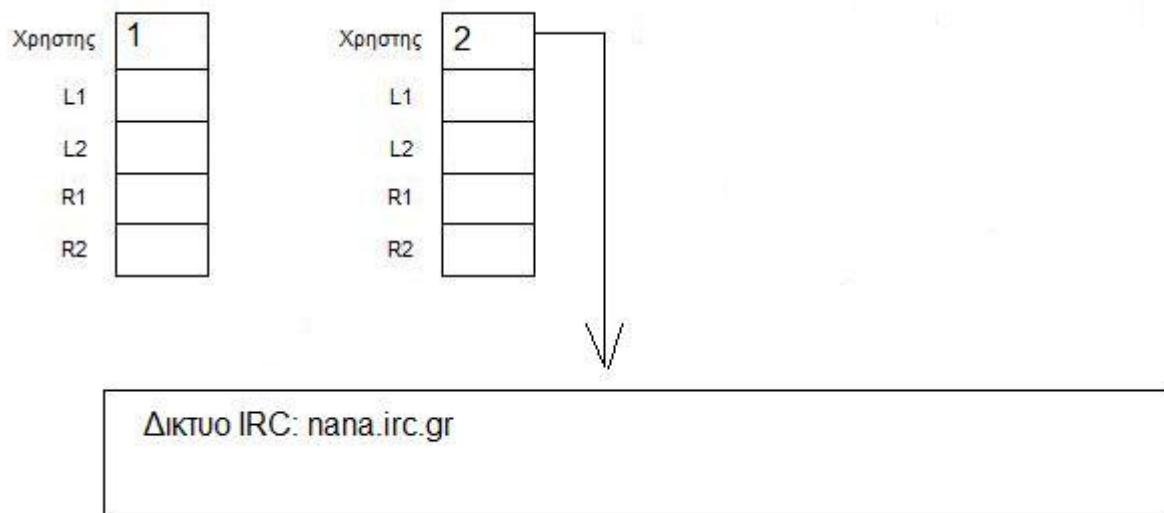


Συνδέεται στο δίκτυο nana.irc.gr και παίρνει το όνομα ircshare1. Στο δίκτυο θα μπορούν να τον δουν και κοινοί χρήστες, ακόμα και να του μιλήσουν. Το πρόγραμμα όμως καταλαβαίνει μόνο σε συγκεκριμένες εντολές οπότε οποιαδήποτε άλλη θα την αγνοεί.

Η μόνη εργασία που κάνει είναι να στέλνει στον server την εντολή PONG όταν λαμβάνει PING για να κρατήσει την σύνδεση ζωντανή.

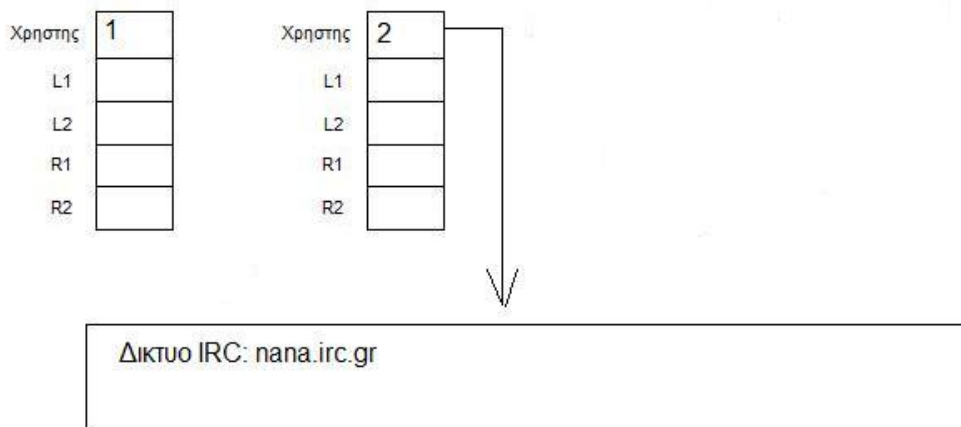
Ο χρήστης 1 περιμένει να συνδεθεί ο επόμενος χρήστης.

Δεύτερος χρήστης



Ο δεύτερος χρήστης συνδέεται στο ίδιο δίκτυο όπως και ο πρώτος.

Προσπαθεί να πάρει το όνομα `ircshare1` αλλά ο IRC server στέλνει μήνυμα ότι το όνομα είναι ήδη σε χρήση (in use). Αυτό το μήνυμα περνάει μέσα από την διαδικασία `ircCom` και ο χρήστης 2 αλλάζει το όνομα του σε `ircshare2`. Επίσης στέλνει μήνυμα στον `ircshare1` να βγει (`getout`).



Εδώ φαίνεται πως ο χρήστης 1 έχει βγει από το δίκτυο IRC.

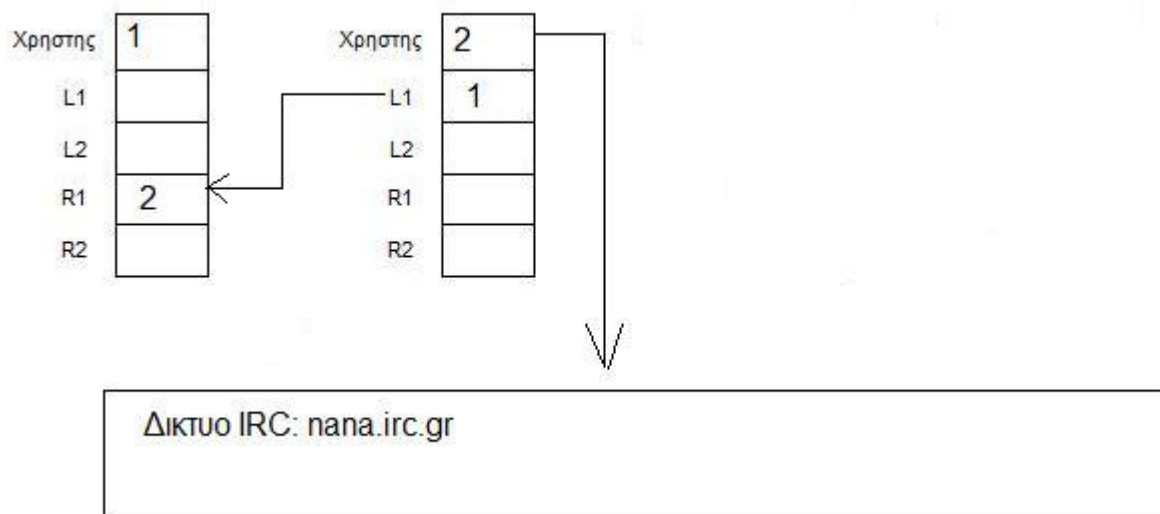
Ο χρήστης 1 λαμβάνει το μήνυμα και το περνάει μέσα από την διαδικασία ircCom οπότε στέλνει την IP διεύθυνση του στον χρήστη 2 και μετά από αυτό βγαίνει από το δίκτυο.

Ο χρήστης 2 λαμβάνει το μήνυμα του χρήστη 1 και φιλτράρει την διεύθυνση που του έστειλε. Με αυτήν την διεύθυνση θα δημιουργήσει σύνδεση με τον χρήστη 1.

Τη διεύθυνση που πήρε θα την καταχωρίσει στον πίνακα του στην θέση L1 και μετά θα πραγματοποιήσει την σύνδεση.

Μετά την δημιουργία της σύνδεσης θα στείλει μέσω του L1 την εντολή user μαζί με το όνομα του υπολογιστή.

Μετά σαν τελευταίο βήμα θα αλλάξει το όνομα του στο δίκτυο και θα το κάνει ircshare1 (αφού ο προηγούμενος ircshare1 έχει βγει από το δίκτυο).



Εδώ βλέπουμε τον χρήστη 2 να έχει δημιουργήσει μια σύνδεση με τον χρήστη 1.

Αυτό γίνεται στην ρουτίνα ircCom του χρήστη 2 αφού έχει πάρει πληροφορίες σχετικά με την διεύθυνση του χρήστη 1.

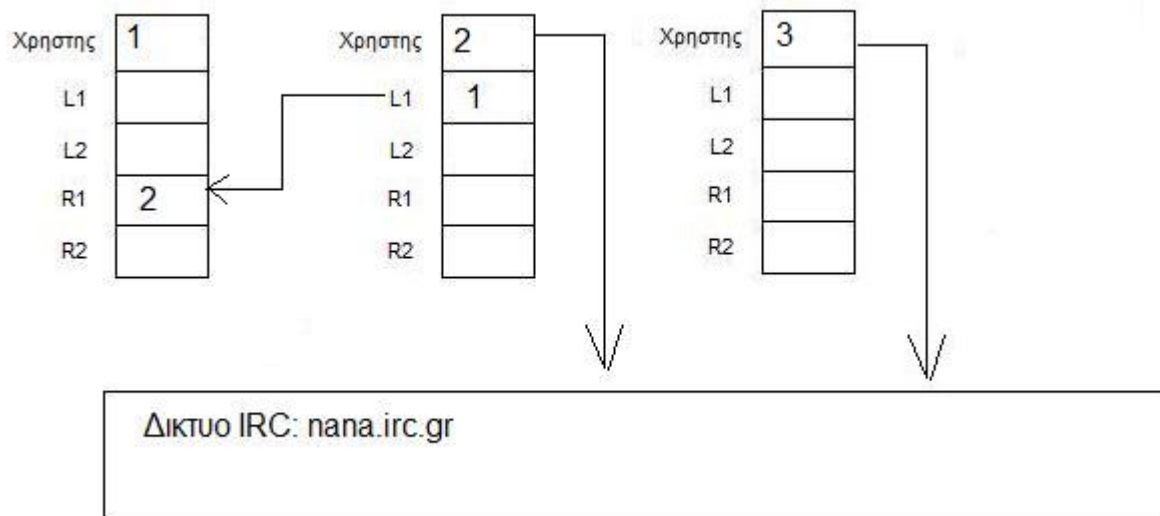
Με την σειρά του ο χρήστης 1 καταλαβαίνει connection request στο στοιχείο R1 αφού σε εκείνο γίνεται η σύνδεση και δέχεται την νέα σύνδεση. Με την αποδοχή της σύνδεσης ενημερώνονται η θέση R1 του πίνακα με την διεύθυνση του χρήστη 2 και τα διάφορα στοιχεία στη διεπαφή για να το καταλάβει ο χρήστης ότι έγινε σύνδεση.

Μετά την δημιουργία της σύνδεσης θα παραλάβει το μήνυμα που είχε στείλει προηγουμένως ο χρήστης 2 (το user) και θα καταχωρήσει το όνομα στο κατάλληλο στοιχείο για να φαίνεται στον χρήστη.

Επίσης με κάθε νέο connection request που συμβαίνει σε αυτό το στοιχείο το πρόγραμμα ψάχνει να δει αν έχει καταχώριση στην θέση L1 του πίνακα.

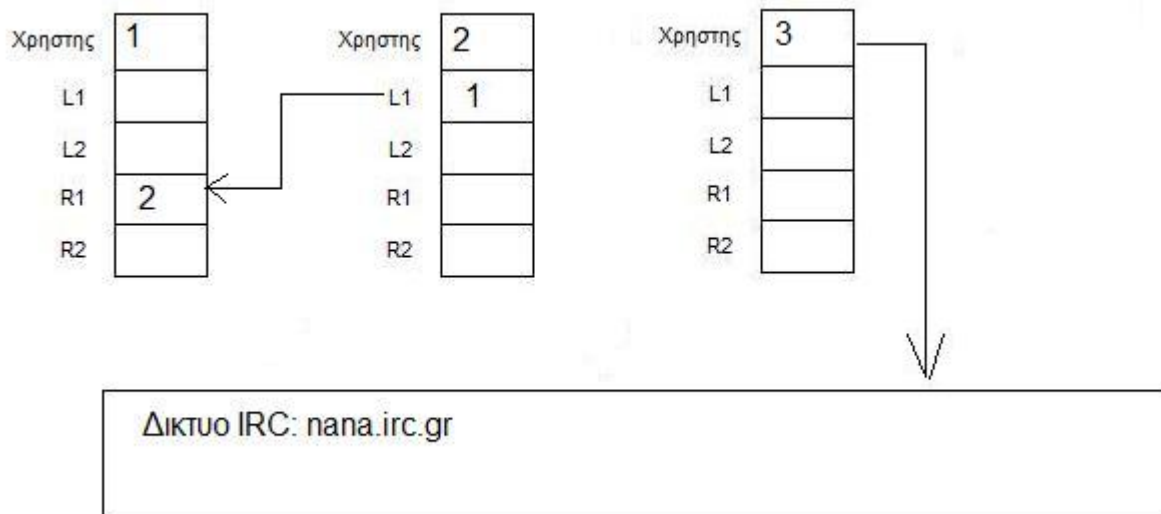
Αυτό όμως δεν συμβαίνει εδώ και θα αναλυθεί στα επόμενα βήματα.

Τρίτος χρήστης



Ο χρήστης 3 συνδέεται στο δίκτυο IRC.

Προσπαθεί να πάρει το όνομα `ircshare1` αλλά ο IRC server στέλνει μήνυμα ότι το όνομα είναι ήδη σε χρήση (in use). Αυτό το μήνυμα περνάει μέσα από την διαδικασία `ircCom` και ο χρήστης 2 αλλάζει το όνομα του σε `ircshare2`. Επίσης στέλνει μήνυμα στον `ircshare1` να βγει (`getout`).



Εδώ φαίνεται πως ο χρήστης 2 έχει βγει από το δίκτυο IRC

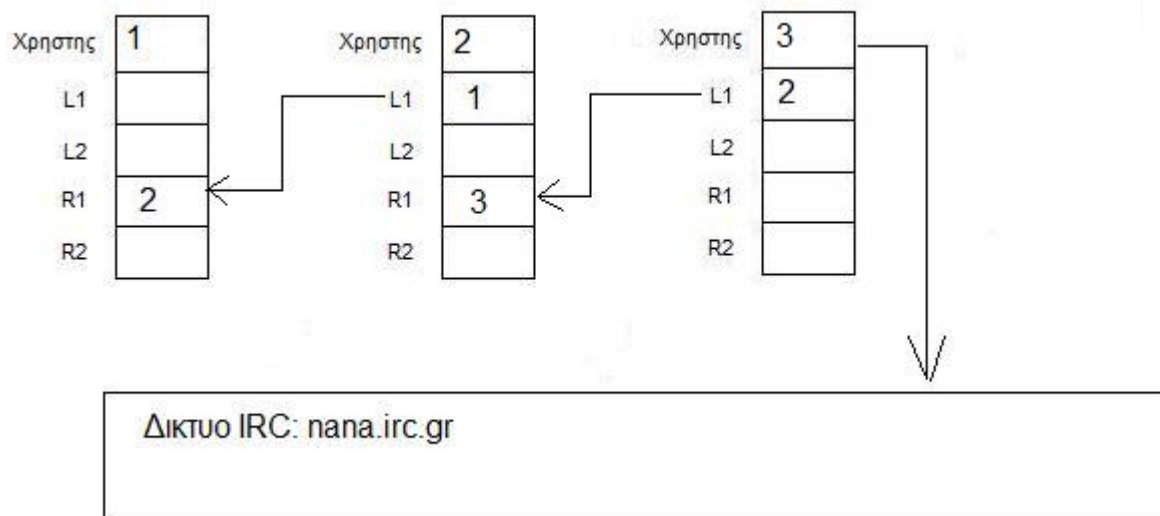
Ο χρήστης 2 λαμβάνει το μήνυμα και το περνάει μέσα από την διαδικασία ircCom οπότε στέλνει την IP διεύθυνση του στον χρήστη 3 και μετά από αυτό βγαίνει από το δίκτυο.

