

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Харківський національний університет радіоелектроніки
Кафедра ЕОМ

Метод самовідновлення високомобільної комп'ютерної мережі на базі рою БПЛА

Кваліфікаційна робота

Другий (магістерський) рівень

Автор:
Афанасьєва А.М.
студ. гр. СПм-22-6

Керівник:
Ткачов В.М.
доц. кафедри ЕОМ

Харків 2024

Опис предметної області та актуальність теми

Високий рівень мобільності мережі вирішує низку сучасних проблем, таких як:

- швидкість реакції на зміни,
- гнучкість у використанні ресурсів
- підвищена ефективність комунікацій.

З цим може допомогти мережа на базі рою БПЛА, яка забезпечує:

- швидке розгортання
- динамічну адаптацію до потреб
- високу мобільність
- масштабованість

Проте недбале ставлення до створення та введення в експлуатацію мобільної мережі може призвести до постійних втрат вузлів, що призводить до втрати даних та швидкості передачі.

Мета і задачі кваліфікаційної роботи

Метою кваліфікаційної роботи є проведення досліджень з метою розробки та вдосконалення методів самовідновлення висококомобільної комп'ютерної мережі на базі рою БПЛА.

Задачі:

- розглянути основні аспекти побудови висококомобільних мереж
- розглянути можливість використання БПЛА для побудови висококомобільних мереж
- проаналізувати методи побудови та налагодження мережі на базі рою БПЛА
- проаналізувати способи покращення процесу побудови мережі на базі рою БПЛА

3

Розроблений метод

Проблема

Прогнозування проблем та їх уникнення виглядає простим: будуємо мережу, тестуємо, виявляємо проблеми, перебудовуємо. Такий процес марно витрачає ресурси та час. Потреба в постійних ітераціях побудови та тестування підкреслює необхідність покращення цього процесу.

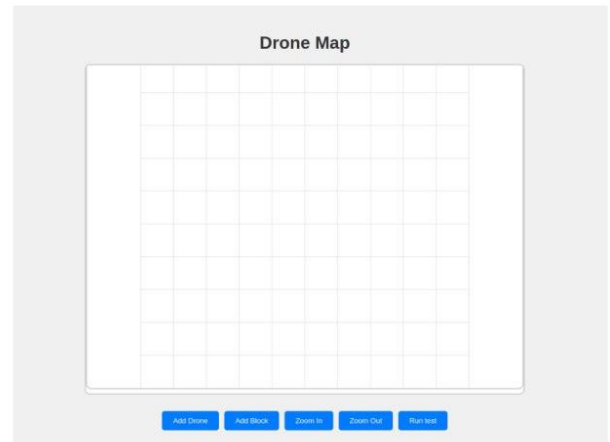
Рішення

Для того, щоб покращити цей процес, варто перед введенням мережі в експлуатацію проводити попереднє тестування та прогнозування її поведінки в різних ситуаціях. Це дозволяє виявити можливі проблеми та підготуватися до них заздалегідь, зменшуючи ризики та зберігаючи ресурси.

4

Підходи до автоматизації тестування мережі

Один з можливих підходів до вирішення даної задачі - це створення додатку, який дозволить створити тестову мережу та включати базові тести для різноманітного тестування, що будуть імітувати роботу мережі.

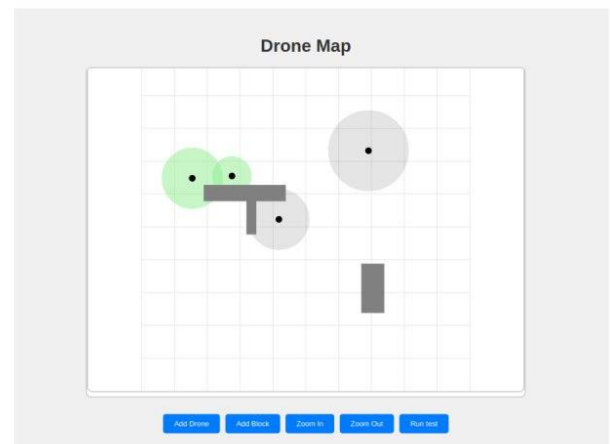


5

Особливості функціоналу програмного застосунку

Базовий функціонал додатку включає

- можливість побудови мережі
- візуалізацію з'єднань між БПЛА
- можливість розміщення блоків
- зміна масштабу мапи
- огляд детальної інформації про кожен БПЛА
- запуск тестів мережі



