

HY-335b - Δίκτυα Υπολογιστών

Project Spring 2017

“Building Short Anonymous Networks”

Παράδοση: **14/05/2017 | 23:59** ηλεκτρονικά (turnin)

Προφορική εξέταση: **18/05/2017** στην αίθουσα **B.310**

1 . Εισαγωγή

Για την υλοποίηση της εργασίας η προτεινόμενη γλώσσα προγραμματισμού είναι η Python. Άλλες αποδέκτες γλώσσες προγραμματισμού είναι C και C++. Για τις υλοποιήσεις μπορείτε να επεκτείνετε τον κώδικα που δόθηκε στο φροντιστήριο. Η πορεία της υλοποίησής σας θα διακρίνεται σε 2 φάσεις. Αρχικά, χρησιμοποιήσετε είτε τον υπολογιστή σας είτε τα μηχανήματα της σχολής για να κατασκευάσετε και να αποσφαλματώσετε τον κώδικά σας. Στην συνέχεια, θα δοθούν σε κάθε ομάδα 2-3 virtual machines στο σύννεφο για να δοκιμάσετε την εργασία σας σε περιβάλλον πραγματικών συνθηκών και από την λειτουργία του συστήματός σας θα εξάγετε χρήσιμα συμπεράσματα.

2. Τι θα μάθετε

Μέσα από την επίλυση της συγκεκριμένης άσκησης θα αποκτήσετε τριβή στις παρακάτω θεματικές ενότητες:

- Εξοικείωση με βασικά δικτυακά εργαλεία όπως είναι το **ping** και το **traceroute**
- Εκμάθηση τεχνικών **προγραμματισμού sockets**
- Κατανόηση πρότυπου επικοινωνίας **client-server** για ανταλλαγή δεδομένων
- Εισαγωγή στις βασικές έννοιες που διέπουν τη λειτουργία ενός **overlay δικτύου**

3. Στόχος

Στην άσκηση αυτή σας ζητείται να υλοποιήσετε το πρότυπο μιας καινοτόμας υπηρεσίας η οποία 1) θα κρύβει την “ταυτότητα” (IP διεύθυνση) του client από τον server λόγω της χρήσης ενδιάμεσου κόμβου και 2) θα βρίσκει καλύτερα μονοπάτια χρησιμοποιώντας ως κριτήρια την καθυστέρηση (round trip time) ή τον αριθμό των hops, από αυτά που δίνει το standard Internet. Αυτό επιτυγχάνεται με την χρήση relays, όπως έχει αποδειχθεί σε προγενέστερες μελέτες δημιουργώντας ουσιαστικά ένα overlay network.

Η λειτουργία της εφαρμογής σας θα διαρθρώνεται στα παρακάτω στάδια:

- ❖ Αρχικά, ο client θα επεξεργάζεται τα 2 αρχεία, τα ονόματα των οποίων θα δέχεται ως input και από αυτά θα αντλεί πληροφορίες σχετικά με τους end-servers και τους relay nodes που περιέχει το δίκτυο. Ένα παράδειγμα εκτέλεσης σε python για το script client.py θα ήταν το εξής:

```
>> python client.py -e end_servers.txt -r relay_nodes.txt
```

Τα ορίσματα που θα δέχεται το script περιγράφονται στον ακόλουθο πίνακα:

-e <end servers filename>	δίνετε το όνομα του αρχείου με την λίστα των end-servers
-r <relay nodes filename>	δίνετε το όνομα του αρχείου με την λίστα των relay nodes

Στο παραπάνω παράδειγμα το argument -e περιγράφει το αρχείο με τις πληροφορίες για τους end-servers που στην συγκεκριμένη περίπτωση περιέχονται στο αρχείο end_servers.txt και το όρισμα -r περιγράφει το αρχείο με τις πληροφορίες για τους relay nodes που στο συγκεκριμένο παράδειγμα περιέχονται στο αρχείο relay_nodes.txt. Για την δομή των 2 αρχείων παρέχονται οδηγίες στην επόμενη ενότητα.

