Санкт-Петербургский политехнический университет Петра Великого
Институт кибербезопасности и защиты информации

На правах рукописи

Овасапян Тигран Джаникович

# ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЕЙ С ИСПОЛЬЗОВАНИЕМ ОБУЧАЮЩИХСЯ АВТОМАТОВ

Направление подготовки: 10.06.01 Информационная безопасность

Направленность: 10.06.01_01 «Методы и системы защиты информации, информационная безопасность»

**НАУЧНО-КВАЛИФИКАЦИОННАЯ РАБОТА**

Автор работы: Овасапян Т.Д.
Научный руководитель:
Кандидат технических наук
Москвин Д.А.

Санкт-Петербург –2021

# Содержание

# ВВЕДЕНИЕ

В настоящее время стремительно растет тенденция использования устройств, способных взаимодействовать с внешней средой и обмениваться между собой информацией по внутренней сети или сети Интернет. Число подобных устройств непрерывно увеличивается, что свидетельствует о переходе к инфомационно-технологической концепции киберфизических систем (КФС). Значимая часть устройств киберфизических систем являются необслуживаемыми распределенными в пространстве вычислительными узлами. Объединение таких устройств в самоорганизующуюся систему сбора, обработки и передачи информации – беспроводную сенсорную сеть (БСС, Wireless Sensor Network, WSN) – позволит расширить возможности представления информации о производственных процессах, об окружающей среде, а также улучшить человеко-машинное взаимодействие.

Обеспечение безопасности играет решающую роль в повсеместном использовании WSN-сетей, поскольку данных класс сетей применяется в критических сферах, напрямую связанных с жизнедеятельностью общества. Необходимо также отметить, что из-за специфичных особенностей беспроводные сенсорные сети подвержены не только атакам злоумышленников, но и ошибкам в сети без каких-либо вмешательств извне. Это может произойти, если, например, один или несколько узлов потеряют часть своего функционала из-за ошибок в программном обеспечении или неблагоприятных условий внешней среды. Таким образом, развитие WSN-сетей вынуждает разрабатывать эффективные методы противодействия угрозам безопасности.

Использование множества различных методов защиты от конкретных атак в беспроводных сенсорных сетях далеко не всегда является целесообразным из-за ограниченности узлов в ресурсах. Некоторые из методов защиты не имеют на сегодняшний день ни одной практической реализации. Другие методы противоречат друг другу, например, по требуемым протоколам маршрутизации, что делает невозможным их одновременную работу в рамках одной сети. К тому же обновление функционала, для противодействия новым атакам, требует

физического доступа к некоторым узлам, что не всегда представляется возможным и увеличит стоимость обслуживания сети. В связи с этим необходим универсальный механизм безопасности, позволяющий защитить беспроводные сенсорные сети как от искусственных угроз безопасности, так и от естественных. В рамках данной работы предлагается использовать адаптивную систему управления на базе обучающегося автомата, позволяющей узлам изменять свое поведение в зависимости от внешних факторов.

Целью работы является обеспечение защищенности БСС от угроз безопасности путем использования адаптивного поведения узлов на основе обучающегося автомата.

Для достижения поставленной цели в работе решались следующие **задачи**:

1. Исследование особенностей функционирования беспроводных сенсорных сетей и выявление актуальных угроз безопасности.
2. Разработка методики проверки защищенности узлов беспроводной сенсорной сети на основе распределенного анализа показателей функционирования.
3. Построение конечно-автоматной модели, описывающей взаимодействие узлов и позволяющей узлам изменять принципы взаимодействия со своими соседями.
4. Создание метода обеспечения защиты на основе адаптивного поведения узлов с применением обучающегося автомата и анализа показателей поведения и функционирования узлов.
5. Экспериментальная оценка эффективности разработанного метода защиты путем моделирования его работы в системе симуляции беспроводной сенсорной сети.

**Научная новизна** полученных результатов:

1. Предложен подход к оценке безопасности узлов беспроводной сенсорной сети на основе анализа показателей функционирования.
2. Впервые предложено использование обучающегося автомата для изменения принципов взаимодействия узлов друг с другому в беспроводной сенсорной сети.
3. Разработана конечно-автоматная модель, позволяющая узлу изменять свое поведение по отношению к соседним узлам для обеспечения устойчивости функционирования беспроводной сенсорной сети.

4. Разработан алгоритм адаптивного выбора действия узла по отношению к соседнему узлу на основе обучающегося автомата и анализа показателей поведения и функционирования.

5. Создан имитационный стенд беспроводной сенсорной сети, позволяющий оценить зависимость количества вредоносных/неисправных узлов от показателя устойчивости функционирования сети.

Теоретическую значимость работы составляют предложенные показатели функционирования и поведения для оценки безопасности узлов БСС, а также представление принципов взаимодействия узлов БСС друг с другом в виде конечно-автоматной модели.

Практическая значимость результатов работы заключается в возможности применения предложенных методов и алгоритмов для реализации интеллектуально-адаптивной системы управления для защиты беспроводной сенсорной сети от угроз безопасности. Полученные в ходе работы результаты позволяют:

– выявить аномалии в работе беспроводной сенсорной сети путем анализа показателей функционирования устройств;

– представить в виде автоматной модели процесс взаимодействия узлов друг с другом в беспроводной сенсорной сети;

– путем имитационного моделирования работы сети в разработанном симуляторе определить граничные значения количества вредоносных и неисправных узлов для сохранения работоспособности сети.

Положения, выносимые на защиту:

1. Подход к оценке безопасности узлов беспроводной сенсорной сети на основе анализа показателей функционирования устройств.

2. Модель обучающегося автомата, обеспечивающая устойчивость функционирования беспроводной сенсорной сети путем адаптивного поведения узлов.

3. Алгоритм адаптивного выбора действия узла по отношению к соседнему узлу на основе обучающегося автомата и анализа показателей поведения и функционирования.

4. Полученные в результате имитационных испытаний оценки устойчивости функционирования беспроводной сенсорной сети в условиях работы с неисправными и вредоносными узлами.

Основные результаты исследований и научных разработок докладывались и обсуждались на следующих конференциях: научно-техническая конференция «Методы и технические средства обеспечения безопасности информации» (Санкт-Петербург, 2017, 2018, 2019 и 2020 гг.), научно-практическая конференция «РусКрипто» (Москва, 2020 г.), научно-практическая конференция с международным участием «Неделя науки СПбГПУ» (Санкт-Петербург, 2017 г.), международная конференция «Региональная информатика (РИ-2016)» (Санкт-Петербург, 2016 г.), межрегиональная конференция «Информационная безопасность регионов России (ИБРР-2017)» (Санкт-Петербург, 2017 г.), межрегиональная научно-практическая конференция «Перспективные направления развития отечественных информационных технологий» (Севастополь, 2018 г.), международная конференция «Fourth World Conference on Smart Trends in Systems, Security and Sustainability WorldS4» (Лондон (Великобритания), 2020 г.), международная конференция «SIN 2020: 13th International Conference on Security of Information and Networks» (Турция, 2020 г.).

Работа победила в конкурсе грантов Правительства Санкт-Петербурга для студентов вузов, расположенных на территории Санкт-Петербурга, аспирантов вузов, отраслевых и академических институтов, расположенных на территории Санкт-Петербурга, в 2017, 2018, 2019 и в 2020 годах.

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Беспроводная сенсорная сеть (БСС, WSN) - самоорганизующаяся сеть из множества датчиков и исполнительных устройств.

Модуль связи – аппаратное устройство, реализующее передачу информации по беспроводному каналу связи.

Узел сети (OBU) – программно-аппаратный комплекс, состоящий из модуля связи, ЭВМ и пакета программного обеспечения, способный посредствам модуля связи организовывать связи с другими узлами по заданному протоколу, формируя VANET.

Базовая дорожная станция (RSU) – стационарный узел VANET, расположенный в непосредственной близости к автомобильным дорогам.

Топология сети – это конфигурация графа, вершинами которого являются узлы сети, а рёбрами – информационные связи между вершинами.

Служебное сообщение – сообщение, передаваемое в компьютерной сети, предназначенное для поддержания корректного функционирования протоколов, применяемых в сети.

Информационное сообщение – сообщение, передаваемое в компьютерной сети, содержащее информацию прикладного уровня.

# 1 ИССЛЕДОВАНИЕ ПРИНЦИПОВ ФУНКЦИОНИРОВАНИЯ БСС И ПРИМЕНЯЕМЫХ В НИХ МЕХАНИЗМОВ БЕЗОПАСНОСТИ

В рамках данного раздела проводится исследование принципов функционирования и особенностей работы беспроводных сенсорных сетей, рассматриваются основные направления применения сетей, а также такие общие параметры как структуры, топологии, типы датчиков и протоколы.

Для того, чтобы исследовать работу БСС необходимо рассмотреть типы узлов, архитектуру и особенности построения сети. Важным этапом в исследовании работы БСС является определение угроз безопасности, а также существующих типов атак. В рамках данного раздела будет уделено внимание каждому из вышеперечисленных аспектов.

В дальнейшем эта информация будет базой для детального рассмотрения основных задач, решающих проблемы безопасности.

## 1.1 Принципы работы беспроводных сенсорных сетей

Беспроводные сенсорные сети (Wireless Sensor Networks или сокращенно WSN) — это беспроводные сети без инфраструктуры, развернутые в виде большого количества беспроводных датчиков (Node) определённым образом, которые используются для мониторинга некоторой системы или условий. WSN может быть либо мобильной беспроводной сенсорной сетью (MWSN), либо статической беспроводной сенсорной сетью (SWSN).

Каждый датчик по беспроводной сети передает информацию к базовой станции (Base Station), где данные сохраняются и могут быть проанализированы. Можно получать необходимую информацию из сети, производя запросы и собирая информацию с базовой станции. Стандартная беспроводная сенсорная сеть может содержать несколько тысяч сенсорных узлов. Узлы беспроводной сети датчиков оснащены измерительными, радиоприемниками и элементами энергопитания, иногда вычислительными устройствами. Узлы датчиков передают информацию по радиосвязи. Ресурсы беспроводной сенсорной сети по своему устройству ограничены в объеме памяти, скорости обработки данных и полосах волн

пропускания. При развертывании узлы датчиков часто должны проводить самоорганизацию в соответствии с сетевой инфраструктурой. После процесса развертывания сенсоры начинают собирать целевую информацию. Беспроводные сенсорные устройства реагируют на запросы, посылаемые с базовой станции, для выполнения специальных инструкций или предоставления образцов зондирования. Режим работы узлов датчиков может быть как постоянным с некоторым временным интервалом, так и управляемым с базовой станции. Для получения информации о местоположении можно использовать глобальную систему позиционирования (GPS) и алгоритмы локального позиционирования.

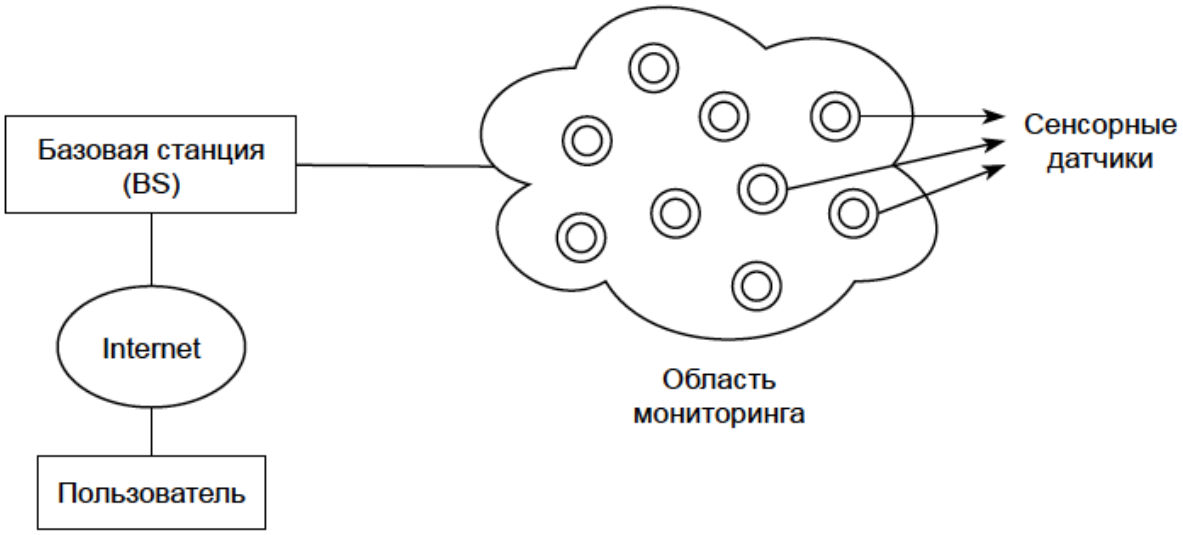



Рисунок 1 – Пример схемы беспроводной сенсорной сети

Беспроводные сенсорные сети (WSN) требуют нетрадиционных подходов для разработки протоколов из-за ряда ограничений. Простота устройства в сочетании с низким энергопотреблением (т. е. длительным сроком службы сети) требует нахождения правильного баланса между коммуникационными возможностями и возможностями обработки данных. В настоящее время большинство исследований WSN фокусируется на разработке энерго-эффективных и вычислительно-эффективных алгоритмов и протоколов, а прикладная область ограничивается базовыми приложениями для мониторинга и отчетности.

Таблица сравнение “области применения краткая статья.pdf”

## 1.2 Существующие решения для построения БСС и сферы их применения

Беспроводные сенсорные сети приобрели значительную популярность благодаря своей универсальности в решении многих задач в различных областях применения и имеют потенциал для изменения жизни общества. Области применения WSN-сетей можно классифицировать в соответствии с характером их использования на шесть основных категорий, к которым, как показано на рисунке 2, относятся: военная сфера, здравоохранение, окружающая среда, флора и фауна, промышленная и городская. Далее в этом подпункте объясняется характер каждой из этих категорий и подкатегорий. Кроме того, посредством ориентировочного рассмотрения характерных примеров, объясняются их особенности, а также указываются их преимущества и проблемы.

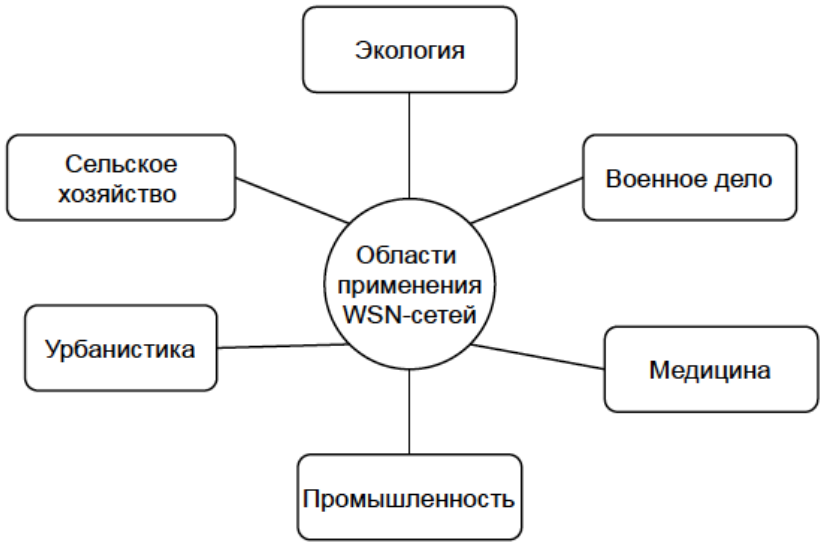



Рисунок 2 – Основные области применения WSN

Кроме того, обсуждаются различные методологии и технические средства, которые используются в каждой из сфер либо для зондирования, либо для обработки, и выявляются существующие между ними сходства и различия.

### *1.2.1 Решения для экологического и сельскохозяйственного применения*

Экология требует постоянного мониторинга условий окружающей среды, который может быть усовершенствован с использованием WSN. Основные категории экологических применений WSN:

— мониторинг воды (оценка качества свежей питьевой воды, мониторинг морской среды в целях предотвращения ущерба биосфере, контроль рыбных хозяйств по кормовым и фекальным отходам);

— мониторинг воздуха (в целях предотвращения опасных заболеваний и превышения нормы загрязняющих веществ и риска для здоровья людей);

— аварийное оповещение: сейсмической активности, вулканической активности, лесных пожаров и цунами (превентивный мониторинг причин стихийных бедствий может помочь избежать этих бедствий и/или снизить их ущерб).

— мониторинг тепличных хозяйств, сельскохозяйственных культур, мониторинг животноводства.

Т. Ахонен совместно с другими [1] представил беспроводной сенсорный узел, оснащённый радиоинтерфейсом ZigBee, для отслеживания состояния теплицы, но работает он в рамках протокола 6LoWPAN и стандартного сокета API, который позволяет передавать сжатые IP-пакеты версии 6 (IPv6) через сети IEEE 802.15.4. Работает в ISM диапазоне 2,4 ГГц и имеет пропускную способность 250 кбит/с. Дальность связи – до 10 метров, а потеря пакетов составляет приемлемые 5%.

