Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра радіотехніки та телекомунікацій

# МЕТОДИЧНІ ВКАЗІВКИ

До лабораторних робіт з дисципліни
«Широкосмугові технології телекомунікацій»
Для студентів спеціальності 172 «Телекомунікації та радіотехніка»
всіх форм навчання

Частина I

2019

Методичні вказівки до лабораторних робіт з дисципліни «Широкосмугові технології телекомунікацій» для студентів спеціальності 172 «Телекомунікації та радіотехніка» всіх форм навчання. Частина I / Укл.: В.С. Кабак, Г.Ф. Вишник. – Запоріжжя: НУ «Запорізька політехніка», 2019. – 56 с.

Укладачі: В.С.Кабак, доцент, к.т.н.,
Г.Ф. Вишник, зав. лаб.

Рецензент: С.В. Морщавка, доцент, к.т.н.

Відповідальний.
за випуск: Г.Ф. Вишник, зав. лаб.

Затверджено:
на засіданні кафедри
радіотехніки та телекомунікацій
Протокол № 1 від 11.09.2019 р.

Рекомендовано до видання НМК
факультету радіоелектроніки та
телекомунікацій
Протокол № 2 від 19.09.2019 р.

# ЗМІСТ

1 МОДЕЛЮВАННЯ ПРОТОКОЛУ МНОЖИННОГО ДОСТУПУ ALOHA ……... 5
1.1 Мета роботи ..... 5
1.2 Завдання на підготовку до виконання лабораторної роботи..... 5
1.2.1 Структурна схема системи з пакетною передачею даних..... 5
1.2.2 Канал зв'язку ..... 6
1.2.3 Генерація пакетів ..... 8
1.2.4 Колізія ..... 8
1.2.5 Трафік..... 9
1.2.6 Пропускна здатність ..... 11
1.2.7 Середній час затримки передачі ..... 11
1.2.8 Оцінка якості протоколів доступу..... 12
1.3 Базовий алгоритм програмного комплекесу..... 13
1.3.1 Опис головної програми..... 15
1.4 Опис підпрограм, які необхідні для функціонування головної програми..... ……..25
1.4.1 Підпрограма розташування терміналів користувачів..... 25
1.4.2 Підпрограма розрахунку відстані між точкою доступу і терміналом користувача ..... 26
1.4.3 Підпрограма виводу результатів теоретичного аналізу ..... 27
1.4.4 Підпрограма розташування терміналів користувачів..... 27
1.5 МОДЕЛЮВАННЯ ЧИСТОЇ СХЕМИ ALOHA ..... 27
1.5.1 Підпрограма моделювання протоколу ALOHA ..... 29
1.6 ПОРЯДОК ПРОВЕДЕННЯ ЛАБОРАТОРНОЇ РОБОТИ ..... 32
1.6.1 Моделювання протоколу ALOHA для провідної мережі.... 32
1.6.2 Моделювання протоколу ALOHA для безпровідної мережі ..... 36
1.7 Зміст звіту ..... 36
1.8 Контрольні запитання ..... 37
2 МОДЕЛЮВАННЯ ПРОТОКОЛУ МНОЖИННОГО ДОСТУПУ S-ALOHA ..... 38
2.1 Мета роботи ..... 38
2.2 Завдання на підготовку до виконання лабораторної роботи..... 38
2.2.1 Обґрунтування базового алгоритму програмного комплексу ..... 38

2.2.2 Підпрограма моделювання протоколу S-ALOHA ..... 40
2.3 Порядок проведення лабораторної роботи ..... 40
2.3.1 Моделювання протоколу S-ALOHA
для провідної мережі ..... 41
2.3.1 Моделювання протоколу S-ALOHA
для безпровідної мережі ..... 42
2.4 Зміст звіту ..... 42
2.5 Контрольні запитання ..... 43
ПЕРЕЛІК ПОСИЛАНЬ..... 44
ДОДАТОК А Лістинги програм для моделювання протоколу ALOHA…….45
ДОДАТОК Б Лістинг підпрограми saloha для моделювання протоколу S- ALOHA ……..55

# 1 ЛАБОРАТОРНА РОБОТА №1 МОДЕЛЮВАННЯ ПРОТОКОЛУ МНОЖИННОГО ДОСТУПУ ALOHA

## 1.1 Мета роботи

1.1.1 Ознайомитися з принципами побудови протоколів множинного доступу у безпровідних мережах.

1.1.2 За допомогою ЕОМ провести симуляцію роботи конкурентного протоколу доступу з повторенням ALOHA.

1.1.3 Отримати показники ефективності протоколу ALOHA.

## 1.2 Завдання на підготовку до виконання лабораторної роботи

### 1.2.1 Структурна схема системи з пакетною передачею даних

По-перше, визначимося з основними загальними припущеннями, на яких ґрунтується комп’ютерне моделювання протоколів множинного доступу.

Узагальнена структурна схема системи з пакетною передачею даних і використанням точки доступу, що пропонується для моделювання, зображена на рис.1.1 [1].

Припустимо, що характеристики терміналів усіх користувачів однакові. До того ж, будемо вважати, що кожен термінал має буфер. Коли відбувається генерація відповідним користувачем пакета, то він зберігається у буфері. Потім пакети, що зберігалися у буфері передаються відповідно до надходження пакетів у буфер. Тобто, буфер працює за алгоритмом FIFO (першим прийшов – перший вийшов).

Сам буфер може підрозділятися на два типи: буфер з нескінченою пам’яттю і буфер з кінцевою пам’яттю. Якщо використовується буфер з кінцевою пам’яттю і пам’ять заповнена, то пакет сформований у даного користувача втрачається. Такий варіант трактується як блокування виклику, але він відрізняється від спотворень пакета під час передачі до точки доступу.

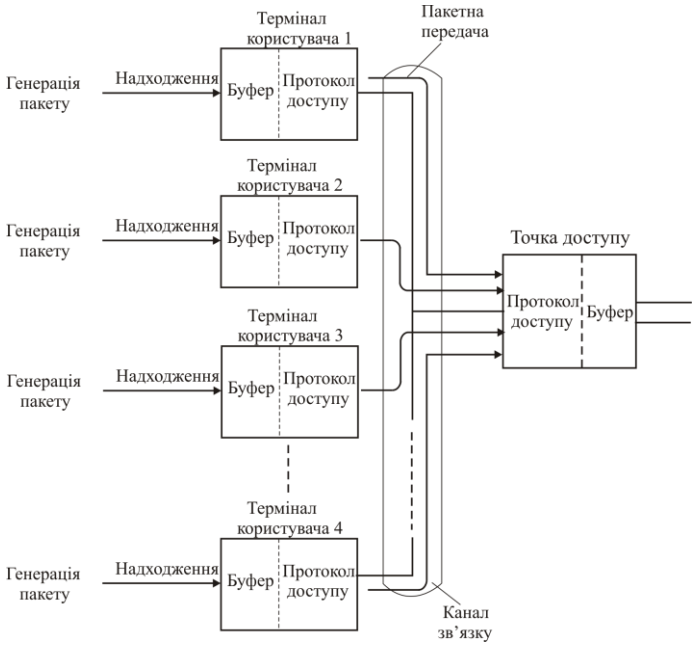



Рисунок 1.1 – Узагальнена схема телекомунікаційної системи з пакетною передачею

Окрім того, якщо кількість терміналів нескінчена, то таку модель називають моделлю з нескінченою кількістю джерел-терміналів.

З іншого боку, модель для якої кількість користувачів обмежена, називають моделлю з кінцевою кількістю джерел-терміналів.

### 1.2.2 Канал зв'язку

Суттєва різниця між провідними і безпровідними мережами полягає саму у каналі зв'язку. Визначимо основні ознаки і відмінності провідного і безпровідного каналів зв'язку.

Для провідних мереж ніякого спотворення сигналу у спільному середовищі (каналі зв'язку) не відбувається і потужність сигналів, що

надходять до точки доступу від терміналів користувачів не змінюється. Саме з цього фундаментального положення здійснюється оцінка роботи протоколу доступу для провідної мережі.

Для безпровідних систем зв'язку характеристики середовища (каналу зв'язку) є варіабельними у часі.

По-перше необхідно враховувати втрати розповсюдження, які залежать від відстані між терміналом і точкою доступу, а також враховувати ефект затінення, який виникає через наявність перешкод на шляху розповсюдження сигналу.

Для втрат розповсюдження пропонується розглядати монотонне зменшення прийнятого у точці доступу сигналу під час збільшення відстані між терміналом і точкою доступу. Математично, звичайно, втрати розповсюдження, тобто зменшення потужності сигналу, подають через коефіцієнт пропорційності $\gamma^{-\alpha}$, де $\gamma$ – медіанне значення прийнятого сигналу (для великомасштабних завмирань) і $\alpha$ – коефіцієнт затухання. Як правило, в залежності від місцевості значення $\alpha$ знаходиться у межах від 2 до 5 [2,3].

Окрім того, сигнал, що передається, поширюється в умовах багатопроменевого розповсюдження радіохвиль і надходить до точки прийому за різними шляхами. При цьому під час розповсюдження він підлягає впливу від таких фізичних ефектів як дифракція, перевідбиття і розсіювання. В результаті прийнятий сигнал у точці прийому набуває флуктуацій за амплітудою, фазою та кутом надходження. Цей ефект від багатопроменевого розповсюдження отримав назву федінга. Ці завмирання сигналу трактують як дрібномасштабні флуктуації.

Якщо провести усереднення рівня сигналу на інтервалі декілька десятків довжини хвилі, то отримане значення трактують як медіанне для дрібномасштабних завмирань. Причому, це середнє значення також підлягає додатковим флуктуаціям через наявність перешкод, будівель, пагорбів і такі флуктуації у дрібномасштабному сенсі отримали назву затінення. Математично затінення описується логарифмічно-нормальним законом зі стандартним значенням дисперсії 6…7 дБ [2,3].

### 1.2.3 Генерація пакетів

Припускаємо, що генерація пакетів кожним терміналом користувача відбувається випадково і незалежно. Вважається, що за таких умов процес генерації пакетів підлягає Пуасонівському розподілу [1, 2]. Відповідно можна відзначити основні вимоги під час генерації пакетів:

– незалежність, тобто генерація пакету повністю не залежить від попереднього процесу генерації;

– постійність, під якою розуміють однакову ймовірність з'явлення пакету для кожного часового слоту при моделюванні;

– рідкість з'явлення пакетів, яку трактують для малого часового інтервалу, а саме, вважатимемо, що ймовірність генерації більш двох пакетів за дуже малий інтервал часу дуже мала.

Крім того, якщо кількість пакетів, що згенерована, підлягає Пуасонівському розподілу, то момент часу, для якого здійснюється генерація пакету також описується законом Пуассона.

### 1.2.4 Колізія

Під терміном колізія, будемо визначати ситуацію, яка виникає, коли два або більше пакетів одночасно передаються у спільному середовищі (рис.1.2), де затемнені ділянки відповідають конфлікту декілька пакетів, що одночасно передаються.