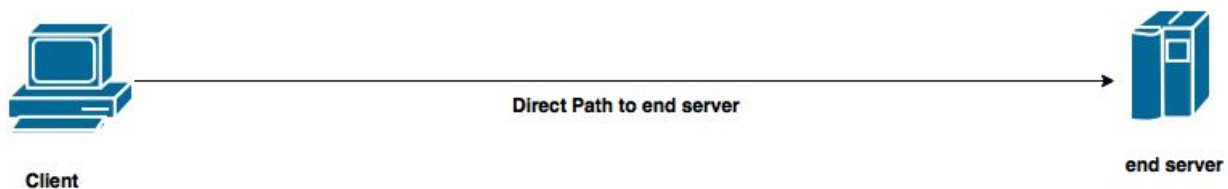
- ❖ Στην συνέχεια, ο client θα ζητάει ως input από το command line το alias του end-server με τον οποίο θα πρέπει να επικοινωνήσει καθώς και το πλήθος των εκτελέσεων του δικτυακού εργαλείου ping για τον υπολογισμό των στατιστικών δεδομένων. Ο server θα πρέπει να περιέχεται μέσα στη λίστα των end-servers που έχει δοθεί αρχικά αλλιώς θα τυπώνεται σχετικό μήνυμα λάθους. Επίσης στο στάδιο αυτό θα εισάγεται το κριτήριο με βάση το οποίο θα επιλέγεται η κατάλληλη διαδρομή ανάμεσα σε client και end-server.

Παράδειγμα εκτέλεσης

```
>> endserver1 120 latency
```

όπου ο client αιτείται να επικοινωνήσει με τον end-server με alias endserver1, χρησιμοποιώντας 120 pings για την εξαγωγή μετρικών και η επιλογή της διαδρομής θα γίνεται χρησιμοποιώντας ως κριτήριο το Round Trip Time (RTT).

- ❖ **Direct Mode:** Σαν πρώτο είδος επικοινωνίας μεταξύ client - end-server, ο client θα κάνει ping κατευθείαν στην IP του end-server χωρίς την παρεμβολή κάποιου ενδιάμεσου κόμβου (relay node) (Σχήμα 1). Το πλήθος των pings καθορίζεται από το input που θα εισάγει ο χρήστης και με βάση τα αποτελέσματα των pings θα εξαχεται η μέση τιμή των RTT και θα περιγράφει το latency για την κατευθείαν επικοινωνία μεταξύ client και end-server. Επιπλέον, από τον client θα εκτελείται traceroute με προορισμό τον end-server ώστε να υπολογίζεται το πλήθος των ενδιάμεσων hops μέχρι τον τελικό προορισμό. Η παραπάνω πληροφορία μαζί με το μέσο RTT είναι 2 μετρικές οι οποίες χαρακτηρίζουν την κατευθείαν επικοινωνία μεταξύ των 2 πλευρών και θα κρατείται στην πλευρά του client.