Благодаря побудкам, длящимися 15 секунд, между периодами бездействия, длящимися 4 минуты и 15 секунд, были соблюдены требования по энергоэффективности архитектуры беспроводной сенсорной сети. Каждый сенсорный узел поочередно отправлял и получал пакеты по опросу готовности

координирующего узла. Время бездействия узла составляло 93.75%, которое могло быть увеличено до 97.50% путем сокращения времени работы с 15 до 5 секунд.

Г. Мендес [2] разработал умную БСС, основанную на соединении Wi-Fi, которая способна продуманно отслеживать состояние агрокультур по заранее заданному алгоритму. Она состоит из трех станций: сенсорная станция, точка доступа и узловая серверная станция.

В научных статьях сообщается о применении технологии РЧИД в различных сферах сельского хозяйства.

Токихиро Фукацу и Масаюки Хирафудзи разработали систему отслеживания с полевыми серверами данных, предназначенную для отслеживания информации по полям с возможностью удаленного управления в любом поле при помощи технологии Wi-Fi.

Иззатдин Абдул Азиз совместно с другими предлагает систему удаленного отслеживания температуры посредством беспроводного датчика и технологии GSM-SMS.

Б. Видья Сагар совместно с другими разработал вариант слежения за состоянием теплицы на базе технологии GSM- SMS, в котором фиксация данных о климатических условиях, обновляемых в режиме реального времени, и других параметрах окружающей среды, а также управление решениями осуществляется системой отслеживания (включающей в себя множество сенсорных узлов и блок управления). Решения могут корректироваться автоматизированной системой, которая также отправляет СМС о том, какая операция производится в данный момент.

А. Рахали совместно с другими [2] занимался дизайном и внедрением электронной системы, основанной на технологии GSM, для управления климатическими параметрами внутри теплицы через СМС сообщения. Основная задача концепции этой системы – удаленное управление климатическими параметрами, которые влияют на урожай в теплице. Несколько датчиков и исполнительных устройств установлены и подключены к блоку управления и блоку сбора данных. Эти датчики предоставляют необходимую информацию,

которая используется для управления вентиляцией, отоплением и насосом через СМС. Данная схема, используемая в нашей системе, позволяет владельцу вести контроль своего хозяйства удаленно без необходимости лично присутствовать на месте. Эта система идеально подходит для сельскохозяйственных теплиц в Марокко. При разработке графического интерфейса было использовано ПО LabVIEW для сбора, отслеживания через ПК и хранения всех данных посредством платы ввода-вывода PCL812PG.

Марко Манкузо совместно с другими [20] представил беспроводную сеть датчиков для отслеживания параметров окружающей среды в теплице с помидорами посредством узлов RTD204 на базе радиочастот IEEE 802.15.4 (которые являются радиочастотами широкополосной модуляции с прямым расширением спектра) с использованием модуляции со скачкообразной перестройкой частоты. Благодаря разнесению частот данная система превосходит ZigBee, у которой выходная мощность ограничена в Европе из-за нормативных ограничений.

### *1.2.2 Решения для применения в промышленности*

WSN могут применяться в различных промышленных целях для решения проблем широкого спектра. Основные подкатегории промышленных применений WSN: логистика, робототехника и машинное оборудование.

WSN подходят для мониторинга в режиме реального времени многих параметров окружающей среды и обеспечивают сбор точных данных, отвечающих требованиям Cold Chain Logistics. Исследователи использовали специальную сетевую модель Zigbee для построения структуры системы.

Применение WSN в логистике: с помощью технологии GPS система отслеживает в реальном времени состояние товара и имеет встроенный терминал, который определяет местонахождения товара, и облачную платформу сервисов, которая используется для идентификации получателя.

Применение WSN в робототехнике. В настоящее время существует множество приложений, объединяющих WSN и роботов. Роботы могут

нивелировать некоторые основные проблемы WSN – мобильность сенсорных узлов, передислокация узлов, обслуживание сети.

Разработаны роботизированные системы обслуживания сети, которые действуют по следующим принципам. Робот перемещается по сети в поисках дыр. Когда обнаруживается область, которая не покрывается датчиком, робот находит узел, который перекрывается другими узлами, и перемещает его в непокрытую область. Если робот обнаружит избыточные датчики, он переместит узлы, чтобы покрыть большую область. Также робот используется для передачи данных по сети между узлами, которые по разным причинам недоступны друг другу. Это так называемая проблема коммивояжера. Робот, перемещающийся между узлами, которые находятся вне радиуса действия беспроводной связи, позволяет сети быть объемной и при этом экономить энергию датчиков.

Применение WSN в мониторинге состояния оборудования. Целью мониторинга работоспособности оборудования является проверка работоспособности различных типов технического оборудования, а также выявление или прогнозирование возникновения неисправностей.

Для обнаружения утечек в системе нефтепроводов и газопроводов используются узлы беспроводных датчиков, которые контролируют давление и температуру трубопроводной жидкости в нескольких точках на трубопроводах, протянутых на большие географические территории.

Т. Ненов совместно с другими [3] представил сенсорный модуль для отслеживания качества воздуха. Модуль включает в себя датчик температуры, датчик влажности и датчик загрязнения воздуха. Данный модуль реализован на базе беспроводного интерфейса по стандарту IEEE 802.15.4 (ZigBee). На базе этого модуля была разработана беспроводная сенсорная сеть для отслеживания качества воздуха.

Гэньгун Ван [4] сообщает о технологиях WirelessHART и ISA-100.11, а также о вариантах их применения в промышленной области.

Дельфин Кристин совместно с другими испытывает и исследует технологии БСС, связанные с промышленной автоматизацией. Они определяют требования

качества обслуживания и проводят анализ возможных угроз, на который мы опираемся для оценки текущего уровня развития следующих технологий: WirelessHART, ISA100.11a и ZigBee.

Хеймердингер разработал беспроводной измерительный датчик (БИД) для отслеживания влажности древесины во время процесса сушки в текущий момент времени. БИД, установленные в разных местах древесины, оснащены радиопередатчиками с автономным питанием, которые отправляют данные на приемник главной станции посредством безлицензионных частотных ISM диапазонов (900 МГц в Северной Америке и 433 МГц в Европе). Эксперимент показал, что благодаря БИД точность и эффективность процессов сушки увеличилась, а затраты на систему сбора данных снизились.

В [5] БСС была разработана для проведения оценки энергопотребления и отслеживания состояния электрических машин. КПД электродвигателя и его состояние рассчитываются неинтрузивным образом посредством беспроводных узлов, которые отслеживают конечные параметры двигателя (т.е. линейный ток, фазный ток и температуру) без вмешательства в работу электромашин.

В [6] представлена система слежения за состоянием нефте- и газопроводов, основанная на БСС – REMONG. Чтобы определить наличие утечек, данная система использует беспроводные сенсорные узлы, которые отслеживают давление и температуру текучей среды в трубопроводе в нескольких ключевых точках трубопроводов, проложенных в пределах огромных территорий.

В [7] БСС разработана для того, чтобы улучшить безопасность промышленного оборудования, которое состоит из главного транспорта и присоединенного к нему прицепа. Трехосевой акселерометр и магнитометр, встроенные в устройство сенсорной системы, отслеживают параметры работы прицепа, а на блок обработки данных, который выполняет алгоритм контроля стабильности, беспроводным способом поступают соответствующие данные.

### *1.2.3 Решения для применения в здравоохранении*

В области здравоохранения WSN используют передовые медицинские сенсоры для мониторинга пациентов в пределах медицинского учреждения, а также

для обеспечения контроля в режиме реального времени жизненных показателей пациента с помощью носимых аппаратных средств. Основные подкатегории применения WSN:

— мониторинг, показателей с датчиков, носимых пациентами;

— системы домашней помощи, включают в себя определение целостной картины состояния здоровья, сбор данных пациента для узлов первичной и экстренной помощи, вплоть до связи и координации с назначенным центром помощи – больницей;

— больничное наблюдение за пациентами: системы WSN могут быть интегрированы, чтобы обеспечить постоянный мониторинг пациента и аварийное оповещение для более точного и быстрого реагирования.

Чжау-Шэнь Лин совместно с другими испытал мобильное устройство с передатчиком Bluetooth, на котором используется скачкообразное изменение частоты в диапазоне 2.4 ГГц. Скачок происходит в относительно быстром темпе, а максимальная физическая скорость передачи данных составляет 1 Мбит/с. Это эквивалентно 700 кбит/с фактической полезной скорости передачи данных. В основном данное устройство применятся в такой технологии здравоохранения, как мобильный блок для сбора физиологических сигналов.

Д. Вироне совместно с другими представил архитектуру системы умного здравоохранения на базе продвинутой БСС и ZigBee (MicaZ с сенсорной панелью MTS310). Она ориентирована главным образом на тех пациентов, которым требуется уход, и тем, кому может помочь постоянное и удаленное наблюдение за показателями здоровья.

Чон Ги совместно с другими представил нательные сенсорные платформы для слежения за физиологическими параметрами. В основе платформы лежит процессор TI MSP430, радиоинтерфейс CC2420 IEEE 802.15.4, трехосевой акселерометр и подзаряжаемый литий-полимерный аккумулятор. Помимо этого, данная платформа включает в себя слот для MicroSD с поддержкой флэш-памяти до 2 Гб.

Димитар Недев совместно с другими представил проект “Умный Кампус”, в котором датчики размещаются в коридорах и комнатах и собирают различные данные, например, уровень освещения и температуру воздуха посредством интерфейсов связи IEEE 802.15.4 6LoWPAN.

Во время исследования авторы испытывали датчики, работающие в режиме реального времени, для диагностики пациентов, страдающих от сердечных недугов, посредством смартфона и нательных датчиков. Авторы представили это приложение как приложение для обеспечения наблюдения за пациентами с сильными сердечными недугами, которые не могут ходить на прием к врачу на регулярной основе. Одна из главных причин, лежащих в основе данной технологии, – запуск таких устройств в серийное производство для помощи малообеспеченным пациентам. Более того, эта система разработана таким образом, чтобы при появлении критических показателей, сообщение об этом отправлялось лечащему врачу пациента.

БСС могут быть интегрированы в такие лечебные заведения, как, например, больницы для обеспечения наблюдения за состоянием пациентов в режиме реального времени и оповещения при появлении показателей критического характера, чтобы реакция на такие показатели была более точна и быстра. Типичный пример подобного рода применения представлен в. В частности, авторы описывают случай применения БСС, в котором беспроводные датчики были размещены в палатах больницы им. Джонса Хопкинса для отслеживания уровня кислорода в крови и частоты сердечных сокращений пациентов в режиме реального времени. Исследователи собрали статистику по производительности сети, и, несмотря на сложность больничной среды из-за помех и радиошумов, пришли к выводу, что применение БСС может улучшить работу лечебного заведения.

### *1.2.4 Решения для применения в военной сфере*

Военная сфера – не только первая сфера, в которой применялись БСС, также она считается сферой, которая послужила толчком для первых исследований в области сенсорных сетей. Умная пыль [8] – типичный пример таких первых исследовательских работ, которые проводились в конце 90-х для разработки

сенсорных узлов, способных, не смотря на свой очень малый размер, вести разведывательную деятельность.

Теперь, благодаря технологическим преимуществам, которых удалось достичь с того момента, БСС способны оказывать поддержку во время проведения военных операций [5]. На Рисунке 4 иллюстрированы варианты применения БСС (вместе с типами датчиков, которые наиболее часто применяются в них) в военных целях, а именно: для наблюдения в районе боевых действий, для наблюдения за ходом боевых действий, а также для обнаружения проникновения посторонних на территорию объекта.

Кроме того, подобные датчики могут использоваться для обнаружения химических, биологических, радиоактивных, ядерных и взрывоопасных веществ, а также для обнаружения вредных промышленных материалов. Для обнаружения проникновения к объекту БСС могут использовать инфракрасные (ИК), фотоэлектрические, лазерные, акустические и вибрационные датчики. Аналогичным образом в узлах БСС используются радиолокационные (РЛС), светолокационные (ЛИДАР), лазерно-локационные (ЛАДАР) и ультразвуковые датчики, чтобы определить расстояние до определенных объектов. Также ЛАДАР и ИК датчики используются для сканирования.

Помимо этого, гибкость структуры БСС позволяет подстраиваться им под различные требования. Например, во время тактических операций используются широкомасштабные БСС, состоящие из тысячи узлов, размещение которых происходит без физического участия. Во время ведения боевых действий в городских условиях и проведения операций по защите своих войск, используются БСС, состоящие из сотни размещаемых вручную узлов [5].

В [6] рассматривается вопрос применения технологии БСС для ведения наземного наблюдения. В частности, авторы предлагают систему, которая состоит из дешевых общих узлов, способных улавливать магнитные и акустические излучения, которые производятся различными движущимися целями. Данная система направлена на обнаружение и классификацию различных целей, таких как

движущийся транспорт и отряд, на основе пространственной разности мощности сигнала, улавливаемого датчиками.

В [7] представлена система обнаружения подводных лодок для противолодочной обороны (ПЛО). Эта система состоит из недорогих многопозиционных сенсорных блоков, совмещающих в себе как активные, так и пассивные гидролокаторы. Указанная система может быть масштабирована до большого количества датчиков, размещенных в прибрежных водах в зависимости от конкретных условий и глубины океана. Упомянутые выше сенсорные блоки используют встроенные пассивные гидролокаторы для обнаружения дизель-электрических подводных лодок, а активные гидролокаторы – для подтверждения целей. Блок, подтвердивший цель, уведомляет об этом соседние сенсорные блоки посредством сигнала оповещения, в котором содержится код ID.

В [8] представлено использование группы акустических датчиков, подвешенных к низу привязных аэростатов, для обнаружения и определения координат движущихся транспортных средств, для фиксации импульсных сигналов от минометов, огневых средств артиллерии и ружейно-пулеметного огня, а также для определения местоположения их источников.

Данная система обнаружения может быть использована в связке с акустическим векторным датчиком, чтобы увеличить вероятность определения местоположения угрозы при помощи ударной волны, которую создает сверхзвуковая пуля, и дульной волны, создаваемой орудием при выстреле [9].

### *1.2.5 Решения для применения в городских условиях*

Разнообразие возможностей, предлагаемых WSN, также дает возможность получить беспрецедентный уровень информатизации о целевой области, будь то комната, здание или наружная территория. Сети WSN являются инструментом для измерения пространственно-временных характеристик любых явлений в городской среде, предлагая неограниченное число возможностей. Наиболее популярные приложения WSN в городской среде связаны с "умными домами", "умными городами" и транспортными системами.

В больших городах для поддержания оптимальной жизни горожан важно следить за различными параметрами. У БСС имеется огромное количество вариантов применения для передачи данных в режиме реального времени властями в целях обеспечения оптимального функционирования города. Например, возросший транспортный поток создает проблемы и является причиной потери времени, когда большое количество автомобильного транспорта движется по загруженным дорогам. БСС могут применяться для отслеживания ситуации на дорогах в целях уменьшения транспортного потока, обозначения парковочных мест и т.д. [46].

В [47] представлена умная система парковки для автомобилей, основанная на БСС. В частности, рассматривается ее применение в пределах парковочной зоны для автомобилей: каждое парковочное место оснащается одним недорогим сенсорным узлом, который определяет свободно или занято данное место. На центральной станции происходит сбор данных по статусу всех парковочных мест, которые затем отправляются в виде периодических отчетов в базу данных. Доступ к базе данных имеется у системы управления верхнего звена для осуществления нескольких задач, таких как поиск свободных парковочных мест, контроль безопасности, автоматический сбор платы или/и создание статистических отчетов.

В [48] представлена система, в основе которой лежит БСС, для измерения и классификации дорожного движения. Узлы этой системы со встроенными магнитными датчиками размещаются вдоль дороги для осуществления наблюдения за дорожным движением в режиме реального времени. Данная конфигурация была представлена как простая и дешевая для внедрения альтернатива индукционным катушкам, которые, как правило, размещаются на дорогах для выполнения основных задач по управлению движением. Эта архитектура, используемая в представленной системе, иллюстрирована на Рис. 17.

В [49] представлена система для контроля дорожного движения возле затопленных туннелей. Указанная система состоит из БСС и централизованной системы управления. Аналоговые выходные сигналы датчиков, расположенных внутри двух подземных туннелей, конвертируются в цифровые, которые затем

передаются в точку доступа через последовательные Ethernet преобразователи. По той же схеме сигналы доходят до программируемого логического контроллера, который отслеживает уровень воды внутри двух подземных туннелей и соответствующим образом регулирует водоотлив затопленных туннелей при помощи насосов, а также движение автомобильного транспорта, приближающегося ко въезду в эти туннели.

В [55] представлена система, которая анализирует параметры транспорта на дороге посредством телематической БСС. В системе используются размещенные на дороге сенсорные узлы и размещенный на обочине узел, который получает отправляемые данные. Сенсорные узлы фиксируют проезжающий автотранспорт и передают данные на узел-получатель данных. Далее, когда происходит событие, создается пакет, а узел-получатель данных ставит временную отметку на этом пакете, и отправляет его на узел связи, который использует статический протокол маршрутизации. После этого узел связи направляет этот пакет пользователю, который анализирует полученную информацию для определения различных параметров дороги, например, скорость автотранспорта, интенсивность движения автотранспорта на полосе и т.д.

