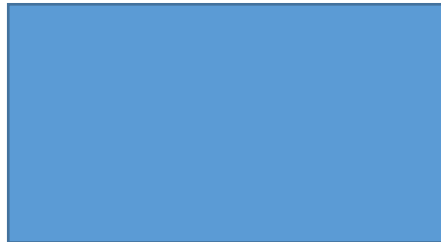




**Τεχνολογικό Εκπαιδευτικό ίδρυμα Κρήτης**

**Σχολή Τεχνολογικών Εφαρμογών**

**Τμήμα Μηχανικών Πληροφορικής**



**Κατανεμημένο σύστημα για την ανίχνευση γνώσης  
μέσω P2P δικτύου**

Συγγραφή: Λιώσης Ευάγγελος-Μάριος (ΤΠ 3630)

Υπεύθυνος διδάσκων: κος Παπαδάκης Νικόλαος

Κατεύθυνση: Μηχανικοί Λογισμικού

Ηράκλειο Κρήτης, Φεβρουάριος 2019



## ΠΕΡΙΕΧΟΜΕΝΑ

Ευχαριστίες .....	σελ. 5
Σύνοψη .....	σελ. 8
1. Εισαγωγή .....	σελ. 9
1.1. Σχετικά με τα Peer-to-Peer δίκτυα .....	σελ. 10
1.1.1. Κατηγορίες Peer-to-Peer δικτύων .....	σελ. 10
1.1.1.1. Πρώτης γενιάς P2P δίκτυα .....	σελ. 10
1.1.1.2. Αποκεντρωτικά P2P δίκτυα .....	σελ. 11
1.1.1.3. P2P δίκτυα τρίτης γενιάς .....	σελ. 12
1.2. Πλεονεκτήματα P2P δικτύων .....	σελ. 12
1.3 Μειονεκτήματα P2P δικτύων .....	σελ. 13
1.4 Διαμοιρασμός αρχείων μέσω Peer-to-Peer δικτύων .....	σελ. 13
2. Τεχνολογίες Συγγραφής κώδικα κι Ανάπτυξης της Εφαρμογής .....	σελ. 14
2.1. Ανάπτυξη εφαρμογής Python .....	σελ. 14
2.1.1. Εισαγωγή στην Python .....	σελ. 14
2.1.2. Δημιουργία αρχείου Βάσης χρηστών (db.py) .....	σελ. 15
2.1.3. Δημιουργία αρχείου Peer (peer.py) .....	σελ. 16
2.1.4. Δημιουργία αρχείου Registry (registry.py) .....	σελ. 17
Βιβλιογραφία .....	σελ. 18
Πίνακας Εικόνων .....	σελ. 19

## **Ευχαριστίες**

*Θα ήθελα να εκφράσω τις ευχαριστίες μου προς τον υπεύθυνο καθηγητή-επόπτη (supervisor), κο Παπαδάκη Νικόλαο, για τη συνεχή βοήθεια, καθοδήγηση κι υποστήριξη που μου παρείχε τόσο κατά τις σπουδές μου ως προπτυχιακός φοιτητής του τμήματος όσο και, στο πλαίσιο της πτυχιακής εργασίας μου, κατά τη διάρκεια δημιουργίας του κώδικα καθώς και συγγραφής της εργασίας. Φυσικά, δεν θα μπορούσα να μην ευχαριστήσω τους γονείς μου που είναι οι πιο θερμοί υποστηρικτές μου σε κάθε βήμα και στάδιο της ζωής μου.*





