

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

Кваліфікаційна робота магістра

**ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНА ТЕХНОЛОГІЯ МОДЕЛЮВАННЯ
СТІЛЬНИКОВИХ МЕРЕЖ 4G**

Здобувач освіти гр. ІН.м-12ан/2у

Марина СУХОСТАВЕЦЬ

Науковий керівник,
к.ф.-м.н., ст.викладач

Дмитро ВЕЛИКОДНИЙ

В.о. завідувача кафедри
к.т.н., доцент

Ігор ШЕЛЕХОВ

Суми 2022

Факультет ЕІТ Кафедра Комп'ютерних наук
Спеціальність «І22 - Комп'ютерні науки»

Затверджую:

в.о. зав.кафедри _____

“ _____ ” _____ 20__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Сухоставець Марині Віталіївній

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Інформаційно-комунікаційна технологія моделювання стільникових мереж 4G _____

затверджую наказом по інституту від “ _____ ” _____ 20__ р. № _____

2. Термін здачі здобувачем закінченого проєкту (роботи) _____

3. Вхідні данні до проєкту (роботи) _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Інформаційно-аналітичний розділ; 2) Вибір методів рішення задач; 3) Програмна реалізація

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6. Консультанти до проєкту (роботи), із значенням розділів проєкту, що стосується їх

Розділ	Консультант	Підпис, дата
--------	-------------	--------------

		Завдання видав	Завдання прийняв

7. Дата видачі завдання _____

Науковий керівник

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проєкту (роботи)	Термін виконання проєкту (роботи)	Примітка
1.	Інформаційно-аналітичний огляд		
2.	Постановка задачі та формування завдань дослідження		
3.	Вибір методів рішення		
4.	Програмна реалізація		
5.	Оформлення пояснювальної записки до кваліфікаційної магістерської роботи		

Здобувач вищої освіти

(підпис)

Науковий керівник

(підпис)

РЕФЕРАТ

Записка: 60 стор., 38 рис., 1 додаток, 11 літературних джерел.

Об'єкт дослідження — стільникові мережі зв'язку 4G (LTE)

Мета роботи — розробка веб-додатку тренажера побудови LTE мереж з метою навчання користувача.

Результати — проведено аналіз предметної області в напрямі розробки інформаційно-комунікаційної технології проектування систем LTE мереж 4G. За результатами дослідження розроблено описову модель предметної області. У середовищі програми для імітації роботи комунікаційних мереж Cisco Packet Tracer створено схему мережі. Розроблений веб додаток-тренажер для побудови LTE мереж із наданих основних компонентів.

4G, LTE, SIP, INTERNET, IP-PACKETS, JAVASCRIPT, CISCO
PACKET TRACER.

Зміст

ВСТУП.....	5
1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ ОГЛЯД	6
1.1 Еволюція мереж від стандарту 1G до 4G	6
1.2 Архітектура мереж 4G	12
1.3 SIP-телефонія	17
1.4 Постановка задачі	21
2 ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ.....	22
2.1 Опис симулятора Cisco Packet Tracer.....	22
2.2 Моделювання мережі стільникового зв'язку у симуляторі Cisco Packet Tracer	26
2.3 Java Script	32
3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	35
3.1 Розробка тренажеру.....	35
3.2 Тестування тренажеру.....	42
ВИСНОВОК	44
СПИСОК ЛІТЕРАТУРИ	45
ДОДАТКИ.....	47

ВСТУП

Стільникові мережі — це високошвидкісні мережі голосового зв'язку та даних із високою пропускнуою здатністю з розширеними можливостями мультимедіа та безперервного роумінгу для підтримки стільникових пристроїв. Із зростанням популярності стільникових пристроїв ці мережі використовуються не лише для розваг і телефонних дзвінків. Свою назву вони отримали, бо мають схожий спосіб розміщення з бджолиними стільниками.

Вишки стільникового зв'язку є важливою частиною надання послуг стільникового зв'язку. Набір веж стільникового зв'язку створює стільникову мережу, яка забезпечує покриття великого регіону. Компанії стільникового зв'язку часто прагнуть побудувати нові вишки стільникового зв'язку, щоб посилити свої послуги в певній місцевості. Але такі проекти вимагають великих коштів, тому гарно продуманий план того де що розміщувати є важливим етапом розширення мереж.

Метою даної роботи є написання програмного веб застосунку-тренажера для побудови 4G стільниковий мереж, для засвоєння знань про компонентну складову LTE. Також за мету ставиться застосувати сучасні технології і тенденції веб-розробки: JavaScript, CSS3, HTML5, для майбутнього розширення додатку, а також для кращої індексації сторінки.

1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ ОГЛЯД

1.1 Еволюція мереж від стандарту 1G до 4G

В цьому підрозділі розглянемо еволюцію технічних засобів комунікації від 1G до сучасного 4G.

Технологія мобільного зв'язку стартує з 1G (the first generation) – першої генерації цього самого зв'язку. Ця технологія включала в собі лише аналогову передачу голосу за допомогою габаритних, незграбних мобільних аналогових телефонів. Існувало багато різних стандартів зв'язку 1G, розроблених і використовуваних у різних країнах, але найбільш широко поширеними в усьому світі були системи Nordic Mobile Telephone (NMT) і Advanced Mobile Phone System (AMPS). Багато мереж 1G були вимкнені в більшості країнах до 2000 року, хоч в деяких проіснували до 2010-х років[1].

Переваги цифрових технологій перед аналоговими були очевидні, тому мережі 2G згодом повністю замінили 1G. Особливістю цифрових технологій було, те що дані передавалися закодованими в одинички і нулики, а не аналоговим способом. Другим поколінням зв'язку було 2G, що додало цифру в процес мобільного спілкування. Також додалась можливість обміну SMS повідомлень - SMS (Short Message Service) або служба коротких повідомлень, яку зазвичай називають «текстовими повідомленнями», — це служба для надсилання коротких повідомлень довжиною до 160 символів.

Наступним поколінням було 3G, та зробив інтернет мобільним, тепер не лише за допомогою настільних комп'ютерів можливо було під'єднатись до глобальної мережі Інтернету, а також використовувати портативні прилади (смартфони, планшети). Тобто стало можливо відправляти електронні листи, читати онлайн газети та журнали, відвідувати соціальні мережі, переглядати невеликі відеоролики та інше.[2]

Наступною ітерацією мобільного зв'язку є стільникові мережі 4G – вважається супершвидкісним зв'язком на даний час. Його найвища швидкість завантаження – отримання даних на девайс користувача 4G мережі –

приблизно 86Мб/с; та швидкістю вивантаження до 28.8Мб/с – відправки користувачем даних у мережу зі свого девайсу [3].

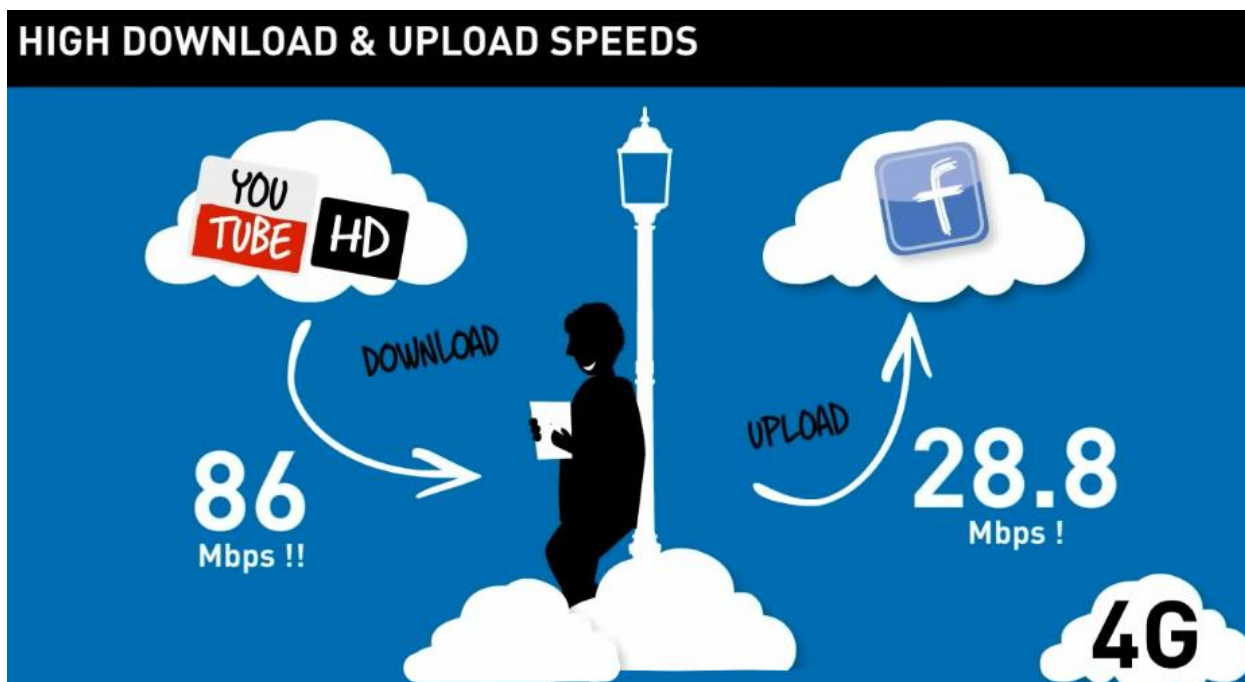


Рисунок 1.1 – Граничні найвищі швидкості 4G мереж

Серед інших переваг 4G мереж є швидша реакція мережі через низький рівень затримки та стабільного з'єднання, навіть в зонах з низьким покриттям.

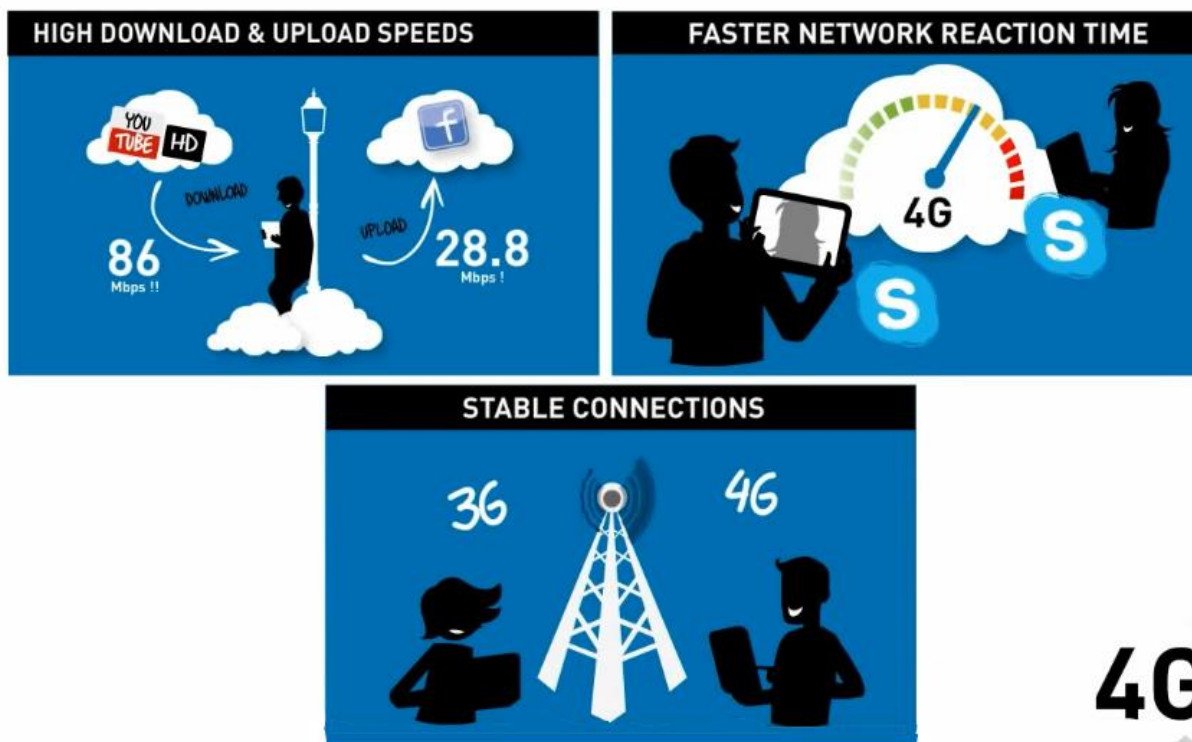


Рисунок 1.2 – Переваги 4G

Але коли, справа доходить до вимірювання швидкості мобільної мережі, існують дві ситуації, які є для нас цікавими, теоретична та реальна ситуація зі швидкістю. Хоч і згадана вище швидкість може бути фізично досягнута, але фактична швидкість обміну даних залежить від великої кількості факторів. Зокрема:

- відстанні до антени чи вишки стільникового зв'язку;
- того чи перебуває користувач в середині приміщення чи зовні на вулиці;
- кількості користувачів в одній зоні;
- ресурсів доступних мережі;
- тип девайсу користувача та застосунку, який використовує послуги мережі.

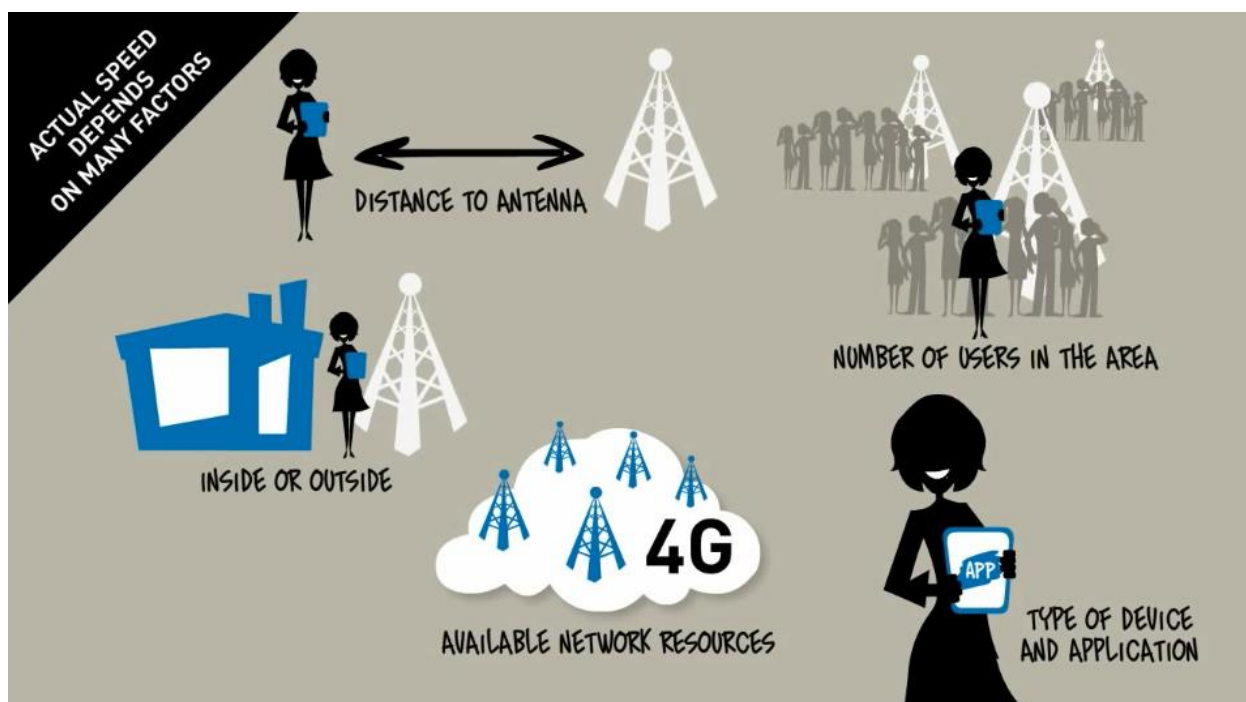


Рисунок 1.2 – Чинники, що впливають на швидкість передачі даних

Теоретичні швидкості – це ті, які ми можемо отримати в лабораторних умовах з ідеальними умовами експерименту, тоді як реальні швидкості – це ті, які ми можемо отримати щодня, використовуючи телефон у реальній мобільній мережі з усіма чинниками, які ми маємо.

Тому зважаючи на вплив всіх аспектів, наша реальна швидкість відрізняється від лабораторної і в нормальних умовах складає за приблизними підрахунками [3]:

- 20 Мб/с на завантаження;
- 12 Мб/с на вивантаження.



