

ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΕΡΓΑΣΙΕΣ ΣΤΑ ΠΛΑΙΣΙΑ ΤΟΥ ΜΑΘΗΜΑΤΟΣ ΕΠΙΛΟΓΗΣ ΧΕΙΜΕΡΙΝΟΥ
ΕΞΑΜΗΝΟΥ «ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΔΙΚΤΥΩΝ ΚΙΝΗΤΩΝ ΕΠΙΚΟΙΝΩΝΙΩΝ»
ΔΙΔΑΣΚΩΝ ΜΑΘΗΜΑΤΟΣ: ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ ΧΡΗΣΤΟΣ ΒΕΡΥΚΟΥΚΗΣ

ΕΡΓΑΣΙΑ 1) Προσομοίωση δικτύου με NS-3

Αυτή η εργασία, εστιάζεται στη δημιουργία μιας προσομοίωσης δικτύου χρησιμοποιώντας το [NS-3](#). Η προσομοίωση χωρίζεται σε δύο μέρη (1) προσομοίωση μιας απλής σύνδεσης σημείου-προς-σημείο μεταξύ δύο κόμβων με UDP echo και (2) προσομοίωση ενός πιο πολύπλοκου δικτύου με δυναμική δρομολόγηση, δημιουργία ροών πακέτων, εισαγωγή αποτυχίας συνδέσεων, και ανάλυση απώλειας πακέτων.

Μέρος 1: Προσομοίωση Σημείου-Προς-Σημείο: Δημιουργήστε μια προσομοίωση με 2 κόμβους που συνδέονται μέσω σύνδεσης point to point. Ορίστε τα χαρακτηριστικά της σύνδεσης (ταχύτητα μετάδοσης δεδομένων και καθυστέρηση) και αντιστοιχίστε διευθύνσεις IPv4 στις διεπαφές των κόμβων. Εγκαταστήστε δύο servers UDP echo στον κόμβο 1 σε διαφορετικές θύρες. Δημιουργήστε δύο πελάτες UDP echo στον κόμβο 0. Εκτελέστε την προσομοίωση και καταγράψτε τα δεδομένα ροής (πακέτα, καθυστέρηση, εύρος ζώνης).

Μέρος 2: Προσομοίωση Δυναμικής Δρομολόγησης: Δημιουργήστε ένα δίκτυο με 5 κόμβους με την παρακάτω τοπολογία:

1. Κόμβος 0 με Κόμβο 1
2. Κόμβος 1 με Κόμβο 2
3. Κόμβος 0 με Κόμβο 3
4. Κόμβος 3 με Κόμβο 4
5. Κόμβος 4 με Κόμβο 2

Ορίστε τις ταχύτητες μετάδοσης δεδομένων και τις καθυστερήσεις για κάθε σύνδεση. Η καθυστέρηση πρέπει να είναι ρυθμιζόμενη μέσω παραμέτρων από τη γραμμή εντολών. Εγκαταστήστε το internet stack και ορίστε τις διευθύνσεις IP για κάθε διεπαφή. Ενεργοποιήστε τη δυναμική δρομολόγηση και εισάγετε συμβάντα αποτυχίας συνδέσεων (π.χ., αποτυχία της σύνδεσης μεταξύ Κόμβου 0 και Κόμβου 1 στο 2ο δευτερόλεπτο και επαναφορά στο 3ο δευτερόλεπτο). Δημιουργήστε ροές) πακέτων CBR (Constant Bit Rate) από τον Κόμβο 0 προς τον Κόμβο 2 και από τον Κόμβο 0 προς τον Κόμβο 3 με διαφορετικές ταχύτητες και χρόνους εκκίνησης. Εγκαταστήστε packet sinks στους κόμβους προορισμούς για τη λήψη της κίνησης. Εφαρμόστε μια γρήγορη “επικοινωνία” μεταξύ των κόμβων για να ρυθμιστούν οι πίνακες διευθύνσεών τους, ώστε να μπορούν να επικοινωνούν πιο γρήγορα και χωρίς καθυστερήσεις κατά τη διάρκεια της κύριας προσομοίωσης



Χρησιμοποιήστε την παρακολούθηση ροής (Flow Monitoring) για να καταγράψετε στατιστικά όπως μεταδοθέντα πακέτα, απωλεσθέντα πακέτα και εύρος ζώνης. Υπολογίστε το ποσοστό απώλειας πακέτων σε τακτά χρονικά διαστήματα και καταγράψτε τα αποτελέσματα σε ένα αρχείο. Στο τέλος της προσομοίωσης, συνοψίστε τον συνολικό αριθμό των πακέτων (μεταδοθέντα, απωλεσθέντα και ποσοστό απώλειας).

Σημειώσεις:

Χρησιμοποιήστε ως γλώσσα εκτέλεσης τη C++ ή python μαζί με τα python bindings για το ns-3. Συνίσταται η χρήση του docker για την εγκατάσταση ενός container με όλα τις απαραίτητες εξαρτήσεις για το NS-3.

Dockerfile για το container με ubuntu & NS-3

```
FROM ubuntu:22.04
ENV DEBIAN_FRONTEND=noninteractive

RUN apt-get update && apt-get install -y \
    software-properties-common \
    g++ \
    python3 \
    python3-pip \
    git \
    nano \
    vim \
    wget \
    sudo \
    net-tools \
    iputils-ping \
    cmake \
    build-essential \
    pkg-config

#ns-3 dependencies
RUN apt-get install -y \
    libsqlite3-dev \
    libxml2-dev \
```



```
libc6-dev \  
python3-setuptools \  
libgtk2.0-0 \  
qt5-default \  
gsl-bin \  
libgsl-dev \  
libgsl23 \  
libgslcblas0 \  
openmpi-bin \  
openmpi-common \  
openmpi-doc \  
libopenmpi-dev \  
autoconf \  
cvs \  
bzip2 \  
unrar \  
unrar-free \  
unrar \  
p7zip-full \  
p7zip-rar \  
libgcrypt20-dev \  
clang  
  
# GCC10  
RUN add-apt-repository -y ppa:ubuntu-toolchain-r/test && \  
apt-get update && \  
apt-get install -y gcc-10 g++-10 && \  
update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-10 10 && \  
update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-10 10  
  
ENTRYPOINT ["/bin/bash"]
```



ΕΡΓΑΣΙΑ 2) Προσομοίωση κινητών δικτύων με Python βάσει containers

Ο στόχος αυτής της εργασίας είναι να σχεδιαστεί και να εφαρμοστεί μια προσομοίωση ενός cellular network χρησιμοποιώντας Docker containers. Ζητείται να αναπτυχθεί η προσομοίωση συσκευών χρήστη και εφαρμογών με δυναμική κατανομή πόρων άλλα και δυναμικό placement των containers με μηχανισμούς παράδοσης (handover mechanism). Τέλος η οπτικοποίηση της τοπολογίας του δικτύου για και η ανάλυση της απόδοσης του συστήματος.