Ο χρήστης 3 λαμβάνει το μήνυμα του χρήστη 2 και φιλτράρει την διεύθυνση που του έστειλε. Με αυτήν την διεύθυνση θα δημιουργήσει σύνδεση με τον χρήστη 2.

Τη διεύθυνση που πήρε θα την καταχωρίσει στον πίνακα του στην θέση L1 και μετά θα πραγματοποιήσει την σύνδεση.

Μετά την δημιουργία της σύνδεσης θα στείλει μέσω του L1 την εντολή user μαζί με το όνομα του υπολογιστή.

Μετά σαν τελευταίο βήμα θα αλλάξει το όνομα του στο δίκτυο και θα το κάνει ircshare1 (αφού ο προηγούμενος ircshare1 έχει βγει από το δίκτυο).



Εδώ βλέπουμε τον χρήστη 3 να έχει δημιουργήσει μια σύνδεση με τον χρήστη 2.

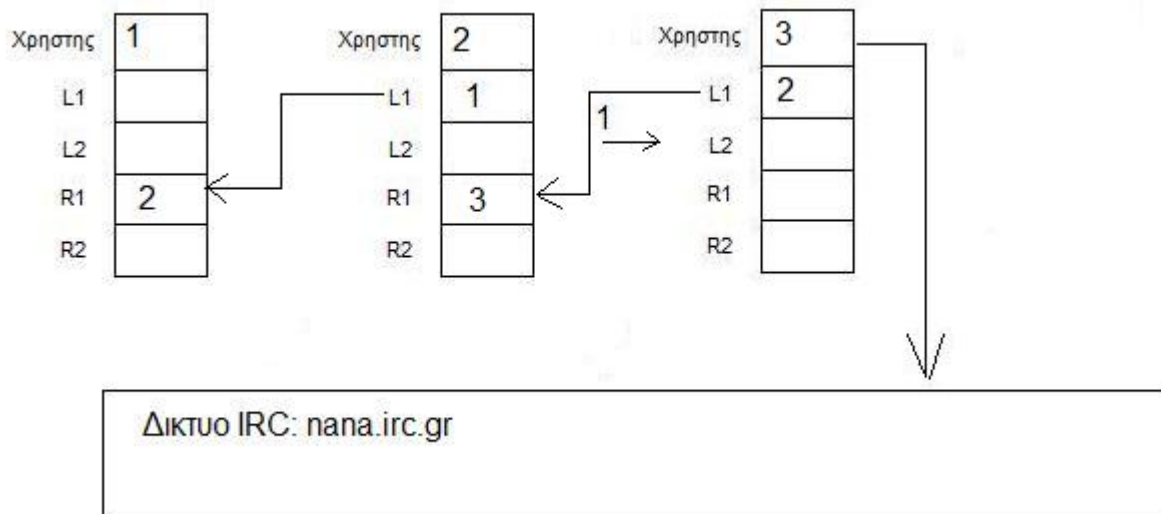
Αυτό γίνεται στην ρουτίνα ircCom του χρήστη 3 αφού έχει πάρει πληροφορίες σχετικά με την διεύθυνση του χρήστη 2.

Με την σειρά του ο χρήστης 2 καταλαβαίνει connection request στο στοιχείο R1 αφού σε εκείνο γίνεται η σύνδεση και δέχεται την νέα σύνδεση. Με την αποδοχή της σύνδεσης ενημερώνονται η θέση R1 του πίνακα με την διεύθυνση του χρήστη 3 και τα διάφορα στοιχεία στη διεπαφή για να το καταλάβει ο χρήστης ότι έγινε σύνδεση.

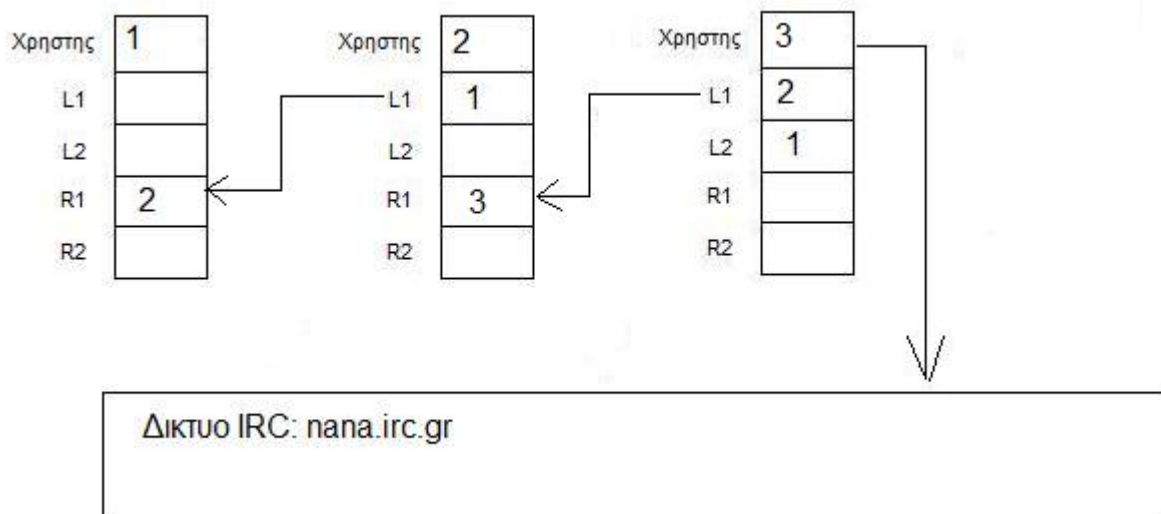
Μετά την δημιουργία της σύνδεσης θα παραλάβει το μήνυμα που είχε στείλει προηγουμένως ο χρήστης 2 (το user) και θα καταχωρήσει το όνομα στο κατάλληλο στοιχείο για να φαίνεται στον χρήστη.

Επίσης με κάθε νέο connection request που συμβαίνει σε αυτό το στοιχείο το πρόγραμμα ψάχνει να δει αν έχει καταχώριση στην θέση L1 του πίνακα. Σε αυτό το σημείο ο χρήστης 2 έχει καταχωρημένο στο L1 του την διεύθυνση του χρήστη 1.

Τώρα το πρόγραμμα θα στείλει αυτή τη καταχώριση στον χρήστη 3 μέσω της σύνδεσης που έχουν. Θα του στείλει την εντολή newip ώστε να καταλάβει ο 3 γιατί πρόκειται μαζί και την διεύθυνση που έχει καταχωρημένη στο L1.



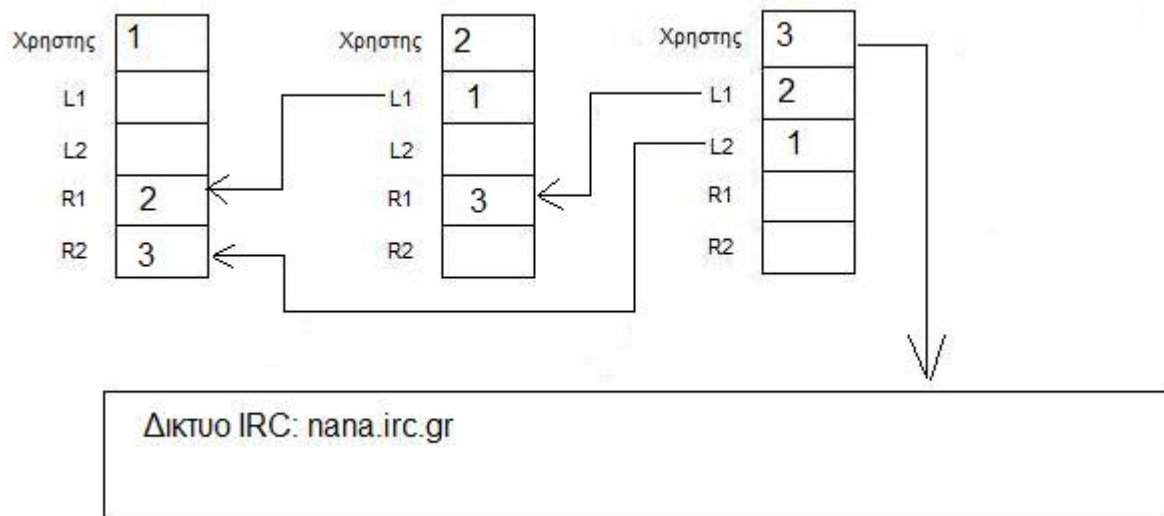
Ο χρήστης 2 στέλνει την διεύθυνση που είχε αποθηκευμένη στο L1 του



Και την καταχωρεί ο χρήστης 3 στο L2 του πίνακα του

Αφού ο χρήστης 3 έχει πλέον παραλάβει την διεύθυνση από τον χρήστη 2 και την έχει καταχωρήσει στο L2 του τώρα είναι η ώρα να δημιουργήσει την σύνδεση.

Έτσι λοιπόν συνδέεται στην συγκεκριμένη διεύθυνση μέσω του στοιχείου L2 και χρησιμοποιώντας την θύρα 4004. Αυτό σημαίνει ότι θα συνδεθεί στο R2 του χρήστη 1.

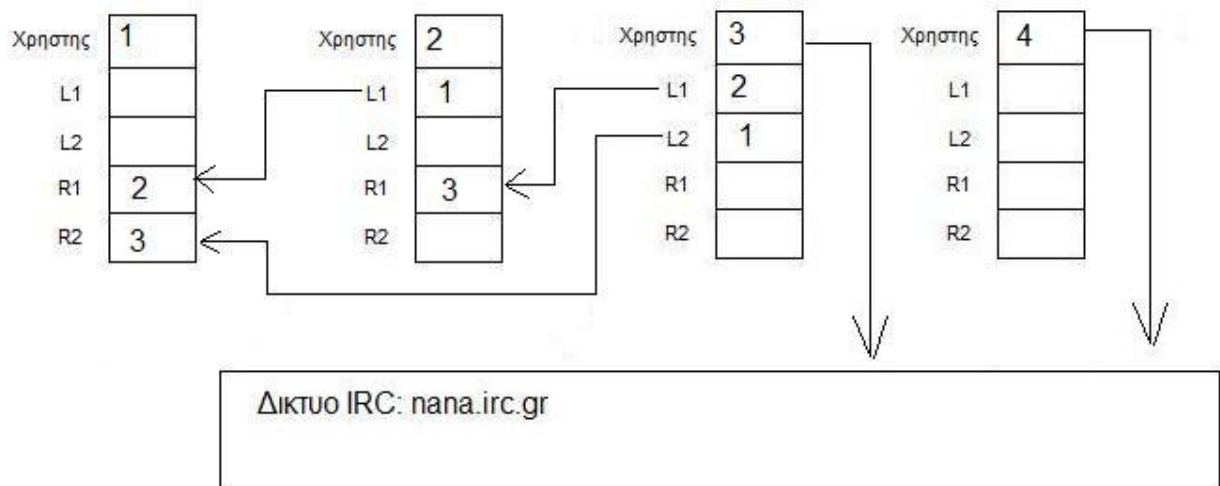


Παράλληλα με την σύνδεση ενημερώνονται και τα κατάλληλα στοιχεία για τον χρήστη ώστε να βλέπει τι γίνεται.

Με το που δημιουργεί την σύνδεση ο χρήστης 3, ο 1 την αποδέχεται μέσω του στοιχείου R2 που έχει και ενημερώνει τον πίνακα του. Επίσης ο 3 στέλνει μέσω αυτού του καναλιού μετάδοσης και το όνομα του υπολογιστή που τρέχει. Ο χρήστης 1 μετά την αποδοχή της σύνδεσης παραλαμβάνει και αυτό το μήνυμα και το καταχωρεί στα απαραίτητα στοιχεία για την ενημέρωση του χρήστη.

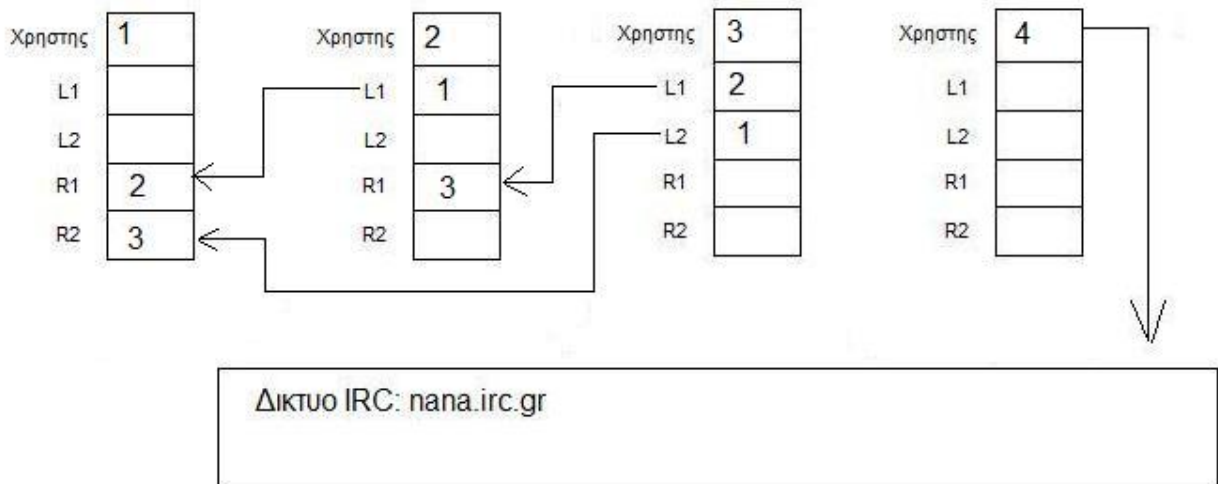
Τώρα έχουμε ένα δίκτυο με 3 υπολογιστές οι οποίοι είναι άμεσα συνδεδεμένοι μεταξύ τους. Μόνο ο χρήστης 3 είναι συνδεδεμένος στο δίκτυο IRC για να περιμένει τον επόμενο χρήστη που θα συνδεθεί.