Рисунок 1.4 – Швидкість завантаження та вивантаження в нормальних життєвих умовах

Все одно ця швидкість є достатньо високою і достатньою в більшості випадків для користувачів 4G. Розглянемо чинники, які роблять мережі цього покоління такими швидкісними.

- Покращене управління частотами

Користувачі розділяють пропускну спроможність мережі з іншими користувачами, це те що 3G та 4G роблять по різному. В 3G швидкість є фіксована: обмежена та регулюється, наче при їзді на авто з обмежуючими швидкість дорожніми знаками. Напротивагу 4G використовує більш ефективний алгоритм радіо кодування (спосіб кодування радіохвиль), сенс цього алгоритму заключається в тому, що кожен користувач отримує свою пропускну здатність, яка виділена тільки для нього, тут можливо привести

іншу автомобільну аналогію: наче користувачу виділили його власну ділянку дороги на автомобільному треку і він може рухатися настільки швидко настільки він захоче та якщо є інші не зайняті автомобільні треки, користувач може використовувати і їх для збільшення своєї продуктивності [3].

- Використання технології IP

4G мережі використовують IP технології. IP технології є загальною мовою інформаційних систем. Попереднє покоління 3G використовує різні мови взаємодії інформаційних систем: свою власну для передачі пакетів у мережі 3G та IP для передачі пакетів у зовнішніх мережах, тому у мережах витрачається більше часу на упаковку та переупаковку пакетів даних, так щоб вони могли переміщатися мережею з відповідною мовою. На порівняння з 4G мережею таких затрат на пакування немає, бо спосіб (мова) передачі є одна – IP пакети – якими їх пакує девайс користувача, такими вони і подорожують мережею.

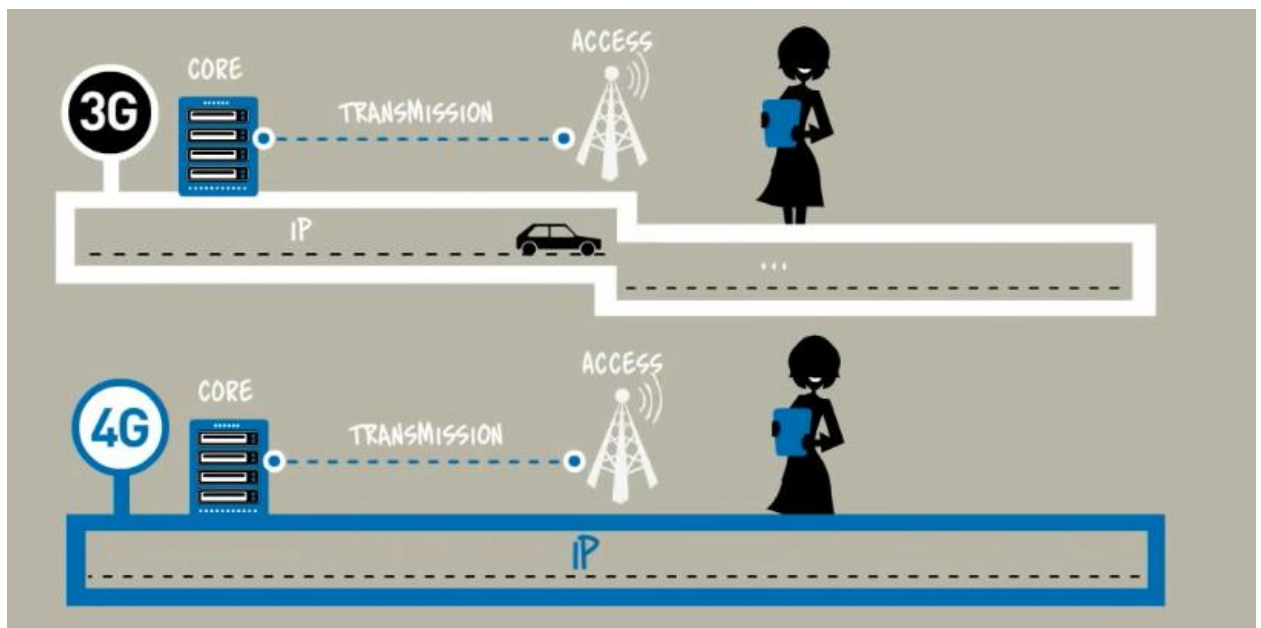


Рисунок 1.5 – Порівняння мережі 4G та 3G

Та останнім переважаючим аспектом є:

- Простіша та розумніша архітектура 4G

Коли користувач під'єднаний до мережі, то йому не потрібно, як в 3G передавати дані через контролера, який регулює трафік. Завдяки розумній

архітектурі 4G вишки моментально знаходять найбільш ефективний і кращий спосіб, щоб під'єднати користувача [4].

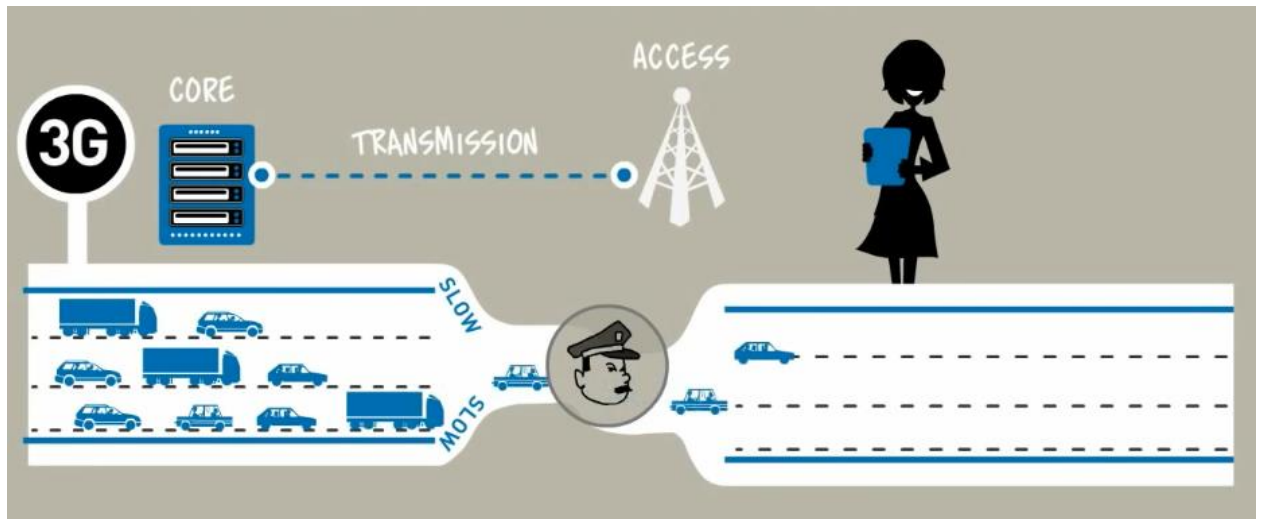


Рисунок 1.1 – Менш ефективний спосіб регулювання пакетів у 3G

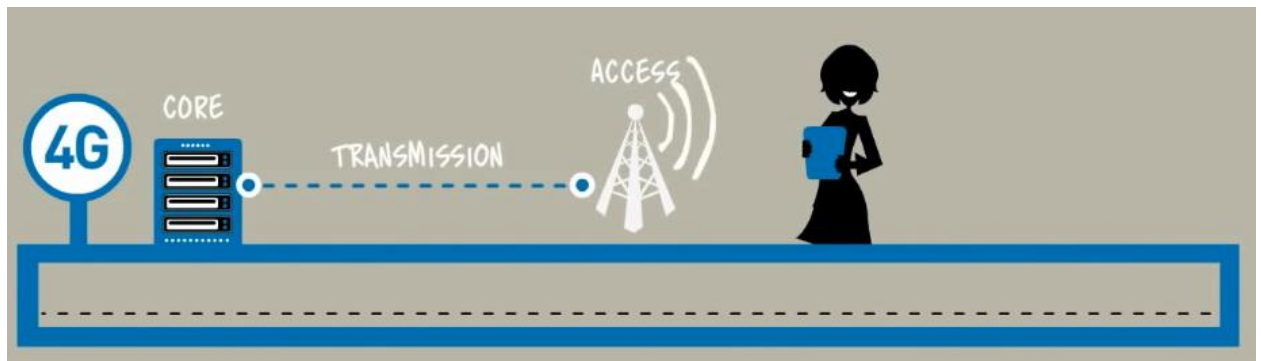


Рисунок 1.6 – Схема ефективнішого способу під'єднання в 4G

Щоб користуватися можливостями сучасних мереж 4G необхідні чотири речі:

- Проста підписка (тариф мобільного оператора), яка включає послуги 4G
- Мобільний пристрій, що підтримує 4G, це може бути планшет, смартфон, переносний 4G роутер чи міні модем.
- Зона покриття 4G в якій користувач має знаходитися.

Тому з 4G користувачі отримують швидкісний інтернет, що може порівнюватися з домашнім інтернетом, навіть коли користувач рухається і переміщується.

Даний підрозділ був невеликим ревію розвитку мобільних комунікаційних технологій, згадавши покоління зв'язку від першого до четвертого, нововведення кожного та переваги кожного над попереднім.

1.2 Архітектура мереж 4G

4G – четверте покоління стільникових мереж. Мережі 4G, вперше представлені в 2009 році, запропонували значно вищу швидкість передачі даних (4G-мережа забезпечує швидкість завантаження до 700 Мбіт/с та передавання — до 90 Мбіт/с.), меншу затримку та більш ефективне використання радіочастотного спектру. Сьогодні велика кількість, а саме: 50 відсотків усіх мобільних з'єднань здійснюється в мережах 4G. 4G рівноцінно називати мережами LTE, бо саме ця технологія і є мережею 4G.

LTE - аббревіатура, розшифровується як Long Term Evolution, що означає “довгостроковий розвиток”, найшвидша на сьогоднішній день технологія, яка забезпечує послуги швидкісного бездротового мобільного зв'язку з високою пропускнуою спроможністю, на даний час це найшвидша технологія, яку пропонує ринок.

Високий показник швидкості став можливим завдяки трьом компонентам:

- більший спектр радіочастот, що використовуються для з'єднання і передачі інформації;
- збільшення кількості відповідних антен – вишок;
- більш ефективне кодування даних, які приходять та відправляються мережею.

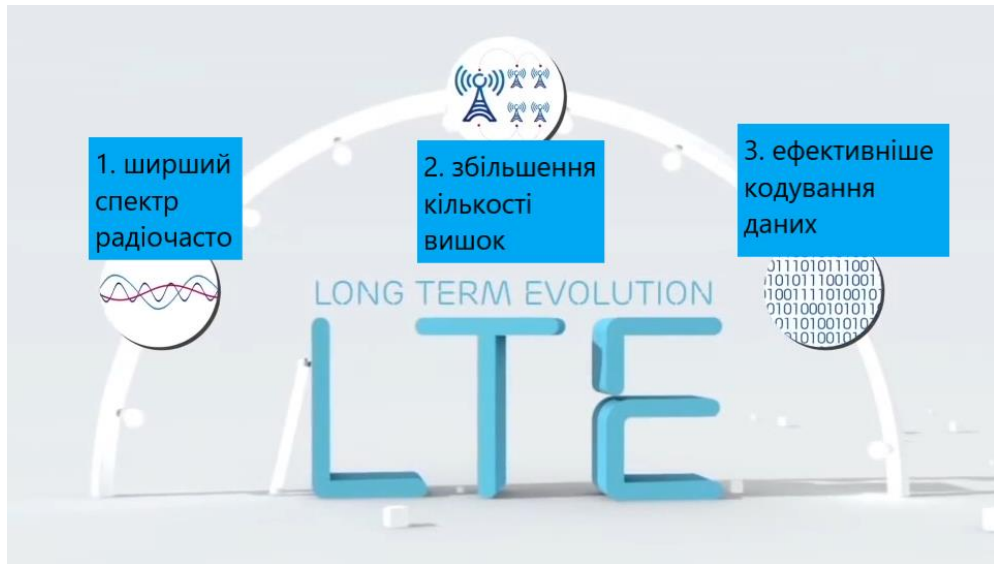


Рисунок 1.7 – Аспекти можливості LTE мереж

Для побудови та функціонування LTE мережа потребує наступних hardware та software елементів:

- антени в базових радіостанціях, так званні ENode B;
- транспортна мережа, включаючи радіорелейні лінії зв'язку та оптичні кабелі, IP маршрутизатори.
- підключення до ір мереж, відоме, як шлюзи чи гетвеї (gateway);
- контролер для керування мобільністю, який називають Mobility Management Entity (MME);
- домашній абонентський сервер бази даних, який зберігає інформацію про всі індивідуальні підписки, англійською home subscriber database (HSS);
- система управління політикою, для того щоб впевнитися у тому, що послуги на які підписаний користувач надаються відповідно;
- мультимедійна підсистема IP для обробки голосових даних над LTE;
- інші мультимедіа системи та пристрої.

LTE System Architecture

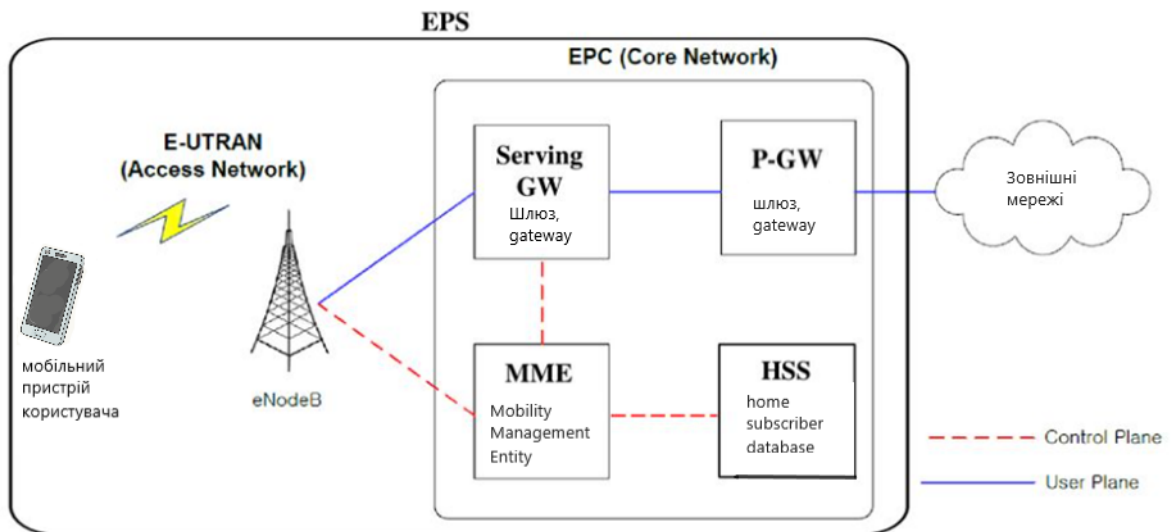


Рисунок 1.8 – Архітектура 4G (LTE) систем

Припустимо користувач, бо все починається з персонального пристрою юзера, хоче виконати передачу голосу через мережу по LTE, простіше кажучи користувач хоче зателефонувати, чи користувач хоче виконати відеозвінок чи відправляти великі за розміром електронні листи чи надіслати листівку тіткі у Вайбері на Великодень разом з іншими користувачами - одночасно. Контролер, він же Mobility Management Entity встановлює з'єднання та контролює і керує з'єднанням з терміналом. В мережах LTE всі дані передаються і отримуються в IP-пакетах. IP-пакети, як коробки на Новій Пошті, у яких запаковані клієнтські музика, фільми, електронні повідомлення та листи, та згажані листівки з Вайберу, також у IP-пакети пакується голос при розмовах і далі ці пакунки даних очікує подорож мережею.

Технологічний шлях ір-пакетів починається з прийняття даних з мобільного пристрою користувача (телефону, планшету тощо) і прискіпливого сортування пакетів, які відправляються на базову радіостанцію ENode B. Enode B закидає ці пакети в величезну транспортну мережу, яка складається з радіорелейних ліній зв'язку, оптичних кабелів і IP-маршрутизаторів. Ця мережа відправляє дані юзера з антен і базової станції ENode B на шлюз. Шлюз

– система, яка складається із певної кількості рівнів. Шлюз також відомий під назвою Gateway. Обслуговуючий шлюз вибирає маршрут і просуває пакети даних далі за їх шляхом. Гейтвей чи шлюз є основним ядром, який зв'язує телефон користувача та мережу IP, в той час як користувач може переміщуватися в просторі разом зі своїм девайсом.