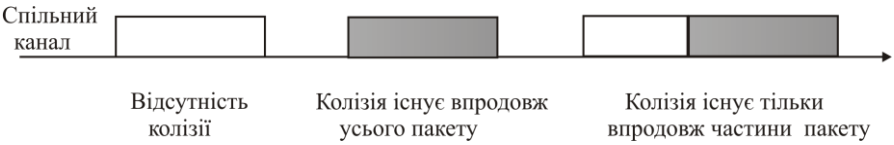



Рисунок 1.2 – Можливі варіанти колізій, які виникають під час пакетної передачі

Особливості обробки пакетів у випадку такого конфлікту для провідних і безпровідних систем наступні:

– для провідної мережі усі пакети, що формуються у спільному середовищі під час колізії спотворюються, оскільки рівень сигналів для усіх пакетів однаковий і, відповідно, пакет вважається втраченим

– тобто для системи зв’язку передача пакету не здійснилася. Якщо ж колізія не виникає, то пакет вважається таким, як успішно доставлений до отримувана;

– для безпровідної мережі потужність прийнятого сигналу від кожного користувача залежить від розташування термінала і умов розповсюдження. Тому, якщо під час колізії у каналі розповсюджується декілька пакетів, то цілком ймовірна ситуація, коли пакет від найближчого терміналу, тобто, з максимальною потужністю у точці прийому, може інколи вважатися переданим, оскільки для нього спотворення мінімальні. Звичайно таку ситуацію з технічної точки зору трактують як **ефект захоплення**.

З іншого боку, навіть якщо колізія не виникає, то пакет також може вважатися втраченим, якщо потужність прийнятого сигналу у точці доступу менша за деякий пороговий рівень, при якому ще можлива достовірна передача інформації.

Для реальних комунікаційних мереж саме точка доступу вирішує вважати пакет від конкретного термінала успішно доставленим або ні, і результати рішення передаються до терміналу-джерела. Крім того, пакети, які не вважаються успішно доставленими, передаються повторно через деякий часовий інтервал.

### 1.2.5 Трафік

У програмному комплексі під терміном трафік будемо розглядати повну кількість заново згенерованих пакетів, а також пакетів, що передавалися повторно за визначений інтервал часу, тобто повний інформаційний обсяг.

Припустимо, що для роботи деякої системи необхідна визначена середня частота успішної передачі повідомлень (пакетів) до адресату, яку позначимо як $\lambda_{усп}$. Через виникнення конфліктних ситуацій деякі з цих пакетів не будуть отримані корисним користувачем або будуть відхилені. Середню частоту виникнення відхилених або втрачених пакетів позначимо як $\lambda_{втр}$. Відповідно загальну (результуючу) частоту надходження пакетів в такій системі можна визначити як суму частоти успішного надходження і частоти відхилення пакетів [2]:

$$\lambda_{\Sigma}=\lambda_{усп}+\lambda_{втр}. \qquad (1.1)$$

Припустимо, що розмір одного повідомлення або пакету рівняється b біт. Тоді повний інформаційний обмін або повний трафік у каналі можна визначити як:

$$G' = b\lambda_{\Sigma} \text{ [ббіт/с.} \tag{1.2}$$

Під час порівняльного аналізу різних протоколів доступу доцільно проводити порівняння з точки зору нормованого трафіку, де нормалізація проводиться відносно швидкості передачі інформації.

Так, якщо максимальна швидкість передачі інформації у каналі рівняється R біт/с, то нормований повний трафік визначиться як:

$$G = \frac{b\lambda_{\Sigma}}{R} \ , \tag{1.3}$$

і вже виступає безрозмірною величиною. Відзначимо, що нормований повний трафік визначає повний інформаційний обмін як частину $(0 \le G \le \infty)$ ємності каналу. Відповідно значення нормованого трафіку може бути більшим за 1. Якщо не відбувається генерації жодного пакету, то, очевидно, що G=0.

Нормований трафік можна також визначити через тривалість одного пакету τ, що передається. Так, якщо розмір одного пакету складає b біт і ці біти передаються зі швидкістю R, то тривалість одного пакту рівняється:

$$\tau = \frac{b}{R} \text{ [ссекунд/пкет]} . \tag{1.4}$$

Тоді з (1.3) для нормованого трафіка можна отримати:

$$G = \lambda_{\Sigma}\tau \ . \tag{1.5}$$

### 1.2.6 Пропускна здатність

Під пропускною здатністю у програмі приймається повна кількість пакетів, що успішно доставлені до точки доступу за визначений час. Відповідно, якщо розмір пакету рівняється b біт, то середній об'єм успішно переданих даних, а іншими словами, пропускна здатність каналу складе [2]:

$$\rho' = b\lambda_{усп} \text{ [ббіт/с.} \tag{1.6}$$

Подальший аналіз також а будемо проводити у термінах нормованої пропускної здатності, яку визначимо як:

$$\rho = \frac{b\lambda_{усп}}{R} \ . \tag{1.7}$$

На відміну від нормованого трафіку, нормована пропускна здатність не може перевищувати 1, тобто ρ може приймати значення від 0 до 1 $(0 \le \rho \le 1)$.

Якщо ніякої генерації пакетів не відбулося, або усі передані пакети втрачені через колізії, то ρ набуває мінімального – нульового значення. Якщо ж усі пакети успішно доставлені до точки доступу, то ρ=1.

З виразу (1.7) можна визначити нормовану пропускну здатність через час передачі одного пакету:

$$\rho = \lambda_{усп}\tau \ . \tag{1.8}$$

### 1.2.7 Середній час затримки передачі

Інтервал часу впродовж якого відбувається генерація пакету у терміналі користувача, передача його до точки доступу і прийом переданого пакету у точці доступу будемо трактувати як середню затримку передачі. Середня затримка передачі залежить від довжини пакету. Тоді нормалізовану до довжини пакету τ середню затримку позначимо як D. В принципі, для реальних телекомунікаційних

систем, середній час затримки пакетів залежить від поточного часу генерації пакету і передачі його терміналом користувача, відстані між точкою доступу і терміналом користувача і часу обробки сигналу у точці доступу.

### 1.2.8 Оцінка якості протоколів доступу

Якість протоколів доступу звичайно оцінюють за трьома введеними фундаментальними показниками – нормованим інформаційним обсягом або нормованим трафіком, нормованою пропускною здатністю і нормованим середнім часом затримки передачі.

Для ідеального протоколу доступу для пропускної здатності повинно виконуватися наступне [1]:

$$\rho = \begin{cases} G & (G < 1) \\ 1 & (G \geq 1) \end{cases} . \qquad (1.9)$$

Але, як показано на рис.1.3, для реальних систем нормована пропускна здатність зростає до деякого порогового значення G під час збільшення нормованого трафіку, а потім пропускна здатність починає зменшуватися під час збільшення G.

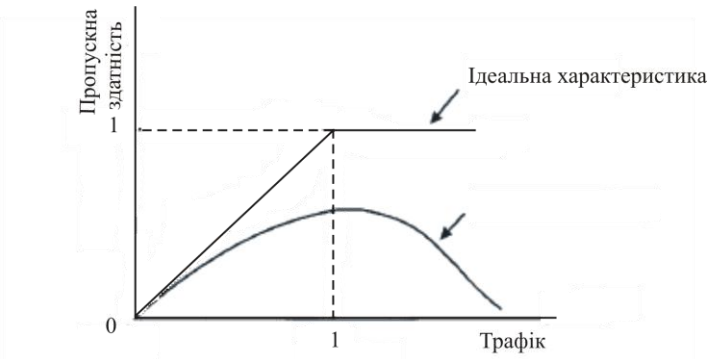



Рисунок 1.3 – Реальна і ідеальна залежності нормованої пропускної здатності від нормалізованого трафіка

Залежності для реального середнього часу затримки і ідеалізованої кривої наведені на рис.1.4. Ідеалізована характеристика середнього часу затримки отримана за припущенням що відбувається ідеальне планування передачі усіх пакетів і ніякого часу на планування не витрачається [1].

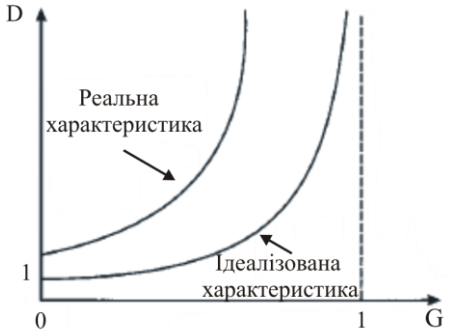



Рисунок 1.4 – Реальна і ідеальна залежності нормованого середнього часу затримки передачі від нормалізованого трафіка

Якщо нормований трафік стає більшим за одиницю, середній час затримки D суттєво зростає і устромляється до нескінченності.

### 1.3 Базовий алгоритм програмного комплексу

Блок-схема алгоритму програмного комплексу наведена на рис.1.5 [1].

Основною метою моделювання є визначення пропускної здатності і затримки передачі з метою порівняння різних протоколів доступу до каналу.

Як видно зі схеми, спочатку алгоритмом передбачається виконання двох незалежних задач: по-перше, це реалізація трафіка, що підлягає пуасонівському розподілу, і, по-друге, визначення геометричного місця розташування терміналів користувачів і точки доступу у зоні обслуговування.

Далі необхідно розглянути умови, які відповідають каналам зв’язку провідних і безпровідних мереж. При моделюванні безпровідної мережі припускаємо, що втрати розповсюдження і затінення мають постійне значення.

Рисунок 1.5 – Базовий алгоритм моделювання

Процес моделювання закінчується після того, як кількість успішно переданих пакетів досягне необхідного заданого значення.

Застосовуючи декілька підпрограм можливо за базовим алгоритмом дослідити різні протоколи доступу з відповідними додатковими змінами.

### 1.3.1 Опис головної програми

Повний лістинг головної програми, а також необхідних підпрограм, наведено у додатку А. Розглянемо основні процедури, які необхідно виконати у головній програмі. Спочатку охарактеризуємо усі змінні, які будемо використовувати у програмному комплексі. (таблиця 1.1).

Таблиця 1.1 – Параметри і змінні, що використовуються у головній програмі

| Позначення змінної | Фізичний сенс |
|---|---|
| **brate** | Бітова швидкість (біт/секунда) |
| **srate** | Символьна швидкість (символ/секунда) |
| **plen** | Довжина пакету у символах |
| **Ttime** | Час передачі пакету (секунда) |
| **Dtime** | Нормалізована затримка розповсюдження |
| **alfa** | Коефіцієнт затухання |
| **sigma** | Стандартне значення девіації для затінення |
| **capture** | Наявність ефекту захоплення (0 – присутній, 1 - відсутній) |
| **r** | Радіус зони обслуговування |
| **bxy** | Розташування точки доступу |
| **mxy** | Розташування терміналів користувачів |
| **Mnum** | Кількість терміналів користувачів |
| **tcn** | Коефіцієнт захоплення (децибели) |

Продовження таблиці 1.1

| Позначення змінної | Фізичний сенс |
|---|---|
| **mcn** | Відношення сигнал/шум у точці доступу для терміналу, що знаходиться на границі зони (децибел) |
| **Mplen** | Довжина пакета у символах |
| **Mstate** | Стан терміналу користувача |
| **mgtime** | Час генерації пакету у терміналі користувача (секунда) |
| **mtime** | Час на зміну стану користувача (секунда) |
| **Mstime** | Час передачі пакету від терміналу |
| **G** | Трафік |
| **Tint** | Очікуване значення інтервалу генерації пакету (секунда) |
| **Rint** | Очікуване значення інтервалу повторної передачі (секунда) |
| **Spnum** | Кількість успішно переданих пакетів |
| **spend** | Кількість пакетів для закінчення моделювання |
| **Splen** | Кількість інформації в успішно переданих пакетах (символ) |
| **Tplen** | Кількість інформації для якої здійснювалася спроба передачі (символ) |
| **Wtime** | Час затримки передачі (секунда) |

Структурно головну програму можна розділити на декілька функціональних блоків [1].

