Казахский национальный университет имени аль-Фараби

УДК 621.3 (043)

На правах рукописи

ТУРЛЫКОЖАЕВА ДАНА АБДИКУМАРОВНА

Информационно-энтропийный метод маршрутизации беспроводных сетей

8D06201 – Радиотехника, электроника и телекоммуникации

Диссертация на соискание степени
доктора философии (PhD)

Научный консультант:
доктор Phd,
Ахтанов С. Н.

Зарубежный научный консультант:
доктор технических наук,
профессор
Вуйцик В.

Республика Казахстан
Алматы, 2025

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей диссертации применяются термины с соответствующими определениями

ACO (Ant Colony Optimization) − Оптимизация с использованием алгоритма муравьиной колонии: алгоритм, основанный на моделировании поведения муравьёв в природе для нахождения оптимальных решений задач, таких как маршрутизация.

Ad hoc сети − Беспроводные сети, состоящие из устройств, которые соединяются друг с другом напрямую без централизованного управления или фиксированной инфраструктуры.

Adaboost (Adaptive Boosting) − Алгоритм машинного обучения, который объединяет несколько слабых классификаторов в один сильный, последовательно обучая их на ошибках предыдущих моделей и придавая больший вес неправильно классифицированным объектам.

AGC (Automatic Gain Control) − Автоматическое управление усилением. Система, которая автоматически регулирует уровень усиления сигнала.

AMC (Automatic Modulation Classification) − Автоматическая классификация модуляций: процесс, используемый в беспроводных системах для автоматического определения типа модуляции получаемого сигнала.

ANN (Artificial Neural Networks) − Искусственные нейронные сети. Классификатор, имитирующий работу человеческого мозга, используемый для распознавания сложных закономерностей и обработки больших объемов данных.

AODV (Ad hoc On-Demand Distance Vector) − Протокол маршрутизации для мобильных ad hoc сетей (MANET), который создает маршруты только по запросу, оперативно реагируя на изменения в топологии сети.

AWGN (Additive White Gaussian Noise) − Аддитивный белый гауссов шум: тип шума, который добавляется к сигналу и имеет равномерное распределение мощности на всех частотах.

Babel − векторный протокол маршрутизации, предназначенный для использования в беспроводных и проводных сетях, который сочетает в себе преимущества алгоритмов маршрутизации на основе векторного расстояния и протоколов на основе состояния.

BATMAN (Better Approach to Mobile Ad-Hoc Networking) – это протокол маршрутизации, специально разработанный для мобильных Ad Hoc сетей, который использует механизм OGM (Originator Message) для распространения информации о топологии и маршрутах.

BCR (Box Covering Based Routing) − это алгоритм маршрутизации, основанный на методе кластеризации сети. Он минимизирует сложность поиска маршрутов, представляя сеть в виде групп узлов (кластеров), что упрощает определение кратчайших путей.

BER (Bit Error Rate) − Мера, определяющая частоту ошибок при передаче битов в цифровых системах, выражаемая как отношение количества ошибочно принятых битов к общему количеству переданных битов
Box covering − метод, предложенный Сонгом, который используется для оценки и оптимизации кластеризации данных, позволяя эффективно покрывать объекты в пространстве минимальным числом кластеров, что упрощает анализ распределения данных и улучшает производительность алгоритмов.
BPSK (Binary Phase Shift Keying) − Двуфазная манипуляция по фазе: схема модуляции, в которой фаза несущей изменяется между двумя значениями для представления двоичных данных.
CEDAR (Core Extraction Distributed Ad Hoc Routing) − Протокол маршрутизации с извлечением основного набора узлов: гибридный протокол, использующий проактивные и реактивные методы для улучшения масштабируемости.
CIEA (Centre Including Eccentricity Algorithm) − Центрвключающий эксцентриситетный алгоритм: алгоритм, предназначенный для оптимального разделения WMN на кластеры и определения фрактальной размерности сети.
Client-oriented Traffic − Трафик, ориентированный на клиента: трафик в WMN, при котором данные передаются непосредственно клиентам сети.
CBB (Compact Box Burning) − Алгоритм кластеризации, в котором кластеры создаются путём случайного выбора новых узлов, находящихся в пределах заданного расстояния.
Dijkstra (Dijkstra Algorithm) − Алгоритм для нахождения кратчайшего пути в графе с неотрицательными весами рёбер, который последовательно выбирает узел с минимальным расстоянием и обновляет расстояния до его соседей.
DSDV (Destination-Sequenced Distance-Vector) − Протокол маршрутизации на основе векторного расстояния с последовательностью назначения: протокол маршрутизации для мобильных ad hoc сетей, который использует метод векторного расстояния и включает последовательности для предотвращения зацикливания, обеспечивая надежность маршрутов.
DSR (Dynamic Source Routing) − Протокол динамической маршрутизации: протокол для беспроводных сетей, позволяющий узлам динамически определять оптимальные маршруты.
DYMO (Dynamic MANET On-demand) − Динамический протокол маршрутизации MANET: улучшенная версия AODV с эффективным управлением маршрутами.
erfc (Complementary Error Function) − Комплементарная функция ошибки: математическая функция, используемая для описания вероятности ошибки в статистике и теории вероятностей, определяемая как 1 минус функция ошибки (erf), и применяемая в различных областях, включая обработку сигналов и теорию информации.
Escherichia coli − Модельная биологическая сеть, где узлы представляют собой гены, белки или другие молекулы, а рёбра обозначают взаимодействия или связи между ними.

Fractal Dimension, *D* (Фрактальная размерность): Мера, характеризующая сложность или степень фрагментации структуры, описывающая, как изменяется детализация объекта при изменении масштаба.
Feature-Based (FB) − метод, использующий характеристики (признаки) данных для анализа и классификации, позволяя выделять значимые аспекты, которые могут улучшить точность модели и упростить интерпретацию результатов.
Gateway-oriented Traffic (Трафик, ориентированный на шлюз): Трафик в WMN, при котором данные передаются через шлюз в Интернет.
GPA (Gateway Placement Algorithm) − Алгоритм размещения шлюзов: алгоритм, предназначенный для оптимального расположения шлюзов в WMN, что способствует максимизации пропускной способности сети.
GPP (Gateway Placement Problem) − Проблема размещения шлюзов: задача оптимизации расположения шлюзов в WMN для максимизации эффективности и производительности сети.
Graph coloring − метод, используемый для назначения «цветов» вершинам графа таким образом, чтобы ни две смежные вершины не имели одинакового цвета, что позволяет решать задачи, связанные с планированием, распределением ресурсов и оптимизацией.
Greedy coloring (GC) − метод кластеризации сети, который основывается на алгоритме раскраски графа.
Grid (Сеточная топология) – Структура сети, в которой узлы организованы в виде решетки, обеспечивая равномерное распределение соединений и упрощая маршрутизацию.
GW/Gws − Gateways (Шлюзы): Узлы в WMN, соединяющие сеть с внешними сетями, например, с Интернетом.
HSLS (Hazy-Sighted Link State) − Протокол маршрутизации с уменьшенным обновлением таблиц: гибридный протокол, уменьшающий частоту обновлений для отдаленных узлов.
HWMP (Hybrid Wireless Mesh Protocol) − Гибридный протокол маршрутизации в Mesh-сетях: сочетает проактивные и реактивные методы маршрутизации.
KNN (k-Nearest Neighbors) − Метод k-ближайших соседей: классификатор, который относит объект к тому классу, к которому принадлежит большинство его (k) ближайших соседей в пространстве признаков.
Likelihood-Based (LB) − метод, используемый для оценки параметров моделей, основываясь на вероятности наблюдаемых данных при заданных параметрах, позволяющий формулировать задачи оптимизации через максимизацию функции правдоподобия.
MANETs (Mobile Ad Hoc Networks) − Мобильные беспроводные сети Ad Hoc: сеть, состоящая из мобильных устройств, которые формируют связь без центральной инфраструктуры.
Matlab+Simulink − Программное обеспечение, используемое для реализации программных компонентов экспериментальной установки.
MEMB (Maximal Excluded Mass Burning) − алгоритм кластеризации, основанный на выборе центра кластера с максимальной исключённой массой.

MCs (Mesh Clients) − устройства, такие как компьютеры, мобильные телефоны и ноутбуки, которые подключаются к WMN для доступа к данным.
MI (Mutual Information) − мера, количественно оценивающая количество информации, которую одна случайная величина содержит о другой, позволяющая оценить зависимость и взаимосвязь между переменными в вероятностной модели.
MIMO (Multiple Input Multiple Output) − технология беспроводной связи, использующая несколько антенн как на передающей, так и на приёмной стороне для увеличения пропускной способности канала связи.
MLP (Multi-Layer Perceptron) − тип искусственной нейронной сети, состоящий из нескольких слоёв (входного, скрытых и выходного), где каждый нейрон в одном слое связан со всеми нейронами следующего слоя, что позволяет модели обучаться сложным нелинейным функциям и применять её для задач классификации и регрессии.
MRs/APs (Mesh Routers/Access Points) − узлы в WMN, ответственные за передачу данных к шлюзам и обратно.
NP (Non-deterministic Polynomial) − класс задач, решение которых можно проверить за полиномиальное время с использованием недетерминированной машины Тьюринга.
NS-3 (Network Simulator 3) − Открытая платформа для симуляции сетей, предназначенная для исследований в области сетевых технологий, позволяющая моделировать и анализировать различные протоколы, архитектуры и сценарии работы сетей.
OGM (Originator Message) – это сообщение, генерируемое исходным узлом для информирования других узлов о своём существовании и передаче информации о топологии сети в протоколе маршрутизации BATMAN.
OLSR (Optimized Link State Routing) – Проактивный протокол маршрутизации для беспроводных сетей, который поддерживает постоянное обновление информации о топологии и обеспечивает эффективную маршрутизацию данных.
OONP (Order-One Network Protocol) − Протокол маршрутизации Order-One Network: гибридный протокол, использующий проактивный подход для соседних узлов и реактивные методы для передачи данных.
QPSK (Quadrature Phase Shift Keying) − Квадратурная фазовая манипуляция: схема модуляции, в которой фаза несущей изменяется между четырьмя значениями, что позволяет передавать два бита данных за один символ.
Random Forest − метод машинного обучения, использующий ансамбль деревьев решений для классификации и регрессии, который обучает множество деревьев на случайных подмножествах данных и признаков, а затем объединяет их предсказания для повышения точности и устойчивости к переобучению.
Random Sequential (RS) − алгоритм кластеризации, в котором узлы сети случайным образом выбираются в качестве центров кластеров, после чего происходит их «сжигание» (burning).
Renyi Dimension, d_R (Размерность Реньи): Мера информационной размерности, зависящая от параметра α, определяемая через обобщённую энтропию Реньи.

RERR (Route Error) – Пакет сообщения об ошибке маршрута в AODV, используемый для уведомления о сбоях в соединении и обновления таблиц маршрутизации.
RREP (Route Reply) – Пакет ответа маршрута в AODV, отправляемый целевым узлом или узлом с маршрутом к целевому узлу, содержащий информацию о маршруте.
RREQ (Route Request) – Пакет запроса маршрута в AODV, отправляемый узлом для поиска пути к целевому узлу.
SNR (Signal-to-Noise Ratio) − Отношение сигнал/шум. Мера отношения мощности сигнала к мощности шума.
SPANs − Беспроводные сети на основе смартфонов (Smartphone Ad Hoc Networks). Сети, состоящие из мобильных устройств, таких как смартфоны, которые соединяются друг с другом без фиксированной инфраструктуры.
SVM (Support Vector Machines) − Опорные векторные машины: классификатор, который находит оптимальную гиперплоскость для разделения классов в пространстве признаков, максимизируя зазор между классами.
TBRPF (Topology Broadcast based on Reverse-Path Forwarding) − Протокол маршрутизации на основе передачи топологической информации по обратному пути: протокол, передающий топологическую информацию для оптимизации таблиц маршрутизации.
TORA (Temporally-Ordered Routing Algorithm) − Протокол маршрутизации с временным порядком: реактивный протокол, обеспечивающий направленную маршрутизацию.
Tsallis Dimension, d_T (Размерность Цалиса): Мера информационной размерности, зависящая от параметра q, определяемая через обобщённую энтропию Цалиса.
UV-flower − модельная сеть с топологией, напоминающей цветок, используемая для изучения сетевых алгоритмов и маршрутизации.
WMN (Wireless Mesh Network) − Беспроводная ячеистая сеть: тип сетевой инфраструктуры, состоящий из взаимосвязанных узлов, которые могут передавать данные между собой и в интернет через маршрутизаторы или шлюзы.
IERA (Information Entropy-based Routing Algorithm) − Алгоритм маршрутизации на основе информационной энтропии: алгоритм, использующий теорию информационной энтропии для оптимизации маршрутов передачи данных в WMN.
YGGDRASIL – децентрализованный протокол маршрутизации, использующий иерархическую структуру дерева для создания масштабируемых сетей без централизованного управления.
Zedboard − Аппаратная платформа, подключенная к ПК через Ethernet-кабель, используемая в экспериментальной установке.
ZHLS (Zone-based Hierarchical Link State) − Протокол маршрутизации с иерархическим подходом: гибридный протокол, использующий иерархический подход для эффективного распределения информации.

ZRP (Zone Routing Protocol) − Протокол маршрутизации с разделением на зоны: гибридный протокол, использующий проактивные методы для основной сети и реактивные для одноранговых частей сети.
16QAM (16-Quadrature Amplitude Modulation) − 16-квадратурная амплитудная модуляция: схема модуляции, которая сочетает в себе изменение фазы и амплитуды для представления 16 различных комбинаций битов.
64QAM (64-Quadrature Amplitude Modulation) − 64-квадратурная амплитудная модуляция: схема модуляции, в которой используется 64 различных комбинаций фазы и амплитуды для передачи данных.
8PSK (8-Phase Shift Keying) − 8-фазная манипуляция по фазе: схема модуляции, в которой фаза несущей изменяется между восьмью значениями, позволяя передавать три бита данных за один символ.

ВВЕДЕНИЕ

Актуальность работы. В настоящее время WMNs (от англ. Wireless Mesh Networks) представляют собой одну из наиболее значимых и динамично развивающихся технологий связи. Их популярность обусловлена такими ключевыми преимуществами, как простота развертывания, способность к самовосстановлению, самоорганизации (масштабная инвариантность), а также высокая экономичность. Благодаря этим характеристикам WMNs обеспечивают устойчивую и гибкую связь в условиях динамических и высоконагруженных сетевых сред.

Одним из ключевых преимуществ WMNs является их высокая отказоустойчивость. В отличие от традиционных беспроводных сетей, где выход из строя одного узла может привести к разрыву соединения, в ячеистых сетях информация передается через множество маршрутов, что позволяет автоматически перенаправлять трафик в случае сбоя одного или нескольких узлов. Такая особенность делает WMNs незаменимыми при построении надежных сетевых инфраструктур, особенно в условиях сложной городской застройки, промышленных предприятий, а также в ситуациях, требующих высокой степени автономности сети. WMNs обладают исключительной масштабируемостью. Новые узлы могут быть легко добавлены в сеть без необходимости проведения глобальных изменений в существующей инфраструктуре. Это делает ячеистые сети особенно привлекательными для использования в быстро развивающихся урбанизированных районах, на промышленных объектах и в системах автоматизации. Более того, гибкость данной технологии позволяет применять WMNs в самых разных областях, включая домашние сети для создания стабильного и равномерного покрытия Wi-Fi в больших помещениях, образовательные учреждения для организации беспроводного доступа в университетах и школах, здравоохранение для обеспечения связи в медицинских учреждениях, мониторинга пациентов и передачи медицинских данных, автоматизацию зданий в системах управления «умным домом», безопасности и климат-контроля, кризисное управление и спасательные операции для обеспечения оперативной связи в зонах бедствий и чрезвычайных ситуаций, а также военные приложения для организации защищенных коммуникаций на поле боя.

Архитектура WMNs строится на основе трехуровневой структуры, включающей три типа узлов. На первом уровне расположены узлы-шлюзы (Gateways, GW/GWs), которые соединены с физическим уровнем сети и обеспечивают подключение к проводным сетям или Интернету. Они служат основными точками выхода во внешнюю инфраструктуру. На втором уровне находятся квазистатические беспроводные ячеистые маршрутизаторы (Mesh Routers, MRs/APs), которые выполняют роль промежуточных узлов и обеспечивают передачу пакетов данных между клиентскими устройствами и сетевыми шлюзами. Эти маршрутизаторы обладают более высокой вычислительной мощностью, несколькими интерфейсами приема/передачи,

большей мощностью сигнала и, как правило, неограниченным источником питания. На третьем уровне располагаются ячеистые клиенты (Mesh Clients, MCs), к которым относятся различные беспроводные устройства, такие как настольные компьютеры, мобильные телефоны, ноутбуки, планшеты и другие гаджеты, подключающиеся к сети.

Передача данных в WMNs включает два типа трафика: трафик, ориентированный на шлюз, и трафик, ориентированный на клиента. В первом случае данные передаются от клиентских устройств через маршрутизаторы к узлу-шлюзу, который отправляет их в глобальную сеть или Интернет. Во втором случае передача данных осуществляется непосредственно между клиентами ячеистой сети по маршрутам с несколькими переходами, что позволяет минимизировать задержки и повысить эффективность обмена информацией. Таким образом, беспроводные ячеистые сети представляют собой современное решение для построения масштабируемых, отказоустойчивых и гибких сетевых инфраструктур, способных адаптироваться к меняющимся условиям эксплуатации.

Несмотря на значительный потенциал WMN, ряд существующих проблем затрудняет их оптимальное функционирование. Динамичная топология сети и постоянно меняющиеся условия канала затрудняют равномерное распределение узлов, что снижает быстродействие маршрутизации и может приводить к неравномерной загрузке. Традиционные алгоритмы маршрутизации испытывают трудности с адаптацией к этим изменениям, что отражается на пропускной способности и надежности связи. Влияние шума и мультипутевого распространения приводит к ошибкам приема сигналов и повышению BER, что негативно отражается на качестве демодуляции.

Для обеспечения надежной и адаптивной работы WMN в условиях динамично меняющейся топологии, переменных характеристик канала, высокого уровня шума и интерференции необходимо разработать инновационные методы, способные решать актуальные проблемы сети. В частности, требуется создание алгоритмов, обеспечивающих достоверный прием сигналов, равномерное распределение узлов для повышения быстродействия, а также выбор маршрутов с высокой пропускной способностью канала и минимизацией ошибок (BER). Такая потребность обусловлена тем, что традиционные методы маршрутизации не всегда справляются с адаптацией к постоянным изменениям условий передачи, что негативно сказывается на качестве связи. Таким образом, разработка методов автоматической классификации модуляции, кластеризации сети и алгоритма маршрутизации на основе информационной энтропии является необходимой для оптимизации маршрутизации и повышения общей эффективности работы WMN.

Цель работы заключается в оптимизации сети и маршрутизации в WMN посредством разработки инновационных алгоритмов, включая алгоритм классификации типов модуляции для достоверного приема сигналов, CIEA для повышения быстродействия и IERA для улучшения надежности приема и пропускной способности сети.

Задачи исследования:

1. Разработка алгоритма классификации типов модуляций сигналов, направленного на уменьшение ошибок при приеме и повышение производительности маршрутизационных алгоритмов.

2. Разработка алгоритма CIEA основанного на теории ‘box covering’ для оптимального разделения WMN на кластеры, расчета фрактальной размерности сети и обеспечения быстродействия алгоритма маршрутизации.

3. Создание алгоритма маршрутизации, основанного на теории информационной энтропии (IERA), который строит маршруты на основе максимального значения условной информации пути и пропускной способности в WMNs для повышения скорости передачи данных и уменьшения битовой ошибки.

Методы исследования:

В данной работе используются методы теории графов, информационной теории, численного моделирования, включая NS3 и программную реализацию на Python, а также экспериментальная передача сигнала через MIMO-антенны.

Метод ‘box covering’ из теории графов применяется для разбиения сложных сетей на кластеры, что позволяет анализировать их, оптимизируя процессы маршрутизации.

Методы информационной теории сосредоточены на анализе информации, передаваемой через канал связи, что позволяет оценивать пропускную способность и битовую ошибку в сети. Взаимная информация используется как ключевой показатель для достоверной классификации сигналов, обеспечивая устойчивость алгоритма к шуму.

Метод численного моделирования (NS3) применяется для тестирования существующих алгоритмов в контролируемых условиях. Он позволяет анализировать влияние параметров сети и сигналов, прогнозировать поведение системы и выявлять оптимальные настройки.

Кроме того, проведен реальный эксперимент, включающий передачу сигнала через MIMO-антенны, что является трудоемким процессом, требующим точной настройки аппаратуры и учета физических характеристик канала связи. Полученные реальные сигналы были обработаны с использованием предложенного алгоритма классификации модуляции, что позволило подтвердить его эффективность на практике. Предложенные методы были реализованы на Python, что обеспечило их верификацию в численном моделировании. Для этого использовались библиотеки Python, включая NetworkX для моделирования сетей, NumPy и SciPy для вычислений, а также инструменты визуализации, такие как Matplotlib и Seaborn.

Объекты исследования: беспроводная ячеистая сеть (WMN).

Основные положения, выносимые на защиту

1. Алгоритм классификации модуляции, основанный на теории взаимной информации, обеспечивает корректное определение типа модуляции сложных телекоммуникационных сигналов при низком уровне отношения сигнал/шум (SNR), более чем 5 дБ.

2. Центрвключающий эксцентриситетный алгоритм (CIEA), основанный на теории ‘box covering’ обеспечивает равномерное разделение беспроводной ячеистой сети (WMN) на кластеры по количеству маршрутизаторов (Routers), повышая быстродействие алгоритма маршрутизации на 20 микросекунд при 3280 узлах в сравнении с классическим алгоритмом Dijkstra.

3. Алгоритм маршрутизации (IERA), определяющий максимум условной информации как разность совместной и условной энтропии ансамбля (множества значений принятых сигналов с шумом), в сравнении с классическим алгоритмом Dijkstra обеспечивает повышение пропускной способности WMN на 5 Мбит/с по результатам численного моделирования для 100 узлов.

Научная новизна данной работы заключается в разработке инновационных алгоритмов оптимизации сети и маршрутизации в WMN, учитывающих фрактальные свойства сети, характеристики среды передачи и информационно-энтропийные критерии.

1. Разработанный алгоритм классификации типов модуляций впервые использует взаимную информацию для извлечения статистического признака модулированного сигнала, что позволяет учитывать взаимозависимости между переменными и повышает точность классификации на 50 % по сравнению с HOC-based алгоритмом.

2. Разработанный алгоритм CIEA, основанный на теории ‘box covering’, впервые использует концепцию эксцентриситета в своей реализации, что позволяет более оптимально покрыть сеть по сравнению с существующими алгоритмами кластеризации GC, OBCA, CBB и MEMB.

3. Разработанный алгоритм маршрутизации IERA впервые строит маршруты на основе максимального значения условной информации пути в WMN, что позволяет оптимизировать скорость передачи данных и снизить вероятность битовой ошибки по сравнению с существующими алгоритмами маршрутизации Dijkstra, ACO, OLSR и AODV.

Теоретическая и практическая значимость работы

Разработанные и реализованные алгоритмы, включая алгоритм классификации типов модуляции сложных сигналов, CIEA и IERA, обладают высокой теоретической и практической значимостью для WMNs. Их теоретическая значимость заключается в том, что они представляют собой инновационные подходы к оптимизации сети и маршрутизации, учитывая такие ключевые аспекты, как равномерное и оптимальное разделение сети на кластеры, вероятности ошибки канала и расположение шлюза. Практическая значимость подтверждается тем, что эти алгоритмы не только разработаны, но и успешно реализованы в сценариях WMNs, решая конкретные задачи, с которыми сталкиваются операторы сети и конечные пользователи.

Алгоритм CIEA реализует равномерную кластеризацию сети, что способствует улучшению быстродействия алгоритма маршрутизации. В свою очередь, IERA, основанный на теории информационной энтропии, реализует выбор маршрутов с максимальной условной информацией, что повышает

надежность передачи данных и снижает битовую ошибку. Доказательством практической применимости и эффективности этих алгоритмов является их успешное внедрение в патент на полезную модель, в рамках которого предложены кластерный маршрутизатор и маршрутизатор, основанный на информационной энтропии.

Личный вклад автора

Все основные результаты, выносимые на защиту, получены соискателем. Личный вклад автора включает проведение обзора литературы, написание и редактирование научных статей, что способствовало структурированию и обоснованию проводимого исследования; разработку и моделирование сетей в среде Python и NS3, что позволяет проводить детальный анализ работы беспроводных сетей и оценивать эффективность предложенных решений, включая задачи кластеризации сети; проведение расчетов фрактальных размерностей, необходимых для изучения топологических характеристик сети и анализа их влияния на процессы маршрутизации; реализацию существующих алгоритмов маршрутизации в Python и NS3 для проведения сравнительного анализа с разработанными методами; участие в разработке, программной реализации и тестировании предложенных алгоритмов, включая оптимизацию параметров и отладку кода; а также проведение эксперимента по получению реальных сигналов MIMO для автоматической классификации модуляции, что подтвердило практическую работоспособность метода. Теоретическая часть разрабатывалась под руководством научного руководителя, а расчеты, программная реализация и экспериментальная работа выполнялись автором, при этом постановка задач и обсуждение результатов осуществлялись совместно с научными консультантами.

Достоверность результатов

Достоверность научных выводов работы подтверждается моделированием, численным анализом, экспериментальным анализом, литературными исследованиями, а также обоснованными теоретическими моделями и принципами, применяемыми в области WMNs и маршрутизации. Эти модели включают в себя взаимную информацию, условную информацию и фрактальную размерность, что обеспечивает глубокое понимание структуры сетей и эффективности маршрутизации, а также способствует разработке более совершенных алгоритмов для оптимизации работы WMNs.

Апробация работы

По материалам диссертационной работы опубликовано 8 печатных работ.

Статьи с высоким импакт-фактором по базе данных Thomson Reuters или в изданиях, входящих в международную научную базу данных Scopus:

1. Ussipov N., Akhtanov S., Zhanabaev Z., Turlykozhayeva D., Karibayev B., Namazbayev T., & Tang X. (2024). Automatic modulation classification for MIMO system based on the mutual information feature extraction. *IEEE Access*.

2. Akhtanov S. N., Turlykozhayeva D. A., Ussipov N. M., Ibraimov M. K., Zhanabaev Z. Zh., 2022. Centre including eccentricity algorithm for complex networks. Electronics Letters, 58(7), 283-285.

3. Zhanabaev Z., Akhtanov S., Turlykozhayeva D., Ussipov N., & Ibraimov M., (2022). Cluster router based on eccentricity. Eurasian Physical Technical Journal, 19(3 (41)), 84-90.

4. Turlykozhayeva D. A., Akhtanov S. N., Ussipov, N. M., Akhmetali A., Bolysbay A., Shabdan, Y., 2023. Routing Algorithm for Software Defined Network Based on Boxcovering Algorithm. In 2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM) (pp. 1-5). IEEE.

5. Turlykozhayeva D. A., Akhtanov S. N., Baigaliyeva, A., Temesheva, S., Zhexebay D. M., Zaidyn M. & Skabylov A. (2024). EVALUATING ROUTING ALGORITHMS ACROSS DIFFERENT WIRELESS MESH NETWORK TOPOLOGIES USING NS-3 SIMULATOR. *Eurasian Physical Technical Journal*, *21*(2 (48)), 70-82.

6. Turlykozhayeva, D., Waldemar, W., Akhmetali, A., Ussipov, N., Temesheva, S., & Akhtanov, S. (2024). Single Gateway Placement in Wireless Mesh Networks. *Physical Sciences and Technology*, *11*(1-2).

7. Turlykozhayeva, D., Ussipov, N., Baigaliyeva, A., Temesheva, S., Bolysbay, A., Abrakhmatova, G., & Akhtanov, S. ROUTING METRIC AND PROTOCOL FOR WIRELESS MESH NETWORK BASED ON INFORMATION ENTROPY THEORY. Eurasian Physical Technical Journal. – 2023. – Т. 20. – №. 4 (46). – С. 90-98.

8. Turlykozhayeva, D. A., Akhtanov, S. N., Zhanabaev, Z. Z., Ussipov, N. M., & Akhmetali, A. (2025). A routing algorithm for wireless mesh network based on information entropy theory. *IET Communications*, *19*(1), e70011.

Связь темы диссертации с планами научных работ

Диссертационная работа выполнена в соответствии с планами фундаментальных научно-исследовательских работ КН МНВО РК «Грантовое финансирование научных исследований» по теме: «AP19674715 – Маршрутизация беспроводных ‘mesh’ сетей на основе ‘box-covering’ алгоритмов». В рамках данного проекта автор работает научным сотрудником по настоящее время, участвуя в разработке и внедрении алгоритмов оптимизации сети и маршрутизации для WMN.

Структура и объём диссертации

Диссертация состоит из введения, четырех разделов, заключения, списка литературы и содержит три приложения. Работа изложена на 119 страницах, иллюстрируется 60 рисунками, приведено 46 формулы, 7 таблиц, 125 наименований библиографических источников.

1. ОБЗОР СОВРЕМЕННЫХ МЕТОДОВ АНАЛИЗА И МАРШРУТИЗАЦИИ БЕСПРОВОДНЫХ СЕТЕЙ

Современные беспроводные сети и их алгоритмы маршрутизации находятся в постоянном развитии, стремясь к обеспечению эффективной передачи данных в изменяющихся условиях окружающей среды. Современные алгоритмы маршрутизации фокусируются на адаптивности к изменениям в сети, пропускной способности передачи данных, энергоэффективности, безопасности и многопутевой маршрутизации для повышения надежности и производительности сети. Основное внимание уделяется оптимизации маршрутов и управлению трафиком, что включает выбор наилучших путей передачи данных, учитывая пропускную способность каналов и загруженность узлов, а также равномерное распределение трафика и предотвращение перегрузок [1, 2, 99].

Параллельно с развитием алгоритмов маршрутизации ключевую роль играют алгоритмы классификации модуляций телекоммуникационных сигналов. Эти алгоритмы помогают правильно идентифицировать тип модуляции и применить соответствующие методы демодуляции, обеспечивая точное получение данных и снижая количество повторных передач и потерь. Глубокое понимание характеристик сигналов и точная идентификация модуляции позволяют разрабатывать более эффективные алгоритмы маршрутизации, которые лучше адаптируются к изменяющимся условиям сети. Это ведет к повышению пропускной способности, энергоэффективности и надежности передачи данных [1, 3, 5].

1.1 Алгоритмы классификации модуляций телеккомуникационных сигналов

Классификация модуляции телекоммуникационных сигналов является ключевой задачей в современных системах связи, позволяющей повысить эффективность и надежность передачи данных. В данном обзоре рассмотрены основные подходы и методы, используемые для классификации модуляции, а также их преимущества и недостатки. Модуляция является процессом изменения параметров несущего сигнала для передачи информации. Различные виды модуляции, такие как амплитудная (AM), частотная (FM) и фазовая (PM), используются в телекоммуникационных системах для различных целей [1, 2]. Классификация модуляции заключается в определении типа модуляции полученного сигнала, что важно для его последующей демодуляции и декодирования. Существуют два основных класса алгоритмов распознавания модуляции: основанные на вероятностном подходе (likelihood-based, LB) и методы, основанные на анализе признаков (feature-based, FB). Вероятностные методы (LB) используют вероятностные модели для оценки параметров модуляции и определения наиболее вероятной модуляции исходя из наблюдаемых сигналов. Примеры таких методов включают в себя байесовские сети и модели максимального правдоподобия, которые могут эффективно учитывать априорные знания о сигналах и шуме. Эти методы хорошо работают в условиях, когда статистические свойства сигналов и шума известны и

стабильны, но их точность может снижаться в условиях изменяющейся среды или при наличии неизвестных помех. Методы анализа признаков (FB) полагаются на извлечение ключевых признаков из сигналов, таких как амплитуда, частота и фаза, и последующее применение алгоритмов машинного обучения для классификации этих признаков. По сравнению с методами LB, методы анализа признаков (FB) более применимы на практике и более популярны в автоматической классификации модуляций (AMC) из-за их простоты реализации [3, 4]. Наиболее часто используемые признаки (FB) для автоматической классификации модуляций (AMC) можно разделить на пять типов: мгновенные временные признаки, признаки преобразования, статистические признаки, признаки формы созвездия и признаки пересечения нуля [3-5]. Мгновенные временные признаки охватывают амплитуду, фазу и частоту сигнала в каждом моменте времени. Эти признаки детализируют все изменения модулированного сигнала, позволяя точнее захватывать его динамические характеристики. Например, мгновенная амплитуда может указывать на уровень модуляции, мгновенная фаза − на фазовые изменения, а мгновенная частота − на частотные модификации. Анализ временных признаков помогает выявлять вариации, которые могут быть критичными для классификации модуляции [6, 7]. Признаки преобразования извлекаются посредством применения различных преобразований, таких как преобразование Фурье [6, 8] или вейвлет-преобразование [9, 10]. Эти преобразования переводят сигнал в частотный домен или другие представления, которые могут раскрыть скрытые особенности. Признаки включают спектральные плотности, частотные компоненты и их распределение. Предварительная обработка сигналов, такая как сглаживание, нормализация и медианная фильтрация, позволяет улучшить точность анализа, убирая шум и выделяя важные частотные компоненты [11, 12]. Статистические признаки анализируют сигнал с помощью высших порядков моментов (HOMs), высших порядков кумулянтов (HOCs), циклических кумулянтов высших порядков (HOCCs) и циклостатики [13-15]. Высшие порядки моментов и кумулянты предоставляют информацию о распределении и зависимости между значениями сигнала. Циклические кумулянты и циклостатика помогают выявить периодические и циклические особенности, которые важны для различения типов модуляций, особенно в условиях шума [16, 17]. Признаки формы созвездия анализируют распределение точек на диаграмме созвездия, которая отображает символы модулированного сигнала. Методы включают подсчет уровней точек созвездия и их распределение, а также сравнение с эталонными точками созвездия. Это позволяет выявить специфические геометрические особенности модуляции, такие как формы и плотности точек, которые помогают различать разные схемы модуляции. Признаки пересечения нуля относятся к анализу точек, в которых сигнал пересекает ось времени. Эти признаки помогают выявлять частотные и временные характеристики сигнала, которые могут быть полезны для определения его модуляции. Например, частота пересечений нуля может быть связана с частотой сигнала, а распределение этих пересечений может дать

информацию о периодичности и структуре сигнала [18, 19]. Каждый из этих типов признаков предоставляет уникальную и полезную информацию о сигналах, улучшая точность и надежность автоматической классификации модуляций. Сложное сочетание этих признаков помогает более точно идентифицировать и различать различные схемы модуляции, обеспечивая высокое качество связи и обработки данных.

Помимо традиционных методов, таких как вероятностный подход (likelihood-based, LB) и методы, основанные на анализе признаков (feature-based, FB), в последние годы всё большее внимание уделяется алгоритмам классификации модуляции, основанным на машинном и глубоком обучении [20, 21, 23, 25]. Эти методы обладают способностью обучаться на больших наборах данных, автоматически извлекая скрытые признаки сигналов, что позволяет достигать высокой точности классификации. Благодаря использованию сложных нейронных сетей и адаптивных моделей, методы машинного и глубокого обучения обеспечивают гибкость и эффективность в условиях сложных и изменяющихся характеристик сигналов. Многочисленные методы автоматической классификации модуляции (AMC), основанные на машинном и глубоком обучении [20-26], были разработаны для классификации сигналов как SISO, так и MIMO. Как показано в недавних обзорах [27-31], эти методы имеют значительный потенциал для достижения близкой к оптимальной производительности при приемлемом уровне вычислительной сложности. Некоторые из этих методов используют инновационные технологии для подавления помех в сложных MIMO-сигналах и повышения точности классификации. Например, в работе [32] предложен метод автоматической классификации модуляции (AMC) для MIMO-систем, использующий сочетание сверточных нейронных сетей (CNN) и нулевого принуждения (Zero-Forcing, ZF) для подавления межканальной интерференции. На первом этапе применяется ZF-эквализация, которая уменьшает влияние наложения сигналов с разных антенн путем умножения принятого сигнала на псевдообратную матрицу канала. Это позволяет компенсировать перекрестные помехи между передающими и принимающими антеннами. Однако такой метод эквализации не учитывает шум, который при низком уровне сигнала может значительно усиливаться. После эквализации сигнал представляется в виде двумерных изображений, содержащих временные и частотные характеристики, и передается в CNN. Архитектура сети включает сверточные слои для автоматического извлечения признаков, пулинговые слои для снижения размерности и повышения устойчивости к шумам, а также полносвязные слои, выполняющие классификацию модуляции. В работе [33] предложено усовершенствование данного метода с использованием переноса обучения (Transfer Learning), что позволяет эффективно классифицировать модуляцию даже при ограниченном количестве размеченных данных. Как и в [32], на первом этапе применяется ZF-эквализация, а затем обработанный сигнал преобразуется в изображение и подается в CNN. В отличие от предыдущего подхода, здесь используется заранее обученная нейросетевая модель, которая адаптируется к новым данным с

помощью fine-tuning. Это позволяет сохранить полезные признаки, извлеченные на предварительном этапе обучения, и настраивать только финальные слои под новую выборку данных. Такой подход сокращает потребность в больших объемах размеченных примеров, повышает устойчивость классификации при изменении условий канала и ускоряет процесс обучения. Несмотря на эффективность предложенных методов [32, 33] , их применение в реальных MIMO-системах остается ограниченным. ZF-эквализация подавляет межканальные помехи, но одновременно усиливает шум, особенно при низком уровне сигнала. Кроме того, она не учитывает нелинейные искажения, возникающие из-за аппаратных особенностей передатчика и приемника. Это приводит к снижению точности классификации в сложных условиях связи. Таким образом, необходимы новые методы, не требующие устранения помех сигналов MIMO и большого объема обучающих данных, но обеспечивающие устойчивость к неопределенности шума и пространственной корреляции. В данной работе решается эта проблема путем применения классификатора, основанного на извлечении признаков взаимной информации, что позволяет учитывать взаимозависимости между переменными и добиваться высокой точности классификации даже при низком SNR.

В работе [116] предложен метод классификации модуляции, основанный на использовании кумулянт высокого порядка (High-Order Cumulants, HOC) для извлечения признаков сигнала и последующей классификации модуляции в кооперативных ретрансляционных сетях с дифференциальным пространственно-временным блоковым кодированием (D-STBC). Принцип работы метода включает несколько этапов. На первом этапе выполняется предобработка принятого сигнала, которая включает дифференциальное декодирование, учитывающее особенности D-STBC, и центрирование для устранения средних значений, которые могут повлиять на точность вычислений. На втором этапе производится извлечение признаков с помощью вычисления кумулянт четвертого и шестого порядков. Кумулянты являются статистическими характеристиками сигнала, отражающими его распределение и отклонение от гауссовой модели. Они обладают инвариантностью к аддитивному гауссовому шуму (AWGN), что делает их надежными для использования в системах классификации модуляции. Вычисление кумулянт выполняется на основе математического ожидания комбинаций значений сигнала и их комплексных сопряжений, что позволяет выявлять характерные различия между различными схемами модуляции. На третьем этапе формируется признаковый вектор, состоящий из нормализованных значений кумулянт, что обеспечивает инвариантность метода к изменениям мощности сигнала. Затем, на заключительном этапе, выполняется классификация модуляции с использованием порогового метода или машинного обучения. На основе анализа полученных признаков алгоритм определяет тип модуляции, например, BPSK, QPSK или 16-QAM. Метод обладает рядом преимуществ, включая устойчивость к гауссовому шуму за счет использования кумулянт высокого порядка, относительно низкую вычислительную сложность по сравнению с методами

глубокого обучения. Однако существуют проблемы, ограничивающие применение метода [116]. Чувствительность к нелинейным искажениям в канале, таким как фазовые сдвиги и нелинейности усилителя мощности, снижает точность классификации. При низком уровне SNR эффективность метода значительно падает, поскольку шумовые искажения вносят ошибки в расчет кумулянт, что затрудняет различение типов модуляции. Для корректного вычисления статистических характеристик требуется достаточно большой объем выборок, что в реальных условиях связи может быть ограничением. Эти недостатки снижают надежность метода при сложных условиях передачи сигнала и требуют дальнейшей оптимизации алгоритма для повышения его устойчивости к помехам и нелинейным искажениям.

В дополнение к вышеописанным методам, основанным на машинном и глубоком обучении, значительный интерес представляет использование информационной энтропии для классификации коммуникационных сигналов. В работах [34-37] предложены алгоритмы классификации модуляций, применяющие энтропийные признаки, что способствует повышению точности классификации в условиях сложных и изменяющихся характеристик сигналов.

В работе [34] разработан метод классификации модуляции, основанный на комбинации энтропийных признаков и ансамблевого обучения. На первом этапе вычисляются характеристики сигнала, такие как энтропия Шеннона, энтропия Реньи и относительная энтропия, которые отражают степень неопределенности сигнала и помогают выявить ключевые различия между различными схемами модуляции. Затем эти признаки передаются на вход ансамблевого классификатора, включающего несколько базовых алгоритмов, таких как случайный лес (Random Forest) и градиентный бустинг. Использование ансамблевых методов позволяет повысить устойчивость классификации за счет комбинирования решений нескольких моделей. В работе [35] предложен метод, использующий энтропийные признаки в сочетании с алгоритмом случайного леса. Извлекаются различные энтропийные показатели, включая кросс-энтропию, энтропию Цаллиса и спектральную энтропию, которые затем подаются на вход случайного леса. Случайный лес строит множество деревьев решений и усредняет их предсказания, что повышает устойчивость модели к шумам и вариациям сигнала. Этот метод отличается хорошей обобщающей способностью и подходит для работы в условиях изменяющихся каналов связи.

В работе [36] предложен гибридный метод распознавания сигналов, основанный на использовании нескольких энтропийных признаков в сочетании с теорией свидетельств Демпстера-Шафера (DS evidence theory). На первом этапе вычисляются основные энтропийные характеристики сигнала, затем они комбинируются с помощью теории Демпстера-Шафера, которая позволяет учитывать неопределенность и неполноту данных. Такой подход особенно полезен в условиях низкого отношения сигнал/шум (SNR), так как он снижает влияние ошибок классификации за счет взвешенной обработки показателей надежности. Несмотря на эффективность энтропийных методов [38-40], они имеют ограничения, связанные с тем, что энтропия измеряет лишь

неопределенность отдельных переменных, не учитывая их взаимосвязь. В отличие от этого, взаимная информация (Mutual Information, MI) позволяет измерять степень зависимости между двумя или более переменными, что делает ее более информативной для классификации сложных сигналов.

