M162 - Τεχνικές Ιδιωτικότητας
2019 Project, Part A

Αυτό είναι το πρώτο μέρος του project που θα δοθεί **αντί** γραπτής εξέτασης. Η ημερομηνία παράδοσης είναι η Παρασκευή 14/2/2020. Η εργασία είναι **ατομική** : μπορείτε να συζητήσετε με τους συμφοιτητές σας τη μεθοδολογία επίλυσης (το οποίο και σας προτείνω να κάνετε), αλλά σε καμία περίπτωση δεν πρέπει να ανταλλάσσετε κώδικα. Λίγες μέρες μετά την παράδοση θα ακολουθήσει προφορική εξέταση πάνω στην εργασία (και μόνο).

Η εργασία μπορεί να υλοποιηθεί σε οποιαδήποτε γλώσσα προγραμματισμού επιθυμείτε, υπό την προϋπόθεση:

- Η γλώσσα να είναι ευρέως χρησιμοποιούμενη (ρωτήστε με αν αμφιβάλλετε)
- Το πρόγραμμα να κάνει compile από command line και να εκτελείται σε περιβάλλον Ubuntu linux, χρησιμοποιώντας compiler και libraries από το Ubuntu repository (όχι proprietary compilers).
- Ο κώδικας θα πρέπει να συνοδεύεται από σαφείς οδηγίες εκτέλεσης από command line, αν δεν μπορώ να τον εκτελέσω δε θα μπορώ να βαθμολογήσω την εργασία.

Η παράδοση των εργασιών γίνεται μέσω email. Μαζί με τον κώδικα θα πρέπει να παραδώσετε και ένα report που ζητείται στην τελευταία ενότητα, και οτιδήποτε επεξηγήσεις θεωρείτε σκόπιμες.

1 Γενικά

Σκοπός της εργασίας είναι η ανάλυση του μοντέλου ανώνυμης επικοινωνίας Crowds μέσω **προσομοίωσης**. Αντί δηλαδή να υλοποιήσετε το σύστημα, θα κάνετε προσομοίωση της εκτέλεσής του, και θα παράγετε σε κάθε εκτέλεση τις πληροφορίες που γίνονται γνωστές στον αντίπαλο, με βάση ένα συγκεκριμένο μοντέλο αντιπάλου. Στη συνέχεια θα χρησιμοποιήσετε το εργαλείο F-BLEAU για να εκτιμήσετε την πιθανότητα επιτυχίας ενός αντιπάλου που προσπαθεί να βρει τον αποστολέα του μηνύματος. Θα πρέπει να μελετήσετε την πιθανότητα αυτή σε διαφορετικά instances του πρωτοκόλλου και να τη συγκρίνετε με την ανάλυση που έχουμε δει στο μάθημα.

Κάθε σύστημα που θα προσομοιώσετε αποτελείται από m χρήστες (δίνεται ως παράμετρος). Κάθε χρήστης μπορεί να επικοινωνήσει μόνο με ένα υποσύνολο των υπολοίπων χρηστών, με βάση ένα **γράφο δικτύου**, του οποίου οι κορυφές είναι χρήστες και κάθε ακμή δηλώνει ότι οι αντίστοιχοι χρήστες μπορούν να επικοινωνήσουν. Ο **πίνακας γειτνίασης** του γράφου (πληροφορίες από wikipedia) θα δίνεται ως είσοδος υπό μορφή αρχείου, κάθε γραμμή του πίνακα σε ξεχωριστή γραμμή του αρχείου, με τις στήλες να χωρίζονται με κενό. Πχ ο πίνακας

```
0 1 0
1 0 1
0 1 0
```

περιγράφει ένα γράφο 3 χρηστών ενωμένων σε "γραμμή". Από το μέγεθος του πίνακα εξάγετε και τον αριθμό των χρηστώ m. Ο γράφος θα είναι πάντα μη κατευθυνόμενος και δε θα περιέχει ακμές από μια κορυφή προς τον εαυτό της.