В [59] описывается дизайн, разработка и размещение стадионной структурной платформы на базе БСС, которая находится в США, для отслеживания показателей здоровья. Система собирает данные в режиме реально времени во время спортивных и других крупных событий для сверки структурного поведения стадиона в корреляции с поведением зрителей.

## 1.3 Особенности работы и угрозы безопасности

Сенсорные сети, помимо основных проблем безопасности, унаследованных от ad-hoc сетей, имеют также и уникальные проблемы, из-за присущих им особенностей [2]. Из особенностей, напрямую влияющих на безопасность, можно выделить следующие:

1. Ограниченность узлов в ресурсах

Ресурсы узлов включают в себя память для программного кода и данных, вычислительные ресурсы контроллера, а также энергетические ресурсы элемента питания. Для создания механизмов безопасности необходимо ограничить размер программного кода, реализующего алгоритм обеспечения безопасности. Другим важным ограничением возможностей узлов является энергопотребление. Данный аспект влияет на время функционирования всей сети, поскольку время работы одного отдельного узла влияет на работу всей сети в целом.

2. Необходимость использования беспроводных каналов связи

В кабельных сетях, для подключения к сети, необходимо иметь доступ к физической составляющей среды коммуникации, а также пройти защитные механизмы (firewall, VPN и т. п.). В БСС среда передачи беспроводная, для доступа к сети, необходимо лишь быть в зоне её действия.

Специфика беспроводной связи, необходимой для работы WSN-сетей, приводит к повреждению пакетов с данными или потере пакетов сильно нагруженными узлами. На качество передачи данных также влияет окружающая среда (деревья, стены зданий и т.п.). Большинство протоколов WSN-сети используют широковещательную передачу, которая может привести к конфликтам в процессе ретрансляции пакетов.

3. Работа в общедоступных местах

В большинстве случаев узлы беспроводной сенсорной сети (БСС) работают на неконтролируемом периметре. Узлы БСС могут находиться внутри неконтролируемого периметра, из-за чего появляется возможность захвата или подмены узла. Злоумышленник может легко получить физический доступ к компонентам БСС и реализовать любую атаку путем перепрограммирования одного или нескольких узлов.

4. Отсутствие технического обслуживания в течение длительного времени

В связи с областью применения WSN-сетей, узлы сети находятся в общедоступных местах без технического обслуживания в течение длительного периода времени. Это делает сеть незащищенной от злоумышленника, который

легко может получить узел в распоряжение. Также на работу сети могут повлиять различные стихийные бедствия, такие как пожары наводнения и т.д.

5. Отсутствие инфраструктуры и полная децентрализация сети

Стоимость WSN-сетей, которые функционируют на неконтролируемом периметре, должна быть очень низкой, что не подразумевает использования дополнительной инфраструктуры в сети. Сами узлы выполняют функции датчиков, ретрансляторов и маршрутизаторов. В децентрализованной большой сети очень трудно идентифицировать вредоносную атаку. При этом становится проблематичным использование таких систем безопасности, как центры сертификации и центральные серверы.

## 1.4 Угрозы безопасности и актуальные атаки

Реализуя угрозы безопасности, злоумышленник может совершить определенную атаку [3]. В таблице 1 представлены наиболее актуальные атаки на WSN-сети.

Уязвимость беспроводных сенсорных сетей к атакам обусловлена в первую очередь присущим им характеристиками. Поскольку данные передаются по беспроводной связи, злоумышленнику довольно просто перехватить трафик. Чтобы соответствовать строгим бюджетным требованиям, сенсорные узлы, как правило, не защищены от взлома. Поэтому системы безопасности необходимы для обеспечения безопасности сети и защиты от угроз безопасности. Для обеспечения безопасности обычно используются системы на основе криптографии. Однако из-за недостатка памяти и мощности (низкая вычислительная мощность, ограниченные запасы энергии) сенсорных узлов эти способы защиты могут приводить к краху или быстрому износу системы. Поэтому обеспечение безопасности с учетом специфических ограничений датчиков и строения сети представляет собой одну из наиболее важных исследовательских проблем в беспроводных сенсорных сетях. Далее будут рассмотрены наиболее известные атаки и методы защиты против них. Атаки можно классифицировать в соответствии с уровнями в модели OSI. В соответствии с вторым методом

классификации атаки можно разделить на внутренние или внешние, а также пассивные или активные [11].

### *1.4.1 Атаки на физическом уровне*

Физический уровень – нижний уровень модели OSI, который описывает способы передачи данных через физические связи, определяющий различные параметры как такие как частота, обнаружение сигнала, модулирование и шифрование данных. Как и в любой среде, в WSN существует возможность глушения. Кроме того, узлы в WSN могут быть развернуты во враждебной или небезопасной среде, где злоумышленник имеет легкий физический доступ. Эти две уязвимости рассматриваются в данном подразделе.

Глушение — это тип атаки, при котором производятся помехи в радиочастотах, используемые узлами сети. Глушение источника может быть довольно мощным, и привести к нарушению работы сети целиком, или менее сильным и сможет нарушить работу небольшой части сети. Средства защиты от глушения включают в себя способы связи с использованием спектра распространения. Например, метод с расширенным или частотно-скачкообразным распространением спектра (FHSS), при котором сигнал передается, быстро переключаясь между несколькими частотными каналами с использованием псевдослучайного перемежения, которое известно передатчику и приемнику. Так как злоумышленник не может следить за изменяющейся частотой, то у него не получится заглушить частоту, по которой передаются данные в конкретный момент. В таком случае злоумышленник может вместо глушения отдельных частот заглушить широкий спектр частотного диапазона ввиду того, что диапазон используемых частот сети ограничен. Для гарантии низкой стоимости и обеспечения условиям требований к энергопотреблению, сенсорные устройства ограничиваются использованием одной частоты и по этой причине сильно подвержены атакам глушения [12].

Подмена

При возможности физического доступа к узлу, злоумышленник может извлечь конфиденциальные данные, например, криптографические данные, метод

аутентификации и другую информацию сети. Одним из способов обороны от этой атаки является защита от несанкционированного физического доступа к узлу. Как правило получается, что сенсорные сети в WSN не защищаются от фальсификации узлов ввиду экономической неэффективности процесса. Можно сделать вывод, что схема безопасности должна брать в расчет вероятную возможность компрометации сенсорных узлов.

### *1.4.2 Атаки на канальном уровне*

Канальный уровень – второй уровень модели OSI, который отвечает за образование нескольких отдельных потоков данных, контроль ошибок, передачу данных в своем сегменте локальной сети, доступ к среде. Он обеспечивает надежные соединения «один к одному» и «один ко многим» в коммуникационной сети. Атаки на канальном уровне включают целенаправленно создаваемые коллизии, исчерпание ресурсов [13].

Коллизии

Коллизии происходят, когда два узла пытаются одновременно передать сигнал на одной и той же частоте. Когда пакеты сталкиваются, в данных может произойти изменение, что вызовет несоответствие контрольной суммы на принимающей стороне. Пакет будет отброшен как недействительный. Противник может стратегически вызвать коллизии в определенных пакетах, например, в управляющих сообщениях ACK. Результатом атак коллизий является задержка с экспоненциально растущим временем ожидания в протоколах управления доступа к среде (MAC). Один из способов защиты от коллизий – это использование контроля ошибок. Лучше всего обнаружение работает в случае малого количества коллизий, в частности вызванных аномалиями вследствие изменений окружающей среды или вероятностными ошибками. С другой стороны, этот способ обнаружения добавляет дополнительные затраты на передачу и обработку данных. Можно логично предположить, что злоумышленник может испортить больше, чем можно исправить. Хотя злонамеренные коллизии можно обнаружить, на данный момент не существует полноценной защиты от них.

Исчерпание ресурсов

Повторяющиеся коллизии также могут быть использованы злоумышленником, чтобы вызвать исчерпание ресурсов. Например, примитивная реализация канального уровня может пытаться повторно передавать поврежденные пакеты. Если эти повторные передачи не будут обнаружены и предотвращены, запасы энергии передающего узла и окружающих его узлов будут быстро исчерпаны. Возможным решением является применение ограничения скорости в управлении допуском MAC, при котором сеть будет игнорировать чрезмерные запросы, таким образом уменьшая утечку энергии из-за повторных передач [14]. Второе возможное решение – это использование образование нескольких отдельных потоков данных с временным разделением, когда каждому узлу выделяется персональный таймслот для передачи данных.

### *1.4.3 Атаки на сетевом уровне*

Сетевой уровень в сенсорных сетях как правило создается по следующим принципам:

– энергоэффективность является первостепенным фактором;

– сенсорные сети в основном ориентированы на данные;

– идеальная сенсорная сеть имеет адресацию на основе атрибутов и осведомленность о местоположении;

– атаки в сети и на уровне маршрутизации включают следующие.

Подделка, изменение или воспроизведение информации о маршрутизации – это атака на обмен маршрутной информации между узлами. Вероятными последствиями являются появление петель маршрутизации, увеличение исходных сетевых маршрутов, притягивание или сброс сетевого трафика для выбранных узлов, появление фальшивых сообщений об ошибках и увеличение времени задержки между конечными пунктами. Контрмерой против подделки и изменения является добавление кода аутентификации сообщения (MAC) после сообщения. Получатель может проверить, не были ли сообщения подделаны или изменены, если будет добавлен MAC код к сообщению. Также счетчики или временные метки

могут быть включены в сообщения для защиты от повторной передачи информации [15].

Выборочная пересылка

Злоумышленник может реплицировать вредоносные узлы, которые будут избирательно пересылать конкретные сообщения и отбрасывают остальные. Атака «черная дыра» является одной из форм атаки выборочной пересылки, при которой узел отбрасывает все получаемые сообщения. В качестве защиты от атак селективной пересылки можно предложить использовать несколько маршрутов отправки данных. И как альтернативный способ защиты – при обнаружение вредоносного узла или узла, который мог выйти из строя обращаться к поиску другого маршрута.

Атака Воронки (Sinkhole)

При атаке sinkhole злоумышленник делает скомпрометированный узел более привлекательным для окружающих узлов, подделывая информацию о маршрутизации. В результате окружающие узлы выбирают захваченный узел в качестве следующего узла для пересылки данных.

Атака Сивиллы (Sybil) — это атака, при которой один узел выдает себя в качестве нескольких идентичностей. К протоколам и алгоритмам, которые легко подвержены атаке, относятся схемы отказоустойчивости, распределенного хранения данных и поддержания топологии сети.

Wormhole атаки (червоточины) – это соединение атакующего с низкой задержкой между двумя частями сети, через которую злоумышленник пересылает сообщения. Такая связь может устанавливаться одним узлом, пересылающим сообщения между двумя соседними, или парой узлов в разных частях сети. Существует два типа отметок для обнаружения и защиты от атак Wormhole: географические и временные.

Атаки переполнения (Hello Flood) – многие протоколы, использующие пакеты HELLO, делают неверное предположение, что получение такого пакета означает, что отправитель находится в радиусе связи и, следовательно, является соседом. Злоумышленник может использовать мощный передатчик, чтобы

обмануть большую область узлов и заставить их поверить, что они являются соседями узла злоумышленника. Если атакующий ложно передает лучший маршрут к базовой станции, все эти узлы попытаются передать сообщение атакующему узлу.

IP-спуфинг (Acknowledgment Spoofing) – алгоритмы маршрутизации, используемые в сенсорных сетях, иногда требуют использования номеров последовательности и подтверждения (sequence number, acknowledgement number). Атакующий узел может подделать номера SEQ и ACK подслушанных пакетов, предназначенных для соседних узлов, передавая фальшивую информацию другим узлам. Часто в качестве подмены злоумышленник утверждает, что узел работоспособный, хотя на самом деле он уже не работает [16].

### *1.4.4 Атаки на транспортном уровне*

Транспортный уровень отвечает за доставку данных. Здесь будут рассмотрены две возможные атаки на этом уровне – флудинг и десинхронизация.

Флуд атаки – протоколу требуется поддерживать состояние на обоих концах соединения, он становится уязвимым к истощению памяти в результате лавинной маршрутизации. Злоумышленник может много раз подряд запрашивать соединение, пока ресурсы, необходимые для каждого нового установления соединения, не будут исчерпаны и сеть не перестанет работать. Последующие присылаемые легитимные запросы будут отброшены. Для решения этой проблемы можно запрашивать у каждого подключающегося клиента подтверждать свою готовность к соединению, решив головоломку. Учитывая, что атакующий, скорее всего, не обладает бесконечными ресурсами, он не сможет создавать новые соединения достаточно быстро, чтобы вызвать голодание ресурсов на обслуживающем узле.

Десинхронизация – это нарушение существующего соединения. Например, злоумышленник может подделывать сообщения на конечном узле, заставляя этот узел запрашивать повторную передачу пропущенных кадров. При правильном выборе времени атакующий может ухудшить или даже предотвратить способность конечных узлов успешно обмениваться данными, заставляя их вместо этого

тратить энергию на попытки восстановления после ошибок, которых на самом деле не было. Для решения этой проблемы можно требовать проводить аутентификацию всех пакетов, передаваемых между узлами. При условии, что метод аутентификации сам по себе безопасен, злоумышленник не сможет отправлять поддельные сообщения конечным узлам [18].

Таблица 1 – Актуальные атаки на WSN-сети

| **Атака** | **Нарушаемое свойство безопасности** | **Описание атаки** |
|---|---|---|
| Атака выброса пакетов | Доступность | Атака заключается в отбрасывании вредоносным узлом всех или только некоторых пакетов, проходящих через него |
| Атака модификации пакетов | Целостность | Вредоносный узел вносит изменения в пакеты, которые проходят через него |
| Атака засорения канала | Доступность | Атакующий узел генерирует помехи в канале передачи данных посторонними шумами |
| Атака поглощения энергии | Доступность | Атакующий узел заставляет соседние узлы обрабатывать специально сформированные пакеты, что в свою очередь приводит к быстрому расходу ресурсов батареи |
| Атака червоточины | Доступность, конфиденциальность | Атакующий перехватывает пакеты от одного узла сети и посредством созданного тоннеля передает их на другой узел, тем самым влияя на структуру сети |
| Атака черная дыра | Доступность, конфиденциальность, целостность | Атака на протокол маршрутизации, вынуждающая все соседние узлы в сети отправлять пакеты с данными атакующему узлу |
| Атака Сибиллы | Доступность, конфиденциальность | Атака, в ходе которой узел злоумышленника эмулирует работу нескольких узлов сразу, тем самым внося изменения в топологию сети |

Можно сделать вывод, что большинство атак влияют на доступность сети. Конфиденциальность является важным свойством безопасности, но в контексте рассматриваемых сетей, ее нарушение не влечет за собой критических последствий. Таким образом, необходимы средства, позволяющие не только обнаружить атаки и предотвратить их, но и сохранить устойчивость сети даже при совершении атак.

## 1.5 Анализ специфичных атак

Как отмечалось ранее, в данной дипломной работе будет акцентироваться внимание на программные атаки, так как процесс их обнаружения является наиболее трудоемким, а их воздействие на устройство может быть критичным. В качестве примеров таких атак выбрано несколько типичных для сетей Интернета вещей способов вредоносного воздействия на устройства и сеть в целом.

В качестве примеров для экспериментов в данной работе рассматриваются следующие типовые для устройств Интернета вещей атаки:

— атака истощения энергоресурсов;

— атака зашумления канала связи;

— атака Black Hole.

Данные атаки были выбраны для исследования в рамках данной дипломной работы, так как являются одними из наиболее распространенных и опасных для сетей Интернета вещей [28-31].

Атаку истощения энергоресурсов оказывается довольно сложно обнаружить [28]. Стандартные методы, такие как статический анализ приложения или динамический анализ системных вызовов, оказываются неэффективными при попытке обнаружения вредоносного кода, который потребляет много ресурсов.

Атака зашумления канала связи позволяет нарушить не только работу сети, в которой находится зараженный узел, но и также превратить устройство в бот для ботнета, что является одной из самых больших современных пробелм сетей Интернета вещей [30].

Атака Black Hole также является одной из наиболее распространенных и опасных атак для WSN-сетей [31]. Ее также бывает сложно обнаружить стандартными методами. Применение интеллектуальных методов может быть эффективным для борьбы с атаками такого типа.

### *1.5.1 Атака истощения энергоресурсов*

Цель данной атаки заключается в том, чтобы за короткий промежуток времени исчерпать энергоресурс устройства (например, аккумуляторную батарею), тем самым выведя его из строя, либо замедлить его работу, нарушая нормальное функционирование.

Достигается такой эффект несколькими способами. Первый способ – Denial-of-Sleep. Он заключается в том, что устройство принудительно выводится из режима сна (режима низкого энергопотребления), который часто используется на устройствах Интернета вещей. Вторая атака основана на несанкционированном увеличении объема сетевого трафика, который получает и отправляет устройство, что также увеличивает затраты энергетических ресурсов. Третий способ организации ататки истощения энергоресурсов – это создание электромагнитных помех на беспроводных каналах, вынуждая тем самым устройства Интернета вещей генерировать сигнал большей мощности, чем изначально было рассчитано. И, наконец, несанкционированное использование вредоносного ПО, которое может производить множественный запуск приложений, принудительно включать GPS, Bluetooth, а также другие датчики и интерфейсы, которые сильно влияют на энергозатраты устройства.