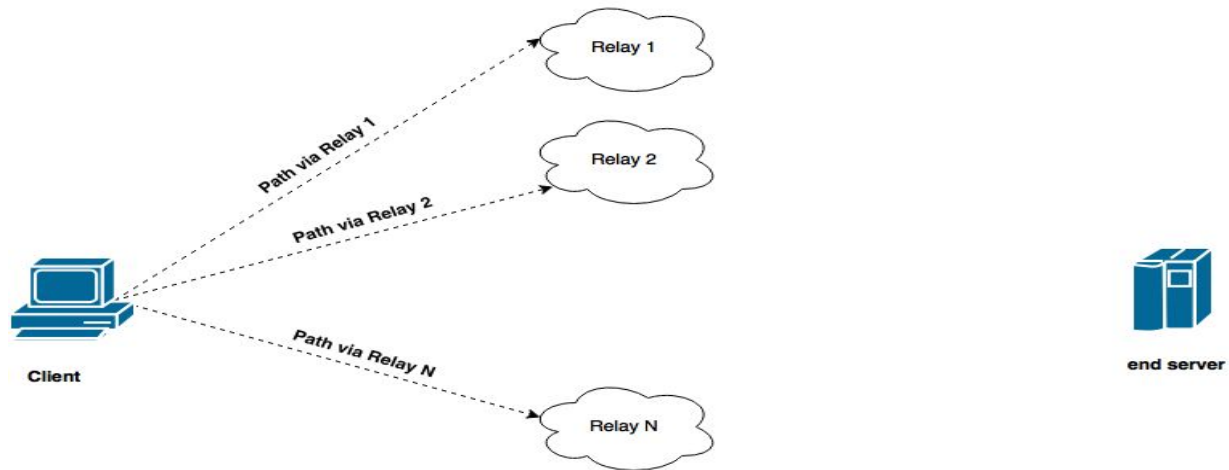


Σχήμα 1

- ❖ **Relay Mode:** Σαν δεύτερος τρόπος επικοινωνίας και βασικό αντικείμενο μελέτης των overlay networks είναι η χρήση ενδιάμεσων κόμβων (relay nodes) ανάμεσα στον client και στον εκάστοτε end-server. Οι πληροφορίες σχετικά με τους relay nodes περιέχονται στο αρχείο relays_list.txt, η δομή του οποίου περιγράφεται στην επόμενη ενότητα. Στην λειτουργία αυτή θα εξετάζονται τα μονοπάτια από τον client προς τους relay nodes αλλά και από τους relay nodes προς τον end-user. Αναλυτικότερα:

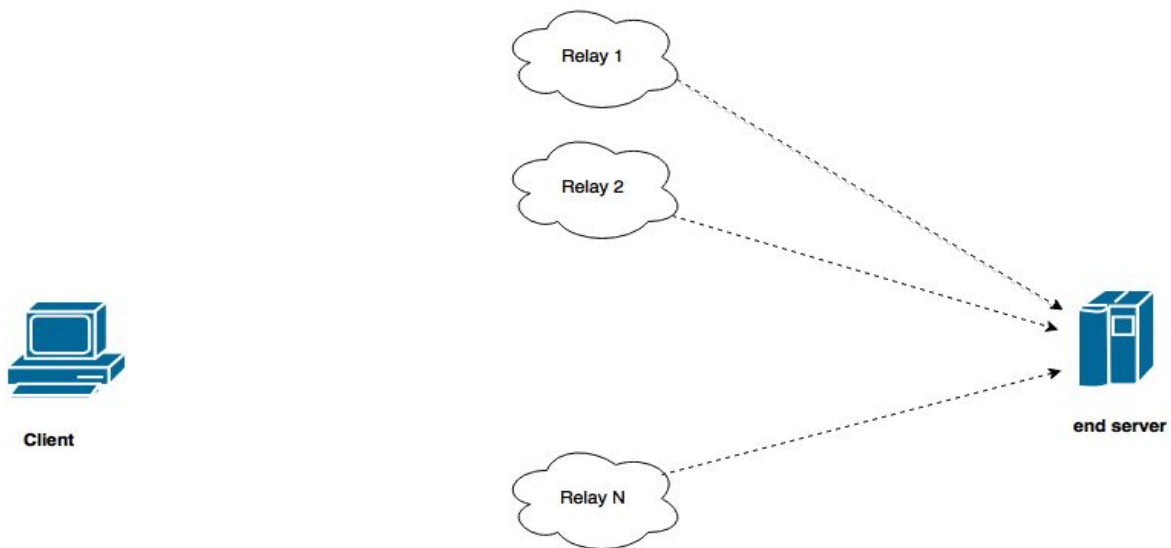
- Από τον client στέλνονται pings προς όλους τους relay nodes (Σχήμα 2). Το πλήθος των pings προς κάθε ενδιάμεσο κόμβο περιγράφεται από το input του χρήστη. Από τα αποτελέσματα των pings, ο client θα υπολογίζει

το μέσο RTT για την κάθε διαδρομή προς κάθε ενδιάμεσο κόμβο. Στην συνέχεια θα εκτελούνται traceroutes προς κάθε ενδιάμεσο κόμβο και θα υπολογίζεται το πλήθος των ενδιάμεσων hops για κάθε διαδρομή από τον client προς όλους τους relay nodes. Η παραπάνω πληροφορία, δηλαδή το μέσο RTT και το πλήθος των hops θα κρατούνται στην μεριά του client.