Μέρος 1: Προσομοίωση της υποδομής του δικτύου από containers, όπου το καθένα θα λειτουργεί ως συσκευή χρήστη που δημιουργεί συγκεκριμένους τύπους κίνησης. Δημιουργήστε εικόνες Docker για διαφορετικές εφαρμογές (π.χ. streaming service κ.α). Τα containers θα πρέπει να μιμούνται μία πραγματική κίνηση και χρήση του δικτύου.

Δημιουργήστε σταθμούς βάσης ως δίκτυα Docker χρησιμοποιώντας Python και Docker SDK. Κάθε δίκτυο Docker θα αντιπροσωπεύει έναν μοναδικό σταθμό βάσης στο προσομοιωμένο δίκτυο.

Μέρος 2: Αναπτύξτε ένα python script που παρακολουθεί τα containers που είναι συνδεδεμένα σε κάθε base station. Εφαρμόστε μια πολιτική προγραμματισμού που εκχωρεί το εύρος ζώνης με βάση τις απαιτήσεις της εφαρμογής και το τρέχον φορτίο στο σταθμό βάσης. Ορίστε ένα τυχαίο μοντέλο κινητικότητας που αλλάζει τη «θέση» των container με την πάροδο του χρόνου. Υπολογίστε την απόσταση μεταξύ της τρέχουσας θέσης του container και κάθε σταθμού βάσης για να αποφασίσετε σε ποιο σταθμό βάσης θα συνδεθεί το container. Υπολογίστε την απόσταση για να προσδιορίσετε πότε ένα container πρέπει να μετακινηθεί σε νέο σταθμό βάσης. Σταματήστε το container, αποσυνδέστε το από το τρέχον δίκτυο και επανασυνδέστε το στο δίκτυο του πλησιέστερου σταθμού βάσης.

hint: μπορείτε να αναπαραστήσετε κάθε σταθμό βάσης ως ένα python object που περιέχει πληροφορίες σχετικά με τους πόρους του (π.χ. εύρος ζώνης, αριθμό συνδεδεμένων container κλπ) και τη θέση του (π.χ σε 2D grid). Εάν θέλετε να προσθέσετε πολυπλοκότητα (bonus),

μπορείτε να εισάγετε ένα απλό μοντέλο εξασθένησης της ισχύος του σήματος ή ακόμα να προσομοιώσετε handover failures.

Μέρος 3: Ανάλυση απόδοσης και οπτικοποίηση. Συλλέξτε δεδομένα είτε μέσω του docker, είτε χρησιμοποιώντας άλλα εργαλεία συλλογής και οπτικοποίησης, για την απόδοση του δικτύου, συμπεριλαμβανομένων:

- Χρήση πόρων και κίνηση δικτύου ανά σταθμό βάσης.
- Αριθμός και ποσοστό επιτυχίας παραδόσεων (handover success rate)

Επιβλέπων: Δημήτρης Σελής, email επικοινωνίας: dselis@isi.gr

ΕΡΓΑΣΙΑ 3) Προσομοίωση σε Python ενός χρονοπρογραμματιστή 4G

Ο στόχος αυτής της εργασίας είναι η υλοποίηση μιας προσομοίωσης ενός απλοποιημένου χρονοπρογραμματιστή σε ένα τηλεπικοινωνιακό δίκτυο, όπου ένας σταθμός βάσης κατανέμει πόρους σε διάφορους χρήστες (UEs) χρησιμοποιώντας διαφορετικούς αλγορίθμους χρονοπρογραμματισμού υπό διάφορες συνθήκες traffic και ποιότητας καναλιού.

Βασικά ζητούμενα

1. **Ανάπτυξη του περιβάλλοντος προσομοίωσης:** Δημιουργήστε ένα μοντέλο δικτύου που περιλαμβάνει ένα σταθμό βάσης και πολλαπλά UEs, ορίζοντας βασικά χαρακτηριστικά για κάθε UE – όπως ένα μοναδικό ID, διαφορετικές απαιτήσεις κίνησης (traffic), τυχαίους δείκτες ποιότητας καναλιού και επίπεδα προτεραιότητας
2. **Εφαρμογή αλγορίθμων χρονοπρογραμματισμού:** Ανάπτυξη απλουστευμένων αλγορίθμων χρονοπρογραμματισμού, συμπεριλαμβανομένων των αλγορίθμων Round-Robin και Proportional Fair (που κατανέμει τους πόρους με εξισορρόπηση του στιγμιαίου ρυθμού δεδομένων και της μέσης απόδοσης κάθε UE για τη βελτιστοποίηση της δικαιοσύνης και της αποδοτικότητας), οι οποίοι κατανέμουν δυναμικά τους πόρους στα UE με βάση τα καθορισμένα κριτήρια.
3. **Δημιουργία μοντέλων παραγωγής traffic:** Σχεδιάστε απλά μοντέλα δημιουργίας traffic για UEs προσομοιώνοντας διάφορους τύπους - όπως συνεχής ζήτηση υψηλού εύρους ζώνης για ροή βίντεο, σποραδικά bursts για περιήγηση στον ιστό και σταθερό χαμηλό εύρος ζώνης για φωνητικές κλήσεις - εισάγοντας τυχαιότητα και μεταβλητότητα για να μιμηθείτε τη συμπεριφορά των χρηστών του πραγματικού κόσμου.
4. **Προσομοίωση τυχαίων συνθηκών καναλιού:** Μοντελοποίηση τυχαίων συνθηκών καναλιού επιτρέποντας στους δείκτες ποιότητας καναλιού για κάθε UE να αυξομειώνονται τυχαία με την πάροδο του χρόνου, επηρεάζοντας τις αποφάσεις κατανομής πόρων του χρονοπρογραμματιστή και αντικατοπτρίζοντας την απρόβλεπτη φύση των ασύρματων καναλιών.
5. **Εκτελέστε την προσομοίωση και αναλύστε την απόδοση:** Σε όλη τη διάρκεια της προσομοίωσης, μετρήστε βασικές μετρικές, όπως το συνολικό throughput, τον δείκτη δικαιοσύνης (fairness) και την καθυστέρηση, και αξιολογήστε την απόδοση κάθε αλγορίθμου χρονοπρογραμματισμού σε διαφορετικά σενάρια δικτύου (διαφορετικός αριθμός χρηστών, διαφορετικές ανάγκες traffic κτλ.) αναλύοντας αυτές τις μετρικές που συλλέξατε για να εντοπίσετε τα δυνατά σημεία, τις αδυναμίες και τους συμβιβασμούς στις στρατηγικές κατανομής πόρων.