## Σύνοψη





## 1. Εισαγωγή

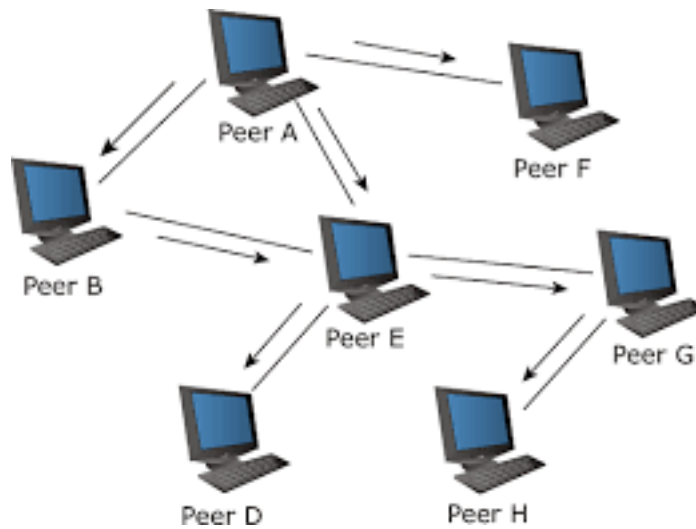
### 1.1. Σχετικά με τα Peer-to-Peer δίκτυα

#### 1.1.1. Κατηγορίες Peer-to-Peer δικτύων

Τα ομότιμα δίκτυα, είναι κατηγορία δικτύωσης η οποία σε σχέση με τα παραδοσιακού τύπου δίκτυα καταργεί την ιεραρχική μορφή των κόμβων.

Στα peer-to-peer δίκτυα, η λειτουργία των κόμβων του γίνεται με τέτοιο τρόπο όπου κάθε ένας κόμβος μπορεί να λειτουργήσει είτε ως πελάτης είτε ως εξυπηρετητής ανάλογα με τις ανάγκες της δικτύωσης

Με άλλα λόγια, η λειτουργία των peer-to-peer δικτύων βασίζονται απο πολλούς υπολογιστές οι οποίοι είναι συνδεδεμένοι μεταξύ τους και ανταλλάσσουν δεδομένα.

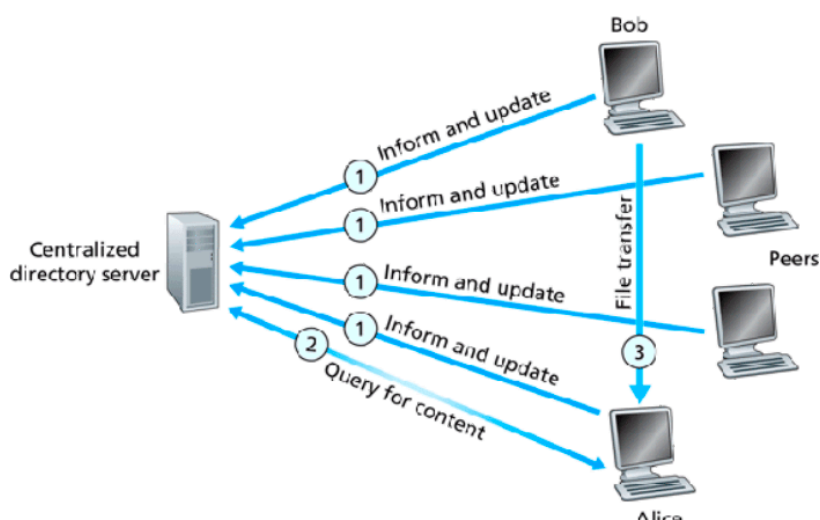


Αυτό εξυπηρετεί την ανάγκη του να μην υπάρξει επιβάρυνση της δικτύωσης αν παρουσιαστεί πρόβλημα σε ένα ή περισσότερους υπολογιστές και σταματήσουν να λειτουργούν .

Σε αυτό το είδος δικτύωσης, για τα ζητήματα ασφαλείας και σωστής διαχείρισης του εκάστοτε κόμβου δεν υπάρχει κάποιος κεντρικός υπολογιστής ο οποίος να τα διαχειρίζετε και να τα επιλύει, οπότε αποκλειστική υπευθυνότητα έχει ο κάθε ένας κόμβος για τα ζητήματα που θα παρουσιαστούν στο δίκτυο.