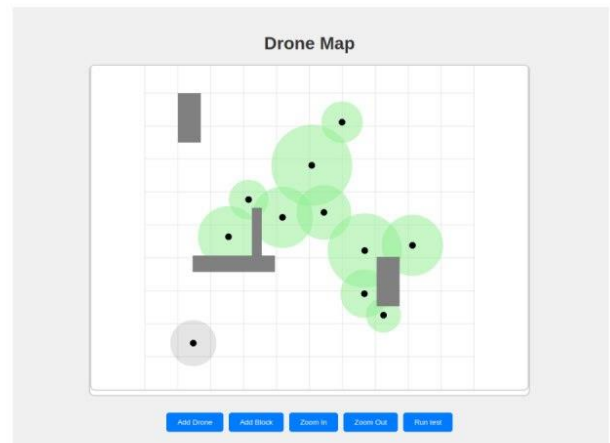
6

Дослідження мережі

Було проведено дослідження ситуації, де з ладу вийде один або кілька вузлів системи.

Також проведено порівняльний аналіз часу заміни дефектного вузла, використовуючи наступну формулу:

$$t = \frac{d(v_2 - v_1)}{v_1 v_2} + \Delta t_1 - \Delta t_2$$



7

Аналіз результатів досліджень

Під час тестування мережі з рою БПЛА, були виявлені наступні недоліки

- Якщо з ладу виходить важливий вузол системи і резервні БПЛА знаходяться на вагомій відстані від даного вузла, то це може призвести до затримки передачі даних або, в критичних випадках, їх втрати
- Важче підтримувати мережу, де кожен окремий БПЛА має невелику кількість з'єднань з іншими БПЛА

Покращення рекомендованих характеристик при проведенні наступних етапів тестування, покращило результати тестів, що доводить вагомість підходу.

8

Апробація результатів дослідження

А. М. АФАНАСЬЄВА
магістр кафедри електронних обчислювальних машин
Харківський національний університет радіоелектроніки
ORCID: 0009-0006-4622-8053

В. М. ТКАЧОВ
кандидат технічних наук, доцент,
доцент кафедри електронних обчислювальних машин
Харківський національний університет радіоелектроніки
ORCID: 0000-0002-4524-9937

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ ПОБУДОВИ ВИСОКОМОБІЛЬНИХ КОМП'ЮТЕРНИХ МЕРЕЖ НА БАЗІ РОЮ БПЛА

Існує чимала кількість переваг використання безпілотних літальних апаратів для побудови мережі на великій території в порівнянні з класичними рішеннями. А у зв'язку зі збільшенням популярності та використання безпілотних літальних апаратів (БПЛА), виникають нові проблеми, спрямовані на вивчення поточних проблем, які, теоретично, можуть бути вирішені за допомогою дронів або роїв дронів.

БПЛА відрізняються великою мобільністю, оскільки вони можуть легко переміщатися на великі відстані. Це означає, що їх можна вивести розгорнутими та переносити з одного місця на інше в залежності від потреб мережі. Наприклад, у випадку зміни топографії або виникнення нових вимог до покриття зв'язку, БПЛА можуть бути легко перенесені в нові місця для оптимального покриття. Одним з ключових переваг використання БПЛА є їхній потенціал для автономної роботи. Вони можуть бути запрограмовані для виконання різних завдань без значного людського втручання. Наприклад, вони можуть автоматично відслідковувати стан мережі та, у випадку виявлення проблем або відмови, самостійно вираховувати ситуацію шляхом відмовлення зв'язку або резервування.