Σχήμα 2

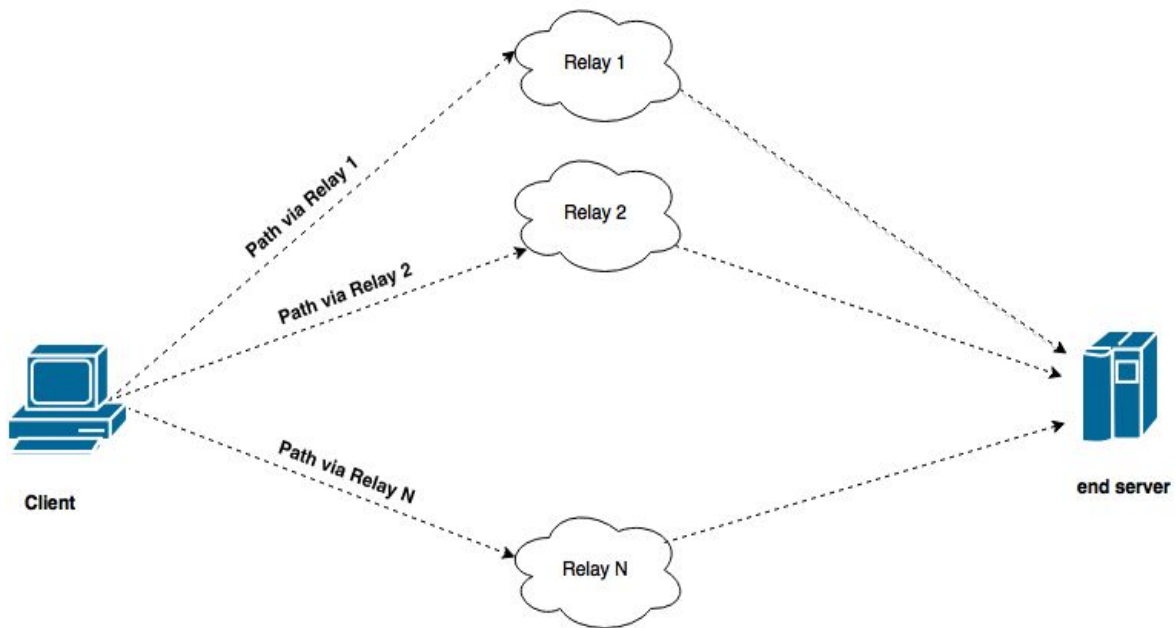
- Ταυτόχρονα με το προηγούμενο στάδιο, θα εκτελείται ίσος αριθμός rings από κάθε relay node προς τον end-user (Σχήμα 3). Με βάση τα αποτελέσματα των rings, θα υπολογίζεται το μέσο RTT για κάθε διαδρομή μεταξύ relay node και end-server. Όμοια με το προηγούμενο βήμα, θα εκτελείται επίσης traceroute από κάθε relay node προς τον end-server και θα εξαχθεί το πλήθος των hops για κάθε διαδρομή. Η πληροφορία σχετικά με το μέσο RTT αλλά και το πλήθος των hops θα πρέπει να κρατείται στην μεριά του client.



Σχήμα 3

- Σαν επόμενο βήμα, ο client θα πρέπει να συνδιάζει τα αποτελέσματα από το monitoring που έγινε στις διαδρομές μέσω των relay nodes κατά τα προηγούμενα στάδια. Πιο συγκεκριμένα, θα πρέπει να υπολογίσει το end-to-end (από τον client προς τον end-server) μέσο RTT και πλήθος hops κάθε διαδρομής που χρησιμοποιεί κάποιον ενδιάμεσο κόμβο (Σχήμα 4). Για παράδειγμα, αν ο client έχει υπολογίσει κατά τις προηγούμενες φάσεις για την διαδρομή client-> relay 1 μέσο χρόνο ίσο με $RTT1$ και πλήθος hops $h1$ και αντίστοιχα για την διαδρομή relay1->end-server μέσο RTT ίσο με $RTT2$ και πλήθος hops ίσο με $h2$, για την διαδρομή client->relay1->end-server (end-to-end) θα έχουμε μέσο **συνολικό** RTT ίσο με $RTT1+RTT2$ και **συνολικό** πλήθος από hops ίσο με $h1+h2$. Η προαναφερθείσα διαδικασία θα γίνει για κάθε διαδρομή από κάθε relay κόμβο. Η παραπάνω συσχέτιση, θα έχει σαν αποτέλεσμα ο client να έχει συνολικά μια εικόνα για όλες τις ενδιάμεσες διαδρομές προς τον end-server σχετικά με το latency και το number of hops.