Розглянемо прийом пакетів мережевим шлюзом, він визначає призначення подорожуючих IP-пакетів, сортує їх і відправляє їх правильним шляхом, чимось знову цей процес нагадує відділення Нової Пошти. В шлюзі перетинаються всі види трафіка, не залежно від того хто чи що джерело даних чи технологія використана для їх використання.

В той час коли всі дані обробляються, політика управління підписками підраховує пакети і застосовує правила політики відповідно до персонального плану підписки клієнта. Персональний план підписника-користувача це менш загальновідома назва тарифного плану – домовленості клієнта з провайдером мобільного зв'язку про обсяг і кількість надання комунікаційних послуг.

Також половини шляху, який проходять IP-пакети в мережах LTE, складає мережа Інтернет та інші IP мережі. Весь шлях може подати у зручній таблиці:

Таблиця 1.1 Компоненти мережі LTE

Стадія дії	Учасники	Операція
1.	Мобільний девайс користувача.	Відправка імейлу.
2.	Конектор, він же Mobility Management Entity (MME)	Встановлення з'єднання з терміналом та контроль зв'язку з ним.
3.	Девайс користувача.	Пакування інформацію в IP-пакети.
4.	Базова радіостанція ENode B	Пункт до якого пакети приходять бездротовим

		шляхом і направляються далі за маршрутом.
5.	Шлюз (Gateway)	Сортування пакетів.
6.	Система управління політикою та домашній абонентський сервер.	Перевірка умов тарифу (персональної підписки) користувача.
7.	Шлюз (Gateway)	Відправка до наступного пункту призначення.
8.	Зовнішня мережа	Передача даних далі по маршруту до кінцевого користувача.

LTE (4G) мережі мають ряд переваг в порівнянні з мережами минулих поколінь [5]:

- з LTE серфінг (бездумний або з певною метою перегляд великої кількості веб сторінок) та відвідування веб ресурсів відчуватиметься більш швидко, завдяки зниженому часу очікування та відсутності небажаних затримок;
- краще покриття додатків: тобто більш висока продуктивність у мультимедійних додатках за рахунок більшої швидкості завантаження та вивантаження даних;
- розширення голосового зв'язку з вищою якістю передачі голосу та меншим часом на очікування встановлення з'єднання.
- багато програм Інтернету речей (Internet of Things) покладаються на 4G LTE або його спеціалізовані «категорії»: LTE-M (Long Term Evolution for Machines) і LTE-Cat 1 (категорія 1).

Також перевагою, яка по суті є і важливою умовою існування LTE є сумісність з існуючими телекомунікаційними інфраструктурами та

екосистемами, тобто LTE може взаємодіяти з мережами меншого покоління та з глобальною мережею Інтернет тощо.

Підсумовуючи даний підрозділ ми розглянули необхідні складники мережі LTE (4G), алгоритм за яким передаються IP-пакети, та характеристики мережі, що є переваги перед мережами минулих поколінь зв'язку.

1.3 SIP-телефонія

SIP-Session Initiation Protocol, - Протокол ініціалізації сеансу, що означає протокол початку сесії чи сеансу комунікації. Фактично, IP сеанс використовує, і IP пристрої, і IP мережу як основу. SIP - це сигнальний протокол для встановлення IP сеансу зв'язку. Якщо провести аналогію з реальним живим спілкуванням, то будь яка розмова починається з початкового сигналу[6].



Рисунок 1.9 – SIP - Session Initiation Protocol

Якщо уявити розмову двох людей, нехай це будуть уявні Євгеній і Сергій. Євгеній хоче поговорити з сусідом Сергієм і повідомляє про свої наміри сигналом Сергію – initial signal. Сигнал Євгенія це привітання зі зверненням.



Рисунок 1.10 – Початковий сигнал від співрозмовника

Тобто звук (аналогова хвиля) проходить повітрям до органів чуття Сергія та він побачивши бажання поговорити, передає сигнал згоди назад, та відповідає: «Що таке таке, Євгенію?» - Ok response signal.

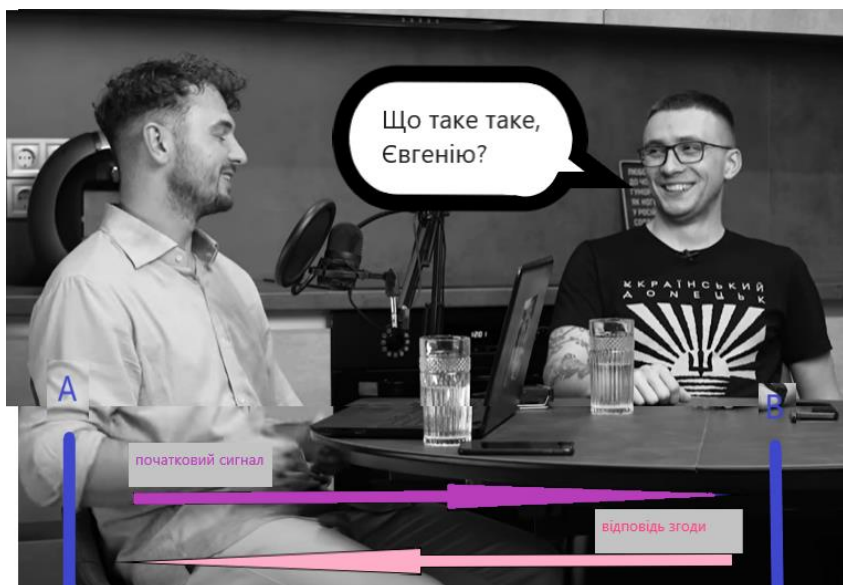


Рисунок 1.1 – Відповідь на ініціацію з'єднання

Тобто співрозмовники почали розмову і далі обмінюються своїми думками. Коли один із співрозмовників вважає, що розмова закінчена, то він відправляє, ще один специфічний сигнал, Євгеній каже: «Бувай, Сергій», а Сергій у відповідь прощається також – сигналізуючи, що сесія завершена.

Це і є принцип роботи сесій у IP світі. Але замість аналогової хвилі – голосу, компоненти мережі використовують IP пакети. Існує безліч видів IP пакетів, головне те що відправник та отримувач повинні обидва розуміти структуру пакета, та знати, що робити з вміщеними даними.

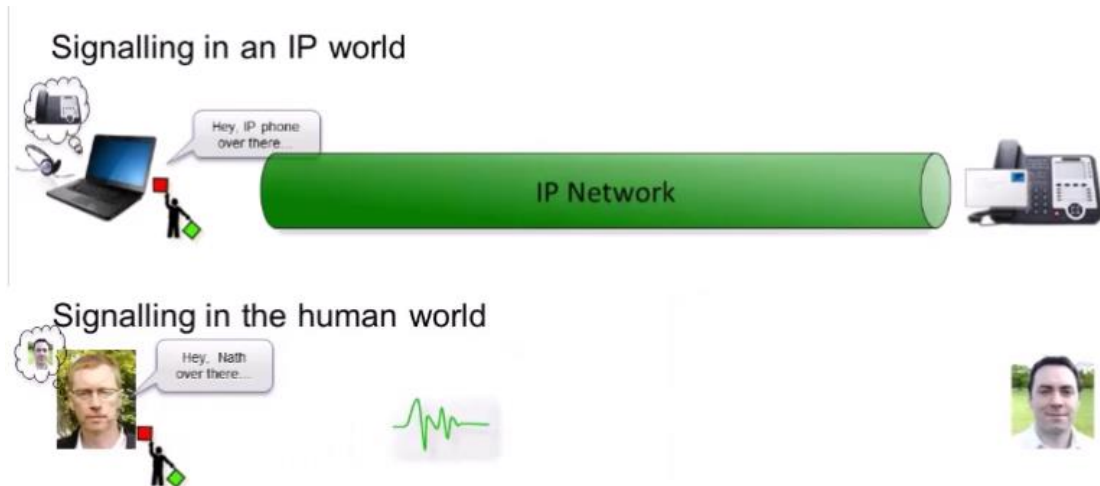


Рисунок 1.12 – Принцип роботи сесії IP

В мережі з IP комунікацією є два види пакетів:

- Сигнальні пакети. Використовуються для створення та встановлення з'єднання.
- Медіа пакети. Використовуються для подальшого обміну аудіо, відео чи інших інтернет даних.

Вхідні аналогові дані кодуються в байти, переносяться мережею і декодуються на стороні отримувача та вже там відтворюються. Та ж сама процедура відбувається при передачі відео, зображення з камери кодується, вкладається в IP пакет, і декодується по прибуттю на девайс іншого учасника сесії.

Two Types of Packet

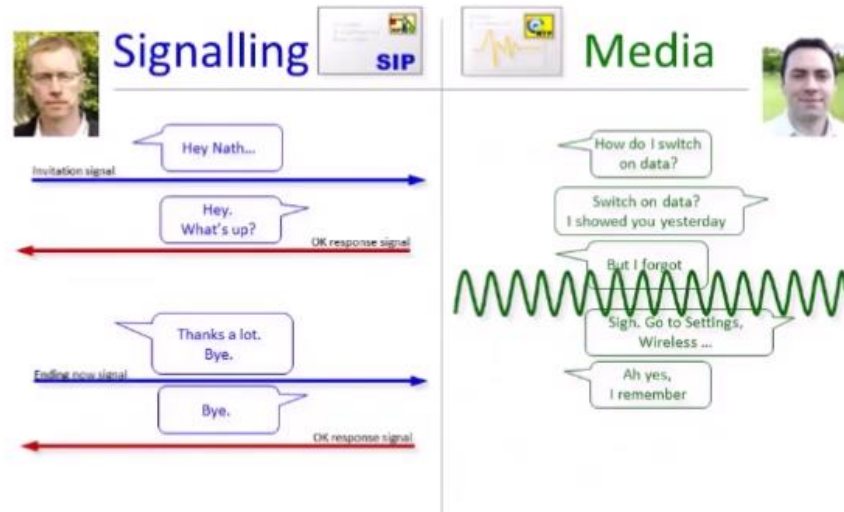


Рисунок 1.13 – Типи IP пакетів

Кожної секунди відбувається обмін великої кількості пакетів, щоб імітувати розмову у реальному часі. Така можливо завдяки тому, що обидві сторони визначили дві важливі речі:

- сторони знають де знаходиться інша сторона і куди відправляти пакети;
- та обидві сторони використовують один ключ для шифрування та дешифрування інформації в пакетах.

Саме для забезпечення цих двох аспектів і використовується SIP протокол. Він визначає де знаходиться інша сторона в мережі, домовляється про ключ шифрування, визначає як конструювати та відправляти пакети, та як керувати обміном інформацією – спілкуванням.

Одна сторона виконує всі умови SIP, щоб сформувати спеціальний IP пакет заповнений необхідними даними для встановлення сигналу та відправки його через IP мережу. З іншої сторони, отримувач знаючи правила SIP розпізнає пакет на спонукання до розмови (девайс видає рінгтон), користувач піднімає слухавку, девайс відправляє декілька пакетів підтвердження установки сесії та інтенсивний обмін медіа пакетами починається. Потім в кінці одна зі сторін сигналізує про завершення сеансу і передача контенту завершується.

Voice over IP



Рисунок 1.14 – Завдання SIP

Після короткого огляду SIP протоколу стає зрозуміло, що це простий звід правил для девайсів, які сигналізують про бажання встановити з'єднання та розірвати з'єднання, для обміну пакетами даних.

1.4 Постановка задачі

На основі зібраних та проаналізованих літературних джерел можна сформулювати ключові етапи кваліфікаційної магістерської роботи.

1. Провести аналіз предметної області за напрямом структура стільникових мереж різних поколінь зв'язку, акцентувати увагу на мережах LTE - четверте покоління.

2. Використовуючи симулятор віртуальної побудови комунікаційних мереж Cisco Packet Tracer відтворити архітектуру LTE, конфігурувати її та продемонструвати приклад її роботи.

3. Підібрати технології веб програмування, визначити ключовий функціонал веб системи, та сформулювати план подальшої розробки тренажера для вивчення архітектури мереж стільникового зв'язку.

4. Розробити веб систему: тренажер для вивчення архітектури мереж стільникового зв'язку, фокусуючись на темі мереж LTE.

5. Перевірити розроблений застосунок на відповідність вимогам, які були висунуті щодо його функціонування.

2 ВИБІР МЕТОДІВ РІШЕННЯ ЗАДАЧІ

2.1 Опис симулятора Cisco Packet Tracer

Cisco Systems, американська технологічна компанія, що працює в усьому світі, найбільш відома своїми продуктами для комп'ютерних мереж, вона виробляє роутери, модеми, свічери та іншу техніку, яка є складовою комп'ютерних мереж Ethernet, Internet, тощо. Будучи компанією, яка продавала свою продукцію переважно іншим компаніям, Cisco не стала загальновідомою, але у другому десятилітті 21 століття вона була однією з найбільших корпорацій у Сполучених Штатах. Cisco була заснована подружньою парою Леонард Босак (Leonard Bosack) і Сандра Лернер (Sandra Lerner) в 1984 році і має штаб-квартиру в Сан-Хосе, Каліфорнія[7].



Рисунок 2.1 – Штаб квартира Cisco, Каліфорнія

Однією із популярних продуктів, який розробила Cisco в сфері моделювання комп'ютерних мереж є Packet Tracer. Packet Tracer — це безкоштовний інструмент, застосунок для симулювання мережі для підготовки до сертифікаційних іспитів, особливо для студентів CCNA (Cisco Certified Network Associate). Він доступний безпосередньо через мережеву академію Cisco. Тобто ви завантажуєте та інстальюєте програмне забезпечення Packet

Tracer, зареєструвавшись на курсі «База в Packet Tracer», який навчить вас основам використання інструменту на освітній платформі від компанії Cisco.

Взагалі Cisco, як будь-яка капіталістична компанія зацікавлена в збільшенні свого прибутку, тому проводить багато навчальних курсів та інших навчальних заходів, які допомагають людям здобувати навички роботи з їх продукцією. Зокрема CCNA — що означає Cisco Certified Network Associate — це сертифікація початкового рівня інформаційних технологій (IT), видана компанією з виробництва мережевого обладнання Cisco. CCNA призначений для перевірки ваших знань про фундаментальні мережеві концепції, які часто вимагають на посадах у сфері мережевих технологій. Даний сертифікат може отримати майже кожен бажаючий, хто проходить курси від компанії Cisco.

Здебільшого за цим сертифікатом звертаються багато людей зацікавлені в накопиченні своїх навичок. Основним інструментом освоєння знань про роботу мережі на курсах Cisco є Packet Tracer, ним користуються зазвичай мережеві допитливі та шанувальниками, CCNA, CCNA Security та студенти CCNP, а також інженери, педагоги чи тренери. Перш ніж запроваджувати будь-який протокол, інженери полюбляють тестувати його на Cisco Packet Tracer[7].

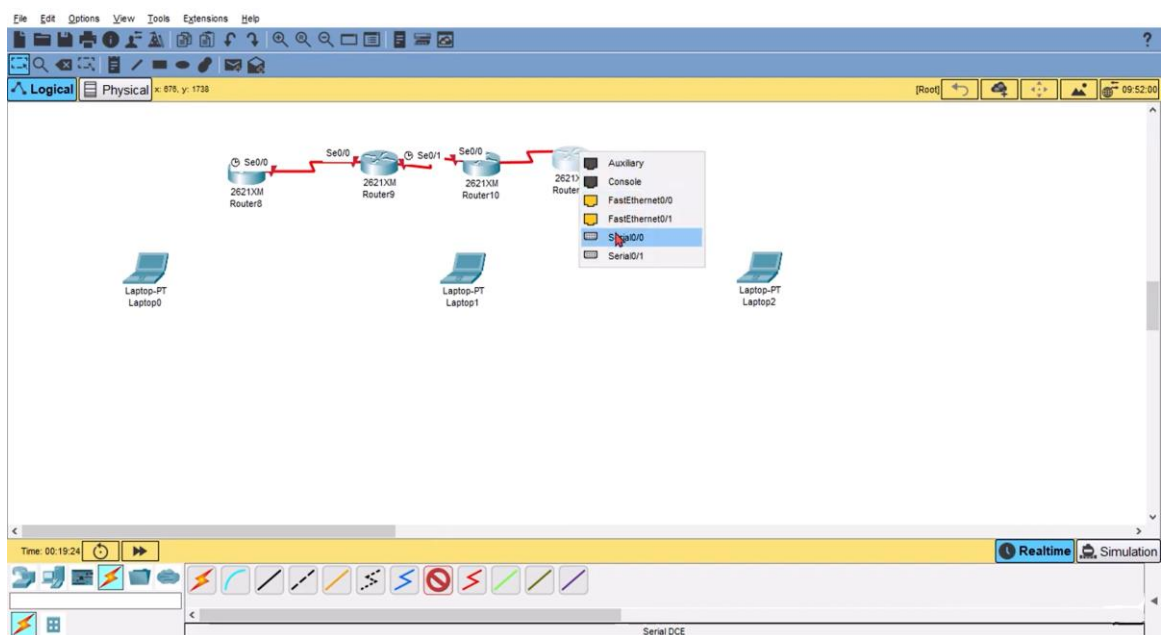


Рисунок 2.2 – Юзер-інтерфейс Packet Tracer

Адже Cisco Packet Tracer надає можливість з'мітувати майже будь-яку мережу: Ethernet, VPN, 3G, 4G. Адже має великий функціонал і включає в себе моделі багатьох реально існуючих продуктів компанії Cisco.