### 1.1.1.1. Πρώτης γενιάς P2P δίκτυα

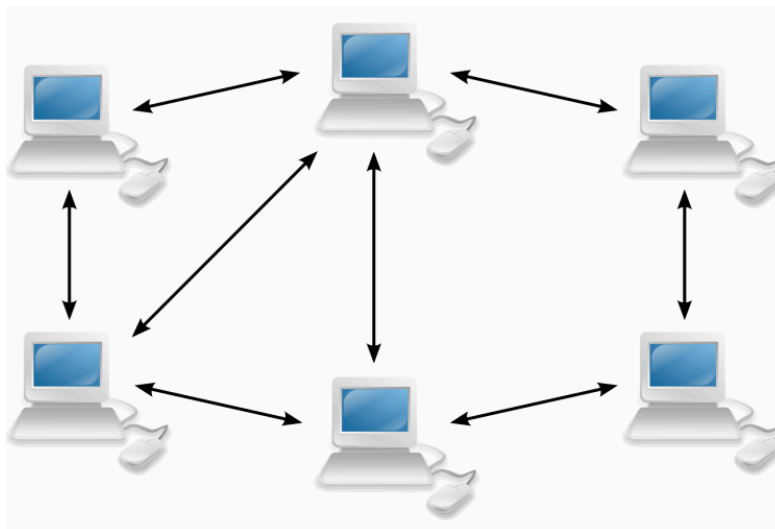
Γενικώς, τα δίκτυα αυτά ονομάζονται απο πολύ κόσμο ως “πρώτης γενιάς P2P δίκτυα” ή “Συγκεντρωτικά”. Στα P2P δίκτυα πρώτης γενιάς, υπάρχει ένας κεντρικός σέρβερ, ο οποίος κρατά αποθηκευμένα όλα τα δεδομένα τα οποία μπορούν να μοιραστούν οι χρήστες του δικτύου. Επίσης οι χρήστες έχουν την δυνατότητα να αναζητήσουν τα αρχεία και τα δεδομένα που επιθυμούν να διαμοιραστούν με την χρήση κατάλληλων client προγραμμάτων.



Τα προγράμματα αυτά, αναζητούν τα δεδομένα που επιθυμεί ο χρήστης, και μόλις η διαδικασία αυτή ολοκληρωθεί τότε μεταξύ των εκάστοτε χρηστών δημιουργείται ένας διάλογος ανταλλαγής δεδομένων μεταξύ τους ώστε να γίνει με επιτυχία η μεταφορά τους.

### 1.1.1.2. Αποκεντρωτικά δίκτυα P2P

Τα αποκεντρωτικά δίκτυα αποτελούν την εξέλιξη των peer-to-peer δικτύων και εν δυνάμει την δεύτερη γενιά δικτύωσης μεταξύ τερματικών και κόμβων σε ρόλο client - server.

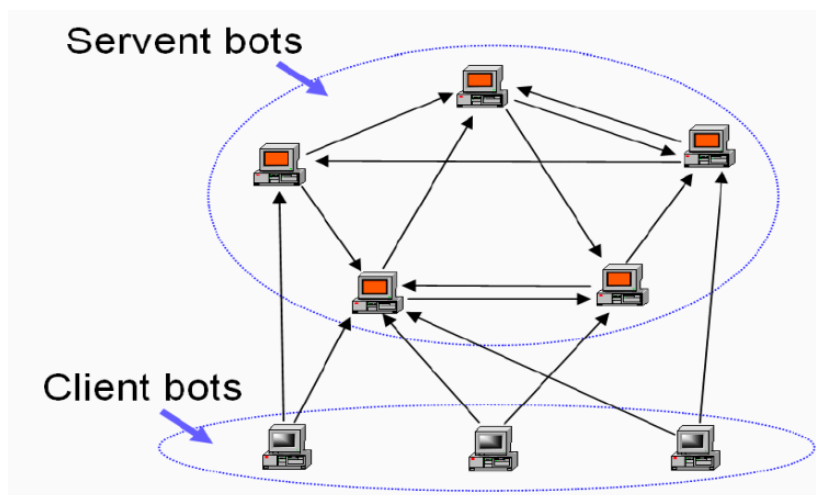


Στην δομή αυτού του δικτύου δεν υπάρχει κάποιος index server. Χαρακτηριστικό αυτού του δικτύου είναι πως όταν ένας κόμβος προσπαθήσει να συνδεθεί, τότε μεταξύ των κόμβων του δικτύου που προπήρχαν διαδίδεται η πληροφορία η οποία φανερώνει την παρουσία του στο δίκτυο καθώς και ότι κάθε κόμβος είναι παράλληλα και client και server.