Προσοχή: ανάμεσα στον client και τον end-server θεωρείστε ότι παρεμβάλλεται **ένας μόνο** relay κόμβος.



Σχήμα 4

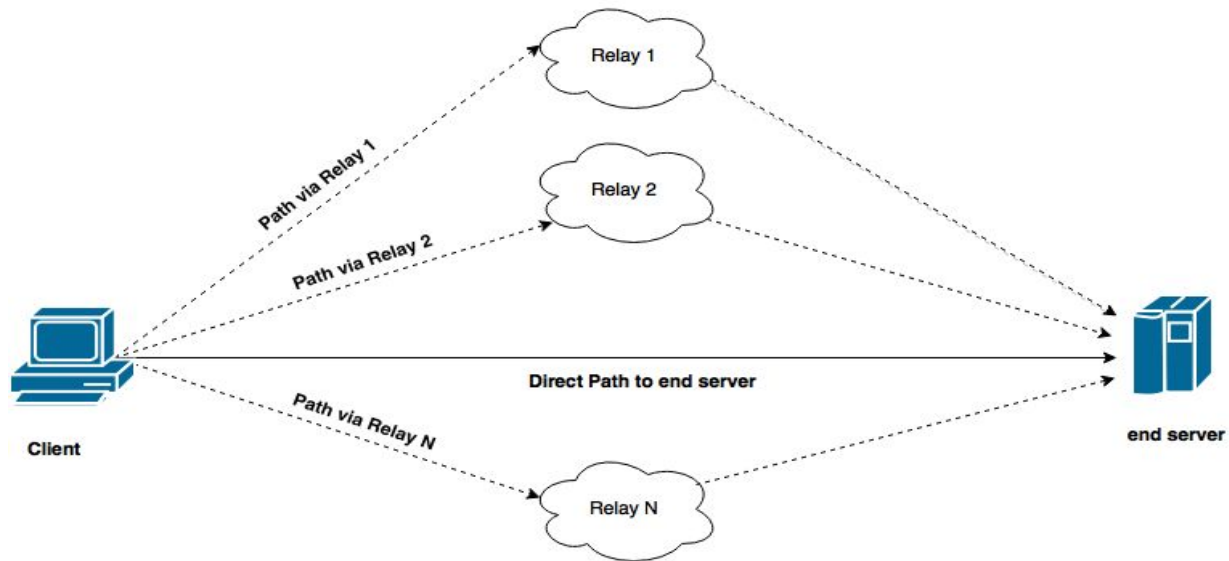
- Στο τελευταίο στάδιο, ο client καλείται με βάση ένα από τα 2 κριτήρια που έχει επιλέξει στο 2ο bullet, να επιλέξει το κατάλληλο μονοπάτι για την επικοινωνία του με τον end-server. Το μονοπάτι αυτό μπορεί να είναι είτε το απευθείας (Direct Mode) είτε κάποιο με χρήση ενδιάμεσου κόμβου (Relay Mode) (Σχήμα 5). Ένα μονοπάτι χαρακτηρίζεται ως καλύτερο αν ελαχιστοποιεί το κριτήριο με βάση το οποίο πρέπει να γίνει η επιλογή (latency ή number of hops). Έτσι, στην περίπτωση που υπάρχουν N ενδιάμεσοι κόμβοι, το συνολικό πλήθος διαδρομών P ισούται με N+1 και έτσι η τελική διαδρομή p^* θα προκύπτει από την παρακάτω σχέση:

$$p^* = \{p \in P: \arg \min C(p)\}$$

όπου $C(p)$ είναι το κριτήριο (latency ή number of hops) για τη εκάστοτε διαδρομή p .

