

Метою моделювання є порівняння характеристик розробленого алгоритму із базовим, та перевірка самого алгоритму. Моделювання складалося з наступних етапів:

- визначення характеристик
- перевірка коректності роботи алгоритму
- моделювання окремих випадків та вимірювання
- отримання характеристик у вигляді графіків
- порівняння характеристик

#### 4.1. Визначення характеристик

Оскільки модель може дуже значно відрізнятись від технічних реалізацій і передбачити усі можливі варіанти не має можливості, при розробці програмну модель було орієнтовано на отримання найбільш незалежних від конкретної реалізації вихідних параметрів. Це пов'язано з тим, що деякі характеристики можуть не тільки змінюватися у часі та від технології до технології, а ще й мають конкретну фізичну розмірність, що заважає їх аналізу, прив'язуючи до конкретних стандартів. До того ж вони можуть залежити не від фізичної реалізації мережі, а від того контексту, у якому вона використовується. Тому перед моделюванням необхідно, базуючись на характеристиках мережі перейти до безрозмірних величин.

Основними характеристиками децентралізованих мобільних мереж є:

- середня швидкість руху вузла ( $v$ , м/сек)
- середня швидкість обміну даними ( $v_{\text{обм.}}$ , біт/сек)
- радіус зони дії ( $r_{\text{max}}$ , м)
- середня відстань між вузлами ( $r$ , м)
- розмір пакетів ( $l$ , біт)
- середній потік інформації від вузла ( $\lambda$ , біт/сек)

Основною характеристикою швидкодії мережі є середня затримка пакету у мережі ( $T$ , сек). У мережах з превентивною маршрутизацією вона залежить від потоку пакетів та швидкості передачі. Але дистанційно-векторна маршрутизація вносить додаткову затримку при встановленні або

оновленні маршруту. Також необхідно врахувати додаткове навантаження широкомовними пакетами. Отже, ефективність алгоритму маршрутизації залежить від наступних характеристик:

- середній потік службової інформації ( $\lambda_{serv}$ , біт/сек)
- середній час встановлення маршруту ( $T_{route}$ , с)

Оскільки маршрути встановлюються випадково і на невідомий час, не має сенсу простежувати залежність цих характеристик від часу, швидкості або інших параметрів мережі. Ефективність встановлення конкретного маршруту залежить від метрики цього маршруту ( $M$ , хопи). Оскільки у найгіршому випадку широкомовлення службових пакетів ведеться по усій мережі, службовий потік (для такого випадку) може бути залежним від кількості вузлів у мережі. Для того щоб не бути залежними від цієї характеристики необхідно моделювати алгоритми у режимі пошуку з зоною, що адаптивно розширюється. Було використане експоненціальне розширення зони (кожен раз збільшувалася в декілька раз), бо воно призводить до найменших затримок при установці маршруту. Але у такому випадку необхідно враховувати вплив параметра  $TTL_{base}$ , тобто початкового TTL. Оскільки дуже вірогідним (не залежно від вірогідності групування) є завжди випадіння одної ланки (флуктуація) у режимі репарації цей параметр було встановлено рівним двом. Але щоб спростити залежності друге кільце можна вважати більшим за середній маршрут при встановленні і більшим ніж максимальний розмір ураженої ділянки при відновленні маршруту, бо його величину можна підібрати виходячи із статистичних характеристик конкретної мережі.

При підтримці маршруту впливає наступний важливий фактор – вірогідність руйнування маршруту  $p_{disroute}$  за час перевірки маршруту  $T_{upd}$ . Від цієї вірогідності залежить те, як часто алгоритм повинен оновлювати маршрут. Оскільки для руйнування маршруту достатньо випаду лише однієї ланки, вірогідність руйнування можна оцінити як вірогідність виходу одного вузла із зони мовлення іншого. Оскільки траєкторії переміщення вузлів

невідомі, цю величину можна оцінити тільки приблизно засновуючись на характеристиках мережі:

$$p_{disroute}(t < T_{upd}) \sim T_{upd} * v(t)/r$$

Перевірка розмірності [(с\*м/с)/м] показує, що  $p_{disroute}$  - безрозмірна величина, тому її зручно використовувати у якості вхідного параметра моделювання. Оскільки продуктивність розробленого алгоритму залежить безпосередньо від місця та кількості зруйнованих ланок, необхідно ввести параметр вірогідності групування пошкоджених ланок  $p_{group}$ , що може залежити від різних факторів і показує наскільки великими можуть бути зруйновані частки ланцюга і наскільки далеко від відправника вони можуть знаходитися.

Також на середню затримку пакетів впливає оптимальність маршрутів, але оскільки обидва алгоритми можуть встановлювати оптимальні маршрути, а рішення про використання неоптимальних приймається алгоритмами оптимізації, цей фактор не враховується при порівнянні.

Від  $\{p_{disroute}, p_{group}\}$  неважко перейти до статистичних параметрів:

$$n_{disroute} = p_{disroute} * M - \text{кількість зруйнованих ланок}$$

$$n_{dislink} = n_{disroute} * p_{group} - \text{середній розмір пошкоджених ланцюгів}$$

$$n_{dislink}^{max} = n_{disroute} * \max(p_{group}) - \text{максимальний розмір ланцюгів}$$

Таким чином параметри  $\{n_{disroute}, n_{dislink}, M\}$  і будуть вхідними при моделюванні. Усі параметри безрозмірні і мало залежать від конкретних реалізацій, тому дозволяють більш наочно представити характеристики. Слід окремо зазначити, що якщо розраховувати  $n_{disroute}$  не для періоду  $T_{upd}$ , а для періоду відновлення або встановлення маршруту  $T_{route}^{кр.}$ , то цей період необхідно обирати таким, щоб ця кількість була менше одиниці, бо інакше алгоритм AODV не встигне відреагувати на зміни. Хоча розроблений алгоритм лише підвищить кількість службових пакетів за рахунок репарації тимчасових шляхів, тому  $T_{route}^{кр.}$  для нього пов'язано з умовною вірогідністю руйнування ланки, що відповідає за репарацію і залежить від  $n_{dislink}$ .