Με την ίδια διαδικασία γίνεται και η αναζήτηση δεδομένων καθώς και ελέγχεται η σωστή λειτουργία σύνδεσης μεταξύ των κόμβων.

### 1.1.1.3. P2P δίκτυα τρίτης γενιάς

Μια εξέλιξη του προτύπου λειτουργίας και της δομής των αποκεντρωτικών δικτύων είναι τα “peer-to-peer δίκτυα τρίτης γενιάς”. Κύριο χαρακτηριστικό τους είναι πως η επιλογή τους από χρήστες τους επιτρέπει να διατηρούν την ανωνυμία τους με χρήση κωδικοποίησης καθώς και την αποτροπή εξωτερικών χρηστών να ελέγξουν το συνολικό δίκτυο.



Επιπλέον στοιχείο που χαρακτηρίζει αυτά τα δίκτυα είναι η υψηλή βιωσιμότητα τους καθώς και συνεχόμενη εισροή δεδομένων λόγω του ότι βασίζονται στους χρήστες και τους κόμβους που τα αποτελούν.

## 1.2. Πλεονεκτήματα P2P δικτύων

Όπως προαναφέραμε, τα P2P δίκτυα δεν βασίζονται σε κεντρικό υπολογιστή ώστε να ελέγχει το σύστημα δικτύωσης, με αποτέλεσμα να παρέχοντε ασταμάτητα δεδομένα μεταξύ των χρηστών χωρίς να επηρεαστεί η λειτουργία του δικτύου από ύπαρξη βλάβης σε κόμβο με πολλούς χρήστες, καθώς και ότι δεν υπάρχει κανένα data traffic λόγω της αρχιτεκτονικής του. Αυτό το καθιστά αρκετά ισχυρό σε Dos επιθέσεις.

Ένα σημαντικό στοιχείο των ομότιμων δικτύων είναι πως μπορούν να αποτελούνται από προσωπικούς υπολογιστές με αποτέλεσμα η ισχύς τους να είναι σε πολύ υψηλό επίπεδο σε σχέση με άλλα δίκτυα που χρησιμοποιούν servers.

Επιπλέον χαρακτηριστικό αυτών των δικτύων είναι πως η ανταλλαγή δεδομένων μεταξύ των χρηστών μπορεί να γίνει με πολύ μεγάλες ταχύτητες όσο μεγαλώνει ο αριθμός των κόμβων.

Τέλος το πιο σημαντικό στοιχείο που τα χαρακτηρίζει είναι ότι υπάρχει πολυ-επίπεδη μεταφορά δεδομένων μεταξύ των χρηστών καθώς και ότι επιτυγχάνεται η απόκρυψη των στοιχείων του εκάστοτε χρήστη εξασφαλίζοντας την ανωνυμία του. Παρόλα αυτά με την εφαρμογή ειδικών τεχνικών και μεθόδων μπορεί να γίνει εύρεση των στοιχείων του χρήστη.

### **1.3 Μειονεκτήματα P2P δικτύων**

Η ανωνυμία η οποία μπορεί να παρέχετε με την χρήση αυτών των δικτύων μπορεί να οδηγήσει σε αρκετά θέματα σχετικά με την ποιότητα των δεδομένων που διαμοιράζονται καθώς δεν υπάρχει κάποιος μηχανισμός ο οποίος να ταταυτοποιεί τους χρήστες του δικτύου, επομένως κάθε χρήστης είναι αποκλειστικά υπεύθυνος για την σωστή και αποδοτική χρήση του.

Κλείνοντας, μπορούμε να καταλάβουμε πως σε αυτού του είδους τα δίκτυα ο χρήστης είναι ο κύριος υπεύθυνος για την σωστή λειτουργία τους καθώς και της ποιότητας του.

Για τον λόγο αυτό θα πρέπει να υπάρχει ένα σύνολο χρηστών το οποίο θα έχει ενεργό ρόλο στην ανάπτυξη και διατήρηση του σκοπού της ανταλλαγής ποιοτικών δεδομένων και όχι παράνομων διαδικασιών.