Παραδοτέα

1. Αρχείο κώδικα σε Python για την υλοποίηση όλων των ερωτημάτων, με τα απαραίτητα σχόλια για κάθε ερώτημα.
2. Αναλυτική αναφορά (μέχρι περίπου 10 σελίδες) που θα περιέχει την μεθοδολογία προσέγγισης κάθε ερωτήματος, τις απαντήσεις στις ερωτήσεις σύγκρισης των αποτελεσμάτων, καθώς και τις γραφικές παραστάσεις των μετρικών για κάθε σενάριο προσομοίωσης.

Επιβλέπων: Βασίλης Αυγερινός, email επικοινωνίας: avgerinos@isi.gr

ΕΡΓΑΣΙΑ 4) Scheduling αλγόριθμοι σε MATLAB, για downlink σε LTE-4G

Γενική περιγραφή. Στα **LTE-4G δίκτυα τηλεπικοινωνιών**¹, ένα κυρίαρχο ζήτημα στο δικτυακό επίπεδο MAC είναι αυτό του scheduling των downlink και uplink requests². Χωριστά στις δύο αυτές περιπτώσεις έχουν αναπτυχθεί αντίστοιχες **scheduling policies**³ (χρονοδρομολογητές).

Στόχοι του project. Ο στόχος του project είναι να αποτιμηθεί η λειτουργία πέντε schedulers που επιλύουν το παραπάνω ζήτημα. Για το **downlink**, προτείνονται οι ακόλουθοι schedulers: round robin, proportional fair, maximum throughput (είναι ήδη υλοποιημένοι), και maximum carrier-to-interference, quality guaranteed, packet loss ratio, earliest deadline first, modified earliest deadline first – proportional fair, modified largest weighted delay first, opportunistic packet loss fair, και exponential proportional fair (δεν είναι ήδη υλοποιημένοι). Η αποτίμηση θα γίνει βασικά με χρήση ως ανεξάρτητης μεταβλητής (input) **το πλήθος των χρηστών**, και με χρήση εξαρτημένων μεταβλητών (outputs) τα **throughput, delay, και fairness** της λειτουργίας του δικτύου. Πρέπει στο τέλος να παραχθούν όμορφα γραφήματα (scatter plots, bar plots), που να αποτυπώνουν το αποτέλεσμα της αποτίμησης. Υπάρχει το ελεύθερο να επιλεγούν και άλλες μεταβλητές input, που να συνδέονται με mobility patterns ή παραμέτρους scheduling. Η επιλογή εκ των schedulers αφήνεται στο φοιτητή ή στη φοιτήτρια, όπως λέγεται παρακάτω.

Αρχές υλοποίησης. Η υλοποίηση οφείλει να γίνει σε **MATLAB**, με χρήση είτε του **toolbox LTE**⁴ (προτείνεται), είτε του **toolbox 5G**⁵ (εναλλακτικά). Θα περιέχει την υλοποίηση της υποδομής του δικτύου, την εφαρμογή **δύο** υλοποιημένων schedulers, και την υλοποίηση ως **τριών** extra. Αν υλοποιηθούν οι τρεις extra, επιτυγχάνεται το άριστα στην εργασία. Ο κάθε extra scheduler ενδέχεται βάσει των δικών του χαρακτηριστικών να περιέχει και περαιτέρω παραμέτρους. Οι παράμετροι αυτοί θα κάνουν την αρχικοποίηση του δικτύου πιο σύνθετη. Έχετε το ελεύθερο να κάνετε παραδοχές που να στέκουν, με τις παραδοχές αυτές να φαίνονται στην υλοποίηση. Έχετε, επίσης, το ελεύθερο να χρησιμοποιήσετε τις παραμέτρους αυτές ως μεταβλητά inputs, και έτσι να παρέχετε γραφήματα των μετρικών του δικτύου βάσει κι αυτών των παραμέτρων. Σε κάθε περίπτωση, η βασική input parameter είναι ο αριθμός των χρηστών, με τις μετρικές της λειτουργίας του δικτύου (συναρτήσεις αυτής) να είναι οι throughput, delay, και fairness.

Τελικά παραδοτέα. Η διαδικασία του μαθήματος περιλαμβάνει μια ενδιάμεση και μια τελική παρουσίαση. Μια καλή πολιτική για το εν λόγω project είναι η **ενδιάμεση παρουσίαση (20%** του βαθμού) να περιέχει μια βασική υλοποίηση της δικτυακής υποδομής, και την εφαρμογή **δύο** schedulers. Η **τελική παρουσίαση (40%** του βαθμού) πρέπει να περιέχει μια πιο σύνθετη υλοποίηση της δικτυακής υποδομής, που να λαμβάνονται υπόψη και οι επιπλέον παραδοχές των extra schedulers, και η υλοποίηση έως και **τριών** extra schedulers. Η τελική παρουσίαση συνοδεύεται από μια **τελική αναφορά (40%** του βαθμού), και το αρχείο του κώδικα MATLAB.

Επιβλέπων: Γιάννης Λακουμέντας, email επικοινωνίας: lakoumen@upatras.gr

¹ <https://www.nxp.com/docs/en/white-paper/LTEPTCLOVWWP.pdf>

² <https://www.3gpp.org/technologies/scheduling>

³ <https://arxiv.org/pdf/2204.11369>

⁴ <https://www.mathworks.com/help/lte/>

⁵ <https://www.mathworks.com/help/5g/>

ΕΡΓΑΣΙΑ 5) Scheduling αλγόριθμοι σε MATLAB, για uplink σε LTE-4G

Γενική περιγραφή. Στα **LTE-4G δίκτυα τηλεπικοινωνιών**⁶, ένα κυρίαρχο ζήτημα στο δικτυακό επίπεδο MAC είναι αυτό του scheduling των downlink και uplink requests⁷. Χωριστά στις δύο αυτές περιπτώσεις έχουν αναπτυχθεί αντίστοιχες **scheduling policies**⁸ (χρονοδρομολογητές).

Στόχοι του project. Ο στόχος του project είναι να αποτιμηθεί η λειτουργία πέντε schedulers που επιλύουν το παραπάνω ζήτημα. Για το uplink, προτείνονται οι ακόλουθοι schedulers: low complexity, low complexity delay, proportional fair delay, first maximum expansion, adaptive resource allocation based packet scheduling (**δεν είναι ήδη υλοποιημένοι**), και οι round robin, proportional fair, και maximum throughput (**είναι ήδη υλοποιημένοι**). Η αποτίμηση θα γίνει βασικά με χρήση ως ανεξάρτητης μεταβλητής (input) **το πλήθος των χρηστών**, και με χρήση ως εξαρτημένων μεταβλητών (ως outputs) τα **throughput, delay, και fairness** της λειτουργίας του δικτύου. Πρέπει στο τέλος να παραχθούν όμορφα γραφήματα (scatter plots, bar plots), που να αποτυπώνουν το αποτέλεσμα της αποτίμησης. Υπάρχει το ελεύθερο να επιλεγούν και άλλες μεταβλητές input, που να συνδέονται με mobility patterns ή παραμέτρους scheduling. Η επιλογή εκ των schedulers αφήνεται στο φοιτητή ή στη φοιτήτρια, όπως λέγεται παρακάτω.