БПЛА мають здатність переміщатися у тривимірному просторі, що дозволяє їм забезпечувати покриття на широкій території порівняно з традиційними методами. Це дозволяє створювати мережі з більш широким охопленням та забезпечувати доступ до зв'язку в регіонах, де раніше це було складним або неможливим. Проте створення мережі з безпілотних літальних апаратів може зіткнутися з рядом недоліків та складнощів. Спочатку, складність конфігурації може бути значним викликом. Налаштування мережі з БПЛА вимагає розуміння принципів роботи кожного дрона, вибір відповідної техніки зв'язку та налаштування параметрів мережі. Це може вимагати значних зусиль і часу від кваліфікованих фахівців. Також, обмежена пропускна здатність може бути проблемою, особливо при передачі великої кількості даних. Залежно від техніки зв'язку та кількості дронів у мережі, може виникнути обмеження на пропуску здатність мережі, що може обмежити її ефективність. Крім того, вартість впровадження та підтримки мережі з БПЛА може бути великою. Додатково, потрібно мати на увазі недовговічність дронів. Залежно від середовища, в якому працюють дрони, можуть виникати проблеми зі стійкістю зв'язку через перешкоди, електромагнітні спотворення або інші фактори, що можуть погіршити продуктивність мережі. Ще вишкляє питання щодо потенційних загроз безпеки. Мережі з БПЛА можуть бути уразливі до кібератак, перешкоди сигналу або фізичних пошкоджень, що може призвести до втрати контролю над дронами або витоку конфіденційної інформації.

Тож, хоча мережі, побудовані роси БПЛА стикаються з рядом викликів і обмежень, що потребують уважного вивчення та розв'язання, вони мають значний потенціал і переваги. Дослідження способів оптимізації конфігурації, забезпечення надійності та зменшення вартості впровадження таких мереж буде корисним для подальшого розвитку цієї техніки. Такий підхід дозволить максимально використувувати потенціал мереж БПЛА та забезпечити їх успішне впровадження в різноманітних областях застосування.

Ключові слова: безпілотні літальні апарати (БПЛА), мобільність, мережі з'єднання, керування станцією, конфі-

Висновки за роботою

Під час тестування були розглянуті різні стратегії розгортання мережі, були протестовані різні можливі сценарії, що дало можливість оглянути різні підходи до побудови мережі.

- Проведення математичного тестування перед початком реальної побудови та експлуатації значно економить час та ресурси
- Вибір стратегії побудови мережі залежить від багатьох факторів, які мають бути враховані
- Зменшення кількості з'єднань одного БПЛА з іншими в системі, збільшує ризики затримки передачі даних або втрати даних у випадку виходу з ладу одного чи кількох вузлів



Перспективні напрями досліджень



Розвиток існуючого додатку

- додавання різних типів тестування
- реалізувати можливість створення власних тестів
- додати можливість створювати акаунти та зберігати створені мережі



Створення можливості моніторингу усієї системи на одному екрані

- Під час аналізу роботи, був помічений недолік, такий як неможливість керувати усією системою з одного пристрою
- Можливість керування всією системою одночасно дозволить покращити час усунення недоліків мережі та покращить аналіз роботи мережі, яка введена в експлуатацію

ДОДАТОК Б

Фрагменти коду створеного програмного застосунку

Лістинг Б.1 – Створення початкового вікна (файл App.js)

```

const [drones, setDrones] = useState([]);
const [blocks, setBlocks] = useState([]);
const [scale, setScale] = useState(1);
const [isDroneModalOpen, setIsDroneModalOpen] =
useState(false);
const [isBlockModalOpen, setIsBlockModalOpen] =
useState(false);

const handleAddDrone = (radius, maxConnections) => {
  setDrones([...drones, {
    id: drones.length + 1,
    x: 50,
    y: 50,
    radius,
    maxConnections,
    connections: 0
  }]);
};

const handleAddBlock = (width, height) => {
  setBlocks([...blocks, { id: blocks.length + 1, width,
height, x: 100, y: 100 }]);
};

const handleDragEnd = (type, id, x, y) => {
  if (type === 'drone') {
    setDrones(drones.map(drone => drone.id === id ? {
...drone, x, y } : drone));
  } else if (type === 'block') {
    setBlocks(blocks.map(block => block.id === id ? {
...block, x, y } : block));
  }
};

return (
  <div className="app-container">
    <h1>Drone Map</h1>
    <div className="map-container">
      <Map scale={scale} drones={drones}
blocks={blocks} onDragEnd={handleDragEnd} />
    </div>
    <div className="controls">