Час встановлення маршруту дорівнює сумі часу передачі широкомовного пакету до цілі і часу відповіді:

$$T_{route} = T_{rreq} + T_{rrep}$$

Можна виразити  $T_{route}$ , через затримку службового пакету пошуку нового маршруту у каналі передачі ( $T_{delay}^{service}$ ):

$$T_{route} = 2 * M * T_{delay}^{service}$$

Ця формула підходить для фази встановлення маршруту у базовому алгоритмі, але при відновленні необхідно урахувати час отримання інформації про помилку (за який усі послані пакети загублюються). Для найгіршого випадку, коли зруйновано тільки останні ланки (або навіть тільки одну) у базовому алгоритмі:

$$T_{route} = (2 * M + (M - 1)) * T_{delay}^{service}$$

Для відновлення маршруту у розробленому алгоритмі:

$$T_{route} = 2 * n_{disroute} * T_{delay}^{service} \text{ (для пасивного режиму)}$$

$$T_{route} = (1 \div 2) * n_{dislink}^{max} * T_{delay}^{service} \text{ (для активного режиму)}$$

Для корекції маршруту у розробленому алгоритмі:

$$T_{route} = (1 \div 2) * (M_{i+1} - M_i) * T_{delay}^{service}$$

Таким чином, поделивши  $T_{route}$  на  $T_{delay}^{service}$  отримуємо безрозмірну характеристику затримки – кількість пересилок службових пакетів у напрямку маршруту:

$$n_{direct}^{service} = T_{route} / T_{delay}^{service}, \text{ де } T_{delay}^{service} = l^{serv} / v_{обм.}$$

Для порівняння навантаження службової інформації на мережу необхідно підрахувати середній потік широкомовних пакетів пошуку. Згідно з формулою Літтла можна оцінити його через середню кількість пакетів у каналі ( $n^{broadcast}$ ) за час встановлення маршруту:

$$\lambda_{serv} = \frac{n^{broadcast}}{T_{delay}^{service}}$$

Оскільки при широкомовленні на кожній ітерації кількість пакетів зростає, середню кількість пакетів можна оцінити як суму пакетів на

кожному етапі поділену на кількість етапів, тобто фактично на TTL пакетів. Для гексагональних мереж зона широкомовлення має форму шестикутника:

$$n^{broadcast} \sim \frac{\sum_{n=1}^{TTL} (18 * n - 12)}{TTL}$$

Для базового алгоритму ця пропорція переходить у рівність. Для розробленого алгоритму при відновленні маршруту ця зона розширюється поки не буде дорівнювати розміру загубленої ділянки. До того ж кількість розширень залежить від кількості ділянок. Якщо підрахувати суму членів арифмітичної прогресії отримаємо:

$$n^{broadcast} = (1 \div 2) * \frac{n_{disroute}}{n_{dislink}} * \frac{18 * TTL * (TTL + 1)/2 - 12 * TTL}{TTL}$$

$$n^{broadcast} = (3 \div 6) * \frac{n_{disroute}}{n_{dislink}} * (3 * TTL - 1)$$

Таким чином параметри  $\{n_{direct}^{service}, n^{broadcast}\}$  і є вихідними параметрами моделювання. Отже процес характеризується відображенням:

$$\{n_{disroute}, n_{dislink}, M\} \rightarrow \{n_{direct}^{service}, n^{broadcast}\}.$$

Оскільки важко аналізувати дві функції від трьох змінних необхідно відкинути залежності, що не мають сенсу. Оскільки для найгірших випадків у базовому алгоритмі усі вихідні параметри залежать тільки від метрики маршруту, при його дослідженні достатньо отримати залежності від неї.

Для вимірювань параметру  $n^{broadcast}$  у розробленому алгоритмі найгіршим випадком є  $n_{dislink}=1$ , до того ж  $n_{disroute}/n_{dislink}$  є кількістю груп видалених ланок, тому можна ввести параметр  $n_{group}$ , який для базового алгоритму можна вважати рівним метриці  $M$ , бо той не відрізняє групи. Таким чином, для дослідження навантаження мережі необхідно отримати графіки залежності від метрики при фіксованому  $n_{group}$ .

Для вимірювань параметру  $n_{direct}^{service}$  найгіршим випадком є групування усіх випадінь ланок, тобто  $n_{dislink} = n_{disroute}$ , тому достатньо отримати залежності від метрики при різних значеннях  $n_{dislink}$ .

Вибір метрики у якості ординати пов'язано з тим, що  $\{n_{disroute}, n_{dislink}\}$  залежать від конкретної мережі і не змінюються, а метрика залежить від обраного маршруту і різна для різних вузлів.

Також, для порівняння активного і пасивного режимів роботи алгоритму, необхідно дослідити залежність затримки від  $n_{group}$ , бо при  $n_{group} = 1$  різниця між обома режимами непомітна, окрім того що активний режим може спрацювати вдвічі швидше за рахунок двосторонньої репарації.