В данной работе предлагается алгоритм AMC для MIMO-систем, основанный на извлечении признаков с использованием взаимной информации. Ключевым элементом метода является построение диаграмм IQ-созвездий MIMO-сигналов для извлечения признаков, что представляет собой уникальный подход, ранее не исследованный в научной литературе.

После извлечения признаков для определения класса модулированных сигналов в автоматической классификации модуляции (AMC) применяются различные типы классификаторов, включая искусственные нейронные сети (ANN), k-ближайших соседей (KNN) и опорные векторные машины (SVM) [40].

Искусственные нейронные сети (ANN) состоят из множества взаимосвязанных нейронов, организованных в несколько слоев (входной, скрытые и выходной). Эти сети обучаются на большом объеме данных, что позволяет им выявлять сложные зависимости между входными признаками и выходными классами. ANN обладают высокой гибкостью и могут адаптироваться к различным типам данных, что делает их особенно полезными для классификации сложных и нелинейных сигналов. ANN могут использовать различные методы обучения. Многослойные перцептроны (MLP) обучаются с использованием алгоритмов обратного распространения ошибки и хорошо подходят для классификации, так как могут моделировать сложные зависимости. Нейронные сети с радиальной базисной функцией (RBF) отличаются более высокой скоростью обучения и могут быть более эффективными в некоторых задачах. Самоорганизующиеся карты (SOM) являются примером сети без контроля, которые могут автоматически определять структуру данных и адаптивно выбирать количество нейронов [41, 42].

Метод k-ближайших соседей (KNN) классифицирует объекты на основе их близости к другим объектам в пространстве признаков. Этот метод не требует обучения как такового: все вычисления происходят на этапе классификации. KNN определяет класс нового объекта, анализируя классы k ближайших соседей и присваивая объекту тот класс, который наиболее часто встречается среди них. KNN прост в реализации и особенно эффективен на небольших наборах данных. Однако его эффективность может снижаться на больших объемах данных из-за высокой вычислительной нагрузки. Важно правильно выбрать значение k, так как оно существенно влияет на точность классификации [43].

Опорные векторные машины (SVM) классифицируют данные, находя гиперплоскость, которая максимально разделяет классы в пространстве признаков. SVM используют ядровые функции для обработки нелинейно разделяемых данных, преобразуя их в более высокоразмерное пространство, где классы могут быть разделены линейно. Основной принцип SVM заключается в максимизации зазора между классами, что способствует высокой степени общности и предотвращает переобучение. Для задач классификации с

несколькими классами SVM могут использовать бинарные классификаторы (BSVM), которые последовательно разделяют каждый класс против всех остальных, или многоклассовые SVM (MSVM), такие как направленный ациклический графический SVM (DAGSVM). SVM особенно выделяются среди других классификаторов благодаря своей способности обеспечивать высокую точность классификации даже при низком уровне сигнала-шум и сложных данных. Они демонстрируют отличные результаты в разнообразных задачах АМС и устойчивы к проблемам переобучения, что делает их одним из наиболее мощных и надежных инструментов в данной области [44-46].

В данной работе для классификации модуляций использовался классификатор SVM, который выделяется своим главным преимуществом − способностью обеспечивать высокую точность и надежность даже при работе с шумными и сложными сигналами. Благодаря своей стратегии максимизации зазора между классами SVM эффективно предотвращает переобучение и демонстрирует отличные результаты в задачах автоматической классификации модуляций.

1.2 Методы кластеризации и фрактального анализа в сложных сетях

Наука о сетях − это быстро развивающаяся область исследований, применяемая для описания различных систем в обществе. Важно отметить, что многие современные сети обладают сложными топологическими особенностями. Методы фрактального анализа с масштабной инвариантностью широко используются для оценки топологии сложных сетей. Этот подход является ценным инструментом для изучения топологических свойств сетей, позволяя выявлять самоподобие и структурную сложность на разных уровнях масштаба. Фрактальный анализ нашёл широкое применение при исследовании различных типов сетей, включая социальные сети, Интернет, беспроводные мобильные сети и биологические сети. Фрактальные свойства были обнаружены в реальных сетях, таких как социальные сети, Интернет и мобильные сети [47].

Применение фрактального анализа к беспроводным сложным сетям полезно для понимания и описания их структуры, динамики и поведения на различных масштабах. Фрактальная размерность позволяет охарактеризовать сложную иерархическую структуру таких сетей, как сети сенсоров или беспроводные сети связи. Использование фрактальных методов помогает выявить топологические особенности сети на разных масштабах, что критично для оптимизации распределения ресурсов и повышения её эффективности. Это включает оптимизацию размещения узлов и маршрутизацию данных с учётом фрактальных характеристик сети. Изучение фрактальных свойств сети даёт понимание её масштабируемости и способности эффективно передавать данные на различных уровнях детализации, что важно для проектирования и оптимизации беспроводных протоколов передачи данных. Фрактальные методы также помогают выявить уязвимости и прогнозировать сбои в беспроводных сетях, что способствует разработке более устойчивых и надёжных сетевых архитектур [47, 48].

Для расчёта фрактальной размерности (D) реальных сетей Сонг предложил метод 'box covering', представляющий собой алгоритм кластеризации сети. Основная цель этого алгоритма − покрыть сеть минимальным числом кластеров N_b при заданном размере кластера l_b.. Если количество кластеров N_b (l_b) (в зависимости от l_b.) следует степенному закону, фрактальная размерность сложной сети D определяется следующим образом:

$$N_b(l_b) \sim l_b^{-D} \quad (1.1)$$

Наиболее распространенными алгоритмами кластеризации являются являются ‘greedy coloring’ (GC), ‘random sequential’ (RS), ‘compact-box-burning’ (CBB) и ‘maximum-excluded-mass burning’ (MEMB) [47, 49]. На рисунке 1.1 ниже показан принцип кластеризации сети с выявлением фрактальности на графике.

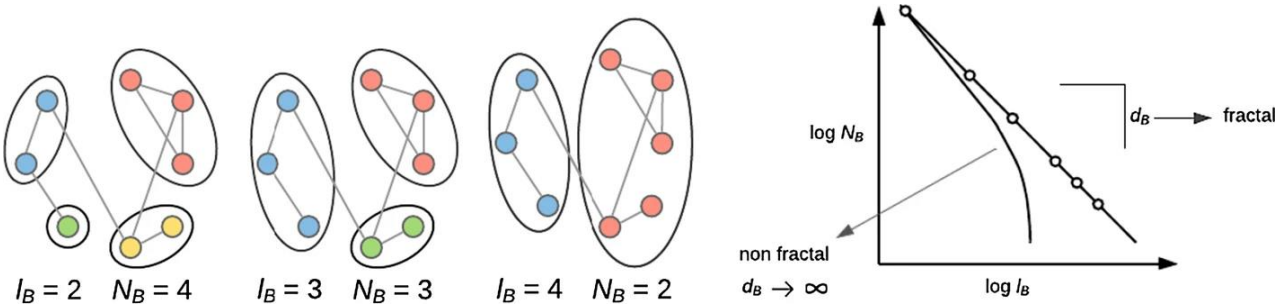

Рисунок 1.1 − Выявление фрактальности путем кластеризации сети, где N_B-количество кластеров, l_B-размер кластера, d_B- фрактальная размерность [47]

В следующем шаге рассмотрим существующие алгоритмы кластеризации. Для ясности размер кластера обозначается как l_b, а радиус кластера − как r_b, где l_b=2r_b+1. Кластеры определяются следующим образом: для каждого кластера расстояние между любыми двумя узлами в его пределах должно быть меньше или равно l_b. В соответствии с этим определением, процесс формирования кластеров требует нахождения кратчайшего пути между двумя узлами для определения их принадлежности к определённому кластеру. Для удобства и во избежание многократного вычисления попарных кратчайших расстояний d (vi, vj) между вершинами v_i, и v_j, принадлежащими множеству вершин графа V, используется подход, при котором эти расстояния сначала вычисляются и сохраняются для всех пар вершин. В данной работе термины ‘граф’ и ’сеть’ используются взаимозаменяемо [47, 48].

Рассмотрим алгоритм кластеризации «жадная раскраска» (greedy coloring, GC), предложенный Сонгом [47, 50] в одной из его первых работ. Проблему кластеризации сети можно сопоставить с задачей «раскраски графа» (graph coloring), что позволяет использовать алгоритм раскраски для решения задачи кластеризации. Метод заключается в создании вспомогательного графа G' на основе исходной сети G, который включает те же вершины и рёбра. В этом вспомогательном графе две вершины соединяются рёбрами только в том случае, если расстояние между ними в исходной сети превышает заданный порог l_b. Процесс построения вспомогательного графа проиллюстрирован на рисунке 1.2, а его краткий код представлен в Алгоритме 1.

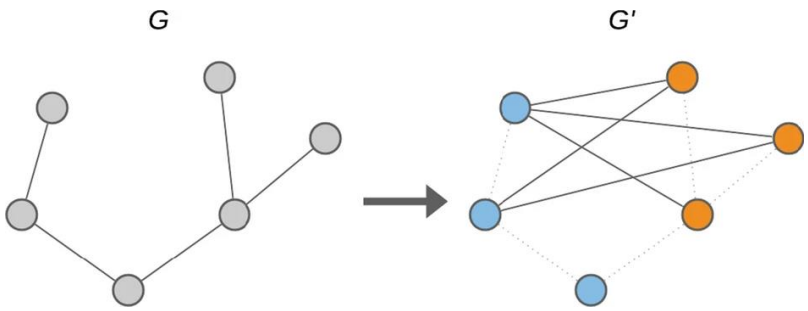

Рисунок 1.2 – Пример построения вспомогательного графа (l_b=2). Узлы с одинаковым цветом в сети G' образуют кластер [50]

После построения вспомогательного графа для заданного значения l_b выполняется его окрашивание. В этом процессе каждой вершине вспомогательного графа присваивается цвет так, чтобы соседние вершины не имели одинакового цвета, и использовалось минимальное количество цветов. Эта задача эквивалентна кластеризации сети методом «box covering» исходной сети G, поскольку назначенные цвета во вспомогательном графе соответствуют кластерам в исходной сети. Минимальное количество цветов, необходимое для окрашивания вершин вспомогательного графа, совпадает с минимальным количеством кластеров, необходимых для полного покрытия исходного графа.

Algorithm 1 Auxiliary graph

Require: $G = (V, E)$, l_B,
  $V' = V$ # the nodes of G' are the same
  $E' = []$ # initialize E' as an empty list
  for v_1 in V' **do**
    for v_2 in V' **do**
      if $d(v_1, v_2) > l_B$ in G **then**
        add (v_1, v_2) edge to E'
    end for
  end for
  return $G' = (V', E')$ auxiliary (simple) graph

Алгоритм ‘graph coloring’ относится к классу NP-задач (Non-deterministic Polynomial) и включает два основных этапа: упорядочивание узлов и повторение действий. На первом этапе узлы графа упорядочиваются, а затем каждому узлу присваивается наименьший возможный идентификатор цвета. Схематическое описание работы алгоритма представлено в Алгоритме 2 [47, 51]. Количество цветов, необходимое для окраски вершин вспомогательного графа, соответствует минимальному числу кластеров, необходимых для полного покрытия исходного графа. Таким образом, этот алгоритм позволяет эффективно группировать узлы в кластеры, минимизируя при этом количество используемых цветов.

Algorithm 2 Greedy coloring

Require: $G = (V, E)$, l_B, and strategy s
  $G' =$ auxiliary_graph(G, l_B) # see Algorithm 1
  $V^{sorted} =$ order(V, s)
  for v in V^{sorted} **do**
    assign lowest allowed color ID to v in G'
  end for
  return (node, color) pairs

Алгоритм ‘Random Sequential’, предложенный Кимом, использует метод случайного разделения сети на кластеры для оптимизации различных задач, включая анализ сети и маршрутизацию. Основной принцип этого алгоритма заключается в процессе, называемом «сжигание» (burning). Сначала выбирается случайный центральный узел, от которого начинается формирование кластеров. После назначения узлам определённых кластеров, они «сжигаются» (Рисунок 1.3).

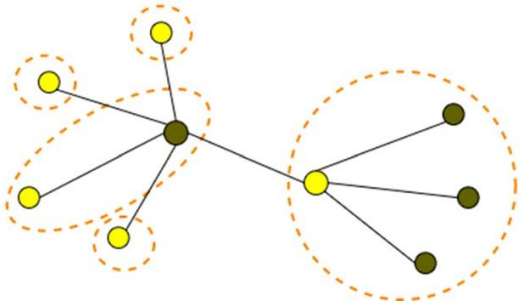

Рисунок 1.3 − Пример покрытия RS алгоритмом (r_B = 1). Центральные узлы кластеров окрашены в желтый цвет [47]

На каждом шаге алгоритма случайным образом выбирается центральный узел C, который еще не был «сожжен». Затем отбираются узлы, находящиеся на расстоянии не более R_B от этого центрального узла и также не «сожженные». Эти недавно «сожженные» узлы объединяются в новый кластер. Важно подчеркнуть, что выбор центрального узла происходит из всего множества узлов V, что может повлиять на результаты масштабирования N_B на графах [47, 51, 52].

Algorithm 3 Random sequential (RS)

Require: $G = (V, E)$, r_B
  $U = V$ # uncovered nodes
  $boxes = [\,]$ # list of the created boxes that will be returned
  while $U \neq \emptyset$ **do**
    $c =$ random_choice(U) # randomly chosen uncovered center node
    append $U \cap B(c, r_B)$ to boxes
    $U = U \setminus B(c, r_B)$
  end while
  return $boxes$

Алгоритм Compact Box Burning (CBB), использующий концепцию «сжигания», был предложен Сонгом вместе с алгоритмами GC и MEMB. Основная идея CBB заключается в том, что кластеры формируются путем случайного выбора новых узлов из набора кандидатов C, который включает все узлы, находящиеся на расстоянии не более l_B от любого узла, уже включенного в набор кандидатов. В отличие от других подходов, CBB стремится обеспечить

компактность формируемых кластеров, что достигается за счет ограниченного расстояния l_B.

Algorithm 4 Compact Box Burning (CBB)

Require: $G = (V, E)$, l_B
  $U = V$ # for uncovered nodes
  $boxes = [\,]$ # list for assigned boxes
  while $U \neq \emptyset$ **do**
    $box = [\,]$ # list for current box
    $C = U$ # candidates for box members
    while $C \neq \emptyset$ **do**
      $p = \text{random}(C)$
      append p to box
      $C = C \setminus \{d(p, v_i) > l_B \mid v_i \in V\}$ # remove p incompatible nodes from C
    end while
    append box to $boxes$
    $U = U \setminus box$
  end while
  return $boxes$

Алгоритм Maximal Excluded Mass Burning (MEMB), также разработанный Сонгом, представляет собой алгоритм кластеризации, сосредоточенный на создании кластеров с максимальной исключённой массой. В отличие от других методов, использующих размер кластера l_B, в этом алгоритме применяется понятие радиуса r_B и центрированных кластеров, где каждый кластер имеет специальный узел − центр. Кластеры формируются таким образом, чтобы каждый узел находился на расстоянии не более r_B от центрального узла. Алгоритм MEMB работает следующим образом: центр выбирается на основе максимальной исключённой массы, то есть количества непокрытых узлов, находящихся на расстоянии не более r_B от потенциального центра (Рисуник 1.4). После выбора центра все непокрытые узлы в этом радиусе «покрываются», но еще не назначаются к конкретным кластерам. Алгоритм MEMB представлен в Алгоритме 9 [47, 52, 53].

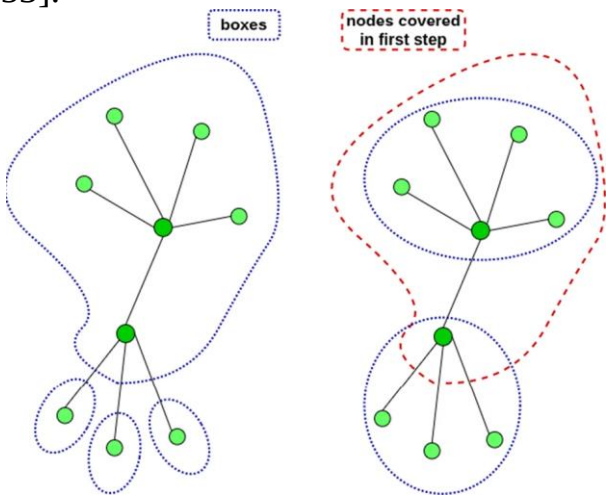

Рисунок 1.4 − Иллюстрация работы алгоритма Maximal Excluded Mass Burning (MEMB) [47]

Algorithm 5 Maximal Excluded Mass Burning (MEMB)

Require: $G = (V, E),\ r_B$
  $U = V$ # uncovered nodes
  $C = \{\}$ # center nodes
  while $U \neq \emptyset$ **do**
    calculate m_i excluded mass for non-center nodes
    let p be the node for which m_p is maximal
    $C = C \cup \{p\}$
    $U = U \setminus B(p, r_B)$
  end while
  let c_{dist} contain the distance to the closest center for every node
  let C' be the list of non-center nodes sorted by their c_{dist} value
  for $n\ in\ C'$ **do**
    $m =$ find a random neighbor of n whose c_{dist} is smaller
    allocate n to the center node which is the center of m
  end for
  return the formed boxes that are identified by their center nodes

Однако стоит отметить, что вышеописанные методы имеют некоторые недостатки. Алгоритмы, такие как GC, RS и CBB, в своей реализации включают случайный выбор центральных узлов и покрывают сеть на расстоянии r_b, что в свою очередь, влияет на кластеризацию сети. Например, на рисунке 1.5 вероятность выбора крайних узлов равна 3/4, тогда как вероятность выбора центрального узла равна 1/4, таким образом, в большинстве случаев крайние узлы остаются в кластерах с одним узлом (Рисунок 1.5).

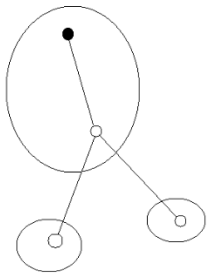

Рисунок 1.5 – Пример кластеризации сети с использованием алгоритмов, основанных на случайном выборе центрального узла при радиусе кластера r_b =1

При делении сети, представленной на рисунке 1.5, на кластеры с помощью алгоритма MEMB выбирается центральный узел с максимальной исключённой массой, что позволяет охватить всю сеть одним кластером. Однако сеть, показанную на рисунке 1.6, алгоритм MEMB не способен оптимально разделить на кластеры, что приводит к образованию максимального числа кластеров с одним узлом.

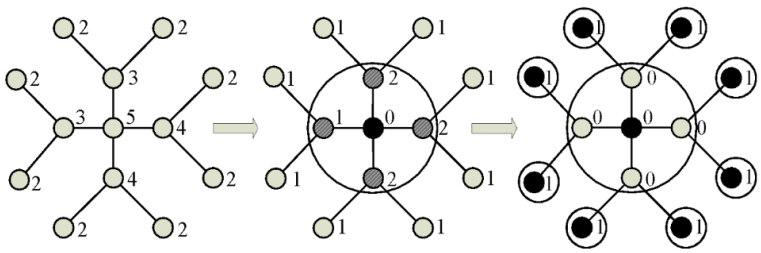

Рисунок 1.6. Реализация метода MEMB (r_b =1)

Если центральным узлом выбирается центральная точка сети, то для полного охвата потребуется большее количество кластеров [47, 54, 55].

1.3 Теоретические основы информационной энтропии

Энтропия в теории информации представляет собой меру неопределенности или хаоса в системе. Понятием энтропии впервые занимался Клод Шеннон в середине XX века, и с тех пор оно стало основополагающим в различных областях науки и технологии, включая связь, криптографию, обработку данных и машинное обучение. Энтропия измеряет количество информации, необходимой для точного описания состояния системы, и играет критическую роль в оценке эффективности передачи и хранения данных [59]. Основная формула, определяющая энтропию Шеннона, выглядит следующим образом:

$$H(X) = -\sum_i P(x_i)\, log_2\, P(x_i), \quad (1.2)$$

где X случайная величина, $P(x_i)$ − вероятность i -го события или значения x_i.

Совместная энтропия используется для измерения общей неопределенности двух случайных величин. Совместная энтропия $H(X,Y)$ определяется как:

$$H(X,Y) = -\sum_{i=1}^{N}\sum_{j=1}^{M} P\left(y_j, x_i\right) log_2(P(x_i)P\left(y_j|x_i\right)) \quad (1.3)$$

где $P\left(y_j, x_i\right)$ − совместная вероятность для значений x_i и y_j, $P(x_i)-$ маргинальная вероятность для события X, то есть вероятность наступления события x_i, $P(y_i|x_i)$ − условная вероятность для события y_i, при условии, что произошло событие x_i, N − количество возможных значений для X, M − количество возможных значений для Y.

Условная энтропия $H(X/Y)$ характеризует неопределенность в случайной величине X, при условии, что известно значение случайной величины Y. Это величина, которая описывает, насколько неопределённым остаётся распределение вероятностей X, если уже известна информация о Y. Условная энтропия позволяет количественно оценить степень, в которой знание одной случайной величины снижает неопределенность в другой. Условная энтропия определяется как:

$$H(X|Y) = -\sum_{i=1}^{N} P(x_i)\sum_{i=1}^{M} P(y_i|x_i)\, log_2\, P(y_i|x_i) \quad (1.4)$$

где $P(x_i) -$ маргинальная вероятность для события X, то есть вероятность наступления события x_i, $P(y_i|x_i)$ − условная вероятность для события y_i, при условии, что произошло событие x_i, N − количество возможных значений для X, M − количество возможных значений для Y [56-58].

Взаимная информация между двумя случайными величинами X и Y измеряет количество информации, которую одна величина предоставляет о другой. Она отражает степень взаимной зависимости между этими величинами, то есть, насколько знание одной из величин уменьшает неопределенность по поводу другой. Взаимная информация показывает, насколько лучше можно предсказать одну величину, зная другую. Взаимную информацию можно выразить через энтропии этих величин и их совместную энтропию:

$$I(X;Y) = H(X) + H(Y) - H(X,Y), \quad (1.5)$$

где $H(X)-$ энтропия случайной величины X, которая измеряет неопределенность или хаос в распределении значений X, $H(Y)$ – энтропия случайной величины Y, $H(X,Y)$ – совместная энтропия, которая измеряет неопределенность системы, состоящую из обеих величин X и Y, с учетом их взаимозависимости [59, 60].

1.4 Информационные размерности Реньи и Цаллиса

Энтропия Реньи, введенная математиком Альфредом Реньи, представляет собой обобщение классической энтропии Шеннона и является параметрическим семейством энтропий, зависящим от параметра α. Она обладает гибкостью, позволяющей измерять неопределенность системы в различных контекстах. Формула энтропии Реньи порядка α выглядит следующим образом:

$$H_\alpha(X) = \frac{1}{1-\alpha}\log\sum_{k=1}^{n} p_k{}^\alpha \tag{1.6}$$

где α ∈ (0, 1) ∪ (1, ∞), а вероятность k-го события p_k. Соответствующие пределы для α ∈ {0, 1, ∞} определяются следующим образом. При α = 1 предел энтропии Реньи соответствует энтропии Шеннона, при α = 0 энтропия Реньи измеряет логарифм числа возможных исходов (энтропия Хартли) [61].

Энтропия Цаллиса, предложенная физиком Константино Цаллисом, является обобщением энтропии Больцмана-Гиббса и предназначена для описания неэкстенсивных систем, где традиционные методы термодинамики не работают. Она определяется параметром q, который регулирует степень неэкстенсивности:

$$S_q = k\frac{1-\sum_{i=1}^{N} p_i^q}{q-1}, \tag{1.7}$$

где N общее количество элементов набора вероятностей, k постоянная, а p_i – вероятность i-го события. Когда порядок q равен 1, энтропия Цаллиса соответствует энтропии Шеннона. Порядок α по энтропии Реньи обычно сравнивают с порядком q по энтропии Цаллиса для анализа их устойчивости в быстро и медленно меняющихся ситуациях [61, 62].

Информационная размерность сети впервые использовалась для оценки информационной нагрузки и измерения странных аттракторов. Впервые Вэй определил информационную размерность на основе информационной энтропии и кластерного алгоритма. Информационная энтропия, содержащаяся в сложной сети, может быть рассчитана следующим образом:

$$H = -\sum_{i=1}^{N_B} p_i \ln p_i, \tag{1.8}$$

где p_i представляет вероятность узлов в i-м кластере, которую можно определить следующим образом:

$$p_i = \frac{n_i}{n}, \tag{1.9}$$

где n_i количество узлов в в i -м кластере, а n – общее количество узлов в сети. Информационная размерность сети определяется следующим образом:

$$d_I = -\lim_{l\to 0}\frac{I}{\log l_B} = \lim_{l\to 0}\frac{\sum_{i=1}^{Nl_B}\frac{n_i(l_B)}{n}\log\left(\frac{n_{i(l_B)}}{n}\right)}{\log l_B}, \tag{1.10}$$

где l_B представляет собой длину кластера, необходимую для покрытия сети, $\frac{n_{i(l_B)}}{n}$ – вероятность узлов в в i-м кластере, когда ребро кластера равно l_B [62, 63].

Информационное измерение Цаллиса предложено Жангом [64] с целью объяснения сложности структуры сетей и для отражения степени самоподобия и фрактальных свойств. Информационная размерность Цаллиса может быть определена следующим образом:

$$d_T = -\lim_{l_B \to 0} \frac{\frac{1-\sum_{i=1}^{N_B} p_i (l_B)^q}{q-1}}{\ln l_B}, \quad (1.11)$$

Согласно уравнению (1.11), информация энтропии Цаллиса переписывается следующим образом:

$$H_T = \frac{\sum_{i=1}^{N_B} p_i (l_B)^q - 1}{1-q}, \quad (1.12)$$

где l_B представляет собой длину, покрывающую кластер, а число $p_i -$ вероятность, связанная с результатами покрытия кластера. $q -$ ограничивающий параметр обобщенной энтропии [61, 64].

Для описания сложности и неопределенности сети существует множество измерений, однако многие из них имеют фиксированные формулы и не являются гибкими. А информационная размерность Реньи имеет параметр α, который может меняться и влиять на само измерение. Размерность Реньи определяется следующим образом:

$$d_{\mathrm{R}} = -\lim_{l_B \to 0} \frac{\frac{1}{1-\alpha} \log \sum_{k=1}^{n} {p_k}^{\alpha}}{\log l_B}, \quad (1.13)$$

где $l_B -$ размер кластера. При α = 1 размерность Реньи соответствует информационной размерности, что легко доказывается с помощью уравнения Лопиталя. Когда α = 0, размерность в точности является классической хаусдорфовой размерностью [63, 64].

1.5 Проблема оптимального размещения узлов маршрутизаторов (шлюзов) в WMN

В современном мире WMN играют ключевую роль в различных приложениях, от городской инфраструктуры до экстренных коммуникаций. Эти сети состоят из множества беспроводных узлов, которые динамически создают сетевую инфраструктуру и обеспечивают соединение с внешними сетями, такими как Интернет, через специальные узлы, называемые шлюзами (GW). Шлюзы в WMN служат важным связующим звеном между локальной сетью и глобальными ресурсами, и их оптимальное размещение имеет решающее значение для обеспечения эффективности сети [65, 66].

Оптимизация размещения шлюзов в WMN представляет собой сложную задачу, которая формализуется как задача оптимизации. Эта задача относится к классу NP-трудных задач, что означает, что нахождение точного решения требует значительных вычислительных ресурсов и времени по мере увеличения размера сети. Основная цель состоит в том, чтобы определить такие узлы для размещения шлюзов, которые обеспечат максимальную пропускную способность сети, минимизируют задержки и увеличат надежность связи.

Важность оптимального размещения шлюзов обусловлена несколькими факторами. Во-первых, оно влияет на производительность сети, включая

пропускную способность и задержки передачи данных. Во-вторых, правильное размещение шлюзов может существенно улучшить управление трафиком и распределение нагрузки, что особенно важно в условиях изменяющихся сетевых условий и экстренных ситуаций. Неправильный выбор узлов для шлюзов может привести к перегрузке сети, увеличению времени задержки и снижению общего качества связи.

Для решения задачи оптимального размещения шлюзов разработаны различные методики и алгоритмы. Жадные алгоритмы используют простую стратегию выбора узлов, обеспечивающую локальную оптимальность, но могут не всегда находить глобально оптимальное решение. Метаэвристические методы, такие как генетические алгоритмы, алгоритмы имитации отжига и алгоритмы муравьиной колонии, позволяют находить хорошие решения для NP-трудных задач, исследуя пространство решений более эффективно, чем традиционные методы. Алгоритмы разделения и ограничений и другие специализированные подходы также применяются для решения задачи оптимального размещения шлюзов [67, 68].

Каждый из этих методов имеет свои преимущества и недостатки. Жадные алгоритмы обычно проще в реализации, но могут не учитывать все возможные варианты размещения. Метаэвристические методы, хотя и могут быть более сложными в настройке, часто предоставляют более качественные решения за счет более глубокой проработки пространства решений. Эффективность алгоритма оценивается по различным критериям, таким как время отклика, масштабируемость, способность адаптироваться к изменениям в сети и удовлетворение потребностей пользователей.

Таким образом, оптимальное размещение шлюзов в WMN является многогранной задачей, требующей комплексного подхода и использования различных методов оптимизации для достижения наилучших результатов. Постоянное развитие технологий и новые подходы к решению этой задачи способствуют улучшению производительности и надежности беспроводных ячеистых сетей [67-69].

1.6 Беспроводные сети и алгоритмы маршрутизации в современных системах связи

В настоящее время сети Ad Hoc приобретают всё большую актуальность благодаря своей способности быстро и эффективно организовывать связь в условиях отсутствия центральной инфраструктуры, что делает их незаменимыми в экстренных ситуациях, военных операциях и при создании временных сетей на больших территориях. Сети Ad Hoc делятся на несколько основных категорий, каждая из которых предназначена для определённых сценариев и имеет свои уникальные характеристики. Основные категории включают: Mobile ad hoc networks (MANETs), Smart phone ad hoc networks (SPANs), Wireless Sensor Networks (WSNs) и Wireless mesh networks (WMNs) [70, 71, 74].

Мобильные ад hoc сети (MANETs) представляют собой тип беспроводной сети, в которой мобильные устройства формируют сеть без фиксированной инфраструктуры. В таких сетях узлы могут свободно перемещаться и

соединяться друг с другом по беспроводным каналам, создавая самоконфигурируемую и самовосстанавливающуюся сеть. Топология MANETs изменяется динамично в зависимости от перемещения узлов, и каждый узел выполняет функции маршрутизатора, пересылая данные к другим узлам. MANETs обладают значительным преимуществом в том, что они могут работать автономно, без необходимости централизованного управления, и могут быть интегрированы в более крупные сети, такие как Интернет. Эти сети используют радиочастоты в диапазоне от 30 МГц до 5 ГГц, что позволяет обеспечивать беспроводное соединение в широком спектре частот. MANETs находят применение в различных областях, включая безопасность дорожного движения, охрану окружающей среды, домашние системы, здравоохранение, аварийно-спасательные операции, а также военные и оборонные системы. Основной задачей MANETs является обеспечение эффективной маршрутизации данных в условиях постоянно меняющейся топологии сети. Поскольку узлы могут перемещаться произвольно и изменять свою связь, необходимо поддерживать актуальную информацию о возможных маршрутах для передачи данных. Это требует постоянного обновления информации о текущем состоянии сети и адаптации к изменениям. В MANETs важно обеспечить, чтобы данные передавались эффективно, с минимальными задержками и перегрузками, даже при частых изменениях в сети. Таким образом, MANETs являются полезным инструментом для создания временных и адаптивных сетей, особенно в ситуациях, когда традиционная инфраструктура отсутствует или недоступна [72-74].

SPANs, или сети с динамическими соединениями мобильных устройств (Smartphone Peer Ad hoc Networks), представляют собой тип беспроводной сети, в которой устройства, такие как смартфоны, могут соединяться друг с другом без использования фиксированной инфраструктуры, такой как базовые станции в традиционных сотовых сетях. Эти сети обладают уникальными характеристиками, отличающимися от традиционных подходов к организации беспроводной связи. SPANs создаются с использованием существующего аппаратного обеспечения, в основном Wi-Fi и Bluetooth, а также программного обеспечения в коммерчески доступных смартфонах. Они являются самоконфигурирующимися и децентрализованными, что означает отсутствие центрального узла или контроллера. Это делает их более устойчивыми к отказам узлов и позволяет устройствам динамически присоединяться и покидать сеть без её разрушения. Топология SPANs может меняться в зависимости от перемещения узлов, что делает их гибкими и адаптивными. Одной из ключевых особенностей SPANs является поддержка многошагового ретранслирования, что отличает их от традиционных сетей Wi-Fi Direct. Это позволяет устройствам эффективно передавать данные, проходящие через несколько промежуточных узлов, и обеспечивает большую гибкость в управлении сетью. Важно отметить, что SPANs обеспечивают возможность пользователям присоединяться и покидать сеть по своему усмотрению, не нарушая её функционирования. SPANs особенно полезны в ситуациях, когда обычная сеть перегружена или недоступна,

такие как на конференциях, музыкальных фестивалях, во время стихийных бедствий или в отдалённых районах, где нет стабильной инфраструктуры. Они позволяют обеспечить связь в условиях, когда традиционные сети не могут обеспечить необходимое покрытие [75, 76].

WSNs представляют собой тип беспроводных сетей, состоящий из множества сенсорных узлов, которые собирают, обрабатывают и передают данные о физических или средовых параметрах. Эти сети применяются в различных областях, таких как мониторинг окружающей среды, управление ресурсами, здравоохранение, умные города и военные операции. Каждый сенсорный узел в WSN состоит из сенсора для измерения определённых параметров, микроконтроллера для обработки данных и радиомодуля для беспроводной передачи информации. Узлы могут быть развернуты в большом количестве по территории, что позволяет собирать данные с большой площади или из труднодоступных мест. Основной особенностью WSN является беспроводная связь, что позволяет легко и быстро развертывать сеть без необходимости прокладки кабелей. Это особенно важно для сетей, создаваемых в труднодоступных местах или временных условиях. Сенсорные узлы часто работают на батарейках и имеют ограниченные ресурсы, поэтому управление энергией является критическим аспектом. Оптимизация потребления энергии и продление срока службы батарей необходимы для обеспечения долговременной работы сети. WSNs обладают высокой скалируемостью, что позволяет создавать сети, состоящие из тысяч узлов, для мониторинга больших территорий или сложных систем. Важным аспектом является обработка и агрегация данных, которую выполняют сенсорные узлы, чтобы сократить объем информации, передаваемой через сеть, и уменьшить нагрузку на коммуникационные каналы. Сети могут адаптироваться к различным условиям окружающей среды, поддерживая надежную работу даже в сложных ситуациях. Беспроводные сенсорные сети находят применение в широком спектре областей. В экологии их используют для мониторинга состояния окружающей среды и управления природными ресурсами. В сельском хозяйстве − для отслеживания состояния почвы и условий роста растений. В здравоохранении − для мониторинга состояния пациентов и управления медицинскими устройствами. В умных городах — для мониторинга инфраструктуры и более эффективного управления городскими ресурсами. В WSN применяются различные топологии и протоколы для управления обменом данными, маршрутизацией и обработкой информации. Эти протоколы адаптированы для различных сценариев использования и включают механизмы для низкого потребления энергии, устойчивости к отказам и поддержки больших объемов данных. Беспроводные сенсорные сети являются мощным инструментом для сбора и анализа данных в реальном времени, что позволяет принимать обоснованные решения и оптимизировать процессы в различных областях применения [77-79].

WMNs представляют собой передовую технологию связи, которая активно внедряется в различных областях благодаря своим многочисленным преимуществам. Эти сети выделяются своей простотой и скоростью

развертывания, возможностью автоматического восстановления, динамической самоорганизацией и самоконфигурацией, что делает их высокоэффективными и экономичными. Универсальность WMNs позволяет их использование в широком спектре приложений, включая домашние широкополосные сети, образовательные учреждения, медицинские учреждения, автоматизацию зданий, управление чрезвычайными ситуациями, спасательные операции и военные приложения. Архитектура WMNs делится на три уровня узлов (Рисунок 1.7). На первом уровне находятся узлы шлюзов (GWs/Gws), которые обеспечивают связь с физическим уровнем сети и выполняют роль точки доступа в Интернет. Второй уровень состоит из квазистатических беспроводных маршрутизаторов (MRs/APs), которые управляют передачей пакетов между различными узлами сети, обеспечивая эффективное распределение трафика. На третьем уровне размещаются клиенты сети (MCs), включая настольные компьютеры, мобильные устройства и ноутбуки, которые получают доступ к сети через промежуточные узлы. Беспроводные маршрутизаторы, которые расположены на втором уровне, отличаются более сложными характеристиками по сравнению с клиентскими устройствами, такими как наличие нескольких интерфейсов передачи данных, повышенная мощность передачи и постоянное питание. Трафик в WMNs делится на два основных типа: шлюзовый трафик, который направляется через шлюз в Интернет, и клиентский трафик, который передается к целевому клиенту через несколько промежуточных узлов [80-84].

В настоящее время активно ведутся исследования в области маршрутизации в WMNs. Проектирование протоколов маршрутизации играет важную роль в обеспечении производительности и надежности WMNs. Основная цель протоколов маршрутизации заключается в обеспечении надежных маршрутов в многоскачковых сетях, работающих в условиях высоконеустойчивой беспроводной среды. В связи с ростом требований к высокой пропускной способности и минимальной задержке в сетях, необходимо разрабатывать методы оптимизации пропускной способности и минимизации задержек в WMNs [85-87].

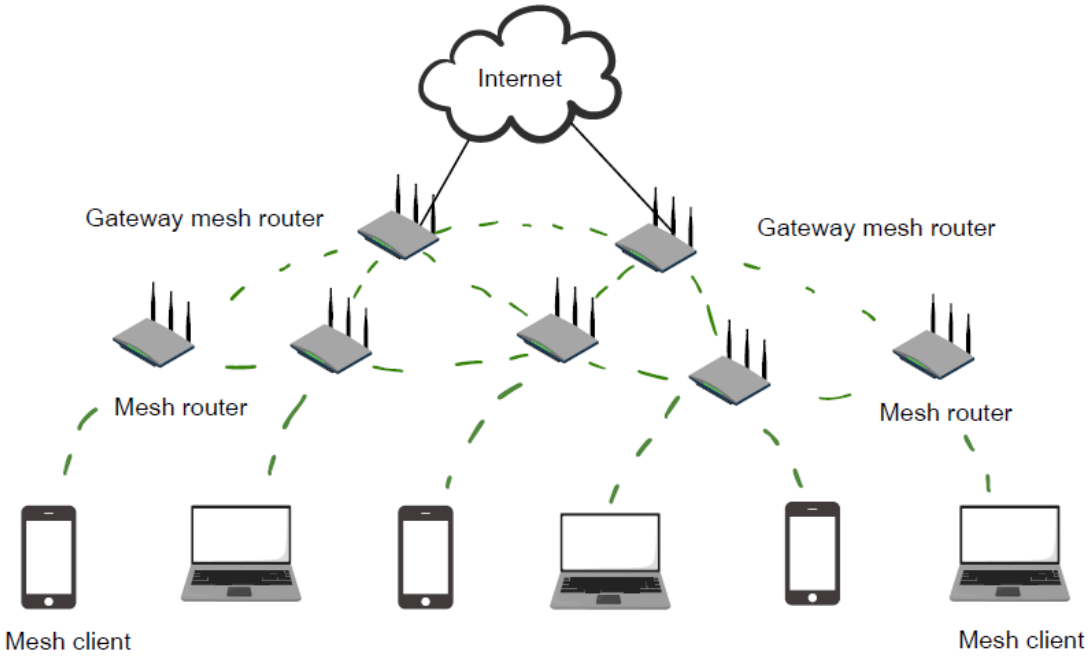

Рисунок 1.7 − Инфраструктура WMN

В данной работе WMN были выбраны объектом исследования ввиду их высокой гибкости, масштабируемости и эффективности в различных приложениях беспроводных сетей. WMNs имеют ряд преимуществ перед другими типами беспроводных сетей, такими как MANETs, SPANs и WSNs. В отличие от MANETs, где каждый узел может выполнять функции маршрутизатора, в WMNs маршрутизация осуществляется с использованием специализированных узлов маршрутизации. Это значительно улучшает производительность и надежность сети, так как эти узлы маршрутизации обеспечивают эффективное управление трафиком и снижают задержки. В отличие от SPANs, где соединения между устройствами могут быть нестабильными из-за перемещения мобильных устройств, WMNs обеспечивают более стабильное и надежное соединение. Это обеспечивается использованием узлов маршрутизации с фиксированными функциями и более развитой инфраструктурой сети, что позволяет поддерживать стабильное соединение, несмотря на изменения в топологии. В сравнении с WSNs, которые ориентированы на сбор и передачу данных о физических параметрах окружающей среды, WMNs предлагают более широкие возможности для масштабируемости и гибкости в применении.

Основная цель маршрутизации в WMN − обеспечение оптимального пути для передачи данных между исходным и конечным узлами сети. Для достижения этой цели используются различные протоколы маршрутизации, которые можно классифицировать на три основные категории: проактивные, реактивные и гибридные. Каждая категория обладает уникальным подходом к обработке и пересылке пакетов данных (Рисунок 1.8) [87-89].