Последний способ отличается тем, что помимо истощения энергоресурсов устройства, злоумышленник также может повлиять на его нормальную работу, сильно ограничивая вычислительные способности.

Исследования показывают, что даже относительно слабая атака позволяет повысить потребление энергии устройством Интернета вещей примерно в 4.5 раза, а значит сильно снизить его автономность [32].

### *1.5.2 Атака зашумления канала связи*

Как известно, главная особенность WSB устройств в том, что они могут быть объединены в сети разных размеров, где различные устройства взаимодействуют между собой, за счет чего и происходит функционирование такой сети. Цель атаки

– нарушить коммуникацию между устройствами Интернета вещей с помощью зашумления каналов передачи между устройствами.

Для успешного проведения атаки, злоумышленник должен установить на устройстве ПО, которое будет генерировать пакеты и отправлять их по каналам связи с другими устройствами в сети. Это не только создаст помехи при отправки важных пакетов и в целом замедлит работу всей сети, но также в целом увеличит объем трафика, который получают и отправляют другие устройства. Также это, как уже было рассмотрено в предыдущем пункте, приведет к ускорению истощения энергетических ресурсов.

### *1.5.3 Атака Black Hole*

Атака «Black Hole (черная дыра)» или атака выброса пакетов – это атака, при которой узел сети, который в нормальном состоянии должен ретранслировать или обрабатывать приходящие пакеты вместо этого отбрасывает их [33].

Обычно, атакуемый узел посылает информацию другим устройствам в сети о том, что он находится ближе всего к главному серверу, за счет чего меняется карта маршрутизации и данные начинают пересылаться через зараженный узел. Злоумышленник делает так, чтобы устройство отбрасывало весь проходящий через него сетевой трафик, поэтому главный сервер перестает получать информацию от других узлов сети. Таким образом, всего нескольких зараженных устройств в сети Интернета вещей достаточно, чтобы нарушить корректную работу большого количества хостов в сети.

Существует два основных подхода к организации атаки выброса пакетов.

Первый способ заключается в том, что злоумышленник внедряет собственный узел в сеть Интернета вещей, с помощью которого в дальнейшем организуется атака. При подобной атаке достаточно легко обнаружить нелегального участника сети стандартными методами и предотвратить атаку.

Гораздо более опасный подход, когда осуществляется компрометация одного или нескольких узлов сети, которые уже учувствуют в передаче данных. В таком случае, при правильном подходе атаку становится очень сложно обнаружить.

## 1.6 Обзор подходов к обеспечению безопасности БСС

Существует множество подходов к обеспечению безопасности распределенных самоорганизующихся сетей [4].

Далее будут рассмотрены основные подходы.

### *1.6.1 Защищенное управление группой*

Данный метод предполагает разбиение всей сети на группы. В каждой группе выделяется один узел (центральный), который анализирует состояние остальных узлов в своей группе. Обмен данными между узлами осуществляется через центральные узлы каждой группы [5]. Недостатком данного подхода является дополнительное требование к вычислительным ресурсам центрального узла каждой группы. Кроме того, внедрение в защищенную группу скомпрометированного узла может принести куда больший вред, чем обычное деструктивное воздействие такого скомпрометированного узла в сети.

### *1.6.2 Безопасное агрегирование данных*

Передача данных существенно влияет на энергетические ресурсы узлов. Для оптимизации объема передаваемых данных и повышения работы сети используется агрегация данных [6]. Для этого вводятся специальные узлы (точки агрегации), целью которых является агрегация данных. При этом точки агрегации данных могут быть подвержены разным видам атак и, как следствие, требуют надежной защиты, поскольку через скомпрометированные узлы могут внедряться ложные данные, которые в будущем гарантированно приведут к ошибкам в агрегированных расчетах. В рамках безопасной агрегации используется множество техник, такие как язык запросов, кластеризация узлов,

### *1.6.3 Использование криптографических методов*

Под данным подходом подразумевается использование в сети шифрования данных, системы управления ключами, а также аутентификации сетевых ресурсов [7]. На сегодняшний день существует ряд работ [8], которые выполняют криптографические функции даже в устройствах с ограниченными вычислительными ресурсами. Однако в большинстве задач, для которых используются WSN-сети, конфиденциальность не играет важную роль. При этом

недостатком метода являются проблемы с масштабируемостью, и меньшая длительность работы узлов, из-за повышенного требования к ресурсам источника энергии.

Выбор наиболее подходящего криптографического метода является крайне важным в WSN, поскольку все службы безопасности гарантируются стойкостью криптографии. Криптографические методы, которые применяются в WSN, должны соответствовать условиям эксплуатации сенсорных узлов и оцениваться по энергопотреблению, размеру передаваемых данных и времени обработки. В этом разделе будет рассмотрена ассиметричная криптография и симметричная криптография.

**Криптография с открытым ключом**

Исследователи считают, что размер кода, объем данных, время обработки и энергопотребление делают нежелательным применение в WSN таких методов алгоритмов с открытым ключом, как протокол согласования ключей Диффи-Хеллмана или подписи RSA. Для ассиметричных алгоритмов (RSA) требуются значительные вычислительные затраты и часто выполняются тысячи инструкций умножения для выполнения одной операции шифрования. Эффективность ассиметричного алгоритма в микропроцессоре определяется количеством тактов, необходимых для выполнения инструкции умножения. [19] Было рассчитано, что для ассиметричных алгоритмов необходимо от нескольких секунд до минут для выполнения шифрования и дешифрования на беспроводных узлах, что делает вероятной уязвимость к DoS-атакам. Алгоритмы криптографии с симметричным ключом и хэш-функции потребляют гораздо меньше вычислительной энергии, чем алгоритмы с открытым ключом.

Последние исследования показали, что применение криптографии с открытым ключом в сенсорных сетях возможно при правильном выборе алгоритмов и соответствующих параметров, оптимизации и использовании маломощных методов. Исследованные алгоритмы с открытым ключом включают схему Рабина, NtruEncrypt, RSA и криптографию эллиптических кривых (ECC). Большинство исследований в литературе посвящено алгоритмам RSA и ECC.

Привлекательность алгоритма на эллиптических кривых заключается в том, что он обеспечивает ту же степень защиты, но при гораздо меньшем размере ключа, таким образом снижаются дополнительные расходы на передачу и обработку данных [20].

**Симметричная криптография**

Ограничения на вычисления и энергопотребление в сенсорных узлах ограничивают применение криптографии с открытым ключом в WSNs. Поэтому большинство исследований посвящено симметричной ключевой криптографии в сенсорных сетях. Пять популярных схем шифрования, RC4, RC5, IDEA, SHA-1 и MD5, были оценены на шести различных микропроцессорах с размером слова 8, 16 и 32 бит. Эксперименты показали примерно одинаковые криптографические затраты для каждого класса шифрования и каждого класса архитектуры.

Производительность симметричной криптографии определяется следующими факторами.

1. Размер шины данных: большинство встроенных процессоров обычно используют шину данных шириной от 8 до 16 бит, а алгоритмы шифрования часто используют 32-битную шину.

2. Набор инструкций: архитектура набора инструкций (ISA) влияет на выбор алгоритма. Большая часть встроенных процессоров не поддерживает инструкцию поворота переменных битов влево (ROL), которая значительно повышает производительность RC5. В WSN предпочтительна симметричная криптография из-за ограничений сенсорных узлов.

**Протоколы управления ключами**

Управление ключами является одним из основных механизмов для обеспечения безопасности в WSN. Цель управления ключами заключается в создании необходимых ключей между сенсорными узлами, которые должны обмениваться данными. Кроме того, схема управления ключами должна поддерживать добавление и аннулирование узлов при работе в неопределенных условиях развертывания. Из-за ограничений, накладываемых на сенсорные узлы,

схемы управления ключами в WSN имеют много отличий от схем в сетях ad-hoc. Из-за большого объема требуемой памяти парные схемы не могут быть использованы при большом размере сети. Далее будут рассмотрены протоколы управления ключами, основанные на структуре сети.

Структура сети, лежащая в основе протокола, играет важную роль в работе протоколов управления ключами. В соответствии со структурой протоколы можно разделить на две категории: схемы с централизованными ключами и схемы с распределенными ключами. В централизованной схеме базовая станция рассматривается как KDC (центр распределения ключей) и корень дерева, а все ключи логически распределены в дереве. Однако при наличии только одного управляющего субъекта центральный сервер является единой точкой отказа. Вся сеть и ее безопасность будут затронуты, если возникнет проблема с контроллером.

В подходах распределенного управления ключами для управления генерацией, регенерацией и распределением ключей используются различные контроллеры, что минимизирует риск сбоя и обеспечивает лучшую масштабируемость. При таком подходе большее количество объектов может выйти из строя, прежде чем пострадает вся сеть. Большинство предложенных схем управления ключами являются распределенными схемами. Эти схемы также делятся на детерминированные и вероятностные, которые рассматриваются в следующем подразделе.

На примере протокола шифрования и аутентификации LEAP (Localized Encryption and Authentication Protocol) рассмотрим четыре основных типа ключей для сенсорных узлов:

– индивидуальный ключ, совместно используемый с базовой станцией (заранее распределенный);

– глобальный ключ, совместно используется всеми узлами в сети (заранее распределенный);

– парный ключ, разделяемый с ближайшими соседними узлами;

– кластерный ключ, разделяемый с несколькими соседними узлами.

Парные ключи, разделяемые с ближайшими соседними узлами, используются для защиты одно ранговой связи, в то время как кластерный ключ используется для локальной трансляции.

### *1.6.4 Системы обнаружения вторжений*

Беспроводные сенсорные сети могут быть подвержены различным вторжениям злоумышленников. Для систем идентификации вторжений (СОВ) основной задачей является мониторинг самой сенсорной сети, проведение идентификации попыток проникновения, а также рассылка узлам уведомлений об этом. Для повышения эффективности работы такие системы могут включать в себя интеллектуальные методы анализа [9]. Существенным недостатком системы являются повышенные требование к вычислительным и энергетическим ресурсам узлов сети, если будет использовать распределенная СОВ. Использование централизованного средства обнаружения вторжений, которое требует выделенного сервера, далеко не всегда возможно из-за отсутствия необходимой инфраструктуры.

Все механизмы безопасности, рассмотренные выше, применяются для предотвращения проникновения в сеть вредоносных узлов. Однако одних этих механизмов бывает недостаточно для обеспечения оптимальной безопасности сети. Чтобы гарантировать высокий уровень безопасности, необходима вторая линия защиты. Эта вторая контрмера представляет собой систему обнаружения вторжений (СОВ), которая занимается обнаружением и предотвращением вредоносных проникновений. СОВ можно классифицировать как: системы, основанные на обнаружении сигнатур и системы с поведенческими аномалиями [22]. Третья категория СОВ может быть выведена из системы поведенческих аномалий под названием системы, основанные на спецификации. Эта категория использует тот же принцип, что и обнаружение аномалий. Однако определение модели поведения осуществляется вручную, а не автоматически с помощью алгоритма обучения. Это упрощает систему обнаружения и значительно снижает

процент ложноотрицательных обнаружений. Этот тип СОВ является наиболее подходящим для ограничений сенсорных сетей.

В результате работы проделанной во втором разделе данной работы были рассмотрены задачи и проблемы безопасности, актуальные для WSN-сетей, затем перечислены распространенные атаки, которым подвержены WSN-сети и соответствующие меры противодействия. На основе изученной информации были определены основные механизмы защиты для беспроводных сенсорных сетей, включающих криптографические, превентивные и физические методы защиты.

### *1.6.5 Распределенный контроль поведения узлов*

Для обеспечения эффективной защиты БСС от атак со стороны скомпрометированных узлов часто применяют механизмы контроля, а именно системы репутации и доверия. Они основаны на расчете величины взаимного доверия узлов друг к другу в процессе мониторинга действий отдельного узла в сети [10]. Разница в подходах к определению уровня доверия может возникнуть из-за особенностей той среды, в которой осуществляется взаимодействие узлов. В имеющихся моделях доверия существуют разные трактовки таких понятий, как репутация и доверие, а также рассматривают разные объекты и субъекты доверия. Этот метод защиты является перспективным, поскольку его применение не требует от узлов затрат больших ресурсов и при этом способен защитить от многочисленных атак разного вида.

Нельзя не отметить, что из-за специфичных особенностей БСС подвержены не только атакам злоумышленников, но и возникновению ошибок в сети без каких-либо вредоносных вмешательств извне. Такое может произойти, если, например, один или несколько узлов потеряют часть своего функционала из-за ошибок в программном обеспечении или неблагоприятных условий внешней среды. Большинство существующих систем безопасности, функционирующих на основе анализа поведения и вычисления уровня доверия, могут идентифицировать такие узлы, как вредоносные и исключить их из сети. Это в свою очередь приведет к

потере отказоустойчивости сети в целом, что является критичным для одноранговых сетей.

## 1.7 Выводы

В рамках раздела выявлены характерные особенности WSN-сетей и влияние этих особенностей на безопасность сети. Из-за ряда особенностей существующие методы обеспечения безопасности являются недостаточно эффективными для сенсорных сетей, работающих длительное время без обслуживания на неконтролируемом периметре. Результаты, полученные в ходе исследования в рамках данного раздела лягут в основу выбора подходов к обеспечению безопасности беспроводных сенсорных сетей.

# 2 МЕТОД ОЦЕНКИ БЕЗОПАСНОСТИ УЗЛОВ НА ОСНОВЕ АНАЛИЗА ПОКАЗАТЕЛЕЙ ФУНКЦИОНИРОВАНИЯ

Сравнивая преимущества и недостатки различных способов обеспечения безопасности устройств беспроводных сенсорных сетей, можно прийти к выводу о том, что наиболее предпочтительными являются динамические методы обнаружения атак в случае, когда анализ выполняется удаленно.

Метод выявления и защиты от атак, который предлагается в рамках данного раздела, заключается в динамическом анализе получаемых с устройства данных о его состоянии. На основе интеллектуального анализа полученных показателей можно выявить узлы атакующего.

Идея одного из таких подходов состоит в следующем: устройства БСС отправляют данные о своих различных показателях на узел сети, на котором запущен алгоритм, основанный на методах машинного обучения, способный определить, заражено ли устройство и, в некоторых случаях, тип заражения с определенной точностью [37].

## 2.1 Выбор показателей функционирования устройств

Важным аспектом в предлагаемом методе выявления и защиты от атак является набор показателей функционирования, на основе которых будет проводиться интеллектуальный анализ состояния узлов.

Одна из ключевых особенностей устройств WSN-сети заключается в том, что они не взаимодействуют и не контролируются напрямую человеком.

Таким образом, показатели, которые можно использовать для определения атакующего устройства в WSN-сетях, представлены в таблице 3.1.

Таблица 3.1 – Используемые показатели для анализа

| Категория | Показатели | Атака от которой защищаемся |
|---|---|---|

| | | |
|---|---|---|
| Сеть | Принятые/отправленные байты, принятые/отправленные пакеты | Сетевые атаки |
| CPU | Загрузка процессора | Атаки истощения ресурсов. Некоторые сетевые атаки, требующие вычислительных мощностей |
| Питание | Уровень заряда, температура, потребление | Атаки истощения ресурсов |
| Процессы | Количество процессов | Атаки истощения ресурсов |
| Память | Общий размер, Shared, Allocated, PSS, VSS, Свободная | Сетевые атаки |

Важно отметить, что использование показателей ресурсов системы имеет существенное преимущество по сравнению с другими методами определения вредоносного ПО. Дело в том, что поменять вызовы функций или в целом поведение программы проще, чем изменить модель использования электроэнергии или других ресурсов устройства. Кроме того, модель использования ресурсов, как правило, сохраняется от устройства к устройству для одной и той же атаки или ВПО.

Дальнейшее исследование будет связанно со способами динамического исследования показателей устройств БСС для выявления атак с применением методов машинного обучения.

## 2.2 Применение методов машинного обучения для выявления атак на устройства

Методы машинного обучения хорошо зарекомендовали себя в задачах классификации. В данном разделе будет описан подход к анализу показателей

работы устройств с использованием методов машинного обучения. Также будут рассмотрены результаты существующих исследований в этом направлении.

Показатели функционирования устройства можно рассматривать в виде временного ряда. Существует два основных подхода в определении метрики расстояния между такими рядами:

— классическое евклидово расстояние (евклидова метрика) – расстояние между двумя векторами в евклидовом пространстве, рассчитывается по формуле:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}.$$

— расстояние из алгоритма динамической трансформации временной шкалы (Dynamic Time Wrapping, DTW).

Использование евклидового расстояния имеет существенный недостаток – если один из двух одинаковых сравниваемых временных рядов незначительно смещен относительного другого, то, согласно евклидовой метрике, такие ряды будут разными. Алгоритм DTW помогает преодолеть этот недостаток и измерять расстояние между временными рядами, не обращая внимания на сдвиги по времени [38].

Далее будет рассмотрен принцип работы алгоритма динамической трансформации временной шкалы. Две последовательности дискретных величин можно обозначить как два временных ряда – $X = (x_1, \dots, x_n)$ и $Y = (y_1, \dots, y_m)$ длины $n$ и $m \in \mathbb{N}$ соответсвенно. Далее, строится матрица расстояний $C \in \mathbb{R}^{n \times m}$. Каждый элемент матрицы $c_{ij}$ показывает расстояние между элементами $x_i$ и $y_i$.