У першому блоці програми усі змінні, що використовуються як у головній програмі, так і в підпрограмах, визначаються як глобальні змінні. Це робиться з метою скорочення пам’яті, що необхідна для моделювання, з відповідним збільшенням швидкості виконання.

Як глобальні змінні обираємо такі параметри системи зв’язку:

– по-перше, вводимо змінні, що характеризують стан користувачів **STANDBY**-ОЧІКУВАННЯ, **TRANSMIT**– ПЕРЕДАЧА, **COLLISION** – КОЛІЗІЯ, **PERMIT** – ДОЗВІЛ НА ПЕРЕДАЧУ;

– по-друге визначаються, як глобальні змінні, параметри системи зв'язку, що відповідають технічному завданню: символьна швидкість – **Srate**, довжина пакету – **Plen**, час передачі пакету - **Ttime**, час затримки розповсюдження – **Dtime**, кількість терміналів користувачів – **Mnum**, час передачі пакету – **Mstime**, стан терміналу користувача – **Mstate**;

– по-третє, задаються очікуване значення часу генерації нового пакету – **Tint** і очікуване значення часу повторної передачі пакету - **Rint**;

– по-четверте, як глобальні змінні, розглядаються параметри, які характеризують якість протоколу множинного доступу: кількість успішно переданих пакетів – **Spnum,** кількість інформації у символах в успішно переданих пакетах – **Splen,** загальна кількість інформації у символах для якої здійснювалася спроба передачі – **Tplen,** середній час затримки передачі **– Wtime**.

Відповідний блок програми у термінах MATLAB набуває такого вигляду:

```
global STANDBY  TRANSMIT  COLLISION  PERMIT
global Srate  Plen  Ttime  Dtime
global  Mnum  Mplen Mstime  Mstate
global  Tint  Rint
global  Spnum  Splen  Tplen  Wtime
```

Визначаємо універсальні константи, які визначають процеси в мережі:

```
STANDBY    =0;
TRANSMIT   =1;
COLLISION  =2;
PERMIT     =3;
```

Для кожного протоколу доступу визначаємо своє ім'я підпрограми і в третьому блоці вводяться імена підпрограм для протоколів, що досліджуються. При цьому використовуємо для вибору підпрограм масив, оскільки за такою побудовою з використанням функції Matlab feval можна легко додавати нові протоколи без ускладнення головної програми:

```
Protocol ={'paloha' ...% Pure Aloha      : pno=1
```

**’ saloha’ ...% Slotted Aloha : pno=2**
**’ npcsma’ ...% Np-CSMA : pno=3**
**’ snpisma’ ...% Np-CSMA : pno=4**.

Введення оператора feval дозволяє використовувати головну програму для дослідження різних протоколів, що досліджуються у наступних лабораторних роботах.

Зокрема, при моделюванні протоколу чистої схеми ALOHA значення параметру **pno** задається рівним 1, для протоколу S-ALOHA pno=2 і т.д., тобто, змінна **pno** дозволяє вибрати необхідний протокол для моделювання з масиву відповідних протоколів.

В четвертому блоці визначаються параметри для моделювання згідно з технічним завданням. Так, змінну **tcn** визначаємо як порогове значення прийнятого сигналу у точці доступу, яке гарантує, що ніякого спотворення або помилки під час прийому не виникає. Тобто, **tcn – це мінімально можливе відношення сигнал/шум, яке гарантує успішну передачі пакета.** Приймаємо його рівним 10 дБ. В програмі трактуємо цю величину як коефіцієнт захоплення.

Змінну **mcn** вводимо як передану потужність сигналу від терміналу і яку задамо у програмі через відношення сигнал/шум у точці доступу, за умови розташування терміналу користувача на границі зони обслуговування і коли в каналі проявляються втрати розповсюдження. Приймаємо **mcn**=30 дБ.

Якщо ефект захоплення не розглядається (**capture=0**), то моделюється ідеальний канал провідної мережі. Якщо ж ефект захоплення включається до моделі (**capture=1**), то моделювання відповідає безпровідній мережі.

Загальну кількість пакетів, яку необхідно донести до точки доступу (тобто розглядати як успішно передані) і по завершенні якої процес моделювання можна вважати закінченим, позначимо як **spend.** Приймаємо це значення рівним 1000. Кількість користувачів в мережі задана рівною 100.

Файл в якому будуть зберігатися результати моделювання позначимо як **outfile.**

Відповідний блок завдання параметрів на моделювання програми набуває такого вигляду:

```
brate  =512e3;
Srate  =256e3;
Plen=128;
```

```
Dtime= 0.01;
alfa  =3;
sigma  =6;
r=100 ;
bxy=(0,0,5);
tcn=10;
Mnum  =100;
mcn=30;
pno=1;
capture=0;
spend=1000;
outfile=  'test.dat';.
```

За заданими у четвертому блоці параметрами, у наступному – п'ятому блоці, розраховуються час передачі пакета як відношення тривалості пакету (у символах) до символьної швидкості:

**Ttime=Plen/Srate;**.

Відношення сигнал/шум у точці доступу визначимо на підставі відомого виразу [1, 2, 3]:

$$P_{прм} = P_{пер} \cdot R^{-\alpha}.$$

Для розрахунку відношення сигнал/шум припускаємо, що усі термінали користувачів розташовані на висоті 0 м, точка доступу піднесена на висоту 5 м (змінна bxy(3)), а рівень вхідних шумів приймачів термінала користувача і точки доступу однаковий:

**Npow=10^(mcn/10)*sqrt(r^2+bxy(3) ^2) ^alfa;**.

Усі параметри, що задані у блоці завдання параметрів зберігаємо у файлі ідентифікаторі fid :

```
 fid= fopen(outfile,'w');
fprinf (fid,'Protocol            =%d\n',pno );
fprinf(fid,'Capture              =%d\n',capture );
fprinf(fid,'Normalised_delay_time   =%f\n',Dtime );
fprinf(fid,'Bit_rate (bps)               =%d\n',brate);
fprinf(fid,'Symbol_rate (sps)            =%d\n',Srate );
fprinf(fid,'Length_of_Packet (sbl)          =%d\n',Plen);
fprinf(fid,'Number_of_mobile                =%d\n',Mnum );
fprinf(fid,'Transmission_power (C/N)     =%f\n',mcn );
fprinf(fid,'Capture_ratio (dB)               =%f\n',tcn );
fprinf(fid,'Number_of_Packet                 =%d\n',spend );.
```

У сьомому блоці визначаємо розташування терміналів користувачів у зоні обслуговування з радіусом r. Реалізація розташування виконується за допомогою підпрограми **position.m**.

Припускаємо, що розташування усіх терміналів користувачів у зоні обслуговування за декартовою системою координат є випадковим. Таке припущення дозволяє задати затінення між терміналом користувача і точкою доступу випадковим значенням з нормальним законом розподілу:

```
mxy=position (r,Mnum,0);
randn('state',sum(100*clock));
mrnd=rand(1,Mnum);
```

У наступному блоці задається нормалізований трафік. Трафік G, що пропонується реалізувати під час моделювання, задаємо циклом за допомогою оператора **for**. Значення G повинно бути меншим за кількість терміналів користувачів. Якщо трафік перевищить кількість користувачів, то процес моделювання закінчується:

```
G=[0.1:0.1:1,1.2:0.2:2]
if G>=Mnum
    break
  end.
```

Даля розраховується очікуваний час генерації пакету, виходячи з кількості терміналів користувачів.

Якщо припустити, що генерація пакетів відбувається за розподілом Пуассона, то ймовірність генерації n пакетів за час t може бути подана як [1,2]:

$$P(t)=\frac{e^{-\lambda t}(\lambda t)^{n}}{n!}. \qquad (1.10)$$

Ймовірність того, що жодного пакета не буде згенеровано впродовж часового інтервалу від 0 до t визначиться як:

$$P(t)=e^{-\lambda t}. \qquad (1.11)$$

Відповідно, можна визначити ймовірність генерації першого пакету після часового інтервалу t як:

$$p(t)=1-e^{-\lambda t} \qquad (1.12)$$

Тоді, у варіанті застосування такого розподілу для протоколу доступу, як поточний час t можна вважати час передачі пакету $T_{time}$, а параметр λ визначається очікуваним часом генерації пакету, $\lambda = 1/T_{int}$. Відповідно (1.12) можна подати у виді:

$$p(t) = 1 - e^{-\frac{Ttime}{Tint}} \tag{1.13}$$

З іншого боку ймовірність передачі пакету через інтервал t можна визначити через повний нормалізований трафік і кількість терміналів користувачів як

$$p(t) = \frac{G}{M_{num}}, \tag{1.14}$$

тобто можна записати рівність:

$$p(t) = 1 - e^{-\frac{Ttime}{Tint}} = \frac{G}{M_{num}} \tag{1.15}$$

З отриманого рівняння визначається очікуваний час генерації пакету:

$$Tint = -\frac{Ttime}{\ln\left(1 - \frac{G}{M_{num}}\right)}. \tag{1.16}$$

Для моделювання необхідно визначити час генерації пакету для кожного користувача. Згідно з (1.13) можна записати:

$$x = 1 - e^{-\frac{t}{Tint}}, \tag{1.17}$$

де $0<x<1$ – рівномірно розподілене випадкове число.

За (1.17), остаточно, для очікуваного часу генерації пакета кожним користувачем визначаємо:

$$t = -T\text{int} \cdot \ln(1 - x). \qquad (1.18)$$

Потім час повторної передачі пакета Rint у програмі встановлюється рівним очікуваному часу генерації пакета Tint:

**Tint= -Ttime/ log (1-G/Mnum);**

**Rint=Tint;**.

Відповідно інтервал повторної передачі Rint також описується експоненціальним розподілом, як і час генерації пакетів:

Для реальних систем час повторної передачі пакета залежить від конкретного протоколу доступу, що використовується. Наприклад, для протоколу ALOHA з виділенням часових інтервалів пакети повторно пересилаються у межах 10 часових слотів.