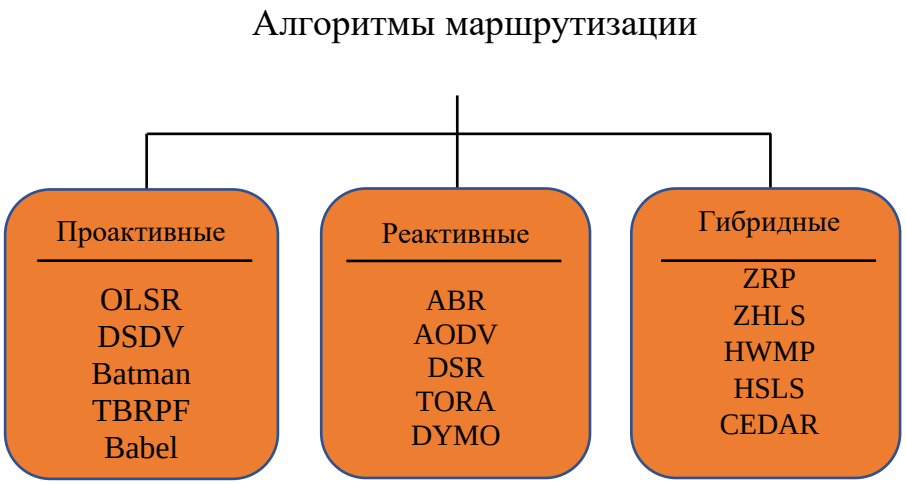

Рисунок 1.8 − Сортировка алгоритмов маршрутизации в WMN

Проактивные протоколы маршрутизации, также известные как табличные методы, поддерживают актуальные маршруты ко всем доступным узлам в сети, независимо от текущих потребностей в передаче данных. Эти протоколы непрерывно оценивают и обновляют маршруты, обеспечивая доступную и точную информацию о возможных путях. Основное преимущество проактивных протоколов заключается в том, что узлы могут мгновенно получать информацию о маршруте и начинать передачу данных без задержек на установку маршрута. Постоянная готовность маршрутов позволяет избежать задержек, что особенно важно для приложений с требованиями к низкой задержке. Узлы периодически обмениваются маршрутной информацией, поддерживая консистентные и актуальные таблицы маршрутизации, что уменьшает вероятность циклов и обеспечивает точность маршрутов. Примеры проактивных протоколов включают OLSR (Optimized Link State Routing), который использует оптимизированные состояния канала и концепцию Multi-Point Relays (MPR) для эффективного распространения информации о маршрутах; DSDV (Destination-Sequenced Distance-Vector), основывающийся на алгоритме векторного расстояния, где каждому маршруту присваивается номер последовательности, предотвращающий образование петель; Babel, векторный протокол с обнаружением петель, обеспечивающий гибкость и надежность, и TBRPF (Topology Broadcast based on Reverse-Path Forwarding), который передает топологическую информацию на основе обратного пути, что позволяет узлам оптимизировать таблицы маршрутизации и минимизировать широковещательный трафик. Эти протоколы обеспечивают эффективность в WMN, делая их подходящими для сред, где требуется минимальная задержка при передаче данных и стабильная поддержка маршрутов, несмотря на более высокие накладные расходы на поддержание актуальности информации [87-92].

Реактивные протоколы маршрутизации в WMN работают по принципу инициирования поиска маршрутов только при необходимости, например, когда требуется передача данных или происходят изменения в топологии сети. В отличие от проактивных протоколов, реактивные не создают маршруты заранее,

что значительно снижает сетевую нагрузку, минимизируя объем обмена маршрутной информацией. Это приводит к более эффективному использованию сетевых ресурсов, поскольку уменьшение объема служебного трафика позволяет экономить пропускную способность и вычислительные ресурсы. Ключевыми преимуществами реактивных протоколов являются снижение нагрузки на сеть за счёт установления маршрутов только по мере необходимости и высокая адаптивность к изменениям топологии. Это делает их особенно эффективными для сетей с динамической топологией, таких как мобильные сети, где маршруты могут часто прерываться из-за перемещения узлов. Однако в WMNs, где узлы обычно стационарны или имеют минимальную подвижность, реактивные протоколы также демонстрируют лучшую масштабируемость, поскольку сокращают необходимость в частых обновлениях маршрутов. Примеры реактивных протоколов включают AODV (Ad hoc On-Demand Distance Vector), который устанавливает маршруты через запросы на маршрут (RREQ) и ответы на маршрут (RREP), и DSR (Dynamic Source Routing), использующий маршрутизацию по источнику, где полный маршрут указывается в заголовке каждого пакета данных. Другие примеры включают DYMO (Dynamic MANET On-demand), улучшенную версию AODV с эффективным управлением маршрутами, и TORA (Temporally-Ordered Routing Algorithm), который обеспечивает направленную маршрутизацию для быстрого реагирования на изменения в сети [89, 90].

Гибридные протоколы маршрутизации объединяют преимущества как проактивных, так и реактивных протоколов маршрутизации, преодолевая их недостатки и обеспечивая оптимальное управление маршрутами с минимальными затратами на управление сетью. Этот тип протокола использует различные алгоритмы в разных частях инфраструктуры: проактивные методы поддерживают постоянную актуализацию маршрутных таблиц в основной сети, тогда как реактивные методы обеспечивают быстрый поиск маршрутов в одноранговых частях сети. Реактивные протоколы используются в области одноранговой сети, в то время как проактивные протоколы используются в беспроводной магистрали, обеспечивая эффективную маршрутизацию. Примеры таких протоколов включают ZRP, который разделяет сеть на зоны для оптимизации маршрутизации, ZHLS, использующий иерархический подход для эффективного распределения информации о состоянии каналов, и HWMP, сочетающий разные режимы маршрутизации для обеспечения надежной и быстрой передачи данных в динамичных беспроводных сетях. Другие примеры включают Hazy-Sighted Link State (HSLS), который уменьшает частоту обновления таблиц для отдаленных узлов, снижая объем управляющего трафика и адаптируясь к изменяющимся условиям сети. Core Extraction Distributed Ad Hoc Routing (CEDAR) создает основной набор узлов для проактивного поддержания маршрутов и использует реактивные методы для маршрутизации до основного набора, улучшая масштабируемость и адаптивность. Order-One Network Protocol (OONP) поддерживает минимальные таблицы маршрутизации, используя

проактивный подход для информации о соседних узлах и реактивные методы для передачи данных через сеть, минимизируя потребление ресурсов [90-92].

В данной работе были исследованы алгоритмы маршрутизации, такие как OLSR, AODV, Dijkstra, и ACO поскольку они являются одними из наиболее распространенных и часто используемых в современных беспроводных сетях. Эти алгоритмы обладают различными особенностями и применяются в различных сценариях, включая беспроводные сети Ad Hoc, MANETs и WMNs. Изучение этих алгоритмов также способствует развитию новых методов и технологий в области сетевой маршрутизации, что имеет важное значение для дальнейшего развития беспроводных и распределенных сетей, включая Интернет вещей (IoT), смарт-города, автономные транспортные системы и другие инновационные приложения.

OLSR − это протокол проактивной маршрутизации, который постоянно обновляет и поддерживает карту топологии сети, обеспечивая эффективную передачу данных между узлами сети. Протокол OLSR, представленный Французским национальным институтом исследований в области компьютерных наук и управления (INRIA), имеет две версии: OLSRv1 и OLSRv2. OLSRv2 − это обновленная версия OLSRv1, сохраняющая тот же механизм, но с некоторыми улучшениями.

В протоколе OLSRv1 каждый узел в сети передает информацию о текущем состоянии соединения всем остальным узлам. Алгоритм OLSRv1 использует два типа управляющих сообщений: HELLO и TOPOLOGY CONTROL (TC), причем сообщения TC периодически передаются только узлами многоточечных ретрансляторов (MPR), которые минимизируют широковещательную передачу и являются важным аспектом протокола OLSRv1. В OLSRv1 каждый узел выбирает набор узлов для многоточечных ретрансляций (MPR) из своих соседей, с которыми у него установлена двунаправленная связь. Набор узлов MPR может изменяться со временем на основе информации, полученной от других узлов в сети. Это изменение управляется данными из HELLO-сообщений, в которых узлы обмениваются информацией о своей видимости и изменениях в топологии сети. При передаче сообщений, узел отправляет их всем своим соседям, но дальнейшую передачу осуществляют только те узлы MPR, которые получают сообщение впервые. Это позволяет снизить накладные расходы, связанные с передачей данных, так как уменьшается количество дублирующих передач. В отличие от традиционных проактивных протоколов, которые используют статические таблицы маршрутизации, OLSRv1 передает информацию о состоянии канала связи. Это позволяет узлам быстро адаптироваться к изменениям в топологии сети, обновляя маршруты на основе актуальной информации о доступности соседних узлов. OLSRv1 использует порядковые номера для версионного контроля и предотвращения циклов маршрутизации. Это улучшает надежность маршрутизации и обеспечивает доставку сообщений в правильном порядке. Протокол OLSRv1 хорошо работает в сетях с высокой плотностью узлов и большим количеством связей. Использование MPR

позволяет снижать нагрузку на сеть за счет минимизации широковещательных передач.

OLSRv2 является развитием протокола OLSRv1, предназначенным для проактивной маршрутизации в беспроводных сетях. OLSRv2 включает два метода выбора узлов MPR: «MPR с переполнением» (Overflow MPR) и «маршрутизация MPR» (Routing MPR). «MPR с переполнением» (Overflow MPR) выбирает минимальное подмножество одношаговых соседей, которые покрывают всех двухшаговых соседей, чтобы минимизировать количество MPR и снизить расходы на передачу данных. Это достигается анализом соседей на один и два перехода, что позволяет контролировать плотность сети и уменьшить объем контрольных данных. В отличие от этого, «маршрутизация MPR» (Routing MPR) выбирает узлы не только для покрытия двухшаговых соседей, но и для оптимизации маршрутизации данных, учитывая качество связей и другие метрики, такие как пропускная способность или задержки. Эти узлы выбираются таким образом, чтобы обеспечить лучшие маршруты для передачи данных, что улучшает общую эффективность маршрутизации, даже если это не минимизирует число ретрансляторов. На рисунке 1.9 приведена схема сети, состоящей из семи узлов (обозначенных от 1 до 7). Узлы соединены направленными стрелками, отражающими поток сообщений между ними, характерный для протокола маршрутизации OLSR. Стрелки указывают на направление передачи HELLO сообщений для обнаружения соседей или на широковещательную передачу сообщений для управления топологией (TC) через выбранные MPR для распространения информации по сети.

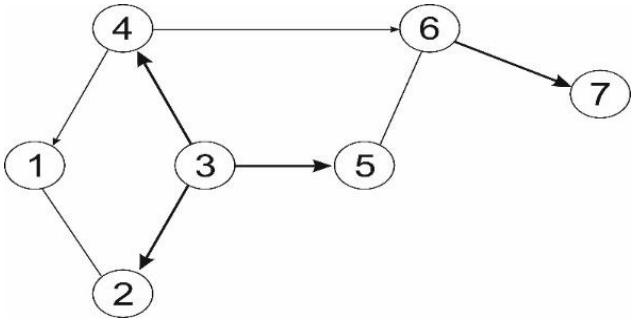

Рисунок 1.9 – Пример маршрутизации протокола OLSR

Протокол OLSRv2 может использовать различные метрики для выбора оптимальных маршрутов, что позволяет лучше учитывать специфические характеристики сети, такие как качество связи или задержки. OLSRv2 снижает количество контрольных сообщений и объем передаваемых данных за счет оптимизации выбора MPR, что особенно важно в плотных сетях. Протокол лучше справляется с увеличением числа узлов в сети, так как ограничивает распространение контрольных сообщений [93-96].

Ad hoc On-Demand Distance Vector (AODV) − это протокол маршрутизации, разработанный для MANETs и WMNs. Он применяет on-demand подход, что означает, что маршруты создаются только тогда, когда они

действительно нужны. Это делает AODV особенно эффективным, так как он минимизирует управляющий трафик и накладные расходы на передачу данных. AODV облегчает создание динамических, самоинициирующихся маршрутов с несколькими переходами между мобильными узлами. В режиме реагирования маршруты устанавливаются только в ответ на конкретные запросы на связь, позволяя узлам быстро находить маршруты к новым пунктам назначения, не тратя ресурсы на обслуживание маршрутов к неактивным узлам. Такой подход обеспечивает быструю адаптацию к изменениям в топологии сети и оперативное реагирование на сбои в соединении. Протокол работает без циклов маршрутизации, что устраняет проблему «счета до бесконечности», известную из алгоритма Беллмана-Форда. Это позволяет AODV быстро конвергировать при изменениях в топологии сети, особенно в условиях высокой мобильности узлов. При сбоях в соединении AODV оперативно уведомляет затронутые узлы, позволяя им аннулировать маршруты, связанные с отключенными соединениями. Отличительной чертой AODV является использование порядковых номеров маршрутов, которые генерируются узлом назначения. Эти номера передаются запрашивающим узлам вместе с информацией о маршруте, что позволяет выбрать наиболее актуальный маршрут и избежать устаревших данных. Узлы выбирают маршрут с самым высоким порядковым номером, что гарантирует корректность и непрерывность маршрутизации. AODV разработан для мобильных сетей с числом узлов от десятков до тысяч. Он справляется с различными уровнями мобильности, от низкой до высокой, и может обрабатывать различные объемы трафика. Особое внимание в AODV уделяется минимизации управляющего трафика и сокращению накладных расходов на передачу данных, что способствует повышению масштабируемости и общей производительности сети. На рисунке 1.10 показан пример маршрутизации в AODV с семью узлами, обозначенными от A до G.

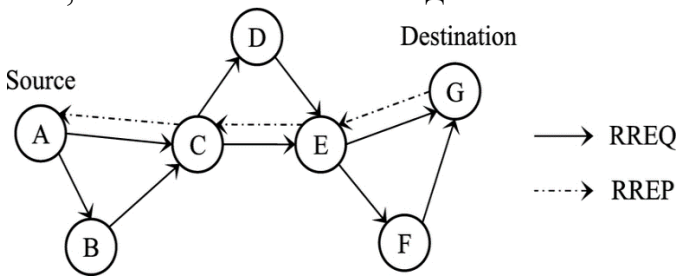

Рисунок 1.10 − Пример маршрутизации AODV

Узел A начинает поиск маршрута к узлу G, отправляя запрос маршрута RREQ с уникальным номером. Промежуточные узлы пересылают RREQ, если не имеют текущего маршрута к G. Когда RREQ достигает узла G или узла с готовым маршрутом к G, он отвечает маршрутным сообщением RREP обратно к узлу A, содержащим информацию о маршруте. Маршруты поддерживаются через сообщения об ошибках маршрута RERR, что позволяет узлам обновлять свои таблицы маршрутизации при обнаружении сбоев. На иллюстрации узел A обозначен как источник, а узел G как пункт назначения. Сплошные стрелки указывают путь запроса маршрута RREQ, передаваемого от A к G. Пунктирные

стрелки показывают путь ответа маршрута RREP, который подтверждает маршрут и возвращается обратно к узлу A. Это иллюстрирует основной механизм обнаружения и установления маршрута в AODV [97-100].

Алгоритм Dijkstra − это классический метод поиска кратчайшего пути в графе от одной вершины ко всем остальным. Основные достоинства этого алгоритма заключаются в его эффективности и простоте реализации. Он работает только с графами, в которых веса рёбер неотрицательны, и вычисляет длины кратчайших путей от начальной вершины ко всем остальным вершинам. Алгоритм итеративно обрабатывает вершины графа, на каждом шаге выбирая вершину с наименьшим расстоянием от начальной и обновляя длины кратчайших путей до остальных вершин через эту вершину. Время выполнения алгоритма пропорционально количеству вершин и рёбер в графе. Важно отметить, что алгоритм Dijkstra не подходит для графов, где ребра могут иметь отрицательные веса, поскольку он не учитывает возможность наличия циклов с отрицательным весом. Для таких графов лучше использовать алгоритм Беллмана-Форда. Кроме того, алгоритм Dijkstra предполагает, что все вершины графа известны заранее, что может быть непрактично для динамически изменяющихся или больших сетей. Если требуется найти кратчайшие пути от нескольких начальных вершин, алгоритм Dijkstra нужно будет запустить многократно, что может быть неэффективным. При наличии циклов с отрицательными весами алгоритм может дать неверные результаты или даже зациклиться [101-104].

Алгоритм оптимизации муравьиными колониями (Ant Colony Optimization, ACO) является метаэвристическим методом, вдохновленным поведением муравьев при поиске оптимального пути к источнику пищи. Он был впервые предложен Марко Дориго в 1992 году для решения комбинаторных задач, таких как задача коммивояжера. Принцип работы алгоритма основан на моделировании поведения муравьев, которые оставляют феромоны на пути к источнику пищи. Другие муравьи обнаруживают эти следы и предпочитают выбирать пути с большим количеством феромонов. Во время выполнения алгоритма строятся кандидаты на решение (потенциальные решения), а затем на основе количества феромонов и дополнительной эвристической информации выбираются пути для последующей оптимизации. Муравьи оставляют феромоны на пройденных путях. Феромон служит индикатором качества пути, и его концентрация обычно увеличивается с увеличением привлекательности пути. После каждой итерации алгоритм обновляет уровень феромонов на каждом ребре графа, основываясь на результатах предыдущей итерации и эвристических данных. Обычно феромоны испаряются со временем, чтобы предотвратить застревание в локальных оптимумах. При выборе пути каждый муравей опирается на уровень феромонов на рёбрах и дополнительную эвристическую информацию (например, расстояние между городами в задаче коммивояжера). Используется вероятностный подход, который позволяет муравьям исследовать новые пути, сохраняя баланс между исследованием и использованием уже известных путей. Алгоритм ACO обеспечивает баланс между интенсификацией

(усилением феромонов на более коротких путях) и диверсификацией (исследованием новых путей) для достижения оптимальных решений. Он широко применяется для решения различных комбинаторных оптимизационных задач, таких как задача коммивояжера, задача раскроя и маршрутизация в сетях. Его гибкость и эффективность делают его популярным инструментом в области оптимизации и исследований операций [105-108].

Необходимо отметить, что в данном работе выбранный нами объект исследования WMN состоит из нескольких беспроводных каскадных каналов, каждый из которых имеет свою вероятность ошибки. Однако существующие алгоритмы маршрутизации не учитывают вероятности ошибок, связанных с этими каскадными каналами. Например, в основополагающей работе [109] алгоритм Dijkstra фокусируется на поиске кратчайшего пути между исходным и целевым узлами, основываясь исключительно на сумме расстояний между узлами. При таком подходе не учитывается вероятность ошибок между каналами, поэтому пропускная способность определенных трактов не является максимальной. Кроме того, алгоритм Dijkstra не адаптируется к динамическим сетевым условиям, что делает его менее подходящим для WMNs. Без механизмов динамической оптимизации маршрутов алгоритм Dijkstra может не обеспечить оптимальную производительность в динамических WMNs.

Авторы в работе [110] предлагают алгоритм оптимизации колонии муравьев (ACO), который использует вероятностный подход для определения оптимальных путей на графах. Однако применение этого метода к WMNs сопряжено с трудностями, такими как необходимость установки таких параметров, как количество итераций и количество муравьев, которые могут быть нетривиальными и не гарантирующими оптимальных результатов в WMN. Кроме того, алгоритм ACO может страдать от проблем масштабируемости при применении к крупномасштабным WMN с большим количеством узлов и связей. Вычислительные затраты, связанные с поддержанием феромонных следов и обновлением перемещений муравьев, могут стать непомерно высокими, что приведет к увеличению времени обработки и потребления ресурсов. Кроме того, алгоритмы ACO могут с трудом адаптироваться к динамическим условиям сети или изменениям топологии, поскольку они обычно полагаются на статическую эвристику или предопределенные параметры.

В работе [111] был проведён анализ полосы пропускания в бесконечном каскаде симметричных каналов. Результаты показали, что полоса пропускания уменьшается с увеличением числа каскадов. Особеным было то, что при бесконечном числе каскадов с равными вероятностями пропускная способность канала стремится к нулю. Анализ был сосредоточен исключительно на симметричных каскадных каналах с равными вероятностями и не учитывал асимметричные или неравные конфигурации каналов.

В работе [112] энтропийный подход был применен в маршрутизации, однако маршруты создавались случайным образом, без учета вероятностей ошибки для каждого канала. Такой случайный выбор маршрутов не всегда

гарантирует максимальную пропускную способность, поскольку могут быть задействованы каналы с высокой вероятностью ошибок.

Кроме того, важно распознавать уязвимости, присутствующие в алгоритмах, используемых в настоящее время в WMNs, таких как OLSR и AODV. OLSR, хотя и эффективен для статических сетей, подвержен переполнению таблиц маршрутизации и требует периодического потока управляющих сообщений. OLSR использует периодический обмен информацией о состоянии канала между узлами для поддержания таблиц маршрутизации, что приводит к увеличению нагрузки на управляющие сообщения и потенциальной перегрузке сети. Постоянный обмен управляющими пакетами потребляет пропускную способность сети и вычислительные ресурсы, снижая общую пропускную способность и скорость реагирования сети. AODV, хотя и предназначен для обеспечения установления маршрута по требованию в WMN, имеет несколько уязвимостей. Одной из заметных уязвимостей является задержка обнаружения маршрута, возникающая при создании новых маршрутов. AODV использует пакеты обнаружения маршрута для поиска путей между исходным и конечным узлами, что может привести к задержкам, особенно в высокодинамичных средах с частыми изменениями топологии [113].

Вывод по пункту 1.1

Классификация модуляции телекоммуникационных сигналов является ключевым аспектом современных систем связи, поскольку она обеспечивает надежность передачи данных и непосредственно влияет на маршрутизацию. В обзоре рассматриваются различные подходы, включая вероятностные методы и анализ признаков, а также современные алгоритмы машинного и глубокого обучения, которые позволяют автоматически извлекать скрытые признаки сигналов. Многие из таких алгоритмов применяют инновационные техники для подавления помех в сложных MIMO-сигналах и повышения точности классификации. Например, некоторые AMC-методы для MIMO-систем используют сверточные нейронные сети в сочетании с методом нулевого принуждения (zero-forcing equalization). Однако в реальных системах связи полностью подавить помехи в MIMO-сигналах невозможно, что снижает эффективность классификации в сложных условиях.

Алгоритмы классификации модуляций, основанные на использовании информационной энтропии для извлечения признаков, способствуют повышению точности классификации в условиях сложных и изменяющихся характеристик сигналов. Энтропия, оценивающая уровень неопределенности и отражающая сложность сигнала, играет важную роль в этих подходах. Однако, несмотря на их конкурентоспособность, взаимная информация (MI) может быть более эффективным инструментом, так как учитывает зависимости между несколькими переменными. В отличие от энтропии, которая сосредотачивается на неопределенности отдельных переменных, MI оценивает, насколько информация об одной переменной снижает неопределенность другой [38-40].

Важно отметить, что для достижения лучших результатов в классификации модуляций необходимо разработать новые алгоритмы, учитывающие сложные и динамичные условия передачи данных.

Вывод по пункту 1.2

Наука о сетях представляет собой динамично развивающуюся область исследований, сосредоточенную на изучении сложных топологических свойств сетей. Методы фрактального анализа с масштабной инвариантностью позволяют эффективно оценивать и описывать структуру этих сетей, выявляя самоподобие и структурную сложность на различных масштабах. В этом контексте кластеризация сети играет ключевую роль, особенно при проектировании алгоритмов маршрутизации, поскольку она позволяет оптимально управлять ресурсами и минимизировать задержки, что критично для обеспечения надежной и эффективной связи. Для вычисления фрактальной размерности реальных сетей предложен метод «box covering», который включает алгоритмы кластеризации, такие как «greedy coloring» (GC), «random sequence» (RS), « compact box burning» (CBB) и «maximum excluded mass» (MEMB). Тем не менее, несмотря на свои преимущества, вышеописанные методы сталкиваются с определенными проблемами, связанными со случайным выбором центральных узлов или выбором узлов с максимальной исключённой массой. Алгоритмы GC, RS и CBB часто приводят к образованию множества кластеров с одним узлом, что делает их недостаточно эффективными для оптимального деления сети на кластеры. Хотя алгоритм MEMB демонстрирует высокую эффективность в ряде случаев, он также может не обеспечивать оптимального разделения сетей на кластеры, поскольку в результате его работы могут формироваться кластеры с одним узлом.Таким образом, для достижения более высокой точности и эффективности при кластеризации сетей необходимы дополнительные исследования и улучшения существующих алгоритмов кластеризации, чтобы преодолеть их ограничения и повысить надежность сетевых архитектур.

Вывод по пункту 1.3

В данном пункте рассмотрены теоретические основы информационной энтропии, представляющей собой меру неопределенности в системе, введенную Клодом Шенноном. Энтропия играет ключевую роль в оценке эффективности передачи и хранения данных, а также в различных областях, таких как связь и обработка сигналов. Обсуждены концепции совместной и условной энтропии, а также взаимная информация, которая позволяет оценить зависимость между переменными и уровень информации, получаемой при наблюдении одной из них. Понимание этих понятий является необходимым для дальнейшего анализа и оптимизации систем связи и обработки данных.

Вывод по пункту 1.4

Информационные размерности Реньи и Цаллиса представляют собой обобщения классической энтропии Шеннона, играя важную роль в анализе неопределенности и сложности сетей. Энтропия Реньи, параметризуемая α, позволяет гибко измерять неопределенность в различных контекстах, тогда как энтропия Цаллиса, регулируемая параметром q, подходит для описания

неэкстенсивных систем. Информационная размерность сети, основанная на этих энтропиях, предоставляет ценную информацию о структурной сложности и самоподобии сетей. Введение параметров в измерения, таких как α и q, усиливает гибкость подходов, позволяя более точно отражать динамику и особенности различных систем. Эти меры открывают новые горизонты для анализа и оптимизации сетевых структур, предоставляя важные инструменты для будущих исследований в области теории информации и маршрутизации.

Вывод по пункту 1.5

Оптимальное размещение шлюзов в беспроводных mesh-сетях (WMN) является критически важной задачей, влияющей на производительность сети, пропускную способность и задержки передачи данных. Эффективное размещение шлюзов улучшает управление трафиком и распределение нагрузки, что особенно актуально в условиях изменяющихся сетевых условий и экстренных ситуаций. Поскольку задача относится к классу NP-трудных задач, разработаны различные методы и алгоритмы, включая жадные алгоритмы и метаэвристические подходы, такие как генетические алгоритмы и алгоритмы муравьиной колонии. Каждый из методов имеет свои преимущества и недостатки, и выбор подхода зависит от конкретных требований к производительности, масштабируемости и адаптивности сети. Таким образом, оптимизация размещения шлюзов в WMN требует комплексного подхода и постоянного совершенствования алгоритмов для повышения надежности и эффективности беспроводных сетей.

Вывод по пункту 1.6

WMN занимают ключевую роль в современных телекоммуникационных системах, предоставляя гибкость, надёжность и адаптивность передачи данных в условиях быстро меняющихся топологий и различных приложений. Однако успешность их функционирования в значительной степени определяется эффективностью применяемых протоколов маршрутизации. Существующие алгоритмы, включая Dijkstra, OLSR, AODV и ACO, имеют определённые преимущества, но также сталкиваются с существенными ограничениями. Например, алгоритм Dijkstra не учитывает вероятность ошибок между каналами и не адаптируется к динамичным сетевым условиям, что может негативно сказываться на надёжности сети. OLSR, будучи эффективным для статичных сетей, подвержен проблемам переполнения таблиц маршрутизации и требует периодической отправки управляющих сообщений. Протокол AODV, который использует пакеты для обнаружения маршрута, сталкивается с задержками, особенно в высокодинамичных средах с частыми изменениями топологии, что может замедлять создание новых маршрутов. Методы маршрутизации на основе колонии муравьёв (ACO) страдают от проблем масштабируемости и сложности настройки параметров. Эти недостатки подчёркивают необходимость разработки более адаптивных и интеллектуальных подходов к маршрутизации в WMN, которые учитывают стохастические свойства каналов связи, изменчивость сетевых условий и задержки, возникающие при обнаружении маршрутов.

2 ОПРЕДЕЛЕНИЕ ТИПОВ МОДУЛЯЦИЙ ТЕЛЕККОМУНИКАЦИОННЫХ СИГНАЛОВ

2.1. Модель системы MIMO, генерация набора данных и экспериментальная установка

А. MIMO модель

В работе [114] рассматривается система MIMO, состоящая из N_s передающих антенн и N_r принимающих антенн ($N_r \geq N_s$). Предполагается, что канал MIMO является плоским замирающим и не зависящим от времени комплекснозначным каналом. Принятый сигнал в n-й момент выборки можно выразить следующим образом:

$$r(n) = Hs(n) + g(n), \quad (2.1)$$

где H обозначает матрицу канала MIMO размером $N_r \times N_s$, которая следует комплексному нормальному распределению с круговой симметрией; $r(n) = [r_1(n), r_{N_r}(n)]^{\mathrm{T}}$ вектор полученного сигнала размером $N_r \times 1$; $s(n) = [s_1(n), s_{N_s}(n)]^{\mathrm{T}}$ вектор переданного сигнала размером $N_s \times 1$; и $g(n) = [g_1(n), g_{N_r}(n)]^{\mathrm{T}}$ вектор аддитивного белого гауссовского шума размером $N_r \times 1$.

Б. Генерация набора данных

В данной работе рассматривается модель сложной много-антенной системы, и процесс генерации набора данных представлен на Рисунок 2.1. В частности, случайные данные модулируются с использованием различных типов модуляции, таких как бинарная фазовая манипуляция (BPSK), квадратурная фазовая манипуляция (QPSK), восьмифазная манипуляция (8PSK), шестнадцатиричная квадратурная амплитудная модуляция (16QAM) и шестидесятичетырехричная квадратурная амплитудная модуляция (64QAM). Модулированный вектор сигнала, обозначаемый как S, имеет размер $1 \times N$ (где N количество символов, и N=256, 512, 1024. Для обеспечения справедливого сравнения S нормализуется с единичной мощностью, то есть $\|S\|_2^2 = 1$. После нормализации S преобразуется в матрицу размером $N_t \times N/N_t$, обозначаемую как $[s_1; s_2; s_{N_t}]$, где $s_i = [s_i(1), s(2) \ldots s_i(\mathrm{N}/N_t)$, $i \in [1, N_t]$; − вектор переданного сигнала на i-й антенне.

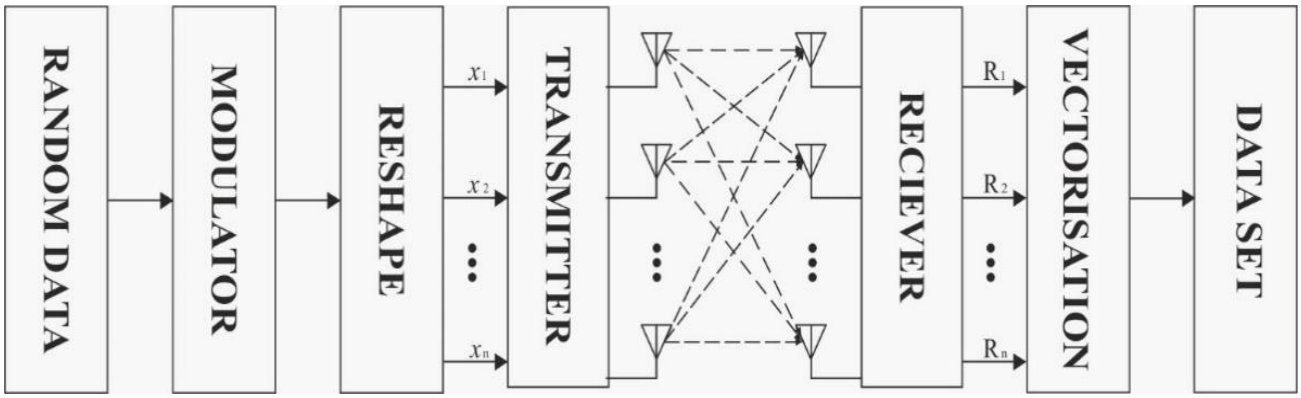

Рисунок 2.1 − Процесс генерации набора данных.

Вектор сигнала, полученного на j-й приемной антенне и прошедшего через MIMO-канал, обозначается как $[r_1; r_2; r_{N_t}]$ и имеет размерность $N_r \times N.\ N/N_t$, где $r_j = [r_j(1), r_j(2), r_j(\mathrm{N}/N_t), j \in [1, N_r]$. Далее матрица полученного сигнала может быть векторизована в вектор размером $1\times N$, обозначаемый как r_*. Тестовые и тренировочные данные извлекаются из r_*. В частности, действительная и мнимая части r: $R\ (r)$ и $I\ (r)$ разделяются и комбинируются в матрицу размером $2\times N$ [$R\ (r)$ $(I\ (r)$], которая является образцом для обучения и тестирования. Следует отметить, что для каждого значения отношения сигнал/шум (SNR) подготовлено 6,000 образцов для обучения и 4,000 образцов для тестирования.

B. Экспериментальная установка

Для оценки эффективности и надежности предлагаемого метода автоматической классификации модуляции (AMC) в реальных условиях связи была проведена серия экспериментов с использованием интегрированного программно-аппаратного комплекса для сбора данных диаграмм созвездий IQ. Программный компонент этого комплекса реализован в программе Matlab+Simulink, а аппаратный компонент включает Zedboard+AD9361 (Рисунок 2.2). Zedboard, подключенный к ПК через Ethernet-кабель, обеспечивает связь между программным компонентом и радиочастотным модулем AD9361. Радиочастотный модуль AD9361 обеспечивает беспроводную связь между приемником и передатчиком.

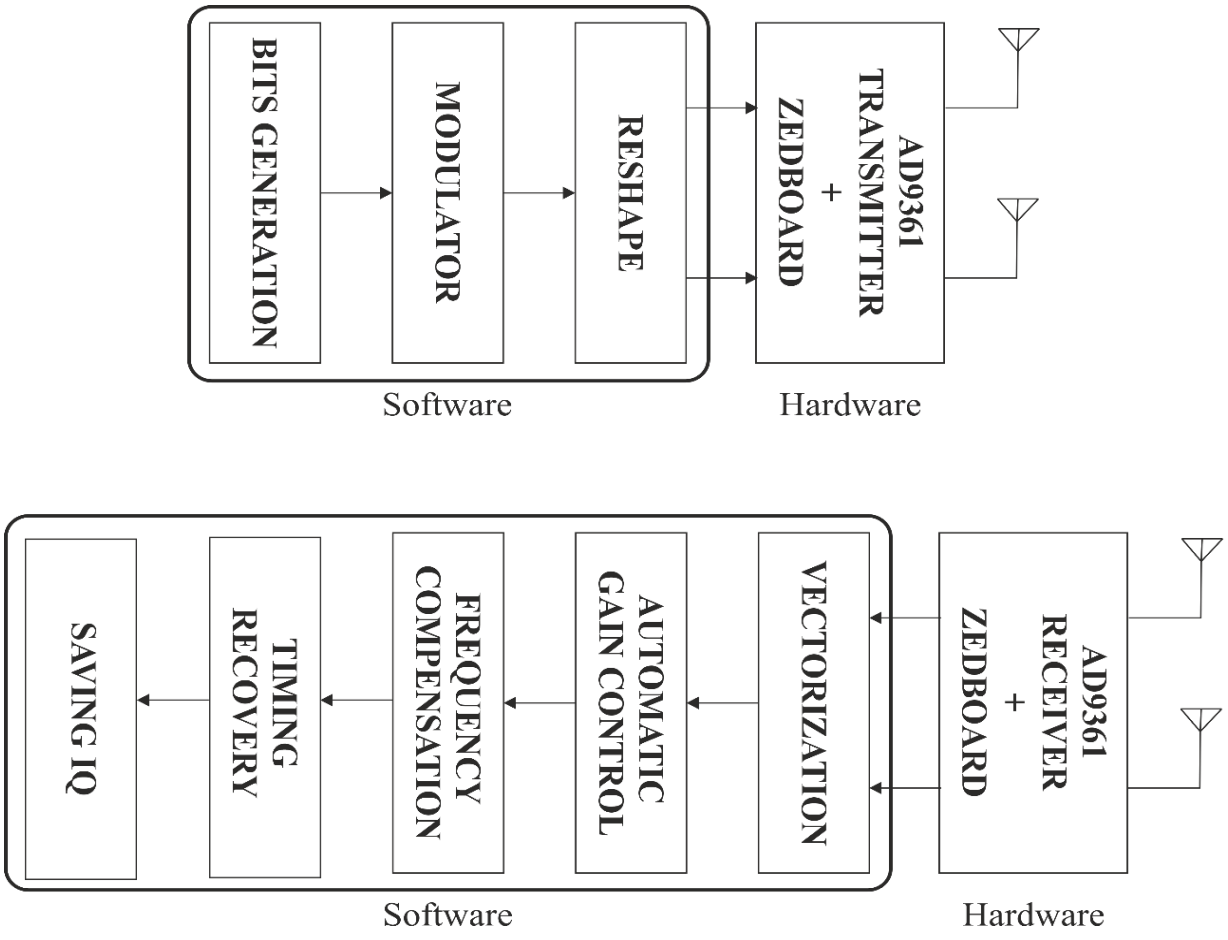

Рисунок 2.2 – Блок-схема передатчика и приемника

Программная часть передающего компонента включает генератор случайных битов, модулятор и устройство демультиплексирования для преобразования формы сигнала, в то время как аппаратная часть состоит из AD9361 с двумя передающими антеннами.

Приемная часть программного компонента выполняет мультиплексирование для векторизации и обработку сигналов для диаграмм созвездий IQ. Обработка сигналов включает задачи, такие как автоматическое управление усилением (AGC), компенсация частоты и восстановление временных характеристик. Аппаратный компонент этой приемной части состоит из AD9361 с двумя антеннами (Рисунок 2.2). Значения диаграмм созвездий IQ были сохранены и использованы для тестирования предлагаемого алгоритма AMC. Рисунок 2.3 иллюстрирует аппаратную часть экспериментальной установки, предназначенной для приема и передачи сигналов в MIMO-каналах.

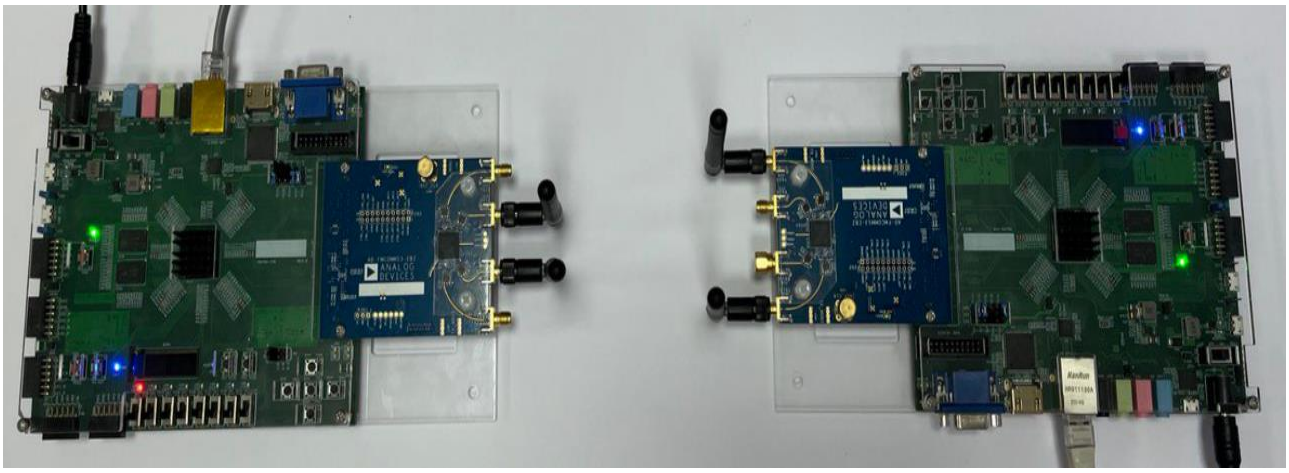

Рисунок 2.3 − Аппаратная часть экспериментальной установки

Конфигурация радиочастотного модуля AD9361 была выполнена с использованием программы Simulink, при этом были выбраны следующие параметры: частота 2,4 ГГц и базовая частота дискретизации 520,841 кГц. Параметры усиления передатчика и приемника были настроены для управления значением отношения сигнал/шум (SNR) [114].

2.2. Алгоритм классификации модуляций сложных сигналов MIMO на основе взаимной инфрмации

Предлагаемый алгоритм классификации модуляций телекоммуникационных сигналов состоит из двух основных частей: извлечения признаков на основе взаимной информации $I(X;Y)$ и классификатора SVM. Структура метода автоматической классификации модуляции (AMC) представлена на Рисунок 2.4.

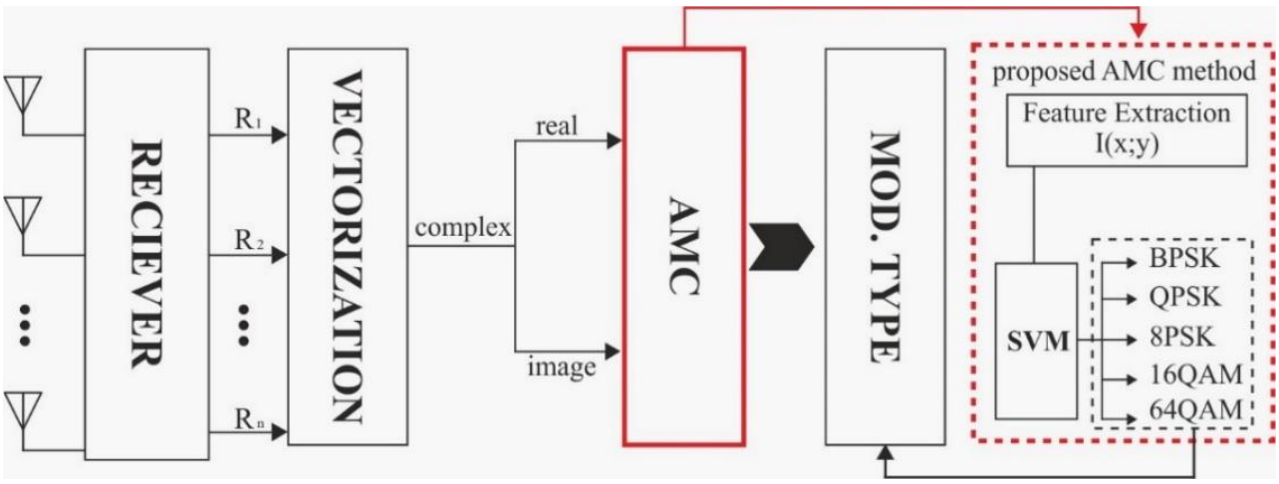

Рисунок 2.4 − Структура предлагаемого метода автоматической классификации модуляции (AMC).

Метод включает извлечение признаков на основе взаимной информации (MI) с использованием гистограммы IQ диаграммы. Процесс начинается с построения диаграммы IQ, которая отображает комплексные значения модулированных сигналов. Синфазный компонент полученного сигнала обозначается переменной *X*, в то время как компонент в квадратуре обозначается переменной *Y*. Затем для расчета MI строится гистограмма (Рисунок 2.5) плоскости *XY*, разделенную на ячейки равного размера ($\Delta X \times \Delta Y$) с координатами i, j. Подсчитывая количество точек $k_{i,j}$ в каждой ячейке (i, j), MI можно рассчитать по следующей формуле:

$$I(X;Y) = \sum_{j=1}^{k_y} \sum_{i=1}^{k_x} \left(\frac{k_{ij}}{N}\right) \log\left(\frac{k_{ij}N}{k_i k_j}\right), \quad (2.2)$$

где *N* общее количество точек.