### **1.4 Διαμοιρασμός αρχείων μέσω Peer-to-Peer δικτύων**

Η ανταλλαγή δεδομένων και αρχείων ή αλλιώς ο διαμοιρασμός τους μέσα στο δίκτυο είναι μια τεχνική η οποία επιτυγχάνεται με την χρήση των torrent tracker.

Οι torrent tracker είναι κατηγορίες σελίδων οι οποίες μπορούν και έχουν είσοδο στην βάση δεδομένων ενός server. Σε αυτές τις ιστοσελίδες αποθηκεύονται πληροφορίες σχετικά με τα δεδομένα που θέλουν οι χρήστες να διαμοιραστούν αναμετάξυ τους. Torrent tracker σελίδες είναι το PirateBay, το Kickass και δεκάδες άλλες. Για κάθε χρήστη που κάνει λήψη οποιονδήποτε δεδομένων αποθηκεύετε ένα αρχείο με στοιχεία σχετικά με το ποιός το διαμοιράζετε και ποιός το κατεβάζει. Αυτό το αρχείο ονομάζεται torrent, το οποίο δεν είναι ένα συνηθισμένο αρχείο, αλλά ένα αρχείο στο οποίο καταγράφονται ποίοι χρήστες μπορούν να αποστείλουν δεδομένα και ποίοι χρήστες να τα κατεβάσουν. Το torrent αρχείο το παίρνει ένας torrent client και μέσω αυτού έχει πρόσβαση στην βάση δεδομένων ώστε να επιτευχθεί λήψη των δεδομένων απο τον χρήστη. Τέτοιου είδους torrent client προγράμματα είναι το μtorrent. Έχοντας περιγράψει τον τρόπο λειτουργίας τους, μπορούμε να καταλάβουμε το πρωτόκολλο BitTorrent το οποίο είναι το επικρατέστερο για την ανταλλαγή και διαμοιρασμό αρχείων. Το πρωτόκολλο αυτό λειτουργεί κατεκερματίζοντας τα αρχεία σε πολύ μικρότερα αρχεία δεδομένων ώστε να επιτευχθεί σωστά η λήψη τους.

## 2. Τεχνολογίες Συγγραφής κώδικα κι Ανάπτυξης της Εφαρμογής

### 2.1. Ανάπτυξη εφαρμογής Python

#### 2.1.1. Εισαγωγή στην Python

Η Python αποτελεί μια ανώτερη ψηφιακή γλώσσα διερμηνίας, με εξελισσόμενες έννοιες. Δημιουργήθηκε μεγάλη κύρια βιβλιοθήκη της (standard library) προκειμένου να εξυπηρετεί το διάβασμα του κώδικα.

Η αναγκαιότητα της γλώσσας Python ήταν μεγάλη που από τη στιγμή που δημιουργήθηκε αναπτύχθηκαν την ίδια στιγμή εφαρμογές με πολύ απλή μορφή.

Αυτή η πολύ απλή μορφή της, την καθιστά εύελικτη ως προς την εκμάθησή της εξασφαλίζοντας μια πολύ ομαλή αναγνωσιμότητα ως προς τον κώδικα που παράγει ο χρήστης, με αποτέλεσμα ο κώδικας να είναι εύκολο να συντηρηθεί και να επεκταθεί.

Έχει υπογραμμιστεί ότι προκειμένου να πραγματοποιηθεί διόρθωση σε κώδικα απαιτείται δυο φορές ανώτερη νοητική διεργασία από όταν γράφτηκε. Δηλαδή, γραμμένος ο πιθανότερα 'εξυπνότερος' -δυσκολότερος κώδικας, εκ των πραγμάτων δεν γίνεται ο δημιουργός να τον αποσφαλματώσει. Συνεπώς, όλα επισημαίνουν τη δυνατότητα της Python για γρήγορη δημιουργία εφαρμογών, κυρίως συγκρινόμενη με γλώσσες χαμηλότερου επιπέδου.