Σε περίπτωση που 2 ή περισσότερα μονοπάτια εμφανίζουν ίδια τιμή για το ένα κριτήριο, θα πρέπει να κάνετε εκ νέου σύγκριση με το άλλο κριτήριο για να λάβετε τελική απόφαση. Αν και πάλι εμφανίζονται παραπάνω από ένα κατάλληλα μονοπάτια, θα επιλέγετε τυχαία.

Με βάση το μονοπάτι που επιλέξατε, **θα πρέπει να κατεβάσετε το αρχείο target-file** όπως περιγράφεται μέσα στο αρχείο files2download.txt και θα υπολογίζετε τον χρόνο που απαιτήθηκε για να κατέβει ολόκληρο το αρχείο. Έτσι αρχικά η εφαρμογή σας θα ενημερώνει τον client για την διαδρομή που εν τέλει επιλέχθηκε προς τον συγκεκριμένο end server και στη συνέχεια θα ζητάει ως input από command line το url του αρχείου που επιθυμείτε να κατεβάσετε προκειμένου να ολοκληρωθεί η διαδικασία.



Σχήμα 5

Υλοποίηση Client:

Ο client θα παίρνει ως όρισμα το όνομα ενός αρχείου που θα περιέχει τις Domain διευθύνσεις των end-servers. Το format του αρχείου θα είναι ως εξής:

Domain Address, alias

Για παράδειγμα

www.google.com, google

www.uoc.gr, uoc

.

.

Με άλλα λόγια θα έχετε μια λίστα με διευθύνσεις IP και ψευδώνυμα που θα περιγράφουν τους end servers με τους οποίους θα επιτρέπεται η επικοινωνία.

Το δεύτερο όρισμα που θα δέχεται το πρόγραμμα που εκτελείται στην πλευρά του client θα είναι το όνομα του αρχείου που θα περιέχει τις IP διευθύνσεις των relay κόμβων καθώς επίσης και την πόρτα που προσφέρουν για επικοινωνία με τους clients, δηλαδή το format του αρχείου θα είναι

alias, IP Address, port number

Για παράδειγμα, ένα αρχείο με όνομα relays_list.txt θα είχε την ακόλουθη μορφή:

```
my_relay1, 18.18.18.18 , 1025
my_relay2, 118.118.118.118 , 1026
```

Η παραπάνω σύνταξη εννοεί ότι ο relay node με όνομα my_relay1 έχει IP διεύθυνση 18.18.18.18 και επιτρέπει σε οποιονδήποτε client να συνδεθεί στην πόρτα με αριθμό 1025 κ.ο.κ.

Μετά την παραπάνω διαδικασία, το script στην πλευρά του client θα ζητάει ως input από command line το alias του end-server με τον οποίο θα υπάρξει επικοινωνία, το πλήθος των pings που θα χρησιμοποιηθούν από την εφαρμογή σας και το κριτήριο που θα χρησιμοποιηθεί για την επιλογή της βέλτιστης διαδρομής (number of iterations).

Η λειτουργικότητα του script περιγράφεται με σαφήνεια από την προηγούμενη ενότητα. Συνοπτικά, οι λειτουργίες του client είναι:

- Αποστολή pings προς τον end-server, ο αριθμός των οποίων προσδιορίζεται από το input από τον χρήστη και traceroute και κατόπιν εξαγωγή του μέσου RTT και number of hops που χαρακτηρίζουν την direct επικοινωνία client-end server.
- Αποστολή εντολής προς κάθε relay node, για να εκτελεστούν pings προς τον end-server ο αριθμός των οποίων προσδιορίζεται από το input του χρήστη και 1 traceroute και

κατόπιν εξαγωγή του μέσου RTT και number of hops που χαρακτηρίζουν την ζεύξη relay node-end server. Τα παραπάνω αποτελέσματα πρέπει να λαμβάνονται από τον client μέσω sockets.

- Αποστολή pings προς κάθε relay node, ο αριθμός των οποίων προσδιορίζεται από το input από τον χρήστη και traceroute και κατόπιν εξαγωγή του μέσου RTT και number of hops που χαρακτηρίζουν την επικοινωνία client-relay node.
- Συνδυασμός των παραπάνω πληροφοριών και υπολογισμός του συνολικού μέσου RTT και number of hops για κάθε διαδρομή μέσα από κάθε relay node.
- Επιλογή της κατάλληλης διαδρομής με βάση το κριτήριο που ζητήθηκε.
- Αίτημα λήψης αρχείου από τον end-server και υπολογισμός χρονικού διαστήματος που απαιτήθηκε.

