

ΑΝΑΦΟΡΑ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗΣ ΑΣΚΗΣΗΣ

Στην ασκηση μας δημιουργήσαμε 2 αρχεία python το ένα είναι το client.py και το άλλο το relay_node.py

CLIENT.PY:

Αρχικά δημιουργούμε 6 αρχεία.

Όταν γίνεται ping από το client στους endservers:

- 1) Rtt.txt: Γράφουμε τα αποτελέσματα από την συνάρτηση ping
- 2) Hops.txt: Γράφουμε τα αποτελέσματα από την συνάρτηση traceroute.

Όταν γίνεται ping από τον relay_nodes στους end servers:

- 1) Relay_Rtt.txt: Γράφουμε τα αποτελέσματα από την συνάρτηση ping
- 2) Relay_Hops.txt: Γράφουμε τα αποτελέσματα από την συνάρτηση traceroute

Αυτά τα 2 αρχεία τα λαμβάνει ο client από τον relay_node μέσω socket.

Όταν γίνεται ping από το client στους relay_nodes:

- 1) My_Relay_Rtt.txt: Γράφουμε τα αποτελέσματα από την συνάρτηση ping .
- 2) My_Relay_Hops.txt: Γράφουμε τα αποτελέσματα από την συνάρτηση traceroute.

- Καλούμε την συνάρτηση ping δύο φορές. Τη πρώτη φορά με το αρχείο end_servers.txt όπου κάνουμε ping με το όνομα του end_server και την δεύτερη με το αρχείο relays_list.txt όπου κάνουμε ping με το ip του Relay_Node.

- Καλούμε την συνάρτηση traceroute δύο φορές. Τη πρώτη φορά με το αρχείο end_servers.txt όπου κάνουμε traceroute με το όνομα του end_server και την δεύτερη με το αρχείο relays_list.txt όπου κάνουμε traceroute με το ip του Relay_Node.

- Καλούμε την συνάρτηση Results με παραμέτρους τα 6 αρχεία που αναφέραμε πριν έτσι ώστε στο αρχείο Rtt.txt και το Hops.txt να περιέχει την πρόσθεση των αποτελεσμάτων μεταξύ Client - Relay_node και Relay_node - End_Server.

- Ζητάμε από τον χρήστη το κριτήριο που θα χρησιμοποιηθεί για την επιλογή της βέλτιστης διαδρομής. Ανοίγοντας το αρχείο end_servers.txt για κάθε γραμμή εκτελούμε την συνάρτηση Find_Best στην οποία διαβάζοντας από το αρχείο Rtt.txt και Hops.txt αντίστοιχα γράφουμε σε ένα string (Res) το

καλύτερο χρόνο Rtt – καλύτερο μονοπάτι Hop αντίστοιχα. Για κάθε end_server γράφουμε σε ένα άλλο αρχείο BEST.txt το string(Res).

- Καλούμε την συνάρτηση Sent_Url με παραμέτρους URL.txt (περιέχει τα url που μας δώθηκαν από τους βοηθούς) και BEST.txt. Για κάθε γραμμή του αρχείου BEST.txt ελέγχουμε αν το Rtt / Hop είναι αρνητικό που σημαίνει ότι ο συγκεκριμένος server είχε 100% packet_loss.

EXTRA:

- 1) Τα συνολικά αποτελέσματα από Rtt / Hop τα γράφουμε σε αρχεία ώστε να μπορεί να τα διακρίνει ποιο εύκολα ο χρήστης.
- 2) Σε περίπτωση που υπάρχουν ίδια Rtt / Hop στο αρχείο BEST.txt διαλέγεται το πρώτο ίδιο Rtt / Hop
- 3) Στην συνάρτηση Sent_Url ελέγχουμε αν το όνομα του server από το αρχείο BEST.txt είναι ίδιο με το όνομα του server από το URL. Δηλαδή αν για κάποιο λόγο το URL.txt περιέχει τα URL ανακατεμένα σε σχέση με τα ονόματα των servers το πρόγραμμά μας θα δουλέψει σωστά.

RELAY_NODE.PY:

Αρχικά παίρνουμε το αρχείο Results.txt μέσω socket από τον client. Χωρίζουμε αυτό το αρχείο στα εξής αρχεία:

- 1) Relays_List.txt
- 2) Server_end_servers.txt

-Καλούμε την συνάρτηση ping από τον Relay_node στον End_Server.

-Καλούμε την συνάρτηση traceroute από τον Relay_node στον End_Server.

Τα αποτελέσματα των παραπάνω 2 συναρτήσεων τα αποθηκεύουμε στην παράμετρο Result την οποία την έχουμε δώσει στην συνάρτηση ping. Αυτήν την παράμετρο την χωρίζουμε σε δύο strings(result_RTT / result_HOP) τα οποία τα στέλνουμε πίσω στον client μέσω socket.

ΑΠΟΤΕΛΕΣΜΑΤΑ:

Επειδή οι relay_nodes είναι ένα Hop παραπάνω από ότι client-end_servers στην πρόσθεση Rtt / Hop ο χρόνος / μονοπάτι του relay_node είναι ίσος με τον χρόνο/ μονοπάτι από client – end server και τον χρόνο/ μονοπάτι (1) από relay_node-end_Server.

Στο URL.txt το:

<https://brand.ucla.edu/wp-content/uploads/2013/09/ucla-identity-logo-image.jpg>

Λόγω ασφάλειας της σελίδας δεν μας αφήνει να πάρουμε την εικόνα