Для реализации алгоритма необходимо выполнить следующие шаги:

1. Построить диаграмму IQ.
2. Построить гистограмму двумерной диаграммы IQ.
3. Рассчитать взаимную информацию (MI) в соответствии с формулой (4).
4. На последнем этапе применить классификатор SVM для определения типа модуляции.

На Рисунок 2.5 гистограмма IQ диаграммы демонстрирует значительные различия в зависимости от типа модуляции. Плотность точек уменьшается по мере увеличения количества символов.

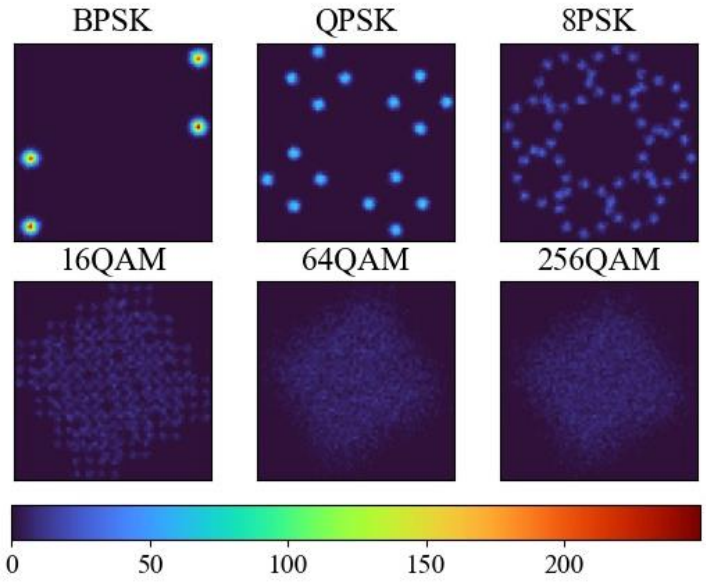

Рисунок 2.5 – Гистограмма IQ диаграмм для шести схем модуляции (SNR=25).

В литературе предложено несколько правил для выбора числа ячеек (b), таких как выбор квадратного корня (где $b = \sqrt{N}$), формула Стёрджеса (где $b = 1 + \log_2 N$), правило Райса (где $b = 2\sqrt[3]{N}$) [115]. Сравнение этих правил представлено на Рисунке 2.7 в результатах.

2.3 Результаты классификации модуляций сложных сигналов MIMO

В этом разделе представлены результаты симуляций для оценки производительности предложенной схемы классификации модуляции. Анализируется система связи MIMO с $N_s = 2$ передающими антеннами и $N_r = 2$ приемными антеннами. Рассматривается канал с замиранием и шумом AWGN, а также шесть схем модуляции: BPSK, QPSK, 8PSK, 16QAM и 64QAM. Извлечение признаков сигнала с использованием взаимной информации рассчитывается путем усреднения 100 симуляций с использованием метода Монте-Карло (Рисунок 2.6).

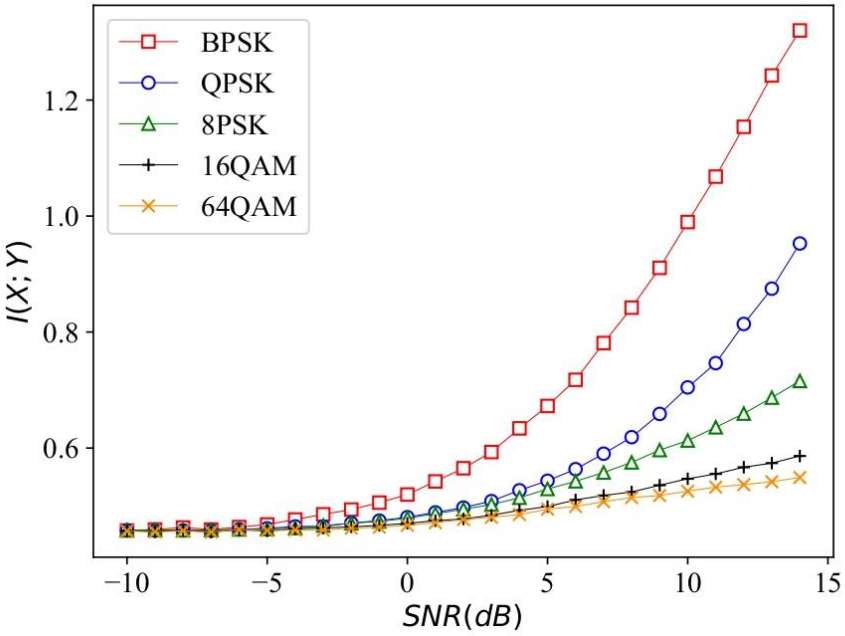

Рисунок 2.6 – Извлечение признаков на основе взаимной информации при различных значениях SNR.

Согласно Рисунку 2.6, по мере увеличения количества символов на диаграмме созвездий расстояние между точками уменьшается, что приводит к увеличению плотности точек на диаграмме и, следовательно, к снижению взаимной информации.

Для расчета вероятности правильной классификации используется следующая формула:

$$P_{cc} = \frac{N_{cc}}{N_{rs}}, \tag{2.3}$$

где, N_{cc} – количество правильно классифицированных сигналов, а N_{rs} – общее количество принятых сигналов. На Рисунке 2.7 показано сравнение существующих правил для выбора числа ячеек.

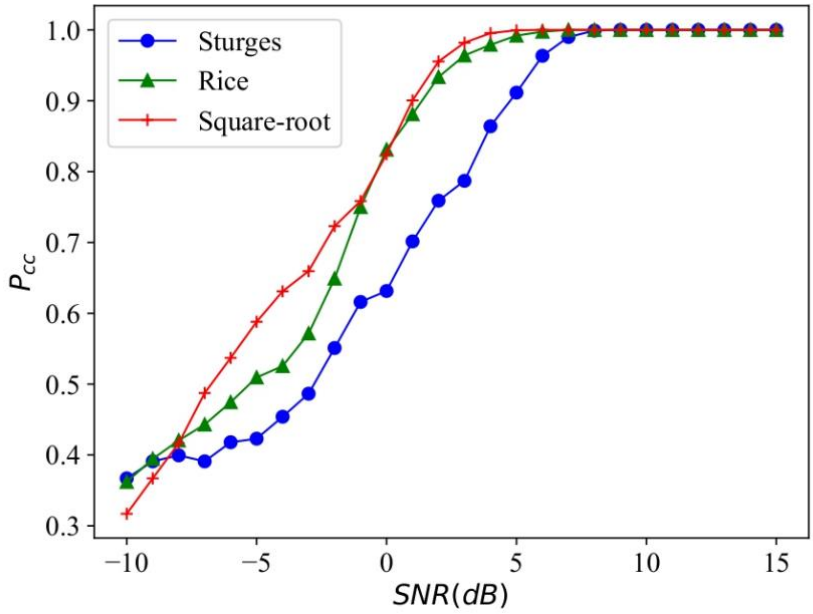

Рисунок 2.7 − Сравнение эмпирических правил для выбора числа ячеек (N=1024)

Согласно Рисунку 2.7, результаты применения правил квадратного корня и Райса демонстрируют лучшую производительность по сравнению с правилом Стёрджеса. Это связано с тем, что с точки зрения интегральной среднеквадратичной ошибки, большее количество ячеек улучшает точность оценки плотности.

Также для предложенного алгоритма оценивалась средняя вероятность правильной классификации различных классификаторов при SNR от -10 до 15, а также рассматривалось время симуляции этих классификаторов (Таблица 1). Вероятность правильной классификации классификатора считается наиболее важным показателем для оценки алгоритма извлечения признаков. В данной работе использовалось пять классификаторов, включая KNN, SVM, Adaboost, Random Forest и MLP для классификации исходного набора признаков, извлеченных предложенным методом. Эти классификаторы оценивались с использованием различных методов определения числа ячеек в гистограмме, включая правила Стёрджеса, Райса и квадратного корня. Параметры для различных классификаторов были установлены следующим образом: для k-ближайших соседей (k-NN) число соседей (k) равно 5. В классификаторе опорных векторов (SVM) использовалось RBF-ядро. Параметры классификатора AdaBoost включали глубину деревьев 10, коэффициент обучения 0,1 и 10 итераций. Классификатор случайного леса (Random Forest) имел глубину деревьев 9, коэффициент обучения 0,1 и 10 итераций. Для многослойного перцептрона (MLP) коэффициент обучения был установлен на уровне 0,1 и было выполнено 10 итераций. В соответствии с набором данных были рассчитаны время симуляции и средняя вероятность правильной классификации различных классификаторов.

Таблица 2.1 − Сравнение производительности классификаторов для предложенного алгоритма

Классификаторы	Sturges		Rice		Square-root	
	P_{cc}	Время (с)	P_{cc}	Время (с)	P_{cc}	Время (с)
SVM	79.21 %	0.56	84.79 %	0.48	85.24 %	0.41
Adaboost	62.60 %	0.25	66.29 %	0.11	68.72 %	0.25
MLP	78.92 %	0.37	83.89 %	0.34	84.37 %	0.62
Random Forest	78.84 %	0.09	81.41%	0.19	82.31%	0.19
KNN	80.25 %	0.10	83.85 %	0.09	84.06 %	0.09

В таблица 2.1 выявлено, что при использовании правила квадратного корня классификаторы SVM, MLP и KNN имеют самую высокую среднюю вероятность правильной классификации по сравнению с методами Стёрджеса и Райса. SVM имеет самую высокую среднюю вероятность правильной классификации, за ним идут MLP и KNN. В то же время классификаторы Adaboost и Random Forest показывают более низкие средние вероятности правильной классификации для всех методов. Что касается времени симуляции, KNN работает быстрее всех трех лучших классификаторов, затрачивая меньше времени по сравнению с SVM и MLP. Хотя SVM требует немного больше времени на симуляцию, его высокая средняя вероятность правильной классификации делает его привлекательным выбором. На основании этих результатов в предложенном методе используется классификатор SVM из-за его высокой средней вероятности правильной классификации и относительно приемлемого времени симуляции. Выбор SVM соответствует цели оптимизации производительности и вычислительной эффективности в рамках предложенного алгоритма.

Рисунок 2.8 демонстрирует результаты симуляций предложенного нами метода с конфигурацией MIMO 2x2 (две передающие и две приемные антенны), сравнивая его с методом на основе схемы D-STBC [116]. Рисунок 2.8 иллюстрирует кривые вероятности правильной классификации P_{cc} в зависимости от различных значений SNR для различных классификаторов, включая классификаторы SVM и KNN.

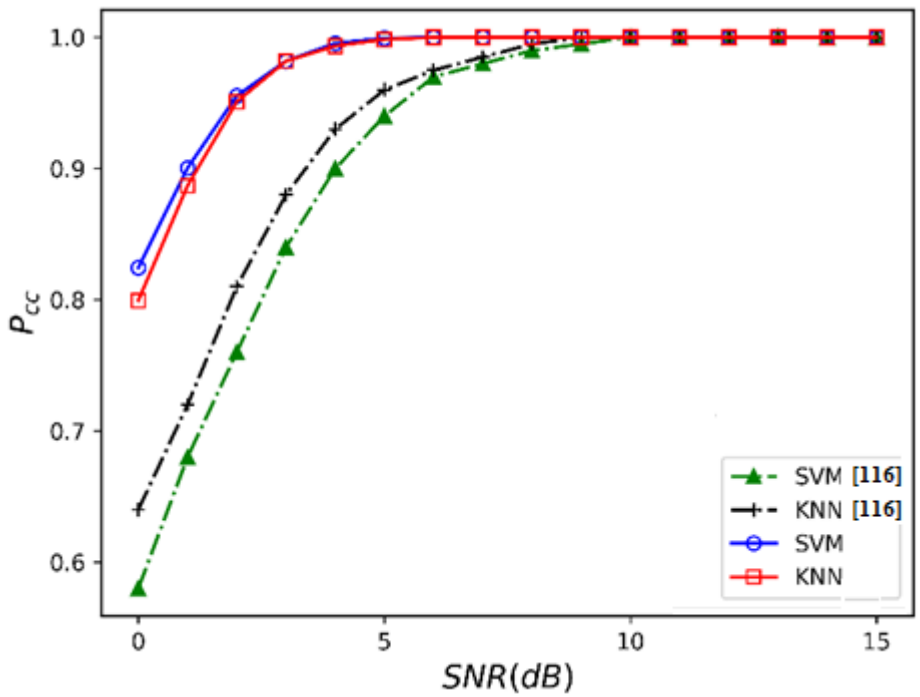

Рисунок 2.8 − Сравнение производительности предложенного метода и метода на основе схемы D-STBC с использованием классификаторов SVM и k-NN.

Предложенный метод превосходит метод на основе схемы D-STBC и достигает идеальной классификации при SNR 5 дБ, в то время как сравниваемый

метод достигает 100% классификации при SNR 13 дБ. Кроме того было проведено сравнение времени симуляции предложенного метода с подходом на основе схемы D-STBC. Для предложенного метода время выполнения составило 0.1 секунды для KNN и 0.56 секунды для SVM, что немного превышает значения подхода на основе схемы D-STBC: 0.07 секунды для KNN и 0.53 секунды для SVM.

Рисунок 2.9 демонстрирует влияние числа ячеек N на производительность предложенного метода при различных значениях SNR, где N = 256, 512, 1024.

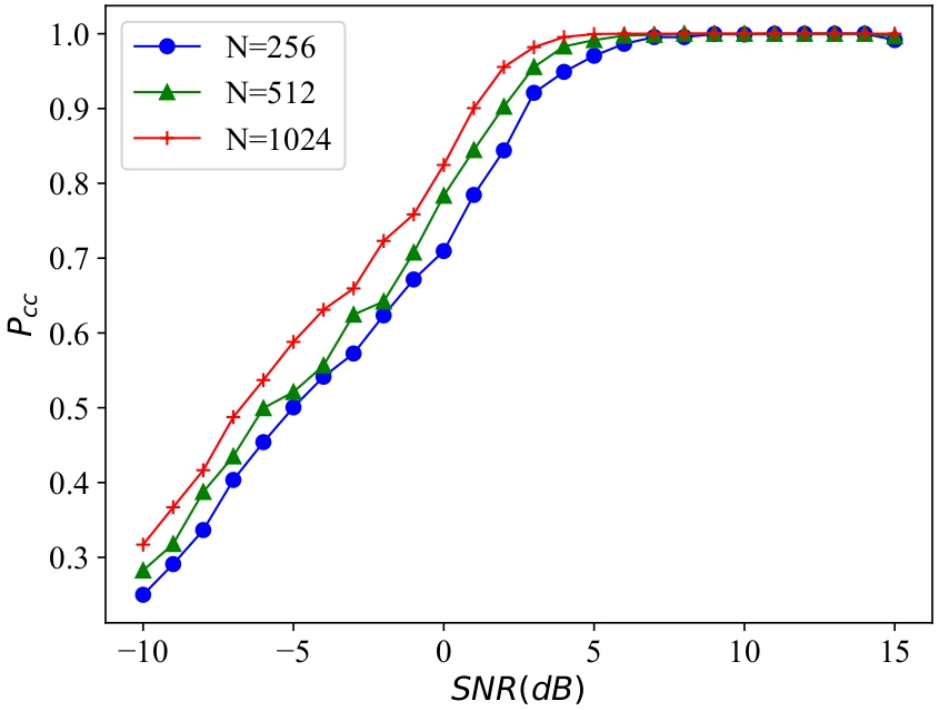

Рисунок 2.9 – Средняя вероятность правильной классификации при различных значениях N

Рисунок 2.9 демонстрирует, что вероятность правильной идентификации увеличивается с ростом SNR, а затем стабилизируется. Четыре типа модуляции: BPSK, QPSK, 8PSK, 16QAM и 64QAM классифицируются правильно с вероятностью 100% при SNR около 5 дБ. Согласно Рисунку 2.9, вероятность правильной классификации P_{cc} изменяется с небольшими ошибками в зависимости от числа точек, поскольку взаимная информация нечувствительна к размеру данных. Рисунок 2.10 ниже показывает сравнение вероятности правильной классификации для различных схем модуляции при N = 1024.

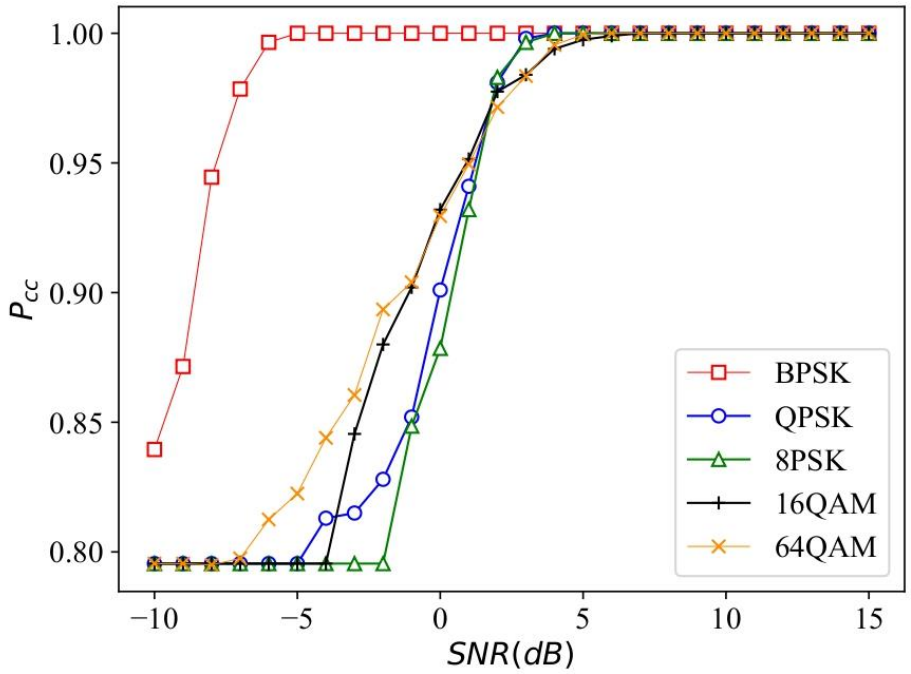

Рисунок 2.10 − Вероятность правильной классификации P_{cc} для различных схем модуляции.

Согласно полученным результатам точность классификации сигналов QAM определяется лучше по сравнению с сигналами PSK, поскольку при увеличении порядка модуляции сигналов QAM точки на диаграмме созвездий распределяются равномернее. Выявлено, что сигналы BPSK можно хорошо идентифицировать с помощью предложенного метода даже при низком SNR = -5 дБ. Сигналы QPSK и 8PSK могут быть правильно распознаны при SNR выше 3 дБ. Идеальное обнаружение сигналов 16QAM и 64QAM достигается при SNR 5 дБ. На рисунке 2.11 представлены вероятности правильной идентификации сигналов BPSK и QPSK, полученные экспериментально.

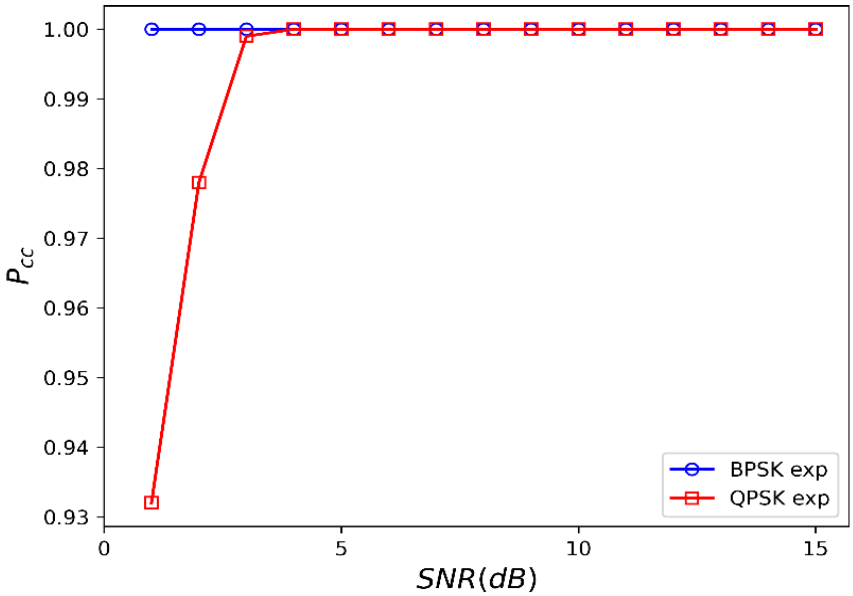

Рисунок 2.11 − Средняя вероятность правильной классификации экспериментальных данных

Как показано на Рисунке 2.11, сигнал QPSK может быть правильно распознан при SNR выше 3 дБ, что соответствует теоретическим результатам.

Вывод к главе 2

В данной главе предложен классификатор, основанный на извлечении признаков взаимной информации для распознавания типов схем модуляции при низком уровне отношения сигнал/шум (SNR) в системах MIMO. Этот метод не требует устранения помех сигналов MIMO и большого объема обучающих данных (шаблонов), так как он более устойчив к неопределенности шума и пространственной корреляции в системах MIMO. Кроме того, следует отметить, что методы извлечения признаков на основе взаимной информации ранее не применялись в алгоритмах автомотической классификации модуляции (AMC) для систем MIMO. Предложенный подход демонстрирует свою эффективность, учитывая взаимозависимости между переменными. Моделирование и экспериментальные результаты показали, что алгоритм способен функционировать корректно даже при низком SNR более чем 5 дБ и небольшом количестве наблюдений, что представляет собой значительный шаг вперед в распознавании схем модуляции в системах MIMO.

3 РАЗДЕЛЕНИЕ WMN НА КЛАСТЕРЫ И ИХ АНАЛИЗ

3.1 Центрвключающий эксцентриситетный алгоритм (CIEA)

Для упрощения понимания предлагаемого Центрвключающего эксцентриситетного алгоритма (CIEA) [117], обозначим размер кластера через l_b, а радиус кластера через r_b, где $l_b = 2r_b+1$. Для сети G, содержащей набор узлов $N = 1, 2,..., n$ и граней $E = 1, 2,..., m$, расстояния между узлами должны быть строго меньше l_b.

С целью устранения недостатков существующих алгоритмов кластеризации (подглава 1.2) в предлагаемом алгоритме CIEA учитывается мера эксцентриситета узлов. Определяется эксцентриситет $e(V)$ узла v в сети G согласно формуле:

$$e(V) = max\{d(v,u|u \epsilon V(G)\} \quad (3.1)$$

В теории графов расстояние определяется как кратчайшее расстояние между узлами v и u. Таким образом, эксцентриситет $e(V)$ узла v в сети G представляет собой максимальное расстояние на графе между между узлами v и u. На рисунке 3.1 представлен пошаговый процесс реализации предлагаемого алгоритма CIEA. На первом шаге вычисляются эксцентриситеты $e(V)$ для всех узлов. Затем выбирается центральный узел, находящийся на расстоянии r_b от узла с максимальным эксцентриситетом. На втором этапе узлы, расположенные на расстоянии r_b от центрального узла, объединяются в один кластер вместе с ним. На следующих шагах эти действия повторяются до тех пор, пока вся сеть не будет поделена на кластеры.

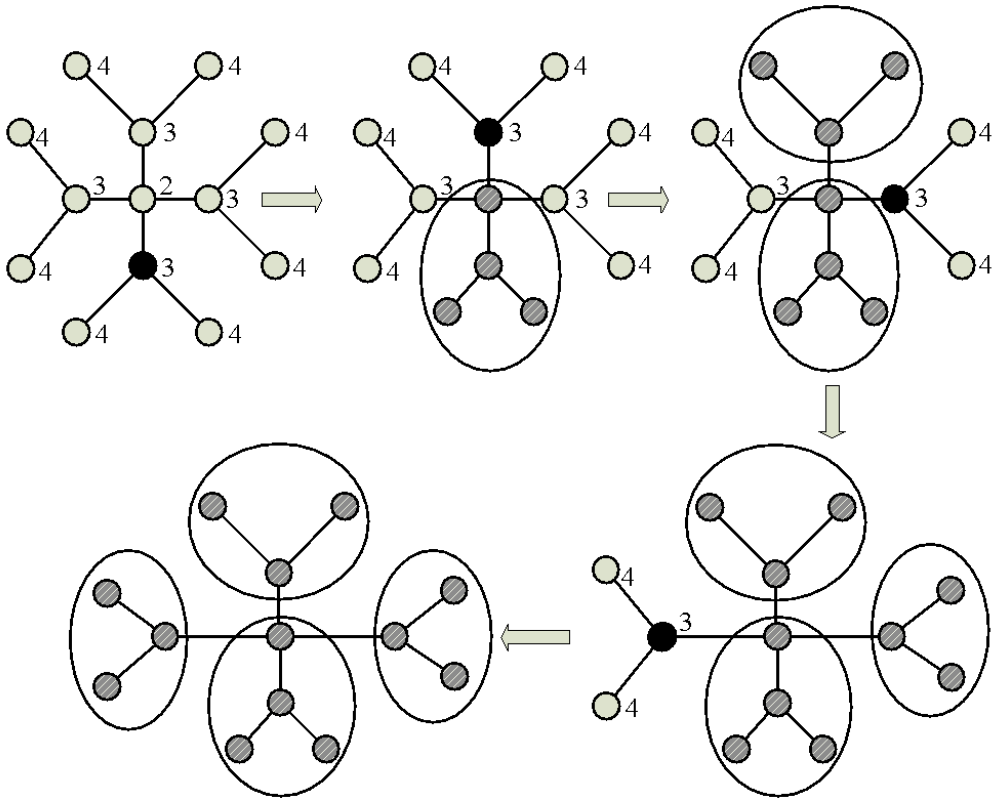

Рисунок 3.1 − Реализация метода CIEA (r_b =1)

Для реализации алгоритма CIEA необходимо выполнить следующие шаги:

1. Вычислить эксцентриситет для каждого узла.

2. От узла с максимальным эксцентриситетом вычесть r_b ($\max(e(V)) - r_b$) и полученный узел отметить центральным.

3. Создать кластер, состоящий из пар узлов в исходной сети, расстояние между которыми равно r_b от выбранного центрального узла.

4. Повторять шаги 2 и 3 до тех пор, пока не будут охвачены все узлы сети.

Структура CIEA аналогична структуре метода MEMB. Тем не менее, CIEA решает задачи с одноузловыми кластерами на краях сети. Кроме того, предлогаемый алгоритм не требует каких-либо сложных вычислений и относительно прост в реализации.

3.2 Кластеризация сети и расчет фрактальной размерности с использованием алгоритма CIEA

Для оценки предлагаемого алгоритма CIEA проведено сравнение с алгоритмами Сонга [50], Чжана [53] и Чжена [48]. В качестве тестовых объектов выбраны реальные социальные сети: Zclub (клуб каратэ), Dolphins (социальная сеть), Polbooks (сеть книг) и Football (футбольная сеть). Результаты сравнительного анализа фрактальных размерностей, рассчитанных с использованием методов Song, Zhang, Zheng и CIEA, представлены в таблице 3.1. Данные в первом, втором и третьем столбцах таблицы заимствованы из работ Вэй Чжэна [48].

Таблица 3.1 − Сравнение фрактальных размерностей, полученных с помощью различных алгоритмов для реальных сетей (Club, Dolphins, Polbooks, football).

Сети	Song	Zhang	Zheng	Предлогаемый метод
Zclub	2.0294	1.3280	1.1347	1.2602
Dolphins	1.8641	1.2993	1.2653	1.1825
Polbooks	2.2846	1.4228	1.2718	1.2708
Football	2.2846	2.5238	2.4949	2.2325

Согласно таблице 3.1, выявлено, что аналогичные значения фрактальной размерности могут быть получены с помощью алгоритма CIEA, что подтверждает его применимость.

Для оценки предлагаемого алгоритма CIEA рассчитывается фрактальная размерность модельной сети UV-flower. Для этого строится модельная сеть UV-flower, где в качестве первого поколения (G1) используется круговой граф, а следующее поколение (G2) получается путём замены каждого узла на два параллельных пути с рёбрами U и V. Эти операции иллюстрируются на рисунке 4.

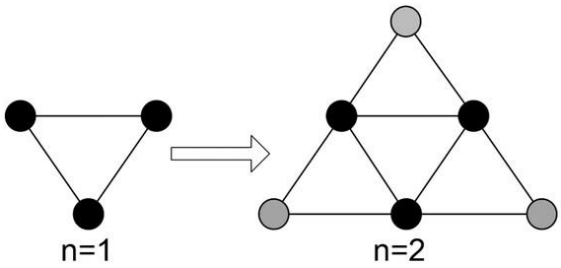

Рисунок 3.2 − Первая и вторая генерации UV- flower модельной сети, где U=1, V=2.

Теоретическая фрактальная размерность сети UV-flower определяется следующей формулой:

$$D = \frac{\ln(U+V)}{\ln U}, \quad U > 1 \tag{3.2}$$

В данном случае параметры модельной сети UV-flower составляют U=2 и V=3. Параметр *n* соответствует шестому поколению, а N − количеству точек, равному 11720. При выполнении теоретического расчета фрактальной размерности по формуле (40) получаем D = 2 321.

Ниже, на Рисунке 3.3, представлены экспериментальные результаты вычисления фрактальной размерности сети UV-flower с использованием алгоритма CIEA.

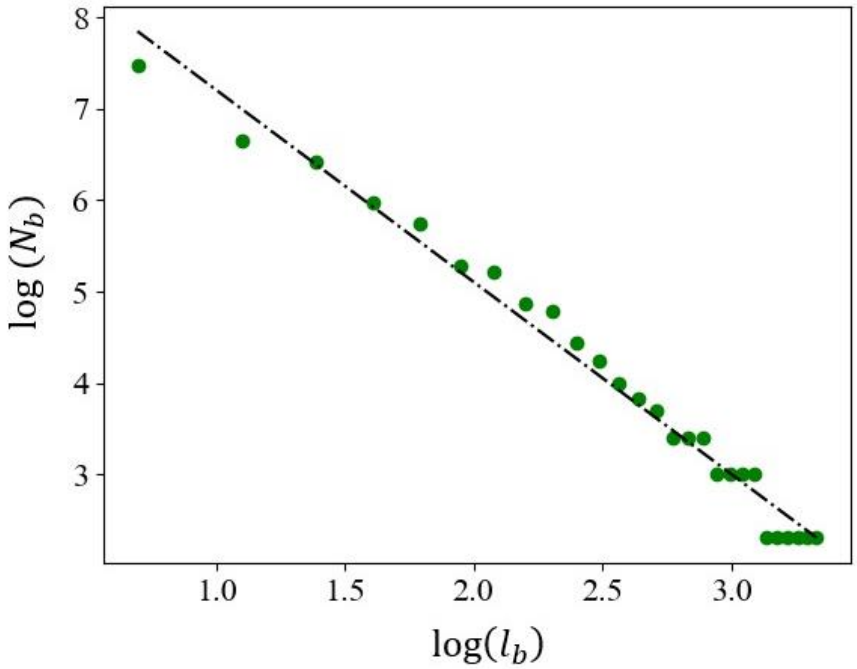

Рисунок 3.3 − Применение алгоритма CIEA к детерминированной сети UV-flower (U=2, V=3, n=6, N=11720, D=2.1).

В таблице 3.2 представлено сравнение полученных экспериментальных расчетов фрактальной размерности известной модельной сети UV-flower, рассчитанных с использованием различных алгоритмов кластеризации.

Таблица 3.2 − Сравнение фрактальных размерностей сети UV-flower, рассчитанных с использованием различных алгоритмов кластеризации.

Алгоритмы	n = 4, N = 470	n = 5, N = 2345	n = 6, N = 11720
CIEA	1.512	1.771	2.100
GC	1.446	1.547	1.684
OBCA	1.447	1.5708	1.754
CBB	1.448	1.535	1.655
MEMB	1.435	1.623	1.859

Согласно данным, представленным в таблице 3.2, значение, наиболее близкое к теоретическому, было получено с помощью алгоритма CIEA в сравнении с другими известными алгоритмами. Это указывает на высокую конкурентоспособность метода CIEA. Кроме того, можно сделать вывод, что алгоритм CIEA обеспечивает более точные результаты с меньшими пределами погрешности.

Для оценки предлагаемого алгоритма CIEA рассчитывается фрактальная размерность реальной сети Escherichia coli (Рисунок 3.4). Фрактальная размерность, полученная с помощью CIEA для реальной сети Escherichia coli, равна D = 2485, при этом количество узлов N=2859 и количество рёбер E=6890.

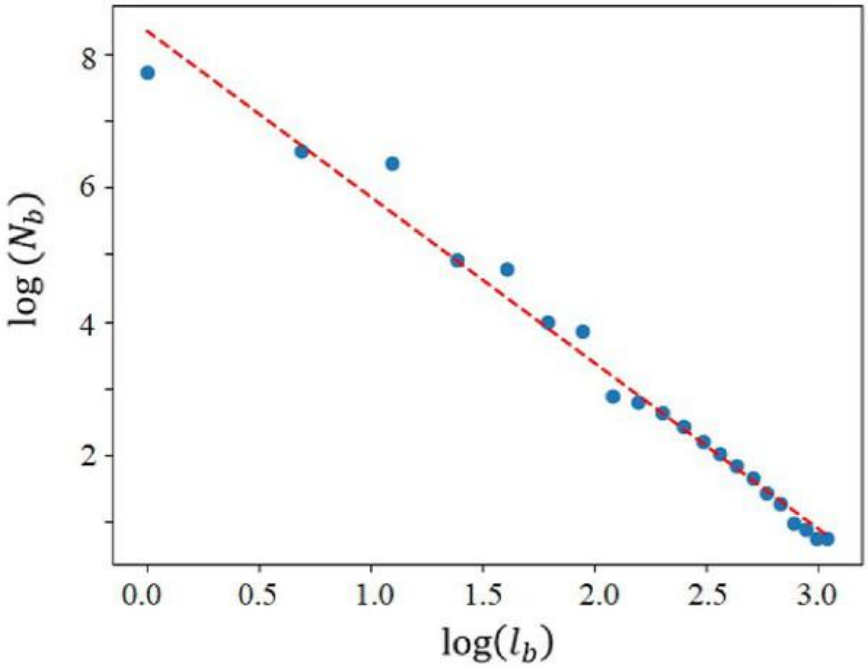

Рисунок 3.4 – Применение алгоритма CIEA к реальной сети Escherichia coli.

В соответствии с Рисунком 3.4, при кластеризации сети Escherichia coli с использованием алгоритма CIEA подтверждается фрактальность сети Escherichia coli, поскольку график представляет собой прямую линию и подчиняется степенному закону, что явно демонстрирует свойство самоподобия [53].

Для оценки предложенного алгоритма CIEA была смоделирована беспроводная сеть из 3000 узлов N, разделённая на кластеры с помощью

алгоритмов CIEA, MEMB и Greedy Coloring (рисунок 3.5). Цель заключалась в определении алгоритма, обеспечивающего покрытие сети с наименьшим числом кластеров.

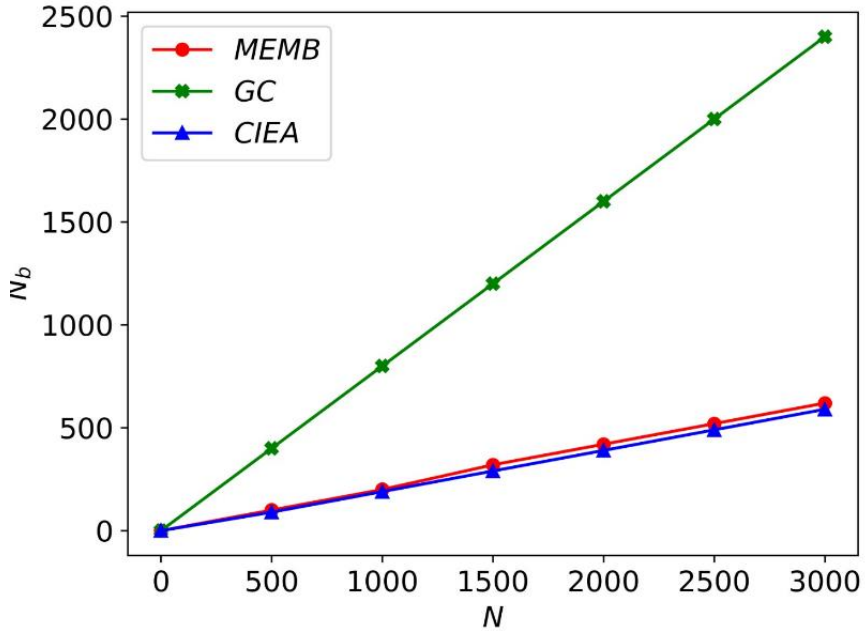

Рисунок 3.5 – Зависимость числа кластеров от размера сети при её покрытии различными алгоритмами кластеризации.

Согласно результатам, представленным на Рисунке 3.5, алгоритм CIEA обеспечивает наилучшее покрытие сети с использованием минимального количества кластеров. Алгоритм MEMB показывает аналогичные результаты, но требует немного большего числа кластеров. В то же время алгоритм Greedy Coloring демонстрирует худшие показатели, значительно увеличивая количество кластеров по сравнению с CIEA и MEMB.

3.3 Применение теории Реньи и Цаллиса для расчета информационной размерности

Для расчета информационной размерности Рени и Цалиса была использована модельная сеть UV-flower. Сеть была разделена на кластеры с помощью различных алгоритмов кластеризации. Результаты расчета информационных размерностей Рени и Цалиса, полученных при использовании известных алгоритмов кластеризации, представлены в Таблице 3.3.

Таблица 3.3. – Информационные размерности Рени, Цалиса модельной сети UV-flower.

Кластерные алгоритмы	Рени размерность при α=0.5	Рени размерность при α=2	Цалиса размерность при q =0.5	Цалиса размерность при q =2
MEMB	1.820	1.674	1.821	1.674
GC	1.823	1.783	1.820	1.798

RS	1.992	1.946	2.042	1.945
CIEA	2.160	2.170	2.158	2.169

Теоретическая фрактальная размерность (D) сети UV-flower, рассчитанная по формуле (3.2), составила 2.321. Сравнение этого теоретического значения с экспериментальными данными, приведёнными в Таблице 3.3, подтверждает оптимальность алгоритма CIEA.

На рисунке 3.6 (а, б) представлена информационная размерность Рени для модельной сети UV-flower, разделенной на кластеры с помощью алгоритма CIEA.

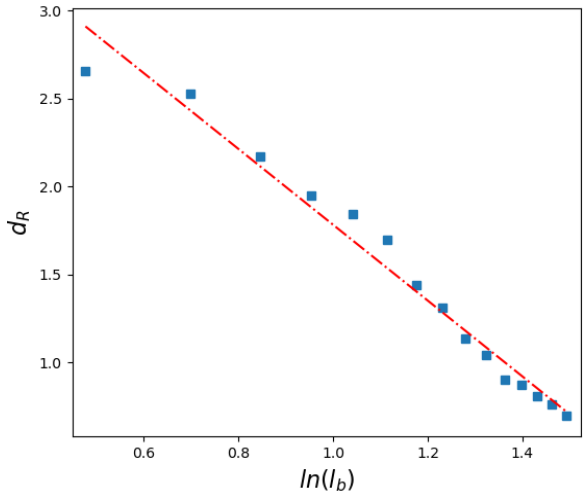

Рисунок 3.6 (а) –Размерность Рени (d_R = 2.16) при разделении сети UV-flower алгоритмом CIEA, при α = 0.5

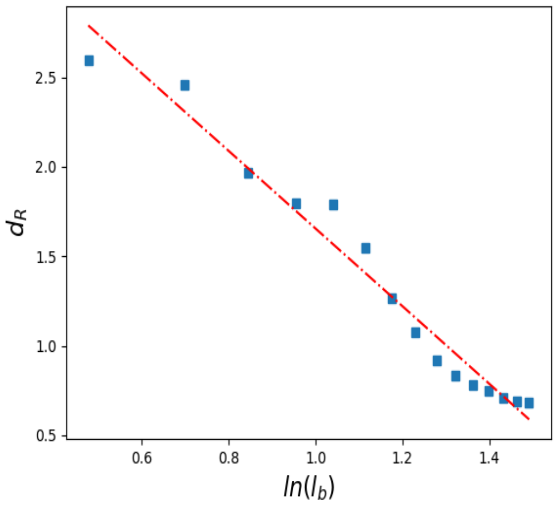

Рисунок 3.6 (б) – Размерность Рени (d_R = 2.17) при разделении сети UV-flower алгоритмом CIEA, при α =2

На рисунке 3.7 (а, б) представлена информационная размерность Цалиса модельной сети UV-flower, разделенной с помощью кластерного алгоритма CIEA.

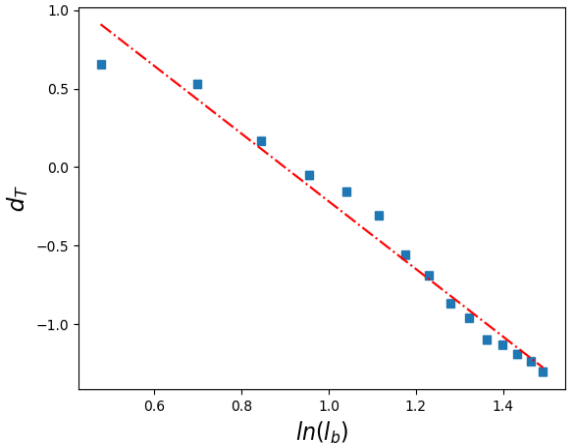

Рисунок 3.7 (а) – Размерность Цалиса (d_T = 2.158) при

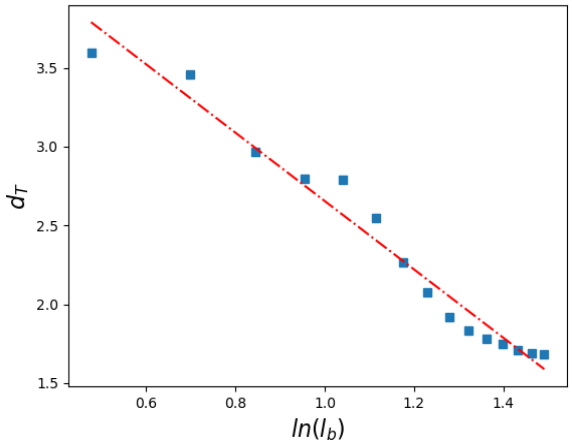

Рисунок 3.7 (б) – Размерность Цалиса (d_T = 2.169) при

разделении сети UV-flower алгоритмом CIEA, при q = 0.5

разделении сети UV-flower алгоритмом CIEA, при q = 2

Согласно результатам, ближайшие значения к теоретической размерности 2.321 достигаются при q и α, равных 2 (d_T = 2,169 и d_R = 2,17), а также немного меньших значениях q и α, равных 0.5.