#### 4.2. Моделювання окремих випадків.

Для перевірки працездатності розробленого алгоритму і порівняння з базовим необхідно провести симуляції для основних окремих випадків. Моделювання окремих випадків проводилося для одного маршруту, бо алгоритми, що досліджуються не залежать від навантаження мережі і у режимі з декількома маршрутами майже неможливо зібрати статистику, яку можна підвергнути оцінці. Також моделювання активного та пасивного режимів виконувалося з вимкненою координацією пакетів, що б дослідити поведінку алгоритму у гіршому випадку. Для пасивного режиму симулювалися наступні ситуації:

- встановлення нового маршруту
- загублення маршруту
- загублення ланок до встановлення маршруту
- загублення ланок після встановлення маршруту
- переміщення кінцевого вузла вправо від напрямку маршруту
- переміщення кінцевого вузла вліво від напрямку маршруту
- переміщення кінцевого вузла вбік

При установці маршруту алгоритми працювали однаково. На рис 4.1 хвиля пакетів розповсюджується у всіх напрямках від вузла-ініціатора (вузол №17). Після отримання пакету RREQ цільовий вузол (вузол №33) відправляє відповідь RREP. Різниця між алгоритмами полягала лише у тому що згідно з розробленим алгоритмом вузли додавали у таблиці маршрутизації не тільки маршрути а й інформацію про транзити.

Слід також відмітити, що обидва алгоритми відсилають пакети одразу після отримання першого пакету відповіді, навіть якщо маршрут не є оптимальним, але при отриманні більш оптимального маршруту потік пакетів одразу перемикається. Це опціональна можливість AODV, але вона дозволяє зменшити затримки на відправку пакетів у мережі, тому її було реалізовано при імплементації алгоритму у моделюючому програмному забезпеченні.

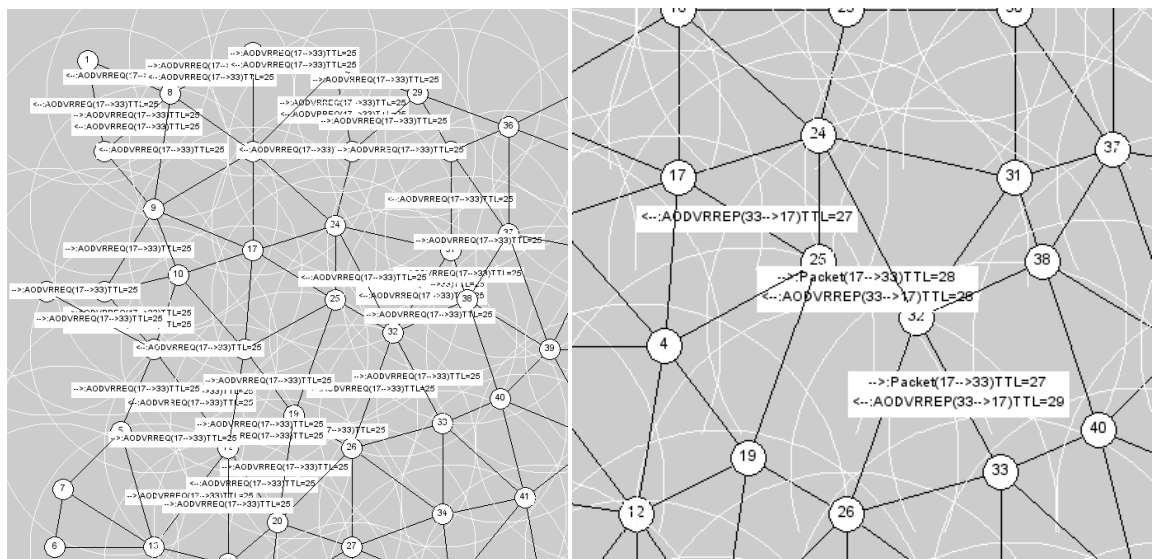


Рисунок 4.1 – Моделювання широкомовлення при установці маршруту.

При загубленні хоч одної ланки у базовому алгоритмі вузол ініціатор отримував повідомлення про помилку і починав пошук маршруту заново. Аналогічно базовий алгоритм поведився і у всіх інших ситуаціях. Поведінка розробленого алгоритму відрізнялась у залежності від кількості загублених ланок. При невеликій кількості алгоритм відновлював маршрут, незалежно від того був це маршрут пакету RREP чи іншого (рис 4.2). Але оскільки алгоритм працює у пасивному режимі і відновлює маршрути тільки, коли

через нього проходить пакет може виникнути ситуація коли маршрут майже зруйновано, тоді більшість ланок, навіть коли вони на правильних позиціях, будуть вважати, що вони випали з маршруту, до того ж відправник може ще раніше загубити зв'язок з маршрутом і намагатися його оновити. При перевірці такої ситуації алгоритм веде себе аналогічно базовому, тобто відсилає широкомовні пакети.