Мережеві тренажери є одним із ключових компонентів навчання для CCNA. Існує кілька мережевих симуляторів, які так широко використовуються, як Cisco Packet Tracer.. За допомогою Packet Tracer ви можете імітувати живе мережеве середовище.

Застосунок доступний декількома мовами:, англійська, німецька, іспанська та французька, на жаль не доступний українською. Packet Tracer дозволяє користувачам створювати складні та величезні мережі, що часто неможливо з фізичним обладнанням через міркування щодо вартості та банальної складності в маніпулюванні великої кількості малих і великогабаритних приладів. Packet Tracer доступний для Linux, Windows, MacOS, Android та iOS.

Інтерфейс застосунку достатньо інтуїтивний більш підготовленій людині. Packet Tracer дозволяє користувачам перетягувати маршрутизатори, комутатори та інші мережеві пристрої для створення моделі-топології мережі. Далі налаштовувати конфіг файли приладів, писати для них команди, проводити імітацію процесу вставлення та витягання кабеля, адже часто в моделях пристрою є інтерактивні форми з їх фізичними інтерфейсами [8].

Цікаво, що цей інструмент містить не лише апаратне забезпечення Cisco, а й широкий спектр інших мережевих пристроїв. Тому користувачі не будуть обмежені лише знаннями продуктів Cisco, а і мати досвід роботи з іншими мережевими приладами, інших компаній.

Ключовими фичами Cisco Packet Tracer є:

- Cisco Packet Tracer підтримує багатокористувацьку систему, що дозволяє багатьом користувачам об'єднувати різні топології в комп'ютерній мережі.

- Підтримує розширення функціоналу за допомогою додаткових програм, які використовують API для покращення можливостей Cisco Packet Tracer. Таку програму можна написати і власноруч, або є варіанти інших користувачів на GitHub.
- Розширений фізичний режим наче транспортує користувача у віртуальну лабораторію, де він можете імітувати роботу з кабельними та гніздами для їх розміщення.

Оновіть ключові навички, такі як розміщення пристроїв (розетки та штекери), перемикання живлення на пристрої, підключення кабелів між портами пристрою – ви маєте, як у реальному житті вмикати кнопку живлення приладу, для імітації повного занування в процес роботи в серверній.

- Packet Tracer можна отримати безкоштовно через обліковий запис Netacad, зареєструватися і завантажити.
- Packet Tracer дозволяє користувачам імітувати конфігурацію, пов'язану з маршрутизаторами Cisco, але процес настільки наближений до реальності, що можна наприклад формувати деякі config файли в Packet Tracer, а потім використовувати їх на реальному обладнанні.
- Мережевий контролер дає вам централізовану інформаційну панель для перегляду стану мережі, миттєвого виявлення та діагностики проблем, а також надсилання змін конфігурації на всі керовані пристрої одночасно, незалежно від того, використовуєте ви графічний веб-інтерфейс чи API.

Ви також можете використовувати реальні програми на своєму комп'ютері для доступу до мережевого контролера та запускати власні сценарії автоматизації інфраструктури. Тобто фактично використовувати побудовану мережу за призначенням і транспортувати нею дані.

Тож даний перелік згаданих переваг та особливостей Cisco Packet Tracer робить його дуже потужним інструментом для демонстрацій роботи телеком мереж.

2.2 Моделювання мережі стільникового зв'язку у симуляторі Cisco Packet Tracer

Однією із цілей даної роботи є побудова мережі LTE - стандарт бездротової високошвидкісної передачі даних для мобільних телефонів та інших терміналів, що працюють з даними, користуючись можливостями застосунку Cisco Packet Tracer.

Дана мережа має мати наступні компоненти:

№	Назва	Основні елементи мережі
1	БС (ENodeB)	-базова станція мережі стандарту LTE.
2	MME (Узел Управління Мобільністю — Mobility Management Entity)	Це основний керуючий елемент у мережі LTE.
3	S-GW (Serving Gateway — обслуговуючий шлюз)	призначений для обробки та маршрутизації пакетних даних, що надходять із/в підсистему базових станцій.
4	PGW (Пакетний шлюз — мережевий шлюз пакетної передачі даних)	Пакетний шлюз забезпечує з'єднання від UE із зовнішніми пакетними мережами даних, які є точкою входу та вихідного трафіку для UE.
5	PCRF (Узел виставлення рахунків абонентам — функція політики та правил тарифікації)	функція політики (управління політикою) також може бути розділена на 2 функції: контроль шлюза (контроль шлюзу) і контроль якості. Charging Function
6	HSS (Home Subscriber Server — сервер абонентських даних мережі)	HSS є більшою базою даних і призначений для зберігання даних в абонентів. HSS фактично змінює набір реєстрів (VLR, HLR,

	AUC, EIR), які використовувалися в мережах 2G і 3G.
--	---

Схематична структура LTE має вигляд представлений на малюнку і зараз відтворимо цю структуру у програмному додатку Cisco Packet Tracer.

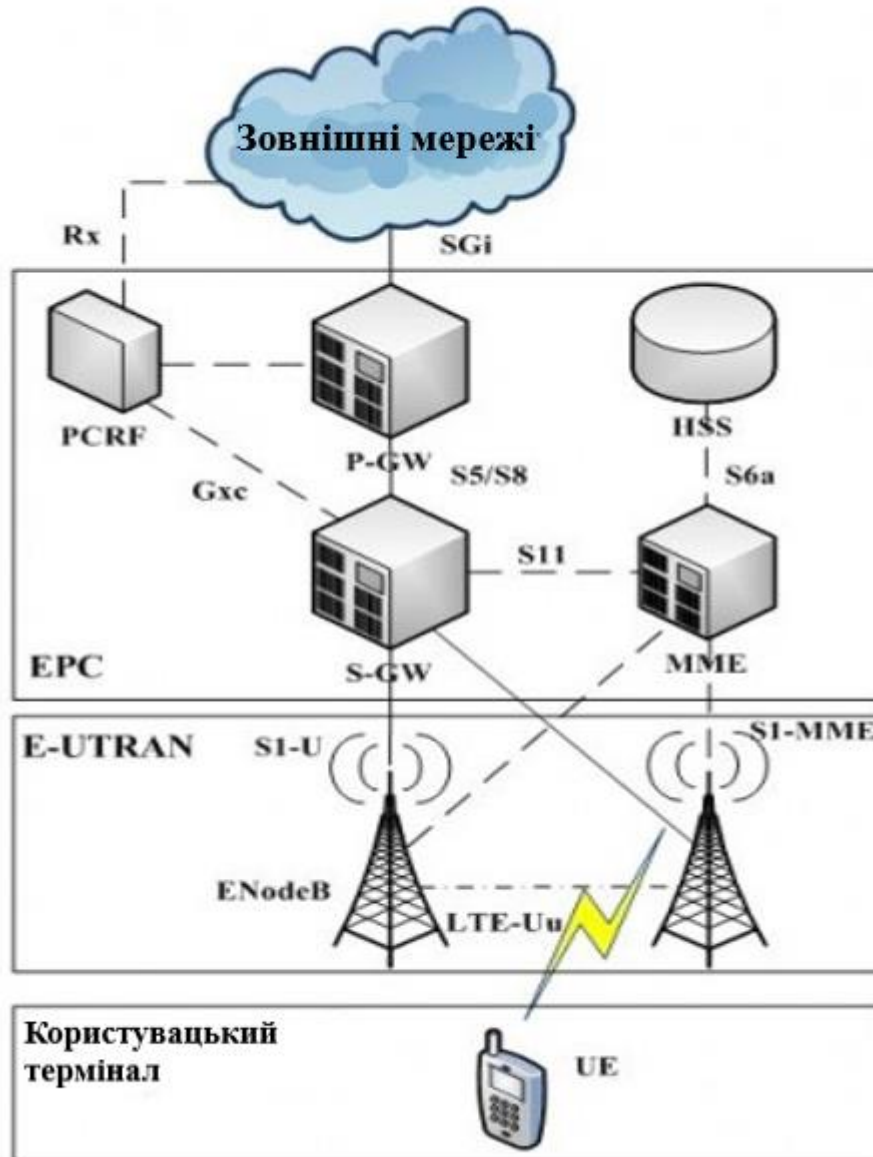


Рисунок 2.3 – Архітектура LTE

Однією із цілей даної роботи є побудова мережі LTE - стандарт бездротової високошвидкісної передачі даних для мобільних телефонів та інших терміналів, що працюють з даними, користуючись можливостями застосунку Cisco Packet Tracer.

Завантажуємо і запускаємо Packet Tracer.

Виставляємо на фото вишки мобільного зв'язку, центральний офіс-сервер до якого вони будуть під'єднуватись та точку доступу.

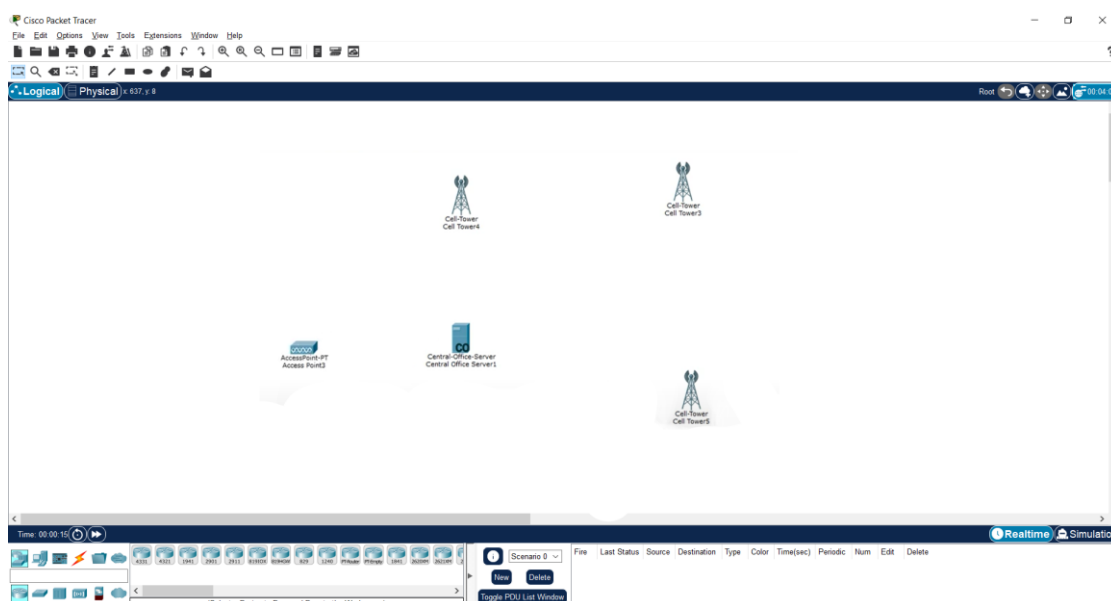


Рисунок 2.4 – Початок схеми

Виставляємо користувачі-девайси майбутньої мережі

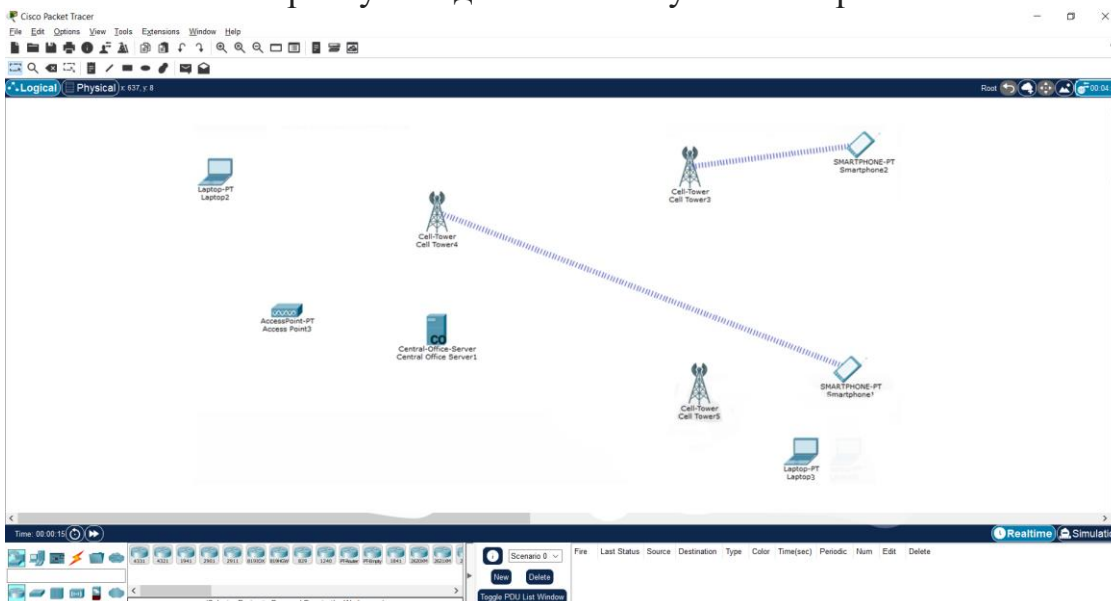


Рисунок 2.5 – Початок схеми

Вносимо конфігурацію на центральному офісі, клацаємо по Central-Office-Server та заходимо у вкладку Config -> Backbone. Даємо IP адресу, адресу гетвею (шлюзу), маска мережі рахується автоматично, також вносимо адресу DNS серверу, на скріншоті бачимо результат [8].

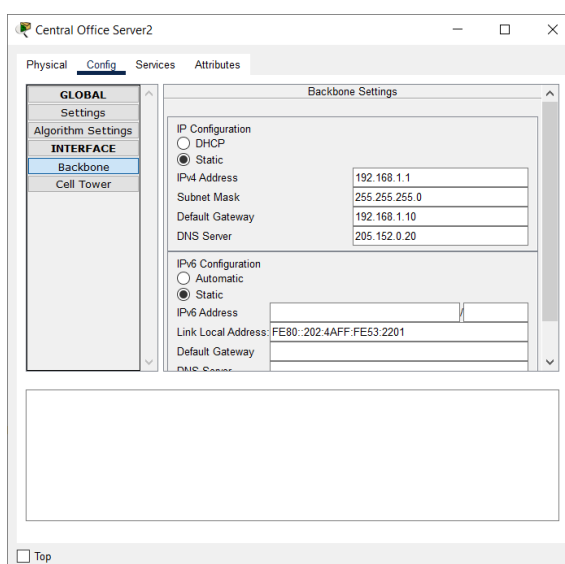


Рисунок 2.6 – Центральний офіс і його налаштування

З'єднуємо чотири наші вежі Cell Tower з центральним офісом Central-Office-Server за допомогою коаксіального (coaxial) кабелю у відповідні порти.