Як було відзначено, основною задачею моделювання є визначення залежності пропускної здатності і середнього часу затримки від значення повного інформаційного обсягу. Прийняття фіксованого значення для часу повторної передачі пакету означає, що реальний трафік, який буде формуватися у моделі, відрізнятиметься від того трафіка, який було заплановано відповідним оператором циклу (**G=[0.1:0.1:1,1.2:0.2:2)**, у бік збільшення.

У десятому блоці виконується ініціалізація змінних, які характеризують якість роботи протоколу доступу і використовуються під час моделювання:

**Spnum=0;**

**Splen=0;**

**Tplen=0;**

**Wtime=0;**.

Визначається поточний час відповідно до обраної підпрограми моделювання і визначеного протоколу доступу:

**now_time=feval(char(protocol(pno)),-1);**.

У наступному блоці формується власне коренева програма, якою виступає цикл, що реалізовано оператором **while.** Конкретний протокол доступу моделюється відповідною підпрограмою.

Фундаментальні положення в процесі моделювання наступні. Стан усіх терміналів користувачів зберігається у змінній Mstate. Тоді під час зміни стану терміналу відбувається повернення до кореневої програми з наданням часу повернення:

**next_time=feval(char(protocol(pno)),now_time);.**

Якщо загальна кількість пакетів від терміналів користувачів, що передані до точки доступу успішно, перевищить визначену кількість, яка задана у змінній **spend=1000**, то процес моделювання для заданого трафіка закінчується:

```
if Spnum>=spend
  break
end.
```

У дванадцятому блоці розраховується кількість терміналів користувачів, що передають пакети до точки доступу:

```
idx=find(Mstate= =TRANSMIT | Mstate= =COLLISION );.
```

У наступному блоці проводиться оцінка пакетів, які успішно доставлені до точки доступу.

По-перше, таке моделювання проводимо для провідної мережі (коефіцієнт захоплення **capture**=0).

В попередньому блоці визначається саме випадок колізії, коли два і більше терміналів передають свої пакети. В такому випадку, оскільки, для провідної мережі це однозначно означає конфлікт з повним спотворенням пакетів, для змінної Mstate присвоюється значення COLLISION.

```
if capture = =0
   if length(idx)>1
      Mstate(idx)=COLLISOIN
   end.
```

Для безпровідної мережі змінна **capture** встановлюється рівною одиниці. За таких умов визначимо успішність доставляння пакету до точки доступу.

Як було відзначено, проявлення ефекту захоплення проявляється у тому, що пакет від терміналу користувача, який у точці прийому має найбільший рівень потужності може підлягати подальшій обробці.

Для оцінки потужності від кожного терміналу у точці прийому спочатку необхідно розрахувати відстань між точкою доступу і терміналом, що на цей момент передає пакет.

Потім у точці доступу необхідно розрахувати відношення сигнал/шум для усіх пакетів від різних користувачів. З цього набору пакетів обирається пакет, який має найбільше значення відношення сигнал/шум.

Для обраного пакету знову розраховується миттєве значення сигнал/шум, але вже з врахуванням завадових сигналів від інших терміналів, які тепер формують рівень системних шумів каналу.

Якщо розраховане значення сигнал/шум більше за порогове значення, то вважається, що дані пакету успішно передані до точки доступу і результат зберігається у змінній Mstate.

Але, якщо розраховане значення сигнал/шум менше за порогове значення, то усі пакети вважаються втраченими через колізію і результат колізії також зберігається у змінній Mstate.

Тобто, значення Mstate =TRANSMIT в процесі передачі означає успішну передачу пакету, у той час як Mstate =COLLSION свідчить про втрату пакетів через конфлікт:

```
else
   if   length(idx)>1
        dxy=distance(bxy,mxy(idx,;),1);
        pow=mpow*dxy.^-alfa.*10. ^(sigma/10*mrnd(idx));
        [maxp no]=max(pow);
        if    Mstate(idx(no))== TRANSMIT
              if length(idx)==1
                    cn=10*log10(maxp);
             else
                    cn=10*log10(maxp/sum(sum(pow)-maxp+1));
             end
             Mstate(idx)== COLLISION
             if cn=tcn
             Mstate(idx(no)) = TRANSMIT
    else
              Mstate(idx)== COLLISION
     end
   end
end
```

Лічильник часу збільшується кожного разу, коли стан кожного терміналу користувача змінюється (тобто відбувається генерація нового пакету):

**now_time=next_time;.**

По закінченні симуляції, відповідно до заданого значення трафіку G, результати моделювання – середній трафік, пропускна

здатність і середня затримка передачі повинні висвітлюватися на екрані зі збереженням цих даних у файлі.

Середній трафік визначається як відношення часу на генерацію і передачу усіх пакетів (з урахуванням повторної пердачі) до повного часу моделювання.

Пропускна здатність визначається як відношення сумарного часу, впродовж якого пакети були успішно доставлені до точки доступу, до повного часу моделювання.

За допомогою підпрограми **theorys.m** обчислюються також значення пропускної здатності і середнього часу затримки передачі, які відповідають результатам теоретичного аналізу. Відповідний блок набуває такого вигляду:

```
traffic =Tplen/Srate/now_time;
ts = theorys(pno,traffic, Dtime);
fprintf(fid,'%f\t%f\t%f\t%f\t%f\t%f\t%f\n',G,now_time…
,Splen/Srate,Tplen/Srate,Wtime,Tint,Rint);
fprintf('G=%f\tS=%f\tTS=%f\n',traffic,Splen/Srate/now_time,ts);.
```

По закінченню виводу результатів моделювання для заданого значення трафіка G, файл, в якому зберігалися результати моделювання, закривається і результати виводяться на екран дисплея:

```
fclose(fid);
graph(outfile);.
```

### 1.4 Опис підпрограм, які необхідні для функціонування головної програми

#### 1.4.1 Підпрограма розташування терміналів користувачів

Розташування точки доступу і терміналів користувачів у зоні обслуговування реалізується у підпрограмі position.m. [1]. Повний лістинг підпрограми наведено у додатку А.

Геометрична модель розташування терміналів користувачів і точки доступу, що реалізована у підпрограмі position.m, зображена на рис.1.6.

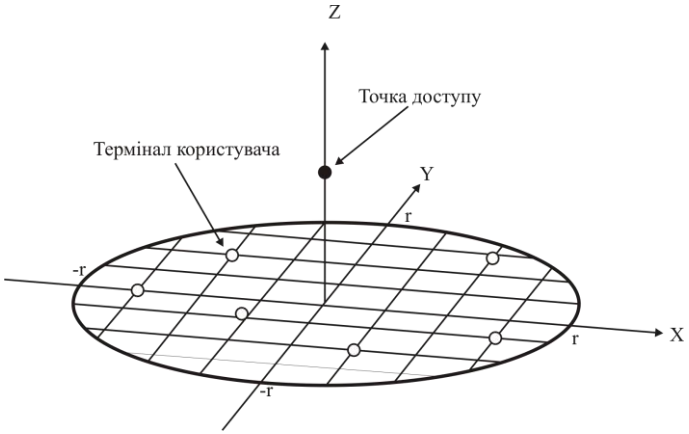



Рисунок 1.6 – Геометричне подання розташування терміналів користувачів у зоні обслуговування

Вхідними аргументами для підпрограми виступають радіус кругової зони обслуговування r, кількість терміналів користувачів n, що випадково розташовані у межах зони з радіусом r, і висота розташування терміналу h. Центр зони розглядається як початок координат (0,0).

Розташування усіх терміналів користувачів у межах зони повинно бути довільним. Висота кожного терміналу задається значенням параметра h. Якщо встановлено значення h=1, то висота розташування кожного терміналу обирається випадково у діапазоні 1…4. Вихідна інформація про розташування терміналів користувачів подається у вигляді матриці розміром n*3.

### 1.4.2 Підпрограма розрахунку відстані між точкою доступу і терміналом користувача

Обчислення відстані між конкретним терміналом і точкою доступу реалізується у підпрограмі distance.m. [1]. Вхідними аргументами для підпрограми виступають координати точки доступу - bstn, координати конкретного терміналу користувача – mstn і масштабний коефіцієнт sc1. Координати, як точки доступу, так і

терміналів користувачів, задаються відповідними значеннями по осях x, y, z згідно з рис.1.6. Якщо масштабний коефіцієнт sc1 рівняється одиниці, то відстань між точкою доступу і терміналом користувача визначається безпосередньо визначеними координатами. Якщо ж значення sc1 відмінно від одиниці, то відбувається збільшення відстані між точкою доступу і терміналом користувача у sc1 раз (масштабний коефіцієнт).

### 1.4.3 Підпрограма виводу результатів теоретичного аналізу

Для порівняння результатів моделювання і результатів теоретичного аналізу передбачається побудова графічних залежностей для кожного з протоколів у відповідності з результатами теоретичного аналізу [1, 2]. Результати теоретичного аналізу моделюються у підпрограмі theorys.m.

Вхідними аргументами програми виступають номер протоколу – **no**, заданий трафік і нормалізована затримка передачі. Для **no =1, 2, 3, 4** виводяться результати для чистої схеми ALOHA, ALOHA з виділенням часових інтервалів (слотована ALOHA), протоколу CSMA і слотованої схеми конкурентного доступу ISMA.

### 1.4.4 Підпрограма виводу результатів моделювання

Вивід результатів моделювання відбувається за допомогою підпрограми graph.m.

Ця програма реалізує вивід графічних залежностей між трафіком і пропускною здатністю, а також між трафіком і середнім часом затримки передачі. Вхідним аргументом graph.m. виступає змінна filename, яка указує ім’я файла, в якому зберігаються результати моделювання.

## 1.5 Моделювання чистої схеми ALOHA

Як відомо, для протоколу ALOHA ніякого планування передачі не відбувається [2]. Перед початком передачі усі користувачі не мають ніякого уявлення про зайнятість каналу. Відповідно передача конкретним терміналом відбувається відразу після з’явлення

інформації для передачі. Принцип роботи протоколу ALOHA, який реалізовано в програмному комплексі зображено на рис.1.7.

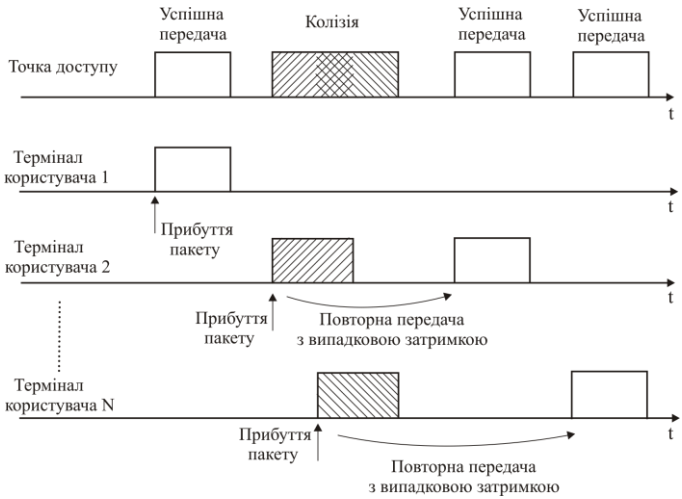



Рисунок 1.7 – Принцип роботи протоколу ALOHA

На рис.1.7 відображено також момент з’явлення колізії між другим і N- тим користувачем і механізм повторної передачі з випадковою затримкою у випадку виникнення колізії.