Τέταρτος χρήστης



Ο χρήστης 4 συνδέεται στο δίκτυο IRC.

Προσπαθεί να πάρει το όνομα `ircshare1` αλλά ο IRC server στέλνει μήνυμα ότι το όνομα είναι ήδη σε χρήση (in use). Αυτό το μήνυμα περνάει μέσα από την διαδικασία `ircCom` και ο χρήστης 3 αλλάζει το όνομα του σε `ircshare2`. Επίσης στέλνει μήνυμα στον `ircshare1` να βγει (`getout`).



Εδώ φαίνεται πως ο χρήστης 3 έχει βγει από το δίκτυο IRC

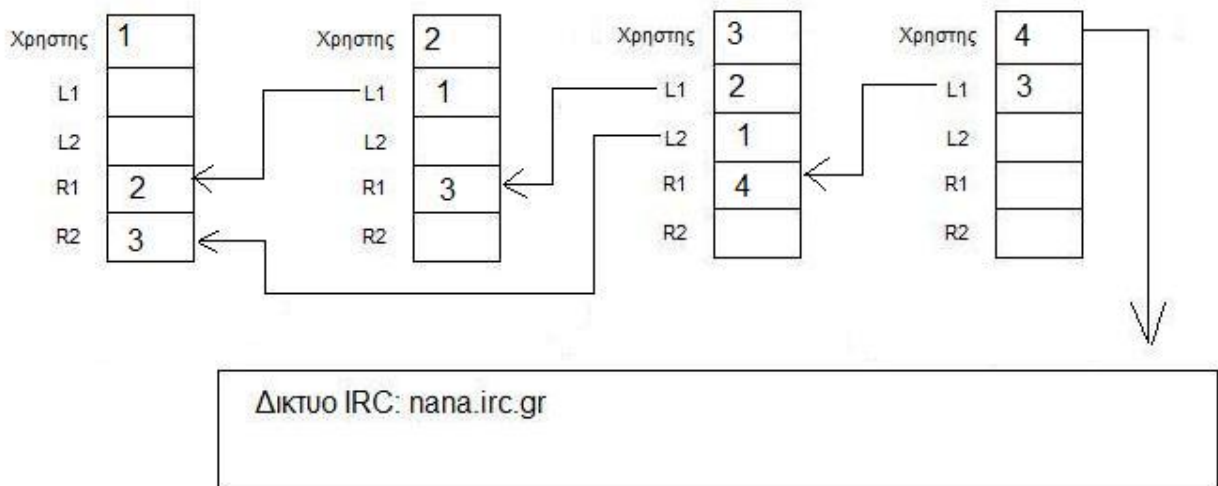
Ο χρήστης 3 λαμβάνει το μήνυμα και το περνάει μέσα από την διαδικασία ircCom οπότε στέλνει την IP διεύθυνση του στον χρήστη 4 και μετά από αυτό βγαίνει από το δίκτυο.

Ο χρήστης 4 λαμβάνει το μήνυμα του χρήστη 3 και φιλτράρει την διεύθυνση που του έστειλε. Με αυτήν την διεύθυνση θα δημιουργήσει σύνδεση με τον χρήστη 3.

Τη διεύθυνση που πήρε θα την καταχωρίσει στον πίνακα του στην θέση L1 και μετά θα πραγματοποιήσει την σύνδεση.

Μετά την δημιουργία της σύνδεσης θα στείλει μέσω του L1 την εντολή user μαζί με το όνομα του υπολογιστή.

Μετά σαν τελευταίο βήμα θα αλλάξει το όνομα του στο δίκτυο και θα το κάνει ircshare1 (αφού ο προηγούμενος ircshare1 έχει βγει από το δίκτυο).



Εδώ βλέπουμε τον χρήστη 4 να έχει δημιουργήσει μια σύνδεση με τον χρήστη 3.

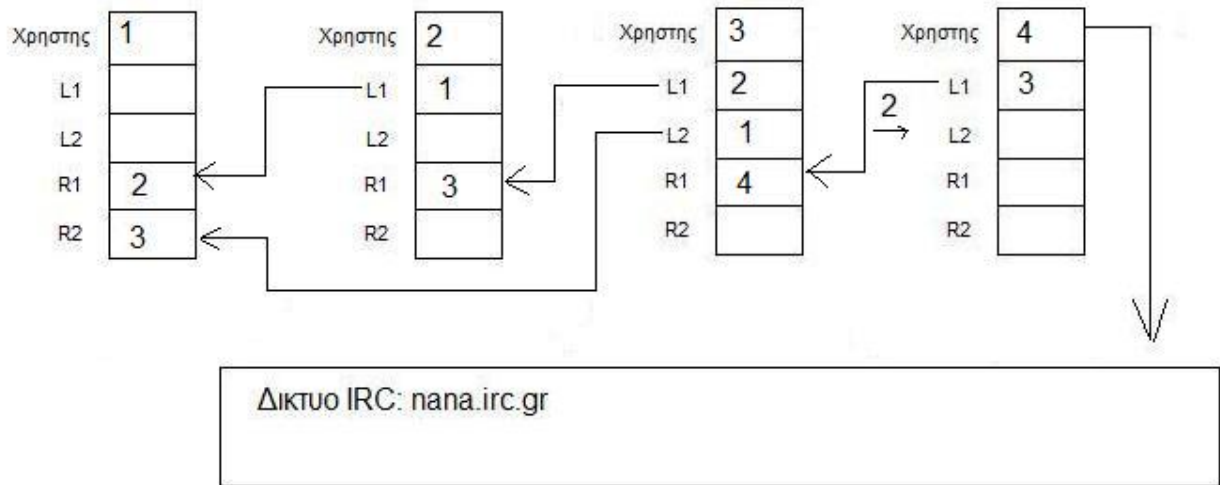
Αυτό γίνεται στην ρουτίνα ircCom του χρήστη 3 αφού έχει πάρει πληροφορίες σχετικά με την διεύθυνση του χρήστη 2.

Με την σειρά του ο χρήστης 3 καταλαβαίνει connection request στο στοιχείο R1 αφού σε εκείνο γίνεται η σύνδεση και δέχεται την νέα σύνδεση. Με την αποδοχή της σύνδεσης ενημερώνονται η θέση R1 του πίνακα με την διεύθυνση του χρήστη 4 και τα διάφορα στοιχεία στη διεπαφή για να το καταλάβει ο χρήστης ότι έγινε σύνδεση.

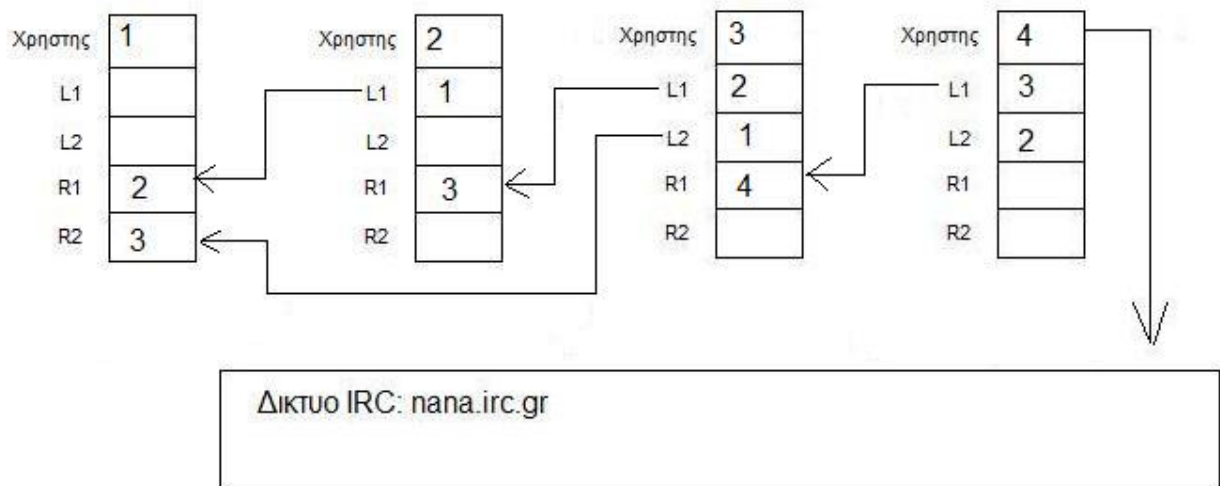
Μετά την δημιουργία της σύνδεσης θα παραλάβει το μήνυμα που είχε στείλει προηγουμένως ο χρήστης 3 (το user) και θα καταχωρήσει το όνομα στο κατάλληλο στοιχείο για να φαίνεται στον χρήστη.

Επίσης με κάθε νέο connection request που συμβαίνει σε αυτό το στοιχείο το πρόγραμμα ψάχνει να δει αν έχει καταχώριση στην θέση L1 του πίνακα. Σε αυτό το σημείο ο χρήστης 3 έχει καταχωρημένο στο L1 του την διεύθυνση του χρήστη 2.

Τώρα το πρόγραμμα θα στείλει αυτή τη καταχώριση στον χρήστη 4 μέσω της σύνδεσης που έχουν. Θα του στείλει την εντολή newip ώστε να καταλάβει ο 4 γιατί πρόκειται μαζί και την διεύθυνση που έχει καταχωρημένη στο L1.



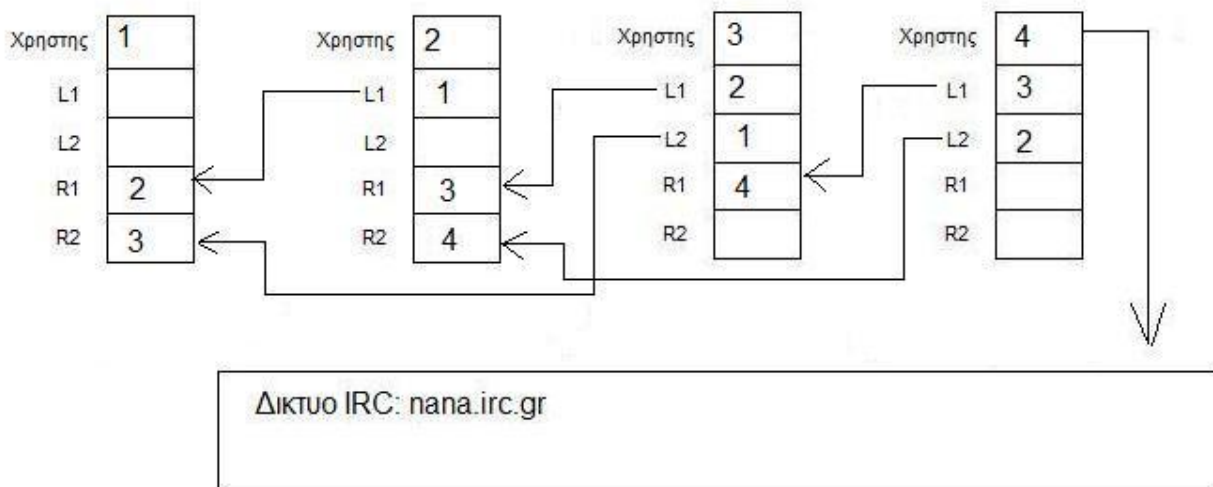
Ο χρήστης 3 στέλνει την διεύθυνση που είχε αποθηκευμένη στο L1 του



Και την καταχωρεί ο χρήστης 4 στο L2 του πίνακα του

Αφού ο χρήστης 4 έχει πλέον παραλάβει την διεύθυνση από τον χρήστη 3 και την έχει καταχωρήσει στο L2 του τώρα είναι η ώρα να δημιουργήσει την σύνδεση.

Έτσι λοιπόν συνδέεται στην συγκεκριμένη διεύθυνση μέσω του στοιχείου L2 και χρησιμοποιώντας την θύρα 4004. Αυτό σημαίνει ότι θα συνδεθεί στο R2 του χρήστη 2.



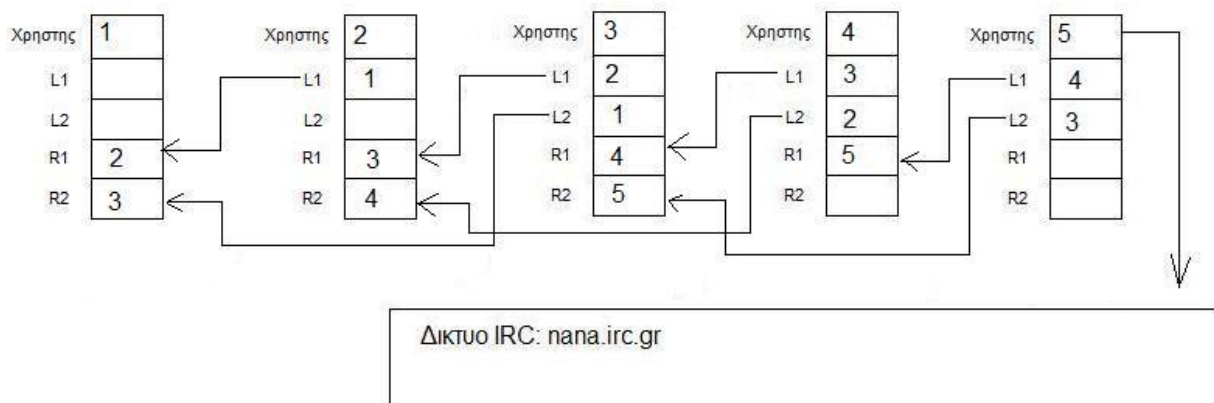
Παράλληλα με την σύνδεση ενημερώνονται και τα κατάλληλα στοιχεία για τον χρήστη ώστε να βλέπει τι γίνεται.

Με το που δημιουργεί την σύνδεση ο χρήστης 4, ο 2 την αποδέχεται μέσω του στοιχείου R2 που έχει και ενημερώνει τον πίνακα του. Επίσης ο 4 στέλνει μέσω αυτού του καναλιού μετάδοσης και το όνομα του υπολογιστή που τρέχει. Ο χρήστης 2 μετά την αποδοχή της σύνδεσης παραλαμβάνει και αυτό το μήνυμα και το καταχωρεί στα απαραίτητα στοιχεία για την ενημέρωση του χρήστη.

Τώρα έχουμε ένα δίκτυο με 4 υπολογιστές οι οποίοι είναι άμεσα συνδεδεμένοι μεταξύ τους. Μόνο ο χρήστης 4 είναι συνδεδεμένος στο δίκτυο IRC για να περιμένει τον επόμενο χρήστη που θα συνδεθεί.