Για τη μετατροπή του προγράμματος από γλώσσα προγραμματισμού -που πλησιάζει τον άνθρωπο- σε τεχνητή γλώσσα κυριαρχεί ο διαχωρισμός των προγραμμάτων σε δύο: οι διερμηνείς (interpreters) και οι μεταφραστές (compilers). Από τη μία, οι διερμηνείς μετατρέπουν άμεσα γραμμή προς γραμμή τον πηγαίο κώδικα του προγράμματος σε τεχνητή γλώσσα, αφετέρου οι μεταφραστές προβαίνουν σε μετατροπή όλου του προγράμματος σε γλώσσα μηχανής προκειμένου αυτό ακολούθως να μπορεί να εκτελεστεί. Ωστόσο, υπάρχει η ειδική περίπτωση όπου η γλώσσα προγραμματισμού χρησιμοποιεί εικονική μηχανή (virtual machine) στην οποία «τρέχει» ο κώδικας (ίδια περίπτωση με την Python). Πριν μετατραπεί σε τεχνητή γλώσσα (γλώσσα του υπολογιστή), παίρνει τη μορφή μιας ενδιάμεσης γλώσσας (bytecode).

Ο κώδικας πλήρως μορφοποιημένος πια, ονομάζεται αντικειμενικός κώδικας. Η κύρια βιβλιοθήκη διαθέτει μια μεγάλη ποικιλία, από ασύγχρονη επεξεργασία έως συμπιεσμένα αρχεία. Επειδή έχουν συμβάλει πολλοί ειδικοί κι έξυπνοι άνθρωποι στη γραφή του κώδικα, ο τελευταίος είναι ταχύτατος για τον μεγαλύτερο αριθμό πιθανών προς χρήση εφαρμογών. Οι νέες δυνατότητες που προσφέρει ο κώδικας ανταποκρίνονται σε πληθώρα πιθανών προβλημάτων που θα συναντήσει κανείς, αποκλείοντας έτσι την ανάγκη προσπάθειας ανακάλυψης από την αρχή του τροχού.

Η γλώσσα έχει δυνατότητα επέκτασης χάρη στο γεγονός ότι ένα βασικό τμήμα της καλύπτει τον πυρήνα της, εν αντιθέσει με τα όλα τα υπόλοιπα τμήματά της είναι modules που διευρύνουν τη λειτουργικότητά της. Μάλιστα, σε συνδυασμό με την ιδιότητά της ως γλώσσα ανοικτού κώδικα, δεν παραμένει σε στάσιμη μορφή αλλά είναι δυναμικής μορφής, όντας, συνεπώς, σε θέση να παρακολουθεί διαρκώς τις αναβαθμίσεις.

### 2.1.3. Δημιουργία αρχείου βάσης χρηστών (db.py)

Το παρών αρχείο που φέρει την ονομασία db.py είναι αυτο το οποίο δημιουργεί μια βάση δεδομένων στην οποία θα γίνεται ο έλεγχος για το εαν υπάρχει ο χρήστης που θα προσπαθεί να κάνει είσοδο στο σύστημα. Αν δεν υπάρχει ο χρήστης τότε το σύστημα θα καταχωρεί καινούριο χρήστη τα στοιχεία του, “Όνομα Χρήστη (username)”, “Μυστικός Κωδικός(password)” .

```
# checks if an account with the username exists
def is_account_exist(self, username):
    if self.db.accounts.find({'username': username}).count() > 0:
        return True
    else:
        return False

# registers a user
def register(self, username, password):
    account = {
        "username": username,
        "password": password
    }
    self.db.accounts.insert(account)

# retrieves the password for a given username
def get_password(self, username):
    return self.db.accounts.find_one({"username": username})["password"]

# checks if an account with the username online
def is_account_online(self, username):
    if self.db.online_peers.find({"username": username}).count() > 0:
        return True
    else:
        return False

# logs in the user
def user_login(self, username, ip, port):
    online_peer = {
        "username": username,
        "ip": ip,
        "port": port
    }
    self.db.online_peers.insert(online_peer)

# logs out the user
def user_logout(self, username):
    self.db.online_peers.remove({"username": username})

# retrieves the ip address and the port number of the username
def get_peer_ip_port(self, username):
    res = self.db.online_peers.find_one({"username": username})
    return (res["ip"], res["port"])
```