Целью алгоритма DTW является поиск оптимального выравнивания между элементами $X$ и $Y$ с минимальным общим расстоянием между рядами. Для этого строится матрица деформаций (трансформаций). Каждый элемент такой матрицы вычисляется следующим образом:

$$d_{i\,j} = c_{i\,j} + \min(d_{i-1\,j}, d_{i-1\,j-1}, d_{i\,j-1}).$$

После этого находится путь деформации $p = \{p_1, \dots, p_L\}, p_l = (n_l, m_l) \in [1{:}N] \times [1{:}M], l \in [1{:}L]$. Путь деформации удовлетворяет следящим условиям:

— граничные условия: начало пути $p_1 = (1,1)$, его конец – $p_L = (N, M)$. Это ограничение гарантирует, что путь деформации содержит все точки обоих временных рядов;

— монотонность: $n_1 \leq n_2 \leq \ldots \leq n_L$ и $m_1 \leq m_2 \leq \cdots \leq m_L$;

— размер шага: $p_{l+1} - p_l = \{(1,0), (0,1), (1,1)\}$ для $l \in [1: L]$.

Суммирование локальных расстояний из таблицы деформаций дает общую стоимость пути или DTW-расстояние. Путь с наименьшей стоимостью и есть оптимальный путь деформации.

Для измерения сходства рядов $X$ и $Y$ или, другими словами, расстояния между ними, оценивается их общая стоимость оптимального пути деформации.

Итак, общая стоимость пути деформации между временными рядами $X$ и $Y$ находится по формуле:

$$c_p(X, Y) = \sum_{i=1}^{L} c(x_{nl}, y_{ml}).$$

DTW-расстояние между временными рядами $X$ и $Y$:

$$DTW(X, Y) = c_{p^*}(X, Y) = \min\{c_p(X, Y) \mid p = (N, M)\}$$

Для оценки качества классификации вводится понятие матрицы ошибок. Это способ разбить объекты на четыре категории в зависимости от комбинации истинного ответа и ответа алгоритма. Через элементы таблицы 3.2 можно, например, выразить долю правильных ответов:

Таблица 3.2 – Описание категорий возможного вывода классификатора

| | y = 1 | y = –1 |
|---|---|---|
| a(x) = 1 | Верные положительные<br>True Positive (TP) | Ошибки 2-го рода<br>False Positive (FP) |
| a(x) = –1 | Ошибки 1-го рода<br>False Negative (FN) | Верные отрицательные<br>True Negative (TN) |

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN}.$$

Также выделяют критерии точность (precision) и полнота (recall):

$$precision = \frac{TP}{TP + FP};$$

$$recall = \frac{TP}{TP + FN}.$$

Точность показывает, какая доля объектов, выделенных классификатором как положительные, действительно является положительными. Полнота показывает, какая часть положительных объектов была выделена классификатором.

Существует несколько способов получить один критерий качества на основе точности и полноты. В данной работе будет использоваться F-мера – гармоническое среднее точности и полноты:

$$F = \frac{2 * precision * recall}{precision + recall}.$$

Перед использованием в различных методах классификаций, показатели делятся на подвыборки. В качестве показателей могут быть, например, данные энергопотребление процессов, которые были считаны с интервалом в 500 мс.

Для разделения выборок используются интервалы (окна) фиксированного размера. Интервал $w$ является временным значением. Например, $w = 6$ означает, что подвыборка содержит значения 6 интервалов записей, полученных в ходе измерений.

Обучение проходит по множествам подвыборок. Для этого полученные выборки разбиваются на заданные интервалы – подвыборки, и с каждой такой подвыборкой ассоциируется метка класса.

Алгоритм подготовки набора данных представлен на рисунке 3.1.

**Input:** Sample set $S = \{Sample_1, Sample_2, \ldots, Sample_n\}$
& window size $w$
**Output:** Subsamples set DB
$DB \leftarrow \{\}$
**for** $i \leftarrow 1$ *to* $n$ **do**
  $l \leftarrow 1$
  **while** $(l + w) < length(sample_i)$ **do**
    $Append \quad \langle Label_i, P_l, ., P_{l+w} \rangle \quad to \quad DB$
    $l \leftarrow l + 1$
  **end**
**end**
**return** $DB$

Рисунок 3.1 – Алгоритм подготовки набора данных.

После этого классификатор обучается на полученной выборке.

При классификации, на вход алгоритму также подается выборка, которая разбивается на интервалы-подвыборки. Классификатор получает на вход интервал и возвращает спрогнозированную метку класса. Метка присваивается выборке путем простого голосования – выборка соотносится с классом, которому принадлежит большая часть ее подвыборок.

**Input:** Subsample database $DB\&Sample$
**Output:** $Label\epsilon\{R, G\}$
$Labels \leftarrow \{\}$
$Classifier = TrainClassifier(DB)$
$Subsamples = Grind\ Sample\ Using\ Algorithm\ 1$
$i \leftarrow 1$
**while** $i < size(Subsamples)$ **do**
  $L \leftarrow Classifier(Subsample_i)$
  $Append\ L\ to\ Labels$
  $i \leftarrow i + 1$
**end**
**return** $most\ frequent\ item\ in\ Labels$

Рисунок 3.2 – Алгоритм подготовки набора данных.

Размер окна подбирался в диапазоне от 5 до 50 с шагом 5.

Для примера, в качестве показателя функционирования будет взят параметр, показывающий энергопотребление устройства. Для других показателей применение методов машинного обучения будет аналогичным.

В работе [50] авторы использовали для выявления известного вредоносного программного обеспечения для ОС Android. Выявить ВПО удалось в среднем с вероятностью 0.9.

В работе [51] использовался параметр, показывающий потребление устройством на ОС Android электроэнергии. Для каждого алгоритма, обучение и валидация проходили на всем диапазоне. В алгоритмах использовалось два варианта метрики: евклидово расстояние и DTW-расстояния. В результате работы в лучшем случае с вероятностью 0.76 удалось выявить вредоносное программное обеспечение, реализующее атаку. Данный результат был получен при использовании DTW-расстояния и алгоритма KNN.

Также существует ряд других работ, демонстрирующих . Таким образом, применение методов машинного обучения является эффективным подходом.

В результате обучения и валидации удалось получить следующую точность для различных алгоритмов, представленную в таблице 3.3.

Таблица 3.3 – Результаты обучения модели, использующей информацию о потреблении электроэнергии устройством

| | **Accuracy (%)** | **Recall (%)** | **Precision (%)** | **F-Measure (%)** |
|---|---|---|---|---|
| KNN (K = 1) | 71.85 | 71.11 | 56.14 | 62.75 |
| KNN (K = 5) | 72.59 | 72.22 | 57.02 | 63.73 |
| KNN (K = 10) | 72.22 | 71.11 | 56.64 | 63.05 |
| **KNN (K = 1 и DTW)** | **83.70** | **78.89** | **73.96** | **76.34** |
| Neural Network | 75.93 | 73.33 | 61.68 | 67.01 |

| | | | | |
|---|---|---|---|---|
| Random forest | 80.74 | 76.67 | 69.00 | 72.63 |
| SVM | 78.52 | 74.44 | 65.69 | 69.79 |

## 2.3 Создание выборок для тестирования

Выборки для тестирования будут собираться с эмулируемых устройств WSN-сети, которые совершают определенную атаку. В качестве атак выбрано несколько типичных для БСС способов вредоносного воздействия на устройства и сеть в целом.

В качестве примеров для экспериментов в данной работе рассматриваются следующие типовые для устройств БСС атаки:

— атака истощения энергоресурсов;

— атака зашумления канала связи;

— атака Black Hole.

Данные атаки были выбраны для исследования в рамках работы, так как являются одними из наиболее распространенных и опасных для сетей Интернета вещей [28-31].

Атаку истощения энергоресурсов оказывается довольно сложно обнаружить [28]. Стандартные методы, такие как статический анализ приложения или динамический анализ системных вызовов, оказываются неэффективными при попытке обнаружения вредоносного кода, который потребляет много ресурсов.

Перечень функционала, который активировался на устройстве для реализации атаки истощения энергии:

— CPU Load – искусственно загружает процессор устройства, выполняя операции в цикле;

— GPU Load – работает аналогично CPU Load, но загружает графический процессор;

— GPS – принудительно активирует GPS модуль;

— WiFi – принудительно оставляет включенным WiFi модуль;

— Bluetooth – принудительно оставляет включенным Bluetooth модуль.

Атака зашумления канала связи позволяет нарушить не только работу сети, в которой находится зараженный узел, но и также превратить устройство в бот для ботнета, что является одной из самых больших современных проблем сетей Интернета вещей [30].

Для эмуляции данной атаки в Nox Player устанавливается программа Packets Generator. Она позволяет отправлять большое количество в фоне на указанный IP адрес. Графический интерфейс программы представлен на рисунке ниже.

Во время эмуляции атаки производилась отправка большого числа пакетов с устройства в течение 30 минут.

Атака Black Hole также является одной из наиболее распространенных и опасных атак для сетей Интернета вещей [31]. Ее также бывает сложно обнаружить стандартными методами и поэтому применение интеллектуального анализа показателей устройства может быть эффективным для борьбы с атаками такого типа.

Для эмуляции данной атаки используется программа «Blackhole» для ОС Android, которая позволяет отбрасывать любые исходящие пакеты на устройстве. Таким образом эмулируется невозможность дальнейшей передачи информации в сети Интернета вещей.

В процессе эмуляции каждой из перечисленных выше атак в течение 30 минут проводилось сохранение всех выделенных ранее параметров функционирования.

## 2.4 Сбор и подготовка тестовых данных

Измерения показателей устройства в каждом из экспериментов проводились в течение 30 минут с интервалом в 500 мс.

Полученные ранее данные представлены различных форматах. Для того чтобы использовать показатели, полученные в ходе эмуляции атак на устройства Интернета вещей, в различных моделях машинного обучения, требуется совершить предобработку этих данных.

Для каждого эксперимента по эмуляции атаки удалось получить следующие данные от устройства:

— суммарная загрузка процессора (в процентах);

— суммарное использование оперативной памяти;

— количество доступной в системе оперативной памяти;

— количество свободной оперативной памяти;

— кэш.

Фрагмент данных, полученных в программе AnotherMonitor представлен на рисунке 4.1.

| | Total CPU usage (%) | Memory used (kB) | Memory available (kB) | MemFree (kB) | Cached (kB) |
|---|---|---|---|---|---|
| 0 | 3.000000 | 426680 | 2684152 | 2555620 | 128532 |
| 1 | 4.784689 | 424916 | 2685916 | 2557388 | 128528 |
| 2 | 2.512563 | 424592 | 2686240 | 2557596 | 128644 |
| 3 | 2.955665 | 424420 | 2686412 | 2557768 | 128644 |
| 4 | 1.515151 | 424344 | 2686488 | 2557844 | 128644 |
| 5 | 2.955665 | 424352 | 2686480 | 2557836 | 128644 |
| 6 | 2.000000 | 424468 | 2686364 | 2557720 | 128644 |
| 7 | 0.505050 | 424644 | 2686188 | 2557544 | 128644 |
| 8 | 1.015228 | 424468 | 2686364 | 2557720 | 128644 |
| 9 | 0.000000 | 424620 | 2686212 | 2557568 | 128644 |
| 10 | 1.522843 | 424344 | 2686488 | 2557844 | 128644 |

Рисунок 4.1 – Данные, полученные в программе AnotherMonitor

При этом количество доступной в системе памяти, как оказалось, складывается из количества свободной памяти и размера кэша, поэтому данный признак исключается из конечной выборки.

Для сетевых интерфейсов устройства удалось получить следующие параметры:

— интерфейс отправки пакета;
— интерфейс приема пакета;
— IP адрес отправки;
— порт отправки;
— IP адрес получателя;
— порт получателя;
— размер пакета.

Названия интерфейсов, IP адресы, а также порты не несут никакой полезной информации, кроме возможности различать входящий и исходящий сетевой трафик. По этой причине, в конечной выборке были объединены данные по всем входящим и исходящим пакетам соответственно.

Помимо этого, устройство также отправляет информацию о расходе электроэнергии. Выходные данные программы PowerTutor представлены в виде простых текстовых файлов, а не файлов формата «csv», который представляет данные сразу в табличном виде. Поэтому данные о потребляемой энергии сначала следует отфильтровать из полученного лог-файла. Для этого показателя также доступно количество энергии, которое было потрачено на подсветку экрана. Это значение также следует отфильтровать и вычесть из общего расхода, так как, как правило, устройства Интернета вещей не обладают собственным экраном.

Также потребовалось синхронизировать временные метки. Приложения, которые собирают различные данные, были настроены так, чтобы обновлять информацию об интерфейсах и показателях с интервалом 500 мс, но при этом могут быть запущены в разное время. Поэтому записи были синхронизированы по времени последнего запущенного приложения для снятия показателей, что позволило получить наиболее точную и полную выборку.

После подготовки и синхронизации данных, полученные таблицы были собраны в один большой набор данных, который содержит все зафиксированные

показатели устройства и метки классов, обозначающие атаку, которая была симулирована. На рисунке 4.2 представлен фрагмент полученной в итоге таблицы.

| Total CPU usage (%) | Memory used (kB) | Memory available (kB) | MemFree (kB) | Cached (kB) | Out Length | In Length | CPU Power Usage | Class |
|---|---|---|---|---|---|---|---|---|
| -0.795390 | 0.099825 | 1.251255 | -0.425317 | 0.936501 | -0.491033 | -0.076786 | -0.691810 | 2 |
| -0.760427 | 1.261268 | 0.808397 | -1.247406 | 1.036530 | 0.470928 | -0.076786 | -0.510075 | 3 |
| -0.795390 | -0.996752 | -0.943859 | 1.018134 | -1.015785 | -0.483708 | 0.022848 | -0.669093 | 1 |
| -0.795390 | 1.232118 | 0.819512 | -1.232308 | 1.036475 | 0.120559 | -0.076786 | -0.413528 | 3 |
| -0.783502 | -0.063514 | 1.313536 | -0.340763 | 0.936213 | -0.488684 | -0.076786 | -0.686131 | 2 |
| -0.795390 | -0.887558 | -0.985495 | 0.974078 | -1.021124 | -0.477419 | 0.122482 | -0.686131 | 1 |
| -0.783620 | 0.253287 | 1.192740 | -0.425008 | 0.901392 | -0.491033 | -0.076786 | -0.691810 | 2 |
| 1.570450 | -0.941462 | -0.964941 | 0.965725 | -1.005135 | -0.491033 | -0.076786 | 1.659391 | 0 |
| 0.320319 | 0.927121 | 0.935807 | -1.061993 | 1.030422 | 2.005091 | -0.076786 | -0.691810 | 3 |
| -0.783620 | -0.959350 | -0.958120 | 0.996941 | -1.014907 | -0.464324 | 0.222116 | -0.527112 | 1 |
| -0.771849 | -0.899061 | -0.981108 | 0.955763 | -1.010378 | -0.491033 | -0.076786 | -0.510075 | 1 |

Рисунок 4.1 – Фрагмент таблицы обработанных данных

Дальнейшая задача заключается в том, чтобы обучить различные модели по аналогии с методом, в котором применялось только энергопотребление устройств, но теперь с использованием всех доступных показателей устройства, которые удалось получить во время эмуляции различных атак.

## 2.5 Результаты применяемых классификаторов Применяемые классификаторы для обработки данных

Рассмотрим подробнее некоторые из методов, которые можно применять для обучения систем обнаружения атак на устройства Интернета вещей.

В качестве классификаторов для построения моделей и тестирования в рамках данной дипломной работы были выбраны три алгоритма:

— алгоритм k ближайших соседей (KNN);

— метод опорных векторов (SVM);

— случайный лес (RF);

Алгоритм k ближайших соседей (KNN) – это простой и мощный классификатор, который ищет K ближайших к объекту элементов в выборке, и

присваивает ему метку класса большинства соседей, основываясь на теории компактности [38].

Метод опорных векторов (SVM) – в основе метода лежит поиск разделяющей гиперплоскости с максимальным отступом от векторов признаков двух классов – опорных векторов. Использование различных ядер позволяет строить нелинейные разделители [39].

Случайный лес (RF) – в основе метода лежит построение множества решающих деревьев во время обучения и генерирования метки класса. Процесс классификации основан на голосовании входящих в это множество деревьев [40].

Метрическими называются методы обучения, основанные на анализе сходства объектов, то есть на предположении, что схожим объектам очень часто соответствуют схожие ответы. Алгоритм k ближайших соседей (KNN) является метрическим классификатором, поэтому для него необходимо задать функцию расстояния. В рамках исследования было использовано описанное ранее классическое евклидово расстояние.

Обработка данных происходит при помощи библиотек Pandas и NumPy для языка программирования Python. Реализации классификаторов для обучения моделей взяты из библиотеки Scikit-learn.

1. Алгоритм k ближайших соседей (KNN)

Если подробнее рассмотреть данный алгоритм классификации, оказывается, что его работа сводится к запоминанию отправленной на вход информации во время обучения, которая в дальнейшем используется при отнесении объекта к тому или иному классу. После прохождения процесса обучения, классы атак, к которым относятся объекты, определяются на основе заданной функции расстояния.