Πέμπτος χρήστης

Για τον πέμπτο και τελευταίο χρήστη θα γίνουν ακριβώς οι ίδιες διαδικασίες με τον τέταρτο. Έτσι έχουμε το δίκτυο που βλέπουμε παρακάτω



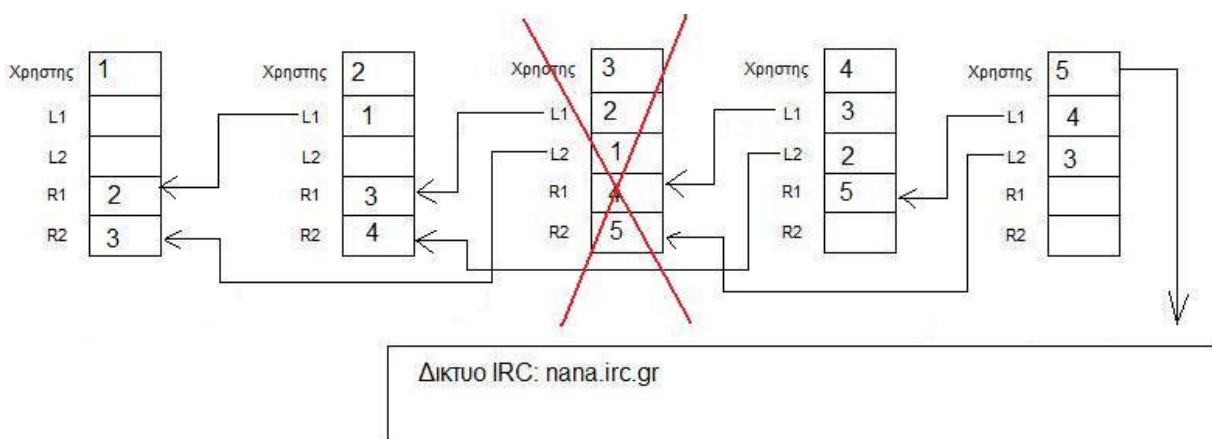
Τώρα έχουμε δημιουργήσει ένα δίκτυο με πέντε υπολογιστές από τους οποίους ο χρήστης 5 είναι συνδεδεμένος στο δίκτυο IRC για να περιμένει τον επόμενο ώστε να τον συνδέσει στο δίκτυο.

5.5 Ανάκτηση συνδέσεων

Τώρα θα δούμε τι γίνεται στην περίπτωση που κάποιος χρήστης βγει από το δίκτυο είτε με την θέληση του είτε απότομα. Το δίκτυο έχει τις κατάλληλες συνδέσεις ώστε να μην καταρρέει σε τέτοια περίπτωση.

Στο επόμενο παράδειγμα θα δείξω πως γίνεται αυτή η ανάκτηση των συνδέσεων με την απομάκρυνση ενός από τους πέντε χρήστες.

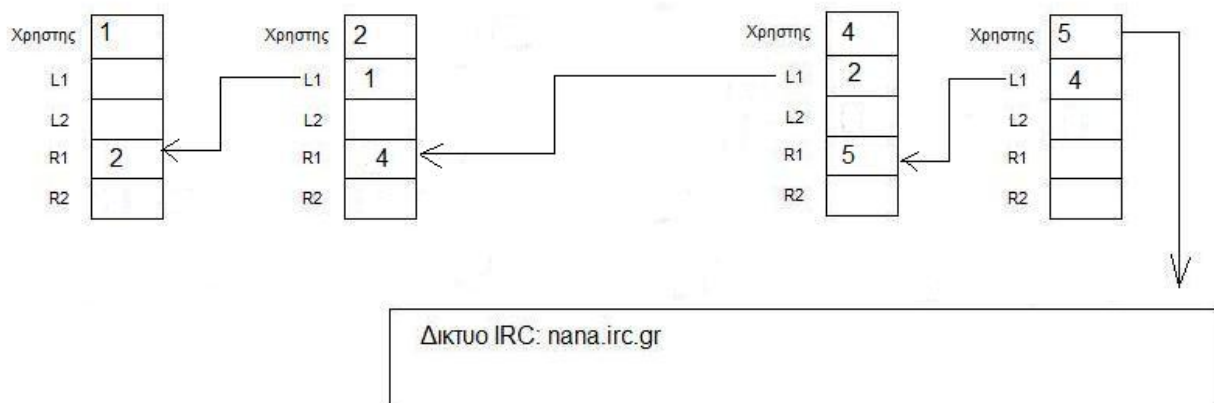
Θα υποθέσουμε ότι κόβεται το η παροχή ρεύματος από τον τρίτο χρήστη ο οποίος βρίσκεται ακριβώς στην μέση του δικτύου και μετά θα δείξω πως αναδημιουργούνται οι συνδέσεις αυτόματα.



Εδώ βλέπουμε ότι ο χρήστης 3 βγήκε από το δίκτυο εντελώς απρόσμενα.

Το πρώτο πράγμα που θα συμβεί είναι στον χρήστη 4 στο στοιχείο του L1 και θα καταλάβει

connection lost. Ελέγχει αν έχει καταχώρηση στο L2 του. Στην περίπτωση μας έχει που σημαίνει ότι είναι συνδεδεμένος με τον δεύτερο αριστερό του. Αυτό που κάνει λοιπόν είναι να μεταφέρει το περιεχόμενο του L2 στο L1 αντικαθιστώντας το παλιό. Επόμενο βήμα είναι να επαναφέρει το στοιχείο L2 σε κατάσταση ακρόασης και να δημιουργήσει μια νέα σύνδεση στο L1 με την καινούρια του διεύθυνση.

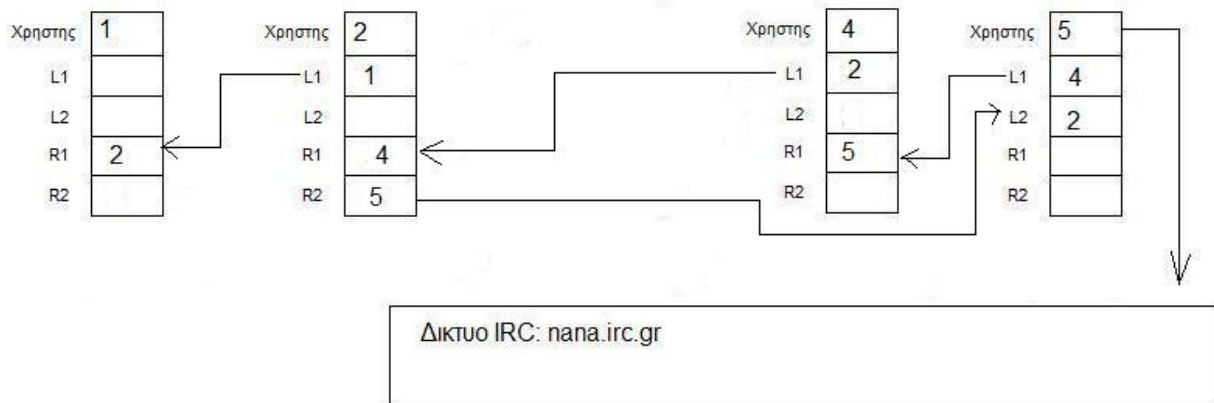


Έπειτα ελέγχει αν διαθέτει (ο χρήστης 4) στην θέση R1 κάποιον χρήστη. Στην περίπτωση μας έχει τον 5 οπότε θα στείλει μήνυμα μέσω του καναλιού L1 (δηλαδή στον χρήστη 2) το μήνυμα lost μαζί με την διεύθυνση που έχει στο R1 του, δηλαδή το 2.

Αν δεν διαθέτει ο χρήστης 4 κάποιον χρήστη στην θέση της λίστας L2, τότε πολύ απλά σημαίνει ότι ο χρήστης που βγήκε από το δίκτυο ήταν ο πρώτος του δικτύου και ο τρέχων παίρνει την θέση του.

Με τη σειρά του ο χρήστης 2 παραλαμβάνει το μήνυμα που έστειλε ο 4 μέσω του καναλιού R1. Με το που δέχεται την εντολή την φιλτράρει και καταχωρεί την διεύθυνση (του 5 δηλαδή) στον πίνακα στην θέση R2.

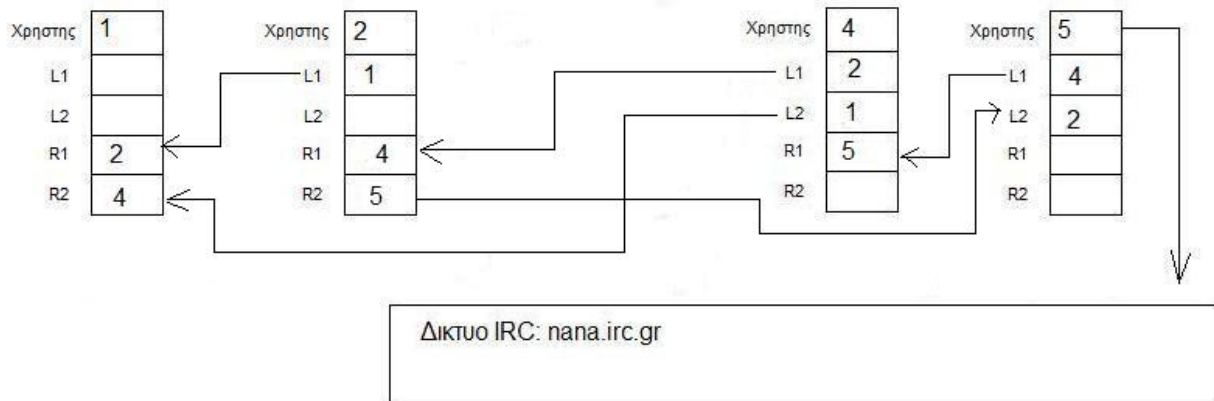
Έπειτα δημιουργεί μια σύνδεση μέσω R2 με τον χρήστη 5.



Και συνδέεται στον στοιχείο L2 του χρήστη 5.

Ο χρήστης 5 με την σειρά του δέχεται την σύνδεση και ανανεώνει τα διάφορα στοιχεία της διεπαφής.

Επίσης ο χρήστης 2 με το που πραγματοποιήθηκε η καινούρια σύνδεση από τον χρήστη 4 στον στοιχείο R1 του, ελέγχει αν έχει καταχωρημένο στην λίστα του στη θέση L1 κάποια διεύθυνση. Στην περίπτωση μας έχει το 1 οπότε το στέλνει μέσω του στοιχείου R1 στον χρήστη 4 με μήνυμα newip.



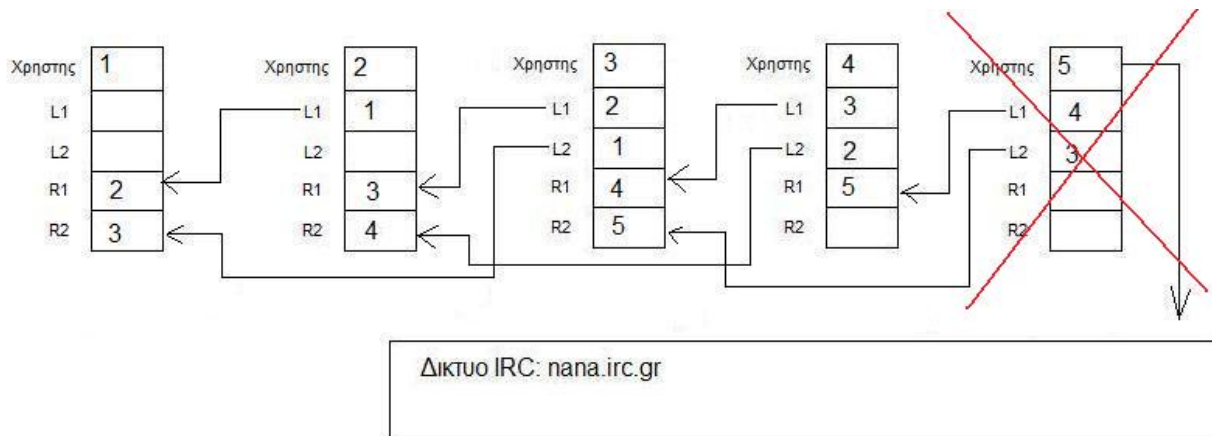
Ο χρήστης 4 λαμβάνει το μήνυμα που έστειλε ο 2 και το καταχωρεί στον πίνακα στην θέση L2.

Έπειτα δημιουργεί την σύνδεση με τον χρήστη 1 στο στοιχείο R2.

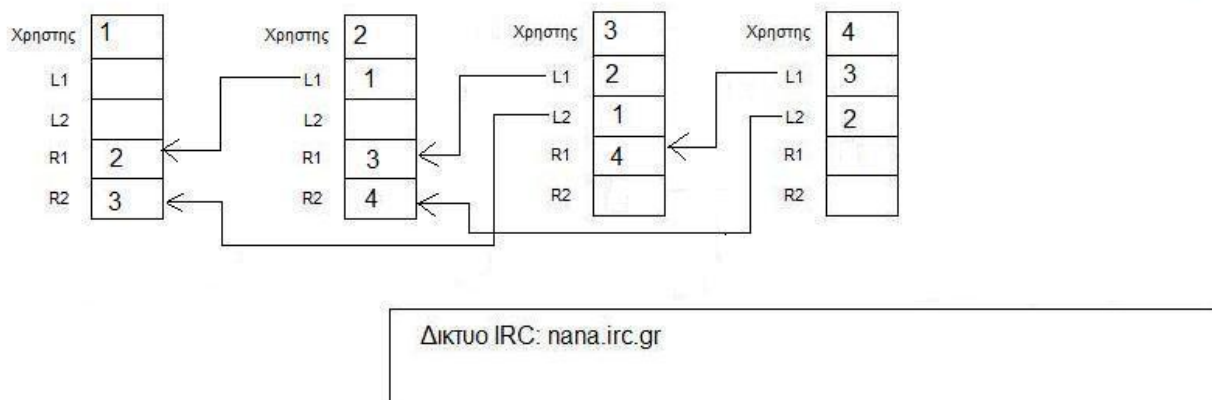
Ο χρήστης 1 με την σειρά του αποδέχεται την σύνδεση και ανανεώνει τα κατάλληλα στοιχεία για την ενημέρωση του χρήστη.

Με αυτό το τελευταίο βήμα τελειώσε η αποκατάσταση συνδέσεων. Μπορεί να φαίνονται πολλά τα βήματα αλλά γίνονται μέσα σε ένα δευτερόλεπτο.

Παρακάτω θα δούμε τι γίνεται αν βγει από το δίκτυο ο τελευταίος χρήστης, δηλαδή αυτός που είναι συνέχεια συνδεδεμένος με το δίκτυο IRC.