### 2.1.4. Δημιουργία αρχείου Peer (peer.py)

Κατά την δημιουργία του peer.py , λάβαμε υπ' όψιν πως ένας χρήστης σε peer- to-peer δίκτυο μπορεί να λειτουργήσει και ως Peer Server, στον οποίο θα συνδέονται άλλοι peers ώστε να επικοινωνούν, και ο ίδιος να γίνεται Peer Client ώστε να συνδέετε με άλλους χρήστες και να επικοινωνεί.

Δημιουργήσαμε τις παρακάτω κλάσεις, με τις οποίες γίνονται όλες οι ενέργειες που αναγράφω παραπάνω.

```
class PeerServer(threading.Thread):
```

Η παραπάνω κλάση, αρχικοποιεί τα δεδομένα που απαιτούνται για την δημιουργία του Peer Server, ελέγχοντας και κρατώντας τα credentials που χρειάζονται, όπως το όνομα χρήστη, το socket και το port του server, την κατάσταση του χρήστη για το αν επικοινωνεί ή όχι μέσα στο δίκτυο, το socket και το port του χρήστη (Peer A) που είναι σε επικοινωνία με κάποιον άλλον (Peer B), την κατάσταση του χρήστη σε σχέση με το δίκτυο και το όνομα χρήστη του Peer B.

Στη συνέχεια δημιουργείται ένα thread το οποίο παίρνει την ip address του χρήστη και ελέγχει εάν η εφαρμογή τρέχει σε περιβάλλον Windows ή σε περιβάλλον MacOS.

Μόλις ολοκληρωθεί αυτή η ενέργεια, το thread, ελέγχει εάν ο χρήστης είναι συνδεδεμένος στο δίκτυο και ελέγχει την κατάσταση του μαζί με όλα τα στοιχεία σύνδεσης. Επιπλέον μορφοποιεί τα δεδομένα για την έναρξη επικοινωνίας άλλων Peers με τον Peer του συστήματος και κάνει επιπλέον έλεγχο για την διαθεσιμότητα του χρήστη για επικοινωνία με άλλους Peers.

```
class PeerClient(threading.Thread):
```

Με την παραπάνω κλάση, γίνεται η αρχικοποίηση του χρήστη ως Client. Η κλάση αυτή κρατάει τα δεδομένα του Client όπως την ip και το όνομα χρήστη του Peer που θα γίνει η σύνδεση, τον αριθμό της θύρας στον οποίο θα συνδεθεί ο χρήστης ως Client όπως και το tcp socket του Client και την κατάσταση στην οποία βρίσκεται. Έπειτα δημιουργείται ένα thread το οποίο παίρνει και ελέγχει τα δεδομένα που αποστέλλει ο Client. Ποιο συγκεκριμένα πραγματοποιεί την σύνδεση του Client με άλλους servers, δημιουργεί το μήνυμα με το οποίο θα ξεκινήσει η επικοινωνία μεταξύ τους και τέλος λαμβάνει τα δεδομένα από άλλους peer που προσπαθούν να επικοινωνήσουν με τον Client.



Επίσης το thread του Client ελέγχει την κατάσταση του και ενημερώνει τον χρήστη αν η κατάσταση επικοινωνίας του με άλλους peers είναι επιτυχής ή όχι. Ο Client μπορεί να επικοινωνεί συνεχώς με τον server όσο το status του server το επιτρέπει. Μόλις ο Client αδρανοποιηθεί, τότε αποστέλλετε ένα μήνυμα στον server το οποίο τον ενημερώνει για την κατάσταση του Client και κλείνει το socket το οποίο είχε ανοίξει για την επικοινωνία μεταξύ τους.

```
class peerMain:
```