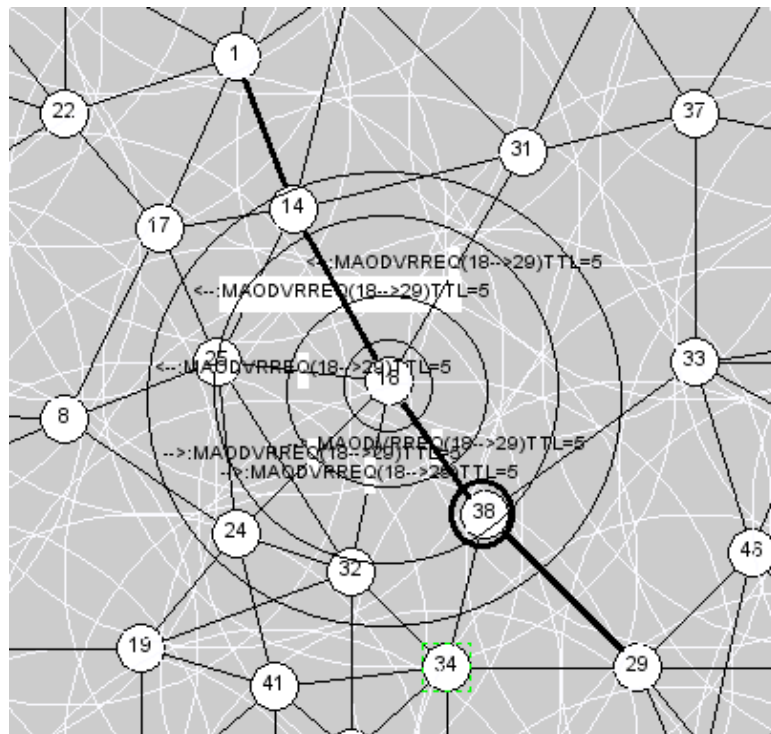


Рисунок 4.2 – Моделювання ситуації з загубленням ланок.

При переміщенні кінцевого вузла поведінка алгоритму залежить від того у яку сторону він перемістився. Якщо вузол перемістився ближче до іншого кінця маршруту останні ланки видаляються, бо оптимальний шлях вже не проходить через них і пакети направляються через передостанні ланки маршруту (рис.4.3). Слід відмітити, що усі пакети, які були у черзі видалених ланок відсилаються за неоптимальним маршрутом, бо ті ланки також отримали RREP і додали вузол у таблицю маршрутизації, а запис у таблиці транзитів видалиться за таймером, бо інших пакетів не приходить. До того ж, як видно з рисунку після корекції вузол посилає пакет RREP з оновленим номером та метрикою маршруту. Тому у наступний раз пакети від застарілої ланки ігноруватимуться.



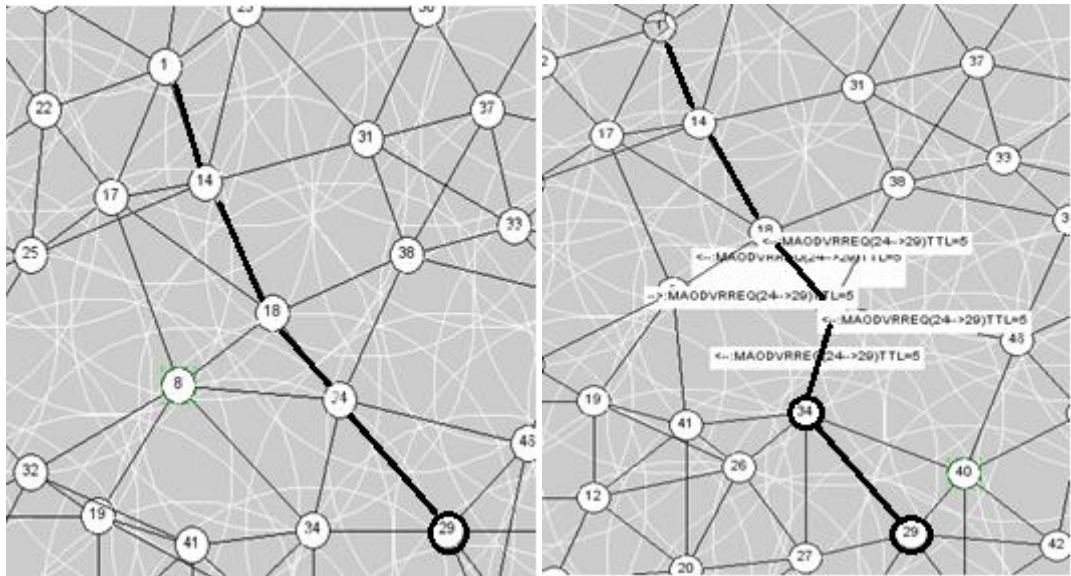


Рисунок 4.3 – Моделювання ситуації з переміщенням вузла.

При переміщенні вузла у напрямку маршруту він також залишається оптимальним, але усі метрики у ланцюгу збільшуються. Але при переміщенні вузла вбік, пакет RREP відсилається з інформацією про відхилення маршруту. Перед відсилкою пакету RREP проводиться перевірка кута, на який відхилено маршрут. Якщо точка відхилення не є точкою зламів, цей кут може бути оцінений безпосередньо алгоритмом координування вузла. Але у найгіршому випадку відсилається додатковий пакет RREQ з іншої точки маршруту і оцінка ведеться за допомогою порівняння метрик до кінцевого вузла.

При моделюванні ситуації із завадою (рис.4.4), ширококомовний пакет від ланки обходить заваду, але маршрут також не оптимальний, бо має декілька зламів, а у оптимального маршруту злам був би одним. Тому також відсилається RREP з інформацією про точки зламів.