Більш детально механізм виникнення колізії зображено на рис.1.8.

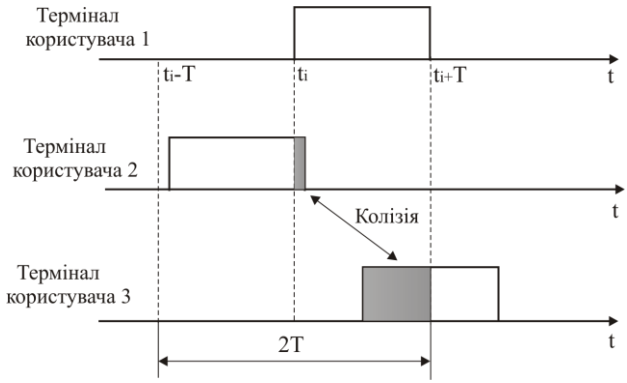



Рисунок 1.8 – Умови виникнення колізії

З аналізу епюр можна зробити висновок, що за умови фіксованої довжини кожного пакету і часу передачі пакету T, пакет може вважатися успішно переданим до точки доступу, якщо жоден з інших користувачів не почне передачу пакета впродовж інтервалу 2T, тобто від моменту часу $t_i$-T до моменту часу $t_i$+T, як це показано на рис.1.8.

За таких умов нормована пропускна здатність визначається таким виразом :

$$\rho = Ge^{-2G},$$

і досягає максимуму ρ=0,178 при G=0,5.

### 1.5.1 Підпрограма моделювання протоколу ALOHA

Моделювання роботи чистої схеми ALOHA реалізовано у підпрограмі paloha.m. [1]. Повний лістинг програми наведено у додатку С. Вхідним аргументом програми виступає змінна now_time, яка характеризує поточний час. Вихідним параметром програми є змінна next_time, яка надає інформацію про час зміни статусу кожного з терміналів користувачів після виконання функцій протоколу ALOHA.

В таблиці 1.2 наведені параметри, які характеризують основні параметри моделювання.

Таблиця 1.2 – Умови моделювання протоколу ALOHA

| Назва змінної | Фізичний сенс | Значення |
|---|---|---|
| r | Радіус зони обслуговування | 100 м |
| bxy | Висота розташування точки доступу | 5м |
| Mnum | Кількість терміналів | 100 |
| Srate | Символьна швидкість | 256 ксимвол/секунда |
| Plen | Довжина пакету | 128 символів |
| alfa | Коефіцієнт затухання | 3 |
| sigma | Стандартна девіація для логарифмічно-нормального закону | 6 дБ |

Продовження таблиці 1.2

| Назва змінної | Фізичний сенс | Значення |
|---|---|---|
| mcn | Відношення сигнал/шум у точці доступу, коли термінал знаходиться на межі зони обслуговування | 30 дБ |
| tcn | Коефіцієнт захоплення | 10 дБ |

Подібно до головної програми у підпрограмі paloha.m можна провести певну структуризацію, що дозволяє виділити в ній шість функціональних блоків.

У першому блоці аналогічно до головної програми вводяться глобальні змінні:

```
global STANDBY  TRANSMIT  COLLISION  PERMIT
global Srate  Plen
global  Mnum  Mplen  Mstate
global  Tint  Rint
global  Spnum  Splen  Tplen  Wtime
persistent mgtime, mtime
```

У другому блоці відбувається ініціалізація стану усіх терміналів користувачів для поточного часу **now_time <0**. Саме в цьому блоці відбувається розрахунок початку часу генерації першого пакету і завдання довжини пакету.

```
if now_time<0
  mgtime= -Tint*log(1-rand(1,Mnum));
        mtime =mgtime;
  Mstate=zeros(1,Mnum);
  Mplen(1:Mnum)=Plen;
        mtime(idx)=now_time+Mplen(idx)/Srate;
        next_time=min(mtime) ;
        return
end.
```

У третьому блоці по закінченні сеансу пакетної передачі підраховується кількість успішно переданих пакетів, розраховується

час затримки і визначається момент генерації нового пакету для усіх терміналів.

Причому для кожного терміналу генерація нового пакету не починається до тих пір поки не відбудеться успішна передача пакету конкретним терміналом до точки доступу (пакет від кожного терміналу не досягне точки доступу):

```
idx=find(ntime==now_time&Mstate==TRANSMIT);
if length(idx)>0
Spnum= Spnum+1;

Splen= Splen+Mplen(idx);
Wtime=Wtime+now_time-mgtime(idx)
Mstate==STANDBY;
Wtime=Wtime+now_time-mgtime(idx)
mgtime(idx)= now_time-Tint*log(1-rand);

mtime(idx)= mgtime(idx) ;
end
```

У четвертому блоці, після закінчення сеансу пакетної передачі, у разі невдалої спроби передачі пакету проводиться обчислення часу повторної передачі пакету для усіх терміналів, що прийняли участь у колізії.

```
idx=find(ntime==now_time&Mstate==COLLISION);
if length(idx)>0
    Mstate(idx)=STANDBY;
    Mtime(idx)=now_time-Rint*log(1-rand(1,length(idx)));
end
```

У п’ятому блоці визначаються усі термінали, які на поточний момент (змінна now_time) здійснюють передачу пакету. Відбувається зміна їх стану на режим передачі – TRANSMIT і проводиться розрахунок кінця передачі.

Потім підраховується кількість переданих пакетів.

```
idx=find(mtime==now_time);
if length(idx)>0
Mstate(idx)=TRANSMIT;
mtime(idx)=now_time+Mplen(idx)/Srate;
Tplen=Tplen+sum(Mplen(idx));
end
```

І, нарешті, у шостому блоці на підставі вибору мінімального значення **mtime**, визначається найближчий поточний час **next_time** для якого відбувається зміна статусу терміналу користувача**next_time=min(mtime);.**

## 1.6 Порядок проведення лабораторної роботи

### 1.6.1 Моделювання протоколу ALOHA для провідної мережі

1.6.1.1 Відкрити середовище Matlab

На екрані з’явиться вікно з панеллю інструментів Matlab (рис.1.9 ).

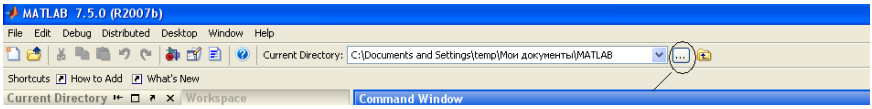



Рисунок 1.9 –Панель інструментів Mathlab

По-перше, необхідно в директорії Matlab створити свою директорію (папку), в яку необхідно розмістити основну програму, а також усі підпрограми, що необхідні для моделювання. Для цього необхідно натиснути кнопку, яка відзначена на рис.1.9. Для нашого прикладу створена директорія LAB1 (рис.1.10).

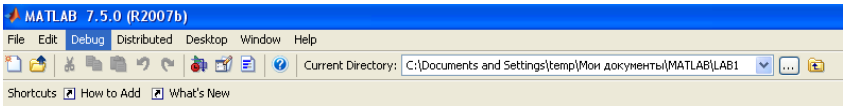



Рисунок1.10 –Створення директорії LAB1 для моделювання

1.6.1.2 Скопіювати файл основної програми main.m, який знаходиться у додатку А методичних вказівок і повернутися до середовища Matlab.

1.6.1.3 Відкрити на панелі інструментів меню File і у вікні, яке відкрилося, задати команду створення нового файлу New, що призведе до відкриття нового вікна, в якому обирається створення, так званого, М-файла (файла середовища Matlab) (рис.1.11).

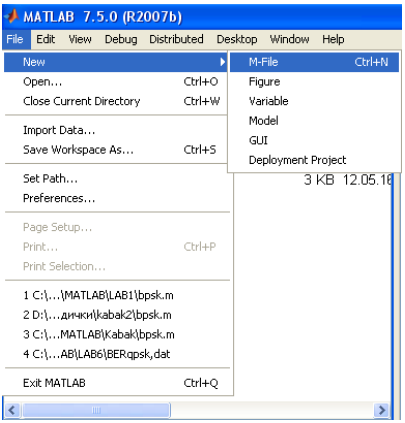



Рисунок.1.11 – Створення М-файла

Натиснути на кнопку M-file. В результаті цих дій відкриється редактор (Editor) середовища Matlab (рис.1.12) з відкритим полем для створення нового М-файла.

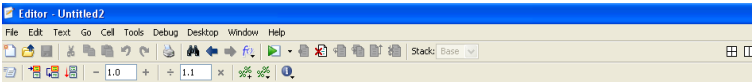



Рисунок 1.12–Вікно редактора (Editor) Matlab

1.6.1.4 Вставити скопійований файл main.m до редактора. Переконатися, що у головній програмі main.m значення параметру **capture=0** (це означає, що проводиться моделювання провідної мережі).

Зберегти скопійований файл під ім’ям main у створеній директорії LAB1 командою Save file as (рис.1.13).

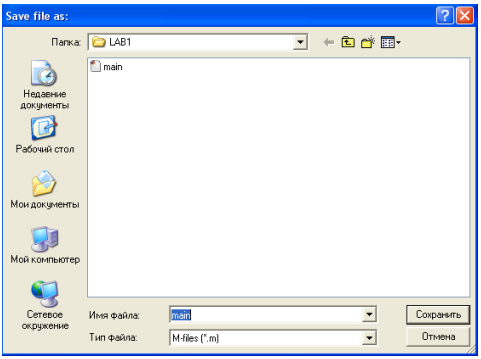



Рисунок.1.13 – Зберігання М-файла у директорії LAB1

В результаті у вікні поточної директорії LAB1 з'явиться скопійований файл з розширенням .m (рис.1.14).

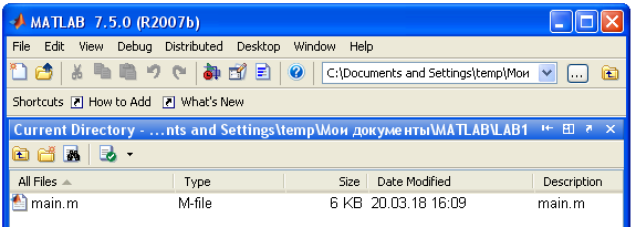



Рисунок.1.14 – Вигляд поточної директорії LAB1 після запису першого файлу main.m

1.6.1.5 Скопіювати у додатку А підпрограму paloha.m, що моделює роботу протоколу ALOHA і повторити процедуру створення нового М-файлу в тій самій директорії LAB1 (тепер можна просто відкрити редактор та скопіювати у новому вікні файл і зберегти його у директорії LAB1 як m.файл). Файл зберегти під ім'ям, що повторює назву функції – тобто, paloha.m.