Помимо выше представленных результатов в данной работе представляется описание самоподобной информационно-структурной среды [118], введенной Жанабаевым Ж. З. В основе, данной теорий, лежит использование экспоненциальной функции вида:

$$exp_{q-1}[x] = (1 + (q-1)x)^{\frac{1}{q-1}}, \quad (3.3)$$

где q-степень однородности, параметр неполноты статистического ансамбля. В пределе $q \to 1$ получаем обычную экспоненту. Для квазиравновесного процесса, характеризуемого параметром q, информацию определим в виде

$$I(x) = -ln_{q-1}P(x) \quad (3.4)$$

Вероятность как функция информации имеет вид:

$$P(I) = exp_{q-1}[-I] = (1 - (q-1)I)^{\frac{1}{q-1}} \quad (3.5)$$

Функция плотности распределения вероятности реализации информации $f(I)$ определяется как:

$$f(I) = \frac{d}{dI} exp_{q-1}[-I] = (1 - (q-1)I)^{\frac{2-q}{q-1}}. \quad (3.6)$$

Для неподвижной точки $f(I) = I$ при $q \to 0$ имеем:

$$I(1+I)^2 = I + 2I^2 + I^3 = 1 \quad I = I_0 = 0.465 \quad (3.7)$$

При значении параметра $q \to 0$ в статистике Цаллиса самоподобное значение информации стремится к значению $I_0 = 0.465$. Критерий I_0 соответствует предельному случаю масштабной инвариантности в динамике хаотических систем, т.е. переходу от хаоса к масштабно-инвариантной структурности (фрактальности). Данный факт может служить ориентиром для выбора оптимального маршрута в телекоммуникационной сети. Полученная размерность I_0 описывает направленный (односторонний) процесс, что соответствует оптимальной маршрутизации.

3.4 Кластерный маршрутизатор на основе алгоритма CIEA

Кластерный маршрутизатор на основе алгоритма CIEA [119] оснащен контроллером с матрицей коммутации, двунаправленными портами для подключения к управляющему автомату, а также программируемой логической интегральной схемой (ПЛИС) с энергонезависимой памятью, что позволяет осуществлять маршрутизацию по нескольким подсетям. На рисунке 3.8 изображена блок-схема кластерного маршрутизатора, включающего контроллер с коммутационной матрицей 1, двунаправленный порт 2 для подключения к системе автоматизации управления 3 и программируемую логическую интегральную схему (ПЛИС) 4 с энергонезависимой памятью 5.

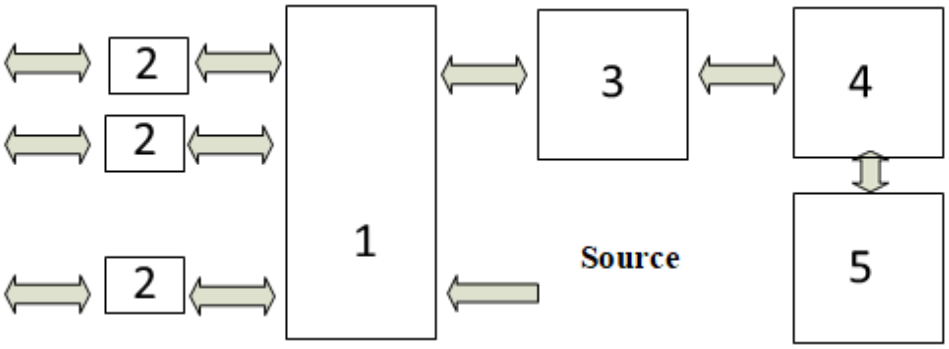

Рисунок 3.8 − Блок-схема кластерного маршрутизацатора на основе алгоритма CIEA

Известные маршрутизаторы строят маршрут по всей сети преимущественно используя алгоритм Dijkstra. Однако эти маршрутизаторы неэффективны при большом количестве сетевых узлов. В отличие от них, кластерный маршрутизатор на основе алгоритма CIEA делит сеть на кластеры, рассчитывая эксцентриситет узлов, что позволяет уменьшить количество операций по поиску кратчайшего пути в сетевой топологии.

Для описания кластерного маршрутизатора обозначим размер сетевого кластера как l_b и радиус кластера как r_b, при этом $l_b = 2r_b+1$. Пусть G – это сеть (Рисунок 3), содержащая набор узлов $N = \{1, 2, ..., n\}$ и ребра $E = \{1, 2, ..., m\}$, при условии, что расстояния между маршрутизаторами должны быть строго меньше l_b. Далее определяется эксцентриситет узлов $e(v)$ в связной сети G по формуле (3.1). На рисунке 3.9 показана реализация разделения сети на кластеры с учётом эксцентриситета на примере сети из 13 узлов и 12 рёбер.

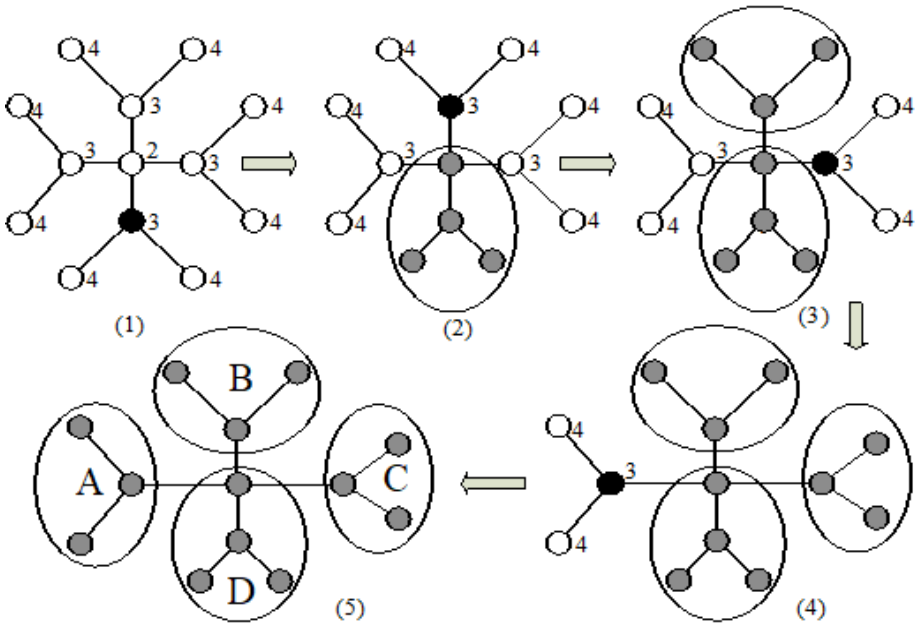

Рисунок 3.9 − Разделение сети G на кластеры (A, B, C, D) на основе алгоритма CIEA (r_b=1). Белые узлы − не покрытые, серые − покрытые, чёрные − центры. Числа в узлах обозначают значения эксцентриситета.

На рисунке 3.9 первым шагом определяется эксцентриситет *e(V)* узлов, после чего в качестве центрального узла выбирается узел, расположенный на расстоянии r_b от узла с максимальным эксцентриситетом (max(*e(V)*) − r_b). На втором шаге узлы, расположенные на расстоянии r_b от центрального узла, покрываются одним кластером с центральным узлом. Затем эти операции повторяются, пока вся сеть не будет разделена на кластеры. После разбиения сети на кластеры (A, B, C, D) с использованием алгоритма CIEA (рисунок 3.9), каждый кластер представляется как один узел (рисунок 3.10 (а)). Сначала определяется кратчайший путь между кластерами, а затем − между узлами внутри кластеров, как показано на рисунках 3.9 и 3.10 (а, б). Такой подход минимизирует сложность поиска маршрутов, представляя сеть в виде групп узлов (кластеров), что упрощает определение кратчайших путей.

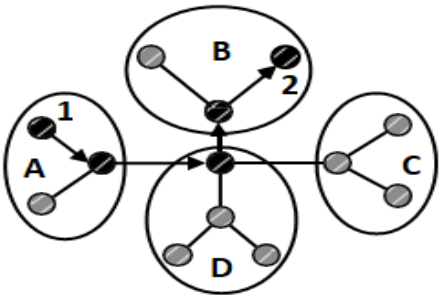

Рисунок 3.10 (а) − Кратчайший путь между кластерами сети

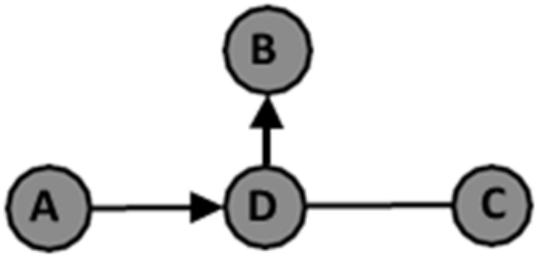

Рисунок 3.10 (б) − Кратчайший путь между кластерами сети r_b=1.

Для оценки эффективности предложенного маршрутизатора была выполнена интеграция алгоритма CIEA с алгоритмом Dijkstra, получившая название BCR (Box Covering Based Routing). На рисунке 3.11 представлены результаты сравнительного анализа эффективности алгоритмов маршрутизации BCR и Dijkstra. В таблице 5 представлено точное время выполнения алгоритмов BCR и Dijkstra при различных размерах сети (в микросекундах).

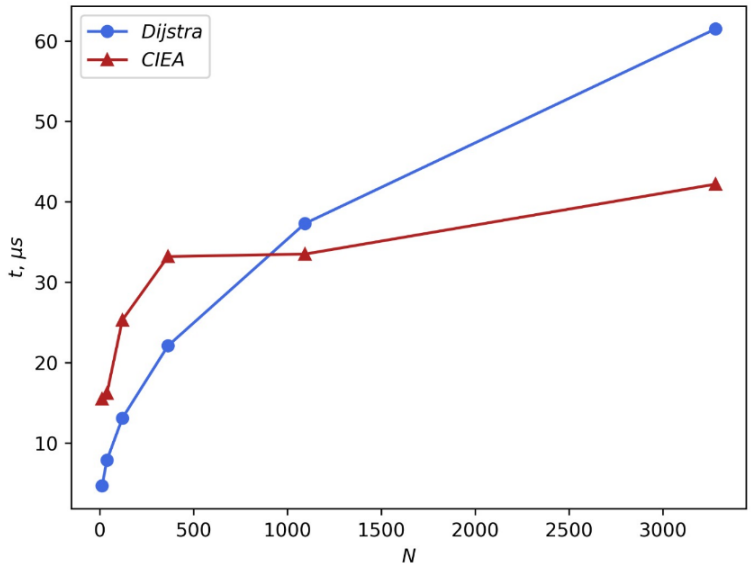

Рисунок 3.11− Время поиска кратчайшего пути для алгоритмов Dijkstra и BCR в зависимости от количества узлов.

Таблица 3.4 − Сравнение времени поиска кратчайшего пути для алгоритмов Dijkstra и BCR (в микросекундах).

Количество узлов	Dijkstra алгоритм	BCR алгоритм
13	4.68	15.5
39	7.89	16.2
121	13.1	25.3
364	22.1	33.2
1093	37.3	33.5
3280	61.5	42.2

Согласно рисунку 55 и таблице 3.4, алгоритм Dijkstra показывает хорошие результаты при небольшом количестве узлов, но его быстродействие значительно снижается с увеличением размера сети. В то же время BCR алгоритм демонстрирует более высокую скорость выполнения, особенно при 3280 узлах, где он превосходит алгоритм Dijkstra на 20 микроскунд, подтверждая эффективность интеграции методов кластеризации и маршрутизации.

Вывод по главе 3

Разработка и анализ алгоритма кластеризации CIEA для беспроводных сетей показали его эффективность и конкурентоспособность по сравнению с существующими методами. В частности, CIEA учитывает эксцентриситет узлов и позволяет более эффективно формировать кластеры, что улучшает покрытие сети и снижает количество необходимых вычислений при маршрутизации. Сравнительный анализ фрактальных размерностей, рассчитанных с использованием алгоритма CIEA и других методов, продемонстрировал его применимость для реальных социальных сетей, таких как Zclub, Dolphins, Polbooks и Football. Дальнейшие эксперименты с модельной сетью UV-flower и реальной сетью Escherichia coli подтвердили точность алгоритма CIEA в расчете фрактальной размерности. Кроме того, результаты кластеризации в беспроводной сети из 3000 узлов продемонстрировали, что алгоритм CIEA обеспечивает наилучшее покрытие сети с минимальным числом кластеров. Применение теории Реньи и Цаллиса также подтвердило, что алгоритм CIEA эффективно разделяет сеть на кластеры, обеспечивая точные значения информационных размерностей. В частности, результаты при различных параметрах α и q показали близость к теоретической размерности, что подчеркивает его надежность. Кластерный маршрутизатор, основанный на алгоритме CIEA, обеспечивает более эффективное решение задач маршрутизации в крупных сетях, минимизируя сложность поиска кратчайшего пути. Таким образом, алгоритм CIEA демонстрирует высокую степень

адаптивности и является полезным инструментом для улучшения производительности беспроводных сетей, обеспечивая не только оптимизацию маршрутов, но и более эффективное управление ресурсами.

4 МЕТОДОЛОГИЯ МАРШРУТИЗАЦИИ И ЕЕ РЕАЛИЗАЦИЯ В WMN

4.1 Создание WMN топологии в среде Python

Топология беспроводных сетей играет ключевую роль в обеспечении эффективной передачи данных и стабильности соединения. От выбора структуры сети зависят такие параметры, как пропускная способность, задержки, отказоустойчивость и энергопотребление. Существует несколько подходов к организации топологий WMN, включая случайную, ячеистую и визуализацию графов с помощью алгоритма Фрухтермана-Рейнгольда, используемого для анализа сетевой структуры [120]. В данной работе топологии WMN реализованы в NS-3, включая случайную и ячеистую структуры, что позволяет моделировать различные сценарии взаимодействия узлов в сети. Для анализа сетевой структуры используется визуализация графов с помощью алгоритма Фрухтермана-Рейнгольда. Этот алгоритм помогает наглядно представить взаимосвязи между узлами и минимизировать перекрытия, что облегчает исследование производительности и стабильности сети в разных топологиях.

Случайная топология в беспроводных сетях характеризуется тем, что узлы располагаются произвольным образом, без определенной структуры или фиксированных связей. Такое расположение может быть полезным в условиях динамически меняющейся среды, где узлы часто перемещаются или выходят из строя. Преимущества случайной топологии включают высокую гибкость и простоту развертывания, особенно в сценариях, где невозможно заранее спланировать размещение узлов, например, в аварийных ситуациях или временных сетях. Однако случайное расположение может привести к неравномерному распределению нагрузки и снижению пропускной способности, так как некоторые каналы могут иметь высокие вероятности ошибок из-за ухудшенного качества связи. Поэтому важно учитывать вероятности ошибок при маршрутизации, чтобы улучшить эффективность сети. Ниже, на рисунке 4.1, построена случайная сеть, которая впоследствии используется при исследовании алгоритмов маршрутизации [120].

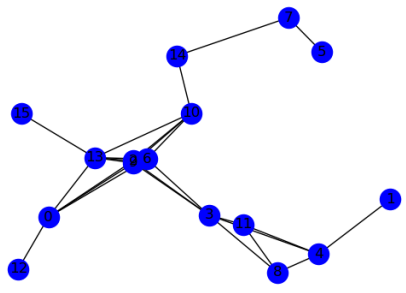

Рисунок 4.1 (а) – Случайная топология 4x4 (16 узлов)

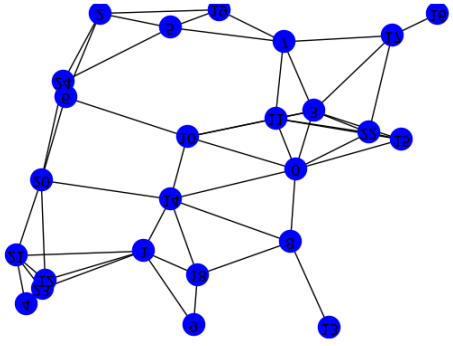

Рисунок 4.1 (б) – Случайная топология 5x5 (25 узлов)

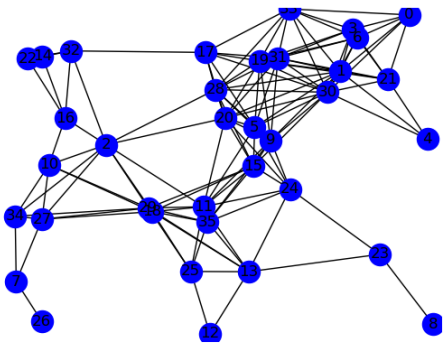

Рисунок 4.1 (в) − Случайная топология 6x6 (36 узлов)

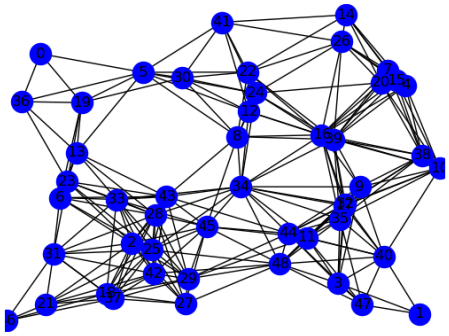

Рисунок 4.1 (г) − Случайная топология 6x6 (36 узлов)

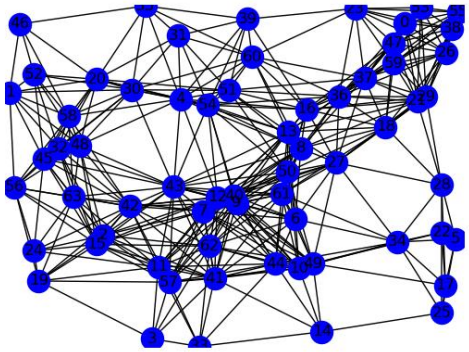

Рисунок 4.1 (д) − Случайная топология 8x8 (64 узла) (1)

Ячеистая топология представляет собой структурированное расположение, при котором узлы организованы по принципу сетки и соединяются с ближайшими соседями. Это обеспечивает плотную сеть взаимосвязанных узлов, создавая несколько резервных путей для передачи данных, что повышает надежность сети. Если один маршрут становится недоступным, другие альтернативные пути могут быть использованы для минимизации перебоев в связи.

Одним из ключевых преимуществ ячеистой топологии является легкость масштабирования. Добавление новых узлов в заранее определенные места позволяет расширить покрытие и улучшить подключение, что делает топологию особенно подходящей для развертывания на больших территориях. Низкая задержка передачи данных, обусловленная близким расположением узлов, делает такую сеть пригодной для приложений с требованием к реальному времени, таких как видеопотоки и онлайн-программы [120].

Предсказуемая организация узлов упрощает планирование сети и ее развертывание. Размещение узлов можно оптимизировать для равномерного покрытия, что сводит к минимуму появление мертвых зон и обеспечивает стабильное качество сигнала по всей территории. Однако ячеистая топология требует наличия фиксированной инфраструктуры, такой как опоры или здания, для установки узлов, что может усложнить развертывание в некоторых условиях.

Кроме того, в условиях высокой плотности населения или сильных беспроводных помех близкое расположение узлов может привести к увеличению уровня интерференции, отрицательно сказываясь на производительности сети. Стоимость развертывания может также быть выше по сравнению с более простыми или децентрализованными топологиями, так как требуется больше оборудования и затрат на установку. Ниже, на рисунке 4.2, представлена ячеистая сеть, которая затем используется при рассмотрении алгоритмов маршрутизации.

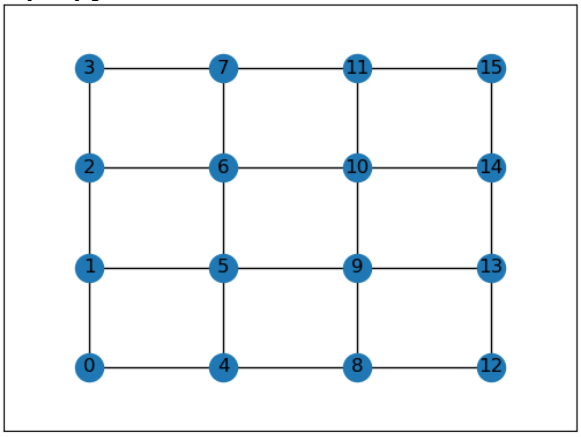

Рисунок 4.2 (а) – Ячеистая топология 4x4 (16 узлов)

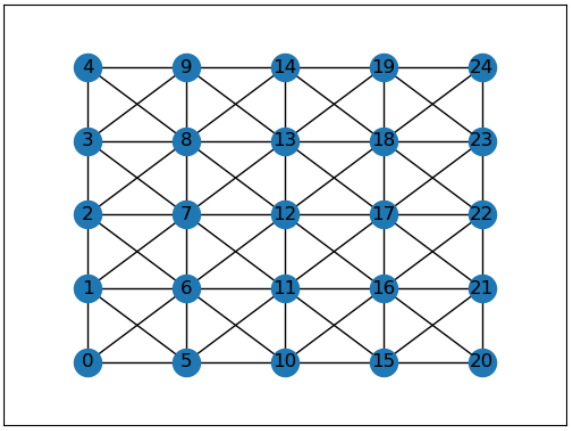

Рисунок 4.2 (б) – Ячеистая топология 5x5 (25 узлов)

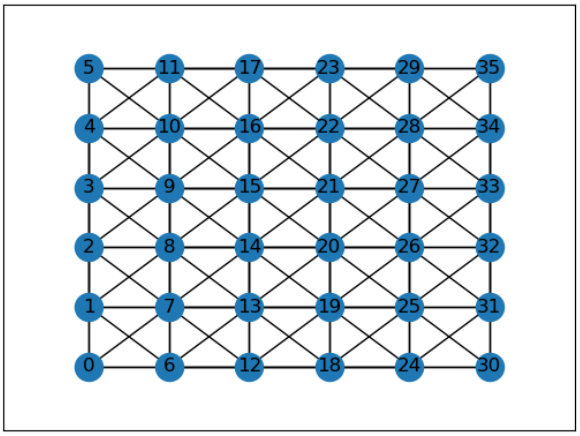

Рисунок 4.2 (в) – Ячеистая топология 6x6 (36 узлов)

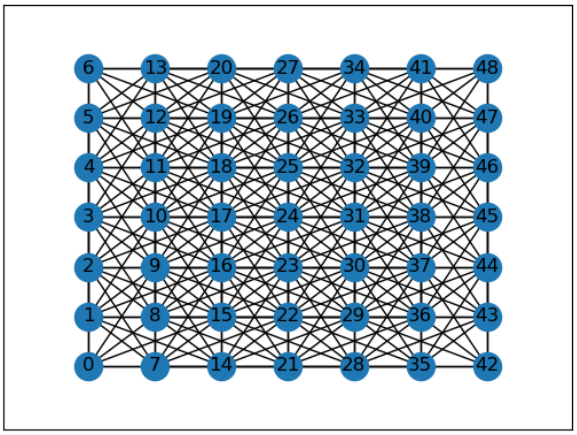

Рисунок 4.2 (г) – Ячеистая топология 7x7 (49 узлов)

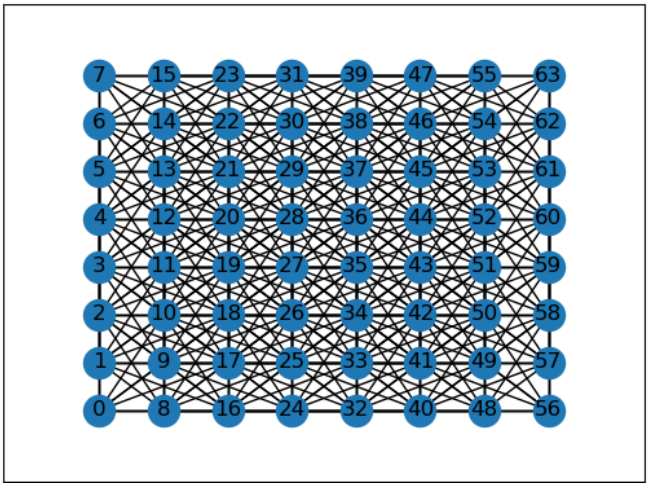

Рисунок 4.2 (д) – Ячеистая топология 8x8 (64 узла) (2)

Топология Фрухтермана-Рейнгольда, разработанная Томасом М. Дж. Фрухтерманом и Эдвардом М. Рейнгольдом, представляет собой алгоритм компоновки графов, часто применяемый для визуализации сетей. В отличие от типичных топологий WMN, таких как случайная или сеточная, этот алгоритм используется для удобного представления узлов и их взаимосвязей. В алгоритме моделируется физическая система, в которой узлы (вершины графа) отталкиваются друг от друга, а соединяющие их связи (ребра) притягиваются, подобно пружинам. Это позволяет достичь равновесной конфигурации, в которой узлы распределены равномерно, а пересечения связей сведены к минимуму. Такой подход делает алгоритм полезным для анализа сложных сетевых структур, так как он наглядно выявляет ключевые взаимосвязи и узлы.

Хотя алгоритм Фрухтермана-Рейнгольда не используется непосредственно для управления работой WMN, он служит важным инструментом для сетевых инженеров, позволяя визуализировать топологию, анализировать сеть, выявлять слабые места и оптимизировать ее конфигурацию. Это способствует лучшему планированию и устранению неполадок, а также может помочь в принятии решений по улучшению маршрутизации и распределению нагрузки в сети. На рисунке 4.3 ниже представлена топология, созданная с использованием алгоритма Фрухтермана-Рейнгольда и впоследствии использованная для изучения алгоритмов маршрутизации [120].

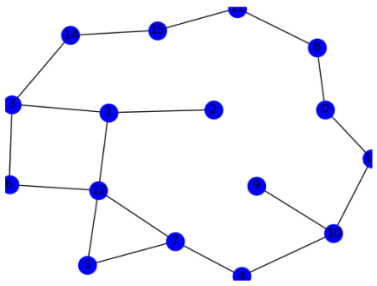

Рисунок 4.3 (а) – Топология Фрухтермана-Рейнгольда 4x4 (16 узлов)

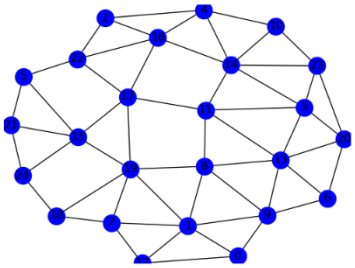

Рисунок 4.3 (б) – Топология Фрухтермана-Рейнгольда 5x5 (25 узлов)

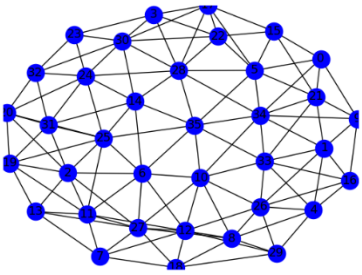

Рисунок 4.3 (в) – Топология Фрухтермана-Рейнгольда 6x6 (36 узлов)

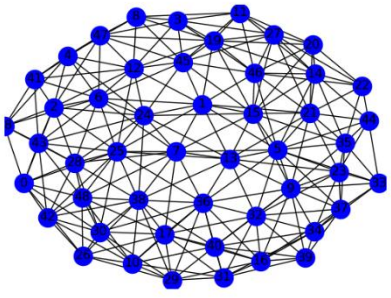

Рисунок 4.3 (г) – Топология Фрухтермана-Рейнгольда 7x7 (49 узлов)

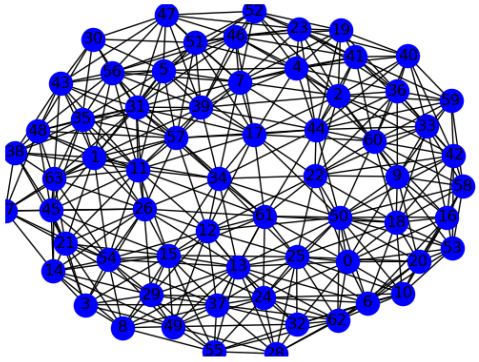

Рисунок 4.3 (д) − Топология Фрухтермана-Рейнгольда 8x8 (64 узла) (3)

WMN преимущественно использует ячеистую топологию, где узлы размещаются в виде сетки, обеспечивая стабильные и надежные соединения, а также повышенную отказоустойчивость. Такая структура идеально подходит для сценариев с высоким трафиком и обеспечением гарантированного качества связи, где важны низкие задержки и высокая пропускная способность. Однако для исследования сетевых алгоритмов и моделирования различных условий работы сети также применяются другие топологии, такие как случайная и визуализируемая с помощью алгоритма Фрухтермана-Рейнгольда. Случайная топология позволяет исследовать сети в динамичных и изменяющихся средах, что особенно полезно для анализа сетевого поведения в нестабильных условиях, когда топология сети может изменяться из-за движения узлов или изменения состояния соединений. В свою очередь, ячеистая топология гарантирует низкие задержки и высокую отказоустойчивость, что делает её оптимальной для приложений, требующих реального времени, таких как системы мониторинга или управление транспортом. Алгоритм Фрухтермана-Рейнгольда, применяемый для визуализации этих топологий, эффективно отображает связи между узлами, позволяя провести оптимизацию планирования сети и выявить критические точки для улучшения работы сети, таких как перегруженные узлы или узкие места в соединениях. Все эти топологии играют ключевую роль в анализе и разработке алгоритмов маршрутизации, что способствует улучшению общей производительности и надежности сети, а также предоставляет гибкие подходы к решению специфических задач в разных условиях эксплуатации.

4.2 Сравнительный анализ классических протоколов AODV, DSDV и OLSR в NS-3

В данном разделе проводится сравнительный анализ классических протоколов маршрутизации AODV, DSDV и OLSR в различных топологиях WMN, с использованием сетевого симулятора NS-3. Моделирование осуществляется с применением стандарта IEEE 802.11p, который является расширением IEEE 802.11, специально разработанным для беспроводной связи в условиях транспортных сред (WAVE). Это расширение учитывает специфические требования, связанные с высокой мобильностью устройств,

обеспечивая эффективную передачу данных в таких условиях, как дороги и автомагистрали. Для более точного моделирования потерь сигнала используется модель затухания Two-Ray Ground Propagation Loss Model. Эта модель описывает распространение сигнала с учетом как прямой волны, так и отраженной от земли, что позволяет более точно учитывать реальное поведение сигнала в условиях наружных пространств.

Оценка эффективности четырех упомянутых протоколов маршрутизации будет основана на таких метриках, как пропускная способность, задержка и потеря пакетов, что позволит провести всестороннюю оценку их производительности и пригодности для WMN. Для анализа влияния количества узлов на производительность этих протоколов были проведены симуляции с использованием NS-3.40 (ver3.14.1) на платформе Ubuntu 22.04 LTS. Параметры симуляции представлены в Таблице 4.1.

Таблица 4.1 – Параметры, используемые в программе NS-3.

Сетевой симулятор	NS-3.40 (ver3.14.1)
Тип канала	Беспроводной канал (Wireless channel)
Модель распространения	Friis Propagation Loss Model
Тип сетевого интерфейса	Phy/Wireless Phy
Тип MAC	Mac/802.11ac
Тип интерфейса	Drop Tail/PriQueue
Тип канального уровня	LL
Модель антенны	Одна антенна (Single Antenna)
Тип трафика	CBR
Транспортный протокол	UDP
Время моделирования	50 s
Размер пакета	1024
Область моделирования	1000m*1000m
Модель мобильности	Постоянная (Constant)
Протоколы	OLSR, AODV, DSDV
Количество узлов в топологиях случайная, ячеистая и Фрухтермана-Рейнгольда	16, 25, 36, 49, 64

Результаты измерения пропускной способности отражают объем данных, успешно переданных по сети за определенный промежуток времени. Пропускная способность рассчитывается по следующей формуле:

$$Throughput = (\sum successfully\ receiver\ bits\ /\ Time\ of\ simulation\ \times 1024) \quad (4.1)$$

Для достижения наилучших результатов пропускная способность должна быть максимальной. На рисунке 4.4 представлена пропускная способность, достигнутая алгоритмом маршрутизации AODV в различных топологиях WMN. Конфигурации случайной топологии и топологии Фрухтермана-Рейнгольда

демонстрируют наивысшую пропускную способность при оптимальном числе узлов, равном 50. Эти результаты подчеркивают их эффективность в условиях эксперимента, свидетельствуя о более высоком уровне производительности и надежности по сравнению с ячеистой топологией.

На рисунке 4.5 показано, что алгоритм маршрутизации OLSR достигает более высокой пропускной способности при оптимальном числе узлов (50) в случайной топологии, по сравнению с ячеистой топологией и топологией Фрухтермана-Рейнгольда. Это подчеркивает превосходство случайной топологии в обеспечении более высокой скорости передачи данных, что может быть полезно в динамических и изменяющихся условиях сети.

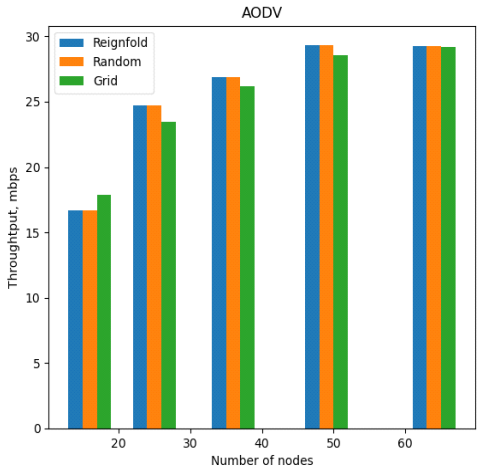

Рисунок 4.4 – Пропускная способность для алгоритма маршрутизации AODV

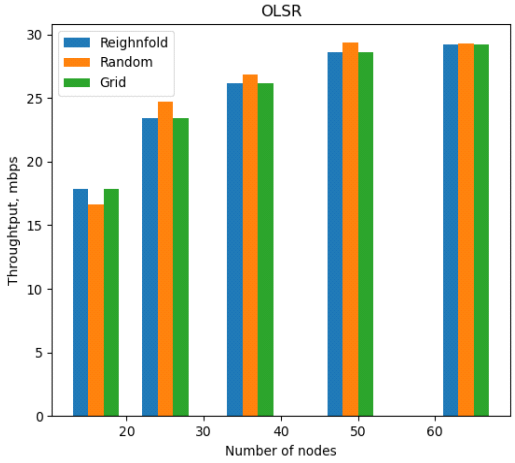

Рисунок 4.5 – Пропускная способность для алгоритма маршрутизации OLSR

На следующем рисунке 4.6 алгоритм маршрутизации DSDV демонстрирует превосходную производительность в топологиях Фрухтермана-Рейнгольда и случайной при оптимальном числе узлов (50) по сравнению с ячеистой топологией. Эти результаты свидетельствуют о более высоких показателях пропускной способности, что указывает на эффективность алгоритма в условиях этих топологий и преимущества в обеспечении более высокой скорости передачи данных.

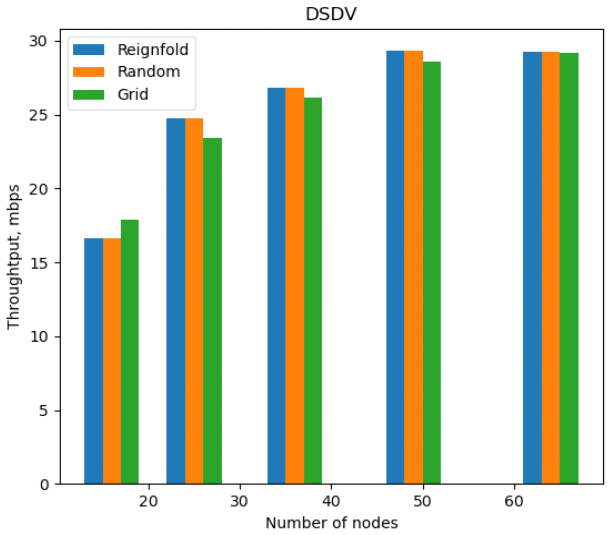

Рисунок 4.6 – Пропускная способность для алгоритма маршрутизации DSDV

Потеря пакетов – это доля данных, которые не достигают своей целевой точки назначения из-за различных факторов, таких как перегрузка сети, отказ соединения или помехи. Потеря пакетов определяется по следующей формуле:

$$Packet\ Lost = (\sum Packets\ send\ by\ sources\ \sum Packets\ successfully\ received) \quad (4.2)$$

Низкое значение потери пакетов указывает на высокую эффективность работы протокола. На рисунке 4.7 показаны уровни потерь пакетов для алгоритма маршрутизации AODV в различных топологиях. Было установлено, что наименьшие потери пакетов наблюдаются в ячеистой топологии при количестве узлов 50, по сравнению с двумя другими топологиями. На рисунке 4.8 представлены уровни потерь пакетов для алгоритма маршрутизации OLSR в различных топологиях. Потери пакетов снижаются в случайной топологии при оптимальном числе узлов 50, что свидетельствует о более высокой эффективности этой топологии по сравнению с остальными двумя.

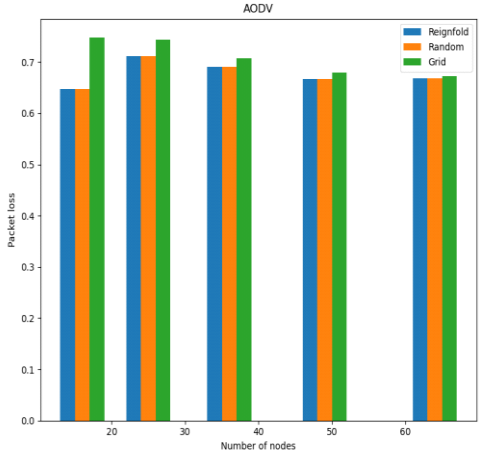

Рисунок 4.7 – Потеря пакетов для алгоритма маршрутизации AODV

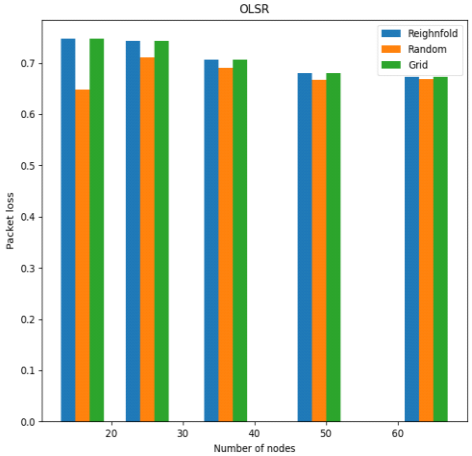

Рисунок 4.8 – Потеря пакетов для алгоритма маршрутизации OLSR

Рисунок 4.9 иллюстрирует уровни потерь пакетов в разных топологиях. Установлено, что для алгоритма маршрутизации DSDV топологии Фрухтермана-Рейнгольда и случайная показывают наименьшие уровни потерь пакетов в большинстве протестированных сценариев. Оптимальное количество узлов, при котором достигалась минимальная потеря пакетов, составило 50.

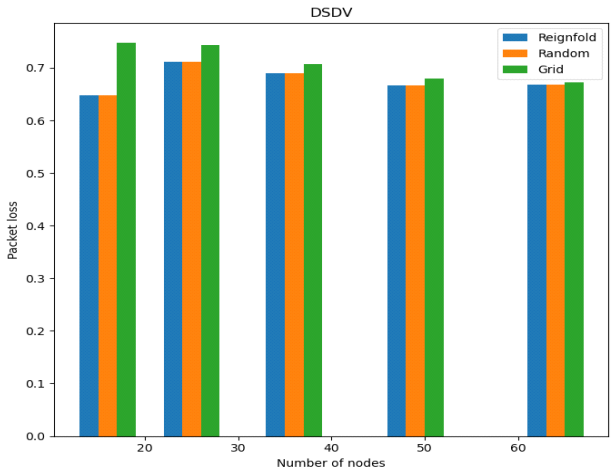

Рисунок 4.9 – Потеря пакетов для алгоритма маршрутизации DSDV

Задержка времени характеризует интервал, необходимый для передачи данных от отправителя к получателю. Задержка времени рассчитывается по следующей формуле:

$$Time\ delay = |EndTime_i - StartTime_i|, \quad (4.3)$$

где $EndTime_i$ – это время, когда пакет i был отправлен источником и успешно получен приемником, а $StartTime_i$ – это время начала отправки пакета i источником.

Рисунок 4.10 демонстрирует задержку времени, с которой сталкиваются пакеты данных при использовании алгоритма маршрутизации AODV в различных топологиях. При оптимальном количестве узлов 60 наименьшая задержка времени наблюдается как в случайной, так и в топологии Фрухтермана-Рейнгольда.

Рисунок 4.11 иллюстрирует задержку времени для пакетов данных при использовании алгоритма маршрутизации OLSR в различных топологиях. На данном рисунке видно, что с увеличением числа узлов задержка времени снижается. При оптимальном количестве узлов 60 случайная топология показывает наименьшую задержку времени по сравнению с остальными.