Αρχές υλοποίησης. Η υλοποίηση οφείλει να γίνει σε **MATLAB**, με χρήση είτε του **toolbox LTE**⁹ (προτείνεται), είτε του **toolbox 5G**¹⁰ (εναλλακτικά). Θα περιέχει την υλοποίηση της υποδομής του δικτύου, την εφαρμογή **δύο** υλοποιημένων schedulers, και την υλοποίηση ως **τριών** extra. Αν υλοποιηθούν οι τρεις extra, επιτυγχάνεται το άριστα στην εργασία. Ο κάθε extra scheduler ενδέχεται βάσει των δικών του χαρακτηριστικών να περιέχει και περαιτέρω παραμέτρους. Οι παράμετροι αυτοί θα κάνουν την αρχικοποίηση του δικτύου πιο σύνθετη. Έχετε το ελεύθερο να κάνετε παραδοχές που να στέκουν, με τις παραδοχές αυτές να φαίνονται στην υλοποίηση. Έχετε, επίσης, το ελεύθερο να χρησιμοποιήσετε τις παραμέτρους αυτές ως μεταβλητά inputs, και έτσι να παρέχετε γραφήματα των μετρικών του δικτύου βάσει κι αυτών των παραμέτρων. Σε κάθε περίπτωση, η βασική input parameter είναι ο αριθμός των χρηστών, με τις μετρικές της λειτουργίας του δικτύου (συναρτήσεις αυτής) να είναι οι throughput, delay, και fairness.

Τελικά παραδοτέα. Η διαδικασία του μαθήματος περιλαμβάνει μια ενδιάμεση και μια τελική παρουσίαση. Μια καλή πολιτική για το εν λόγω project είναι η **ενδιάμεση παρουσίαση (20%** του βαθμού) να περιέχει μια βασική υλοποίηση της δικτυακής υποδομής, και την εφαρμογή **δύο** schedulers. Η **τελική παρουσίαση (40%** του βαθμού) πρέπει να περιέχει μια πιο σύνθετη υλοποίηση της δικτυακής υποδομής, που να λαμβάνονται υπόψη και οι επιπλέον παραδοχές των extra schedulers, και η υλοποίηση έως και **τριών** extra schedulers. Η τελική παρουσίαση συνοδεύεται από μια **τελική αναφορά (40%** του βαθμού), και το αρχείο του κώδικα MATLAB.

Επιβλέπων: Γιάννης Λακουμέντας, email επικοινωνίας: lakoumen@upatras.gr

⁶ <https://www.nxp.com/docs/en/white-paper/LTEPTCLOVWVP.pdf>

⁷ <https://www.3gpp.org/technologies/scheduling>

⁸ <https://arxiv.org/pdf/2204.11369>

⁹ <https://www.mathworks.com/help/lte/>

¹⁰ <https://www.mathworks.com/help/5g/>

ΕΡΓΑΣΙΑ 6) Προσομοίωση δικτύου 4G και αλγόριθμοι χρονοπρογραμματισμού (PF, MT)

Η παρούσα εργασία έχει ως στόχο την ανάπτυξη μιας προσομοίωσης ενός δικτύου 4G χρησιμοποιώντας Python. Θα χρειαστεί να προσομοιώσετε την συμπεριφορά του δικτύου υπό διάφορες συνθήκες φορτίου και να αξιολογήσετε την απόδοση του με βάση διάφορες μετρικές απόδοσης.

Απαιτούμενες Υλοποιήσεις:

1. Προσομοίωση Δικτύου 4G:

- a. Αναπτύξτε μια προσομοίωση ενός δικτύου 4G χρησιμοποιώντας Python, το οποίο να περιλαμβάνει έναν σταθμό βάσης και χρήστες(UE). Κάθε UE θα έχει μοναδικά χαρακτηριστικά, όπως αναγνωριστικό(ID), διαφορετικές απαιτήσεις κίνησης, τυχαίους δείκτες ποιότητας καναλιού(CQI), και διαφορετικά επίπεδα προτεραιότητας.
- b. Υλοποίηση μοντέλου κινητικότητας όπου τα UEs μετακινούνται τυχαία εντός της περιοχής κάλυψης του σταθμού βάσης. Οι UEs θα πρέπει να έχουν ποικίλες ταχύτητες και μοτίβα κίνησης (π.χ. πεζοί, οχήματα) για να αντικατοπτρίζουν διαφορετικά σενάρια του πραγματικού κόσμου.

2. Υλοποίηση Αλγορίθμων Χρονοπρογραμματισμού:

- a. **Proportional Fair (PF):** Ισορροπεί την απόδοση με τη δικαιοσύνη, προσπαθώντας να διασφαλίσει ότι όλοι οι χρήστες παίρνουν δίκαιη πρόσβαση στο κανάλι, ενώ επιτρέπει σε χρήστες με καλύτερη ποιότητα καναλιού να παίρνουν περισσότερους πόρους.
- b. **Max Throughput (MT):** Κατανέμει δυναμικά πόρους με βάση τις τρέχουσες συνθήκες του καναλιού. Ο αλγόριθμος δίνει προτεραιότητα στους χρήστες με την καλύτερη ποιότητα καναλιού. Καθώς οι UEs μετακινούνται και η ποιότητα του σήματος τους αλλάζει, ο αλγόριθμος χρονοπρογραμματισμού θα προσαρμόσει την κατανομή πόρων για να μεγιστοποιήσει τη συνολική απόδοση του δικτύου.

3. Δημιουργία Μοντέλων Παραγωγής Κινητικότητας: Προσομοιώστε διάφορα μοντέλα κινητικότητας για τα UEs:

- a. **Constant Bit Rate (CBR):** Σταθερός ρυθμός ροής δεδομένων, όπως voice calls.
- b. **Variable Bit Rate (VBR):** Μεταβλητός ρυθμός ροής δεδομένων, όπως video streaming ή multimedia.
- c. **HTTP-Like traffic:** Προσομοίωση συμπεριφοράς περιήγησης στον ιστό, όπου τα δεδομένα ζητούνται σε ριπές, όπως λήψη ιστοσελίδων.

4. Μοντελοποίηση Συνθηκών Καναλιού: Η ποιότητα καναλιού του κάθε UE θα επηρεάζεται από την απόσταση του από τον σταθμό βάσης. Τα UEs που βρίσκονται κοντύτερα στο σταθμό βάσης σε κάθε χρονική στιγμή θα έχουν και την καλύτερη ποιότητα καναλιού.