1.6.1.6. Скопіювати у додатку А підпрограму position.m, що реалізує розташування точки доступу і терміналів користувачів у зоні обслуговування, і повторити процедуру створення нового М-файла в

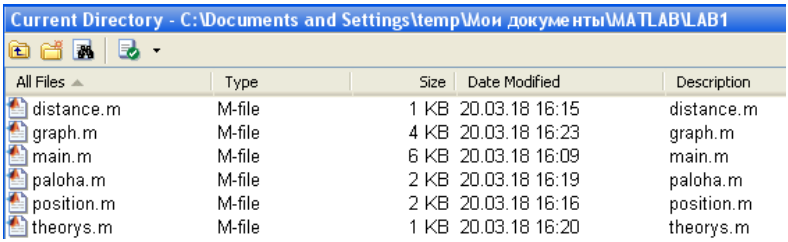
тій самій директорії LAB1. Файл зберегти під ім’ям, що повторює назву функції – position.m.

1.6.1.7 Скопіювати у додатку А підпрограму distance.m, яка обчислює відстані між конкретним терміналом і точкою доступу, та повторити процедуру створення М-файла в директорії LAB1. Файл зберегти під ім’ям, що повторює назву функції –distance.m.

1.6.1.8 Скопіювати у додатку А підпрограму theorys.m яка репрезентує теоретичні вирази для оцінки пропускної здатному різних протоколів множинного доступу і повторити процедуру створення М-файла в директорії LAB1. Файл зберегти під ім’ям, що повторює назву функції –theorys.m .

1.6.1.9 Скопіювати у додатку А підпрограму graph.m, яка реалізує побудову графічних залежностей результатів моделювання і повторити процедуру створення М-файла в директорії LAB1. Файл зберегти під ім’ям, що повторює назву функції – graph.m.

Після виконання вище визначених пунктів в основному вікні середовища Matlab усі шість файлів повинні відобразитися у полі поточної директорії (Current Directory) – рис.1.15.


Current Directory - C:\Documents and Settings\temp\Мои документы\MATLAB\LAB1

| All Files | Type | Size | Date Modified | Description |
|---|---|---|---|---|
| distance.m | M-file | 1 KB | 20.03.18 16:15 | distance.m |
| graph.m | M-file | 4 KB | 20.03.18 16:23 | graph.m |
| main.m | M-file | 6 KB | 20.03.18 16:09 | main.m |
| paloha.m | M-file | 2 KB | 20.03.18 16:19 | paloha.m |
| position.m | M-file | 2 KB | 20.03.18 16:16 | position.m |
| theorys.m | M-file | 1 KB | 20.03.18 16:20 | theorys.m |

Рисунок.1.15 – Вигляд поточної директорії LAB1 після запису усіх підпрограм, що використовуються при моделюванні

1.6.1.10 У командній строчці командного вікна (Command Window) набрати назву основної програми main і запустити аналіз натисканням клавіші Enter.

Якщо синтаксичні помилки у програмах відсутні, то на дисплеї у командному вікні повинен відобразитися надпис “Simulation Start”, а потім рядок “paloha without capture effect”.

Далі відповідно до заданого трафіку буде поступово виводитися таблиця з реальним значенням трафіку G, пропускної здатності за

результатами моделювання S і значеннями теоретичної пропускної здатності TS.

Ознакою закінчення моделювання, тобто коли трафік досягне приблизно заданої межі G=2, буде висвітлення рядка “Simulation End”. Додатково, згідно з підпрограмою **graph.m,** будуть виведені два графіки – перший для пропускної здатності провідної мережі, а другий для середнього часу затримки передачі пакетів.

1.6.1.11 Зафіксувати табличні значення моделювання і відобразити їх та отримані графічні залежності у звіті.

### 1.6.2 Моделювання протоколу ALOHA для безпровідної мережі

1.6.2.1 В поточній директорії LAB1 відкрити файл головної програми main.m. Відповідно лістинг програми висвітлиться у вікні редактора Matlab. Змінити значення параметра capture з 0 на одиницю (**capture=1**). Зберегти файл натиснувши на кнопку Save.

1.6.2.2 Перейти до командного вікна і набрати у командному рядку main і запустити моделювання. Почнеться розрахунок безпровідної мережі, ознакою якого буде виведення рядку “ paloha with capture effect ”.

Аналогічно до провідної мережі буде поступово виводитися таблиця з реальним значенням трафіку G, пропускної здатності за результатами моделювання S, але тепер для безпровідної мережі.

1.6.2.3 Зафіксувати табличні значення моделювання і відобразити їх та отримані графічні залежності для безпровідної мережі у звіті.

## 1.7 Зміст звіту

1.7.1 Базовий алгоритм моделювання. Перелік змінних програми main.m і paloha.m з визначенням їх сенсу.

1.7.2 Результати експериментальних досліджень у вигляді табличних та графічних залежностей як для провідної , так і для безпровідної мереж.

1.7.3 Висновки за результатами роботи.

### 1.8 Контрольні запитання

1. Класифікація протоколів множинного доступу.

2.До якого виду протоколів відноситься протоколALOHA . Основні дії які виконуються під час застосування протоколу.

3. Пояснити графічно процес виникнення колізії у протоколах ALOHA.

4. Які характеристики вводяться для оцінки ефективності протоколів множинного доступу.

5. Як в лабораторній роботі проводиться моделювання провідної і безпровідної мереж.

6. Приведіть ідеальні залежності для пропускної здатності і середнього часу затримки.

7. Умови відсутності колізій для протоколу ALOHA.

8. Провести аналіз ефективності протоколу ALOHA.

9. Пояснити отримані залежності для пропускної здатності для провідної і безпровідної мереж.

10. Пояснити отримані залежності для середнього часу затримки передачі пакетів  для провідної і безпровідної мереж.

# 2 ЛАБОРАТОРНА РОБОТА № 2 МОДЕЛЮВАННЯ ПРОТОКОЛУ МНОЖИННОГО ДОСТУПУ S-ALOHA

## 2.1 Мета роботи

**2.1.1** Ознайомитися з принципами побудови протоколів множинного доступу у безпровідних мережах.

**2.1.2** За допомогою ЕОМ провести симуляцію роботи конкурентного протоколу доступу з повторенням S-ALOHA.

**2.1.3** Отримати показники ефективності протоколу S-ALOHA.

## 2.2 Завдання на підготовку до виконання лабораторної роботи

### 2.2.1 Обґрунтування базового алгоритму програмного комплексу

Базовий алгоритм програмного комплексу залишається повністю тотожним алгоритму, запропонованому у першій лабораторній роботі для алгоритму ALOHA.

Модифікація протоколу S-ALOHA порівняно з класичним протоколом ALOHA полягає у тому, що повідомлення можуть бути переданими тільки впродовж часового інтервалу (слоту), який розташований між двома синхронізуючими імпульсами.

Причому передача пакету може відбуватися тільки на початку часового слоту. Відзначається, що кількість колізій з дотриманням такої роботи зменшується практично у два рази [1, 2].

Принцип дії протоколу зображено на рис.2.1. Пакет згенерований у визначеному часовому слоті передається до точки доступу у наступному слоті.

Для успішної передачі пакету до точки доступу кількість пакетів, що згенеровані в одному слоті, не повинна перевищувати одиницю. Якщо в слоті формуються декілька пакетів, то виникає колізія.

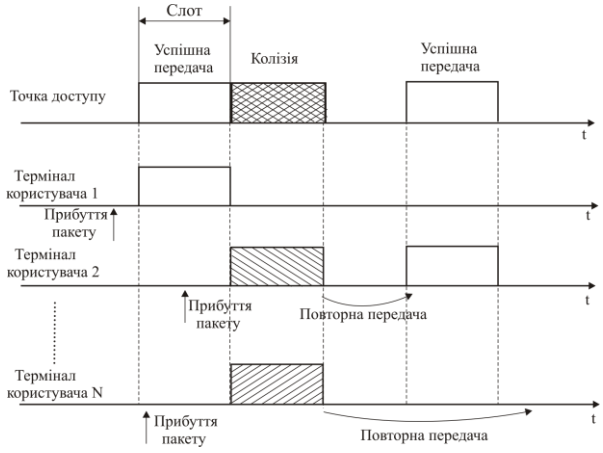



Рисунок 2.1 – Принцип роботи протоколу ALOHA з виділенням часових інтервалів

Більш детально механізм виникнення колізії зображено на рис.2.2.

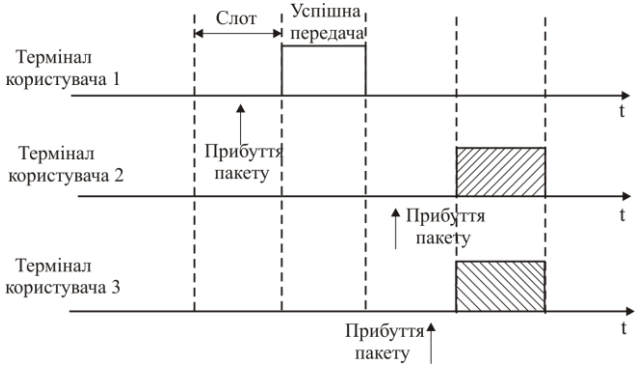



Рисунок 2.2 – Механізм виявлення колізій у протоколі ALOHA з виділенням часових інтервалів

Пропускна здатність протоколу S-ALOHA визначається виразом [1, 2]:

$$\rho = Ge^{-G},$$

і досягає максимуму ρ=0,368 при G=1.

### 2.2.2 Підпрограма моделювання протоколу S-ALOHA

Моделювання протоколу S-ALOHA реалізується за допомогою підпрограми saloha.m, повний лістинг якої наведено у додатку Б.

В головному підпрограма saloha.m подібна до підпрограми моделювання чистої схеми ALOHA - paloha.m. Різниця, як було визначено вище, полягає у синхронізації передачі пакетів з формуванням часових слотів.

## 2.3 Порядок проведення лабораторної роботи

Відкрити середовище Matlab. Створити нову директорію (папку) LAB2, в яку необхідно розмістити основну програму, а також усі підпрограми, що необхідні для моделювання (рис.2.3).

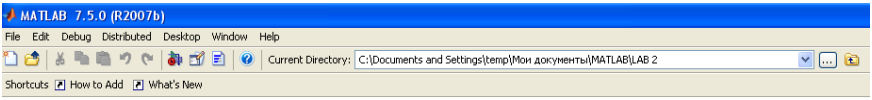
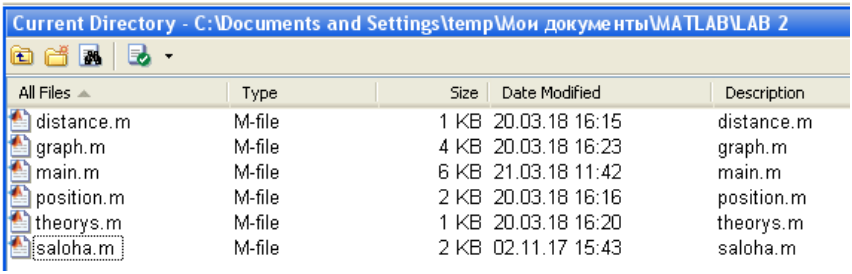



Рисунок 2.3 – Створення директорії LAB2 для моделювання

Для моделювання S-ALOHA необхідним є практично той же набір підпрограм, який застосовувався у лабораторній роботі №1, за винятком підпрограми paloha.m, а саме:

**main.m;**
**position.m;**
**distance.m;**
**graph.m;**
**theory.m.**

Тому, необхідно скопіювати ці файли, які були створені у директорії LAB1 (або з методичних вказівок) і зберегти їх під тими ж іменами у директорії LAB2. Додатково необхідно у додатку Б скопіювати підпрограму saloha.m і зберегти її у директорії LAB2. В результаті у вікні поточної директорії повинні відобразитися повний набір М-файлів, які необхідні для моделювання (рис.2.4).