Προφανώς ο προτελευταίος θα πάρει την θέση του.



Τώρα έχουμε ένα δίκτυο με τέσσερις υπολογιστές το οποίο όμως βρίσκεται στον αέρα και δεν μπορεί να συνδεθεί νέος χρήστης. Και αυτό γιατί δεν υπάρχει κάποιος στο δίκτυο IRC να περιμένει τον επόμενο.

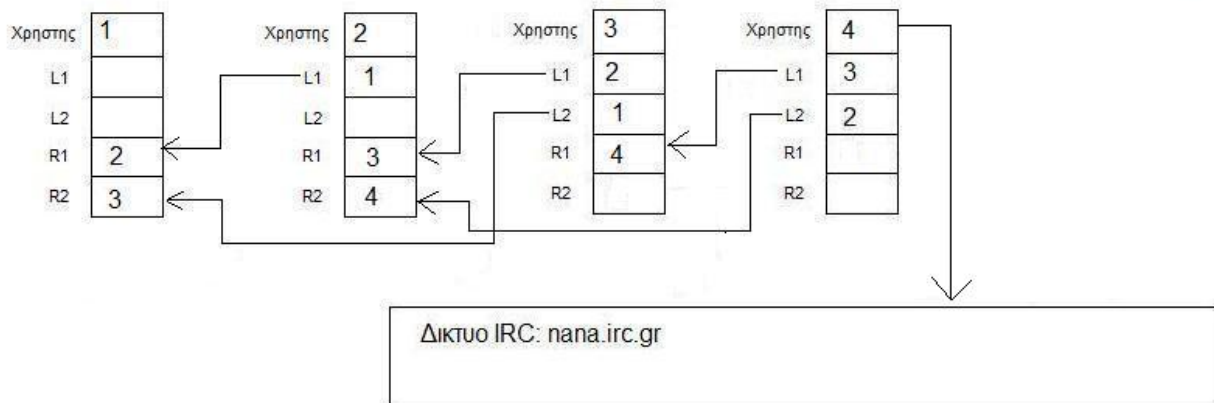
Ο χρήστης 3 καταλαβαίνει στο στοιχείο R2 connection lost. Έτσι θέτεται σε κατάσταση αναμονής και ενημερώνει τον χρήστη μέσω της διεπαφής κατάλληλα.

Ο χρήστης 4 καταλαβαίνει connection lost στο στοιχείο R1.

Το στοιχείο μπαίνει αμέσως σε κατάσταση αναμονής και ενημερώνει τον χρήστη κατάλληλα μέσω της διεπαφής.

Επίσης αυτό που είναι και το πιο βασικό είναι ότι ελέγχει αν έχει καταχώριση στο R2 του. Στην περίπτωση μας δεν έχει που σημαίνει ότι ήταν ο προτελευταίος χρήστης. Και αυτό γιατί ξέρει ότι είναι συνδεδεμένος μόνο με μια θέση δεξιά και όχι δύο. Άρα ο δεξιός του είναι και ο τελευταίος.

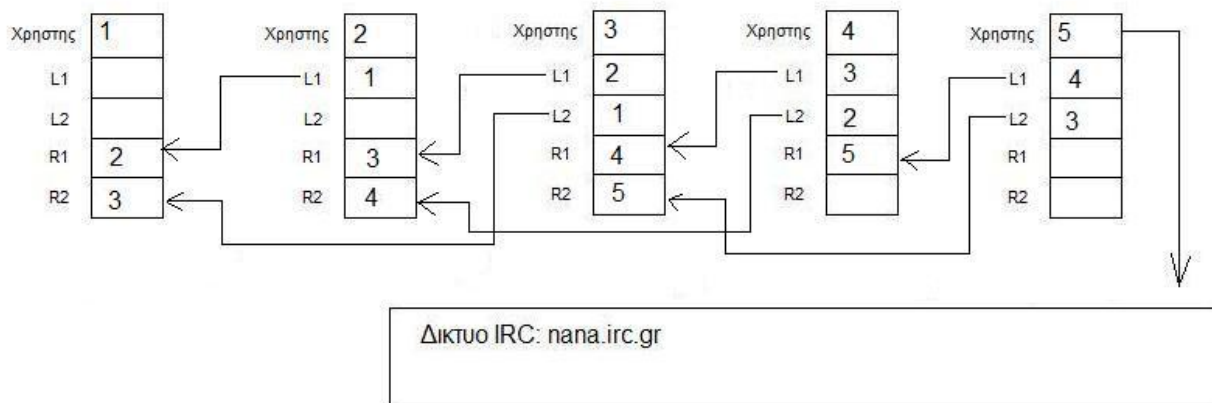
Σε αυτή την περίπτωση ενεργοποιείτε ξανά η διαδικασία σύνδεσης στο δίκτυο IRC μέσω του χρήστη 4. Αυτή η διαδικασία γίνεται πολύ απλά πατώντας το κουμπί connect to server αυτόματα.



5.6 Αναζήτηση

Το πρόγραμμα διαθέτει διαδικασία αναζήτησης η οποία θα περιγραφεί αναλυτικά παρακάτω. Η αναζήτηση μπορεί να πραγματοποιηθεί με δύο ή περισσότερους υπολογιστές στο δίκτυο.

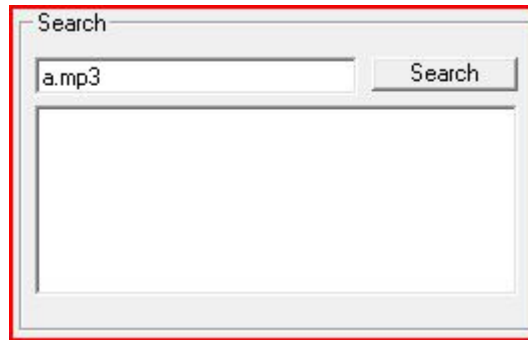
Θα υποθέσουμε πως έχουμε το παρακάτω δίκτυο αποτελούμενο από 5 υπολογιστές όπως και στο προηγούμενο παράδειγμα.



Την αναζήτηση μπορεί να την προκαλέσει οποιοσδήποτε χρήστης από τους πέντε αρκεί να είναι συνδεδεμένος στο δίκτυο.

Στο παράδειγμα μας θα την κάνει ο 2.

Για να γίνει η αναζήτηση ο χρήστης θα πρέπει να εισάγει στο κατάλληλο πεδίο το ακριβές όνομα αρχείου που επιθυμεί να ψάξει.



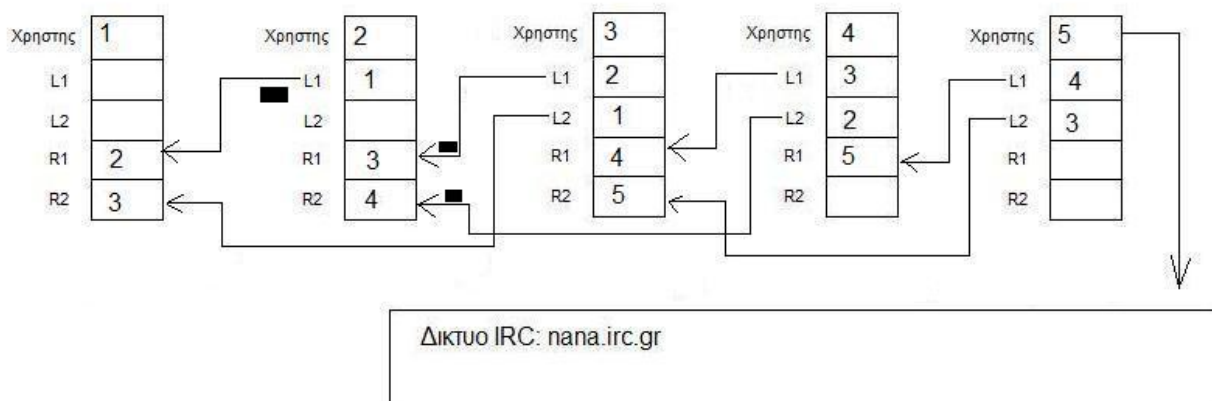
Στο παράδειγμα μας ξεκινάμε την αναζήτηση με την λέξη κλειδί a.mp3
Για να ξεκινήσει θα πρέπει ο χρήστης να πατήσει το κουμπί Search.

Το πρώτο πράγμα που θα ελέγξει το πρόγραμμα είναι να δει ποιους χρήστες έχει καταχωρημένους στην λίστα του.

Ο χρήστης 2 έχει τον L1,R1 και R2.

Οπότε θα στείλει σε αυτούς τους χρήστες το μήνυμα για αναζήτηση.

Το μήνυμα θα αποτελείτε από το εξής: την λέξη QUERY , την τοπική διεύθυνση και μετά την λέξη προς αναζήτηση.



Εδώ φαίνεται με μαύρα τετραγωνάκια τα πακέτα αναζήτησης που στέλνει ο χρήστης 2. Τα πακέτα αυτά θα τα παραλάβει ο χρήστης 1, 3 και 4 στα αντίστοιχα στοιχεία τους.

Την αναζήτηση θα την παραλάβει ο χρήστης 1 από το στοιχείο R1 και θα την περάσει μέσα από if για να διαπιστωθεί το περιεχόμενο της. Μετά ελέγχει να δει αν είναι διαθέσιμος από πλευρά συνδέσεων, δηλαδή αν εκείνη τη χρονική στιγμή στέλνει κάποιο αρχείο ή αν λαμβάνει αποτελέσματα αναζήτησης από κάποιον άλλο χρήστη.

Στη περίπτωση που είναι διαθέσιμος, τότε φιλτράρει την εντολή και την χωρίζει σε δύο μέρη, ένα με την διεύθυνση αποστολέα και το άλλο με το περιεχόμενο προς αναζήτηση. Παίρνει λοιπόν το περιεχόμενο και αναζητάει στην λίστα αρχείων που έχει να το βρει. Στην περίπτωση που βρεθεί το αρχείο στην λίστα, τότε δημιουργεί μια απευθείας σύνδεση με τον αποστολέα της αναζήτησης και τον ενημερώνει για την διάθεση του και αμέσως μετά την τερματίζει για να ελευθερώσει το κανάλι στους υπόλοιπους χρήστες.

Από την μεριά του ο χρήστης 2 δέχεται την σύνδεση και καταχωρεί στο πεδίο αποτελεσμάτων το αρχείο και την διεύθυνση του αποστολέα.

Αφού τελειώσει η αναζήτηση στον τρέχων χρήστη τότε κοιτάει να δει αν έχει καταχώρηση στην λίστα του για τον L2. Στην περίπτωση που έχει τότε προωθεί το πακέτο σε αυτόν τον χρήστη.

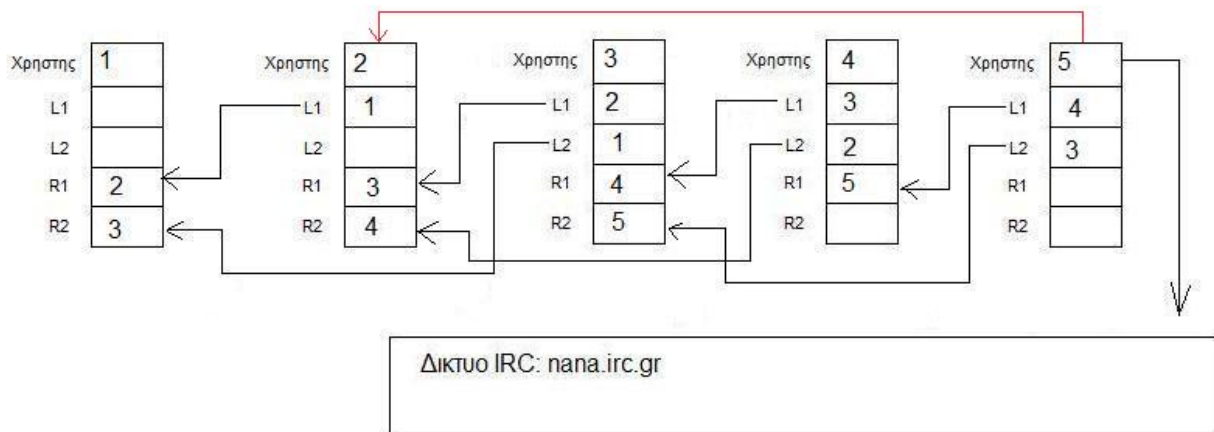
Θα υποθέσουμε ότι το αρχείο το έχει ο χρήστης 5.

Η διαδικασία της αναζήτησης που περιγράφηκε προηγουμένως θα πραγματοποιηθεί με τον ακριβώς ίδιο τρόπο και στους χρήστες 3 και 4. Η μόνη διαφορά είναι ότι στο τέλος για την προώθηση της αναζήτησης δεν θα κοιτάξουν στον πίνακα για τον L2 αλλά για τον R2. Αυτό γίνεται γιατί η προώθηση προχωράει προς τα δεξιά σε αυτή τη φάση. Αυτό που μας ενδιαφέρει εδώ είναι η προώθηση του πακέτου από τον χρήστη 3 προς τον 5 μέσω του καναλιού R2 που είναι συνδεδεμένοι.

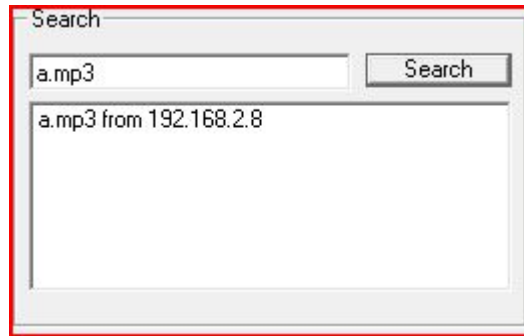
Την αναζήτηση θα την παραλάβει ο χρήστης 5 από το στοιχείο L2 και θα την περάσει μέσα από if για να διαπιστωθεί το περιεχόμενο της. Μετά ελέγχει να δει αν είναι διαθέσιμος από πλευρά συνδέσεων, δηλαδή αν εκείνη τη χρονική στιγμή στέλνει κάποιο αρχείο ή αν λαμβάνει αποτελέσματα αναζήτησης από κάποιον άλλο χρήστη.

Στη περίπτωση που είναι διαθέσιμος, τότε φιλτράρει την εντολή και την χωρίζει σε δύο μέρη, ένα με την διεύθυνση αποστολέα και το άλλο με το περιεχόμενο προς αναζήτηση. Παίρνει λοιπόν το περιεχόμενο και αναζητάει στην λίστα αρχείων που έχει να το βρει. Στην περίπτωση που βρεθεί το αρχείο στην λίστα, τότε δημιουργεί μια απευθείας σύνδεση με τον αποστολέα της αναζήτησης και τον ενημερώνει για την διάθεση του και αμέσως μετά την τερματίζει για να ελευθερώσει το κανάλι στους υπόλοιπους χρήστες.