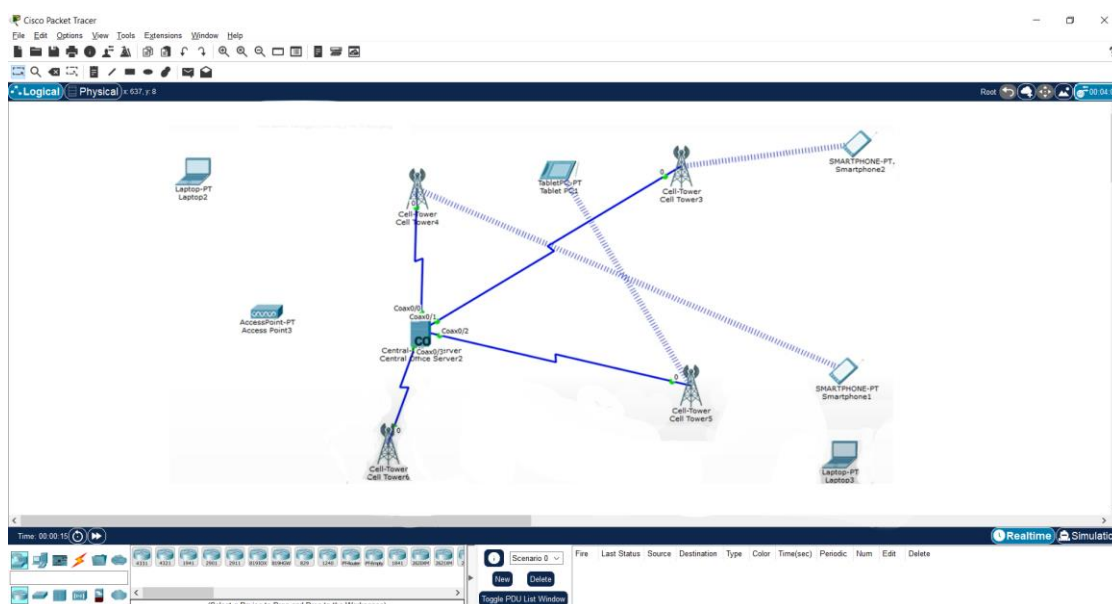


Рисунок 2.7 – Під'єднання кабелю до вузлів

Налаштовуємо смартфони, ноутбуки, планшет та інші девайси, які будуть комунікувати в мережі. Клацаєм на смартфон відкриваємо Desktop вкладку налаштувань, далі кнопкою refresh DHCP підключаємось до найближчої вежі Cell Tower. Dynamic Host Configuration Protocol (DHCP) або Протокол динамічної конфігурації хоста (DHCP) — це клієнт-серверний протокол, який автоматично надає хосту Інтернет-протоколу (IP) його IP-

адресу та іншу пов'язану конфігураційну інформацію, таку як маска підмережі та шлюз (gateway) за замовчуванням, тобто вводити ці дані статично нам не потрібно, даний протокол робить це за нас.

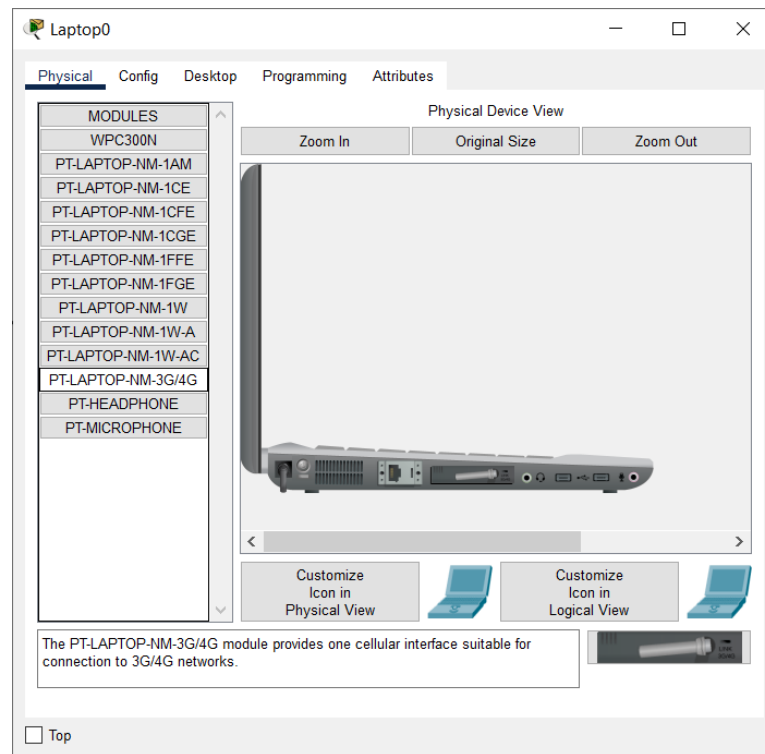


Рисунок 2.8 – Налаштування ноутбуку

Далі налаштовуємо ноутбуки, планшет та ще один смартфон. Планшет налаштовується майже аналогічно, що і смартфон, а от ноутбук має відмінності, адже для того, щоб підключатися до мережі йому потрібно мати необхідний модуль. Тому вимикаємо ноутбук, вставляємо модуль PT-LAPTOP-NM-3G/4G, далі вмикаємо і ноутбук може підключатися до вежі так само як і телефон.

Додатково модуль 3G/4G можна встановити на настільний ПК, хоча таке виконання можливе, але не дуже практичне в реальному житті, хіба, якщо бути впевненому в тому, що місце встановлення ПК ніяк не буде обмежувати бездротовий сигнал. Після проведених маніпуляцій схема має вигляд приведений на наступному скріншоті.

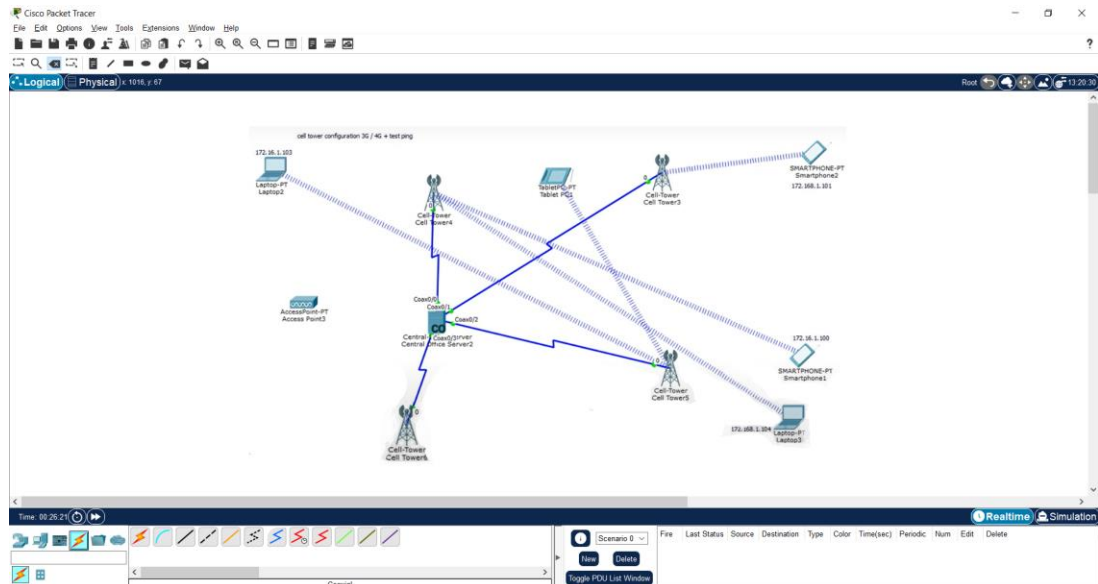


Рисунок 2.9 – Схема після налаштування девайсів

Для перевірки працездатності мережі використаємо вбудовану функцію Packet Tracer, зверху робочої області натискаємо на схематичне зображення листа та відправляємо повідомлення від одного кінцевого вузла до іншого, наприклад від Smartphone1 до Laptop3 і бачимо, що статус доставки пакету (Latest Status) є успішним.

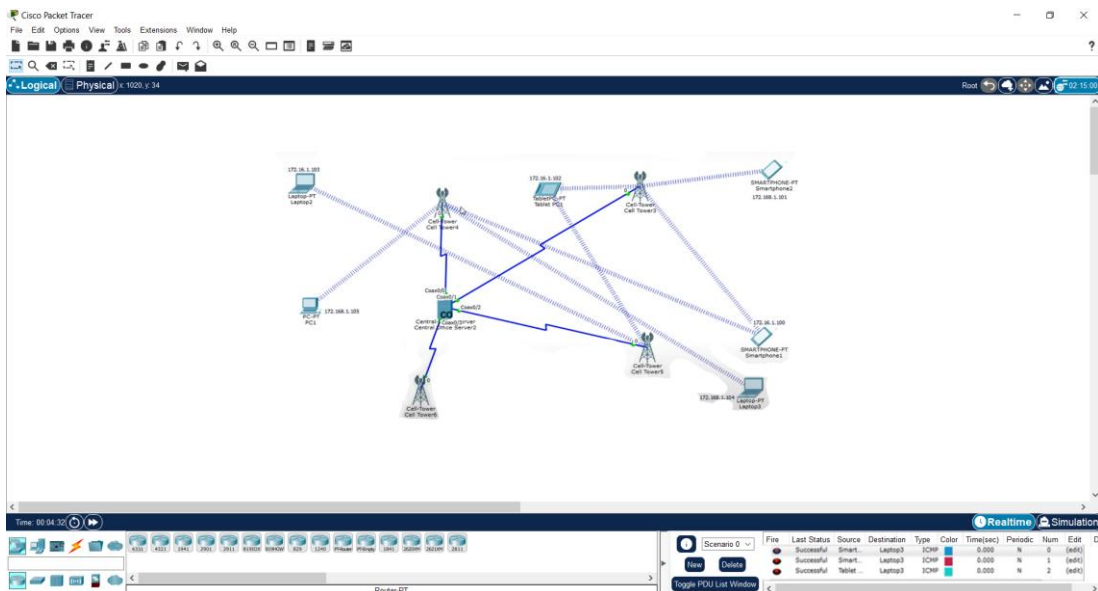


Рисунок 2.10 – Перевірка работоспроможності мережі

Тож можемо зробити висновок, що ми маємо робочу схему, яка імітує живу мережу стільникового зв'язку 4G.

2.3 Java Script

Так як інформаційна система буде виконана у вигляді веб сторінки, то вибір технології програмування був очевидним. JavaScript є мовою монополістом в розділі веб розробки. Інтерактивність та живість сайту по більшій мірі залежить саме від JavaScript.

JavaScript — це динамічна комп'ютерного мова програмування. Вона легка і найчастіше використовується як частина веб-сторінок, рішення, які реалізуються з її використанням дозволяють сценарію на стороні клієнта взаємодіяти з користувачем, тим самим створювати динамічні сторінки. Це інтерпретована мова програмування з об'єктно-орієнтованими можливостями, також JS вважається подіє орієнтованим в тому числі. У світі існує понад 1,8 мільярда веб-сайтів, і на 95% із них використовується JavaScript. Її популярність можна порівнювати до дівчаток блондинок з рожевими гардеробом у фільмах для підлітків у 2000-х, тобто популярність – висока [9].

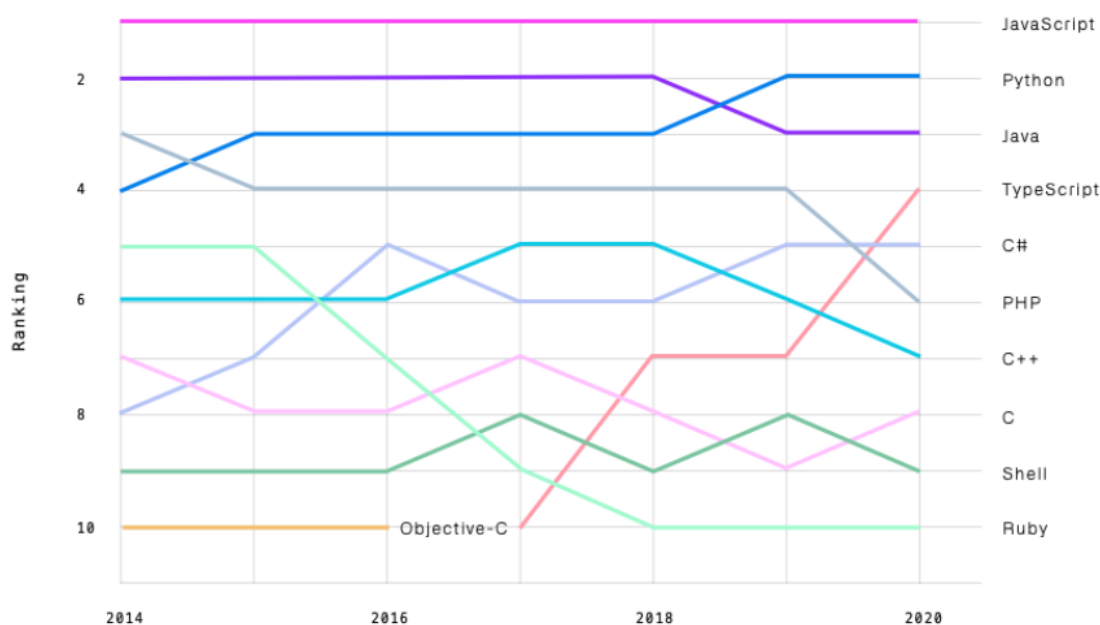


Рисунок 2.11 – Популярність JavaScript протягом багатьох років.

Спочатку JavaScript був відомий як LiveScript, але Netscape (Netscape Communications розробники браузеру Netscape) змінив його назву на

JavaScript, можливо, через тодішню популярність іншої мови програмування, а саме Java. JavaScript вперше з'явився в Netscape 2.0 у 1995 році під назвою LiveScript. Ядро мови загального призначення було вбудовано в Netscape, Internet Explorer та інші веб-браузери її розробника.

Затверджують стандарти для JavaScript Ecma International. Ecma International (офіційно Європейська асоціація виробників комп'ютерів) є некомерційною організацією, яка розробляє стандарти комп'ютерного обладнання, засобів зв'язку та мов програмування [10].



Рисунок 2.12 – Організація розробки стандартів

Розроблена ними специфікація ECMA-262 визначила стандартну версію основної мови JavaScript.

- JavaScript — це легка інтерпретована мова програмування.
- Призначений для розширення можливостей мережевих додатків.
- Доповнює та інтегрується з Java – одною із серверних мов програмування
- Доповнює та інтегрується з HTML – мова розмітки сторінок та документів у браузерах.
- Відкрита та кросплатформна

Так як основна ціль це розробка веб додатків, то JavaScript також являється подієвою мовою програмування. Однією з найбільш вражаючих особливостей JavaScript є те, що він включає програмування на основі подій. Він має вбудовані події, такі як «onClick» і «onHover», які очікують взаємодії користувача на веб-сайті перед виконанням певного сегмента коду. Наприклад, коли ви натискаєте перемикач нічного режиму, це подія, яка

запускає сегмент коду JavaScript, який змінює CSS на всьому веб-сайті зі світлих кольорів на темні.

Ця можливість дуже знадобиться при розробці веб додатку, адже сторінка має реагувати на кліки, перетягування блоків користувачем. Також JavaScript може допомогти з анімаціями та переходами. Хоча багато анімацій стали доступніші за допомогою CSS3.

Ми будемо використовувати мікс всіх вище згаданих технологій для розробки веб застосунку: HTML, JS, CSS3.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Розробка тренажеру

В цьому підрозділі опишемо покроково реалізацію веб застосунку. Для початку розробимо діаграму UML для відображення взаємодії користувача і системи. Для цього застосуємо візуальний редактор для малювання діаграм draw.io. Use case діаграм - діаграма варіантів використання UML є основним графічно виконаним форматом вимог до системи/програмного забезпечення чи іншого проекту, адже UML хоч і сфокусовано на роботі в сфері комп'ютерних наук, все ж може адаптуватися під інші бізнес потреби

Часто use-case застосовується для нової чи недостатньо розробленої і продуманої програми. Варіанти використання вказують на очікувану поведінку - на те що має, повинна робити система, але не на точний спосіб її здійснення- не описує функції, не те що вони мають повертати, як реалізуватися, лише що робити.

Ключовий аспект моделювання діаграм варіантів використання полягає в тому, що вони допомагають розробникам проектувати програмні продукти з точки зору кінцевого користувача. Це ефективна і популярна техніка для узагальнення інформації про поведінку системи в термінах користувача користуючись шляхом визначення всієї видимої ззовні поведінки системи – тобто кружки діаграми є поведінкою системи доступної користувачу.

Use case діаграма нам потрібна для деяких цілей, а саме указати контекст системи, врахувати вимоги до системи, розробити аналітику спільно з експертами в області (керівником роботи) про подальші розширення системи.

Опишемо кожен use-case і розкриємо його суть.

1. Choose type of test. PopUp форма, яка буде пропонувати юзеру обрати те що він хоче зробити, чи виконати тест чи вивчати мережі
2. Test your knowledge with test. Варіант використання, коли користувач у попередньому варіанті (1. Choose type of test) обирає пройти тест, даний

use—case включає include use-case, адже в будь-якому разі користувач отримає результат тестування.

3. Learn LTE structure by test with hints. Варіант використання, коли користувач у попередньому варіанті (1. Choose type of test) обирає навчатися, цей use-case не включає (include) додаткових usecases, котристувач просто вивчає складові мережі.