Зачастую, к недостаткам данного алгоритма относят отсутствие настраиваемых параметров и то, что он очень сильно зависит от выбора метрики. Тем не менее, исходя из результатов рассмотренных ранее исследований, эта особенность классификатора не ухудшает результаты. Наоборот, использование метрического классификатора позволило добиться наибольшей точности и полноты классификации при анализе потребления электроэнергии. Это дает

основание предположить, что использование расширенных данных позволит улучшить результаты обнаружения.

2. Метод опорных векторов (SVM)

Применение метода опорных векторов предполагает линейную разделимость объектов выборки, то есть возможность с помощью единственной гиперплоскости однозначно отделить один класс от другого.

3. Случайный лес (RF)

Метод случайного леса (Random Forest) показал не лучшие результаты в исследовании, проведенном с использованием данных о энергопотреблении. Тем не менее, данный классификатор может хорошо обрабатывать полученные в ходе работы данные, так как число признаков увеличилось в несколько раз по сравнению с предыдущими работами. Кроме того, RF может помочь в оценке значимости признаков.

Обучение проходило на полученных ранее данных, которые были случайным образом перемешаны и разбиты на обучающую выборку и проверочную выборку таким образом, что 75% всех данных использовались для обучения, а остальные 25% - для тестирования. Также для финальной проверки были собраны дополнительные данные с другого устройства.

Экспериментальным путем было обнаружено, что наилучших результатов метод k ближайших соседей показывает наилучший результат при значении параметра k = 10. Результаты обучения классификатора на полученных показателях устройства представлены в таблице 4.1.

Таблица 4.1 – Результаты классификации методом kNN

| Accuracy (%) | Полнота (Recall) (%) | Точность (precision) (%) | F-мера (%) |
|---|---|---|---|
| 99.8273 | 99.8368 | 99.8307 | 99.8335 |

Также были проведены тесты метода опорных векторов. В качестве параметра C для линейного ядра экспериментальным путем было выбрано число $10^{-2}$. Результаты тестирования метода SVM представлены в таблице 4.2.

Формула для алгоритма классификации kNN:

$$w(i, u) = [i \leq k];$$

$$a(u; X^{\ell}, k) = \arg\max_{y \in Y} \sum_{i=1}^{k} [y_u^{(i)} = y].$$

— $X^{\ell}$ – выборка

— $u$ – объект

— $k$ – количество соседей

— $Y$ – множество классов

— $y$ – класс объекта

Таблица 4.2 – Результаты классификации методом SVM

| Accuracy (%) | Полнота (Recall) (%) | Точность (precision) (%) | F-мера (%) |
|---|---|---|---|
| 97.2366 | 97.2603 | 97.4790 | 97.2233 |

Затем был протестирован метод случайного леса. Размер леса, который является параметром алгоритма, выбран также экспериментальным методом – 42. Результаты проверки работоспособности метода представлены в таблице 4.3.

Таблица 4.3 – Результаты классификации методом RF

| Accuracy (%) | Полнота (Recall) (%) | Точность (precision) (%) | F-мера (%) |
|---|---|---|---|
| 97.4093 | 97.5220 | 97.6611 | 97.4696 |

Таким образом, по результатам можно сделать вывод о том, что использование нескольких параметров устройства Интернета вещей улучшает

точность и полноту выявления типовых атак. Наилучший результат достигается при использовании алгоритма k ближайших соседей (kNN) с параметром k = 10. Также можно обратить внимание, что метод случайного леса также показал хорошие результаты по сравнению с результатами при использовании всего одного признака.

## 2.6 Выводы

В рамках работы, описанной в данном разделе, разработан метод определения атак на устройства беспроводной сенсорной сети с помощью сбора и последующего анализа различных показателей устройств.

Выдвинуто предположение о том, что в качестве данных для динамического анализа могут быть использованы различные показатели устройства, которые можно получить из операционной системы. Для проверки работоспособности этого метода были воспроизведены некоторые из известных атак на беспроводные сенсорные сети и собраны необходимые данные о состоянии устройств во время атак. После подготовки и обработки, полученные показатели были использованы в нескольких алгоритмах машинного обучения.

Опираясь на полученные в ходе обучения классификаторов результаты, можно сделать вывод о том, что использование показателей устройств Интернета вещей для обнаружения атак является эффективным методом. Это подтверждают результаты тестирования обученных на полученных данных классификаторов. Они показали, что использование набора показателей можно эффективно использовать для обнаружения атак на устройства Интернета вещей и позволили правильно обнаружить, в лучшем случае, 99.84% всех атак во время теста.

# 3 ПРИМЕНЕНИЕ АВТОМАТОВ ДЛЯ ОБЕСПЕЧЕНИЯ АДАПТИВНОЙ БЕЗОПАСНОСТИ

Применение методов контроля поведения хорошо сочетается с концепцией распределенных сетей из маломощных устройств [17, 19], однако имеет один недостаток, связанный с потерей отказоустойчивости. Недостаточная отказоустойчивость обусловлена полным исключением узла с низким показателем доверия из сети.

Для повышения отказоустойчивости как при атаках, так и при неблагоприятных условиях работы сети, предлагается использовать адаптивное поведение, в рамках которого узел будет изменять принципы взаимодействия со своими соседями. Повышение доступности будет обеспечено за счет изменения поведения относительно вредоносного или неисправного узла. Это связано с тем, что узел не будет полностью исключен из сети. Для реализации такого поведения необходим механизм принятия решений, который приводят к адаптивному поведению.

В рамках раздела исследуется концепция адаптивной системы, которую можно применить для обеспечения безопасности узлов беспроводной сенсорной сети. Представлен анализ существующих адаптивных систем, их классификация и выбор наиболее оптимальной системы в рамках решаемых задач. Представлен математический аппарат для реализации выбранной адаптивной системы. В ходе работы будет осуществлён также выбор наиболее подходящих параметров для построения адаптивной системы.

## 3.1 Анализ существующих адаптивных систем и их классификация

Необходимо ввести понятие адаптивной системы в контексте текущего исследования. Адаптивная система – система, которая имеет возможность менять свое поведение во время работы в зависимости от изменений, которые происходят в окружающей её среде. Такие системы применимы в различных областях, например, в областях искусственного интеллекта [17], эволюционных и

биологических алгоритмах [18], отказоустойчивых вычислениях [19], робототехнике и так далее.

В рамках данной работы адаптивное поведение позволит узлам беспроводной сенсорной сети подстроиться под действия злоумышленников или неисправных узлов и выработать методологию взаимодействия с ними для минимизации деструктивного воздействия.

Далее будет рассмотрена классификация адаптивных систем, которая подразделяет их на 4 обширных типа [42].

Первый тип описывается следующим образом: подход возникает за счет анализа всевозможных воздействий, на которые система должна реагировать. На стадии проектирования требуется предусмотреть всевозможные ответные реакции, которые будет осуществлять система. На этом этапе разрабатывается поведенческая модель, которая содержит точки принятия решения: точка принятия решения представляет собой конструкцию «если … то … иначе…». Адаптация такого типа имеет смысл в следующих ситуациях:

1. Требуется интеллектуальная система, которая способна работать в зависимости от текущего состояния окружающей среды.
2. Изменения, которые происходят в системе, затрагивают только наблюдаемые атрибуты окружающей среды.
3. На стадии проектирования существует возможность предвидеть все возможные изменения окружающей среды, которые представляют интерес.
4. Работа с неопределенностью не является центральной в процессе работы системы.

Адаптация второго типа заключается в следующем: Она предлагает несколько альтернативных стратегий для достижения одной и той же цели. Различные стратегии требуются для анализа различных цепочек развития событий. Для реализации этого типа адаптации требуется провести детальное исследование большого числа альтернативных способов развития событий. Для выбора стратегии выполнения, требуется провести комплексный анализ окружающей

среды, а также учесть желаемое качество обслуживания. Условия применения адаптации такого типа:

1. Когда изменения могут повлиять на условия.
2. Если в системе требуется учитывать некоторую неопределенность.
3. Не представляется возможным рассчитать все изменения на стадии разработки, поэтому предпочтительнее их обнаружить во время выполнения.
4. Требуется наличие интеллектуальной системы, функционал которой включает способность обнаруживать и восстанавливать отклонения во время выполнения между поведением и условиями.

Третий тип адаптации является усовершенствованием описанных ранее адаптаций. Усовершенствование заключается в том, что данная система обладает набором определенных функциональных возможностей. Такая система может быть использована для создания специальных моделей поведения, которые не предусмотрены ни одной из предопределенных политик. Такую систему удобно использовать в областях, в которых не определены знания об окружающей среде, а также заранее не определены условия. Такую адаптивную систему, как правило, используют в следующих ситуациях:

1. В системах, в которых не существует возможности полностью предвидеть все проблемные ситуации.
2. Происходит частое изменение условий работы системы из-за внешних факторов.
3. Требуется наличие возможности собирать специальные функциональные возможности во время работы системы.
4. Функциональные возможности системы могут динамически изменяться извне.

Последний, четвертый тип адаптивных систем, заключается в следующем: адаптация такого типа является представителем самого высокого уровня интеллектуальных систем, так как она способна отслеживать своё состояние, обучаться во время работы и производить изменения самой себя. Такие системы полезны в тех случаях, когда могут возникнуть ситуации, для которых нет

известной стратегии решений или каких-либо действий. В таких случаях система может пересмотреть свою модель действия и подстроиться под не заданную заранее ситуацию.

Наиболее оптимальным адаптивным подходом к обеспечению безопасности беспроводных сенсорных сетей, с учетом специфики их работы, является третий тип адаптации. Более совершенным является четвёртый тип адаптации, но в рамках применимости в БСС он является неоправданно дорогим с точки зрения требуемых ресурсов и сложности в разработке. В качестве модели, реализующей третий тип адаптации предлагается использовать обучающийся автомат (ОА, Learning Automaton, LA).

Автомат можно представить в виде автоматической машины или механизма, который определенным образом отвечает на последовательность команд для достижения определенного результата. Автомат отвечает либо на заранее установленный набор правил, либо подстраивается под изменения среды, в которой он действует.

Термин обучение в психологии означает акт приобретения знаний и изменение поведения на основе полученного опыта. Так, предлагаемый автомат, адаптируется под ответы среды путем ряда взаимодействий с ней. Далее он пытается найти оптимальное действие из возможных вариантов, которые предлагает ему случайная стационарная или нестационарная среда, в которой он функционирует. Таким образом, автомат работает как программа принятия решений для нахождения оптимального действия.

Таким образом, ОА применяются при оптимизации задач, в которых из множества действий требуется определить оптимальное. Следует отметить, что обучение в этом плане эффективно только тогда, когда в системе, в которой функционирует автомат, присутствуют высокие уровни нечеткости. В системах с низким уровнем нечеткости обучение на основе ОА может быть неподходящим средством при выборе [12.3]. Среда применения беспроводных сенсорных сетей удовлетворяет данному условию.

Первые научно-исследовательские работы, связанные с моделями ОА, относятся к трудам советского математика М. Л. Цетлина, который подробно изучал детерминированные ОА.

### 3.2 Марковский процесс

Состояние обучающегося автомата в произвольный момент времени описывается цепью Маркова. В связи с этим для полного понимания принципов работы обучающихся автоматов, в рамках беспроводных сенсорных сетей, необходимо сначала представить математическую модель цепи Маркова.

Марковский процесс принятия решений – это математическая основа моделирования и решения задач, в которой необходимо предпринять последовательность взаимосвязанных решений в условиях неопределенности. Чтобы определить Марковский процесс принятия решений, нужно задать кортеж из четырех элементов (S,A,P.(·,·),R.(·,·)), где:

1. S – конечное число состояний.
2. A – конечное число действий (часто представляется в следующем виде: $A_s$ – конечное число действий, доступных из состояния s).
3. $P_a(s, s') = \Pr(s_{t+1} = s' | s_t = s, a_t = a)$ вероятность того, что действие $a$ в состоянии $s$ во время $t$ перейдет в состояние $s'$ ко времени $t + 1$.
4. $R_a(s, s')$ вознаграждение, получаемое после перехода в состояние $s'$ из состояния $s$ с вероятностью перехода $P_a(s, s')$.

Фактически Марковский процесс принятия решений представляет собой конечный автомат, где происходят переходы между различными состояниями S. Построив правильную цепочку переходов, можно эффективно взаимодействовать со злоумышленником, создавая таким образом политику взаимодействия с ним.

На примере будет рассмотрено использование Марковского процесса принятия решений, описывающего взаимодействие узла беспроводной сенсорной сети с другим неисправным/вредоносным узлом. Модель в виде конечного автомата представлена на рисунке 1.

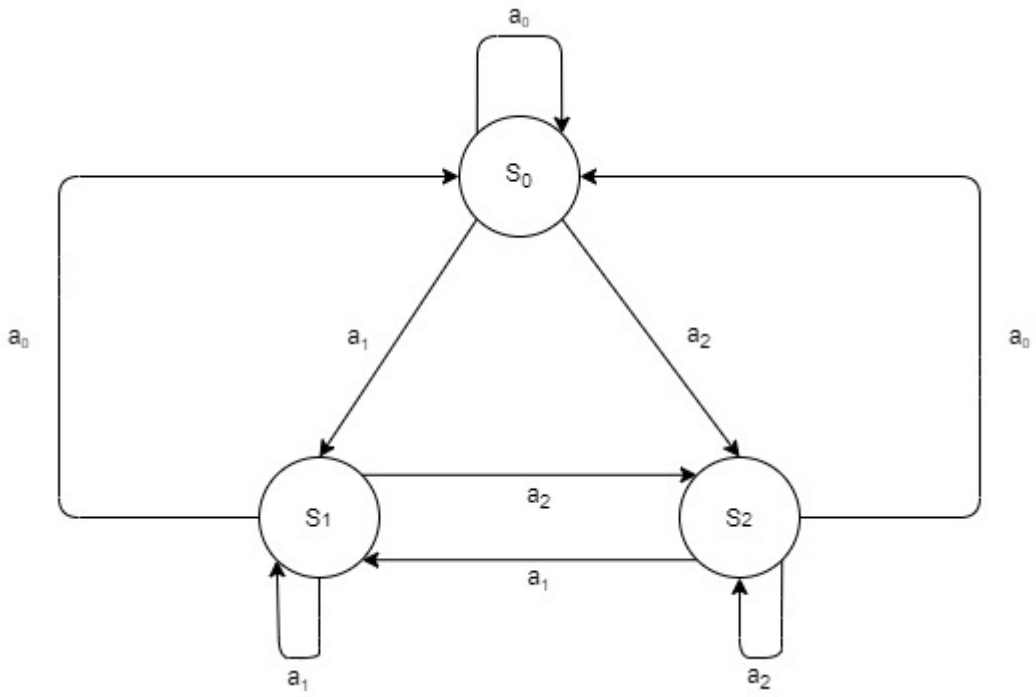



Рис. 1. Представление Honeypot в виде конечного автомата

На рисунке представлен набор состояний: $S_0, S_1, S_2$, а также соответствующие действия $a_0, a_1, a_2$, с помощью которых производятся переходы между состояниями. Из начального состояния $S_0$ можно равновероятно перейти в состояния $S_1$, $S_2$ или остаться в $S_0$. При поступлении некоторого сообщения от злоумышленника выполняется одно из действий $a_0, a_1, a_2$. В зависимости от ответной реакции злоумышленника на это действие (либо соединение было сохранено, либо соединение было разорвано) пересчитываются вероятности выбора действий, либо в сторону увеличения вероятности выполнения некоторого действия, $a$, либо в сторону уменьшения вероятности выполнения этого действия при таком сообщении от злоумышленника. Таким образом, происходит перерасчет вероятностей для различных действий. После того, как в какой-то момент соединение будет разорвано, происходит возвращение в состояние $S_0$, и процесс запускается заново для новых сообщений, поступающих от злоумышленника, однако при осуществлении переходов теперь будут учитываться ранее полученные вероятности. Следовательно, на ходу будет строиться политика ответов на запросы злоумышленника, которая позволит более оптимально с ним взаимодействовать

(не расходовать ресурсы, обрабатывая его сообщения, не передавать ему сообщения и т.д.).

### 3.3 Обучающиеся автоматы

В области теории автоматов обучающийся автомат может быть определен как структура, состоящая из пяти элементов: множества состояний, множества выходных координат и действий, входных координат, функции, которая преобразовывает текущее состояние и выходную координату в следующее состояние и функции, которая преобразовывает текущее состояние (и входные координаты) в текущую выходную координату [12.3, 8–11].