Current Directory - C:\Documents and Settings\temp\Мои документы\MATLAB\LAB 2

| All Files | Type | Size | Date Modified | Description |
|---|---|---|---|---|
| distance.m | M-file | 1 KB | 20.03.18 16:15 | distance.m |
| graph.m | M-file | 4 KB | 20.03.18 16:23 | graph.m |
| main.m | M-file | 6 KB | 21.03.18 11:42 | main.m |
| position.m | M-file | 2 KB | 20.03.18 16:16 | position.m |
| theorys.m | M-file | 1 KB | 20.03.18 16:20 | theorys.m |
| saloha.m | M-file | 2 KB | 02.11.17 15:43 | saloha.m |

Рисунок.2.4 – Вигляд поточної директорії LAB2 після запису усіх підпрограм, що використовуються при моделюванні

### 2.3.1 Моделювання протоколу S-ALOHA для провідної мережі

Для проведення моделювання за протоколом S-ALOHA необхідно відкрити у редакторі Matlab файл головної програми main.m і провести такі зміни:

**– замінити значення параметру pno=1 (протокол ALOHA), який визначає номер відповідного протоколу доступу, на pno=2 (протокол S-ALOHA);**

**– встановити значення параметру capture рівним 0 (задаємо алгоритм моделювання провідної мережі);**

**– оскільки очікувана пропускна здатність протоколу і, відповідно. реальний трафік збільшаться, то необхідно провести заміну заданого трафіку, який надається в операторі циклу (максимальний нормований трафік збільшено до 4 , порівняно з ALOHA):**

**for G=[0.1:0.1:1,1.2:0.2:4].**

Зберегти зміни у файлі натисканням кнопки Save**.**

У командному вікні (Command Window) набрати назву основної програми main і запустити аналіз натисканням клавіші Enter.

Якщо синтаксичні помилки у програмах відсутні, то на дисплеї у командному вікні відобразиться надпис “ Simulation Start ”, а потім рядок “ saloha without capture effect ” – тобто, моделювання роботи провідної мережі.

Далі відповідно до заданого трафіку буде поступово виводитися таблиця з реальними значеннями трафіку G, пропускної здатності S за результатами моделювання і значеннями теоретичної пропускної здатності TS.

Ознакою закінчення моделювання, тобто коли трафік досягне приблизно заданої межі G=4, буде висвітлення рядка “Simulation End”.

Подібно до алгоритму ALOHA будуть виведені два графіки – перший для пропускної здатності провідної мережі, а другий для середнього часу затримки пакетів.

Зафіксувати табличні значення моделювання і відобразити їх та отримані графічні залежності у звіті.

### 2.3.2 Моделювання протоколу S-ALOHA для безпровідної мережі

У поточній директорії LAB2 відкрити файл головної програми main.m. Відповідно лістинг програми висвітлиться у вікні редактора Matlab. Змінити значення параметра capture з 0 на одиницю (**capture=1**). Зберегти файл натисканням на кнопку Save.

Перейти до командного вікна і набрати команду main та запустити моделювання. Почнеться розрахунок безпровідної мережі, ознакою якого буде виведення рядку “ saloha with capture effect ”.

Аналогічно до провідної мережі буде поступово виводитися таблиця з реальними значенням трафіку G, пропускної здатності за результатами моделювання S, але тепер для безпровідної мережі.

Зафіксувати табличні значення моделювання і отримані графічні залежності для провідної мережі у звіті.

## 2.4 Зміст звіту

2.4.1 Базовий алгоритм моделювання. Перелік змінних програми main.m і saloha.m з визначенням їх сенсу.

2.4.2 Результати експериментальних досліджень у вигляді табличних та графічних залежностей як для провідної, так і для безпровідної мереж.

2.4.3 Висновки за результатами роботи.

## 2.5 Контрольні запитання

1. До якого виду протоколів відноситься протокол S-ALOHA . Основні дії які виконуються під час застосування протоколу.

2. Пояснити графічно процес виникнення колізії для S-ALOHA.

3. Умови відсутності колізій для протоколу S-ALOHA.

4. Провести аналіз ефективності протоколу S-ALOHA.

5. Приведіть залежності для пропускної здатності і середнього часу затримки протоколу S-ALOHA.

6. Пояснити отримані залежності для пропускної здатності як провідної, так і безпровідної мереж.

7. Пояснити отримані залежності для середнього часу затримки передачі пакетів для провідної і безпровідної мереж.

8. Порівняти протоколи ALOHA і S-ALOHA за основними показниками ефективності.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Harada H. Simulation and software radio for mobile telecommunications [Текст] / H. Harada, R. Prasad. – Artech House, 2003. – 465 p.

2. Скляр Б. Цифровая связь. Теоретические основы и практическое применение [Текст] / Б. Скляр; пер. с англ.; – М.: Издательский дом “Вильямс”, 2003. – 1104 с.

3. Феер К. Беспроводная цифровая связь [Текст] / К. Феер; пер. с англ.; – М.: Радио и связь, 2000. – 519 с.