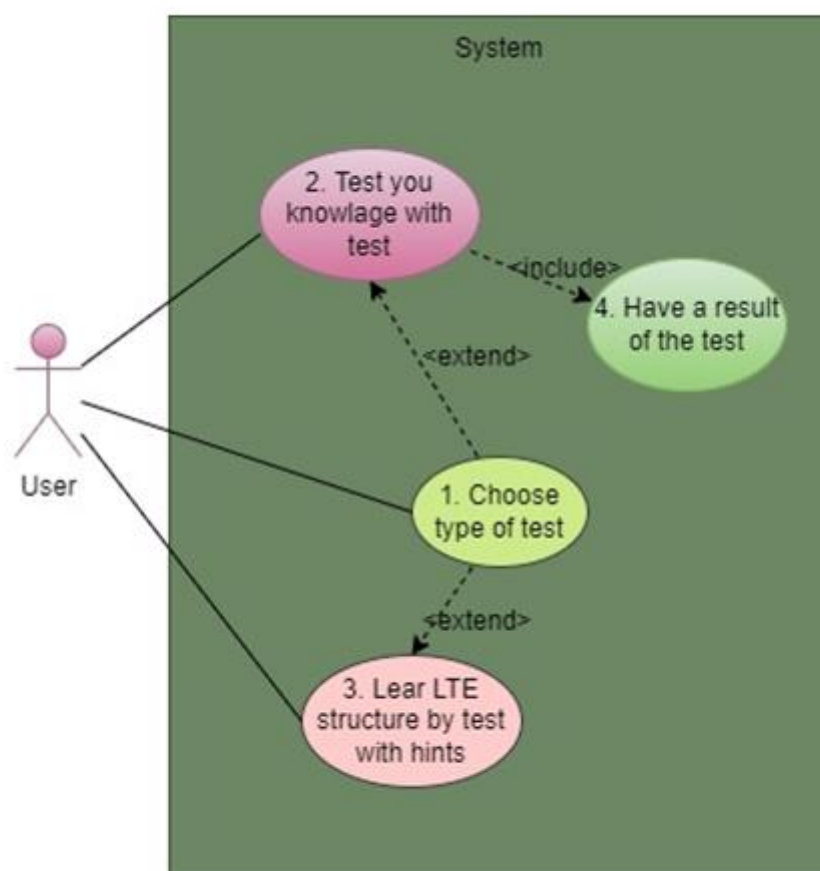


Рисунок 3.1 – Use-case UML діаграма зв'язку системи і користувачів

Далі потрібно розробити схематичний вид веб сторінки, та намалювати його в додатку векторної та растрової графіки, Paint 3D. Paint 3D — це безкоштовна програма від корпорації Microsoft, яка містить як базові, так і розширені художні інструменти, але нам буде достатньо базових можливостей для приблизного розміщення елементів інтерфейсу.

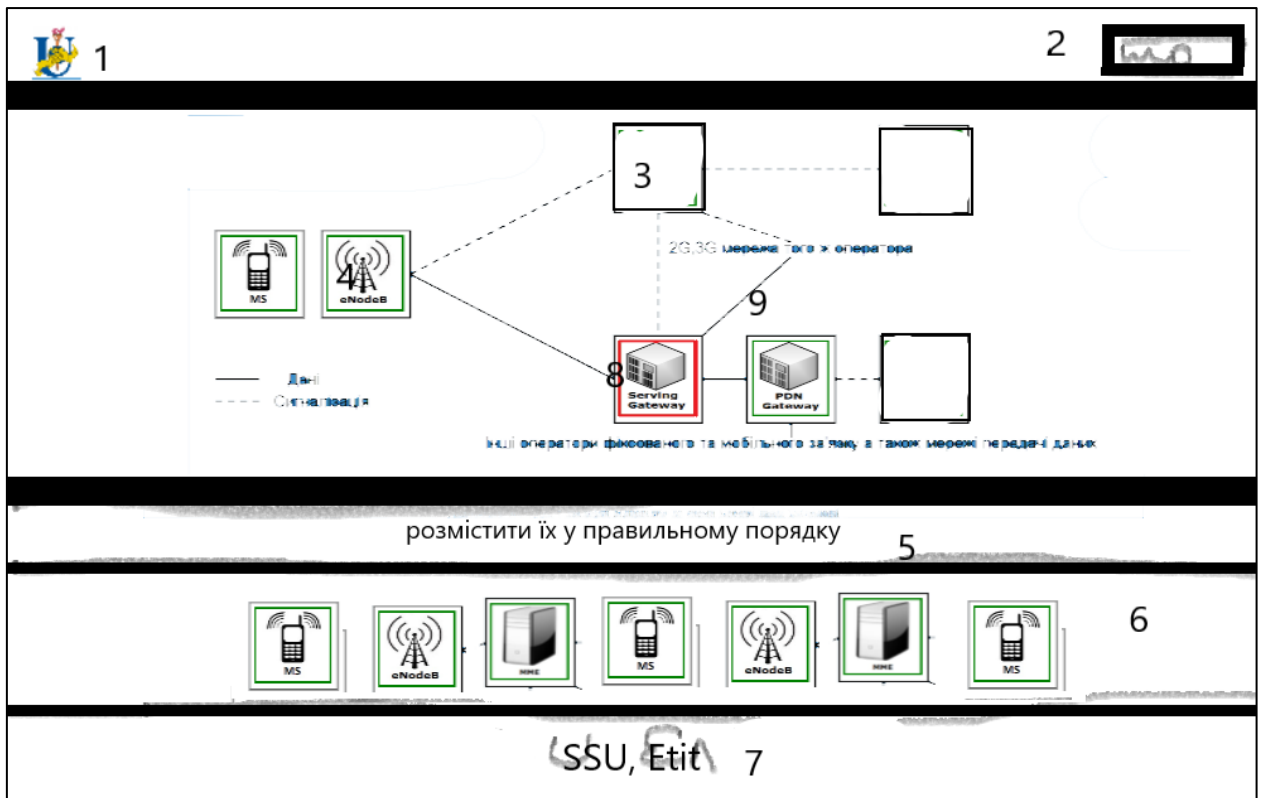


Рисунок 3.2 – Макет сайту

Макет сайту складається із 5 горизонтальних компонентів: хедер, футер, та три блоки контенту. У хедері будуть розміщені відповідно до рисунку:

1. Емблема СумДУ
2. Кнопка на зміну моду

У футері містяться:

7. Посилання на сайт СумДУ

У головному блоці контенту буде розміщена схема з описами шляхів переходу ір пакетів по мережі, яка має деякі позначення:

9. Пуста схема мережі LTE

По схемі розкидані у відповідних місцях квадрати, які є цілю для елементів, які можна переміщати, тобто drag-target. В залежності від того, який елемент буде вставлений у це квадрат, він підсвітиться зеленим або, червоним в залежності чи правильно був вставлений компонент.

3. Пустий квадрат цілю в яку будуть переміщувати користувач рухомі елементи з блоку нижче.

4. Блок, що є в правильній позиції – підсвічується зеленим

8. Блок, що є не в правильній позиції - підсвічується червоним

Центральний горизонтальний блок – це завдання, яке треба виконати

5. Завдання

Блоком нижче знаходиться контейнер з контентом, а саме із зображеннями, які користувач може брати і перетаскувати в пусті блоки. Та відповідно бачити чи правильно він додав їх відповідно до кольору яким вони підсвітілись.

6. Зображення компонентів мережі, які користувач може переміщувати мишкою

Створюємо index.html та за допомогою display: flex будуюмо прототип сайту за наведеним макетом.

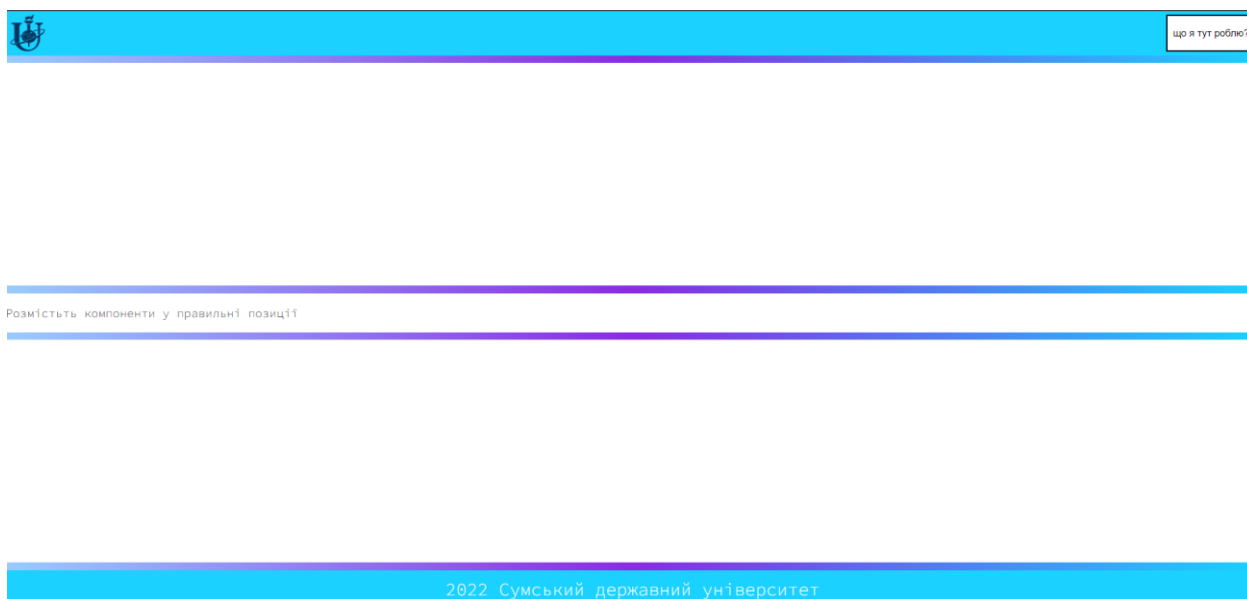


Рисунок 3.3 – Прототип сайту

На прототип була додана емблема СумДУ, кнопка з інструкцією . В головній частині три блоки контенту, який поділений різнокольоровими блоками.

Опишемо технологію flex-box. Довгий час основним надійними, хоч і важким, але кросбраузерними інструментами, доступними для створення макетів CSS – схожий на наведений вище, були такі властивості: як плаваючі елементи (flow) та позиціонування (position). Вони є працюючим варіантом

розробки сайту, але певним чином вони важкі в реалізації та обмежують і можуть розчарувати, бо не завжди можливо виконати ними задумане.

Наприклад наступні варіанти реалізації були б неможливі з комбінації flow та position:

- Вертикальне центрування блоку, в нашому проєкті ця річ знадобилась.
- Зробити так, щоб усі дочірні елементи контейнера займали рівну кількість ширини/висоти – ця річ хоч і не згадобилась, ає вона є часто нецільовою.
- Зробити стовпці в макеті рівної висоти, але не зважаючи на те яка кількість вмісту в них лежить.

Як бачите flexbox значно полегшив нам розробку макету сайту, адже він робить побудову сайту набагато простішою.

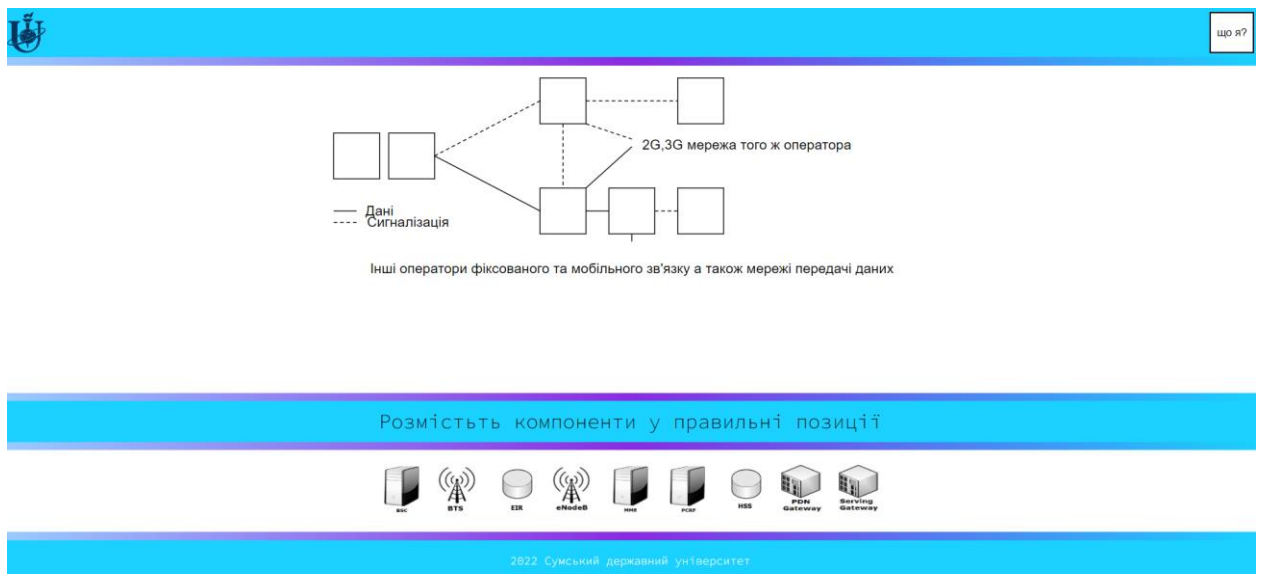


Рисунок 3.4 – Прототип сайту

Для того, щоб додаток залишився розширюваним то картинки в блоці з якого вони вибираються, додаються за допомогою коду генератора в файлі JS. Серед обов'язкових властивостей додаємо id, margin, source (всі адреси файлів зберігаються у масиві файлів)

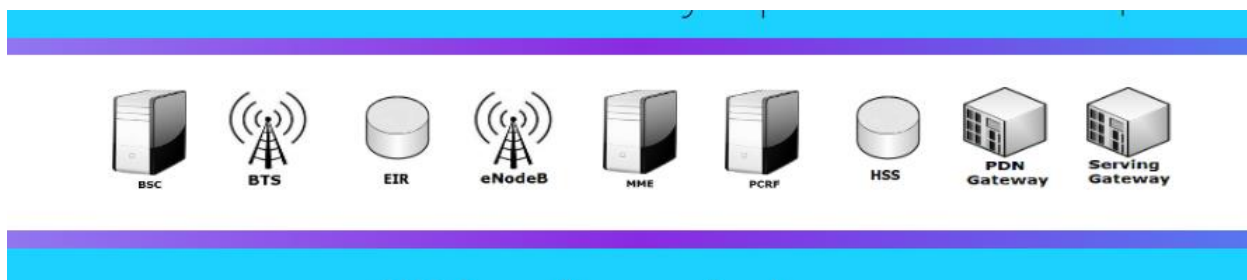


Рисунок 3.5 – Прототип сайту

Схема мережі зроблена з використанням технології векторного зображення SVG. Масштабована векторна графіка (SVG) — це формат векторних файлів, котирий був застосований на нашому сайті, він найбільш зручний для Інтернету. На відміну від растрових файлів на основі пікселів, які мають конкретний розмір, таких як JPEG, PNG чи інші, векторні файли зберігають зображення з використанням математичних формул і розрахунків, тобто за допомогою математичних формул та рівнянь на основі точок і прями. Для нас це означає, те що такі векторні зображенн, як SVG, можна значно зменшувати чи збільшувати, робити зміни майже без втрати якості, що робить їх прекрасним рішенням для логотипів і складної онлайн 2д графіки та не тільки.