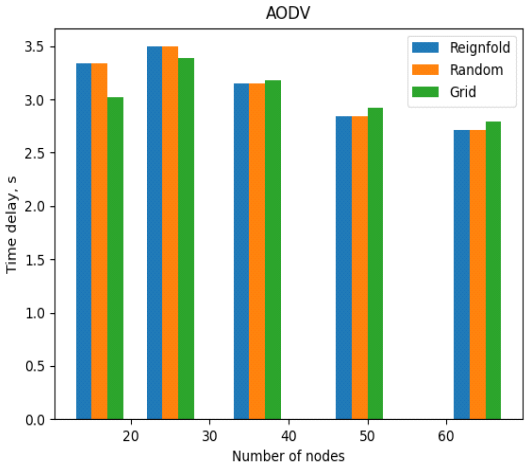

Рисунок 4.10 – Задержка времени для алгоритма маршрутизации AODV

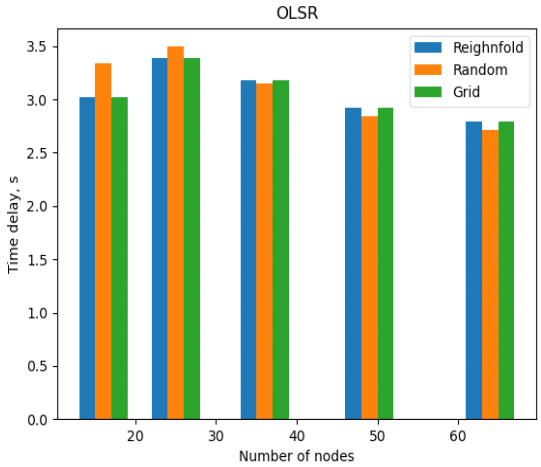

Рисунок 4.11 – Задержка времени для алгоритма маршрутизации OLSR

Рисунок 4.12 показывает задержку времени, с которой сталкиваются пакеты данных при использовании алгоритма маршрутизации DSDV в различных топологиях. Наблюдается, что с увеличением числа узлов задержка времени уменьшается. При оптимальном количестве 60 узлов случайная и топология Фрухтермана-Рейнгольда демонстрируют наименьшую задержку времени по сравнению с остальными топологиями.

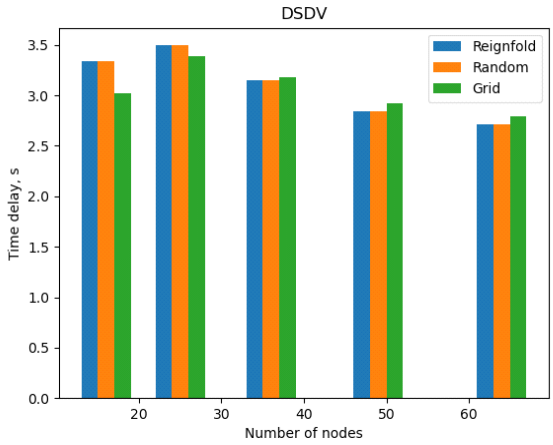

Рисунок 4.12 – Задержка времени для алгоритма маршрутизации DSDV

Джиттер – это изменение задержки между полученными пакетами в сети. Он может возникать из-за перегрузки сети, изменений маршрутизации или вариаций во времени прибытия пакетов. Джиттер вычисляется по следующей формуле:

$$Jitter = |(D_{i+1} - D_i) - (S_{i+1} - S_i)| \quad (4.4)$$

где S_i – это время, когда пакет i был отправлен источником, а D_i – время, когда он был получен приемником.

Рисунок 4.13 демонстрирует джиттер, с которым сталкиваются пакеты данных при использовании алгоритма маршрутизации DSDV в различных топологиях. На графике видно, что с увеличением числа узлов в сети джиттер

снижается, что свидетельствует о более стабильной передаче данных. Это также указывает на более высокую эффективность ячеистой топологии при использовании алгоритма маршрутизации DSDV по сравнению с другими топологиями, особенно когда оптимальное количество узлов составляет 60.

Рисунок 4.14 иллюстрирует джиттер для пакетов данных при использовании алгоритма маршрутизации OLSR в различных топологиях. На этом рисунке видно, что алгоритм OLSR демонстрирует лучшие результаты в ячеистой топологиях и Фрухтермана-Рейнгольда, с меньшими значениями джиттера по сравнению с случайной топологией.

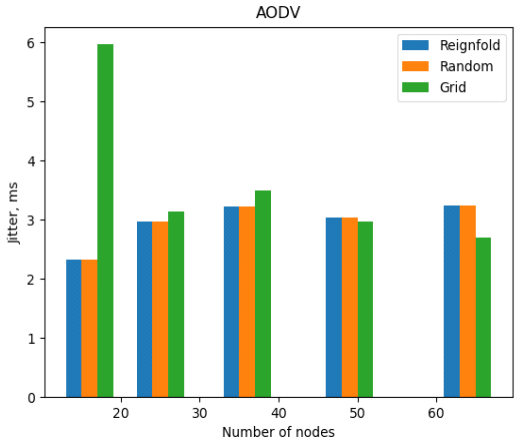

Рисунок 4.13 – Джиттер для алгоритма маршрутизации AODV

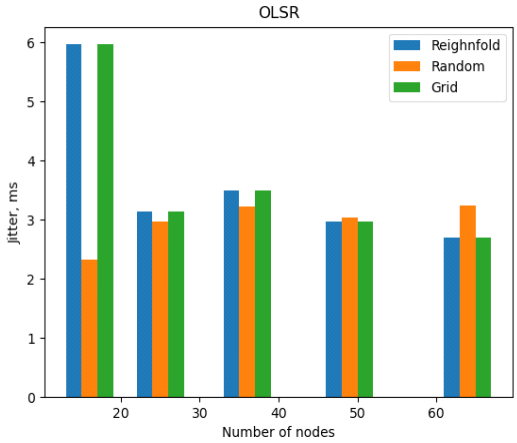

Рисунок 4.14 – Джиттер для алгоритма маршрутизации OLSR

Рисунок 4.15 иллюстрирует джиттер, с которым сталкиваются пакеты данных при использовании алгоритма маршрутизации DSDV в различных топологиях. График, представленный на рисунке 4.15, показывает, что для алгоритма маршрутизации DSDV ячеистая топология более эффективна по сравнению с двумя другими топологиями.

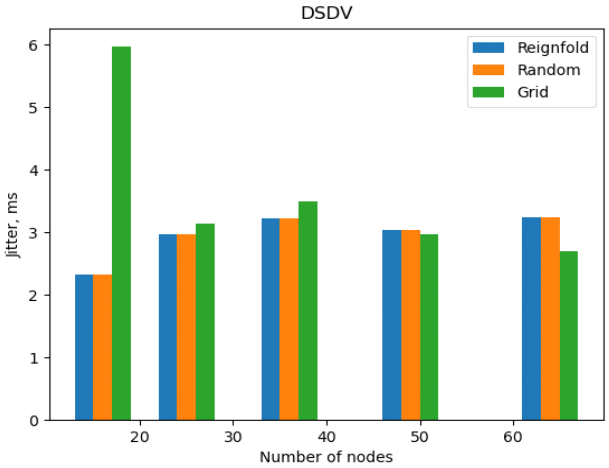

Рисунок 4.15 – Джиттер для алгоритма маршрутизации DSDV

На основе анализа пропускной способности, джиттера, задержки времени и потери пакетов в различных топологиях можно сделать следующие выводы о

целесообразности использования различных алгоритмов маршрутизации. Для алгоритма маршрутизации AODV ячеистая топология оказывается наиболее подходящей, поскольку она обеспечивает снижение потерь пакетов и стабильную передачу данных, особенно при количестве узлов около 60. Для алгоритма маршрутизации OLSR случайная топология показывает лучшие результаты по пропускной способности и задержке времени при 50-60 узлах. Для алгоритма маршрутизации DSDV как случайная, так и топология Фрухтермана-Рейнгольда, демонстрируют хорошие результаты по различным метрикам, особенно при количестве узлов около 50-60.

4.3 Алгоритм размещения узлов маршрутизаторов (GPA) в WMN

В данном разделе рассматривается предлагаемый алгоритм размещения узлов маршрутизаторов (GPA) [121], который адаптирует метод «box covering», предложенный Жангом [122] и применяемый в фрактальных сетевых алгоритмах. Этот метод позволяет эффективно распределять узлы маршрутизаторов, минимизируя затраты на установление и поддержание связи между ними. Кроме того, для моделирования влияние узлов друг на друга в сети алгоритм GPA использует аналогию с электростатическим взаимодействием [119]. В данной методологии каждый узел сети рассматривается как источник «заряда», который пропорционален его степени (числу соединений с другими узлами). На основе этих зарядов вычисляются силы взаимодействия между узлами с использованием аналогии с физическими законами (закон Кулона). Математически сила взаимодействия между двумя узлами i и j с зарядами q_1 и q_2 на расстоянии r определяется по формуле:

$$F = k\frac{|q_1 q_2|}{r^2}, \quad (4.5),$$

где $F-$ сила взаимодействия между узлами, $r -$ расстояние между узлами.

Сила взаимодействия между узлами зависит от произведения их «зарядов» и обратно пропорциональна квадрату расстояния между ними. Это моделирует влияние узлов на соседей в сети, где более сильное влияние оказывается на узлы, которые находятся ближе и имеют более высокую степень (заряд). В методологии исследования используется алгоритм «отталкивания», который основан на вычислении взаимодействий между узлами. Степень каждого узла рассматривается как его «заряд» в сети, что позволяет определить степень его влияния на соседние узлы. Алгоритм начинается с расчета евклидовых расстояний между всеми узлами, что позволяет оценить их прямолинейное расстояние в двумерном пространстве и определить степень взаимодействия между ними. Затем производится систематическая оценка взаимодействий между узлами. Для каждого узла вычисляется совокупная сила, оказываемая на него соседними узлами. Эта информация необходима для определения важности узла в сети и его общего влияния на структуру сети. Используя по аналогии закон Кулона, рассчитываются силы взаимодействия между узлами в пределах заданного радиуса покрытия. Суммирование полученных сил позволяет учитывать совокупное воздействие всех соседних узлов на каждый отдельный узел. После выполнения расчетов силы взаимодействия между узлами

сохраняются для дальнейшего анализа. Это позволяет сортировать узлы по их «влиянию» и выделять наиболее важные узлы сети, что способствует улучшению маршрутизации и повышению общей эффективности сети. Выявленные влиятельные узлы могут быть использованы для оптимального размещения маршрутизаторов или других сетевых устройств.

Для проверки эффективности предложенного алгоритма GPA, был проведен анализ пропускной способности сети с использованием программы NS3. В этом процессе каждый узел сети был моделирован как шлюз, который соединялся с сервером, отвечающим за передачу пакетов данных. Для оценки производительности сети сервер последовательно соединялся с каждым узлом-шлюзом, и для каждой связи вычислялась пропускная способность.

4.4 Оценка алгоритма GPA для оптимального размещения маршрутизаторов в WMN

В этом разделе представлены результаты, демонстрирующие влияние местоположения шлюза на мощность и пропускную способность для двух типов топологий: сетчатой и случайной [121]. Для построения сетей использовалась программа NS-3, параметры которой представлены в таблице 4.1. Для нахождения пропускной способности между узлами в этих сетях использовался существующий алгоритм маршрутизации OLSR [121].

На рисунке 4.16 (а) показан пример развернутой конфигурации сети 5x5, состоящей из 25 узлов. В этой сети выделены четыре ячеистых маршрутизатора, соответствующее покрытие которых обозначено желтым кружком, а посередине в качестве желтого узла специально выбран узел шлюза. Рисунок 4.16 (б) иллюстрирует расчетное соотношение силы к пропускной способности в сети.

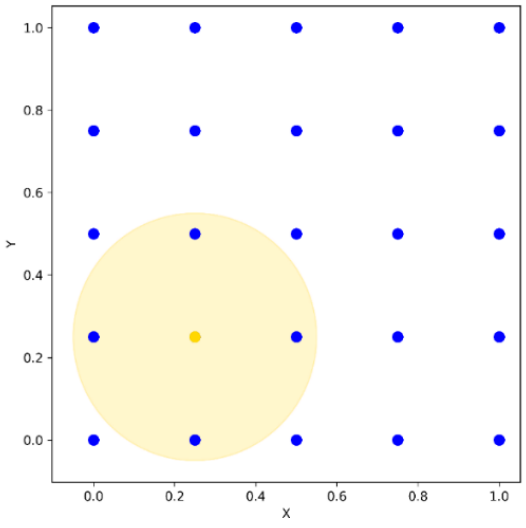

Рисунок 4.16 (а) – Иллюстрация ячеистой сети со шлюзом, расположенным в центре

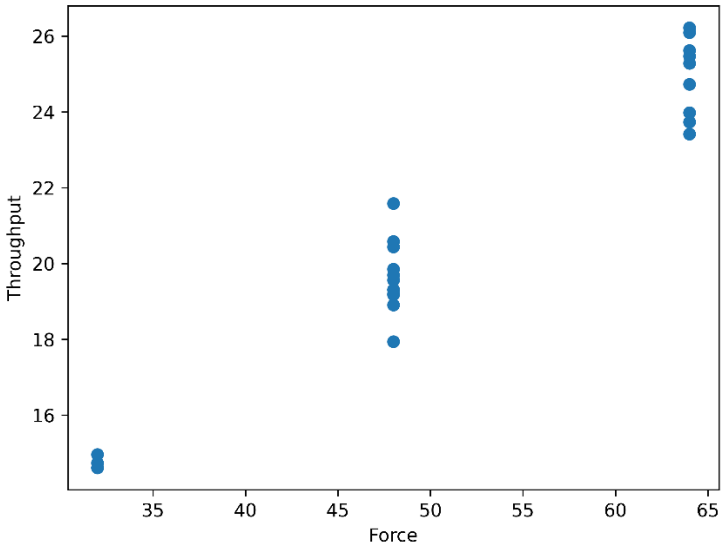

Рисунок 4.16 (б) – Отношение мощности к пропускной способности

Рисунок 4.16 (б) иллюстрирует прямую положительную корреляцию между силой и пропускной способностью, предполагая, что более высокие значения силы соответствуют увеличению пропускной способности. Выбор узла с наибольшей силой может служить эффективной стратегией размещения шлюза. На рисунке 4.17 (а) изображена случайная сеть, состоящая из 25 узлов, где 9 ячеистых маршрутизаторов вместе с их соответствующим покрытием выделены желтым кружком, а центрально расположенный узел специально выбран в качестве узла шлюза. Рисунок 4.17 (б) иллюстрирует расчетное соотношение силы к пропускной способности в случайной сети.

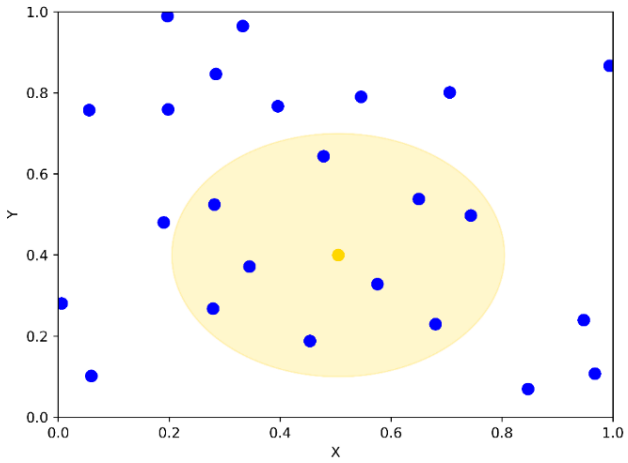

Рисунок 4.17 (а) – Иллюстрация случайной сети со шлюзом, расположенным в центре

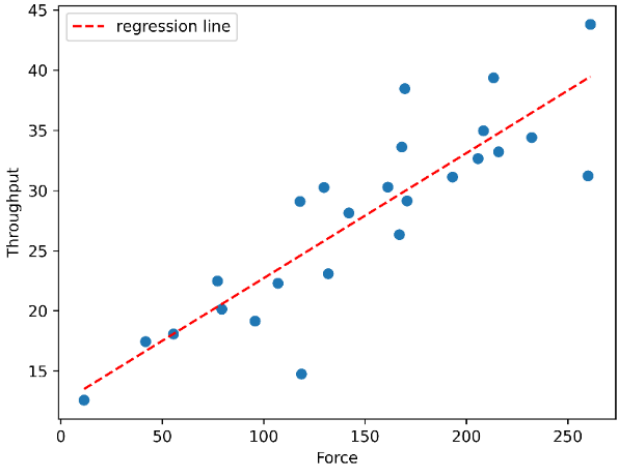

Рисунок 4.17 (б) – Отношение мощности к пропускной способности

Рисунок 4.17 (б) демонстрирует четкую прямую корреляцию между силой и пропускной способностью, указывая на то, что более высокие значения силы связаны с увеличением пропускной способности. Эти результаты подчеркивают критическую роль размещения шлюза в определении пропускной способности сети и подчеркивают важность принятия стратегических решений при проектировании и оптимизации сети. В данном исследовании было изучено влияние размещения шлюзов на две распространенные топологии сети: случайную и сетчатую. В случайной топологии сети, которая отличается децентрализованным и нерегулярным расположением узлов, было обнаружено, что стратегическое размещение шлюзов играет ключевую роль. Несмотря на непредсказуемость сети, результаты показали явную корреляцию между мощностью и пропускной способностью, что указывает на то, что оптимальное расположение шлюза может значительно повысить производительность сети.

В ячеистой топологии, где узлы организованы структурированным и единообразным образом, влияние размещения шлюзов оказалось более значительным. Стратегическое размещение шлюзов в сети продемонстрировало прямую положительную корреляцию между мощностью и пропускной способностью. Этот вывод подчеркивает важность учета топологии сети в

стратегиях размещения шлюзов, поскольку структурированные схемы могут обеспечить более предсказуемые результаты.

4.5 Алгоритм маршрутизации на основе информационной энтропии (IERA)

В этом разделе представлен алгоритм маршрутизации на основе информационной энтропии (IERA), принцип работы которого объясняется с использованием бинарного асимметричного канала [123, 124, 125]. Бинарный канал используется как модель передачи данных, где передаваемые сигналы принимают два возможных значения, при этом переданный сигнал обозначается как X, принятый сигнал − как Y, а их взаимосвязь описывается матрицей переходных условных вероятностей P(Y/X) (Рисунок 4.18). Оба сигнала представляют собой случайные величины, которые принимают значения согласно распределению вероятностей. В двоичном симметричном канале вероятность ошибок для сигналов x_1 и x_2 одинакова. На рисунке 4.18 представлен бинарный дискретный канал с вероятностями p (x_1) и p (x_2) = 1 - p(x_1), которые связаны с входными символами в бинарном алфавите.

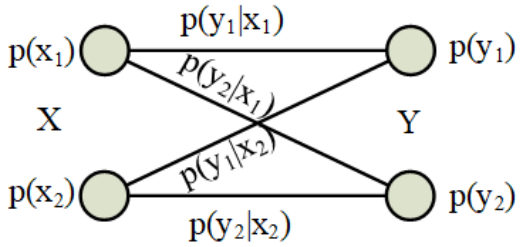

Рисунок 4.18 − Бинарный канал.

Матрица переходных условных вероятностей P(Y|X) для бинарного канала (Рисунок 4.18), определяется следующим образом [111]:

$$P(Y|X) = \begin{bmatrix} p(y_1|x_1) & p(y_1|x_2) \\ p(y_2|x_1) & p(y_2|x_2) \end{bmatrix} = \begin{bmatrix} 1-p_e & p_e \\ p_e & 1-p_e \end{bmatrix} \quad (4.6)$$

Двоичный канал служит моделью канала связи, используемой для передачи данных в битах. В симметричных каналах оба бита демонстрируют одинаковую вероятность ошибки, обозначаемую как p_e, что влияет на точность передачи битов. Асимметричные каналы различают вероятность ошибки для разных битов, что ведет к различным вероятностям ошибок. Для повышения энергетической эффективности предпочтительнее использовать бинарный асимметричный канал, где надежная передача битов с меньшей вероятностью ошибки снижает энергозатраты и повышает устойчивость к ошибкам [125].

Двоичный асимметричный канал, показанный на рисунке 4.19 и называемый Z-каналом, представляет собой канал с двоичным входом и выходом. В этом канале отдельные биты имеют разные вероятности ошибок: бит «1» несет вероятность ошибки, равную $p(1 \mid 0)$, в то время как бит «0» несет вероятность ошибки, равную $p(0 \mid 1)$, согласно формуле (4.7):

$$p(1|0) = \frac{1}{2} erfc\left(\frac{|m_1 - Th|}{\sigma_1}\right), p(0|1) = \frac{1}{2} erfc\left(\frac{|Th - m_0|}{\sigma_0}\right), \quad (4.7)$$

где Th − пороговое значение, используемое для принятия решения, в то время как m_1 и m_0 обозначают средние значения амплитуды сигнала для «1» и «0» соответственно, а σ_1 и σ_0 обозначают их соответствующие стандартные отклонения. Примечательно, что во многих беспроводных каналах связи средние значения амплитуды сигнала для «0» остаются неизменными, m_0 = 0, несмотря на изменения расстояния между передатчиком и приемником. И наоборот, уровень сигнала для «1» уменьшается пропорционально увеличению расстояния между передатчиком и приемником. Средняя энергия принятого сигнала «1» может быть определена как E_1=$m_1^2 T$, где T − длительность сигнала. Стандартное отклонение рассчитывается по формуле $\sigma_1 = \sqrt{N_1/T}$. Порог принятия решения можно установить равным стандартному отклонению сигнала «0», где $Th = \sigma_0 = \sqrt{N_0/T}$. Здесь N_1 и N_0 представляют собой мощность сигналов «1» и «0» соответственно. В результате формула (4.7) примет следующий вид:

$$p(1|0) = \frac{1}{2} erfc\left(\frac{\left|\sqrt{E_1} - \sqrt{N_0}\right|}{\sqrt{N_1}}\right), p(0|1) = \frac{1}{2} erfc(1) \approx 0.0786. \quad (4.8)$$

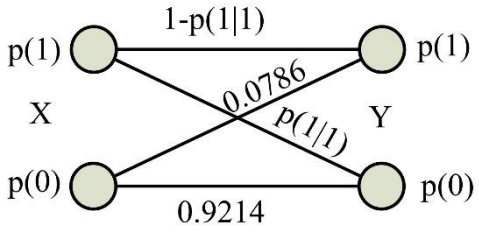

Рисунок 4.19 – Бинарный асимметричный канал.

В коде RZ (возврата к нулю) матрица переходных условных вероятностей для сигналов «1» и «0» действительно демонстрирует асимметрию в сценарии бинарного канала. Эта асимметрия часто характерна для реальных каналов связи, где вероятность ошибки различается при передаче «1» и «0»:

$$P(Y|X) = \begin{bmatrix} 1 - p(1|0) & 0.0786 \\ p(1|0) & 0.9214 \end{bmatrix} \quad (4.9)$$

Маршруты в беспроводных сетях часто включают в себя взаимосвязанные наборы каскадных каналов. В таком контексте каскадная модель представляет собой последовательность бинарных каналов, которые могут быть использованы для описания передачи данных через несколько уровней или звеньев в сети. Эта модель хорошо подходит для WMN сетей, где узлы сети связаны друг с другом через промежуточные маршруты, и каждый канал между узлами может быть моделирован как бинарный канал с вероятностью ошибки. Визуальная схема каскадных каналов представлена на рисунке 4.20.

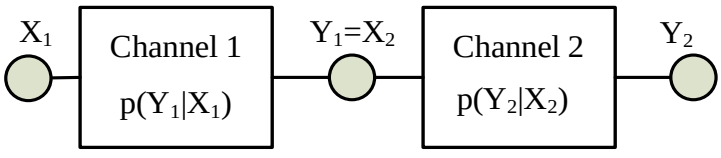

Рисунок 4.20 – Каскадные каналы.

Для удобства выход начального канала обозначается как Y_1, впоследствии служащий входом X_2 для второго канала, и так далее. Это предположение предполагает независимость двух каналов друг от друга. В двухканальной системе условная вероятность устанавливается путем умножения матрицы, полученной из переходных условных вероятностей.

$$P(Y_2|X_1) = P(Y_1|X_1)P(Y_2|X_2) \quad (4.10)$$

Для системы, состоящей из L каскадных каналов, формула может быть выражена следующим образом:

$$P(Y_L|X_1) = \prod_{i=1}^{L} P(Y_i|X_i) \quad (4.11)$$

Соотношение между совместной вероятностью P (Y_L, X_1) и условной вероятностью P($Y_L|X_1$) определяется формулой Байеса:

$$P(Y_L, X_1) = P(X_1)P(Y_L|X_1), \quad (4.12)$$

где X_1 входной сигнал в первый канал, Y_L выходной сигнал каскада из L каналов. Применяя уравнения (4.13) и (4.14), можно вычислить как взаимную, так и условную энтропию.

$$H(Y_L, X_1) = -\sum_{i=1}^{N}\sum_{j=1}^{N} p(y_i, x_j) log_2 p(y_i, x_j), \quad (4.13)$$

$$H(Y_L|X_1) = -\sum_{i=1}^{N}\sum_{j=1}^{N} p(y_i, x_j) log_2 p(y_i|x_j), \quad (4.14)$$

где p(y_i, x_j) и p($y_i \mid x_j$) обозначают элементы в матрицах P(Y_L,X_1) и P($Y_L|X_1$) соответственно, в то время как N представляет количество строк и столбцов в матрице. Формула, используемая для определения взаимной информации L каскадно соединенных каналов, такова:

$$I(X_1; Y_L) = I(Y_L; X_1) = H(X_1) - H(X_1|Y_L) = H(Y_L) - H(Y_L|X_1) \quad (4.15)$$

где $H(X_1)$ представляет энтропию Шеннона.

Вместо взаимной информации (4.15), которая зависит от разности энтропии $H(X_1)$источника и условной энтропии $H(X_1|Y_L)$, будем использовать условную информацию предложенную Жанабаевым З.Ж. [123, 125]. Далее формулы рассмотрим в дискретной форме. Физические процессы характеризуются взаимосвязью (корреляцией) минимум двух характерных величин $X\{x_i\}, Y\{y_i\}$. В этих случаях информация должна определяться через взаимодействие значений элементов множеств X, Y. В практике теории связи и телекоммуникаций широко используется взаимная информация К. Шеннона $I(X_1; Y_L)$ (формула 4.15). Выражение (4.16), используя формулу Байеса (4.12) можно представить в симметричном виде

$$I(X; Y) = I(Y; X) = \sum_i \sum_j p([x_i], [y_j]) \, log_2\left(\frac{p([x_i], [y_j])}{p([x_i]\, p[y_j])}\right) \geq 0 \quad (4.16)$$

Если нет корреляции между X и Y ($p([x_i], [y_j]) = p[x_i]\, p[y_j]$) выполняется равенство. Взаимная информация является симметричной мерой корреляций между $X \rightleftarrows Y$ и может быть использована только для известных сигналов X (переданный), Y (принятый). Однако связи $X = X(Y)$, $Y = Y(X)$ могут быть необратимыми, нелинейными.

В маршрутизации будем использовать асимметричную модель условной информации $I(Y|X)$, определяемой разностью энтропий ансамбля $H(X, Y)$ и наличия условия $H(Y|X)$:

$$I(Y|X) = H(X, Y) - H(Y|X) \quad (4.17)$$

Из формулы (4.17) следует асимметрия $I(Y|X) \neq I(X|Y)$, так как по аддитивности энтропии $H(Y,X) = H(X,Y)$, но $H(Y|X) \neq H(X|Y)$, связь $Y = Y(X)$, используемая нами для телекоммуникационных сигналов X является нелинейной, необратимой. Отметим, что, используя формулу Байеса (4.12) для вероятностей и значений энтропии

$$p([x_i],[y_j]) = p([x_i]|p[y_j])p([x_i]), \quad H(Y,X) = H(X) + H(Y|X), \quad (4.18)$$

вводимую нами меру $I(Y|X)$ можно также назвать условной энтропией сигнала $H(X)$[33]. Но это не будет соответствовать в общем случае физическому смыслу $I(Y|X)$ – уменьшение энтропии всего ансамбля $H(X,Y)$ из-за вычета энтропии части ансамбля $H(Y|X)$ определяет появление порядка-информации. Более важно то, что $H(X)$ определяется через $p([x_i])$, которая будет известной только через $p([x_i],[y_j])$, входящей в определение $H(X,Y)$. В маршрутизации для расчета пропускной способности пути необходимо учитывать связь между условной информацией и энтропией, которая количественно выражает меру порядка и хаоса. Для этой цели разделив формулу (4.19) на $H(X,Y)$ получим своеобразный закон сохранения нормированных значений условных информации и энтропии, описывающий переход порядок ($\tilde{I}$) – хаос ($\tilde{H}$) [123].

$$\tilde{I}(Y|X) + \tilde{H}(Y|X) = 1, \qquad \tilde{H} = \frac{H(Y|X)}{H(X,Y)} \qquad \tilde{I} = \frac{I(Y|X)}{H(X,Y)} \qquad (4.19)$$

В формуле (4.19) примем $Y = Y|(X)$ т. е. условие Y находится через сигнал в виде его первой производной т.е. будем пользоваться фазовым портретом.

Функция условия $Y = Y(X)$ может быть выбрана в различной форме, отражающей специфику процесса. Для процессов с ускорением $Y(X)$ находится через вторую производную по времени $X(t)$, можно использовать вклад стандартных шумовых процессов и т.д. Далее введенную нами формулу условную информацию применим к маршрутизации

$$I(Y_L|X_1) = H(X_1,Y_L) - H(Y_L|X_1) \qquad (4.20)$$

Эта условная информация относится к коллективной информации по L каскадным каналам, а не к источнику X_1. Следовательно, выбор $I(Y_L|X_1)$ принятого сигнала был преднамеренным выбором, сделанным в качестве меры для маршрутизации. Это различие характеризует уровень достоверности в системе передачи информации. Положительное значение, полученное в этом контексте, может быть интерпретировано как энтропия принятого сигнала, означающая определенный объем информации через неопределенность в принятом сигнале. Конечно, энтропия принятого сигнала имеет тенденцию быть ниже, чем энтропия переданного сигнала, в первую очередь из-за потерь, понесенных по каскадным каналам. Оценка изменений энтропии и потерь, возникающих в этих каскадных каналах, позволяет оценить их пропускную способность. Эта оценка становится решающей при определении эффективности и надежности передачи данных по таким каскадным конфигурациям каналов [124, 125].

Максимальное значение условной информации (4.20) по каналам соответствует расчетной пропускной способности маршрута. Следовательно,

доступная расчетная пропускная способность внутри канала может быть аппроксимирована с помощью формулы:

$$C = \frac{max(I(Y_L|X_1))}{T} = \frac{\max\,(H(X_1,Y_L) - H(Y_L|X_1))}{T} \quad (4.21)$$

Взаимосвязь между условной информацией и пропускной способностью в алгоритме маршрутизации определяется формулой (4.21). В более плотных сетях, где традиционные алгоритмы могут бороться с перегрузками, подход, основанный на информационной энтропии, может привести к более сбалансированному и эффективному распределению трафика, тем самым увеличивая среднюю пропускную способность сети. На рисунках 4.21 и 4.22 показаны сравнительные графики, полученные с помощью формул (4.15, 4.20). На рисунке 4.21 каждая кривая обозначает количество каскадов L в диапазоне от 1 до 10 в канале.

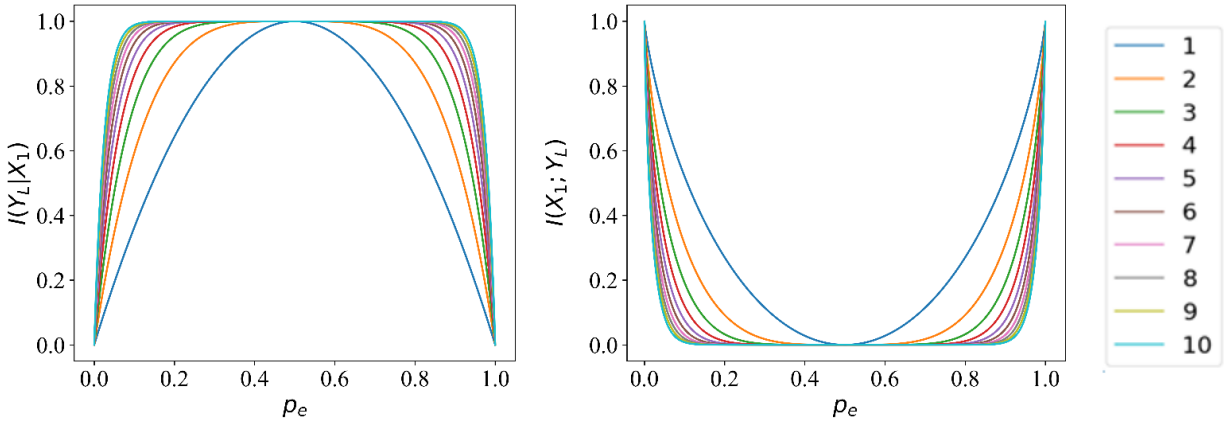

Рисунок 4.21 – Условная информация $I(Y_L|X_1)$(4.20) и взаимная информация $I(X_1;Y_L)$ (4.15) в зависимости от вероятности ошибки p_e для симметричных каналов.

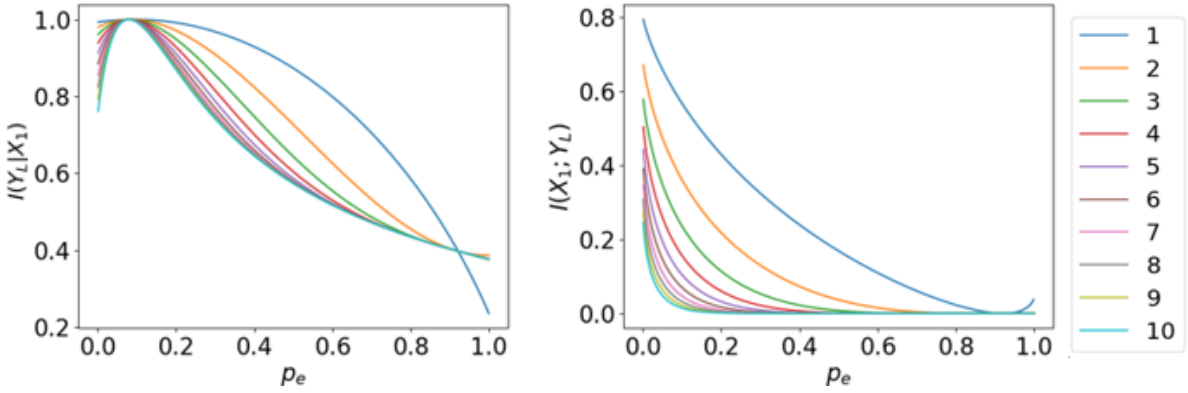

Рисунок 4.22 – Условная информация $I(Y_L|X_1)$ (4.20) и взаимная информация $I(X_1;Y_L)$ (4.15) в зависимости от вероятности ошибки p_e для асимметричных каналов.

Рисунки 4.21 и 4.22 демонстрируют, что с увеличением числа каскадов L взаимная информация $I(X_1;Y_L)$ стремится к нулю быстрее. Однако условная

информация $I(X_1|Y_L)$ неизменно сохраняет положительные значения или же в ассимитричных каналах стеримтся к нулю медленнее. Взаимная информация $I(X_1;Y_L)$ достигает нуля, указывая на отключение канала, продолжая сохранять минимальное ненулевое значение, что особенно заметно в асимметричных каналах. Следовательно, $I(X_1|Y_L)$ является важной мерой для принятия решений о маршруте, учитывая зависимость маршрута от условной информации.

Оценка доступной полосы пропускания внутри канала включает в себя оценку двух узлов, задействованных в передаче и приеме пакетов. В беспроводных сетях вероятность ошибки канала часто возрастает по мере увеличения расстояния между двумя узлами. Предполагая, что узлы n_1 и n_2 работают в пределах диапазона связи, текущие координаты для n_1 и n_2 обозначаются как (x_{n1}, y_{n1}) и (x_{n2}, y_{n2}) соответственно. Расстояние между узлом n_1 и узлом n_2 вычисляется с использованием формулы:

$$d(n_1, n_2) = \sqrt{(x_{n1} - y_{n1})^2 + (x_{n2} - y_{n2})^2}. \quad (4.22)$$

Мощность принимаемого сигнала определяется на основе модели потерь на свободном пути [124, 125].

$$E_1 = E_t - 20log\left(\frac{4nfd}{c}\right). \quad (4.23)$$

В формуле (4.23) E_1, E_t, f, c и d обозначают мощность принятого сигнала, мощность передаваемого выходного сигнала в децибелах, частоту в герцах, скорость света в метрах в секунду и расстояние от передатчика до приемника в метрах соответственно.

В следующем шаге представлено подробное руководство по предложенному методу. Для иллюстрации метода выполняются следующие этапы:

1.Определение ошибки канала: В WMNs случайной величиной является состояние канала или линии связи в любой момент времени. Это состояние может характеризоваться показателями качества канала, такими как частота битовых ошибок или вероятности ошибок канала, которые зависят от расстояния и мощности принимаемого сигнала между узлами.

2. Расчёт распределений вероятностей канала: для каждого канала необходимо построить условные вероятности перехода. Это также можно сделать, собрав данные и отправив пакеты «привет» для каждого канала связи с течением времени, чтобы вычислить условные вероятности, используя вероятности ошибок канала.

3. Расчёт условной информации маршрута: для определённого маршрута в сети вычисляется взаимная и условная энтропия пути.

4. Поиск наилучшего маршрута: вычисляется условная информация в качестве индикатора для принятия решений о маршруте. Маршруты с более высокой условной информацией могут быть более устойчивыми к изменениям, что обеспечивает хорошую производительность передачи сигнала, в то время как маршруты с меньшей условной информацией могут быть нестабильными. Рассмотрим алгоритм маршрутизации, основанный на теории информационной энтропии, на примере сети из шести узлов (см. Рисунок 4.23).

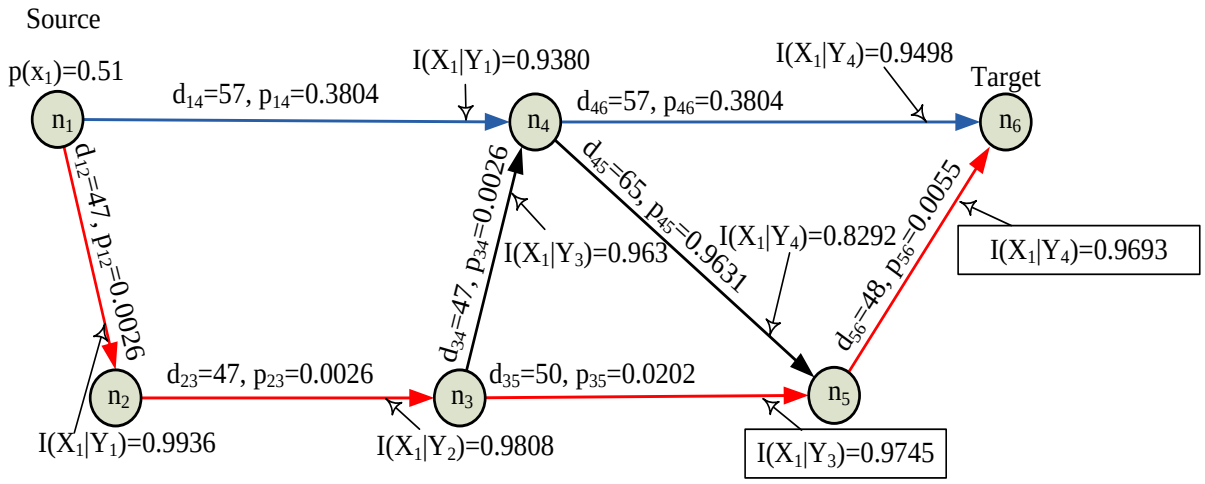

Рисунок 4.23 – Пример маршрутизации в сети с асимметричными каналами, осуществленный на основе теории информационной энтропии.

В приближенной сети с использованием формул (4.8), (4.22), (4.23) определяются вероятности ошибок канала между двумя соседними узлами, которые представлены в виде p_{ij}, где i и j - два соседних узла. Например, $p_{12}=p(y_1|x_2)$ – условная вероятность ошибки канала между двумя соседними узлами, такими как узлы n_1 и n_2 [124, 125].

Мощность принятого сигнала E_1 и вероятность ошибки были вычислены с использованием параметров, представленных в таблице 4.2. Отмечается, что все узлы имеют идентичные параметры, при этом дальность действия каждого маршрутизатора установлена на уровне 80 метров.

Таблица 4.2 – Параметры беспроводной связи между узлами

Значения	параметров
P_t мощность передатчика	53 дБ
f частота сигнала	2,4 ГГц
N_0 мощность шума для сигнала равна '0’	2 мВт
N_1 мощность шума для сигнала ’1‘	1 мВт

В сценариях, где расстояния между узлами неизвестны, эмпирическая оценка вероятности ошибки (p_e) становится решающей. Эта оценка может быть выполнена с помощью практических методов, таких как передача и прием проверочных пакетов приветствия. Учитывая вероятность входных символов для узла-источника (n_1), характеризуемую как $p(x_1)=0,51$ и $p(x_2)=0,49$, эмпирическая проверка с помощью таких пакетных передач помогает получить более точную оценку вероятности ошибки в беспроводных сетях. После определения вероятности ошибки решающее значение приобретает установление матриц условных и совместных вероятностей. Например, в

асимметричном канале между узлами n_1 и n_2 с расстоянием d_{12}=47 метров вычисленная вероятность ошибки p_{12}=p(1|0) составляет 0.0026. Впоследствии эти матрицы, полученные с использованием формул (4.6) и (4.8) соответственно, будут эквивалентны.

$$P(Y_1|X_1) = \begin{bmatrix} 0.9974 & 0.0786 \\ 0.0026 & 0.9214 \end{bmatrix}, P(Y_1, X_1) = \begin{bmatrix} 0.51 \\ 0.49 \end{bmatrix} \times \begin{bmatrix} 0.9974 & 0.0786 \\ 0.0026 & 0.9214 \end{bmatrix} = \begin{bmatrix} 0.5087 & 0.0385 \\ 0.0013 & 0.4515 \end{bmatrix}.$$

Очевидно, что вероятности $P(Y_1)$ равны [0.5472, 0.4528]. Вычисление условной информации по формуле (4.20) дает значение $I(Y_1|X_1)$ = 0.9936.

Для второго каскада при d_{23}=47 м между узлами n_1 и n_3 умножаем на матрицу условные вероятности каскадированных каналов

$$P(Y_2|X_1) = P(Y_1|X_1)P(Y_2|X_2) = \begin{bmatrix} 0.9974 & 0.0786 \\ 0.0026 & 0.9214 \end{bmatrix} \begin{bmatrix} 0.9974 & 0.0786 \\ 0.0026 & 0.9214 \end{bmatrix} = \begin{bmatrix} 0.8488 & 0.0567 \\ 0.1512 & 0.9433 \end{bmatrix}.$$

$$P(Y_2, X_1) = \begin{bmatrix} 0.51 \\ 0.49 \end{bmatrix} \begin{bmatrix} 0.8488 & 0.0567 \\ 0.1512 & 0.9433 \end{bmatrix} = \begin{bmatrix} 0.5075 & 0.0739 \\ 0.0025 & 0.4161 \end{bmatrix}. \quad P(Y_2) = [0.5814 \quad 0.4186].$$

$I(Y_2|X_1)$ = 0.9808. Также есть незначительные потери во втором канале.