Από τους m χρήστες, οι c θεωρούνται ότι ελέγχονται από τον αντίπαλο (corrupted). Τα ids των corrupted χρηστών δίνονται ως παράμετρος.

Τέλος η αρχική γνώση του αντιπάλου είναι μια πιθανοτική κατανομή των $m - c$ έντιμων χρηστών, η οποία δίνεται σε ένα αρχείο με μία μόνο γραμμή, η οποία περιλαμβάνει τις πιθανότητες χωρισμένες με κενό. Πχ το αρχείο

```
0.5 0.25 0.25
```

περιγράφει ότι ο πρώτος χρήστης έχει 50% πιθανότητα να στείλει μήνυμα, ενώ οι άλλοι δύο από 25%.

2 Προσομοίωση

Το πρώτο πρόγραμμα που θα φτιάξετε θα κάνει προσομοίωση του Crowds και θα παράγει την πληροφορία που το σύστημα παρέχει στον αντίπαλο. Το πρόγραμμα `simulate` θα παίρνει τις εξής παραμέτρους από command line:

```
simulate <phi> <graph-file> <corrupted-file> <users-file> <broken-paths> <fix-strategy>
```

Η πρώτη παράμετρος είναι το σύστημα που πρέπει να προσομοιώσετε. Η δεύτερη είναι το αρχείο που περιέχει το γράφο του δικτύου. Η τρίτη είναι ένα αρχείο με τα ids των corrupted users (0-based ints, one per line). Τέλος, το "users-file" περιλαμβάνει μια γραμμή για κάθε εκτέλεση που πρέπει να προσομοιώσετε, η οποία περιέχει το id του χρήστη που στέλνει το μήνυμα. Πχ το αρχείο

```
0
3
3
```

σημαίνει ότι πρέπει να προσομοιώσετε 3 εκτελέσεις, όπου στην πρώτη αποστολέας είναι ο χρήστης 0, ενώ στις άλλες δύο ο χρήστης 3.

Το πρόγραμμά σας θα πρέπει **στην standard έξοδο** (όχι σε κάποιο αρχείο) να παράγει μια γραμμή για κάθε εκτέλεση (δηλαδή για κάθε γραμμή του "users-file") η οποία να περιέχει την πληροφορία που παίρνει ο αντίπαλος κατά τη συγκεκριμένη εκτέλεση. Την ακριβή μορφή της εξόδου την ορίζετε όπως επιθυμείτε ανάλογα με τη λειτουργία του κάθε συστήματος (πχ μπορεί να παράγετε το string "det 4" το οποίο να σημαίνει ότι ο χρήστης 4 έγινε detected). Θα πρέπει όμως να περιγράψετε τη μορφή της εξόδου που χρησιμοποιήσατε στο report που θα συνοδεύει τον κώδικα.

Το Crowds είναι πιθανοτικό σύστημα, οπότε προφανώς η έξοδος μπορεί να είναι διαφορετική σε κάθε προσομοίωση. Η πιθανότητα προώθησης δίνεται στην είσοδο. Σε κάθε προώθηση ένας χρήστης επιλέγει τυχαία (ομοιόμορφη κατανομή) ανάμεσα στους χρήστες με τους οποίους συνδέεται στο γράφο, συμπεριλαμβανομένου πάντα και του εαυτού του (όπως ορίζεται στο πρωτόκολλο). Ο αντίπαλος **δεν** ελέγχει όλο το δίκτυο άλλα μόνο τους corrupted χρήστες και τον τελικό web server.

Τέλος, ο αντίπαλος μπορεί να "καταστρέψει" ένα μονοπάτι που περνάει από έναν corrupted κόμβο, σταματώντας απλά την προώθηση των μηνυμάτων. Η παράμετρος `broken-paths` είναι ένας ακέραιος που καθορίζει το μέγιστο αριθμό των μονοπατιών που καταστρέφει ο αντίπαλος, στην ίδια εκτέλεση του πρωτοκόλλου. Αν είναι 0 το πρωτόκολλο εκτελείται κανονικά, αν είναι 1 ο αντίπαλος καταστρέφει το πρώτο μονοπάτι που θα περάσει από corrupted κόμβο, κλπ.