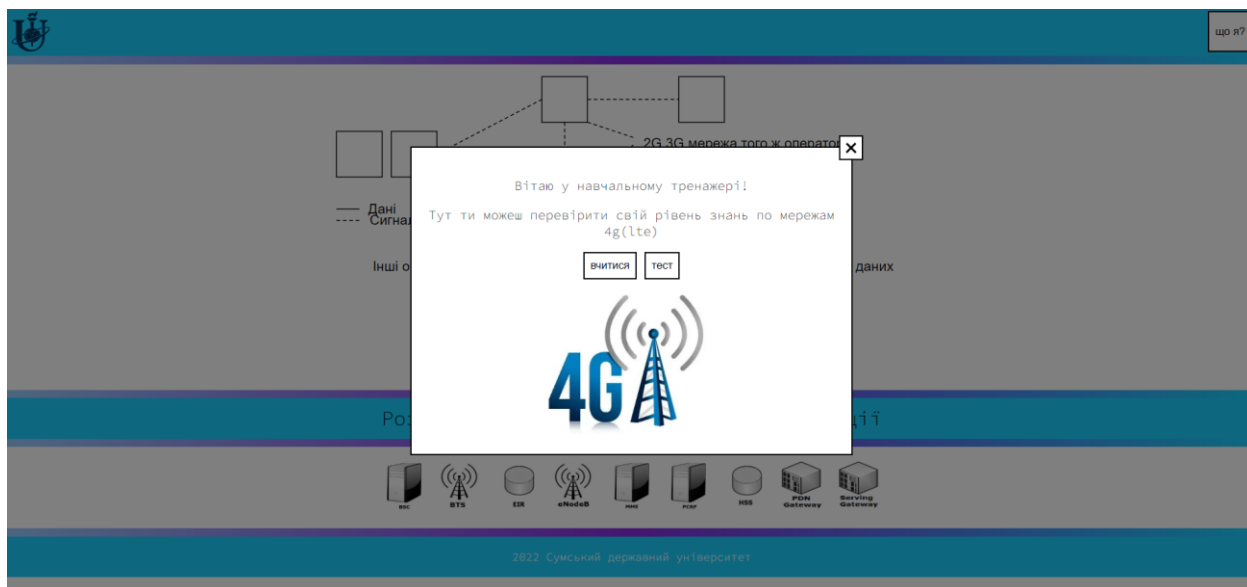


Рисунок 3.6 – Привітання

SVG набули великої популярні серед веб-дизайнерів та веб-розробників. SVG пишуться дуже подібно до коду XML, тобто фактично вони зберігають текстову інформацію як буквальний текст, а не як геометричні форми, тобто

зберігають опис зображення, а не саме зображення. Ця особливість дозволяє пошуковим системам, таким як Google чи іншим, вважати графіку SVG за ключові слова – мета дані, що потенційно може допомагати сайту краще індексуватися і просуватися в пошуковому рейтингу.

Для більш живого вигляду сайту були використанні псевдо елементи та псевдокласи. Псевдоклас CSS `:hover` відповідає, коли користувач взаємодіє певним чином з елементом: користувач за допомогою мишки наводить на елемент ціль – це і є тригером спрацювання цього псевдокласу. Буває велика кількість псевдоелементів та псевдокласів, зокрема для створення закриваючої кнопки із хрестиком були застосовані `:after` та `:before`.

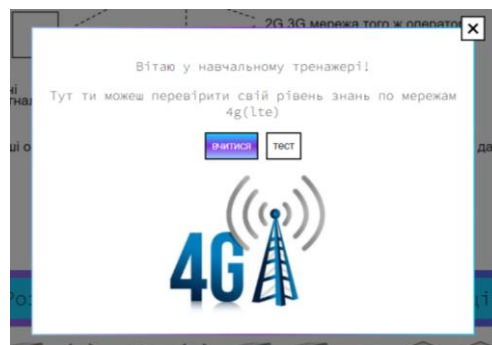


Рисунок 3.7 – Робота hover

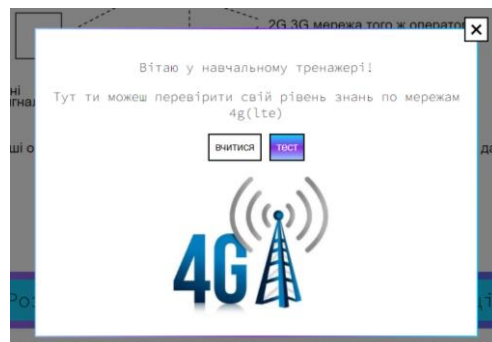


Рисунок 3.8 – Робота hover

З використанням сучасних можливостей JS, HTML5, CSS3 в цьому підрозділі, ми розробили сайт, далі нам треба його протестувати.

3.2 Тестування тренажеру

Це підрозділ виділено для того щоб протестувати додаток, на те чи виконує він всі поставлені йому задачі. Перевіримо роботу функції Test your knowledge with test.

Запускаємо додаток, в поп ап вікні обираємо тест, встановлюємо деякі елементи в правильнв позиціїї, деякі ні, отримуємо результат тесту.

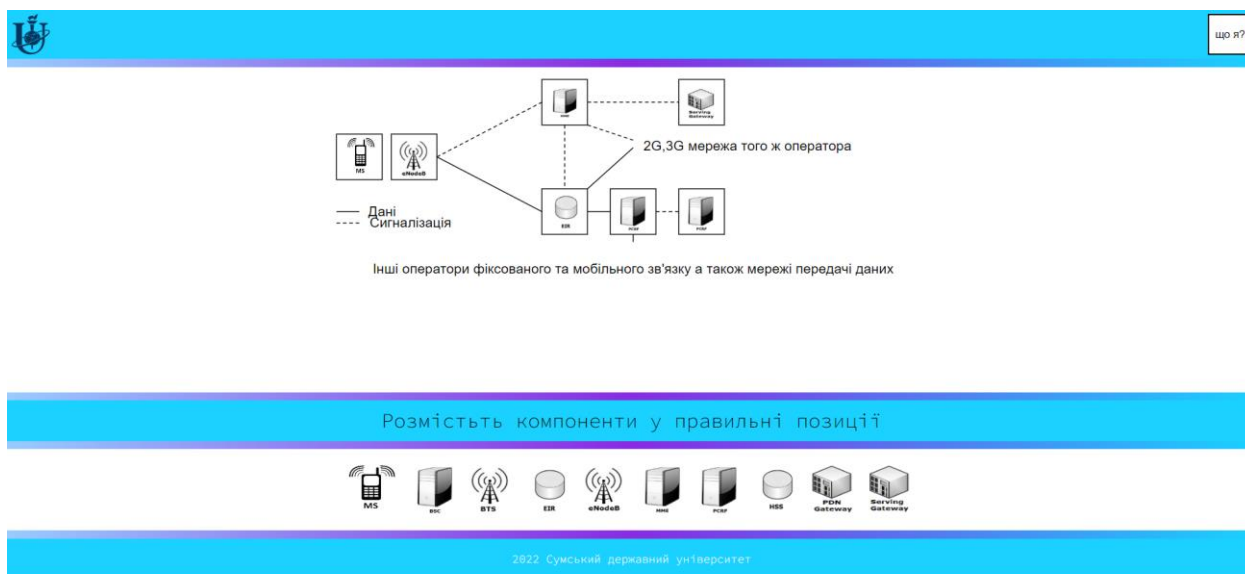


Рисунок 3.9 – Проходження тесту

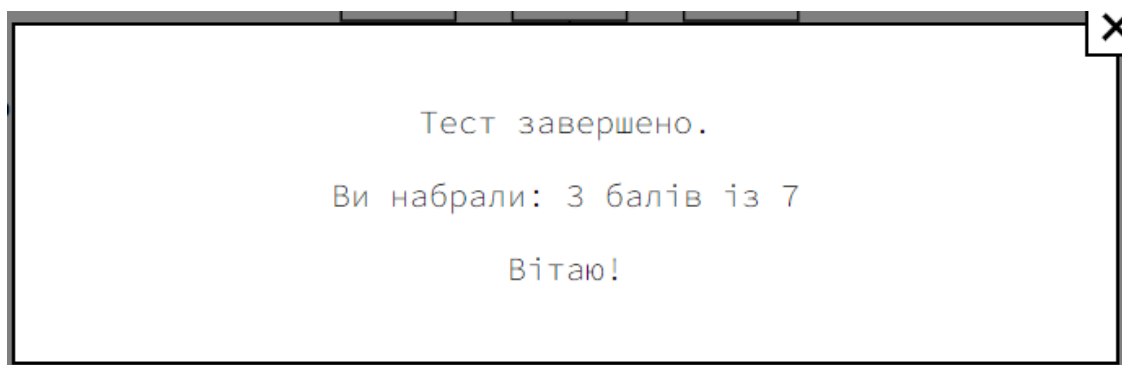


Рисунок 3.10 – Результат тесту

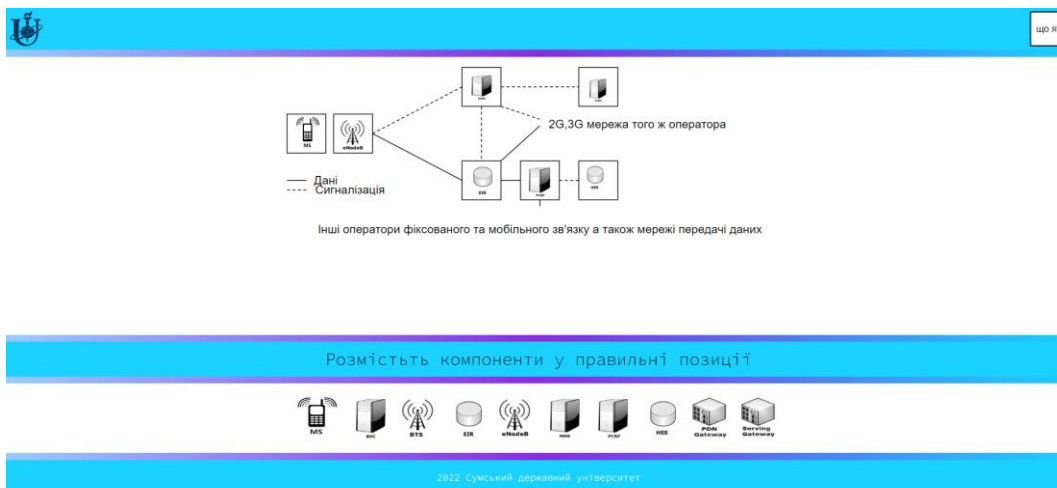


Рисунок 3.11 – Проходження тесту

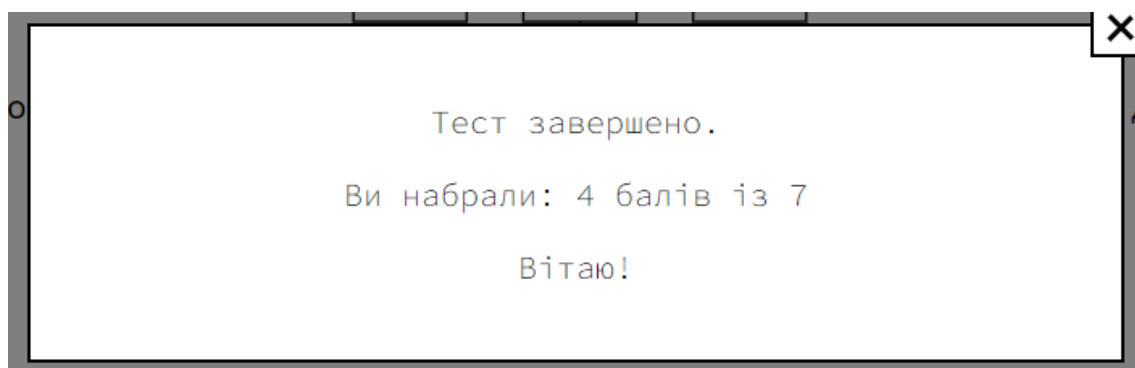


Рисунок 3.12 – Результат тесту

Виконаєм ще одну ту ж саму перевірку, щоб впевнитися, що ця функція працює. Відповідно до схеми додаємо лише 4 елементи на правильне місце, і в отриманому результаті бачимо, що ми вибрали 4 пункти з 7 правильно.

Після завершення всіх запланованих тестів виконаних у цьому підрозділ будемо вважати етап тестування завершеним.

ВИСНОВОК

У кваліфікаційній магістерській роботі проведено аналіз предметної області за напрямом розробки, експлуатації, використання, та особливостей функціонування мереж сучасного мобільного зв'язку, а саме 4G(LTE). Був розглянутий шлях еволюції від 1G до 4G, переваги та недоліки кожного покоління перед попереднім та наступним. Розглянуто в подробицях особливості LTE, які роблять дану технологію лідером ринку мобільних мереж.

Була широко описана архітектура систем LTE та основні компоненти та їх роль в передачі даних-пакетів. Був розглянутий протокол встановлення з'єднання SIP, приклад та принцип його застосування.

У кваліфікаційній магістерській роботі було приведено оцінку функціональності застосунку для моделювання та симуляції робот мереж Cisco Packet Tracer. Були виділені його функціональні характеристики та переваги його використання.

Була побудована мережа стільникового зв'язку, яка підтримує технологію 4G (LTE), та завдяки приладам та протоколам, ще є наявність підтримки технології 3G. Була протестована посилка повідомлень між приладами (ноутбуки, планшети, смартфони).

Додатково в цій роботі була розглянута і коротко описана мова браузерної розробки JS - інструменту розробки даного веб додатку.

Додатково дана робота описує етапи створення веб застосунку для побудови мереж LTE (4G), веб-технологій, які були для цього застосовані. Була збудована діаграма варіантів використання додатку, розроблені функції і функціонал відповідно до діаграми, а також був проведений етап тестування застосунку.

Робота задовільняє поставленій задачі із першого розділу, та відповідає меті, яка була поставлена на початку розробки проекту.

СПИСОК ЛІТЕРАТУРИ

1. Організація комп'ютерних мереж: підручник: для студ. спеціальності 122 / Ю. А. Тарнавський, І. М. Кузьменко. – Київ : КПІ ім. Ігоря Сікорського, 2018. – 259 с.
2. Карпенко М. Ю. Конспект лекцій з курсу «Комп'ютерні мережі» (для студентів усіх форм навчання спеціальностей 122 – Комп'ютерні науки, 151 – Автоматизація та комп'ютерно-інтегровані технології, 126 – Інформаційні системи та технології) / М. Ю. Карпенко, Н. В. Макогон; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2019. – 99 с.
3. Комп'ютерні мережі : курс лекцій / Ю. В. Волосяк. – Миколаїв: МНАУ, 2019. – 203 с.
4. Пропускна здатність мережі LTE [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: <https://ua.waykun.com/articles/ocinkapropusknoi-zdatnosti-merezhi-standartu-lte.php>
5. ДОСЛІДЖЕННЯ РЕЖИМІВ ПЕРЕДАЧІ МЕРЕЖІ LTE 4G / УДК 621.395.743, В.Л. Дунець канд. техн. наук, А.М. Семейн (Тернопільський національний технічний університет імені Івана Пулюя, Україна) 2018. – 1с.
6. Комп'ютерні мережі. Протоколи, технології, обладнання : навч. посіб. для студ. спец. 125 «Кібербезпека» / В. М. Базилевич, Д. Б. Мехед, Ю. М. Ткач. – Ніжин: ФОП Лук'яненко В.В. ТПК «Орхідея», 2018. – 108 с.
7. Cisco Packet Tracer [Електронний ресурс] — Режим доступу до ресурсу: <https://www.netacad.com/>
8. Open Source Used In Cisco Packet Tracer 8.1 / Cisco Systems, Inc., 2021. – 83 с.
9. ECMAScript® 2021 Language Specification / Jordan Harband (@ljharb), Shuyi Guo (@_shu), Michael Ficarra (@smooshMap), Kevin Gibbons (@bakkoting), 2021. – 879 с.