Продолжая эти операции, вычисляются все значения условной информации $I(Y_L|X_1)$ в каждом узле (Рисунок 4.23). Условные информации $I(Y_L|X_1)$ двух маршрутов [n_1→ n_2 →n_3→ n_4] и [n_1 →n_4] сравниваются в узле n_4. После сравнения выбирается маршрут с максимальным значением условной информации, и его матрица переходных условных вероятностей сохраняется на посещаемом узле. Эти операции будут продолжаться до тех пор, пока не будет построен маршрут с максимальным значением условной информации (целевой) до пункта назначения.

На рисунке 4.23 красными стрелками показан маршрут [n_1→ n_2 →n_3→ n_5 →n_6], рассчитанный предложенным методом, при котором достигается максимальное значением условной информации. Синими стрелками показан маршрут [n_1 →n_4 →n_6], рассчитанный в соответствии с алгоритмом Dijkstra для расстояня d_{ij}. Согласно алгоритму Dijkstra, сумма расстояний d_{14}+d_{46} меньше, чем сумма расстояний предлагаемого метода d_{12}+ d_{23}+ d_{35}+ d_{56}. Однако значение условной информации для алгоритма Dijkstra оказывается ниже, что указывает на меньшую пропускную способность. Принцип построения маршрута предлогаемым алгоритмом приведен ниже.

Алгоритм

(1) Составление временного ряда входного сигнала X={[x_i]}
(2) Вычисление расстояние d_{ij} (формула (4.22)) между текущим узлом и соседними. Затем определяются вероятности ошибок каждого p_e канала в сети (Оценка вероятности ошибки p_e-канала также может быть оценена путем передачи и приема проверочных пакетов приветствия. Формируется матрица переходных условных вероятностей $P(Y|X)$.
(3) Вычисление совместной вероятности P(X, Y), определяя условие Y через $P(Y|X)$;

(4) Расчет условной информации $I(Y_L|X_1)$ по формуле 4.20 в исходном узле и называем его «текущим» узлом;
(5) Если информация узла, обусловленная энтропией, больше фактической, то осуществляется обновление ее значения. Каждый узел хранит $I(Y_L|X_1)$. В этом случае в последующих каскадах используется формула (4.6) для определения условной вероятности. Текущий узел заменяется и устанавливается значение true (правда). Далее не учитываются посещенные узлы.
(6) Выбирается следующий текущий узел с наибольшим значением условной информации $I(Y_L|X_1)$ и значением false (ложь) в посещенном поле.
(7) Повторяется операция, начиная с шага 3, пока не будут посещены все узлы.

Вычислительную сложность предлагаемого алгоритма маршрутизации в WMN можно проанализировать с точки зрения операций, выполняемых на каждом шаге алгоритма. Вычисление вероятности входного сигнала. Вероятности могут быть определены путем отправки пакетов “Привет” или получены эмпирически, этот шаг может быть относительно простым с постоянной временной сложностью. Для каждого соседнего узла алгоритм выполняет несколько вычислений, включая определение вероятностей ошибок, формирование матриц условных вероятностей, вычисление совместных вероятностей и вычисление условной информации. Вычислительная сложность этих операций зависит от таких факторов, как количество соседних узлов, размер сети и сложность используемых уравнений. Выбор следующего текущего узла с наибольшим значением условной информации и обеспечение того, чтобы он ранее не посещался, требует поиска и сравнения значений энтропии. Этот шаг обычно включает в себя перебор соседних узлов и выбор узла с максимальным значением условной информации. В целом, вычислительная сложность предлагаемого алгоритма зависит от таких факторов, как размер и топология сети, количество соседних узлов и сложность вычислений, выполняемых на каждом шаге [124, 125].

На рисунке 4.24 представлена таблица маршрутизации (для сети на рисунке 4.23), сформированная с использованием алгоритма IERA. В таблице для каждого узла указаны оптимальные маршруты к другим узлам, определённые на основе максимизации суммарной взаимной информации. Структура таблицы включает целевой узел, следующий узел на маршруте, значение метрики, представляющее собой сумму взаимной информации по маршруту, а также сам маршрут в виде последовательности узлов. Поскольку маршрутизация осуществляется в однонаправленном ориентированном графе, наличие пути от исходного узла к целевому не означает, что существует обратный маршрут.

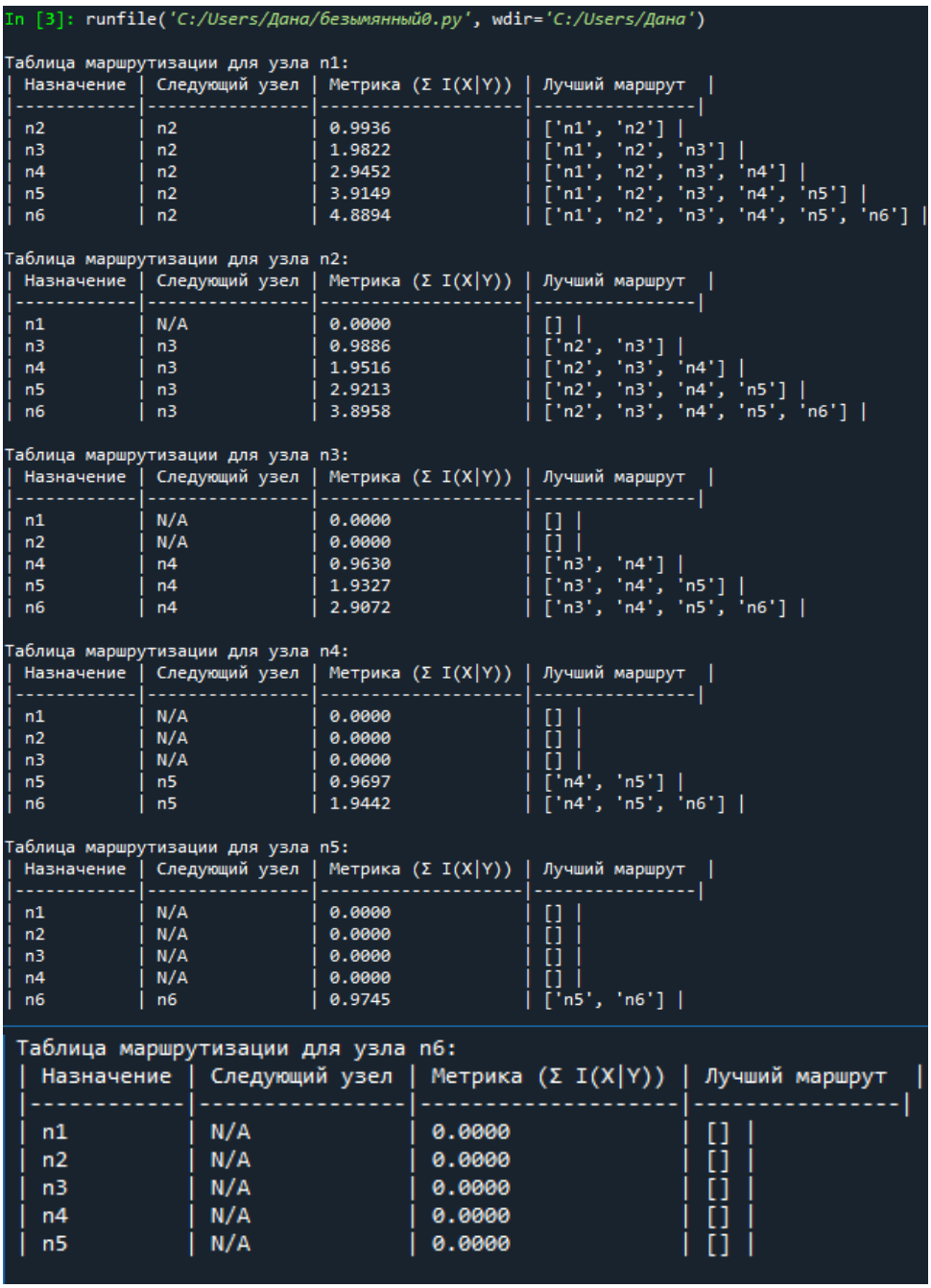

Рисунок 4.24 – Таблица маршрутизации, сформированная с использованием алгоритма IERA.

Представленная таблица маршрутизации демонстрирует работу алгоритма IERA, который формирует маршруты на основе максимизации суммарной взаимной информации. Такой подход обеспечивает выбор путей с наибольшей информационной пропускной способностью, что повышает эффективность передачи данных в сети.

4.6 Сравнение протоколов маршрутизации IERA, Dijkstra, ACO, OLSR, AODV в среде Python

Для оценки эффективности алгоритма была построена модель ячеистой сети, в которой крайние узлы расположены ближе друг к другу, чем центральные узлы (Рисунок 4.25). Используемые параметры оставались в соответствии с ранее заявленными (таблица 1). Расчеты проводились с использованием среды Python на компьютере, оснащенном следующими техническими характеристиками: процессор Intel Core i7 8700 с тактовой частотой 3,4 ГГц. Библиотека networkx использовалась для генерации динамических сетевых топологий с использованием метода компоновки spring, охватывающего диапазон от 10 до 100 узлов [124].

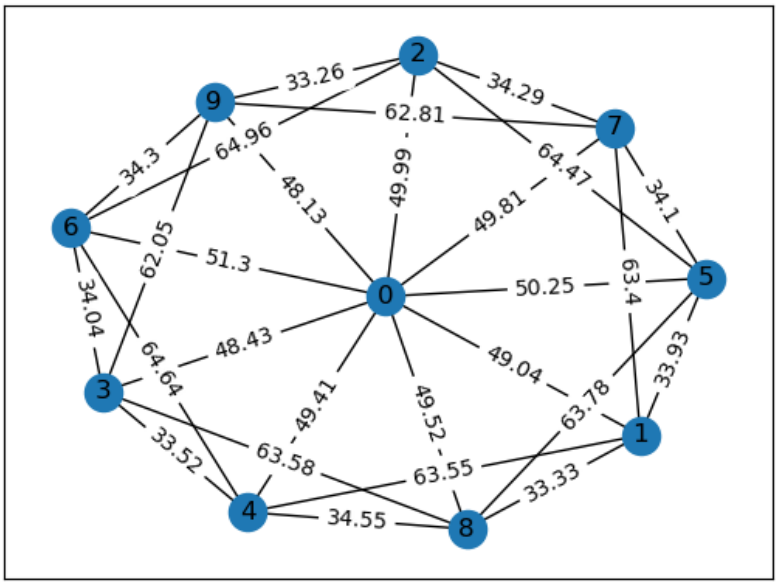

Рисунок 4.25 – Модель ячеистой сети, состоящей из 10 узлов. Цифры на рисунке обозначают расстояния d_{ij} между узлами.

Для сравнения было использовано пять алгоритмов маршрутизации: Dijkstra, ACO, OLSR, AODV и ранее разработанный наш метод. Путем определения путей между всеми узлами случайно сгенерированных сетей (Рисунок 4.25) были рассчитаны средние значения условной информации (Рисунок 4.26) и пропускной способности (Рисунок 4.27) в зависимости от количества узлов.

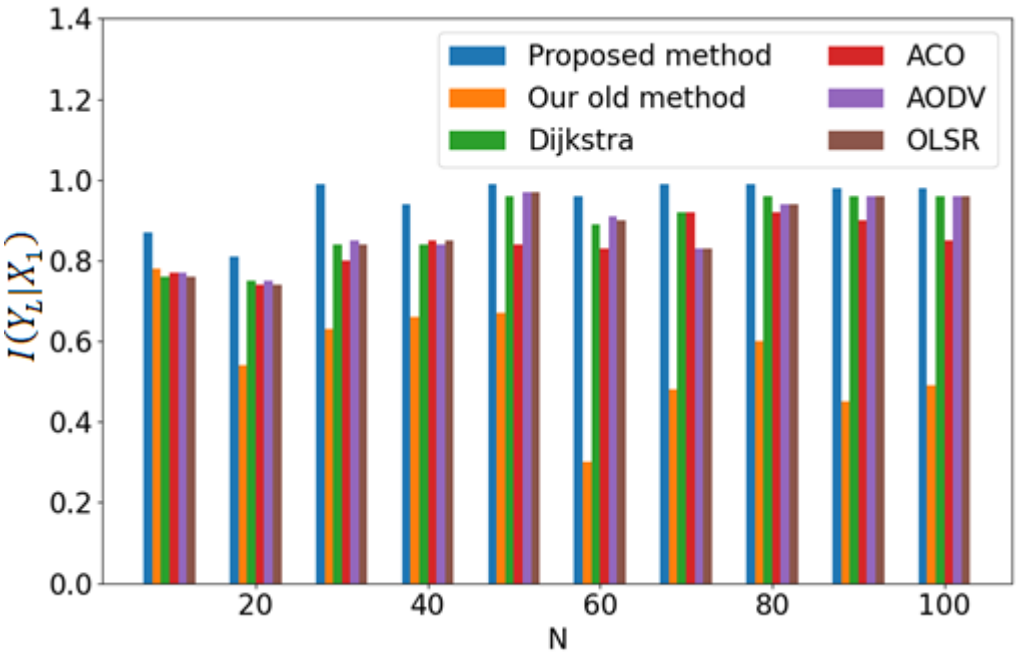

Рисунок 4.26 – Средние значения условной информации (формула (12)) для различных алгоритмов маршрутизации в зависимости от количества узлов N.

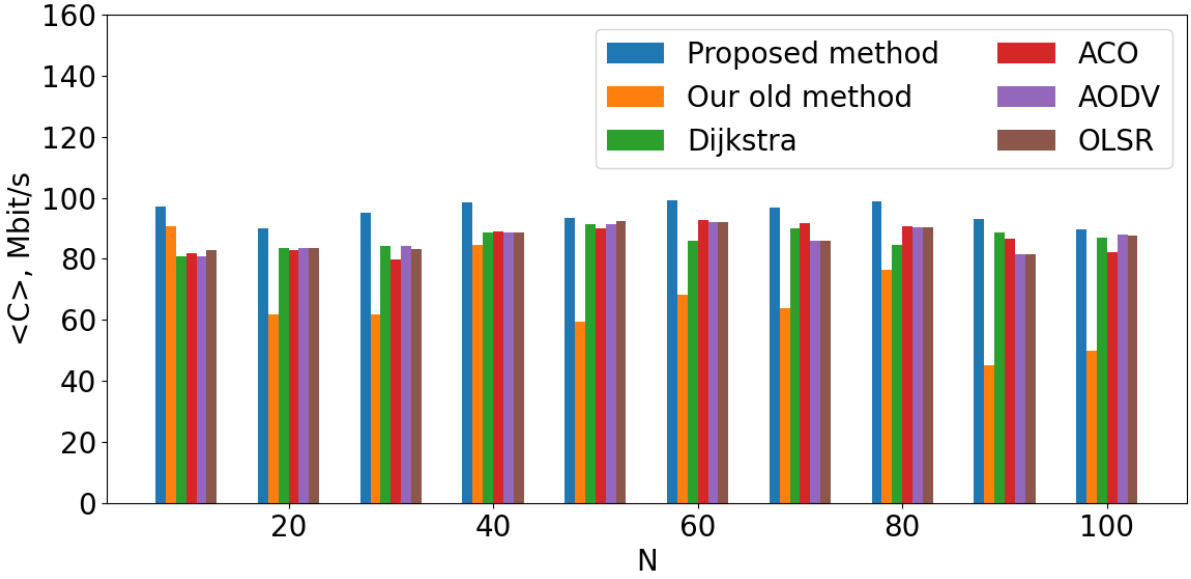

Рисунок 4.27 – Средние значения пропускной способности (формула (13)) для различных алгоритмов маршрутизации в зависимости от количества узлов N.

Согласно рисункам 4.26, 4.27 по мере увеличения количества узлов предложенный алгоритм демонстрирует более высокие средние значения как для условной информации (Рисунок 4.26), так и для пропускной способности (Рисунок 4.27) по сравнению с другими алгоритмами. Разработанный алгоритм маршрутизации обеспечивает значительное улучшение пропускной способности сети, достигая 90 Мбит/с при 100 узлах, что является оптимальным результатом для рассматриваемых сложных условий с высоким уровнем помех и интерференции, тогда как предыдущий наш метод демонстрирует 40 Мбит/с, а классические алгоритмы OLSR, AODV и Dijkstra 85 Мбит/с. Предлагаемый нами алгоритм учитывает вероятность ошибки канала, которая зависит от расстояния

между узлами. Более того, оптимальный маршрут внутри WMN сети определяет приоритет узлов, находящихся в непосредственной близости, что потенциально приводит к общей длине пути между исходным и целевым узлами. Алгоритмы OLSR и AODV демонстрируют высокую производительность при умеренных значениях условной информации и пропускной способности в сетях различного размера. ACO, несмотря на более высокую вычислительную сложность, обеспечивает стабильную работу при увеличении числа узлов, однако уступает по эффективности разработанному методу, особенно при низком SNR. Наш предыдущий метод показывает наименьшую пропускную способность среди рассмотренных алгоритмов, что указывает на ограничения случайного выбора маршрута и подчеркивает значимость более продуманного подхода к построению маршрутов.

Для подтверждения эффективности предложенного алгоритма была проведена оценка средней частоты битовых ошибок на всех маршрутах (Рисунке 4.28). Во всех рассмотренных топологиях наш метод демонстрирует устойчиво более низкие значения ошибки по сравнению с другими алгоритмами, что отражено на Рисунке 4.28.

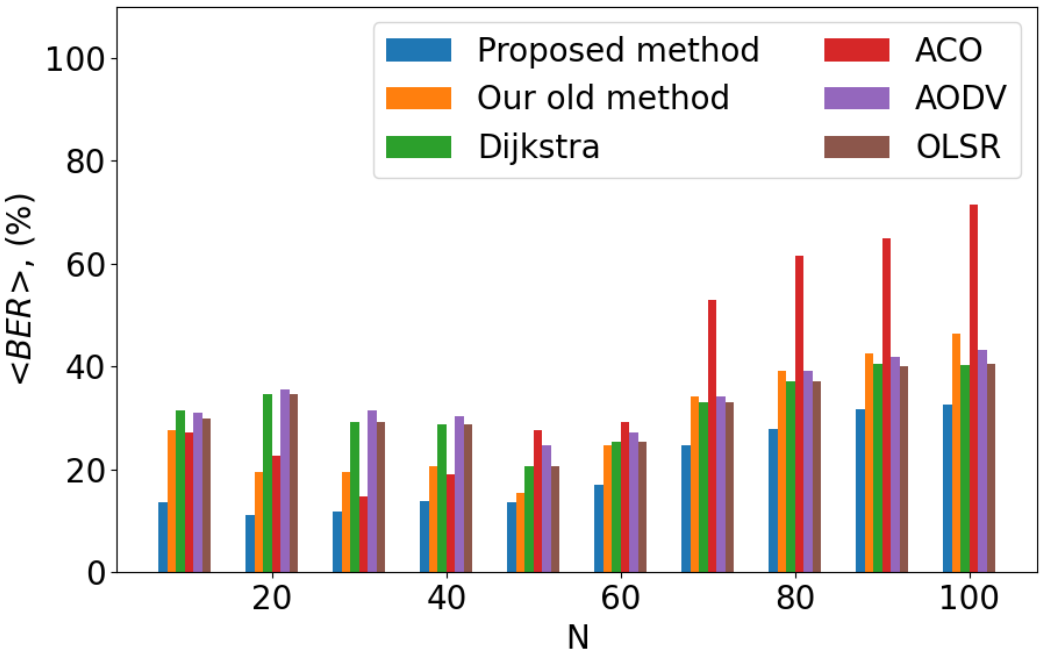

Рисунок 4.28 – Средние BER для различных алгоритмов маршрутизации в зависимости от количества узлов N.

Согласно данным, представленным на рисунке 4.28, алгоритм ACO показывает самый высокий показатель BER (75%), за ним следуют наш предыдущий метод (45%), алгоритмы AODV (41%), OLSR (39%), Dijkstra (39%), а предложенный нами метод IERA демонстрирует самый низкий показатель BER (33%), что свидетельствует о его потенциальном превосходстве в минимизации ошибок при маршрутизации [124, 125].

Вывод по пункту 4.1

В данной главе рассмотрены различные топологии WMN, включая случайную топологию, ячеистую и топологию Фрухтермана-Рейнгольда. Каждая из этих топологий имеет свои особенности, преимущества и недостатки, которые могут существенно повлиять на производительность сети. Случайная топология характеризуется отсутствием четкой структуры, что позволяет создавать гибкие сети, однако может привести к непредсказуемым связям между узлами. Ячеистая топология, напротив, обеспечивает надежное и предсказуемое покрытие благодаря структурированной организации узлов, но требует фиксированной инфраструктуры и может быть дорогостоящей в развертывании. Топология Фрухтермана-Рейнгольда, используемая для визуализации сетевых структур, способствует пониманию взаимосвязей между узлами, однако не определяет схему работы сети. Важность учета топологии сети WMN заключается в том, что она определяет эффективность маршрутизации, пропускную способность и надежность системы. Правильный выбор топологии может значительно улучшить характеристики сети, снизить задержки и повысить устойчивость к сбоям. Таким образом, при проектировании и оптимизации беспроводных сетей необходимо тщательно анализировать и выбирать подходящую топологию в зависимости от конкретных требований и условий эксплуатации.

Вывод по пункту 4.2

В данной главе проведен сравнительный анализ протоколов маршрутизации AODV, DSDV и OLSR в различных топологиях беспроводных сетей WMN с использованием симулятора NS-3. Результаты показали, что различные топологии оказывают значительное влияние на производительность протоколов. Для AODV наилучшие результаты по пропускной способности и минимальным потерям пакетов были получены в ячеистой топологии при количестве узлов 60. Алгоритм OLSR продемонстрировал лучшую пропускную способность и меньшую задержку времени в случайной топологии при 50-60 узлах. Алгоритм DSDV оказался наиболее эффективным в топологиях Фрухтермана-Рейнгольда и случайной, при числе узлов 50-60. В целом, выбор топологии и протокола маршрутизации зависит от специфики задачи и требований к сети, таких как пропускная способность, задержка и устойчивость к потерям пакетов.

Вывод по пункту 4.3, 4.4

Представлен алгоритм размещения узлов маршрутизаторов (GPA) в WMN на основе адаптации метода «box covering» для расчета силы взаимодействий между узлами. Показано, что сила взаимодействия между узлами, определяемая их «зарядом» и расстоянием между ними, служит основой для вычисления влияния узлов на общую производительность сети. Оценка алгоритма GPA в различных топологиях, таких как ячеистая и случайная сеть, демонстрирует явную корреляцию между силой и пропускной способностью. Стратегическое размещение шлюзов, основанное на силах взаимодействия, значительно улучшает эффективность передачи данных. Результаты моделирования в NS3

подтверждают, что в случайной топологии сила и пропускная способность сохраняют положительную зависимость, что указывает на важность выбора оптимального места размещения шлюза. В сеточной топологии влияние размещения становится более заметным, подчеркивая, что структурированная организация узлов способствует предсказуемости и повышению производительности сети. Таким образом, проведенное исследование демонстрирует, что размещение узлов маршрутизаторов является ключевым фактором, влияющим на эффективность работы WMN.

Вывод по пункту 4.5, 4.6

В данном исследовании представлен алгоритм маршрутизации, основанный на информационной энтропии, разработанный для WMN. Этот алгоритм учитывает динамическую природу WMN, оптимизируя использование пропускной способности и минимизируя битовую ошибку (BER). Проведенные моделирования и сравнительный анализ с алгоритмами Dijkstra, ACO, OLSR и AODV подтверждают его преимущества. Использование маршрутов с максимальными значениями условной информации в беспроводных каналах обеспечивает эффективное распределение трафика и снижение средних показателей битовой ошибки. Результаты демонстрируют вычислительную эффективность и конкурентоспособность алгоритма на сетях различных размеров, с существенно более низкими средними значениями BER по сравнению с существующими методами. Адаптивность алгоритма на основе условной информации к изменяющимся условиям сети позволяет достичь масштабируемости и надежности в различных приложениях WMN. Перспективы его дальнейшего использования включают экспериментальную реализацию и применение в реальных устройствах, учитывая совместимость алгоритма с языками программирования и инструментами, такими как Python, C++, NS-3 и NetSim.

ЗАКЛЮЧЕНИЕ

В ходе диссертационного исследования была проведена работа, направленная на оптимизацию сети и маршрутизации в WMN путем разработки инновационных алгоритмов. Разработанные алгоритмы учитывают характеристики среды передачи, фрактальные свойства сети и информационно-энтропийные критерии, что позволяет повысить быстродействие, надежность и скорость передачи данных.

В рамках исследования были разработаны три ключевых алгоритма:

1. Классификатор, основанный на извлечении признаков взаимной информации для распознавания типов схем модуляции в системах MIMO, показал свою высокую эффективность при низком уровне отношения сигнал/шум (SNR) более чем 5 дБ. Этот метод не требует устранения помех MIMO и большого объема обучающих данных, так как он устойчив к неопределенности шума и пространственной корреляции. Моделирование и экспериментальные результаты подтверждают, что алгоритм успешно функционирует даже при низком SNR, что представляет собой значительный шаг вперед в распознавании схем модуляции в системах MIMO.

2. Алгоритм кластеризации CIEA продемонстрировал свою эффективность и конкурентоспособность по сравнению с существующими методами. Учитывая эксцентриситет узлов, алгоритм CIEA обеспечивает оптимальное покрытие сети и снижает вычислительные затраты при маршрутизации. Сравнительный анализ фрактальных размерностей, рассчитанных с использованием алгоритма CIEA и других методов, подтвердил его применимость для реальных социальных сетей, демонстрируя высокую степень адаптивности. Кластерный маршрутизатор на основе CIEA минимизирует сложность поиска кратчайшего пути, обеспечивая быстродействие алгоритма маршрутизации на 20 микросекунд при 3280 узлах в сравнении с классическим алгоритмом Dijkstra.

3. Алгоритм маршрутизации, основанный на информационной энтропии, был разработан для беспроводных каскадных каналов в сетях WMN. Этот алгоритм учитывает динамическую природу сетей и оптимизирует использование пропускной способности, минимизируя битовую ошибку (BER). Проведенные моделирования и сравнительный анализ с алгоритмами Dijkstra, ACO, OLSR и AODV подтверждают его преимущества, включая значительно более низкие средние значения BER. Использование маршрутов с максимальными значениями условной информации обеспечивает эффективное распределение трафика, что делает данный алгоритм перспективным для реализации в реальных условиях. В частности, результаты численного моделирования для 100 узлов показывают, что в сравнении с классическими алгоритмами Dijkstra, OLSR, AODV данный алгоритм обеспечивает повышение пропускной способности WMN на 5 Мбит/с.

В дополнение к основным достижениям диссертации представлено исследование фрактальных и информационных размерностей сетей, рассчитанных на основе алгоритма кластеризации CIEA. Этот анализ углубляет

понимание структурных и динамических характеристик сетей, что критически важно для оптимизации маршрутизации и управления трафиком.

Дополнительно исследуются различные топологии WMN, включая ‘mesh’ и случайные структуры, что позволяет выявить их преимущества и недостатки в маршрутизации и передаче данных. Анализ этих конфигураций помогает определить влияние их характеристик на эффективность передачи, что важно для разработки более эффективных решений в области беспроводных коммуникаций.

Дополнительно представлен алгоритм размещения узлов маршрутизаторов (GPA) в WMN, основанный на адаптации метода «box covering». Исследование показало, что сила взаимодействия между узлами, определяемая их «зарядом» и расстоянием, влияет на общую производительность сети. Эффективное стратегическое размещение шлюзов повышает передачу данных и улучшает пропускную способность сети, подтверждая значимость расположения маршрутизаторов для функционирования WMN.

Эти результаты подчеркивают актуальность и значимость проведенного исследования, открывая перспективы для дальнейшего развития теоретических и практических аспектов в области беспроводной связи и телекоммуникационных систем. Результаты работы могут быть использованы для внедрения новых технологий и улучшений в реальных условиях, что, в свою очередь, будет способствовать повышению качества и надежности связи в различных приложениях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Meng, F., Chen, P., Wu, L., & Wang, X. Automatic modulation classification: A deep learning enabled approach //IEEE Transactions on Vehicular Technology. – 2018. – Т. 67. – №. 11. – С. 10760-10772.
2. Wang, Y., Gui, G., Gacanin, H., Ohtsuki, T., Sari, H., & Adachi, F. Transfer learning for semi-supervised automatic modulation classification in ZF-MIMO systems //IEEE Journal on Emerging and Selected Topics in Circuits and Systems. – 2020. – Т. 10. – №. 2. – С. 231-239.
3. Dileep, P., Singla, A., Das, D., & Bora, P. K. Deep learning-based automatic modulation classification over MIMO keyhole channels //IEEE Access. – 2022. – Т. 10. – С. 119566-119574.
4. Zhou, S., Yin, Z., Wu, Z., Chen, Y., Zhao, N., & Yang, Z. A robust modulation classification method using convolutional neural networks //EURASIP Journal on Advances in Signal Processing. – 2019. – Т. 2019. – С. 1-15.
5. Zhu Z., Nandi A. K. Automatic modulation classification: principles, algorithms and applications. – John Wiley & Sons, 2015.
6. Hazza, A., Shoaib, M., Alshebeili, S. A., & Fahad, A. An overview of feature-based methods for digital modulation classification //2013 1st international conference on communications, signal processing, and their applications (ICCSPA). – IEEE, 2013. – С. 1-6.
7. Al-Nuaimi, D. H., Hashim, I. A., Zainal Abidin, I. S., Salman, L. B., & Mat Isa, N. A. Performance of feature-based techniques for automatic digital modulation recognition and classification—A review //Electronics. – 2019. – Т. 8. – №. 12. – С. 1407.
8. Nanopoulos, A., Alcock, R., & Manolopoulos, Y. Nanopoulos A., Alcock R., Manolopoulos Y. Feature-based classification of time-series data //International Journal of Computer Research. – 2001. – Т. 10. – №. 3. – С. 49-61.
9. Dan, W., Xuemai, G., & Qing, G. A new scheme of automatic modulation classification using wavelet and WSVM //2005 2nd Asia Pacific Conference on Mobile Technology, Applications and Systems. – IEEE, 2005. – С. 5 pp.-5.
10. Park, H., Yang, J., Park, J., Kang, S. G., & Choi, J. K. A survey on peer-to-peer overlay network schemes //2008 10Th international conference on advanced communication technology. – IEEE, 2008. – Т. 2. – С. 986-988.
11. Ghasemzadeh, P., Banerjee, S., Hempel, M., & Sharif, H. Accuracy analysis of feature-based automatic modulation classification with blind modulation detection //2019 International Conference on Computing, Networking and Communications (ICNC). – IEEE, 2019. – С. 1000-1004.
12. Yuan, J., Zhao-Yang, Z., & Pei-Liang, Q. Modulation classification of communication signals //IEEE MILCOM 2004. Military Communications Conference, 2004. – IEEE, 2004. – Т. 3. – С. 1470-1476.
13. Prakasam P., Madheswaran M. Digital Modulation Identification Model Using Wavelet Transform and Statistical Parameters //Journal of Computer Systems, Networks, & Communications. – 2008.

14. Nandi A. K., Azzouz E. E. Algorithms for automatic modulation recognition of communication signals //IEEE Transactions on communications. – 1998. – Т. 46. – №. 4. – С. 431-436.
15. Li, J., He, C., Chen, J., & Wang, D. Automatic digital modulation recognition based on euclidean distance in hyperspace //IEICE transactions on communications. – 2006. – Т. 89. – №. 8. – С. 2245-2248.
16. Dobre O. A., Rajan S., Inkol R. Joint signal detection and classification based on first-order cyclostationarity for cognitive radios //EURASIP Journal on Advances in Signal Processing. – 2009. – Т. 2009. – С. 1-12..
17. Dobre, O. A., Abdi, A., Bar-Ness, Y., & Su, W. Cyclostationarity-based modulation classification of linear digital modulations in flat fading channels //Wireless Personal Communications. – 2010. – Т. 54. – С. 699-717.
18. Hsue, S. Z., Soliman, S. S. Automatic modulation classification using zero crossing // IEE Proceedings F (Radar and Signal Processing). – IET Digital Library, 1990. – Т. 137. – № 6. – С. 459-464.
19. Pedzisz, M., Mansour, A. Automatic modulation recognition of MPSK signals using constellation rotation and its 4th order cumulant // Digital Signal Processing. – 2005. – Т. 15. – № 3. – С. 295-304.
20. Xie, W., Liu, C., Zhang, S., Chen, H., Zhao, Z. Deep learning in digital modulation recognition using high order cumulants // IEEE Access. – 2019. – Т. 7. – С. 63760-63766.
21. Shi, W., Jiang, J., Guo, J., Huang, T., Hu, J., Zheng, Z. Particle swarm optimization-based deep neural network for digital modulation recognition // IEEE Access. – 2019. – Т. 7. – С. 104591-104600.
22. Hu, S., Ma, J., Xiong, W., Guo, J., He, X., Zhang, T. Deep neural network for robust modulation classification under uncertain noise conditions // IEEE Transactions on Vehicular Technology. – 2019. – Т. 69. – №. 1. – С. 564-577.Huang, S., Li, W., Zhang, Y., Liu, C., Zhang, J. Automatic modulation classification using compressive convolutional neural network // IEEE Access. – 2019. – Т. 7. – С. 79636-79643.
23. Lee, J. H., Kim, B., Chung, Y. Deep neural network-based blind modulation classification for fading channels // 2017 International Conference on Information and Communication Technology Convergence (ICTC). – IEEE, 2017. – С. 551-554.
24. Karra, K., Kuzdeba, S., Petersen, J. Modulation recognition using hierarchical deep neural networks // 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN). – IEEE, 2017. – С. 1-3.
25. Kim, B., Kim, S., Song, K., Choi, W., Lee, S. Deep neural network-based automatic modulation classification technique // 2016 International Conference on Information and Communication Technology Convergence (ICTC). – IEEE, 2016. – С. 579-582.
26. Bahloul, M. R., Boukadoum, M., Aliouane, N., Cherroun, H. Modulation classification for MIMO systems: State of the art and research directions // Chaos, Solitons & Fractals. – 2016. – Т. 89. – С. 497-505.
27. Jdid, B., Benyahia, R., Loukil, A., Othman, M., Hamdi, M. Machine learning based automatic modulation recognition for wireless communications: A comprehensive survey // IEEE Access. – 2021. – Т. 9. – С. 57851-57873.

28. Huynh-The, T., Hua, C., Nurunnabi, M., Kim, J. Automatic modulation classification: A deep architecture survey // IEEE Access. – 2021. – T. 9. – C. 142950-142971.
29. Peng, S., Sun, S., Yao, Y. D. A survey of modulation classification using deep learning: Signal representation and data preprocessing // IEEE Transactions on Neural Networks and Learning Systems. – 2021. – T. 33. – №. 12. – C. 7020-7038.
30. Zhang, F., Liu, L., Zhai, S., Han, J. Deep learning based automatic modulation recognition: Models, datasets, and challenges // Digital Signal Processing. – 2022. – T. 129. – C. 103650.
31. Zhou, R., Liu, F., Gravelle, C. W., Yu, L. Deep learning for modulation recognition: A survey with a demonstration // IEEE Access. – 2020. – T. 8. – C. 67366-67376.
32. Wang, Y., He, Q., Cheng, Y., Zhang, Z. Automatic modulation classification for MIMO systems via deep learning and zero-forcing equalization // IEEE Transactions on Vehicular Technology. – 2020. – T. 69. – №. 5. – C. 5688-5692.
33. Wang, Y., Xu, L., Zhou, W., Fang, X. Transfer learning for semi-supervised automatic modulation classification in ZF-MIMO systems // IEEE Journal on Emerging and Selected Topics in Circuits and Systems. – 2020. – T. 10. – №. 2. – C. 231-239.
34. Zhang, Z., Li, S., Ma, C., Wang, H. Modulation signal recognition based on information entropy and ensemble learning // Entropy. – 2018. – T. 20. – №. 3. – C. 198.
35. Zhang, Z., Zhu, Y., Chen, X., Liu, H. A method for modulation recognition based on entropy features and random forest // 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). – IEEE, 2017. – C. 243-246.
36. Wang, H., Zhang, X., Chen, P., Liu, W. A new method of cognitive signal recognition based on hybrid information entropy and DS evidence theory // Mobile Networks and Applications. – 2018. – T. 23. – C. 677-685.
37. Wei, Y., Fang, S., Wang, X. Automatic modulation classification of digital communication signals using SVM based on hybrid features, cyclostationary, and information entropy // Entropy. – 2019. – T. 21. – №. 8. – C. 745.
38. Cover, T. M., Thomas, J. A. Elements of Information Theory. – John Wiley & Sons, 1999.
39. Archer, E., Park, I. M., Pillow, J. W. Bayesian and quasi-Bayesian estimators for mutual information from discrete data // Entropy. – 2013. – T. 15. – №. 5. – C. 1738-1755.
40. Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P. Conditional entropy and mutual information // Numerical Recipes: The Art of Scientific Computing. – 2007. – T. 3.
41. Moulay, H., Djebbar, A. B., Dehri, B., & Dayoub, I. Dendrogram-based Artificial Neural Network modulation classification for dual-hop cooperative relaying communications // Physical Communication. – 2022. – T. 55. – C. 101929.

42. 42. Mingoti S. A., Lima J. O. Comparing SOM neural network with Fuzzy c-means, K-means and traditional hierarchical clustering algorithms //European journal of operational research. – 2006. – Т. 174. – №. 3. – С. 1742-1759.
43. Guo G. et al. KNN model-based approach in classification //On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings. – Springer Berlin Heidelberg, 2003. – С. 986-996.
44. Awad, M., Khanna, R., Awad, M., & Khanna, R. Support vector machines for classification //Efficient learning machines: Theories, concepts, and applications for engineers and system designers. – 2015. – С. 39-66.
45. Zhang, H., Nie, R., Lin, M., Wu, R., Xian, G., Gong, X., & Luo, R. (2021). A deep learning-based algorithm with multi-level feature extraction for automatic modulation recognition // Wireless Networks, – 27(7), – 4665-4676.
46. Ding, S., Zhu, Z., Zhang, X. An overview on semi-supervised support vector machine // Neural Comput. Appl. – 2017. – Т. 28. – №. 5. – С. 969-978.
47. Kovács, P. T., Nagy, M., Molontay, R. Comparative analysis of box-covering algorithms for fractal networks // Applied Network Science. – 2021. – Т. 6. – №. 1. – С. 73.
48. Zheng, W., Sun, C., Wang, Y., Qiu, X. Fractal analysis of mobile social networks // Chinese Physics Letters. – 2016. – Т. 33. – №. 3. – С. 038901.
49. Wen, T., & Cheong, K. H. The fractal dimension of complex networks: A review // Information Fusion, – 2021. – *73*, 87-102.
50. Song, C., Havlin, S., Makse, H. A. How to calculate the fractal dimension of a complex network: The box covering algorithm // Journal of Statistical Mechanics: Theory and Experiment. – 2007. – Т. 2007. – №. 03. – С. P03006.
51. Malaguti E., Toth P. A survey on vertex coloring problems //International transactions in operational research. – 2010. – Т. 17. – №. 1. – С. 1-34.
52. Kosowski A., Manuszewski K. Classical coloring of graphs //Contemporary Mathematics. – 2004. – Т. 352. – С. 1-20.
53. Zhang, Z., Zhou, S., Zou, F.: Self-similarity, small-world, scale-free scaling, disassortativity, and robustness in hierarchical lattices. Eur. Phys. J. B 56(3), 259–271 (2007).
54. Deng Y., Zheng W., Pan Q. Performance evaluation of fractal dimension method based on box-covering algorithm in complex network //2016 IEEE 20th international conference on computer supported cooperative work in design (CSCWD). – IEEE, 2016. – С. 682-686.
55. Song C., Havlin S., Makse H. A. Origins of fractality in the growth of complex networks //Nature physics. – 2006. – Т. 2. – №. 4. – С. 275-281.
56. Шеннон К. Работы по теории информации и кибернетике. – Рипол Классик, 1963.
57. Волькенштейн М. В. Энтропия и информация. – 1986.
58. Martin N. F. G., England J. W. Mathematical theory of entropy. – Cambridge university press, 2011. – №. 12.
59. Бриллюэн Л. Наука и теория информации. – Рипол Классик, 2013.

60. Мартин Н И. Д. Математическая теория энтропии. – Мир, 1988.
61. Duan S., Wen T., Jiang W. A new information dimension of complex network based on Rényi entropy //Physica A: Statistical Mechanics and its Applications. – 2019. – Т. 516. – С. 529-542.
62. Zhang, Q., Luo, C., Li, M., Deng, Y., & Mahadevan, S. Tsallis information dimension of complex networks //Physica A: Statistical Mechanics and its Applications. – 2015. – Т. 419. – С. 707-717.
63. Wei, D., Wei, B., Hu, Y., Zhang, H., & Deng, Y. A new information dimension of complex networks //Physics Letters A. – 2014. – Т. 378. – №. 16-17. – С. 1091-1094.
64. Wen T., Jiang W. An information dimension of weighted complex networks //Physica A: Statistical Mechanics and its Applications. – 2018. – Т. 501. – С. 388-399.
65. Taleb, S. M., Meraihi, Y., Gabis, A. B., Mirjalili, S., & Ramdane-Cherif, A. Nodes placement in wireless mesh networks using optimization approaches: a survey //Neural Computing and Applications. – 2022. – Т. 34. – №. 7. – С. 5283-5319.
66. Wzorek M., Berger C., Doherty P. Router and gateway node placement in wireless mesh networks for emergency rescue scenarios //Autonomous intelligent systems. – 2021. – Т. 1. – С. 1-30.
67. Sakamoto S. Implementation of a cuckoo search intelligent simulation system for node placement problem in wireless mesh networks: Performance evaluation for tuning hyperparameters Pa and γ //Internet of Things. – 2024. – С. 101288.
68. Binh, L. H., & Duong, T. V. T. A novel and effective method for solving the router nodes placement in wireless mesh networks using reinforcement learning. // Plos one, – 2024. – 19(4), e0301073.
69. Taleb, S. M., Meraihi, Y., Mirjalili, S., Acheli, D., Ramdane-Cherif, A., & Gabis, A. B. Mesh router nodes placement for wireless mesh networks based on an enhanced moth–flame optimization algorithm //Mobile Networks and Applications. – 2023. – Т. 28. – №. 2. – С. 518-541.
70. Agrawal, R., Faujdar, N., Romero, C. A. T., Sharma, O., Abdulsahib, G. M., Khalaf, O. I., ... & Ghoneim, O. A. Classification and comparison of ad hoc networks: A review //Egyptian Informatics Journal. – 2023. – Т. 24. – №. 1. – С. 1-25.
71. Rubinstein, M. G., Moraes, I. M., Campista, M. E. M., Costa, L. H. M., & Duarte, O. C. M. A survey on wireless ad hoc networks //IFIP International Conference on Mobile and Wireless Communication Networks. – Boston, MA : Springer US, 2006. – С. 1-33.
72. Ramphull, D., Mungur, A., Armoogum, S., & Pudaruth, S. A review of mobile ad hoc NETwork (MANET) Protocols and their Applications //2021 5th international conference on intelligent computing and control systems (ICICCS). – IEEE, 2021. – С. 204-211.
73. Mirza S., Bakshi S. Z. Introduction to MANET //International research journal of engineering and technology. – 2018. – Т. 5. – №. 1. – С. 17-20.
74. Sharmila S., Shanthi T. A survey on wireless ad hoc network: Issues and implementation //2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS). – IEEE, 2016. – С. 1-6.