Να σημειωθεί ότι για την επικοινωνία μεταξύ client ↔ end-server και client ↔ relay-node θα πρέπει να χρησιμοποιηθούν **TCP sockets**.

Για την **ταυτόχρονη** εκτέλεση πολλαπλών διεργασιών, **θα πρέπει να χρησιμοποιηθούν threads**.

Υλοποίηση Relay-node:

Η λειτουργικότητα του script που θα εκτελείται στην πλευρά του κάθε relay node θα πρέπει να εξυπηρετεί τις ακόλουθες λειτουργίες:

- ➔ Κάθε relay node θα ανοίγει socket στην πόρτα που επιτρέπει επικοινωνία με τον client, όπως θα περιγράφονται στο αρχείο relay_nodes.txt.
- ➔ Ο κάθε relay node θα είναι υπεύθυνος, μετά από υπόδειξη του client, να στέλνει αριθμό pings ίσο με number of iterations προς τον επιλεγμένο end-server και να υπολογίζει το μέσο χρόνο RTT. Επίσης εκτελώντας traceroute, θα πρέπει να υπολογίζεται το πλήθος των hops από τον relay node προς τον end-user. Οι 2 αυτές τιμές θα πρέπει να αποστέλονται με χρήση socket στον client και να αποθηκεύονται εκεί.

Σημείωση: στα scripts σας φροντίστε να εκτυπώνονται ενδεικτικά μηνύματα που θα υποδηλώνουν την σωστή λειτουργία της εφαρμογής σύμφωνα με τις απαιτήσεις που σας ζητήθηκαν.

4. Παραδοτέα

4.1 Οδηγίες σχετικά με τα παραδοτέα

Μέσα στον φάκελο που θα παραδώσετε θα περιέχονται τα ακόλουθα:

- Ένα αρχείο που περιέχει την υλοποίηση της λειτουργικότητας του client με όνομα **client.py** ή **client.c** ή **client.cpp** ανάλογα με την γλώσσα προγραμματισμού που επιλέξατε.
- Ένα αρχείο που περιέχει την υλοποίηση της λειτουργικότητας του relay node με όνομα **relay_node.py** ή **relay_node.c** ή **relay_node.cpp** ανάλογα με την γλώσσα προγραμματισμού που επιλέξατε.
- Μια αναφορά σε μορφή pdf που θα περιέχει περιγραφή της υλοποίησής σας καθώς επίσης και σχολιασμό των αποτελεσμάτων.
- Ένα αρχείο με όνομα **readme.txt** που θα περιέχετε ένα παράδειγμα κλήσης σε καθένα από τα scripts που θα παραδώσετε.
- Το αρχείο με τις πληροφορίες για τους end-servers με όνομα **end_servers.txt**.
- Το αρχείο με τις πληροφορίες για τους relay nodes με όνομα **relay_nodes.txt**.
- Ένα αρχείο με όνομα **members.txt** που θα αναφέρει το ονοματεπώνυμο και το ΑΜ και των 2 μελών της ομάδας.

4.2 Οδηγίες TURNIN

Παραδίδετε το project εκτελώντας την εντολή:

turnin project2017@hy335b your_folder

Ένα μέλος από κάθε ομάδα θα πρέπει να κάνει turnin την παραδοτέα εργασία.

Όλα τα project θα ελεγχθούν για αντιγραφές με την χρήση του MOSS.

4.3 Προφορική Εξέταση

Μετά την παράδοση του project θα εξεταστείτε προφορικά πάνω στην εργασία σας. Είναι υποχρεωτικό να παρευρίσκονται όλα τα μέλη της ομάδας στην εξέταση.

4.4 Βαθμολογία

Η βαρύτητα της άσκησης είναι 20% επί του συνολικού βαθμού.

Για απορίες, ερωτήσεις και οποιαδήποτε διευκρίνιση χρησιμοποιήστε το forum από το Moodle του μαθήματος.

Καλή επιτυχία!