Από την μεριά του ο χρήστης 2 δέχεται την σύνδεση και καταχωρεί στο πεδίο αποτελεσμάτων το αρχείο και την διεύθυνση του αποστολέα.



Αφού τελειώσει η αναζήτηση στον τρέχων χρήστη τότε κοιτάει να δει αν έχει καταχώρηση στην λίστα του για τον L2. Στην περίπτωση που έχει τότε προωθεί το πακέτο σε αυτόν τον χρήστη.



Εδώ βλέπουμε το αποτέλεσμα αναζήτησης στον χρήστη 2. Το αριστερό μέρος αποτελείται από το όνομα του αρχείου που ήταν προς εύρεση και το δεξί η διεύθυνση του χρήστη ο οποίος διαθέτει το αρχείο.

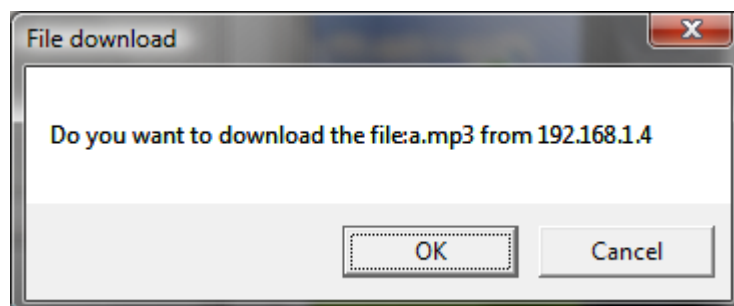
Στην αρχή της λίστας τοποθετείτε ο χρήστης που είχε την πιο γρήγορη σύνδεση με τον χρήστη 2 στην αναζήτηση. Έτσι δίνεται η δυνατότητα να επιλέξουμε την ταχύτερη σύνδεση για μεταφορά του αρχείου.

Τώρα είναι στο χέρι του χρήστη 2 αν θα κατεβάσει αυτό το αρχείο ή όχι από τον συγκεκριμένο χρήστη.

5.7 Μεταφορά αρχείων

Αφού ολοκληρωθεί η διαδικασία αναζήτησης σειρά έχει η μεταφορά αυτού του αρχείου. Ο χρήστης που πραγματοποίησε την αναζήτηση στο προηγούμενο βήμα, δηλαδή ο 2, τώρα έχει την επιλογή να μεταφέρει αυτό το αρχείο από τον απομακρυσμένο υπολογιστή που βρέθηκε, στον δικό του. Αυτό ενεργοποιείτε με διπλό κλικ πάνω στο όνομα του αρχείου. Στα παράδειγμα μας θα μεταφέρουμε το αρχείο με το όνομα a.mp3 από τον υπολογιστή του χρήστη 5 στον υπολογιστή του χρήστη 2.

Όταν ο χρήστης 2 πατήσει πάνω στο όνομα του αρχείου στην λίστα αναζήτησης 2 φορές τότε θα εμφανιστεί το εξής μήνυμα στον χρήστη:



Το πρόγραμμα θα συνεχίσει ανάλογα με την επιλογή του χρήστη. Στην περίπτωση μας θα πατήσει OK ώστε να συνεχίσει η διαδικασία μεταφοράς.

Αμέσως το πρόγραμμα παίρνει την διεύθυνση IP που βρίσκεται δίπλα από το όνομα αρχείου και δημιουργεί μια απευθείας σύνδεση με αυτήν στην θύρα 4005. Επίσης θα θέσει την μεταβλητή down σε κατάσταση true. Με το που ζητηθεί connection request στην μεριά του χρήστη 5 θα το δεχτεί αμέσως. Όταν πραγματοποιηθεί η σύνδεση τότε ο χρήστης 2 θα δει ότι η μεταβλητή down είναι true και θα στείλει στον χρήστη 5 την εντολή GIVE μαζί με το όνομα αρχείου που είναι για μεταφορά.

Ο χρήστης 5 από την μεριά του θα παραλάβει το μήνυμα του 2 δηλαδή το GIVE και θα φιλτράρει την εντολή. Μετά θα στείλει πίσω στον χρήστη 2 το μήνυμα START μαζί με το μέγεθος του αρχείου. Επίσης θα καταχωρίσει σε μια μεταβλητή το μέγεθος του αρχείου και θα προσθέσει το όνομα του στην λίστα uploading.

Επόμενο στάδιο είναι ότι θα ανοίξει το αρχείο σε κατάσταση δυαδικής ανάγνωσης και θα αρχίσει να διαβάζει κομμάτια των 4096 bytes σε μια προσωρινή μνήμη και θα τα στέλνει στον χρήστη 2. Αυτή η διαδικασία θα γίνει έως ότου διαβαστεί όλο το αρχείο. Σε κάθε κομμάτι που διαβάζεται μια μεταβλητή μετράει όλα τα bytes ώστε να ενημερώνει τον χρήστη ανάλογα.

Όταν θα τελειώσει με όλα τα κομμάτια τότε το πρόγραμμα θα στείλει το μήνυμα EOF στον χρήστη 2 που σημαίνει End Of File δηλαδή τέλειωσε η αποστολή του αρχείου. Σε αυτή την φάση κλείνουν οι συνδέσεις, καθαρίζουν οι λίστες και μπαίνει σε κατάσταση αναμονής.

Τώρα από τη μεριά του χρήστη 2.

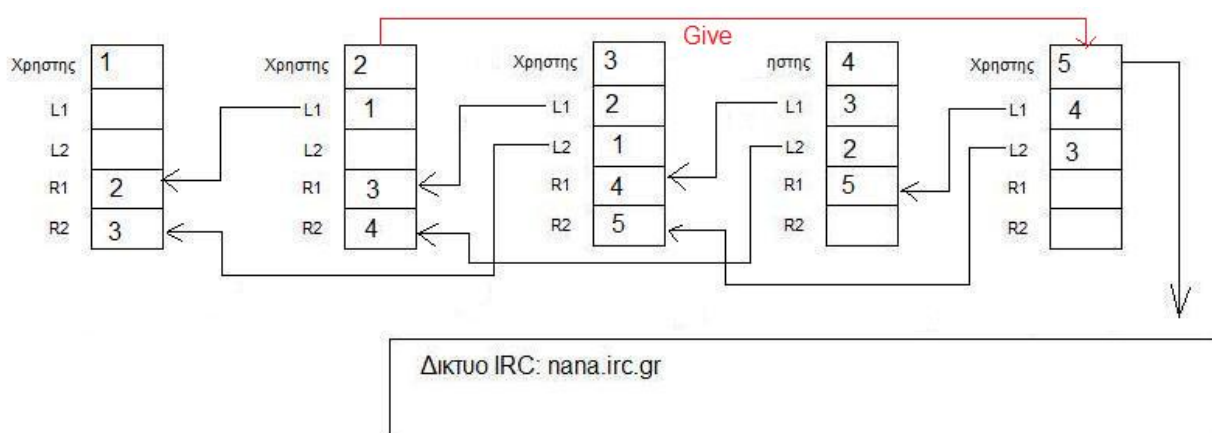
Όταν παραλαμβάνει το μήνυμα START τότε δημιουργεί ένα καινούριο αρχείο στο δίσκο με δικαιώματα εγγραφής με όνομα ίδιο με το αρχικό. Το όνομα το παίρνει από την εντολή με φίλτρο. Έπειτα προσθέτει στην λίστα downloading το όνομα του αρχείου ώστε να ενημερωθεί ο χρήστης

και δημιουργεί μια μεταβλητή που καταχωρεί μέσα της το μέγεθος του αρχείου που είναι για παραλαβή. (το οποίο παίρνει από το πρώτο μήνυμα)

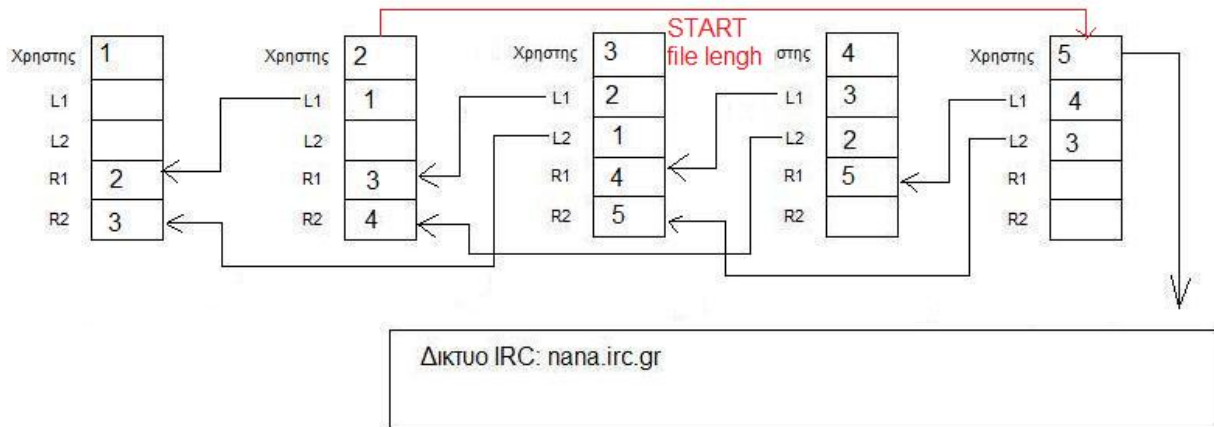
Στην περίπτωση που τα δεδομένα που λαμβάνονται δεν είναι START τότε το πρόγραμμα τα καταχωρεί στο αρχείο που άνοιξε προηγουμένως και αυξάνει την μεταβλητή δεδομένων.

Τώρα στην περίπτωση που ληφθεί το μήνυμα EOF τότε το πρόγραμμα κλείνει το αρχείο και όλες τις συνδέσεις και μηδενίζει όλες τις ανάλογες μεταβλητές. Αυτό σημαίνει ότι η λήψη του αρχείου ολοκληρώθηκε επιτυχώς και είναι έτοιμο για χρήση.

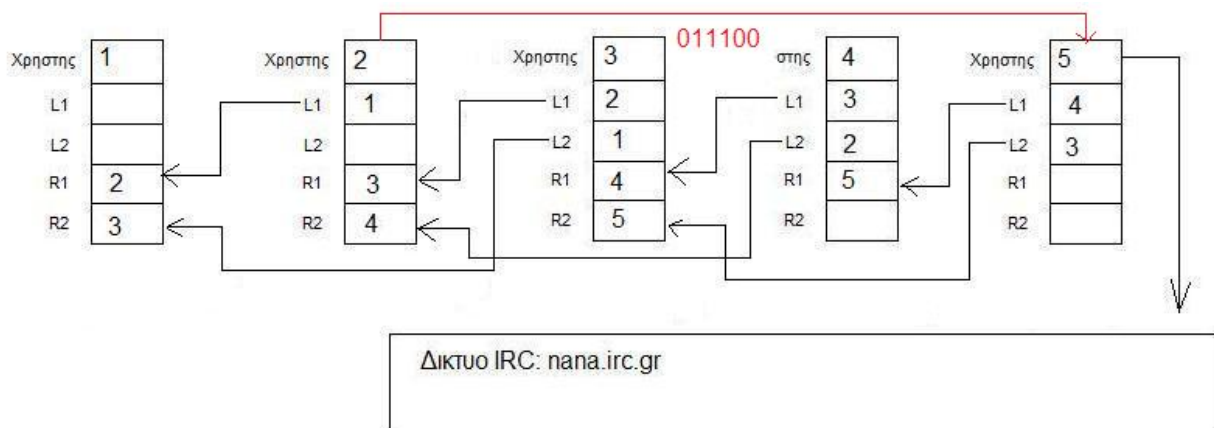
Επίσης ανά ένα δευτερόλεπτο ενημερώνονται τα δύο labels που είναι υπεύθυνα για την ενημέρωση του χρήστη για την κατάσταση μεταφοράς και τα υπολειπόμενα bytes.



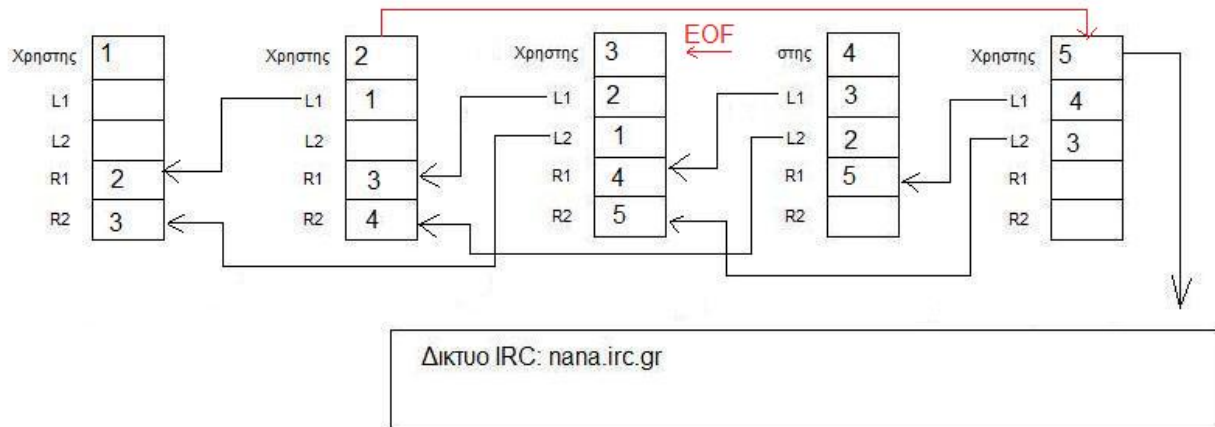
Εδώ βλέπουμε σχηματικά την σύνδεση που δημιουργεί ο χρήστης 2 με τον 5 και τον μήνυμα GIVE που του στέλνει.