Τα μονοπάτια που καταστρέφονται από τον αντίπαλο, επισκευάζονται με βάση την στρατηγική που ορίζεται στο `fix-strategy`. Η παράμετρος αυτή παίρνει 2 τιμές: "last-honest" σημαίνει ότι ο τελευταίος τίμιος χρήστης ξαναρχίζει το πρωτόκολλο προωθώντας το μήνυμα σε κάποιον νέο χρήστη, "initiator" σημαίνει ότι ο αρχικός χρήστης ξαναρχίζει το πρωτόκολλο με τον ίδιο τρόπο.

3 Υπολογισμός επιτυχίας του αντιπάλου

Η ανάλυση της ανωνυμίας του πρωτοκόλλου θα γίνει δίνοντας το αποτέλεσμα μεγάλου όγκου προσωμοιώσεων στο εργαλείο F-BLEAU (`https://github.com/gchers/fbleau`). Το εργαλείο αυτό προσωμοιώνει έναν αντίπαλο που είναι βέλτιστος όσο ο αριθμός των δεδομένων αυξάνει, και επιστρέφει μια εκτίμηση της πιθανότητας επιτυχίας του βέλτιστου αντιπάλου.

Η χρήση του εργαλείου είναι απλή, και η εκμάθησή του αποτελεί μέρος της εργασίας. Για τη χρήστη του F-BLEAU και μόνο μπορείτε να συνεργάζεστε ελεύθερα, να συζητάτε οτιδήποτε έχει να κάνει με το εργαλείο αυτό στο piazza, κλπ. Βοήθεια θα δοθεί και από τον διδάσκοντα στο piazza.

4 Ανάλυση αποτελεσμάτων

Τέλος, χρησιμοποιώντας τα προγράμματα που υλοποιήσατε, θα πρέπει να γράψετε ένα report (μορφή pdf) στο οποίο να κάνετε μια σύντομη ανάλυση των αποτελεσμάτων. Θα χρησιμοποιήσετε παραδείγματα του συστήματος δική σας επιλογής (αριθμός χρηστών, γράφος δικτύου, αρχική κατανομή, κλπ). Χρησιμοποιώντας τα προγράμματα που δημιουργήσατε θα αναλύσετε τα παραδείγματα αυτά και θα περιγράψετε τα αποτελέσματα. Θα πρέπει επίσης να συγκρίνετε τα αποτελέσματα αυτά με τις σχετικές έννοιες από τη θεωρία ποσοτικής ροής πληροφορίας που είδαμε στο μάθημα, και να διαπιστώσετε αν τα εμπειρικά αποτελέσματα συμπίπτουν με τη θεωρητική ανάλυση. Επίσης θα πρέπει να παράγετε γραφικές παραστάσεις που να δείχνουν την πιθανότητα επιτυχίας του αντιπάλου ως συνάρτηση κάποιας παραμέτρου του συστήματος (δική σας επιλογής).

Ενδεικτικά, μπορείτε να αναλύσετε πώς επηρρεάζει την ανωνυμία:

- το φ (για ομοιόμορφη ή μη αρχική κατανομή)
- Η αρχική κατανομή
- Ο αριθμός corrupted χρηστών
- Η καταστροφή μονοπατιών όταν επισκευάζονται από τον τελευταίο χρήστη
- Η καταστροφή μονοπατιών όταν επισκευάζονται από τον αρχικό χρήστη
- Ο γράφος δικτύου
- . . .

Το report που θα φτιάξετε δε χρειάζεται να είναι τεράστιο (10-20 σελίδες αρκούν). Θα πρέπει όμως να περιέχει αρκετές πληροφορίες ώστε να μπορεί να το κατανοήσει ένας τρίτος αναγνώστης (με εμπειρία στην ανώνυμη επικοινωνία), πχ να περιέχει μια σύντομη περιγραφή των συστημάτων που αναλύονται, όχι σκέτα νούμερα.