5. Μετρικές Αξιολόγησης και Ανάλυση Απόδοσης: Κατά τη διάρκεια εκτέλεσης της προσομοίωσης, μετρήστε βασικές μετρικές αξιολόγησης του συστήματος, όπως το συνολικό throughput του συστήματος (system throughput), τον δείκτη δικαιοσύνης

(user fairness index), την καθυστέρηση πακέτων (packet delay), την απώλεια πακέτων καθώς και όποιες άλλες θεωρείτε εσείς απαραίτητες. Αξιολογήστε την απόδοση των αλγορίθμων χρονοπρογραμματισμού που υλοποιήσατε βάσει των παραπάνω μετρικών αξιολόγησης.

Παραδοτέα:

- Πηγαίος κώδικας της Python υλοποίησης με τα απαραίτητα σχόλια.
- Τεχνική αναφορά που περιλαμβάνει:
 - Σχεδιασμό και υλοποίηση του συστήματος
 - Μεθοδολογία προσομοίωσης
 - Ανάλυση αποτελεσμάτων
- Συγκριτική αναφορά απόδοσης των δύο αλγορίθμων με γραφήματα και μετρήσεις.

Επιβλέπων: Νίκος Μάριος Ράπτης, email επικοινωνίας: nraptis@isi.gr

ΕΡΓΑΣΙΑ 7) Προσομοίωση δικτύου 4G και αλγόριθμοι χρονοπρογραμματισμού (RR, WFQ)

Η παρούσα εργασία έχει ως στόχο την ανάπτυξη μιας προσομοίωσης ενός δικτύου 4G χρησιμοποιώντας Python. Θα χρειαστεί να προσομοιώσετε την συμπεριφορά του δικτύου υπό διάφορες συνθήκες φορτίου και να αξιολογήσετε την απόδοση του με βάση διάφορες μετρικές απόδοσης.

Απαιτούμενες Υλοποιήσεις:

1. Προσομοίωση Δικτύου 4G:

- a. Αναπτύξτε μια προσομοίωση ενός δικτύου 4G χρησιμοποιώντας Python, το οποίο να περιλαμβάνει έναν σταθμό βάσης και χρήστες (UE). Κάθε UE θα έχει μοναδικά χαρακτηριστικά, όπως αναγνωριστικό (ID), διαφορετικές απαιτήσεις κίνησης, τυχαίους δείκτες ποιότητας καναλιού (CQI), και διαφορετικές απαιτήσεις Quality of Service (QoS).
- b. Υλοποίηση μοντέλου κινητικότητας όπου τα UEs μετακινούνται τυχαία εντός της περιοχής κάλυψης του σταθμού βάσης. Οι UEs θα πρέπει να έχουν ποικίλες ταχύτητες και μοτίβα κίνησης (π.χ. πεζοί, οχήματα) για να αντικατοπτρίζουν διαφορετικά σενάρια του πραγματικού κόσμου.

2. Υλοποίηση Αλγορίθμων Χρονοπρογραμματισμού:

- a. **Round Robin (RR):** Υλοποιήστε τον αλγόριθμο Round Robin, ο οποίος κατανέμει τους πόρους του δικτύου εξίσου σε όλους τους χρήστες, ανεξάρτητα από τις συνθήκες του καναλιού ή τις απαιτήσεις κίνησης.
- b. **Weighted Fair Queuing (WFQ):** Υλοποιήστε τον αλγόριθμο Weighted Fair Queuing, ο οποίος κατανέμει τους πόρους του δικτύου αναλογικά με βάση τα βάρη που έχουν ανατεθεί στους χρήστες. Αναθέστε βάρη στους χρήστες με βάση την προτεραιότητα ή τις απαιτήσεις Quality of Service (QoS).

3. **Δημιουργία Μοντέλων Παραγωγής Κινητικότητας:** Προσομοιώστε διάφορα μοντέλα κινητικότητας για τα UEs:
- Constant Bit Rate (CBR):** Σταθερός ρυθμός ροής δεδομένων, όπως voice calls.
 - Variable Bit Rate (VBR):** Μεταβλητός ρυθμός ροής δεδομένων, όπως video streaming ή multimedia.
 - HTTP-Like traffic:** Προσομοίωση συμπεριφοράς περιήγησης στον ιστό, όπου τα δεδομένα ζητούνται σε ριπές, όπως λήψη ιστοσελίδων.
4. **Μοντελοποίηση Συνθηκών Καναλιού:** Η ποιότητα καναλιού του κάθε UE θα επηρεάζεται από την απόσταση του από τον σταθμό βάσης. Τα UEs που βρίσκονται κοντύτερα στο σταθμό βάσης σε κάθε χρονική στιγμή θα έχουν και την καλύτερη ποιότητα καναλιού.
5. **Μετρικές Αξιολόγησης και Ανάλυση Απόδοσης:** Κατά τη διάρκεια εκτέλεσης της προσομοίωσης, μετρήστε βασικές μετρικές αξιολόγησης του συστήματος, όπως το συνολικό throughput του συστήματος (system throughput), τον δείκτη δικαιοσύνης (user fairness index), την καθυστέρηση πακέτων (packet delay), την απώλεια πακέτων καθώς και όποιες άλλες θεωρείτε εσείς απαραίτητες. Αξιολογήστε την απόδοση των αλγορίθμων χρονοπρογραμματισμού που υλοποιήσατε βάσει των παραπάνω μετρικών αξιολόγησης.

Παραδοτέα:

- Πηγαίος κώδικας της Python υλοποίησης με τα απαραίτητα σχόλια.
- Τεχνική αναφορά που περιλαμβάνει:
 - Σχεδιασμό και υλοποίηση του συστήματος
 - Μεθοδολογία προσομοίωσης
 - Ανάλυση αποτελεσμάτων
- Συγκριτική αναφορά απόδοσης των δύο αλγορίθμων με γραφήματα και μετρήσεις.

Επιβλέπων: Νίκος Μάριος Ράπτης, email επικοινωνίας: nraptis@isi.gr

ΕΡΓΑΣΙΑ 8) Event-driven network simulation σε Python

Αυτή η εργασία αφορά τη μοντελοποίηση και προσομοίωση λειτουργίας ενός απλού δικτύου με χρήση της γλώσσας προγραμματισμού Python. Αφού ολοκληρωθεί η μοντελοποίηση, καλείστε να συλλέξετε και να απεικονίσετε διάφορες μετρικές του δικτύου, όπως καθυστέρηση πακέτων, packet drops, throughput κ.α. Για εκτέλεση της εργασίας προτείνεται η βιβλιοθήκη SimPy <https://simpy.readthedocs.io/en/latest/>, ωστόσο μπορείτε να χρησιμοποιήσετε όποια βιβλιοθήκη προσομοίωσης δικτύων της Python επιθυμείτε.

Η απλή δομή τους δικτύου σας μπορεί για παράδειγμα να περιέχει δύο hosts (source host και destination host) και ένα router που θα διασυνδέει τους hosts και θα επιτρέπει την μεταφορά

πακέτων. Οι συνδέσεις μεταξύ των hosts και του router ορίζονται από συγκεκριμένο bandwidth καθώς και καθυστέρηση διάδοσης (propagation delay) για κάθε σύνδεση.