Η τελευταία κλάση, η class peerMain, είναι αυτή η οποία είναι υπεύθυνη για όλα τα δεδομένα που δημιουργούνται ώστε να επιτευχθεί η επικοινωνία μεταξύ των peer στο δίκτυο. Η αρχικοποίηση του peer με βάση την οποία θα πραγματοποιηθούν όλες οι ενέργειες για την επικοινωνία μεταξύ των peer περιλαμβάνει την διεύθυνση ip, αριθμό θύρας, το tcp socket, και το udp port για τα αρχικά δεδομένα που θα ανταλλάξει ο εκάστοτε peer με τους υπόλοιπους. Επίσης αρχικοποιείται το udp για τα εισαγωγικά μηνύματα, τις πληροφορίες σύνδεσης ή την αυτόματη σύνδεση του peer, τον server και client του peer καθώς και ένα timer για τον χρόνο που θα χρειαστεί ώστε να γίνουν αυτές οι διαδικασίες. Για τις δραστηριότητες του χρήστη στην εφαρμογή, η κλάση αυτή έχει αρχικοποιημένες μεθόδους που ανάλογα με το τι θα επιλέξει ο χρήστης, θα μπορεί να συνδεθεί στην εφαρμογή με τα στοιχεία του, να δημιουργήσει νέο λογαριασμό, να επικοινωνήσει με άλλους χρήστες καθώς και να τους αναγνωρήσει ανάλογα με το όνομα που έχουν ορίσει ως όνομα χρήστη.

### 2.1.5. Δημιουργία αρχείου Registry (registry.py)

Αυτή η κλάση χρησιμοποιείται για την αποστολή των μηνυμάτων στο registry από κάθε έναν peer που ανήκει στο registry. Για κάθε νέο peer στο registry δημιουργείτε ένα καινούργιο client thread. Ανάλογα με το είδος του μηνύματος, τότε ο χρήστης μπορεί να πραγματοποιήσει συγκεκριμένες ενέργειες στην εφαρμογή όπως, να φτιάξει λογαριασμό, να συνδεθεί και να αποσυνδεθεί και να αναζητήσει διάφορους χρήστες.

Η αρχικοποίηση του client thread στοχεύει στην ενεργοποίηση του timer σύμφωνα με την οποία ο peer είτε θα συνδεθεί για να πραγματοποιήσει επικοινωνία είτε λόγω του timer θα αποσυνδεθεί. Πάλι κατά το registration θα γίνει έλεγχος για το αν η εφαρμογή τρέχει σε Windows ή MacOS.

## Βιβλιογραφία

<https://el.wikipedia.org/wiki/Peer-to-peer>

[https://www.google.com/search?](https://www.google.com/search?q=Συγκεντρωτικά+δίκτυα+P2P&biw=2560&bih=1330&tbm=isch&source=lnms&sa=X&ved=0ahUKEwjn4OCW97XgAhUN4KYKHQuTCzEO_AUICygC#imgrc=IS9qOh5r8L8QwM)

[q=Συγκεντρωτικά+δίκτυα+P2P&biw=2560&bih=1330&tbm=isch&source=lnms&sa=X&ved=0ahUKEwjn4OCW97XgAhUN4KYKHQuTCzEO\\_AUICygC#imgrc=IS9qOh5r8L8QwM:](https://www.google.com/search?q=Συγκεντρωτικά+δίκτυα+P2P&biw=2560&bih=1330&tbm=isch&source=lnms&sa=X&ved=0ahUKEwjn4OCW97XgAhUN4KYKHQuTCzEO_AUICygC#imgrc=IS9qOh5r8L8QwM)

<https://www.pcsteps.gr/100244-ανταλλαγή-αρχείων-πώς-λειτουργεί-p2p-torrent/>

## Πίνακας Εικόνων

[https://www.google.gr/search?q=peer+to+peer&client=opera&hs=wOJ&source=lnms&tbn=isch&sa=X&ved=0ahUKEwic1Z3jjZPhAhVQmbQKHd4kA-EQ\\_AUIDigB&biw=1280&bih=732](https://www.google.gr/search?q=peer+to+peer&client=opera&hs=wOJ&source=lnms&tbn=isch&sa=X&ved=0ahUKEwic1Z3jjZPhAhVQmbQKHd4kA-EQ_AUIDigB&biw=1280&bih=732)