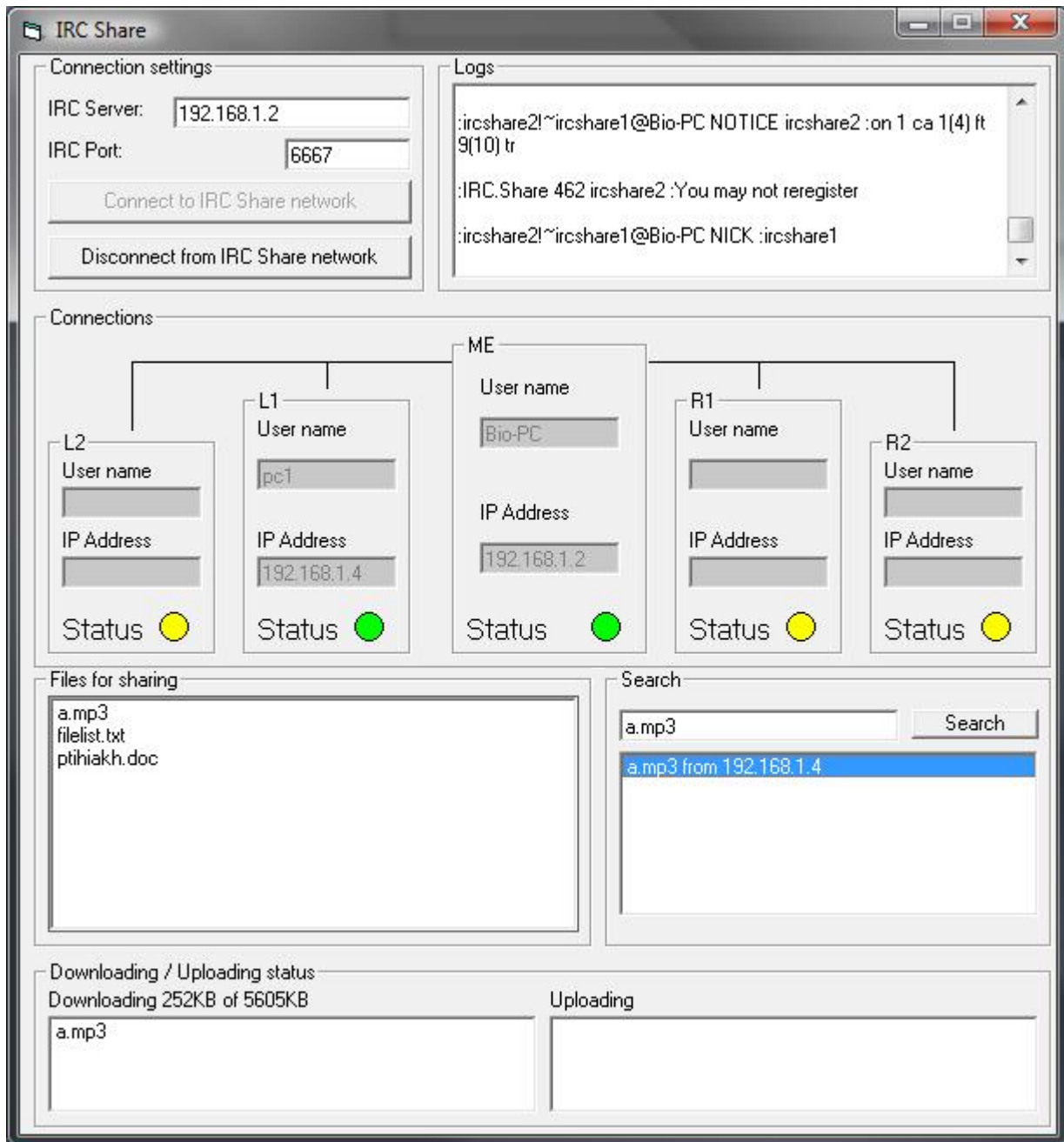
Τώρα ο χρήστης 5 στέλνει στον 2 το μήνυμα START μαζί με το μέγεθος του αρχείου.



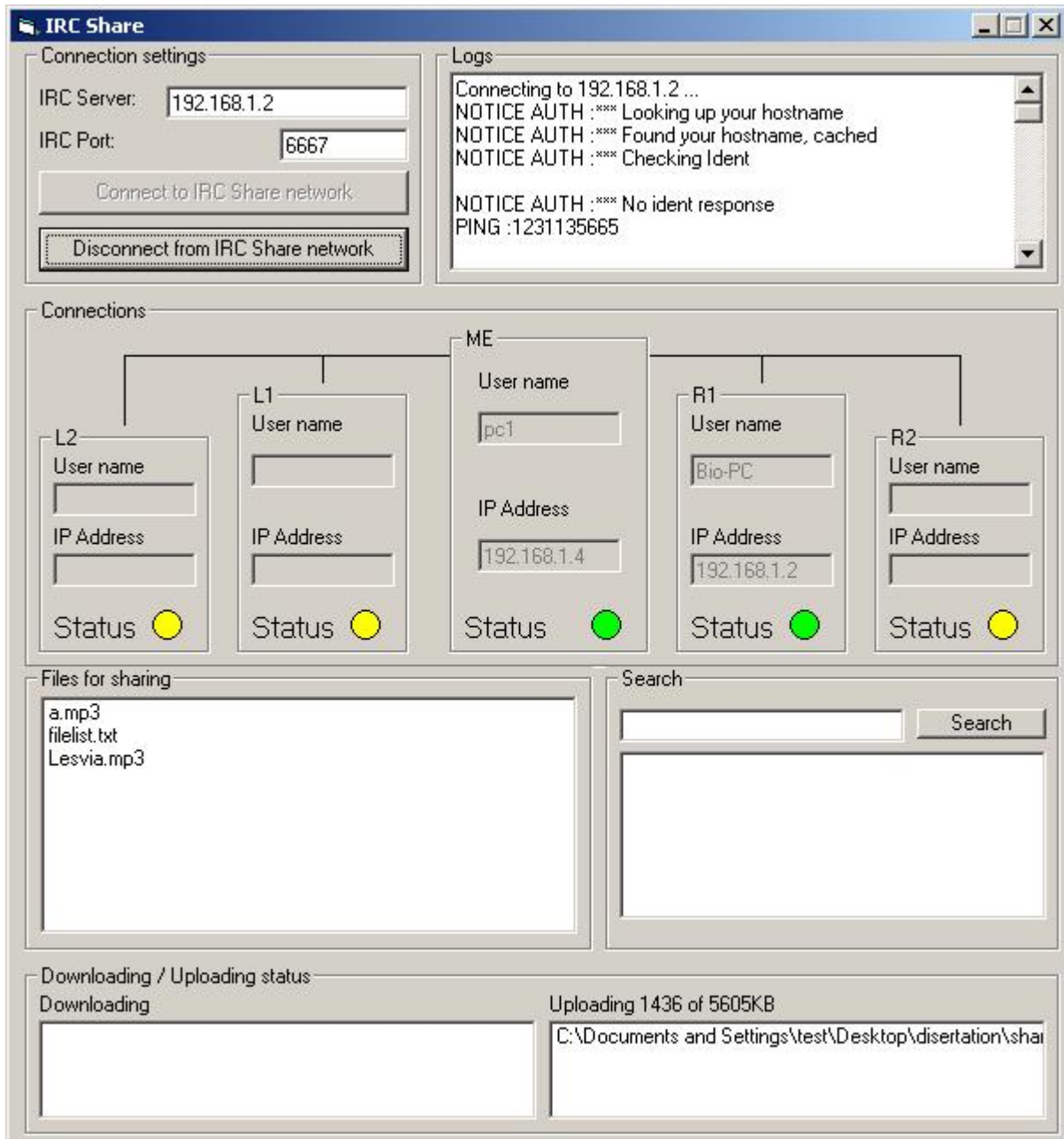
Μεταφορά δεδομένων από τον 5 προς τον 2 με πακέτα των 4096 bytes.



Αποστολή του τελευταίου μηνύματος EOF στον χρήστη 2 για το κλείσιμο του αρχείου.



Εδώ βλέπουμε ένα στιγμιότυπο από τον χρήστη 2 ο οποίος κατεβάζει το αρχείο από τον 5. Μπορούμε να διακρίνουμε ότι έχει προστεθεί στην λίστα downloading το αρχείο και πάνω από την λίστα βλέπουμε την κατάσταση της μεταφοράς.



Εδώ αντίστοιχα βλέπουμε τον χρήστη 5 κατά την διάρκεια αποστολής του αρχείου. Επίσης διακρίνεται το αρχείο που αποστέλλεται στα αντίστοιχα πεδία.

Κεφάλαιο 6: Μελλοντικές προσθήκες

Το πρόγραμμα μου στην φάση που βρίσκεται λειτουργεί αποτελεσματικά και διεκπεραιώνει τους βασικούς στόχους που είναι η σύνδεση στο δίκτυο, η αναζήτηση αρχείων και τέλος η μεταφορά αυτών.

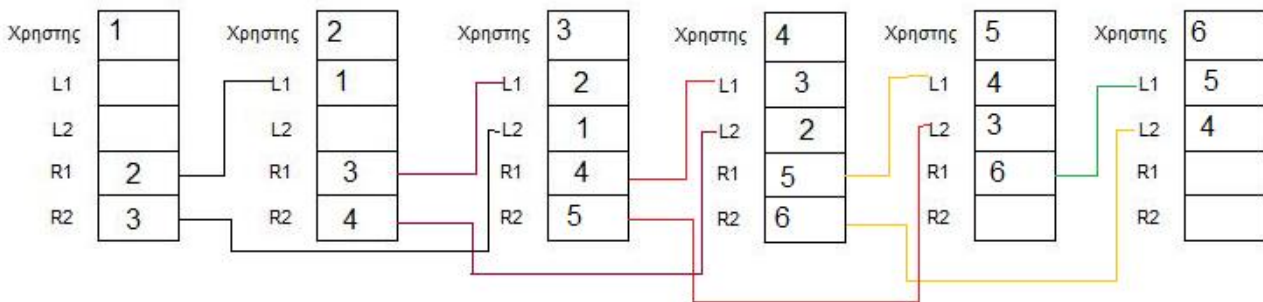
Κατά την φάση του προγραμματισμού όμως διαπίστωσα ότι μπορεί να δεχτεί ορισμένες μελλοντικές βελτιώσεις η οποίες δεν έγιναν ώστε να διατηρηθεί το πρόγραμμα απλό στην εξήγηση και λειτουργικό.

Παρακάτω θα αναφέρω και θα αναλύσω μερικές από αυτές χωρίς αυτό να σημαίνει ότι δεν υπάρχουν και άλλες.

Μια πολλή χρήσιμη προσθήκη στον κώδικα θα ήταν για την αποκατάσταση των συνδέσεων δηλαδή σε περίπτωση που 2 ή περισσότεροι χρήστες βγουν ταυτόχρονα από το δίκτυο.

Στο κεφάλαιο αποκατάσταση συνδέσεων καλύψαμε την περίπτωση που κάποιος μεμονωμένος χρήστης βγει από το δίκτυο ενώ εδώ θα μιλήσουμε για ομάδα χρηστών.

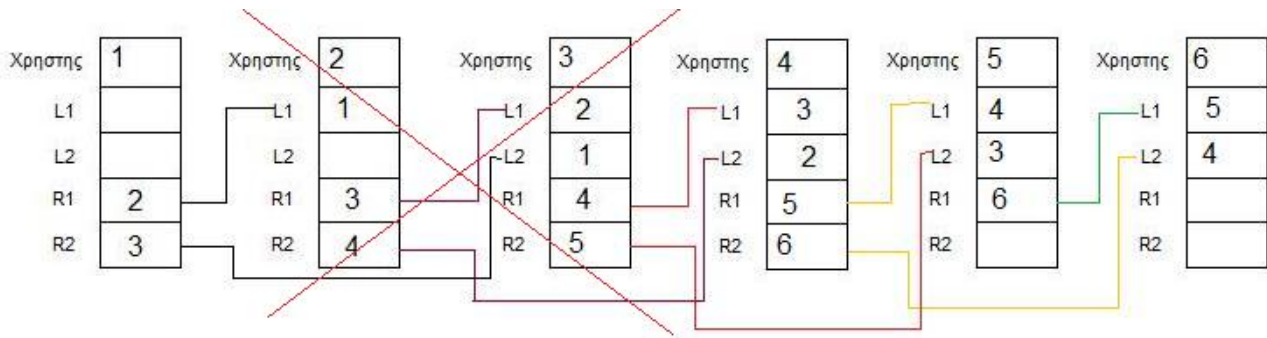
Θα πάρουμε το παράδειγμα με τους έξι χρήστες:



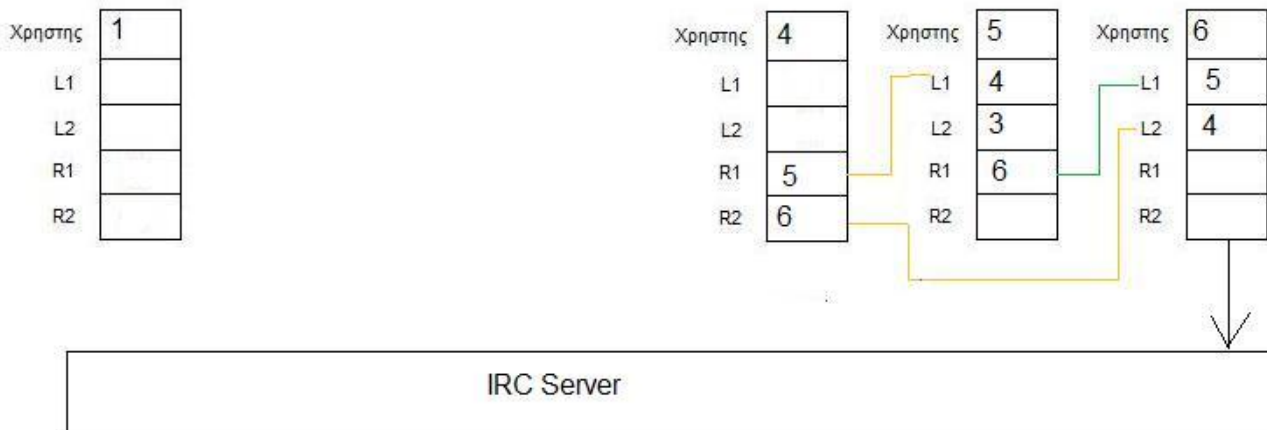
Σε αυτό το παράδειγμα θα βγουν ταυτόχρονα οι χρήστες 2 και 3 και έτσι δεν μπορεί να ισχύσει ο κώδικας που έχω γράψει.

Θα πρέπει όμως να αναφέρω εδώ ότι η περίπτωση να βγουν ταυτόχρονα δύο ή περισσότεροι είναι πολύ απίθανη. Μπορεί να γίνει όμως σε περίπτωση που μια ομάδα χρηστών είναι συνδεδεμένη από κάποιο τοπικό δίκτυο.