Βασικά ζητούμενα

1. **Βασική μοντελοποίηση και προσομοίωση:** Χρησιμοποιείτε την Python για να ορίσετε το δίκτυο. Συγκεκριμένα θα χρειαστεί να ορίσετε τους κόμβους (nodes) καθώς και τις συνδέσεις (links) μεταξύ τους. Θα ορίσετε επίσης τα bandwidth και καθυστέρηση διάδοσης (propagation delay) για κάθε link.
2. **Μεταφορά πακέτων:** Προσομοιώστε με την Python πακέτα που μεταφέρονται μέσω του δικτύου. Θα πρέπει να ορίσετε το μέγεθος των πακέτων (π.χ. σε bytes) καθώς και την καθυστέρηση μεταφοράς (transmission delay) και να υπολογίσετε τον συνολικό χρόνο που κάθε πακέτο χρειάζεται για να φτάσει στον τελικό στόχο (π.χ. destination host) δηλαδή το συνολικό end-to-end delay. Ισχύει ότι:

$$\text{Transmission Delay} = \frac{\text{Packet Size}}{\text{Bandwidth}}$$

Προσοχή στο να είναι κατάλληλες οι μονάδες των μεγεθών (π.χ. packet size μετρημένο σε bytes και bandwidth μετρημένο σε bytes/sec αντίστοιχα).

3. **Προσομοίωση ουράς (queue) στον router:** Προσθέστε μια ουρά στον router όπου υπάρχει αναμονή των πακέτων. Το μέγεθος της ουράς θα είναι περιορισμένο (π.χ. 10 πακέτα) και αν η ουρά είναι γεμάτη θα έχουμε packet drop.
4. **Συλλογή και απεικόνιση μετρικών:** Προσομοιώστε την μεταφορά πολλαπλών πακέτων (π.χ. 2000 πακέτα) και συλλέξτε **end-to-end delay για κάθε πακέτο**, **throughput** (συνολικό ποσό δεδομένων σε bits / χρόνος που χρειάστηκε για να μεταφερθούν), **packet loss rate** (ποσοστό των πακέτων που έγιναν drop εξαιτίας υπερχειλίσσης της ουράς) και **queue size** (μέγεθος ουράς). Αποτυπώστε γραφικά αυτές τις μετρικές ως προς τον χρόνο.
5. **Συμφόρηση δικτύου (network congestion) – συσχέτιση bandwidth και καθυστέρησης:** Εκτελέστε την προσομοίωση για διαφορετικές τιμές του bandwidth (π.χ. 1 Mbps, 10 Mbps, 100 Mbps). Υπολογίστε και απεικονίστε γραφικά τις παραπάνω μετρικές του ερωτ. 4 για τις διάφορες τιμές του bandwidth. Παρατηρείτε μείωση της απόδοσης του δικτύου και συμφόρηση ; Εξηγήστε.
6. **Μέγεθος πακέτου:** Εκτελέστε την ίδια διαδικασία στέλνοντας τον ίδιο αριθμό πακέτων αλλά με διαφορετικό μέγεθος πακέτων (π.χ. 100 bytes, 1000 bytes, 3000 bytes, 5000 bytes κ.λπ.). Δείξτε και εξηγήστε αντίστοιχα και με γραφικές παραστάσεις πως επηρεάζονται οι μετρικές του δικτύου ανάλογα με το μέγεθος των πακέτων.
7. **Επέκταση του δικτύου:** Επεκτείνετε το παραπάνω δίκτυο με περισσότερους hosts και routers. Εκτελέστε τα παραπάνω ερωτήματα για το νέο δίκτυο και εξηγήστε πως η πολυπλοκότητα του νέου δικτύου επηρεάζει τις επιδόσεις και τις μετρικές σε σχέση με τον χρόνο και σε σύγκριση με το προηγούμενο δίκτυο, όταν θέλουμε να στείλουμε τον ίδιο όγκο πληροφορίας μέσω πακέτων.
8. **Packet Scheduling Algorithms:** Εφαρμόστε δύο διαφορετικούς αλγόριθμους scheduling πακέτων στην προσομοίωση σας: First-Come-First-Serve (FCFS) και Round-Robin (RR). Για την προσομοίωση χρησιμοποιείτε ένα από τα δίκτυα που έχετε ήδη μοντελοποιήσει, με συνδυασμό μικρών και μεγάλων πακέτων που φτάνουν με ποικίλους ρυθμούς. Αναλύστε πώς κάθε αλγόριθμος scheduling επηρεάζει μετρικές όπως η καθυστέρηση πακέτων (delay) και το throughput. Εξηγήστε επίσης ποιος αλγόριθμος αποδίδει καλύτερα κάτω

από διαφορετικά φορτία κυκλοφορίας (π.χ. ελαφρύ-μικρά πακέτα, μέτριο-μεσαία πακέτα, βαρύ-μεγάλα πακέτα).

Παραδοτέα

1. Αρχείο κώδικα σε Python για την υλοποίηση όλων των ερωτημάτων, με τα απαραίτητα σχόλια για κάθε ερώτημα.
2. Αναλυτική αναφορά (μέχρι περίπου 10 σελίδες) που θα περιέχει την μεθοδολογία προσέγγισης κάθε ερωτήματος, τα μεγέθη που έχουν επιλεχθεί για κάθε σενάριο προσομοίωσης του δικτύου (π.χ. για packet size, bandwidth, propagation delay κ.λπ.), τις απαντήσεις στις ερωτήσεις σύγκρισης των αποτελεσμάτων, καθώς και τις γραφικές παραστάσεις των μετρικών για κάθε σενάριο προσομοίωσης.

Επιβλέπων: Παναγιώτης Μαράντης, email επικοινωνίας: pmaradis@isi.gr

ΕΡΓΑΣΙΑ 9) Wireless mobile network simulation with Python

Αυτή η εργασία αφορά τη μοντελοποίηση και προσομοίωση λειτουργίας ενός δικτύου κινητών επικοινωνιών με χρήση της γλώσσας προγραμματισμού Python. Βασικό στοιχείο της εργασίας είναι η αλληλεπίδραση μεταξύ των σταθμών βάσης (Base Stations - BS) και των συσκευών των χρηστών (User Equipment – UE). Μέσω της προσομοίωσης των BS-UEs αλλά και προσομοίωσης της αποστολής πακέτων-πληροφορίας μεταξύ τους, καλείστε να συλλέξετε και να απεικονίσετε διάφορες μετρικές του δικτύου.

Βασικά ζητούμενα

1. **Βασική μοντελοποίηση του χώρου:** Χρησιμοποιείτε την Python για να ορίσετε τους χρήστες και τους σταθμούς βάσης σε έναν ορισμένο χώρο. Οι χρήστες θα κινούνται εντός του ορισμένου χώρου και δεν θα μπορούν να κινηθούν εκτός αυτού. Η κίνηση για κάθε UE θα γίνεται με τυχαίο τρόπο που θα επιλέξετε εσείς (π.χ. 1 βήμα προς τυχαία κατεύθυνση σε κάθε χρονική στιγμή, 1 χρονική στιγμή ακινησία και την επόμενη 1 βήμα κίνηση κ.λπ.). Οι σταθμοί βάσης προφανώς θα είναι ακίνητοι.