```

```

        <button className="button" onClick={() =>
setIsDroneModalOpen(true)}>Add Drone</button>
        <button className="button" onClick={() =>
setIsBlockModalOpen(true)}>Add Block</button>
        <button className="button" onClick={() =>
setScale(scale + 0.1)}>Zoom In</button>
        <button className="button" onClick={() =>
setScale(scale - 0.1)}>Zoom Out</button>
    </div>
    <DroneModal
        isOpen={isDroneModalOpen}
        onRequestClose={() =>
setIsDroneModalOpen(false)}
        onAddDrone={handleAddDrone}
    />
    <BlockModal
        isOpen={isBlockModalOpen}
        onRequestClose={() =>
setIsBlockModalOpen(false)}
        onAddBlock={handleAddBlock}
    />
</div>
);
};

```

Лістинг Б.2 – Приклади методів для обробки зображення станів БПЛА на мапі (файл App.js)

```

const handleDrag = (id, x, y) => {
    setDraggedDrones(prevState =>
        prevState.map(drone =>
            drone.id === id ? { ...drone, x, y } : drone
        )
    );
};

const handleDragStop = (type, id, e, data) => {
    onDragEnd(type, id, data.x, data.y);
};

const handleDroneClick = (drone, e) => {
    e.stopPropagation(); // To prevent the map click event
    setSelectedDrone(drone);
};

const handleMapClick = () => {
    setSelectedDrone(null);
};

const isIntersecting = (drone1, drone2) => {
    const distance = Math.sqrt(Math.pow(drone1.x - drone2.x, 2)
+ Math.pow(drone1.y - drone2.y, 2));

```

```

    return distance < drone1.radius + drone2.radius;
};

const isBlocked = (drone1, drone2) => {
  return blocks.some(block => {
    const blockX1 = block.x;
    const blockY1 = block.y;
    const blockX2 = block.x + block.width;
    const blockY2 = block.y + block.height;

    const isDrone1InsideBlock = drone1.x > blockX1 &&
drone1.x < blockX2 && drone1.y > blockY1 && drone1.y < blockY2;
    const isDrone2InsideBlock = drone2.x > blockX1 &&
drone2.x < blockX2 && drone2.y > blockY1 && drone2.y < blockY2;

    return isDrone1InsideBlock || isDrone2InsideBlock;
  });
};

const getConnections = (drone) => {
  return draggedDrones.reduce((count, otherDrone) => {
    if (otherDrone.id !== drone.id) {
      const intersects = isIntersecting(drone,
otherDrone);
      const blocked = isBlocked(drone, otherDrone);
      if (intersects && !blocked) {
        return count + 1;
      }
    }
    return count;
  }, 0);
};

const getFreeSlots = (drone) => {
  return drone.maxConnections - getConnections(drone);
};

const getIntersectingDrones = (drone) => {
  return draggedDrones.filter(otherDrone => {
    if (otherDrone.id !== drone.id) {
      const intersects = isIntersecting(drone,
otherDrone);
      const blocked = isBlocked(drone, otherDrone);
      return intersects && !blocked;
    }
    return false;
  });
};

const isDroneGreen = (drone) => {
  const intersectingDrones = getIntersectingDrones(drone);
  if (intersectingDrones !== undefined &&
intersectingDrones.length > 0) {

```

```
    let droneInfo = '';
    for (const element of intersectingDrones) {
        droneInfo = droneInfo + element.radius + ' = ' +
(getFreeSlots(element) > 0) + ', current free slots = ' +
getFreeSlots(drone) + ', ';
    }
}
const hasConnections = getConnections(drone) > 0;
const hasFreeSlots = getFreeSlots(drone) > 0;
const canConnectToOthers =
intersectingDrones.some(otherDrone => getFreeSlots(otherDrone) >
0);
return hasConnections || (hasFreeSlots &&
canConnectToOthers);
};
```