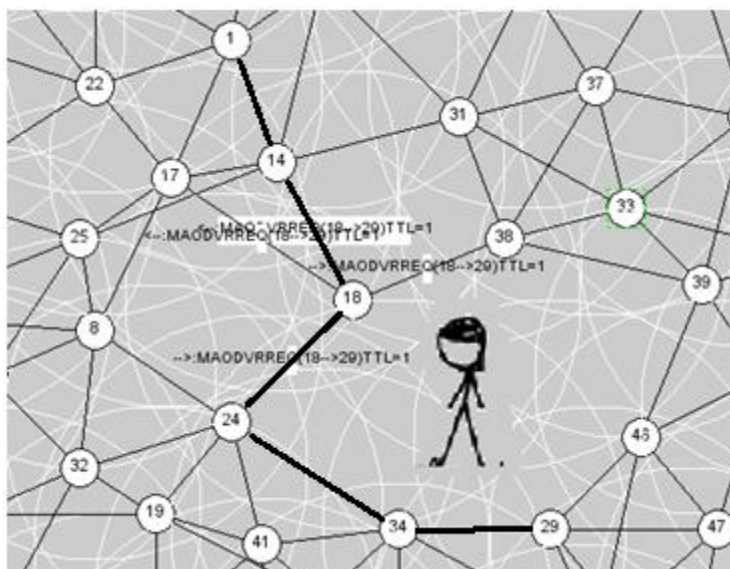


Рисунок 4.4 – Моделювання ситуації з завадою.

У активному режимі було промодельовано загублення ланок та переміщення кінцевого вузла. При загубленні ланки вузол зразу починав відновляти пошкоджену ділянку, не дожидаючись пакету від відправника, але для цього необхідно було обмінюватися повідомленнями привітання. При переміщенні кінцевого вузла, вузол сам відіслав повідомлення RREP не тільки до сусідів а і ширококомовно з малим TTL, що б сусідні ланки знали про його переміщення, тому маршрут було встановлено вдвічі швидше. Оцінка кута при точці відхилу також пройшла без додаткових пакетів, бо ширококомовні пакети RREP дійшли до останніх ділянок ланцюгу маршрутів по оптимальним шляхам і порівняння проводилось при пересилці вторинних пакетів RREP, що призначені для оновлення номеру маршруту і сповіщення про відхил. Такий підхід набагато оптимальніше модифікації Local Repair алгоритму AODV, за рахунок меншої затримки та зони ширококомовлення пакету.

Моделювання виключного ширококомовлення проводилося при відновленні маршруту. На рисунку 4.5. можна помітити, що зона мовлення пакету має форму місяця. Це пов'язано з тим що до того вже проводилася репарація маршруту неподалік і вузли проігнорували пакет для репарації того ж маршруту від правої ділянки.

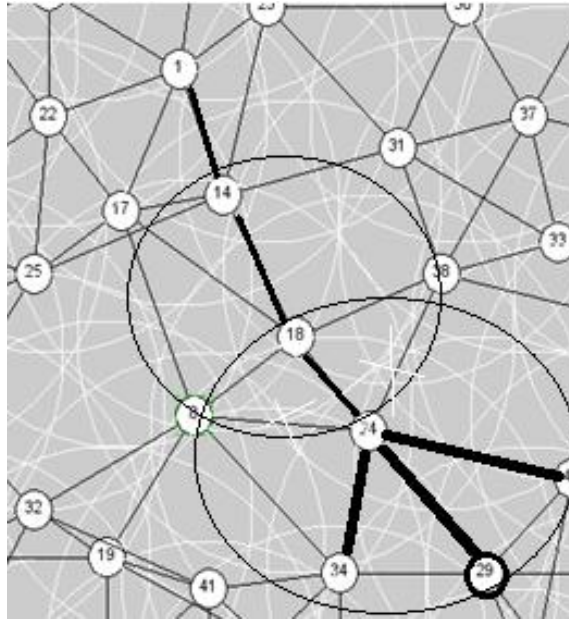


Рисунок 4.5 – Виключне широкомовлення.

Оскільки на моделі похибки координатії мінімальні, замість конусного широкомовлення можна дослідити тільки пряму координатію пакету, що показана на рисунку 4.6. Перевірка проводилася для ділянки мережі, у якій усі вузли мали по шість зв'язків. Для перевірки один із вузлів було переміщено і замість нього центром комірки став інший вузол. На рисунку видно, що пакет пройшов через нову комірку без репарації маршруту. Також можна помітити, що у обох випадках було обрано різні альтернативні маршрути, бо для координатії не важливо через який розтер пройде пакет. Їй необхідно лише що б при цьому було дотримано необхідний кут. У першому випадку компенсація відхилення обрала поворот направо, а у другому – наліво. Оскільки альтернативний маршрут обирається випадково, потік пакетів рівномірно розподіляється по обом каналам.

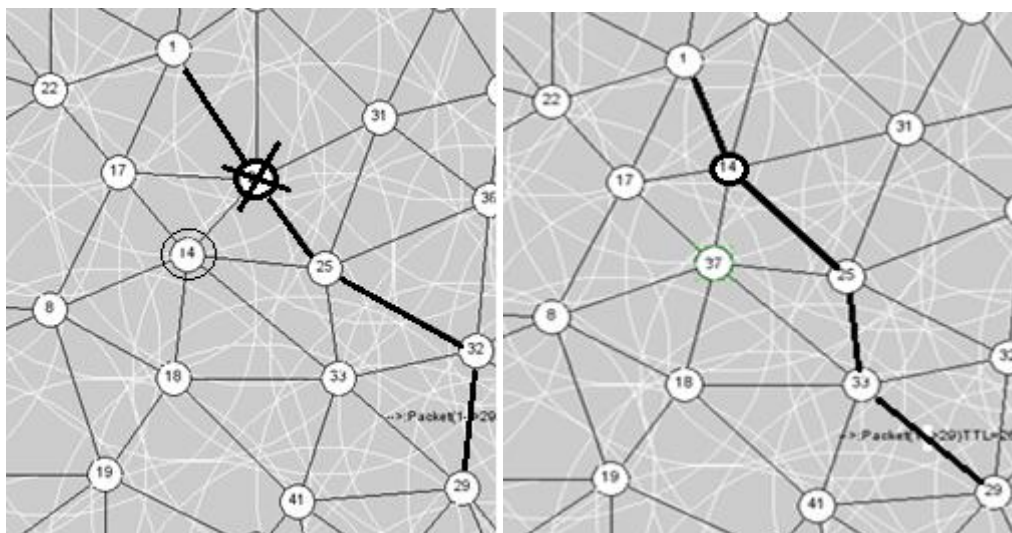


Рисунок 4.6 – Координування пакету.