Η ισχύς του σήματος από τον σταθμό βάσης προς το UE θα ορίζεται ως:

$$\text{Signal Strength} = \frac{1}{1 + \text{distance}^2}$$

Σε κάθε χρονική στιγμή της προσομοίωσης κάθε UE θα πρέπει να είναι συνδεδεμένο στον BS από τον οποίο λαμβάνει το ισχυρότερο σήμα. Μετά από κάθε χρονική στιγμή/βήμα θα ελέγχετε την ισχύ για κάθε UE και αν δέχεται ισχυρότερο σήμα από κάποιο άλλο BS, θα πρέπει να συνδεθεί με αυτό το BS (διαδικασία handover). Στην μοντελοποίηση και προσομοίωσή σας θα πρέπει να περιέχονται 3 Base Stations και 10 UEs.

2. **Απεικόνιση προσομοίωσης:** Χρησιμοποιείτε μία απλή διαδικασία απεικόνισης (π.χ. με την βιβλιοθήκη Matplotlib) όπου για ένα επιλεγμένο χρόνο προσομοίωσης (π.χ. 100 seconds = 100 timesteps) θα φαίνεται η κίνηση των UEs στον χώρο αλλά και με ποιο BS

είναι συνδεδεμένα (π.χ. BS και UE συνδέονται με μία γραμμή και κάθε UE μπορεί να απεικονίζεται με το ίδιο χρώμα με το BS στο οποίο είναι συνδεδεμένο κάθε timestep).

3. **Κίνηση Δεδομένων – Data Traffic:** Κάθε χρήστης θα πρέπει να δημιουργεί κίνηση δεδομένων σε χρονικά βήματα επιλογής σας. Για προσομοίωση της αποστολής πακέτων μπορείτε να χρησιμοποιήσετε την βιβλιοθήκη SimPy χωρίς αυτό να είναι υποχρεωτικό (<https://simpy.readthedocs.io/en/latest/>). Υποθέστε ότι κάθε UE δημιουργεί κίνηση συγκεκριμένης κατανομής επιλογής σας (π.χ. τυχαία κατανομή, κανονική κατανομή, κατανομή Poisson κ.λπ.), που αντιπροσωπεύει τον αριθμό των πακέτων δεδομένων ανά χρονικό βήμα. Επιλέξτε επίσης ελεύθερα το μέγεθος του κάθε πακέτου (π.χ. σταθερό μέγεθος).

Ορίστε συγκεκριμένο και σταθερό εύρος ζώνης (Bandwidth) για κάθε BS και χωρίστε το με ίσο τρόπο σε όλους τους συνδεδεμένους χρήστες κάθε timestep. Για κάθε χρήστη υπολογίστε τον χρόνο αποστολής (latency) με βάση τα πακέτα που έχει στείλει και το διαθέσιμο bandwidth ως εξής:

$$Latency (s) = \frac{Packet Size (bits)}{Assigned Bandwidth (\frac{bits}{s})}$$

Για κάθε χρήστη υπολογίστε το Throughput ως εξής:

$$Throughput = \frac{Total Data Transmitted (bits)}{Time (s)}$$

Απεικονίστε γραφικά τα latency και Throughput ως προς τον χρόνο.

4. **Περιορισμένο bandwidth και Packet Loss:** Επιλέξτε κατάλληλες τιμές bandwidth και packet size ώστε να υπάρχουν χρονικές στιγμές όπου το bandwidth ενός ή περισσότερων BS δεν επαρκεί για μεταφορά όλων των πακέτων. Θεωρήστε ότι όταν συμβαίνει αυτό τα πακέτα που στέλνονται εκείνη τη στιγμή χάνονται (packet loss).
5. **Γραφική απεικόνιση:** Υπολογίστε και απεικονίστε γραφικά σε σχέση με τον χρόνο το Throughput, Latency και τον αριθμό χαμένων πακέτων για κάθε χρήστη. Επίσης υπολογίστε και απεικονίστε σε σχέση με τον χρόνο, το bandwidth κάθε BS ώστε να φαίνεται πόσο κοντά είναι στο να εξαντληθεί το εύρος ζώνης του.
6. **Δυναμική κατανομή Bandwidth:** Κατανέμετε με μέθοδο της επιλογής σας το Bandwidth κάθε BS στους UEs που εξυπηρετεί ώστε να πετύχετε καλύτερη εξυπηρέτηση χρηστών και μεταφορά πακέτων (π.χ. κατανομή με βάση ποιος χρήστης έχει στείλει τα περισσότερα ή τα μεγαλύτερα πακέτα). Προσομοιώστε το σενάριο του προηγούμενου ερωτήματος χρησιμοποιώντας την νέα μέθοδο κατανομής εύρους ζώνης και απεικονίστε τις ίδιες μετρικές. Βάσει των αποτελεσμάτων εξηγήστε αν και γιατί υπάρχει βελτίωση στην απόδοση του δικτύου.

Παραδοτέα

3. Αρχείο κώδικα σε Python για την υλοποίηση όλων των ερωτημάτων, με τα απαραίτητα σχόλια για κάθε ερώτημα.
4. PowerPoint με τα αποτελέσματα για την τελική παρουσίαση.
5. Αναλυτική αναφορά (μέχρι περίπου 10 σελίδες) που θα περιέχει την μεθοδολογία προσέγγισης κάθε ερωτήματος, τα μεγέθη που έχουν επιλεγεί για κάθε σενάριο

προσομοίωσης του δικτύου (π.χ. για packet size, bandwidth, propagation delay κ.λπ.), τις απαντήσεις στις ερωτήσεις σύγκρισης των αποτελεσμάτων, καθώς και τις γραφικές παραστάσεις των μετρικών για κάθε σενάριο προσομοίωσης.

Επιβλέπων: Παναγιώτης Μαράντης, email επικοινωνίας: pmaradis@isi.gr

ΕΡΓΑΣΙΑ 10) Προσομοίωση εναλλαγής κυττάρων σε δίκτυο 4G

Στόχος: Σκοπός αυτού του έργου είναι η προσομοίωση της διαδικασίας εναλλαγής κυττάρων (handover) σε ένα απλοποιημένο δίκτυο 4G, όπου ο εξοπλισμός χρήστη (UE) μετακινείται μεταξύ των περιοχών κάλυψης διαφορετικών σταθμών βάσης. Η διαδικασία εναλλαγής κυττάρων διασφαλίζει ότι οι τρέχουσες συνεδρίες επικοινωνίας (όπως οι φωνητικές κλήσεις ή η μετάδοση δεδομένων) δεν διακόπτονται καθώς ο χρήστης μετακινείται.