Обучающийся автомат определяется как пятерка $<A, B, Q, F(\cdot, \cdot), G(\cdot)>$, где:

— $A = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ – это множество выходных координат или действий, а α(t) – действие, выбранное автоматом в текущий момент времени t;

— $B = \{\beta_1, \beta_2, \ldots, \beta_m\}$ – это множество входных координат для автомата. β(t) – выходная координата в текущий момент времени t. Множество B может быть конечным или бесконечным. В рамках раздела рассматривается случай, когда $m = 2$, т.е. $B = \{0, 1\}$, где $\beta = 0$ – событие, за которое ОА поощряется, а $\beta = 1$ – событие, за которое назначается штраф;

— $Q = \{q_1, q_2, \ldots, q_s\}$ – это множество конечных состояний, где $q(t)$ обозначает состояние автомата в текущий момент времени $t$;

— $F(\cdot, \cdot) : Q \times B \rightarrow Q$ – это преобразование состояния и входных координат в текущий момент времени t так, чтобы $q(t+1) = F(q(t), \beta(t))$. Она называется функцией перехода, т.е. функция, которая определяет состояние автомата в следующий момент времени $t + 1$. Это преобразование может быть, как детерминированным, так и стохастическим;

— $G(\cdot)$ – это преобразование $G : Q \rightarrow A$, именуемое функцией выхода. В зависимости от состояния в конкретный момент времени, эта функция определяет выходную координату автомата в тот момент времени, когда $\alpha(t) = G(q(t))$. Опять же, это преобразование может быть, как детерминированным, так и стохастическим. Без потери общности G является детерминированным.

Если множества $Q$, $B$ и $A$ являются конечными, то и автомат считается конечным.

Средой $E$, обозначается пространство (область), в котором автомат функционирует. Среда включает в себя все внешние факторы, которые влияют на действия автомата. Среда может быть описана математически тройкой $A$, $C$, $B$, которые обозначают:

— $A = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ – множество действий;

— $B = \{\beta_1, \beta_2, \ldots, \beta_m\}$ – множество выходных координат среды. В рамках исследования рассматривается случай, когда $m = 2$, т.е. $\beta = 0$ обозначает поощрение, а $\beta = 1$ – штраф;

— $C = \{c_1, c_2, \ldots, c_r\}$ – множество вероятностей штрафа, где элемент ci ∈ C отвечает на входное воздействие $\alpha i$;

Процесс обучения основан на цикле, включающем в себя два элемента: случайную среду (СС) и ОА, как показано на рисунке 3.4. В процессе обучения ОА непрерывно взаимодействует со средой для обработки ответов на ее различные действия (т.е. ее выборы). Путем необходимых взаимодействий ОА пытается определить оптимальное действие, предлагаемое СС. Сам процесс обучения представлен в виде множества взаимодействий между СС и ОА.

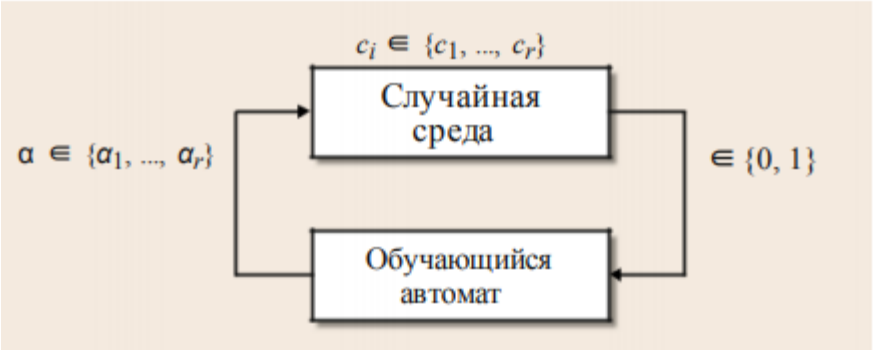



Рисунок 3.4 – Схема работы автомата

Случайная среда предлагает автомату выбор из множества возможных действий $\{\alpha_1, \alpha_2, \ldots, \alpha_r\}$. Автомат выбирает одно из этих действий, например, $\alpha_i$, которое служит входной координатой для СС. Поскольку СС известно о базовом

распределении вероятностей штрафа в системе, в зависимости от вероятности штрафа $c_i$ соответствующей $\alpha_i$, она инициирует либо поощрение (обозначаемое, как правило, значением 0) для ОА, либо штрафом (обозначаемым, как правило, значением 1). Информация о поощрении/штрафе (в зависимости от действия), предоставляемая ОА помогает выбрать ему дальнейшее действие. Повторяя вышеупомянутый процесс посредством ряда взаимодействий со средой, ОА в конечном итоге пытается определить оптимальное действие.

Далее необходимо привести несколько важных определений, которые будут в дальнейшем использоваться. $P(t)$ обозначает вектор вероятности действия, где $P(t) = [p_1(t), p_2(t), \ldots, p_r(t)]$, в котором каждый элемент вектора $p_i(t) = Pr[\alpha(t) = \alpha_i]$, $i = 1, \ldots, r$, удовлетворяет условию (1)

$$\sum_{i=1}^{r} p_i(t) = 1 \forall t \qquad (1)$$

Учитывая вектор вероятности действия $P(t)$ в момент времени $t$, средний штраф можно представить, как математическое ожидание (2).

$$M(t) = E[\beta(t)|P(t)] = Pr[\beta(t) = 1|P(t)] = \sum_{i=1}^{r} Pr[\beta(t) = 1|\alpha(t) = \alpha_i]Pr[\alpha(t) = \alpha_i] = \sum_{i=1}^{r} c_i\, p_i(t) \qquad (2)$$

Средний штраф для случайного автомата имеет вид (3).

$$M_0 = \frac{1}{r}\sum_{i=1}^{r} c_i \qquad (3)$$

Когда $t \to \infty$, если средний штраф хотя бы асимптотически $M(t) < M_0$, автомат в целом считается лучше случайного автомата. $E[M(t)]$ имеет вид: $E[M(t)] = E\{E[\beta(t)|P(t)]\} = E[\beta(t)]$.

### 3.4 Классификация (Разновидности) обучающихся автоматов

Детерминированный обучающийся автомат.

Автомат называется детерминированным автоматом, если и функция перехода $F(\cdot, \cdot)$, и функция выхода $G(\cdot)$, обозначенные в предыдущем п.п.,

являются детерминированными. Так, в детерминированном автомате следующее состояние и действие может быть определено однозначно при наличии текущего состояния и входного значения.

Стохастический (Недетерминированный) обучающийся автомат.

Если функция перехода $F(\cdot, \cdot)$, либо функция выхода $G(\cdot)$ является стохастической, автомат называется стохастическим автоматом. В таком автомате, если текущее состояние и входная координата определены, следующие состояния и действия не могут быть определены однозначно. В таком случае $F(\cdot, \cdot)$ отражает только вероятности достижения различных состояний из данного состояния. Через $\mathbf{F}^{\beta_1}$, $\mathbf{F}^{\beta_2}$, . . ., $\mathbf{F}^{\beta_m}$ обозначаются матрицы условных вероятностей, где каждая из этих условных матриц $\mathbf{F}^{\beta}$ (для $\beta \in B$) – матрица $s \times s$, произвольный элемент которой $f_{ij}^{\beta} = Pr[q(t+1) = q_j | q(t) = q_i, \beta(t) = \beta]$, i, j $= 1, 2, \ldots,$ s.

Каждый элемент $f_{ij}^{\beta}$ матрицы $\mathbf{F}^{\beta}$ обозначает вероятность перехода автомата из состояния $q_i$ в состояние $q_j$ при получении входящего сигнала $\beta$ от СС. $\mathbf{F}^{\beta}$ – это марковская матрица и отсюда следует, что:

$$\sum_{j=1}^{s} f_{ij}^{\beta} = 1, \text{где } \beta \in B; i = 1, 2, \ldots, s\,.$$

Аналогичным образом в стохастическом автомате, если $G(\cdot)$ – стохастическая, получаем $g_{ij} = Pr\{\alpha(t) = \alpha_j | q(t) = q_i\}$, $i, j = 1, 2, \ldots, s$, где $g_{ij}$ обозначает элементы матрицы условных вероятностей размерностью $s \times r$. Таким же образом $g_{ij}$ обозначает вероятность того, что когда автомат находится в состоянии $q_i$, он выбирает действие $\alpha_j$. Таким образом, получается:

$$\sum_{j=1}^{r} g_{ij} = 1\,, \text{где } i = 1, 2, \ldots, s$$

Автомат, описывающий поведение беспроводной сенсорной сети будет являться стохастическим (недетерминированным).

Обучающийся автомат с фиксированной структурой

Стохастический ОА, если условные вероятности fij β и gij являются постоянными, т.е. они не меняются с шагом по времени t и по входной последовательности, называется стохастическим автоматом с фиксированной структурой (САФС). Известные варианты автоматов данного типа были предложены Цетлиным [12.1, 2], Крыловым [12.26] и Кринским [12.27] – все они являются - оптимальными. Более подробно в разд. [12.3].

Обучающийся автомат с переменной структурой

В отличие от САФС, стохастический автомат с переменной структурой (САПС) – это такой автомат, в котором вероятности перехода между состояниями не являются фиксированными. В таком автомате переходы между состояниями или сами вероятности действия корректируются самостоятельно в каждый момент времени по соответствующей схеме. Вероятности перехода $f_{ij}^{\beta}$ и функция выхода $g_{ij}$ меняются со временем, а вероятности действия корректируются, основываясь на входном значении. Речь о данном автомате идет в контексте линейных схем, однако концепты, описанные ниже, могут быть также расширены до нелинейных корректирующих схем. Типы автоматов, которые корректируют вероятности перехода с течением времени, были представлены в 1963 году Варшавским и Воронцовой [12.6]. Во время своего функционирования САПС опирается на генератор случайных чисел. Выбранное действие зависит от вектора распределения вероятности действия, который, в свою очередь, корректируется, основываясь на входящей координате поощрения/штрафа, которую автомат получает от СС.

САПС – это пятерка $< Q, A, B, G, T >$, где $Q$ обозначает различные состояния автомата, $A$ – множество действий, $B$ – множество ответов, которые ОА получает от среды, $G$ – функция выхода, а $T$ – корректирующая схема вероятности действия $T : [0, 1]^r \times A \times B \rightarrow [0, 1]^r$ так, чтобы $P(t + 1) = T [ P(t), \alpha(t), \beta(t)]$ , где $P(t)$ – это вектор вероятности действия.

Как правило, САПС включает в себя корректировку вероятностей как состояния, так и действия. Для простоты на практике принимается, что в таком

автомате каждое состояние отвечает на определенное действие, в случае которого отображение перехода по действию $G$ становится тождественным, а количество состояний $s$ равняется количеству действий $r$ ($s = r < \infty$). САПС можно проанализировать при помощи дискретного марковского процесса, построенного на соответствующем множестве состояний. Если схема корректировки вероятностей $T$ не зависит от времени, $\{P(t)\}_{t\geq 0}$ – дискретно-однородный марковский процесс, а вектор вероятности в текущий момент времени $P(t)$ (вместе с $\alpha(t)$ и $\beta(t)$) определяется полностью $P(t + 1)$. Отсюда следует, что каждая определенная схема коррекции $T$ определяет разные типы алгоритмов обучения. Всего можно выделить четыре таких алгоритма:

— поглощающие алгоритмы – алгоритмы, в которых схема коррекции T, выбирается таким образом, чтобы в марковском процессе имелись поглощающие состояния;

— непоглощающие алгоритмы– алгоритмы, в которых у марковского процесса отсутствуют поглощающие состояния;

— линейные алгоритмы – алгоритмы, в которых $P(t + 1)$ является линейной функцией $P(t)$;

— нелинейные алгоритмы – алгоритмы, в которых $P(t +1)$ является нелинейной функцией $P(t)$.

В САПС, если выбранное действие αi поощряется, вероятность текущего действия возрастает, а вероятности для всех остальных действий понижаются. С другой стороны, если выбранное действие αi наказывается штрафом, вероятность текущего действия понижается, в то время как вероятности для других действий обычно повышаются. Отсюда получаются следующие разнотипные схемы обучения для САПС:

— поощрение-штраф (ПШ): В обоих случаях, т.е. и тогда, когда автомат поощряется, и наказывается штрафом, вероятности действия корректируются;

— нештраф-штраф (НШ): В случае штрафа вектор вероятностей действия корректируется, в то время как в случае поощрения вероятности действия не увеличиваются и не уменьшаются;

— поощрение-нештраф (ПН): Вектор вероятностей действия корректируется при поощрении автомата и не меняется в случае штрафа.

Обучающийся автомат считается непрерывным, если схема корректировки вероятностей $T$ является непрерывной, т.е. вероятность выбора действия может быть любым действительным числом в замкнутом промежутке [0, 1]. В САПС, если действия r осуществляются в стационарной среде с $\beta = \{0, 1\}$, схема корректировки общей вероятности действия для непрерывного автомата представлена ниже. Предположим, что выбрано действие αi и, следовательно, $\alpha(t) = \alpha_i$. Скорректированные вероятности действия могут быть выражены как:

— для $\beta(t) = 0, \forall j \neq i, p_j(t+1) = p_j(t) - g_j[P(t)]$,

— для $\beta(t) = 1, \forall j \neq i, p_j(t+1) = p_j(t) + h_j[P(t)]$.

Поскольку $P(t)$ – вектор вероятности, $\sum_{j=1}^{r} p_j(t) = 1$, поэтому

— при $\beta(t) = 0, p_i(t+1) = p_i(t) + \sum_{j=1, i\neq j}^{r} g_j[P(t)]$;

— при $\beta(t) = 1, p_i(t+1) = p_i(t) - \sum_{j=1, i\neq j}^{r} h_j[P(t)]$.

Функции $h_j$ и $g_j$ являются неотрицательными и непрерывными в [0, 1] и подчиняются $\forall i = 1, 2, ..., r, \forall P \in (0, 1)^R$, $0 < g_j(P) < p_j$ и $0 < \sum_{j=1, i\neq j}^{r} p_j + h_j(P)$

Для непрерывного линейного САПС, следующие четыре схемы обучения широко изучены в научной литературе. Объяснение этих схем идет в рамках случая с двумя действиями; их расширение до случая с r-действием, где $r > 2$, прямолинейно и освещается в [12.3]. Четыре схемы обучения:

— линейная схема поощрение-нештраф ($L_{ПН}$);

— линейная схема нештраф-штраф ($L_{НШ}$);

— симметричная линейная схема поощрение- штраф ($L_{ПН}$);

— линейная схема поощрение - ε-штраф ($L_{П-εШ}$).

Пусть для ОА с двумя действиями $g_i[P(t)] = ap_j(t)$ и $h_j[P(t)] = b(1 - p_j(t))$. В выражении $a$ и $b$ называются параметрами поощрения и штрафа, и подчиняются следующим неравенствам: $0 < a < 1$, $0 \leq b < 1$. Равенство будет использовано в

дальнейшем для получения корректирующих равенств вероятности действия. Вышеупомянутые линейные схемы достаточно широко используются в области ОА из-за удобства аналитической трактовки. Они отличаются существенно разными характеристиками, как видно в таблице 1.

Таблица 1 – Сравнение схем обучения ОА

| **Схема обучения** | **Параметры обучения** | **Полезность** | **Оптимальность** | **Эргодическая/поглощающая** |
|---|---|---|---|---|
| $L_{ПН}$ | a > 1, b = 0 | Хорошая | ε-оптимальная, при a -> 0 | Поглощающая (стационарная E) |
| $L_{НШ}$ | a = 0, b > 0 | Очень плохая | - | Эргодическая (нестационарная E) |
| $L_{ПШ}$ (симметричная) | a = b, a, b > 0 | Плохая | Никогда ε-оптимальная | Эргодическая (нестационарная E) |
| $L_{П-εШ}$ | a > 0, b << a | Хорошая | ε-оптимальная, при a -> 0 | Эргодическая (нестационарная E) |

Схема $L_{ПН}$ была впервые представлена Норманом и в дальнейшем исследована Шапиро и Нарендрой. Она основывается на принципе того, что когда автомат получает предпочтительный ответ (т.е. поощрение) от среды, вероятности действия корректируются линейным образом, а если автомат получает нежелательный ответ (т.е. штраф) от среды, они остаются неизменными.

Равенства корректировки вероятности для этой схемы могут быть упрощены до:

$p_1(t+1) = p_1(t) + a[1 - p_1(t)]$ , если $\alpha(t) = \alpha_1$ и $\beta(t) = 0$,

$p_1(t+1) = (1 - a)\, p_1(t)$, если $\alpha(t) = \alpha_2$ и $\beta(t) = 0$

$p_1(t+1) = p_1(t)$, если $\alpha(t) = \alpha_1$ или $\alpha_2$ и $\beta(t) = 1$.

Видно, что при выборе действия $\alpha_i$ и получении поощрения вероятность $p_i(t)$ возрастает, а другая вероятность $p_j(t)$ (т.е. $j \neq i$) уменьшается. Если выбрано $\alpha_1$ или $\alpha_2$ и получен штраф, $P(t)$ не меняется. Равенство показывает, что у схемы $L_{ПН}$

имеются векторы $[1, 0]^T$ и $[0, 1]^T$ в качестве двух поглощающих состояний. И в действительности при вероятности 1 она поглощается в одно из этих поглощающих состояний. Таким образом, конвергенции схемы $L_{ПН}$ зависит от природы начальных состояний и вероятностей. Указанная схема не подходит для нестационарных сред. С другой стороны, для случайных стационарных сред схема $L_{ПН}$ и абсолютно целесообразна, и $\varepsilon$-оптимальна. Схемы $L_{НП}$ и $L_{ПШ}$ были разработаны похожим образом. Они и их соответствующий анализ представлены. Так называемые условия симметрии для функций $g(\cdot)$ и $h(\cdot)$, которые приводят к абсолютной целесообразности ОА также освещаются в.