10. Marijn Haverbeke. Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming 3rd Edition. – San Francisco: No Starch Press, 2018. – 472 с.
11. Getting Started with GNS3 [Електронний ресурс] — Режим доступу до ресурсу: <https://docs.gns3.com/docs/>
12. Пропускна здатність мережі LTE [Електронний ресурс]:[Веб-сайт]. – Електронні дані. – Режим доступу: <https://ua.waykun.com/articles/ocinkapropusknoi-zdatnosti-merezhi-standartu-lte.php>

ДОДАТКИ

Файл index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Kanit:ital,wght@0,200;0,400;0,600;0,800;1,200;1,400;1,600;1,800" rel="stylesheet">
  <title>4G LTE</title>
  <script src="app.js"></script>
  <script src="jquery.js"></script>
  <script src="jquery-ui.js"></script>
  <script src="raphael.js"></script>
  <script src="DragObject.js"></script>
  <script src="DropTarget.js"></script>
  <script src="dragMaster.js"></script>
  <script src="helpers.js"></script>
</head>
<body>

  <div class="popup-back">
    <div class="rulesPopUp">
      <p>Вітаю у навчальному тренажері!</p>
      <p>Тут ти можеш перевірити свій рівень знань по мережам 4g(lte)</p>
      <button class="button" id="buttonPopUpLearn">вчитися</button>
      <button class="button" id="buttonPopUpTest">тест</button>
      <br>
      
      <button class="close-button" id="buttonPopUpClose"></button>
    </div>
  </div>

  <div class="popup-back">
    <div class="rulesPopUp">
      <p>Тест завершено.</p>
      <p>Ви набрали: <span>3</span> балів із 7</p>
      <p>Вітаю!</p>
      <button class="close-button" id="buttonPopUpClose"></button>
    </div>
  </div>

```



```

<div class="container">
  <div class="wrapper">
    <header>
      
      <button id="buttonPopUpOpen" class="button">щю я?</button>
    </header>
    <div class="color"></div>
    <div class="dropable"></div>
    <div class="schema">

```

```

<svg height="390" version="1.1" width="768" xmlns="http://www.w3.org/2000/svg" \
xmlns:xlink="http://www.w3.org/1999/xlink" style="overflow: hidden; position: relative;">
  <desc style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);">Created with Rapha?l 2.1.2</desc>
  <defs style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);"></defs>
  <rect x="21.224" y="82.3440000000001" width="56.12" height="56.12" rx="0" ry="0" fill="#ffffff" stroke="#000000"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" id="0" stroke-width="1.1224"></rect>
  <rect x="88.568" y="82.3440000000001" width="56.12" height="56.12" rx="0" ry="0" fill="#ffffff" stroke="#000000"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" id="1" stroke-width="1.1224"></rect>
  <rect x="273.76399999999995" y="149.688" width="56.12" height="56.12" rx="0" ry="0" fill="#ffffff" stroke="#000000"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" id="2" stroke-width="1.1224"></rect>
  <rect x="357.94399999999996" y="149.688" width="56.12" height="56.12" rx="0" ry="0" fill="#ffffff" stroke="#000000"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" id="3" stroke-width="1.1224"></rect>
  <rect x="442.12399999999997" y="149.688" width="56.12" height="56.12" rx="0" ry="0" fill="#ffffff" stroke="#000000"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" id="4" stroke-width="1.1224"></rect>
  <rect x="273.76399999999995" y="15" width="56.12" height="56.12" rx="0" ry="0" fill="#ffffff" stroke="#000000"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" id="5" stroke-width="1.1224"></rect>
  <rect x="442.12399999999997" y="15" width="56.12" height="56.12" rx="0" ry="0" fill="#ffffff" stroke="#000000"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" id="6" stroke-width="1.1224"></rect>
  <path fill="none" stroke="#000000" d="M144.688,110.4040000000001L273.76399999999995,43.06"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224" stroke-dasharray="4.4896,3.367200000000004"></path>
  <path fill="none" stroke="#000000" d="M301.82399999999996,71.12L301.82399999999996,149.688"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224" stroke-dasharray="4.4896,3.367200000000004"></path>
  <path fill="none" stroke="#000000" d="M329.88399999999996,43.06L442.12399999999997,43.06"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224" stroke-dasharray="4.4896,3.367200000000004"></path>
  <path fill="none" stroke="#000000" d="M144.688,110.4040000000001L273.76399999999995,177.748"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224"></path>
  <path fill="none" stroke="#000000" d="M329.88399999999996,177.748L357.94399999999996,177.748"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224"></path>
  <path fill="none" stroke="#000000" d="M414.06399999999996,177.748L442.12399999999997,177.748"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224" stroke-dasharray="4.4896,3.367200000000004"></path>
  <path fill="none" stroke="#000000" d="M329.88399999999996,71.12L386.00399999999996,89.82666666666667"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224" stroke-dasharray="4.4896,3.367200000000004"></path>
  <path fill="none" stroke="#000000" d="M329.88399999999996,149.688L386.00399999999996,99.18"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224"></path>
  <path fill="none" stroke="#000000" d="M329.88399999999996,177.748L357.94399999999996,177.748"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224"></path>
  <path fill="none" stroke="#000000" d="M414.06399999999996,177.748L442.12399999999997,177.748"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224" stroke-dasharray="4.4896,3.367200000000004"></path>
  <path fill="none" stroke="#000000" d="M329.88399999999996,71.12L386.00399999999996,89.82666666666667"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224" stroke-dasharray="4.4896,3.367200000000004"></path>
  <path fill="none" stroke="#000000" d="M329.88399999999996,149.688L386.00399999999996,99.18"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224"></path>
  <text x="386.00399999999996" y="93.56800000000001" text-anchor="middle" font-family="&quot;Arial&quot;"
  font-size="17px" stroke="none" fill="#000000"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0); text-anchor: middle; font-family: Arial; font-size: 17px;">
  <tspan dy="8.599995773315442" dx="140.599995773315442"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);">2G,3G мережа того ж оператора</tspan></text>
  <path fill="none" stroke="#000000" d="M386.00399999999996,205.808L386.00399999999996,215.16133333333332"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224"></path>
  <text x="386.00399999999996" y="218.27911111111111" text-anchor="middle" font-family="&quot;Arial&quot;"
  font-size="17px" stroke="none" fill="#000000"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0); text-anchor: middle; font-family: Arial; font-size: 17px;">
  <tspan dy="35.600009167141366"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);">Інші оператори фіксованого та мобільного зв'язку а також мережі передачі даних</tspan>
  <path fill="none" stroke="#000000" d="M21.224,177.748L49.284,177.748"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224"></path>
  <path fill="none" stroke="#000000" d="M21.224,191.778L49.284,191.778"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);" stroke-width="1.1224" stroke-dasharray="4.4896,3.367200000000004"></path>
  <text x="77.344" y="177.748" text-anchor="middle" font-family="&quot;Arial&quot;" font-size="17px"
  stroke="none" fill="#000000" style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0); text-anchor: middle; font-family: Arial; font-size: 17"
  <tspan dy="5.599995468139639" style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);">Дани</tspan></text>
  <text x="88.568" y="191.778" text-anchor="middle" font-family="&quot;Arial&quot;" font-size="17px"
  stroke="none" fill="#000000" style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0); text-anchor: middle; font-family: Arial; font-size: 17"
  <tspan dy="5.59999668842765" dx="23.599995773315442"
  style="-webkit-tap-highlight-color: □rgba(0, 0, 0, 0);">Сигналізація</tspan></text></svg>

```

```

    </div>
    <div class="color"></div>
    <div class="task">
      <p>Розмістіть компоненти у правильні позиції</p>
    </div>
    <div class="color"></div>
    <div class="component-parent">
      <div id="component">

      </div>
    </div>
    <div class="color"></div>
    <footer>
      <p>2022 <a href="https://sumdu.edu.ua/uk/">Сумський державний університет</a></p>
      <!-- <p><br>розробник</p> -->
    </footer>
  </div>
</div>
</html>

```

Файл style.css

```

body {
  font-family: 'Source Code Pro', monospace;
  margin: 0;
  height: 100%;
}

.container {
  display: flex;
  flex-direction: column;
  min-height: 100%;
}

.wrapper {
  flex: 1 1 auto;
}

.task{
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  height: 50px;
  background-color: #1ad1ff;
  color: black;
  font-size: 25px
}

```

```
header {
  display: flex;
  justify-content: space-between;
  min-height: 50px;
  background-color: #1ad1ff;
  color: white;
}

header button {
  margin: 5px;
}

header img{
  width: 50px;
  height: 50px;
  margin: 5px;
}

.color {
  background: linear-gradient(to right, #99ccff, blueviolet, #1ad1ff);
  height: 10px;
}

.schema{
  height: 400px;
  text-align: center;
}

.content {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  background: linear-gradient(to right, #99ccff, blueviolet, #1ad1ff);
  height: 300px;
}

.component-parent {
  display: flex;
  flex-direction: row;
  justify-content: center;
  align-items: center;
  height: 100px;
}

.component {
  display: flex;
  flex-direction: row;
  justify-content: center;
  text-align: center;
  align-items: center;
}

.dragable {
  cursor: move;
}
```

```

.dropable {
  height: 70px;
  width: 70px;
  border: 2px solid black;
  background-color: white;
  margin: 10px;
}

footer {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 50px;
  background-color: #1ad1ff;
  color: white;
  font-size: 1.5em;
}

footer a, footer p{
  font-size: 14px;
  color: white;
  text-decoration: none;
}

footer a:hover {
  text-decoration: underline;
}

.button {
  padding: 0.5em;
  border: 2px solid black;
  color: black;
  background-color: white;
}

.button {
  padding: 0.5em;
  border: 2px solid black;
  color: black;
  background-color: white;
}

.button:hover {
  border: 2px solid black;
  background: linear-gradient(to top, #99ccff, blueviolet, #1ad1ff);
  color: #ffffff;
}

.popup-back {
  background: rgba(0, 0, 0, 0.5);
  width: 100%;
  height: 100%;
  position: absolute;
  top: 0;
  left: 0;
  display: none;
  justify-content: center;
  align-items: center;
  z-index: 5;
}

.rulesPopUp{
  height: auto;
  width: 500px;
  background: white;
  color: black;
  border: 2px solid black;
  padding: 20px;
  position: relative;
  text-align: center;
}

```

```

.rulesPopUp img{
  width: 50%;
  height: 50%;
}

.rulesPopUp:hover {
  border-image: linear-gradient(to right, #99ccff, blueviolet, #1ad1ff) 5;
}

.close-button {
  /* border-radius: 50%; */
  padding: 0.5em;
  width: 30px;
  height: 30px;
  border: 2px solid black;
  color: black;
  position: absolute;
  top: -15px;
  right: -15px;
  background-color: white;
}

.close-button:hover {
  border: 2px solid black;
  background: linear-gradient(to top, #99ccff, blueviolet, #1ad1ff);
  color: #ffffff;
}

.close-button::before {
  content: " ";
  position: absolute;
  display: block;
  background-color: black;
  width: 2px;
  left: 12px;
  top: 5px;
  bottom: 5px;
  transform: rotate(45deg);
}

.close-button::after {
  content: " ";
  position: absolute;
  display: block;
  background-color: black;
  height: 2px;
  top: 12px;
  left: 5px;
  right: 5px;
  transform: rotate(45deg);
}

```

Файл dragobject.js

```

var startingPositions=[];
function DragObject(picture) {
    picture.dragObject = this;

    dragMaster.makeDraggable(picture);

    var rememberingPosition;
    var mouseOffset;

    this.toString = function() {
        |   return picture.id;
    };

    this.onDragStart = function(offset) {
        |   var s = picture.style;
        |   rememberingPosition = {top: s.top, left: s.left, position: s.position};
        |   s.position = 'absolute';

        |   mouseOffset = offset;
    };

    this.hide = function() {
        |   |   |   picture.style.display = 'none' ;
    };

    this.hide = function() {
        |   |   |   picture.style.display = 'none' ;
    };

    this.onDragMove = function(x, y) {
        |   picture.style.top = y - mouseOffset.y +'px';
        |   picture.style.left = x - mouseOffset.x +'px';
    };

    this.onDragSuccess = function(dropTarget) {
        |
        |   };

    this.onDragFail = function() {
        |   var s = picture.style;
        |   s.top = rememberingPosition.top;
        |   s.left = rememberingPosition.left;
        |   s.position = rememberingPosition.position;
    };

    this.show = function() {
        |   picture.style.display = '' ;
    };
}

```

Файл droptarget.js

```
var language;
var typeClass ;
var isTest;
var allDevices=0;
var mistakes=0;
var rectlength;
var browser;

function checkFinal(){
    console.log(allDevices);
    var mass=document.getElementById('dragObjects').getElementsByTagName('img');
    var right=0;
    var wrong=0;
    var all;
    for(var i=0; i<mass.length; i++) {
        if($(mass[i]).data("checker")==="right")right++;
        if($(mass[i]).data("checker")==="wrong")wrong++;
    }
}
```

```

J
function isRight(n,elem,left,top){
    var k
    if(browser=="opera" || browser=="ie"){
        | k=0.06*rectlength;
    }
    else{
        | if(rectlength<$(elem).width()+6){
        |     k=0;
        | }
        | if(rectlength<$(elem).width()+12){
        |     k=0.0028*$(window).width();
        | }
        | else{
        |     k=0.004*$(window).width();
        | }
    }
}

var timer=300;
var perc=70;
top=top+k;
left=left+k;

if(isTest==1){
    |(elem).offset({top:top,left:left});
    |(elem).hide( "scale",
    |{percent:perc , direction: 'both',origin: ['middle']}, timer );
    |(elem).show( "scale",
    |{percent: perc, direction: 'both',origin: ['middle']}, timer );
    /* |(elem).fadeOut(timer);
    |(elem).offset({top:top+k,left:left+k});
    |(elem).fadeIn(timer);*/
    |//$(elem).show("medium");
    |if(n==0){
    |    |(elem).data("checker","wrong");
    }
}

```

Файл dragmaster.js


```
var dragMaster = (function() {  
    var dragObject;  
    var mouseDownAt;  
  
    var currentDropTarget;  
  
    function mouseDown(e) {  
        e = fixEvent(e);  
        if (e.which!=1) return;  
  
        mouseDownAt = { x: e.pageX, y: e.pageY, element: this };  
  
        addDocumentEventHandlers();  
  
        return false;  
    }  
  
    function mouseMove(e){  
        e = fixEvent(e)  
  
        // (1)  
        if (mouseDownAt) {  
            if (Math.abs(mouseDownAt.x-e.pageX)<5 && Math.abs(mouseDownAt.y-e.pageY)<5) {  
                return false;  
            }  
  
            // (2)  
            dragObject.onDragMove(e.pageX, e.pageY);  
  
            // (3)  
            var newTarget = getCurrentTarget(e);  
  
            // (4)  
            if (currentDropTarget != newTarget) {  
                if (currentDropTarget) {  
                    currentDropTarget.onLeave();  
                }  
                if (newTarget) {  
                    newTarget.onEnter();  
                }  
                currentDropTarget = newTarget;  
            }  
  
            // (5)  
            return false;  
        }  
    }  
}
```

```

function mouseUp(){
    if (!dragObject) { // (1)
        mouseDownAt = null;
    } else {
        // (2)
        if (currentDropTarget) {
            currentDropTarget.accept(dragObject);
            dragObject.onDragSuccess(currentDropTarget);
        } else {
            dragObject.onDragFail();
        }
        dragObject = null;
    }
    // (3)
    removeDocumentEventHandlers();
}

```

```

function getMouseOffset(target, x, y) {
    var docPos = getOffset(target);
    return {x:x - docPos.left, y:y - docPos.top};
}

```

Файл app.js

```

document.addEventListener('DOMContentLoaded', () => {
    document.getElementById('buttonPopUpOpen').addEventListener('click', () => {
        document.querySelector('.popup-back').style.display = "flex";
    })

    document.getElementById('buttonPopUpClose').addEventListener('click', () => {
        document.querySelector('.popup-back').style.display = "none";
    })

    document.getElementById('buttonPopUpCloseSmaller').addEventListener('click', () => {
        document.querySelector('.popup-back').style.display = "none";
    })
}

```

```

function generateImg(source) {
  const img = document.createElement('img');
  img.src = 'images/components/'+source;
  img.style.width = '60px';
  img.style.height = '60px';
  img.style.margin = '5px';
  img.id = source.substring(0, source.indexOf('.'));
  img.draggable = true;
  img.classList.add('draggable');
  return img;
}

function generateSVG(){
  const svg = document.createElement('svg');
  svg.setAttribute('width', '100%');
  svg.setAttribute('height', '100%');
  svg.setAttribute('viewBox', '0 0 1000 1000');
  svg.setAttribute('preserveAspectRatio', 'xMidYMid meet');
  svg.setAttribute('xmlns', 'http://www.w3.org/2000/svg');
  svg.setAttribute('xmlns:xlink', 'http://www.w3.org/1999/xlink');
  svg.setAttribute('version', '1.1');
  svg.setAttribute('id', 'svg');
  return svg;
}

```

```

function generateSVG(){
  const svg = document.createElement('svg');
  svg.setAttribute('width', '100%');
  svg.setAttribute('height', '100%');
  svg.setAttribute('viewBox', '0 0 1000 1000');
  svg.setAttribute('preserveAspectRatio', 'xMidYMid meet');
  svg.setAttribute('xmlns', 'http://www.w3.org/2000/svg');
  svg.setAttribute('xmlns:xlink', 'http://www.w3.org/1999/xlink');
  svg.setAttribute('version', '1.1');
  svg.setAttribute('id', 'svg');
  return svg;
}

```

```
document.getElementById('svg').appendChild(generateSVG());
```

```

function generateImgArray() {
  const imgSources = [
    'BSC.jpg',
    'bts.jpg',
    'EIR.png',
    'enodeb.jpg',
    'mme.jpg',
    'pcrf.jpg',
    'hss.png',
    'pdngate.png',
    'servgate.png'
  ]
}

```

```
    let imgArray = [];  
    for (let i = 0; i < imgSources.length; i++) {  
      imgArray.push(generateImg(imgSources[i]));  
    }  
    return imgArray;  
  }  
  
  imgArray = generateImgArray();  
  imgArray.forEach(img => {  
    document.getElementById('component').appendChild(img);  
  });  
});
```