75. Mitra P., Poellabauer C. Emergency response in smartphone-based mobile ad-hoc networks //2012 IEEE International Conference on Communications (ICC). – IEEE, 2012. – С. 6091-6095.
76. Parvin J. R. An overview of wireless mesh networks //Wireless mesh networks-security, architectures and protocols. – 2019.
77. Yang K. Wireless sensor networks. – 2014.
78. Kandris, D., Nakas, C., Vomvas, D., & Koulouras, G. Applications of wireless sensor networks: an up-to-date survey //Applied system innovation. – 2020. – Т. 3. – №. 1. – С. 14.
79. Yick J., Mukherjee B., Ghosal D. Wireless sensor network survey //Computer networks. – 2008. – Т. 52. – №. 12. – С. 2292-2330.
80. Benyamina D., Hafid A., Gendreau M. Wireless mesh networks design—A survey //IEEE Communications surveys & tutorials. – 2011. – Т. 14. – №. 2. – С. 299-310.
81. Eslami M., Karimi O., Khodadadi T. A survey on wireless mesh networks: Architecture, specifications and challenges //2014 IEEE 5th control and system graduate research colloquium. – IEEE, 2014. – С. 219-222.
82. Alabady S. A., Salleh M. F. M. Overview of Wireless Mesh Networks //J. Commun. – 2013. – Т. 8. – №. 9. – С. 586-599.
83. Chai Y., Zeng X. J., Liu Z. The future of wireless mesh network in next-generation communication: a perspective overview //Evolving Systems. – 2024. – С. 1-14.
84. Parvin J. R. An overview of wireless mesh networks //Wireless mesh networks-security, architectures and protocols. – 2019.
85. Kaur J., Singh H. Several Routing Protocols, Features and Limitations for Wireless Mesh Network (WMN): A Review //ICDSMLA 2021: Proceedings of the 3rd International Conference on Data Science, Machine Learning and Applications. – Singapore : Springer Nature Singapore, 2023. – С. 187-200.
86. Alotaibi E., Mukherjee B. A survey on routing algorithms for wireless ad-hoc and mesh networks //Computer networks. – 2012. – Т. 56. – №. 2. – С. 940-965.
87. Siraj M. A Survey on routing algorithms and routing metrics for Wireless Mesh Networks //World Applied Sciences Journal. – 2014. – Т. 30. – №. 7. – С. 870-886.
88. Al-Karaki J. N., Al-Mashaqbeh G. A., Bataineh S. Routing protocols in wireless mesh networks: a survey //International Journal of Information and Communication Technology. – 2017. – Т. 11. – №. 4. – С. 445-495.
89. Kaur J., Singh H. Several Routing Protocols, Features and Limitations for Wireless Mesh Network (WMN): A Review //ICDSMLA 2021: Proceedings of the 3rd International Conference on Data Science, Machine Learning and Applications. – Singapore : Springer Nature Singapore, 2023. – С. 187-200.
90. Dhanasekaran, S., Ramalingam, S., Baskaran, K., & Vivek Karthick, P. (2024). Efficient distance and connectivity-based traffic density stable routing protocol for vehicular Ad Hoc networks. IETE Journal of Research, 70(2), 1150-1166.
91. Al Ajrawi, S., & Tran, B. (2024). Mobile wireless ad-hoc network routing protocols comparison for real-time military application. // Spatial Information Research, 32(1), 119-129.

92. Atti, M., & Yogi, M. K. (2024). Self-Assured Opportunistic Routing Algorithm for Ad Hoc Networks. // Journal of IoT-based Distributed Sensor Networks, 1(1), 15.
93. Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., & Viennot, L. Optimized link state routing protocol for ad hoc networks //Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. – IEEE, 2001. – С. 62-68.
94. Prajapati S., Patel N., Patel R. Optimizing performance of OLSR protocol using energy based MPR selection in MANET //2015 Fifth International Conference on Communication Systems and Network Technologies. – IEEE, 2015. – С. 268-272.
95. Wheeb A. H., Al-Jamali N. A. S. Performance analysis of OLSR protocol in mobile ad hoc networks //iJIM. – 2022. – Т. 16. – №. 01. – С. 107.
96. Barolli, L., Ikeda, M., De Marco, G., Durresi, A., & Xhafa, F. Performance analysis of OLSR and BATMAN protocols considering link quality parameter //2009 International Conference on Advanced Information Networking and Applications. – IEEE, 2009. – С. 307-314.
97. Chakeres I. D., Belding-Royer E. M. AODV routing protocol implementation design //24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings. – IEEE, 2004. – С. 698-703.
98. MMaurya, P. K., Sharma, G., Sahu, V., Roberts, A., Srivastava, M., & Scholar, M. T. An overview of AODV routing protocol //International Journal of Modern Engineering Research (IJMER). – 2012. – Т. 2. – №. 3. – С. 728-732.
99. Al-Dhief, F. T., Sabri, N., Salim, M. S., Fouad, S., & Aljunid, S. A. MANET routing protocols evaluation: AODV, DSR and DSDV perspective //MATEC web of conferences. – EDP Sciences, 2018. – Т. 150. – С. 06024.
100. Singh M., Kumar S. A survey: Ad-hoc on-demand distance vector (AODV) protocol //International Journal of Computer Applications. – 2017. – Т. 161. – №. 1. – С. 38-44.
101. Javaid A. Understanding Dijkstra's algorithm //Available at SSRN 2340905. – 2013.
102. Gunawan, R. D., Napianto, R., Borman, R. I., & Hanifah, I. Implementation Of Dijkstra’s Algorithm In Determining The Shortest Path (Case Study: Specialist Doctor Search In Bandar Lampung) //Int. J. Inf. Syst. Comput. Sci. – 2019. – Т. 3. – №. 3. – С. 98-106.
103. Verma, D., Messon, D., Rastogi, M., & Singh, A. Comparative study of various approaches of Dijkstra algorithm //2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS). – IEEE, 2021. – С. 328-336.
104. Huang Y. Research on the improvement of Dijkstra algorithm in the shortest path calculation //2017 4th International Conference on Machinery, Materials and Computer (MACMC 2017). – Atlantis Press, 2018. – С. 745-749.
105. Sim K. M., Sun W. H. Ant colony optimization for routing and load-balancing: survey and new directions //IEEE transactions on systems, man, and cybernetics-Part A: systems and humans. – 2003. – Т. 33. – №. 5. – С. 560-572.

106. Zhang, H., Wang, X., Memarmoshrefi, P., & Hogrefe, D. A survey of ant colony optimization-based routing protocols for mobile ad hoc networks //IEEE access. – 2017. – Т. 5. – С. 24139-24161.
107. Gao, S., Wang, Y., Cheng, J., Inazumi, Y., Tang, Z. Ant colony optimization with clustering for solving the dynamic location routing problem //Applied Mathematics and Computation. – 2016. – Т. 285. – С. 149-173.
108. Chatterjee S., Das S. Ant colony optimization based enhanced dynamic source routing algorithm for mobile Ad-hoc network //Information sciences. – 2015. – Т. 295. – С. 67-90.
109. Edsger D., Misa T. J. An interview with edsger w. Dijkstra //Communications of the ACM. – 2010. – Т. 53. – №. 8. – С. 41-47. DOI: 10.1145/1787234.1787249.
110. Dorigo M., Gambardella L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem //IEEE Transactions on evolutionary computation. – 1997. – Т. 1. – №. 1. – С. 53-66. DOI: 10.1109/4235.585892.
111. Rohan, Neeraj. Behavior of Cascaded Binary Symmetric Channel for different number of cascades //2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC). – IEEE, 2020. – С. 1159-1164., doi: 10.1109/I-SMAC49090.2020.9243336.
112. Z. Zhanabaev, T. Grevtseva, S. Akhtanov and A. Serikbayev, Informational-Entropic Routing in Wireless Network //2018 International Conference on Computing and Network Communications (CoCoNet). – IEEE, 2018. – С. 14-17., doi: 10.1109/CoCoNet.2018.8476885.
113. Jain M., Pal M. K. Comparative study of AODV And OLSR routing protocols /Soft Computing and Signal Processing: Proceedings of ICSCSP 2018, Volume 2. – Springer Singapore, 2019. – С. 309-317.
114. Ussipov, N., Akhtanov, S., Zhanabaev, Z., Turlykozhayeva, D., Karibayev, B., Namazbayev, T., Tang, X. Automatic modulation classification for MIMO system based on the mutual information feature extraction //IEEE Access. – 2024.
115. Balzanella A., Verde R. Histogram-based clustering of multiple data streams //Knowledge and Information Systems. – 2020. – Т. 62. – №. 1. – С. 203-238.
116. Tayakout, H., Dayoub, I., Ghanem, K., & Bousbia-Salah, H. Automatic modulation classification for D-STBC cooperative relaying networks //IEEE Wireless Communications Letters. – 2018. – Т. 7. – №. 5. – С. 780-783.
117. Akhtanov, S., Turlykozhayeva, D., Ussipov, N., Ibraimov, M., Zhanabaev, Z. Centre including eccentricity algorithm for complex networks // Electronics Letters. – 2022. – Т. 58. – №. 7. – С. 283-285.
118. Жанабаев З. Ж. Квазиканоническое распределение Гиббса и масштабная инвариантность хаотических систем //Мат. 5-й Межд. конф.» Хаос и структ. в нелин. сист»., 15-17 июня. – 2006. – С. 15.
119. Zhanabaev Z., Akhtanov S., Turlykozhayeva D., Ussipov N., & Ibraimov M., Cluster router based on eccentricity //Eurasian Physical Technical Journal. – 2022. – Т. 19. – №. 3 (41). – С. 84-90.
120. Turlykozhayeva D. A., Akhtanov S. N., Baigaliyeva A. N., Temesheva S. A., Zhexsabay D. M., Zaydin M., Skabylov A. A. EVALUATING ROUTING

ALGORITHMS ACROSS DIFFERENT WIRELESS MESH NETWORK TOPOLOGIES USING NS-3 SIMULATOR //Eurasian Physical Technical Journal. – 2024. – Т. 21. – №. 2.
121. Turlykozhayeva D., Waldemar W., Akhmetali A., Ussipov N., Temesheva S., Akhtanov S. Single Gateway Placement in Wireless Mesh Networks // Physical Sciences and Technology. – 2024. – Vol. 11. – No. 1-2.– С. 5-8.
122. Zhang, H., Wei, D., Hu, Y., Lan, X., & Deng, Y. Modeling the self-similarity in complex networks based on Coulomb's law //Communications in Nonlinear Science and Numerical Simulation. – 2016. – Т. 35. – С. 97-104.
123. Жанабаев З. Ж., Усипов Н. М. INFORMATION-ENTROPY METHOD FOR DETECTING GRAVITATIONAL WAVE SIGNALS //Eurasian Physical Technical Journal. – 2023. – Т. 20. – №. 2 (44). – С. 79-86.
124. Turlykozhayeva D. A., Ussipov N. M., Baigaliyeva A. N., Temesheva S. A., Bolysbay A. T., Abrahmatova G. A., & Akhtanov S. T. ROUTING METRIC AND PROTOCOL FOR WIRELESS MESH NETWORK BASED ON INFORMATION ENTROPY THEORY //Eurasian Physical Technical Journal. – 2023. – Т. 20. – №. 4 (46). – С. 90-98.
125. Turlykozhayeva, D. A., Akhtanov, S. N., Zhanabaev, Z. Z., Ussipov, N. M., & Akhmetali, A. (2025). A routing algorithm for wireless mesh network based on information entropy theory. *IET Communications*, *19*(1), e70011.

ПРИЛОЖЕНИЕ А – ПАТЕНТ

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ РЕСПУБЛИКА КАЗАХСТАН

REPUBLIC OF KAZAKHSTAN

ПАТЕНТ
PATENT

№ 7327

ПАЙДАЛЫ МОДЕЛЬГЕ / НА ПОЛЕЗНУЮ МОДЕЛЬ / FOR UTILITY MODEL

(21) 2022/0471.2

(22) 26.05.2022

(45) 05.08.2022

(54) Эксцентриситетке негізделген кластерлік маршрутизатор
Кластерный маршрутизатор на основе эксцентриситета
Eccentricity based cluster router

(73) Ахтанов Саят Нусипбекович (KZ); Турлыкожаева Дана Абдикумаровна (KZ)
Akhtanov Sayat Nusipbekovich (KZ); Turlykozhayeva Dana Abdikumarovna (KZ)

(72) Үсіпов Нұржан Мұсайыпұлы (KZ)
Ибраимов Маргулан Касенович (KZ)
Ussipov Nurzhan Musayipuli (KZ)
Ibraimov Margulan Kasenovich (KZ)

ЭЦҚ қол қойылды
Подписано ЭЦП
Signed with EDS

Е. Оспанов
Е. Оспанов
Y. Ospanov

«Ұлттық зияткерлік меншік институты» РМК директоры
Директор РГП «Национальный институт интеллектуальной собственности»
Director of the «National Institute of Intellectual Property» RSE

Патентті күшінде ұстау ақысы уақытылы төленген жағдайда патенттің күші
Қазақстан Республикасының бүкіл аумағында қолданылады.

Патентке пайдалы модельдің толық сипатта масы www.kazpatent.kz ресми сайтында
«Қазақстан Республикасының пайдалы модельдерінің мемлекеттік тізілімі» бөлімінде қолжетімді

* * *

Действие патента распространяется на всю территорию Республики Казахстан
при условии своевременной оплаты поддержания патента в силе.

Полное описание полезной модели к патенту доступно на официальном сайте www.kazpatent.kz
в разделе «Государственный реестр полезных моделей Республики Казахстан».

* * *

Subject to timely payment for the maintenance of the patent in force
the patent shall be effective on the entire territory of the Republic of Kazakhstan.

Full description of the patent for utility model are available on the official website www.kazpatent.kz
in the section «State Register of Utility Models of the Republic of Kazakhstan».

Қазақстан Республикасы Әділет министрлігінің
«Ұлттық зияткерлік меншік институты» РМК
Нұр-Сұлтан қаласы, Мәңгілік Ел даңғылы, ғимарат 57А

РГП «Национальный институт интеллектуальной собственности»
Министерства юстиции Республики Казахстан
Город Нур-Султан, проспект Мангилик Ел, здание 57А

«National Institute of Intellectual Property» RSE,
Ministry of Justice of the Republic of Kazakhstan
Nur-Sultan, 57A Mangilik El Avenue

Тел./Tel.: +7 (7172) 62-15-15
E-mail: kazpatent@kazpatent.kz
Website: www.kazpatent.kz

ПРИЛОЖЕНИЕ Б – ПАТЕНТ

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ РЕСПУБЛИКА КАЗАХСТАН

REPUBLIC OF KAZAKHSTAN

ПАТЕНТ
PATENT

№ 10114

ПАЙДАЛЫ МОДЕЛЬГЕ / НА ПОЛЕЗНУЮ МОДЕЛЬ / FOR UTILITY MODEL

(21) 2024/1526.2

(22) 23.11.2024

(45) 24.01.2025

(54) Ақпараттық-энтропиялық талдауға негізделген сымсыз желілерге арналған маршрутизатор
Маршрутизатор для беспроводных сетей на основе информационно-энтропииного анализа
Router for wireless networks based on information-entropy analysis

(73) Ахтанов Саят Нусипбекович (KZ)
Akhtanov Sayat Nusipbekovich (KZ)

(72) Ахтанов Саят Нусипбекович (KZ) — Akhtanov Sayat Nusipbekovich (KZ)
Турлыкожаева Дана Абдикумаровна (KZ) — Turlykozhayeva Dana Abdikumarovna (KZ)
Темешева Сымбат Айдынқызы (KZ) — Temesheva Symbat Aidynkyzy (KZ)
Үсіпов Нұржан (KZ) — Ussipov Nurzhan (KZ)
Сқабылов Әлішер Әлиұлы (KZ) — Skabylov Alisher Aliuly (KZ)

ЭЦҚ қол қойылды — С. Ахметов
Подписано ЭЦП — С. Ахметов
Signed with EDS — S. Akhmetov

«Ұлттық зияткерлік меншік институты» РМК директоры
Директор РГП «Национальный институт интеллектуальной собственности»
Director of the «National Institute of Intellectual Property» RSE

Патентті күшінде ұстау ақысы уақытылы төленген жағдайда патенттің күші
Қазақстан Республикасының бүкіл аумағында қолданылады.

Патентке пайдалы модельдің толық сипатта масы www.kazpatent.kz ресми сайтында
«Қазақстан Республикасының пайдалы модельдерінің мемлекеттік тізілімі» бөлімінде қолжетімді

* * *

Действие патента распространяется на всю территорию Республики Казахстан
при условии своевременной оплаты поддержания патента в силе.

Полное описание полезной модели к патенту доступно на официальном сайте www.kazpatent.kz
в разделе «Государственный реестр полезных моделей Республики Казахстан».

* * *

Subject to timely payment for the maintenance of the patent in force
the patent shall be effective on the entire territory of the Republic of Kazakhstan.

Full description of the patent for utility model are available on the official website www.kazpatent.kz
in the section «State Register of Utility Models of the Republic of Kazakhstan».

Қазақстан Республикасы Әділет министрлігінің
«Ұлттық зияткерлік меншік институты» РМК
Астана қаласы, Мәңгілік Ел даңғылы, ғимарат 57А

РГП «Национальный институт интеллектуальной собственности»
Министерства юстиции Республики Казахстан
Город Астана, проспект Мангилик Ел, здание 57А

«National Institute of Intellectual Property» RSE,
Ministry of Justice of the Republic of Kazakhstan
Astana, 57A Mangilik El Avenue

Тел./Tel.: +7 (7172) 62-15-15
E-mail: kazpatent@kazpatent.kz
Website: www.qazpatent.kz

ПРИЛОЖЕНИЕ С – ПРОГРАММНЫЙ КОД

```python
import matplotlib.pyplot as plt
import math
import random
import numpy as np

'''''''''''''''''' Grid ''''''''''''''''''
# # Create a grid layout for the nodes
# num_rows = 5  # Change this value to modify the grid layout
# num_cols = 5  # Change this value to modify the grid layout
# num_nodes = num_rows * num_cols  # Calculate total nodes

# positions = {}
# for i in range(num_nodes):
#     row = i // num_cols
#     col = i % num_cols
#     # Normalize row and col values to fit within the range [0, 1]
#     normalized_row = row / (num_rows - 1)  # Subtract 1 to ensure the range is [0, 1)
#     normalized_col = col / (num_cols - 1)
#     new_pos = (normalized_row, normalized_col)
#     positions[i + 1] = new_pos  # Incrementing ID by 1 to match the second code's output style

'''''''''''''''''' Random uniform ''''''''''''''''''
# Define the number of nodes
num_nodes = 25

# Generate random positions for nodes with distances greater than 1
min_distance = 0.1

def check_distance(point1, point2, min_dist):
    return math.sqrt((point2[0] - point1[0]) ** 2 + (point2[1] - point1[1]) ** 2) >= min_dist

positions = {}
while len(positions) < num_nodes:
    x = random.uniform(0, 1)
    y = random.uniform(0, 1)
    new_pos = (x, y)

    if all(check_distance(new_pos, existing_pos, min_distance) for existing_pos in positions.values()):
        positions[len(positions) + 1] = new_pos

''''''''''''''''''''''''''''''''''''''''''''''''''
G = 1
coverage_radius = 0.25

plt.figure(figsize=(6, 6))
# Plot the nodes
for i, pos in positions.items():
    x, y = pos
    plt.plot(x, y, 'o', markersize=8, color='blue')
    # plt.text(x + 0.01, y + 0.01, str(i), fontsize=8)

forces = {}
total_forces = {}

def euclidean_distance(pos1, pos2):
    return math.sqrt((pos2[0] - pos1[0]) ** 2 + (pos2[1] - pos1[1]) ** 2)

for node, pos in positions.items():
    forces[node] = []
    total_force = 0
    for other_node, other_pos in positions.items():
        if node != other_node:
            distance = euclidean_distance(pos, other_pos)
            if distance <= coverage_radius:
                force = G / (distance ** 2)
                forces[node].append((other_node, force))
                total_force += force
    total_forces[node] = total_force

sorted_total_forces = sorted(total_forces.items(), key=lambda x: x[1], reverse=True)
top_nodes = [node for node, _ in sorted_total_forces[:4]]
worst_node = sorted_total_forces[-1][0]

connected_nodes = set()
for node, force_list in forces.items():
    if node == top_nodes[0]:
        for connected_node, _ in force_list:
            connected_nodes.add(connected_node)
connected_nodes.add(top_nodes[0])

for i, pos in positions.items():
    x, y = pos
    if i in top_nodes:
        if i == top_nodes[0]:
            plt.plot(x, y, 'o', markersize=8, color='gold')
            circle = plt.Circle((x, y), coverage_radius , color='gold', alpha=0.2)
            plt.gca().add_patch(circle)
        #else:
            #plt.plot(x, y, 'o', markersize=10, color='red')
    # elif i == worst_node:
    #     plt.plot(x, y, 'o', markersize=10, color='black')
    else:
        plt.plot(x, y, 'o', markersize=8, color='blue')

print("Top 4 best nodes:", top_nodes)
print("Worst node:", worst_node)
print("Nodes connected to top node:", list(connected_nodes))

plt.xlabel('X')
plt.ylabel('Y')
# plt.title('GateForce Algorithm')
plt.tight_layout()
plt.xlim(0,1)
plt.ylim(0,1)
plt.savefig('random.png', dpi=600)

# Convert positions dictionary to a NumPy array
positions_array = np.array(list(positions.values()))
#np.savetxt('10_pos.txt', positions_array, fmt='%.6f')

forces_array = np.array(list(total_forces.values()))
# np.savetxt('grid_forces.txt', forces_array, fmt='%.6f')
```

```python
import networkx as nx
import numpy as np


def CIE (network, g, rb):
    unburned = set(network.graph.nodes())
    if rb == 0:
        boxes = [[node] for node in network.graph.nodes()]
    else:
        boxes = []
        sp=dict(nx.shortest_path_length(g, weight='length'))
        b={i: sp[(i)][max(sp[(i)], key=sp[(i)].get)] for i in range(0,len(sp))}
        x1=[]
        for i in range(0,len(b),1):
            x1.append(float(b[(i)]))
        x1=np.array(x1)
        n=0
        y1=max(x1)
        j=np.where(x1>=max(x1-rb))
        while unburned:
            j=np.where(x1>=max(x1-rb))
            x3=x1[j]
            jj=np.argmin(x3)
            jjj=j[0][jj]
            x2=np.array(list(b))
            seed = int(x2[jjj])
            box = list(network.ball_of_seed(seed, rb).intersection(unburned))
            if box:
                boxes.append(box)
                unburned -= set(box)
                x1[boxes[n]]=0
                n+=1
    return(boxes)

```

```python
import networkx as nx
import numpy as np
import CIE
import network
import matplotlib.pyplot as plt
from itertools import combinations
import generators
import random
import ProbsdistTh

def probx(p):
    P=np.array([p ,1-p])
    return P


def prob(p1, sigma):
    P=np.array([[1-p1, sigma], [p1, 1-sigma]])
    return P


def Iyx(P):
    HH=(-P[0]*np.log2(P[0])+-P[1]*np.log2(P[1]))
    return HH

def Pym(Pyx):
    Py=[]
    for i in range (len(Pyx)):
        s=0
        for j in range (len(Pyx[0])):
            s+=Pyx[i][j]#вывод суммы отдельной строки
        Py.append(s)#с занесением в одномерный массив
    return Py

```

```python
def Infroute(G, u, v, sigma, N, return_path_directions=False):
    """
    G is the graph in question
    u is the starting node
    v is the destination node

    Returns path, distance
    """
    visited = set()
    unvisited = set(G.nodes)
    Iy = {u:0}

    shortest_paths = {u:[u]}
    for node in unvisited:
        if node == u:
            continue
        else:
            Iy[node] =0
    cur_node = u
    #All informations equals zero
    # print("Iy=", Iy)


    weights = nx.get_edge_attributes(G, "weight")
    Pxi = nx.get_node_attributes(G, "Px")
    # print("weights=", weights)
    # print("Px=", Pxi)


    #All channel error probabilities zero

    Pr=[]
    for v1 in range(N):
    #list(nx.nodes(G)):
        Pr.append(prob(0,0))
    # print("initial probabilities", Pr)


    Pxs=probx(Pxi[cur_node])
    # print("Pxs=", Pxs)


    while len(unvisited) > 0:
        if cur_node == v:
            break

        # if min([Hy[node] for node in unvisited]) == np.inf:
        #     print('There is no path between u and v.')
        #     return np.nan
    #     # Pull up neighbors
        neighbors = G[cur_node]

        # print("cur_node=", cur_node)


        for node in neighbors:
            # print("neigh_node=", node)
            # print("edge=", (cur_node, node))
            if (cur_node, node) in weights:
                pij=np.dot(Pr[cur_node], prob( weights[(cur_node, node)],sigma))
            else:
                pij=np.dot(Pr[cur_node], prob(weights[(node, cur_node)], sigma))

            # print("Pusl=", pij)

            # Pyx=
            # Pyx=np.dot(Pxs, pij)


            Pyx=Pxs*pij
            # print("Pyx=", Pyx)
            Py=Pym(Pyx)

            # print("Py=", Py)



            Iynax=Iyx(Py)

            # print("Iyx=", Iynax)

            # Future update: Add weight update for weighted graphs
            # Set either the distance through the current node or a previous shorter path
            if Iy[node] < Iynax:
                Iy[node] = Iynax
                Pr[node]=pij
                # print(node)
                shortest_paths[node] = shortest_paths[cur_node] + [node]




            #

    #     # Mark current node as visited
        visited.add(cur_node)
        unvisited.remove(cur_node)
        cur_node = sorted([(node, Iy[node]) for node in unvisited], key=lambda x:x[1], reverse=True)[0][0] #
        # print(cur_node)
        # break

```

```python
def boxroute(G, source, target, N):

    G1=network.network(G)
    a=CIE.CIE(G1, G, 1)

    print("boxes=", a)
    G2=nx.Graph()

    #Построение сети кластеров и определение узлов между кластерами

    comb = combinations(a, 2)
    # weights = nx.get_edge_attributes(G,'weight')

    Links=G.edges()

    for i in list(comb):
        # print (i)

        for j in Links:
            if j[0] in i[0]:
                if j[1] in i[1]:
                    G2.add_edge(a.index(i[0]),a.index(i[1]), labels=j)

            if j[0] in i[1]:
                if j[1] in i[0]:
                    G2.add_edge(a.index(i[1]),a.index(i[0]), labels=j)


    sigma=0.0786
    #source=0
    #target=13
    k=0
    for j in a:
        if target in j:
            ct=k
        if source in j:
            cs=k
        k=k+1
    # print(cs, ct)

    if ct==cs:
        GG=G.subgraph(a[ct])


        # nx.draw_networkx(GG, with_labels=True, node_size=500)


        # path=nx.shortest_path(GG, source, target, weight='length')

        path0=Infroute(GG, source, target, sigma ,N)
        pathe=path0[0]
        # print(path0[0])

    else:
        A=nx.shortest_path(G2, cs, ct, weight='length')
        # A1=Infroute(G2, cs, ct, sigma ,N)
        # print("A=", A1)
        labels = nx.get_edge_attributes(G2, 'labels')
        labelss={(i[1], i[0]): labels[(i)] for i in labels}
        labels.update(labelss)
        SG=[]
        for i in A:
            SG.append(G.subgraph(a[i]))

        # path=[]
        pathe=[]
        for j in range(len(A)-1):
            # print(j)
            lab=labels[(A[j],A[j+1])]
            # print(lab)

            if j==0:
                if lab[0] in SG[0]:
                    if source==lab[0]:
                        # path.append(source)
                        pathe.append(source)
                        ns=lab[1]
                    else:
                        # path.append(nx.shortest_path(SG[j], source, lab[0], weight='length'))
                        path0=Infroute(SG[j], source, lab[0], sigma ,N)
                        pathe.append(path0[0])
                        print("IIII", pathe[0])
                        ns=lab[1]

                else:
                    # path.append(nx.shortest_path(SG[j], source, lab[1], weight='length'))
                    path0=Infroute(SG[j], source, lab[1], sigma ,N)
                    pathe.append(path0[0])
                    ns=lab[0]
            else:
                if ns==lab[0]:
                    # path.append(ns)
                    pathe.append(ns)
                    ns=lab[1]
                elif ns==lab[1]:
                    # path.append(ns)
                    pathe.append(ns)

                    ns=lab[0]
                else:
                    if lab[0] in SG[j]:
                        # path.append(nx.shortest_path(SG[j], ns, lab[0], weight='length'))
                        path0=Infroute(SG[j], ns, lab[0], sigma ,N)
                        pathe.append(path0[0])
                        ns=lab[1]
                    else:
                        # path.append(nx.shortest_path(SG[j], ns, lab[1], weight='length'))
                        path0=Infroute(SG[j], ns, lab[1], sigma ,N)
                        pathe.append(path0[0])

                        ns=lab[0]

        # path.append(nx.shortest_path(SG[len(A)-1], ns, target, weight='length'))
        path0=Infroute(SG[len(A)-1], ns, target, sigma ,N)
        pathe.append(path0[0])
        # print(path)

        # path = [item if isinstance(item, int) else item for sublist in path for item in (sublist if isinstance(sublist, list) else [sublist])]
        pathe = [item if isinstance(item, int) else item for sublist in pathe for item in (sublist if isinstance(sublist, list) else [sublist])]

    return pathe
```

```
N=30
G = nx.complete_graph(N)
pos = nx.spring_layout(G)
for i in pos:
    pos[i]=pos[i]*10000
        # print(pos[i]+1)

#Задание вероятности узла
Px=0.51
nx.set_node_attributes(G, Px, "Px")

for ix in range(1,len(G)):
    G.nodes[ix]["Px"]=random.randint(1,99)/100


#Расчет растояния и условной вероятности между узлами

Edges=G.edges()
Dist=[]
p=[]
for j in Edges:
    # print(j[0], j[1])
    dist=np.sqrt((pos[j[0]][0]-pos[j[1]][0])**2+(pos[j[0]][1]-pos[j[1]][1])**2)
    Dist.append(round(dist,2))
    pr=ProbsdistTh.Probsdist(dist)

    p.append(pr)


k=0
for j in Edges:
    G.add_edge(j[0], j[1], length=Dist[k], weight=p[k])
    k=k+1

elarge = [(u, v) for (u, v, d) in G.edges(data=True) if d["length"] > 10000]
esmall = [(u, v) for (u, v, d) in G.edges(data=True) if d["length"] <= 10000]

#удалить лишние линки
for j in elarge:
    G.remove_edge(j[0], j[1])


# nx.draw_networkx(G, pos=pos,node_size=500, with_labels=True)
# edge_labels = nx.get_edge_attributes(G, "length")
# nx.draw_networkx_edge_labels(G, pos, edge_labels)

Nodes=G.nodes()


a=list(combinations(Nodes,2))
source=0
target=25

path=boxroute(G, source, target, N)
print("path", path)
Pxi = nx.get_node_attributes(G, "Px")
weights = nx.get_edge_attributes(G, "weight")
Pxs=probx(Pxi[source])
# print("Pxs=", Pxs)
sigma=0.0786
P1=prob(0,0)
# print(P0)

for ii in range(len(path)-1):

    if (path[ii], path[ii+1]) in weights:

        P2=prob(weights[(path[ii], path[ii+1])], sigma)
    else:
        P2=prob(weights[(path[ii+1]),path[ii] ], sigma)

    P1=np.dot(P1, P2)


    Pyx2=Pxs*P1
# print(P2)
Py2=Pym(Pyx2)
Iynax=Iyx(Py2)-Pxs[0]*(-P1[0][0]*np.log2(P1[0][0])-P1[0][1]*np.log2(P1[0][1]))+Pxs[1]*(-P1[1][0]*np.log2(P1[1][0])-P1[1][1]*np.log2(P1[1][1]))

print(Iynax)
```

```
    """
    # Marcell Nagy <marcessz@math.bme.hu>
    import networkx as nx
    import random


    def uv_flower(u, v, n):
        # Returns an a (u,v)-flower
        # TODO: átnézni
        graph = nx.cycle_graph(u + v)
        for i in range(n - 1):
            for e in list(graph.edges):
                graph.remove_edge(e[0], e[1])

                path_nodes = [e[0]]
                for x in range(u - 1):
                    path_nodes.append(graph.number_of_nodes())
                    graph.add_node(graph.number_of_nodes())
                    graph.add_edge(path_nodes[-2], path_nodes[-1])
                graph.add_edge(path_nodes[-1], e[1])

                path_nodes = [e[0]]
                for x in range(v - 1):
                    path_nodes.append(graph.number_of_nodes())
                    graph.add_node(graph.number_of_nodes())
                    graph.add_edge(path_nodes[-2], path_nodes[-1])
                graph.add_edge(path_nodes[-1], e[1])

        return graph


    def fractal_model(generation, m, x, e):
        """
        Returns the fractal model introduced by
        Song, Havlin, Makse in Nature Physics 2, 275.
        generation = number of generations
        m = number of offspring per edge end-point
        x = number of connections between offsprings
        e = probability that hubs stay connected
        1-e = probability that x offsprings connect.
        If e=1 we are in MODE 1 (pure small-world).
        If e=0 we are in MODE 2 (pure fractal).
        """
        graph = nx.Graph()
        graph.add_edge(0, 1)  # This is the seed for the network (generation 0)
        node_index = 2
        for n in range(1, generation+1):
            all_links = list(graph.edges())
            while all_links: # added m new nodes per link - sufficient to go by edges
                link = all_links.pop()

                new_nodes_a = range(node_index, node_index + m)
                node_index += m

                new_nodes_b = range(node_index, node_index + m)
                node_index += m

                graph.add_edges_from([(link[0], node) for node in new_nodes_a])
                graph.add_edges_from([(link[1], node) for node in new_nodes_b])

                # original version

    #             repulsive_links = list(zip(new_nodes_a, new_nodes_b)) # yields tuples
    #             graph.add_edges_from([repulsive_links.pop() for _ in range(x-1)])  # TODO: ezt meg kell nézni a cikkben

                if x>m: # this is needed in order to avoid triggering error by popping from empty list
                    repulsive_no=m
                    print('invalid arguments: x>m\n set x=m instead')
                else:
                    repulsive_no=x


                if random.random() > e:
                    repulsive_links = list(zip(new_nodes_a, new_nodes_b)) # yields tuples
                    graph.add_edges_from([repulsive_links.pop() for _ in range(repulsive_no)])  # TODO: ezt meg kell nézni a ci
                    graph.remove_edge(link[0], link[1])

        return graph
```

```python

def hub_attraction_dynamical_growth_model(generations, m, a, b, t_cutoff):
    """
    Returns a fractal graph generated by the model, introduced by Li Kuang et al. in:
    "A Fractal and Scale-free Model of Complex Networks with Hub Attraction Behaviors"

    This is a modification of the previous fractal model

    :param generations: number of iterations
    :param m: number of new offsprings per edge end-point
    :param a: connection probability of hubs
    :param b: connection probability of non-hubs
    :param t_cutoff: threshold separating hubs and non-hubs
    :return: graph
    """
    x = 1

    graph = nx.Graph()
    graph.add_edge(0, 1) # this is generation 0 so to speak
    node_index = 2

    t = 1

    while t <= generations:

        all_links = list(graph.edges())
        degrees = dict(graph.degree)
        k_max = max(degrees.values())  # The maximum degree in the graph at time t
        offsprings = {}  # dict: keys are nodes, values are set of new offsprings

        for edge in all_links:

            new_nodes_a = range(node_index, node_index + m)
            node_index += m

            new_nodes_b = range(node_index, node_index + m)
            node_index += m

            if edge[0] in offsprings:
                offsprings[edge[0]] = offsprings[edge[0]].union(set(new_nodes_a))
            else:
                offsprings[edge[0]] = set(new_nodes_a)

            if edge[1] in offsprings:
                offsprings[edge[1]] = offsprings[edge[1]].union(set(new_nodes_b))
            else:
                offsprings[edge[1]] = set(new_nodes_b)

            graph.add_edges_from([(edge[0], node) for node in new_nodes_a])
            graph.add_edges_from([(edge[1], node) for node in new_nodes_b])

            repulsive_links = list(zip(new_nodes_a, new_nodes_b))

            # degrees dictionary contains the degrees from the previous generation

            if degrees[edge[0]] / k_max > t_cutoff and degrees[edge[1]] / k_max > t_cutoff:
                e_prob = a
            else:
                e_prob = b

            if random.random() > e_prob:
                graph.remove_edge(edge[0], edge[1])
                graph.add_edges_from([repulsive_links.pop()])

        # TODO: max degree kellene a boxból

        within_box_edges = []

        for node, offspg in offsprings.items():
            rnd_node = random.choice(list(offspg))
            within_box_edges += [(rnd_node, n) for n in random.sample(offspg - {rnd_node}, degrees[node])]

        graph.add_edges_from(within_box_edges)

        t += 1

    return graph
```

```python

def hub_attraction_dynamical_growth_model(generations, m, a, b, t_cutoff):
    """
    Returns a fractal graph generated by the model, introduced by Li Kuang et al. in:
    "A Fractal and Scale-free Model of Complex Networks with Hub Attraction Behaviors"

    This is a modification of the previous fractal model

    :param generations: number of iterations
    :param m: number of new offsprings per edge end-point
    :param a: connection probability of hubs
    :param b: connection probability of non-hubs
    :param t_cutoff: threshold separating hubs and non-hubs
    :return: graph
    """
    x = 1

    graph = nx.Graph()
    graph.add_edge(0, 1) # this is generation 0 so to speak
    node_index = 2

    t = 1

    while t <= generations:

        all_links = list(graph.edges())
        degrees = dict(graph.degree)
        k_max = max(degrees.values())  # The maximum degree in the graph at time t
        offsprings = {}  # dict: keys are nodes, values are set of new offsprings

        for edge in all_links:

            new_nodes_a = range(node_index, node_index + m)
            node_index += m

            new_nodes_b = range(node_index, node_index + m)
            node_index += m

            if edge[0] in offsprings:
                offsprings[edge[0]] = offsprings[edge[0]].union(set(new_nodes_a))
            else:
                offsprings[edge[0]] = set(new_nodes_a)

            if edge[1] in offsprings:
                offsprings[edge[1]] = offsprings[edge[1]].union(set(new_nodes_b))
            else:
                offsprings[edge[1]] = set(new_nodes_b)

            graph.add_edges_from([(edge[0], node) for node in new_nodes_a])
            graph.add_edges_from([(edge[1], node) for node in new_nodes_b])

            repulsive_links = list(zip(new_nodes_a, new_nodes_b))

            # degrees dictionary contains the degrees from the previous generation

            if degrees[edge[0]] / k_max > t_cutoff and degrees[edge[1]] / k_max > t_cutoff:
                e_prob = a
            else:
                e_prob = b

            if random.random() > e_prob:
                graph.remove_edge(edge[0], edge[1])
                graph.add_edges_from([repulsive_links.pop()])

        # TODO: max degree kellene a boxból

        within_box_edges = []

        for node, offspg in offsprings.items():
            rnd_node = random.choice(list(offspg))
            within_box_edges += [(rnd_node, n) for n in random.sample(offspg - {rnd_node}, degrees[node])]

        graph.add_edges_from(within_box_edges)

        t += 1

    return graph
```

```python
# Создаем направленный граф
G = nx.DiGraph()

# Добавляем рёбра с взаимной информацией
edges = {
    ('n1', 'n2'): 0.9936,
    ('n2', 'n3'): 0.9886,
    ('n3', 'n4'): 0.963,
    ('n3', 'n5'): 0.9745,
    ('n5', 'n6'): 0.9745,
    ('n1', 'n4'): 0.9380,
    ('n4', 'n5'): 0.9697,
    ('n4', 'n6'): 0.9498
}

# Добавляем рёбра в граф
for (u, v), weight in edges.items():
    G.add_edge(u, v, mutual_info=weight)

# Функция построения таблицы маршрутизации
def build_routing_table(graph, source):
    table = {}
    for target in graph.nodes:
        if source == target:
            continue  # Исключаем самого себя

        if nx.has_path(graph, source, target):
            paths = list(nx.all_simple_paths(graph, source, target))
            best_path = max(paths, key=lambda path: sum(graph[path[i]][path[i+1]]['mutual_info'] for i in range(len(path)-1)))
            total_info = sum(graph[best_path[i]][best_path[i+1]]['mutual_info'] for i in range(len(best_path)-1))
            next_hop = best_path[1] if len(best_path) > 1 else None
        else:
            best_path, total_info, next_hop = [], 0, "N/A"

        table[target] = {
            'next_hop': next_hop,
            'total_info': total_info,
            'best_route': best_path
        }

    return table

# Формируем таблицы маршрутизации для каждого узла
routing_tables = {node: build_routing_table(G, node) for node in G.nodes}

# Вывод таблиц маршрутизации
for node, table in routing_tables.items():
    print(f"\nТаблица маршрутизации для узла {node}:")
    print("| Назначение | Следующий узел | Метрика (Σ I(X|Y)) | Лучший маршрут  |")
    print("|------------|----------------|--------------------|-----------------|")

    for target, data in table.items():
        print(f"| {target:<10} | {data['next_hop']:<14} | {data['total_info']:<18.4f} | {data['best_route']} |")
```