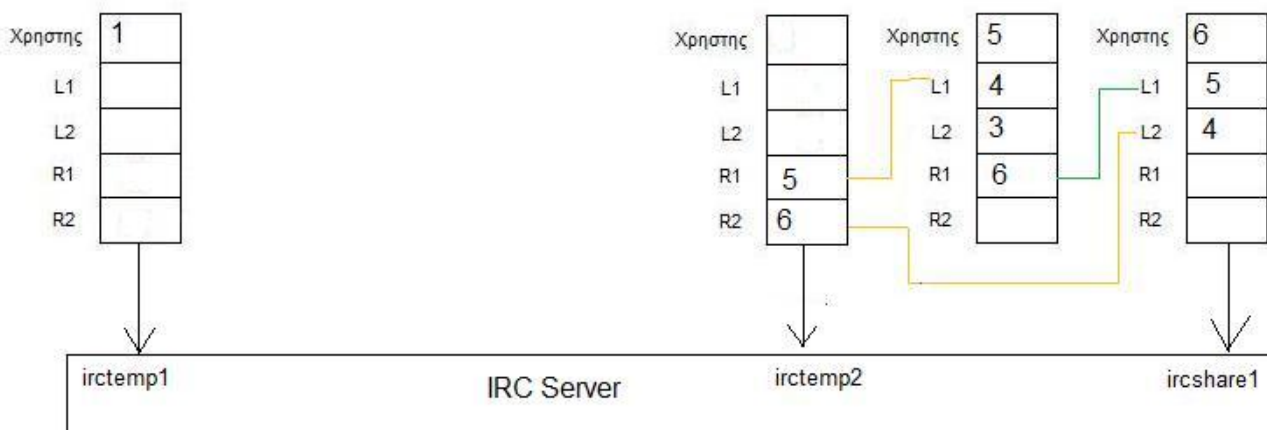
Για αυτό το λόγο δεν πρέπει να αποκλειστεί από μελλοντική προσθήκη αλλά και επειδή είναι πολύ απίθανο να συμβεί δεν το έχω προγραμματίσει.



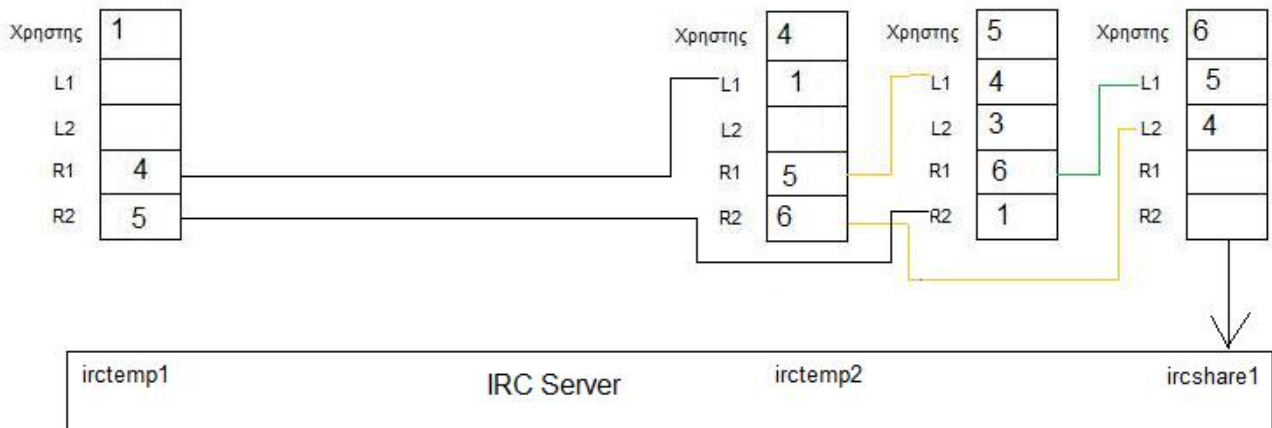
Εδώ βλέπουμε ότι ο χρήστης 2 και 3 έχουν βγει από το δίκτυο. Έτσι έχει μείνει ο 1,4,5 και 6. Το πρόβλημα όμως είναι ότι έχει κοπεί η σύνδεση μεταξύ του 1 και των υπολοίπων.



Ο μόνος τρόπος για να επανακτηθεί η σύνδεση είναι να συνδεθούν τα άκρα της κάθε απομονωμένης ομάδας στον IRC Server και να επικοινωνήσουν μεταξύ τους.



Έτσι όποιο άκρο μπαίνει πρώτο στο IRC δίκτυο παίρνει το όνομα `irctemp1` και το δεύτερο άκρο παίρνει το όνομα `irctemp2`. Μόλις ο ένας χρήστης δει τον άλλον ανταλλάζουν μεταξύ τους πληροφορίες σχετικά με τις διευθύνσεις που έχουν και έτσι επαναφέρουν την χαμένη σύνδεση. Αφού το δίκτυο ενωθεί τότε οι δύο χρήστες βγαίνουν από το irc δίκτυο και συνεχίζεται η λειτουργία του κανονικά.



Ένα άλλο χαρακτηριστικό που αξίζει να προστεθεί είναι να αλλάξει ο τρόπος με τον οποίο διαχειρίζονται οι συνδέσεις στο IRC δίκτυο. Δηλαδή το πρόγραμμα μου δεν θα μπορέσει να διαχειριστεί πολλές ταυτόχρονες συνδέσεις στο IRC δίκτυο αν εκείνη την στιγμή 2 χρήστες ανταλλάζουν μεταξύ τους πληροφορίες μέσω του δικτύου. Θα εξηγήσω με ένα απλό παράδειγμα σύνδεσης.

Όταν ο πρώτος χρήστης προσπαθεί να συνδεθεί στο IRC δίκτυο παίρνει το όνομα `ircshare1` και περιμένει τον επόμενο. Ο δεύτερος που θα συνδεθεί θα δει ότι υπάρχει ήδη ένας χρήστης με το όνομα `ircshare1` και θα ονομαστεί `ircshare2` ώστε να ανταλλάζουν πληροφορίες μεταξύ τους για την σύνδεση. Αν όμως εκείνη την στιγμή μπει και άλλος χρήστης στο δίκτυο τότε δεν θα μπορέσει να συνδεθεί αλλά και ούτε να περιμένει να τελειώσουν οι προηγούμενοι χρήστες.

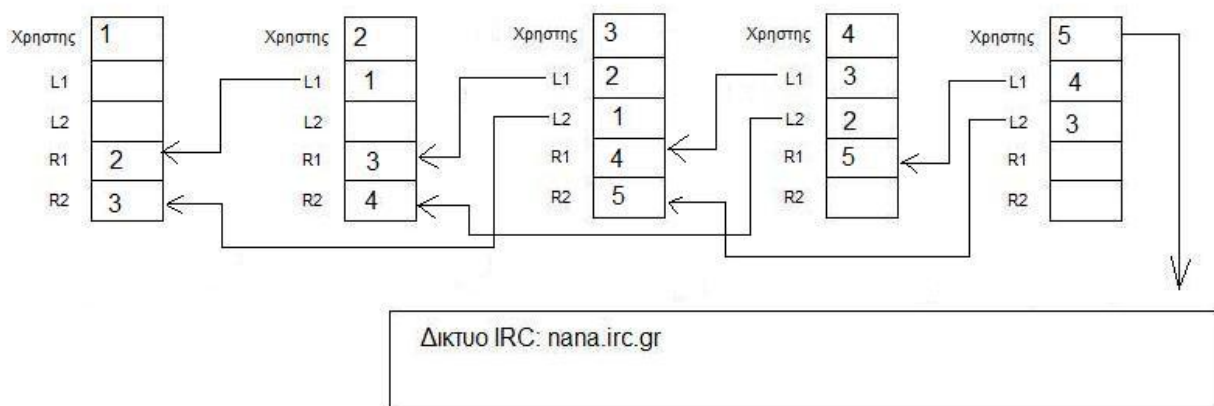
Αυτό που θα πρέπει να γίνει είναι ο τρίτος χρήστης που θα συνδεθεί και δει ότι δεν μπορεί να πάρει κάποιο όνομα, να πάρει κάποιο προσωρινό και μόλις δει ότι τελειώσαν οι προηγούμενοι να πάρει το ανάλογο, δηλαδή το `ircshare2`.

Αυτή η προσθήκη θα ήταν χρήσιμη σε περίπτωση που το πρόγραμμα έβγαине στο διαδίκτυο και το χρησιμοποιούσαν πολλοί χρήστες.

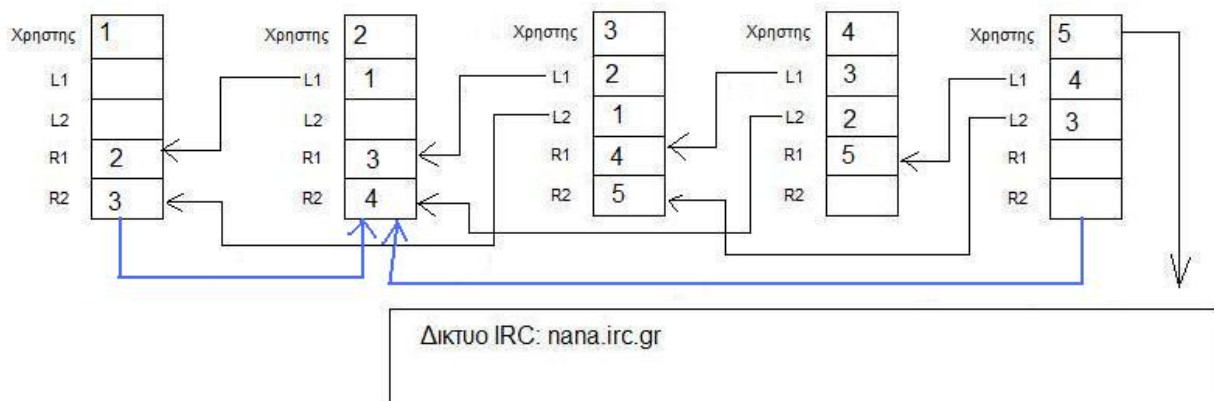
Μια άλλη αλλά και πολύ σημαντική προσθήκη θα ήταν η μεταφορά αρχείων από πολλαπλούς χρήστες ταυτόχρονα ώστε να βελτιωθεί η ταχύτητα λήψης των αρχείων. Το βλέπουμε συνήθως σε μεγάλα προγράμματα που υπάρχουν καιρό στην αγορά και βλέπουμε πως βελτιώνεται η ταχύτητα λήψης.

Θα δείξω με το παρακάτω παράδειγμα πως γίνεται αυτό και πως βοηθάει στην μεταφορά:

Θα υποθέσουμε πως ο χρήστης 2 δημιουργεί μια αναζήτηση για το αρχείο a.mp3 και του επιστρέφουν αποτελέσματα 2 χρήστες ότι το έχουν. (1 και 5)



Στην παρούσα φάση έτσι όπως το έχω προγραμματίσει ο χρήστης 2 μπορεί να κατεβάσει το αρχείο ή από τον 1 ή από τον 5. Αυτό δεν αποτελεί πρόβλημα σε δίκτυο υψηλών ταχυτήτων αλλά στα απλά δίκτυα η ταχύτητα μεταφοράς θα είναι χαμηλή. Αν θέλουμε να ανεβάσουμε την ταχύτητα θα πρέπει ο χρήστης 2 να κατεβάζει το αρχείο παράλληλα και από τους δύο χρήστες. Έτσι θεωρητικά θα έχουμε διπλάσια ταχύτητα μετάδοσης.



Τώρα βλέπουμε στο σχήμα ότι οι χρήστες 1 και 5 στέλνουν ταυτόχρονα το ίδιο αρχείο στον 2. Για να στείλουν το ίδιο αρχείο όμως ταυτόχρονα θα πρέπει να συνεννοηθούν μεταξύ τους ώστε να στείλει ο 1 χρήστης το πρώτο μισό κομμάτι και ο 5 το άλλο μισό. Μόλις τελειώσει η μεταφορά και από τους δύο χρήστες τότε ο 2 θα ενώσει τα δύο μέρη σε ένα και λειτουργικό αρχείο.

Αυτό επίσης μπορεί να γίνει και με περισσότερους από δύο χρήστες. Μπορεί να στέλνουν ακόμα και 10 χρήστες το 1/10 του αρχείου σε έναν χρήστη ώστε να βελτιωθεί η ταχύτητα μεταφοράς. Αρκεί να έχουν πολλοί χρήστες το ίδιο αρχείο.

Ως τελευταίο χαρακτηριστικό που θα μπορούσα να προσθέσω στο πρόγραμμα το οποίο το διαθέτουν όλα τα υπόλοιπα που βρίσκονται στο διαδίκτυο, είναι να συνεχίζεται από το σημείο που διακόπηκε η μεταφορά ενός αρχείου.

Υπάρχουν πολλοί λόγοι οι οποίοι μπορούν να διακόψουν μια μεταφορά αρχείου αλλά ο πιθανότερος είναι να τερματιστεί η σύνδεση πάνω στην μεταφορά.

Αυτό που θα μπορούσε να γίνει για να διορθωθεί το πρόβλημα είναι να γίνεται έλεγχος πριν από κάθε αποστολή ή λήψη ενός αρχείου ώστε να βλέπει το πρόγραμμα σε ποιο ποσοστό έχει ολοκληρωθεί και να ζητάει μόνο τα υπολειπόμενα δεδομένα από τον χρήστη.

Το σημαντικό πλεονέκτημα εδώ είναι η μικρότερος χρόνος αποστολής του αρχείου αφού ο χρήστης θα ζητήσει μόνο το υπόλοιπο, αλλά και η χαμηλότερη κυκλοφορία δεδομένων στο δίκτυο από περιττές ολόκληρες μεταφορές.

Κεφάλαιο 7: Συμπεράσματα

Το πρόγραμμα που παρουσίασα στις παραπάνω σελίδες δημιουργήθηκε στο πλαίσιο της πτυχιακής μου εργασίας για το ΑΤΕΙ Κρήτης. Με λίγα λόγια προσφέρει κάτι ξεχωριστό από τα υπόλοιπα που κυκλοφορούν στο διαδίκτυο εξαιτίας του τρόπου σύνδεσης των χρηστών.

Όπως ανέφερα και παραπάνω μπορεί να δεχτεί ορισμένες βελτιώσεις που πιστεύω ότι θα μπορέσει να συναγωνιστεί τα ήδη υπάρχοντα προγράμματα όπως το eMule, Kazaa κ.α.

Στις μετρήσεις που έκανα στο τοπικό μου δίκτυο δεν διαπίστωσα κάποιο λάθος στην λειτουργία του αλλά οπωσδήποτε σε πραγματικές συνθήκες θα φανούν οι ατέλειες που θα διορθωθούν με τον καιρό.

Βιβλιογραφία

Δίκτυα υπολογιστών : Tanenbaum Τέταρτη έκδοση

Ιστορία των δικτύων: wikipedia

Κώδικας και παραδείγματα: www.planet-source-code.com

Κώδικας και αρχιτεκτονικές δικτύων: sourceforge.net