#### 4.3. Отримання і аналіз характеристик.

За допомогою програмного забезпечення були отримані графіки залежності затримки пакетів та потоку службових пакетів від метрики маршруту для обох алгоритмів. Додатково для розробленого алгоритму було отримано графік залежності затримки службових пакетів від кількості видалених ланок маршруту. В усіх залежностях одиниці вимірювання або безрозмірні (кількості) або хопи (відстані та час). Моделювання основних характеристик проводилося для найгіршого випадку (пасивний режим).

Візуальне представлення залежності потоку службових пакетів від метрики маршруту представляє собою сімейство графіків для різних значень кількості зруйнованих ланок (рис 4.7). Дослідження залежності проводилося для найгіршого випадку, тобто пасивної репарації маршруту. Як видно на графіку, обидва алгоритми вели себе одного тільки при руйнуванні усіх ланок одночасно, що маловірогідно у реальній ситуації, бо трансмітери вузлів мають обмеження на швидкість переміщення вузла. У будь якій іншій ситуації виграш був тим помітніше, чим менше ланок було зруйновано у заданий період часу.

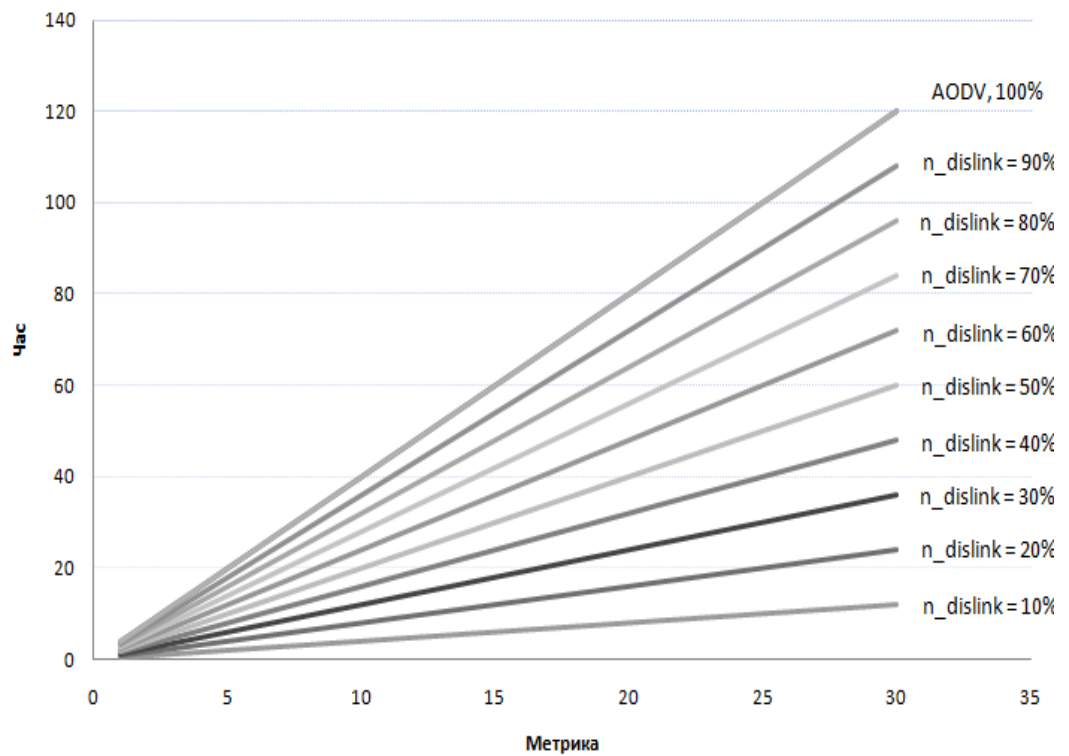


Рисунок 4.7 – Графіки залежності середнього часу затримки пакету маршруту від метрики.

Слід враховувати, що мінімальний період часу, за який алгоритм може детектувати та репарувати пошкоджений маршрут  $T_{react}$  і є періодом за який проходить  $n$  одночасних руйнацій ланок маршруту. Для AODV він складається із часу визначення пошкодження  $T_{detect}$ , часу отримання пакету помилки  $T_{rerr}$ , часу відсилки пакету запиту  $T_{rreq}$  і отримання пакету відповіді  $T_{rrep}$ :

$$T_{react} = T_{detect} + T_{rerr} + T_{rreq} + T_{rrep}$$

Якщо за час реакції буде додатково руйнуватися хоч одна ланка маршруту пакет відповіді не встигне прийти, або пакет запиту може піти не по тому шляху. При більшій частоті руйнації, AODV перестав працювати, бо маршрути постійно загублюються.