# ДОДАТОК А

Лістинги програм необхідних для моделювання протоколу ALOHA

```
%main.m
%Packet communication system
%MATLAB version
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


clear;
            %definition of the global variable

global STANDBY TRANSMIT COLLISION PERMIT
global Srate Plen Ttime Dtime
global Mnum Mplen Mstime Mstate
global Tint Rint
global Spnum Splen Tplen Wtime

           %definition of the fixed number
STANDBY =  0;
TRANSMIT = 1;
COLLISION =  2;
PERMIT = 3;

             %definition of the protocol
protocol = {'paloha' ...  %pure ALOHA:      pno = 1
         'saloha' ...  %Slotted ALOHA:    pno = 2
         'npcsma' ...  %np CSMA:         pno = 3
         'snpisma' ...%Slotted np CS CSMA:  pno = 4
      };


%definition of communication channel
brate = 512e3;          %bit rate
Srate = 256e3;          %symbol rate
Plen = 128;             %length of a packet
Dtime = 0.01;           %normalized propagation delay
alfa = 3;               %decline fixed number of propagation loss
sigma = 6;              %standard deviation of shadowing [dB]
r = 100;                %service area radius [m]
bxy = [0, 0, 5];        %position of the access point(x,y,z) [m]
```

```
tcn = 10;                  %capture ratio [dB]

                           %definition of the access terminals
Mnum = 100;                %number of the access terminal
mcn = 30;
%C/N at the access point when  transmitted from area edge

%simulation condition
pno = 1;                   % protocol number
capture =0;                 %capture effect     0:nothing ,  :consider
spend = 10000;             %number of packets that
                           % simulation is finished
outfile = 'test.dat';      %name of file to store the result

Ttime = Plen /Srate;
%transmission time of one packet
mpow = 10^(mcn/10)*sqrt(r^2+bxy(3)^2)^alfa;
%true value of C/N

fid = fopen(outfile, 'w');
fprintf(fid,'Protocol              = %d\n',pno);
fprintf(fid,'Capture               = %d\n',capture);
fprintf(fid,'Normalize_delay_time     = %f\n',Dtime);
fprintf(fid,'Bit_rate(bps)           = %d\n',brate);
fprintf(fid,'Symbol_rate(sps)        = %d\n',Srate);
fprintf(fid,'Length_of_Packet(sbl)    = %d\n',Plen);
fprintf(fid,'Number_of_mobile_station = %d\n',Mnum);
fprintf(fid,'Transmission_power(C/N)  = %f\n',mcn);
fprintf(fid,'Capture_ratio(dB)        = %f\n',tcn);
fprintf(fid,'Number_of_Packet         = %d\n',spend);
fprintf('\n*********Simulation Start*********\n\n');
            if capture ==0
     fprintf('%s without capture effect\n\n',char(protocol(pno)));
        else
     fprintf('%s with capture effect\n\n',char(protocol(pno)));
            end

mxy = position (r, Mnum, 0);                 %positioning of the access terminals
randn('state', sum(100*clock));               %resetting of the random table
mrnd = randn(1,Mnum);                        %decision of the shadowing

for G=[0.1:0.1:1,1.2:0.2:2]                 % offered traffic
```

```
 if G>=Mnum
  break
end

Tint = -Ttime /log(1-G/Mnum);
%expectation value of the packet generation interval
Rint = Tint;
%expectation value of the packet resending interval

Spnum = 0;
Splen = 0;
Tplen = 0;
Wtime = 0;
now_time = feval (char(protocol(pno)),-1);
%initialize of the access terminals

while 1
    next_time = feval(char(protocol(pno)),now_time);

    if Spnum >= spend
        %disp(Spnum);
         break
    end
    idx = find(Mstate==TRANSMIT | Mstate==COLLISION);
    if capture ==0                          %without capture effect
           if length (idx) >1
             Mstate (idx) = COLLISION;        %collision occurs
           end
   else                                     %with capture effect
      if length (idx) >1
             dxy = distance (bxy, mxy(idx,:),1);
%calculation of the distance
    pow = mpow*dxy.^-alfa.*10.^(sigma/10*mrnd(idx));


%calculation of received power
      [maxp no] = max(pow);
          if Mstate(idx(no)) == TRANSMIT
            if length(idx) == 1
                cn = 10*log10(maxp);
            else
  cn = 10*log10(maxp/(sum(pow)-maxp+1));
```

```
            end
            Mstate(idx) = COLLISION;
            if cn >= tcn          %received power larger than capture ratio

              Mstate (idx(no)) = TRANSMIT;  %transmitting success
            end
          else
            Mstate(idx) = COLLISION;
          end
      end
    end
 now_time = next_time;
 %time is advanced until the next state change time
end

        traffic = Tplen / Srate / now_time;   %calculation of the traffic
        ts = theorys(pno, traffic, Dtime);
%calculation of the theory value of the throughput
fprintf(fid,'%f\t%f\t%f\t%f\t%f\t%f\t%f\n',G,now_time,Splen/Srate,
Tplen/Srate, Wtime, Tint, Rint);
fprintf('G=%f\tS=%f\tTS=%f\n',traffic,Splen/Srate/now_time,ts);
end
fprintf('\n********* Simulation End *********\n\n');
fclose(fid);
graph(outfile);
%%%%%%%%%%%%%%%% end of file %%%%%%%%%%%%%%%%

position.m
%Positioning of the access terminals in the area of the radius r.
%Input arguments
% r : The radius r that an access point is an origin.
% n : The number of access terminals.
% h : h=0 – z=0 h=1 -> z=1-4
% Output argument
% posxy : (x,y,z)
function [posxy] = position(r,n,h)
ms = 4*r;          %calculation of the number of maximum position
ms = ms+4*sum(fix(sqrt(r^2-[1:r-1].^2)));
if n > ms
error('n exceeds the number of position.');
end
```

```
posxy = zeros(n,3); %initialize
  for ii=1:n
    while 1
       xx = round(r*rand)*sign(sin(2*pi*rand));
       %x and y are decided at random
       yy = round(r*rand)*sign(cos(2*pi*rand));
       if xx^2+yy^2 <= r^2 && (xx==0||yy==0)
       % (xx,yy) is not (0,0) in the area
                if isempty(find(posxy(:,1)==xx & posxy(:,2)==yy)) == 0   %
(xx,yy) are vacant
            break
         end
      end
    end
  posxy (ii, [1 2]) = [xx yy];
  if h==1
     while 1
        posxy (ii, 3) = round (50*rand) / 10;
        if 1 <=posxy (ii,3) && posxy (ii, 3) <=4
          break
        end
     end
  end
end
%%%%%%%%%% end of file %%%%%%%%%%


%distance.m
%The calculation of distance between access point and access terminal
%Input arguments
%bstn     : coordinate of access point (x,y,z)
%mstn     : coordinate of access terminals (x,y,z) (mstn vector or matrix)
%scl      : scale (if scl is omitted, scl=1.)
%Output argument
%d        : distance
function [d] = distance (bstn, mstn, scl)
if nargin <3            %scl is omitted
  scl = 1;
end
[v,h] = size (mstn);
d = sqrt(sum(rot90(((repmat(bstn,v,1)-mstn)*scl).^2)));
%%%%%%%%%% end of file %%%%%%%%%%
```

```
%paloha.m
%pure ALOHA System
%Input argument
%now_time  : now time but, now_time <0 initializes
%the access terminals
%Output argument
%next_time : next state change time
%
function [next_time]=paloha(now_time)
global STANDBY TRANSMIT COLLISION        %definition of the global
variable
global Srate Plen
global Mnum Mplen Mstate
global Tint Rint
global Spnum Splen Tplen Wtime
persistent mgtime mtime                  %definition of the static variable
if now_time < 0                          %initialize access terminals
  rand('state', sum(100*clock));         %resetting of the random table
  mgtime =-Tint*log(1-rand(1,Mnum));     %packet generation time
  mtime =mgtime;                         %packet transmitting time
  Mstate   =zeros(1,Mnum);
  Mplen(1:Mnum)=Plen;                    %packet length
  next_time=min(mtime);
  return
end
idx=find(mtime==now_time & Mstate==TRANSMIT); %findind of the
terminal which transmission succeeded
if length(idx)>0
  Spnum    =Spnum+1;
  Splen    =Splen+Mplen(idx);
  Wtime    =Wtime+now_time-mgtime(idx);
  Mstate(idx)=STANDBY;
  mgtime(idx)=now_time-Tint*log(1-rand);   %next packet generation time
  mtime(idx) =mgtime(idx);                 %next packet transmitting time
end
idx=find(mtime==now_time & Mstate==COLLISION);%finding of the
terminal which transmission failed
if length(idx)>0
  Mstate(idx)=STANDBY;
```

```
  mtime(idx)=now_time-Rint*log(1-rand(1,length(idx)));
%resending time
end
idx=find(mtime==now_time);                    %finding of the terminal which
transmission start
if length (idx)>0
  Mstate(idx) =TRANSMIT;
  mtime(idx)  =now_time+Mplen(idx)/Srate;        %end time of transmitting
  Tplen       =Tplen+sum(Mplen(idx));
end
next_time=min(mtime);                         %next state change time
%%%%%%%%%%%%%%% end of file %%%%%%%%%%%%%%%

%graph.m
%The function of drawing the graph of simulation result
%Input argument
%filename : name of the file in which simulation result was stored
%Output argument
%nothing
function graph (filename)
filename='test.dat';
mtitle1 = {'Throughput of Pure ALOHA system'        % definition of the title
        'Throughput of Slotted ALOHA system'
        'Throughput of np CSMA system'
        'Throughput of  Slotted np ISMA system'
    };
mtitle2 = {'Average Delay time of Pure ALOHA system' % definition of the
title
        'Average Delay time of Slotted ALOHA system'
        'Average Delay time of np CSMA system'
        'Average Delay time of Slotted ISMA system'
    };
fid      = fopen (filename,'r');
nul      = fscanf (fid, '%s', 2);
protocol = fscanf (fid, '%d', 1);
nul      = fscanf (fid, '%s', 2);
capture  = fscanf (fid, '%d', 1);
nul      = fscanf (fid, '%s', 2);
dtime    = fscanf (fid, '%f', 1);
nul      = fscanf (fid, '%s', 2);
brate    = fscanf (fid, '%d', 1);
nul      = fscanf (fid, '%s', 2);
```

```
srate    = fscanf (fid, '%d', 1);
nul      = fscanf (fid, '%s', 2);
plen     = fscanf (fid, '%d', 1);
nul      = fscanf (fid, '%s', 2);
mnum     = fscanf (fid, '%d', 1);
nul      = fscanf (fid, '%s', 2);
mcn      = fscanf (fid, '%f', 1);
nul      = fscanf (fid, '%s', 2);
tcn      = fscanf (fid, '%f', 1);
nul      = fscanf (fid, '%s', 2);
spend    = fscanf (fid, '%d', 1);
idx = 0;
while 1
  data0 = fscanf (fid, '%f', 1);
  data1 = fscanf (fid, '%f', 1);
  data2 = fscanf (fid, '%f', 1);
  data3 = fscanf (fid, '%f', 1);
  data4 = fscanf (fid, '%f', 1);
  data5 = fscanf (fid, '%f', 1);
  data6 = fscanf (fid, '%f', 1);
  if feof(fid) == 1                    %eof
    break
  end
  idx = idx+1;
  tg(idx) = data0;                     %offered traffic
  g(idx)  = data3 / data1;             %actual offered traffic
  s(idx)  = data2 / data1;             %actual throughput
  w(idx)  = data4 / spend*srate/plen;  %average delay
end
fclose(fid);
ts = theorys(protocol, g, dtime);      %calculation of the throughput
if protocol <3                         % Pure ALOHA & Slotted ALOHA
  plot(g,s,'bo:',g,ts,'r-');           %normal graph
  legend('result', 'theory',0);        %legend
else                                   %np-CSMA & Slotted np-ISMA
  semilogx(g,s,'bo:',g,ts,'r-');       %semilog graph
  if protocol == 3                     % legend
    legend (strcat('result(a=',num2str(dtime),')'),'theory',0);
  else
    legend (strcat('result(d=',num2str(dtime),')'),'theory',0);
  end
end
```

```
title(char(mtitle1(protocol)), 'FontSize', 16);         %title
xlabel('Traffic (Simulation result)', 'FontSize', 14);    %x axis label
ylabel ('Throughput', 'FontSize', 14);                  %y axis label
figure (2);                         %preparation of the new graph windows (line-
feed)
if protocol < 3                          %Pure ALOHA & Slotted ALOHA
  plot (g,w,'bo:');                        %normal graph
  legend ('result');                       %legend
else                                  %np-CSMA & Slotted np-ISMA
  semilogx (g,w,'bo:');                     %semilog graph
  if protocol == 3                        %legend
    legend (strcat ('result (a=',num2str(dtime),')'),0);
  else
    legend (strcat ('result (d=', num2str (dtime),')'),0);
  end
end
title(char(mtitle2(protocol)),'FontSize', 16);            %title
xlabel('Traffic(Simulation result)','FontSize', 14);      %x axis label
ylabel('Average Delay time (packet)','FontSize', 14);      %y axis label
%%%%%%%%%%% end of file %%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%theorys.m
%The calculation of the theory value of the throughput
%Input arguments
% no: protocol number
%                               1 : Pure ALOHA
%                               2 : Slotted ALOHA
%                               3 : np-CSMA
%                               4 : Slotted np-ISMA
```

```
%  g: offered treaffic (scalar or vector)
%  a: normalized propagation delay
% Output argument
% ts:the theory value of the throughput (scalar or vector)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function[ts] = theorys(no,g,a)
switch no
case 1                    % Pure ALOHA
    ts = g.*exp(-2*g);
case 2                    % Slotted ALOHA
    ts = g.*exp(-g);
case 3                    % Non-Persistent Carrier Sense Multiple Access
    ts = g.*exp(-a*g)./(g*(1+2*a)+exp(-a*g));
case 4                    % Slotted Non-Persistent ISMA
    ts = a *g.*exp(-a*g)./(1+a-exp(-a*g));
end

%%%%%%%%%%% end of file %%%%%%%%%%%
```

# ДОДАТОК Б

Повний лістинг підпрограми моделювання протоколу S-ALOHA

```
%saloha.m
%
%Slotted ALOHA System
%
%Input argument
%now_time: now_time but,
% now_time<0 initializes the access terminals
%Output argument
%next_time: next state change time
%
function[next_time]=saloha(now_time)
global STANDBY TRANSMIT COLLISION        %definition of the global
variable
global Srate Plen
global Mnum Mplen Mstate
global Tint Rint
global Spnum Splen Tplen Wtime

persistent mgtime mtime slot            %definition of the static variable
if now_time < 0                         %initialize access terminal
  rand('state', sum(100*clock));        %resetting of the random table
  slot=Plen/Srate;                      %slot length
  mgtime=-Tint*log(1-rand(1,Mnum));
  mtime=(fix(mgtime/slot)+1)*slot;      %packet transmitting time
  Mstate=zeros(1,Mnum);
  Mplen(1:Mnum)=Plen;                   %packet length
  next_time=min(mtime);
  return
end

idx=find(mtime==now_time & Mstate==TRANSMIT);
%finding of the terminal which transmission succeeded
if length(idx) > 0
  Spnum=Spnum+1;
  Splen=Splen+Mplen(idx);
  Wtime=Wtime+now_time-mgtime(idx);
  Mstate(idx)=STANDBY;
  mgtime(idx)=now_time-Tint*log(1-rand);
 %next packet generation time
```

```
  mtime(idx)=(fix(mgtime(idx)/slot)+1)*slot;
%next packet transmitting time
end

idx=find(mtime==now_time & Mstate==COLLISION);

%finding of the terminal which transmission failed

if length(idx) > 0
  Mstate(idx)=STANDBY;
  mtime(idx)=now_time-Rint*log(1-rand(1,length(idx)));  %waiting time
  mtime(idx)=(fix(mtime(idx)/slot)+1)*slot;    %resending time
end

idx=find(mtime==now_time);
 %finding of the terminal which transmission start
if length(idx) > 0
  Mstate(idx)=TRANSMIT;
  mtime(idx)=now_time+Mplen(idx)/Srate;
 %end time of transmitting
  mtime(idx)=round(mtime(idx)/slot)*slot;
  Tplen=Tplen+sum(Mplen(idx));
end

next_time=min(mtime);
 %next state change time
%%%%%%%%%%%%%%%% end of file
%%%%%%%%%%%%%%%%
```