Βασικά Παραδοτέα:

Ανάπτυξη του Περιβάλλοντος Προσομοίωσης:

1. **Μοντέλο Δικτύου:** Σχεδιάστε ένα απλό μοντέλο δικτύου με πολλαπλούς σταθμούς βάσης (π.χ. 4-6) που είναι διατεταγμένοι σε σχηματισμό πλέγματος, αντιπροσωπεύοντας τις περιοχές κυτταρικής κάλυψης. Κάθε σταθμός βάσης θα έχει καθορισμένη ακτίνα κάλυψης.
2. **Εξοπλισμός Χρήστη (UE):** Μοντελοποιήστε πολλαπλούς UE που κινούνται τυχαία ή κατά προκαθορισμένες διαδρομές εντός του δικτύου. Κάθε UE θα έχει μοναδικό αναγνωριστικό, τοποθεσία και ενεργές συνεδρίες επικοινωνίας (κάθε μια μπορεί να έχει μια μεταβλητή που αντιπροσωπεύει το φόρτο στο δίκτυο).
3. **Ισχύς Σήματος (SNR):** Κάθε σταθμός βάσης θα εκπέμπει σήματα προς τα UE εντός της κάλυψής του. Η ισχύς σήματος θα υπολογίζεται για κάθε UE ανάλογα με την απόστασή του από τον σταθμό βάσης, μοντελοποιώντας την αντίστροφη σχέση μεταξύ απόστασης και ισχύος σήματος (SNR).

Υλοποίηση Αλγορίθμων Εναλλαγής Κυττάρων:

1. **Εναλλαγή Κυττάρου με Ισχυρότερο Σήμα (βάσει RSSI):** Τα UE θα παρακολουθούν την ισχύ σήματος των σταθμών βάσης και θα ενεργοποιούν την εναλλαγή όταν ένας γειτονικός σταθμός βάσης έχει ισχυρότερο σήμα από τον τρέχοντα.
2. **Εναλλαγή Κυττάρου με Όριο (Threshold-based):** Ορίστε μια οριακή τιμή για την ισχύ σήματος. Η εναλλαγή ενεργοποιείται όταν η ισχύς σήματος από τον τρέχοντα σταθμό βάσης πέσει κάτω από αυτό το όριο και ένας γειτονικός σταθμός βάσης προσφέρει ισχυρότερο σήμα.
3. **Εναλλαγή Κυττάρου με Κόστος (Προαιρετικό):** Υλοποιήστε έναν πιο σύνθετο αλγόριθμο εναλλαγής κυττάρων που λαμβάνει υπόψη παράγοντες όπως η εξισορρόπηση φόρτου, η ισχύς σήματος και η απόσταση.

Μοντελοποίηση Κινητικότητας UE και Εναλλαγής Κυττάρων:

1. **Μοντέλα Κινητικότητας:** Προσομοιώστε διάφορα πρότυπα κινητικότητας για τα UE:
 - **Τυχαία Κίνηση:** Τα UE κινούνται τυχαία εντός του δικτύου.
 - **Γραμμική Κίνηση:** Τα UE ακολουθούν μια ευθεία διαδρομή (π.χ. προσομοίωση ενός οχήματος που κινείται κατά μήκος ενός δρόμου).
 - **Προσαρμοσμένες Διαδρομές (Προαιρετικό):** Δημιουργήστε δίκες σας πιο πολύπλοκες γεννήτριες διαδρομών (μπορείτε να αξιοποιήσετε δεδομένα από κάποιο Dataset)
2. **Εκτέλεση Εναλλαγής Κυττάρων:** Όταν πληρούνται οι συνθήκες για την εναλλαγή (με βάση τον επιλεγμένο αλγόριθμο), προσομοιώστε τη διαδικασία εναλλαγής επανατοποθετώντας το UE στον νέο σταθμό βάσης. Κάθε νέα σύνδεση πρέπει να συνοδεύτε από ένα delay που προκύπτει από ένα σταθερό delay για την ανταλλαγή σημάτων και ένα μεταβλητό delay που εξαρτάται από το φορτίο του νέου σταθμού βάσης και την απόσταση του UE από τον σταθμό (ισχύς σήματος).

Προσομοίωση Συνθηκών Δικτύου:

1. **Μεταβαλλόμενος Φόρτος Δικτύου:** Εισαγάγετε τυχαίες ή προκαθορισμένες μεταβολές στον φόρτο που διαχειρίζεται κάθε σταθμός βάσης (π.χ. αριθμός συνδεδεμένων UE ή φόρτος κάθε συνεδρίας που εξυπηρετείτε από τον σταθμό βάσης).
2. **Μεταβαλλόμενη Ισχύς Σήματος:** Μεταβάλλετε τυχαία την ισχύ σήματος με την πάροδο του χρόνου για να αντικατοπτρίσετε συνθήκες του πραγματικού κόσμου, όπως εξασθένηση, σκίαση και εμπόδια.

Ανάλυση Απόδοσης:

1. **Κύριες Μετρήσεις:** Κατά τη διάρκεια της προσομοίωσης, καταγράψτε μετρήσεις όπως:
 - **Καθυστέρηση Εναλλαγής:** Ο χρόνος που απαιτείται για την εκτέλεση της εναλλαγής.
 - **Αριθμός Εναλλαγών:** Ο συνολικός αριθμός των εναλλαγών που εκτελούνται από κάθε UE.
 - **Αποσυνδέσεις (Προαιρετικό):** Καταμετρήστε τον αριθμό των αποσυνδεδεμένων κλήσεων ή συνεδριών δεδομένων κατά τη διάρκεια των εναλλαγών λόγω κακής ποιότητας σήματος ή υπερβολικής καθυστέρησης (απαιτεί να ορίσετε αρκετές ακόμα παραμέτρους στην προσομοίωση σας)
2. **Μεταβολή Σεναρίου:** Προσομοιώστε διάφορα σενάρια αλλάζοντας τον αριθμό των UE, την πυκνότητα των σταθμών βάσης και τα πρότυπα κίνησης, και παρατηρήστε πώς μεταβάλλεται η απόδοση κάθε αλγορίθμου εναλλαγής.

Παραδοτέα

1. Αρχείο κώδικα σε Python για την υλοποίηση όλων των ερωτημάτων, με τα απαραίτητα σχόλια για κάθε ερώτημα.
2. Αναλυτική αναφορά (μέχρι περίπου 10 σελίδες) που θα περιέχει την μεθοδολογία προσέγγισης κάθε ερωτήματος, τις απαντήσεις στις ερωτήσεις σύγκρισης των αποτελεσμάτων, καθώς και τις γραφικές παραστάσεις των μετρικών για κάθε σενάριο προσομοίωσης.

Επιβλέπων: Βασίλης Αυγερινός, email επικοινωνίας: avgerinos@isi.gr