Для розробленого алгоритму час реакції складається тільки із часу, необхідного для репарації ділянки:

$$T_{react} = T_{rreq} + T_{rrep}$$

Тобто алгоритм реагує на зміни у топології набагато швидше, тому може працювати у екстремальних умовах з підвищеною мобільністю вузлів. Хоча апаратні можливості вузла можуть накладати обмеження швидкості встанови зв'язків та можливості по підтримці зв'язку з вузлом, що переміщується, швидкість реакції дуже важлива при підтримці довгих маршрутів (особливо оптимальних), бо для розробленого алгоритму ця швидкість репарації не залежить від метрики, на відмін від AODV. До того ж, мобільність (як характеристика вірогідності руйнування зв'язку) не завжди пов'язана зі швидкістю переміщення, бо вузли можуть переміщуватися не швидко, але одночасно, або коливатися на межі зон мовлення. Ще більша швидкість реакції є у алгоритму координування, бо у нього вона дорівнює швидкості встановлення зв'язків нової комірки, хоча може потребувати попереднього обміну локальними картами між сусідами, що аналогічно репарації однієї ланки з тією різницею, що усі ланки «репаруються» одночасно.

Візуальне представлення залежності потоку службових пакетів від метрики представлено у вигляді сімейства графіків для різних кількостей груп видалених ланок (рис.4.8). Поведінка алгоритму AODV була аналогічна поведінці розробленого алгоритму при максимальній кількості груп, бо коли кількість груп співпадає з кількістю ланок, це означає, що ланцюг повністю зруйновано. У інших випадках кількість широкомовних пакетів була меншою. Лінійна залежність потоку пакетів від радіусу зони мовлення спостерігається за рахунок того, що іспити проводилися для ортогональної сітки. Для мереж без обмеження кількості зв'язків потік пакетів залежить від щільності ділянки мережі, але встановивши базову гексагональну сітку можна розповсюджувати пакети тільки по ній, бо базові вузли у більшості випадків мають маршрути до додаткових.

Також моделювання проводилося без урахування виключного широкомовлення, яке у залежності від ситуації могло зменшити кількість

пакетів максимум на половину (якщо попереднє широкомовлення проводилося на значній відстані).

При використанні координування пакетів широкомовлення не потрібно, але воно може використовуватися в основному для невеликих уражених ділянок з достатньо високою щільністю вузлів.

Слід відмітити що в реальних ситуаціях можливе додаткове підвищення службового трафіку за рахунок неправильного підбору TTL пакетів.

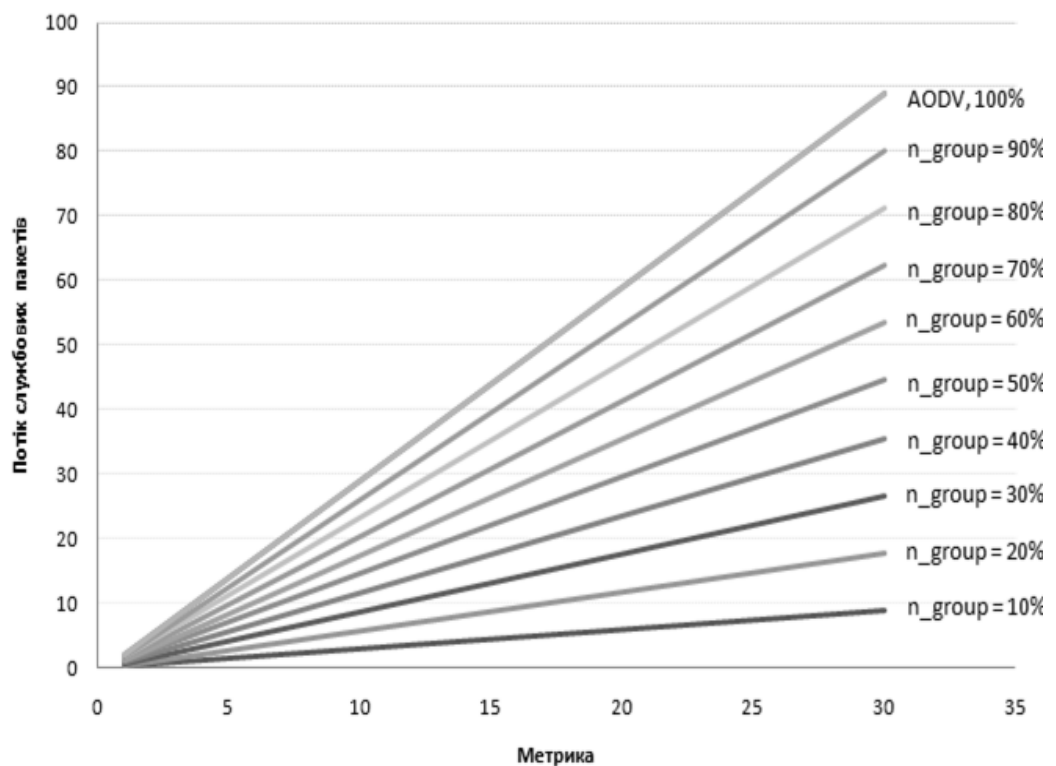


Рисунок 4.8 – Графіки залежності потоку службових пакетів від метрики.

Додатково було отримано графіки залежності затримок встановлення маршруту від кількості видалених ланок для пасивного та активного режиму (рис.4.9). Графіки були отримані при фіксованій вірогідності групування пошкоджень, бо вона незалежна від вірогідності видалення ланок.

Із графіку видно, що для пасивному режиму завжди необхідна затримка, пропорційна кількості видалених ланок у маршруті. Якщо вимірювати затримку у хопх (як на графіку), то вона буде вдвічі більша



кількості видалених вузлів. Для активного режиму затримка пропорційна розміру групи, який завжди менше числа видалених ланок. У гіршому випадку коефіцієнт відношення між затримками у активному та пасивному режимах дорівнює коефіцієнту відношення між середнім розміром пошкоджених груп ланок і кількістю самих ланок. При двосторонній репарації затримка може бути у два рази меншою.

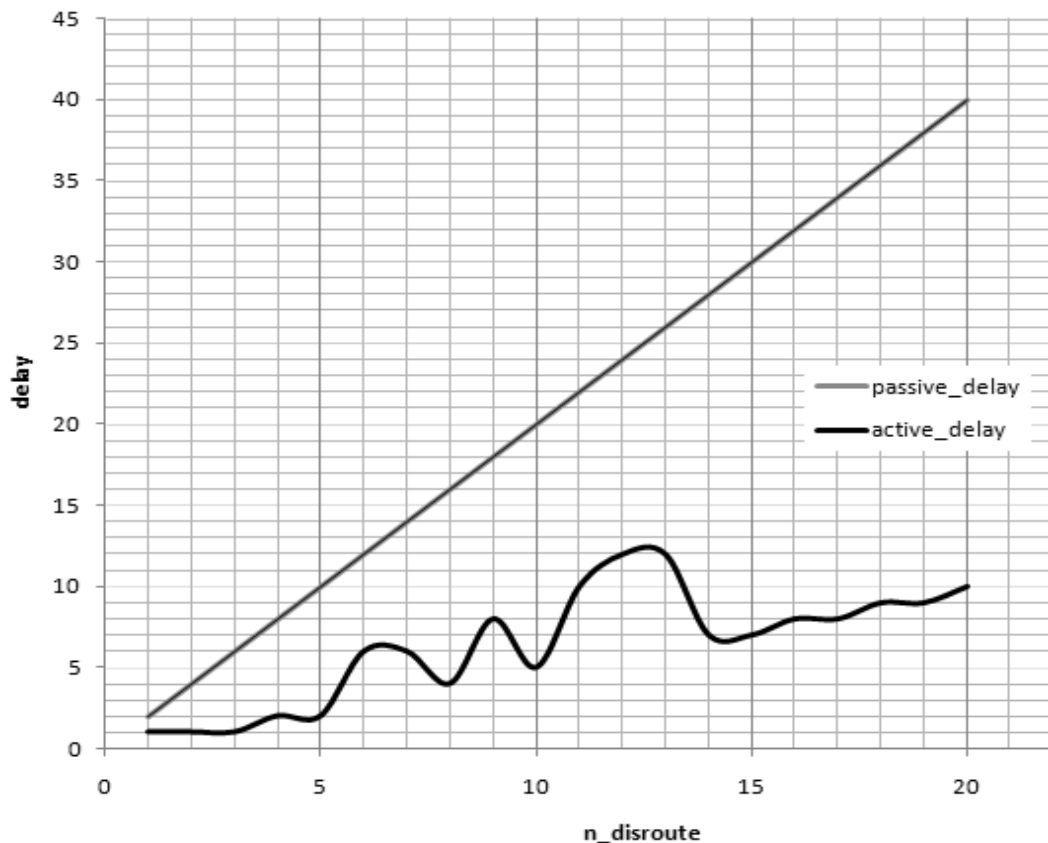


Рисунок 4.9 – Порівняння активного та пасивного режиму

Слід враховувати, що реальний вплив на затримку пакетів залежить від співвідношення між затримкою при передачі службового пакету і пакету в один байт, тобто швидкістю передачі даних у мережі. Чим вона більша тим менший вплив на затримку пакетів мають затримки пошуку маршрутів. До того ж, якщо використовується контроль якості сервісів (QoS) у службових пакетах маршрутизації може бути максимальний пріоритет. З іншого боку їх пріоритет може впасти при великому потоці таких пакетів, що значно збільшить затримки при передачі. До того ж вузол при ретрансляції пакету вносить додаткову затримку, що пов'язана не з перебуванням пакету у



чергах, а з обробкою і аналізом пакетів і залежить від обчислювальних можливостей вузла.

Також при значних відстанях стає помітною затримка, пов'язана із швидкістю світла у вакуумі. При несприятливому середовищі передачі радіосигналу може змінитися і сама швидкість світла за рахунок «ретрансляції» фотонів між атомами. У такому випадку знадобиться координування пакетів по шляхам без оптимальних зламів, яке намагається передати пакет за шляхом, що близький до фізичної прямої.

## Висновки

Моделювання показало що розроблений алгоритм більш адаптований до топології Mesh ніж базовий, за рахунок більшої активності при підтримці маршрутів і використання особливостей топології та алгоритмів встановлення зв'язків. При моделюванні були виявлені наступні переваги алгоритму:

- менша затримка пакетів у чергах
- менша кількість загублених пакетів
- менший службовий потік пакетів
- можливість зменшення фізичної затримки при передачі сигналу
- можливість швидкої корекції маршруту при переміщенні вузла-цілі або вузла призначення
- більша швидкість реакції

У щільних ділянках мережі, які можна використовувати для координації пакетів додаються наступні можливості:

- немає необхідності у підтримці маршрутів
- можливість автоматичного розподілення трафіку між альтернативними шляхами
- можливість зменшення фізичних затримок при передачі даних
- відсутність широкомовних пакетів запиту при підтримці

- можливість корекції маршрутів із зламами

До недоліків можна віднести більш активний трафік при установці зв'язку і незначні сплески трафіку при одночасній репарації. В цілому, розроблений алгоритм можна використовувати у тих же мережах, що й AODV, як більш оптимальний і пристосований, та у мережах з підвищеною мобільністю або чутливістю до затримок пакетів.