Дискретный обучающийся автомат

Алгоритмы САПС, представленные ранее являются непрерывными, т.е. вероятности действий могут принимать любое действительное значение в промежутке [0, 1]. У ОА выбор действия определяется посредством генератора случайных чисел (ГСЧ). Для того чтобы увеличить скорость конвергенции этих автоматов, Татачар и Оммен представили дискретные алгоритмы для САПС, в которых они предложили дискретизацию поля вероятностей. Разные свойства (поглощающие и эргодические) этих обучающихся автоматов и схемы корректировки вероятности действия для данных дискретных автоматов (как, например, их непрерывные аналоги) были позднее подробно исследованы Омменом и другими. Дискретные автоматы можно воспринимать как некий гибрид САФС и САПС. Дискретизнация коцептуализирована через ограничение вероятности выбора действий строго до фиксированного количества значений в замкнутых промежутках [0, 1]. Таким образом, корректировка вероятностей действия достигается пошагово, а не непрерывным образом, как в случае с непрерывным САПС. Очевидно, что, как и САФС, они обладают конечными множествами, однако из-за того, что у них имеются векторы вероятности действия, которые являются случайными векторами, они ведут себя так же, как САПС.

Дискретный ОА также может быть двух типов:

— линейный – значения вероятности действия равномерно распределены в замкнутом промежутке [0, 1];

— нелинейный – значения вероятности распределены неравномерно в промежутке [0, 1].

Пожалуй, лучшим мотивом использования дискретности является преодоление устойчивого ограничения непрерывных обучающихся автоматов, т.е. низкой скорости конвергенции. Это достигается за счет ограничения базовых допущений автомата. Изначально допущение заключалось в том, что ГСЧ могли генерировать действительные значения с произвольной точностью. В случае дискретного ОА, если вероятность действия находится сравнительно близко к величине, равной единице, вероятность выбора этого действия увеличивается до величины, равной единице (при соответствующих условиях), напрямую, а не асимптотически. Вторым важным преимуществом дискретизации является то, что она более практична в том смысле, что ГСЧ, используемые в непрерывных САПС, могут лишь теоретически выдавать любое значение в промежутке [0, 1], в то время как все машинные внедрения ГСЧ используют генераторы псевдослучайных чисел. Другими словами, множество возможных случайных значений не бесконечно в [0, 1], а конечно.

Последний, но не менее важный момент дискретизации в плане внедрения и представления. Дискретные внедрения автомата используют целые числа для отслеживания количества кратных $1/N$ вероятностей действия, где $N$ – это так называемый разрешающий параметр. Он не только увеличивает скорость конвергенции алгоритма, но также уменьшает время в плане количества временных циклов, требуемых процессору для выполнения каждой итерации задачи, как и объем требуемой памяти. Дискретные алгоритмы зарекомендовали себя на практике как более эффективные и в плане времени, и в плане места по сравнению с непрерывными алгоритмами. По аналогии с парадигмой непрерывного ОА, также сообщалось и о дискретных версиях – автоматах $DL_{ПН}$, $DL_{НШ}$ и $DL_{ПШ}$. Их конструкция, анализ и свойства приведены в [12.30, 32–34] и обобщены в таблице 2.

Таблица 2 – Свойства дискретных схем обучения

| **Схема обучения** | **Параметры обучения** | **Полезность (хорошая/плохая)** | **Оптимальность (при $N \to \infty$)** | **Эргодическая** |
|---|---|---|---|---|
| $DL_{ПН}$ | $N > 0$ | Хорошая | $\varepsilon$-оптимальная | Поглощающая (стационарная Е) |
| $DL_{НШ}$ | $N > 0$ | Очень хорошая | Целесообразная | Эргодическая (нестационарная Е) |
| $ADL_{НШ}$ | $N > 0$ | Хорошая, медленная | $\varepsilon$-оптимальная | Искусственно поглощающая (стационарная Е) |
| $DL_{ПШ}$ | $N > 0$ | Допустимая | $\varepsilon$-оптимальная, если $c_{min} < 0.5$ | Эргодическая (нестационарная Е) |
| $ADL_{ПШ}$ | $N > 0$ | Хорошая | $\varepsilon$-оптимальная | Искусственно поглощающая (стационарная Е) |
| $MDL_{ПШ}$ | $N > 0$ | Хорошая | $\varepsilon$-оптимальная | Эргодическая (нестационарная Е) |

### 3.5 Алгоритмы оценивания

Скорость конвергенции обучающихся алгоритмов — один из самых важных факторов, который являлся основной причиной разработки семейства дискретных алгоритмов. С той же целью Татачар и Састри разработали новый класс алгоритмов, именуемых алгоритмами оценивания, скорость конвергенции у которых выше, чем у всех предыдущих семейств. Эти алгоритмы, как и предыдущие, обеспечивают функционирование и корректировку вектора вероятности действия. Однако, в отличие от предыдущих, данные алгоритмы также проводят расчеты каждого действия, которое поощряется при помощи вектора оценивания поощрения, а затем использует эти расчеты в уравнениях корректировки вероятности. Вектор оценивания-поощрения, как правило обозначается $\hat{D}(t) = [\hat{d}_1(t), \ldots, \hat{d}_r(t)]$. Вектор соответствующего состояния обозначается $Q(t) = <P(t), \hat{D}(t)>$.

В случайной среде данные алгоритмы помогают при выборе действия путем увеличения уверенности в возможностях поощрения разных действий; например, сначала эти алгоритмы обрабатывают каждое действие по нескольку раз, а затем (в одной из версий) могут увеличить вероятность определенного действия наивысшим оцениванием поощрения. Из этого вытекает схема с более высокой точностью при выборе правильного действия. Предыдущие алгоритмы САПС без оценивания корректируют вектор вероятностей напрямую, основываясь на ответе, который автомат получает от среды, где, в зависимости от используемого типа схемы корректировки вектора, вероятность выбора поощряемого действия в последующий момент времени увеличивается, а вероятности выбора других действий могут быть снижены. При этом алгоритмы оценивания корректируют вектор вероятностей и на основе вектора оценивания, и на основе текущей обратной связи, которую автомат получает от среды. Среда влияет на вектор вероятностей как напрямую, так и косвенно, последнее влияние является результатом оценивания расчетов поощрения разных действий. Таким образом, это может привести к увеличению вероятностей действия отличных от поощряемого в текущий момент времени действия.

Даже несмотря на наличие дополнительной вычислительной сложности при расчетах поощрения, данные алгоритмы оценивания по производительности на порядок превосходят представленные ранее алгоритмы без оценивания. Ланктот и Ооммен [12.31] позднее представили дискретные версии указанных алгоритмов оценивания, скорость конвергенции у которых оказалась даже выше.

### *3.5.1 Непрерывные алгоритмы оценивания*

Татачар и Састри представили класс непрерывных алгоритмов оценивания [12.35–38], в которых схема корректировки вероятности $T$ является непрерывной, т.е. вероятность выбора того или иного действия может быть выражена любым действительным числом в замкнутом промежутке [0, 1]. Как было сказано ранее, дискретные версии данных алгоритмов были представлены Оомменом в соавторстве с Ланктотом и Агаче [12.31,40].

Алгоритм преследования

Семейство алгоритмов преследования — это класс алгоритмов, которые преследуют действие, которое автомат воспринимает в текущий момент времени как оптимальное. Первый алгоритм преследования, именуемый алгоритмом $CP_{ПШ}$, представленный Татачаром и Састри [12.36, 41], преследует оптимальное действие путем изменения вероятности текущего оптимального действия вне зависимости от получаемого ответа от среды, т.е. поощрения или штрафа. В таком случае поощряется то действие, которое воспринимается в текущий момент времени как оптимальное, а его величина вероятности действия увеличивается вместе с величиной прямопропорциональной его расстоянию до единицы, а именно $1\text{-}p_m(t)$, где менее оптимальные действия штрафуются, а их вероятности уменьшаются пропорционально.

На основе распределения вероятностей $P(t)$, алгоритм выбирает действие $\alpha(t)$. В зависимости от ответа в виде поощрения или штрафа, он увеличивает компонент $P(t)$, который обладает максимальным расчетом текущего поощрения, и он уменьшает вероятность в соответствии с остальными действиями. Наконец, алгоритм корректирует производимые расчеты вероятности поощрения выбранного действия, это является принципом, лежащим в основе сохранения и использования производимых расчетов. Вектор оценивания $\hat{D}(t)$ может быть рассчитан по следующей формуле, которая производит оценивание по методу максимального правдоподобия:

$$d_i(t) = \frac{W_i(t)}{Z_i(t)}, \forall i = 1, 2, \dots, r$$

где $W_i(t)$ — количество раз, когда за действие $\alpha_i$ следует поощрение до текущего момента времени $t$, а $Z_i(t)$ — количество раз, когда было выбрано $\alpha_i$ до текущего момента $t$.

Данный алгоритм схож по принципу с алгоритмом $L_{ПШ}$, поскольку и алгоритм $CP_{ПШ}$, и алгоритм $L_{ПШ}$ повышают/понижают вектор вероятностей действия вне зависимости от того, какой ответ автомат получает от среды (поощрение или штраф). Главное отличие заключается в том, каким образом

производятся расчеты поощрения, а также происходит корректировка по поощрению/штрафу. Следует подчеркнуть, что алгоритм без преследования сдвигает вектор вероятностей в направлении поощряемого последний раз действия, в то время как алгоритм преследования сдвигает вектор в направлении действия с самым высоким показателем поощрения. Татачар и Састри теоретически доказали их оптимальность, и доказали практически, что данные алгоритмы преследования более точны и быстры по сравнению с алгоритмами без преследования.

Алгоритм с временным разделением событий

Более продвинутый алгоритм, который мы называем алгоритмом ВРС для согласования с научными трудами, был разработан Татачаром и Састри.

Как и другие алгоритмы оценивания, алгоритм ВРС обеспечивает функционирование вектора расчетов поощрения $\hat{D}(t)$ и использует его для расчета вектора вероятности действия $P(t)$. При поощрении действия $\alpha_i(t)$ в соответствии с алгоритмом ВРС, компоненты вероятности с величиной поощрения выше $\hat{d}_i(t)$ интерпретируются иначе, чем те, которые имеют значение ниже $\hat{d}_i(t)$. Проявляется это в том, что алгоритм повышает вероятности всех действий, величина у которых выше, чем величина у выбранного действия, и понижает вероятности всех действий с меньшей величиной. Это происходит при помощи индикаторной функции $S_{ij}(t)$, которая принимает значение 1, если $\hat{d}_i(t) > \hat{d}_j(t)$, и значение 0, если $\hat{d}_i(t) \le \hat{d}_j(t)$. Таким образом, для корректировки вероятностей действия алгоритм ВРС использует и вектор вероятностей $P(t)$, и вектор расчетов поощрения $\hat{D}(t)$. Можно увидеть, что $P(t+1)$ косвенно зависит от ответа, который автомат получает от среды. Обратная связь от среды меняет значения компонентов $\hat{D}(t)$, который в свою очередь влияет на значения функций $f(\cdot)$ и $S_{ij}(t)$ [12.31, 37–39].

При работе алгоритма возможны три случая. Если $i$-ое действие поощряется, значения вероятности действий с показателями вознаграждения выше, чем показатели вознаграждения у выбранного в текущий момент времени действия, корректируются по формуле.

$$p_j(t+1) = p_j(t) - \lambda \left\{ f\left[\hat{d}_i(t) - \hat{d}_j(t)\right] \frac{p_i(t) - p_j(t)p_i(t)}{r-1} \right\},$$

когда $\hat{d}_i(t) < \hat{d}_j(t)$ поскольку функция является монотонной и возрастающей, видно, что $f[\hat{d}_i(t) - \hat{d}_j(t)]$ является отрицательной. Это приводит к тому, что значение $p_j(t+1)$ выше значения $p_j(t)$, и это указывает на то, что вероятность выбираемых действий, показатели у которых выше показателей выбранного в текущий момент времени действия, будет возрастать. Вероятности всех действий с показателями поощрения ниже показателя выбранного в текущий момент времени действия корректируются по следующей формуле: $p_j(t+1) = p_j(t) - \lambda f[\hat{d}_I(t) - \hat{d}_j(t)]p_j(t)$.

Знак у функции $f$ $[\hat{d}_i(t) - \hat{d}_j(t)]$ отрицательный, что указывает на то, что вероятность выбираемых действий, показатели которых меньше показателей выбранного в текущий момент времени действия, уменьшится.

Татачар и Састри доказали на практике, что алгоритм ВРС является ε-оптимальным. Они также наглядно показали, что алгоритм ВРС часто конвергирует на несколько порядков быстрее, чем схема $L_{ПН}$.

Общий алгоритм преследования

Агаче и Ооммен представили общий вариант алгоритма преследования ($СР_{ПШ}$), предложенного Татачаром и Састри. Их алгоритм, получивший название общий алгоритм преследования (ОАП), обобщает алгоритм Татачара и Састри путем преследования всех действий, обладающих значениями выше значения выбранного действия. В таком случае вероятность выбора неверного действия снижается до минимума. Агаче и Ооммен экспериментальным путем провели сравнение своего алгоритма преследования с используемыми на тот момент алгоритмами и обнаружили, что их алгоритм является самым оптимальным в плане скорости конвергенции. В алгоритме $СР_{ПШ}$ вероятность просчитанного оптимального действия доведена до максимума путем первоначального понижения вероятности всех действий следующим образом [12.40]

$p_j(t+1) = (1 - \lambda)\, p_j(t), j = 1, 2, \ldots, r.$

Сумма вероятностей действия равняется единице при помощи вероятностной меры Δ, которая получается по формуле

$$\Delta = 1 - \sum_{j=1}^{r} p_j(t+1) = 1 - \sum_{j=1}^{r} (1-\lambda) p_j(t) = 1 - \sum_{j=1}^{r} p_j(t) + \lambda \sum_{j=1}^{r} p_j(t) = \lambda$$

Затем вероятностная мера Δ прибавляется к вероятности оптимального просчитанного действия. Таким образом, алгоритм ОАП равномерно распределяет вероятностную меру Δ по отношению к действию, которое было оценено как превосходящее выбранное действие.

Поэтому $p_m(t+1) = (1-\lambda)\, p_m(t) + \Delta = (1-\lambda)\, p_m(t) + \lambda$, где $\hat{d}_m = \max_{j=1,2,\ldots,r}[\hat{d}_j(t)]$

Так, схема корректировки выводится по формуле:

$$p_j(t+1) = (1-\lambda) p_j(t) + \frac{\lambda}{K(t)}$$

если $\hat{d}_j(t) > \hat{d}_i(t), j \neq i$, $p_j(t+1) = (1-\lambda) p_j(t)$ ,

если $\hat{d}_j(t) \leq \hat{d}_i(t), j \neq i$, $p_i(t+1) = 1 - \Sigma p_j(t+1)$,

где $K(t)$ обозначает количество действий, значения у которых выше, чем у вероятности поощрения выбранного в текущий момент времени действия. Формальный алгоритм здесь опускается, но его можно найти в.

### *3.5.2 Дискретные алгоритмы оценивания*

Как было показано ранее, дискретные ОА превосходят непрерывные аналоги, а алгоритмы оценивания превосходят алгоритмы без оценивания в плане скорости конвергенции алгоритмов обучения. Используя зарекомендовавшие себя в прошлом возможности дискретизации, а именно увеличение скорости конвергенции обучающихся алгоритмов, Ланктот и Ооммен улучшили алгоритмы преследования и алгоритмы ВРС. Это привело к созданию классов алгоритмов обучения, именуемых в научной литературе дискретными алгоритмами оценивания (ДАО). Как и в случае с предыдущими дискретными алгоритмами, компоненты вектора вероятности действия могут принимать конечное множество дискретных значений в замкнутом промежутке [0, 1], который в свою очередь разделяется на несколько подпромежутков пропорциональных разрешающему

параметру N. При этом вектор оценивания поощрения продолжает функционировать для сохранения показателя вероятности поощрения каждого действия. Лактот и Ооммен показали, что каждый член алгоритма, принадлежащий к классу ДАО, должен обладатать двумя свойствами, известными как свойство усреднения и свойство монотонности, чтобы быть -оптимальным. Вместе эти свойства помогают доказать - оптимальность любого ДАО.

Считается, что ДАО с действиями r и разрешающим параметром N облдает свойством усреднения, если максимальное значение, на которое может опуститься вероятность действия за одну итерацию, ограничено 1/(*rN*).

Свойство монотонности

Пусть существует индекс m и в момент $t_0 < \infty$ так, чтобы $\hat{d}_m(t) > \hat{d}_j(t)$, $\forall j$ так, чтобы $j = m$ и $\forall t$ так, чтобы $t \geq t_0$, где $\hat{d}_m(t)$ — максимальный компонент $\hat{D}(t)$. Считается, что ДАО обладает свойством монотонности при присутствии целого числа $N_0$ так, чтобы все разрешающие параметры $N > N_0$, $p_m(t) \rightarrow 1$ с вероятностью 1, когда $t \rightarrow \infty$, где $p_m(t)$ - максимальный компонент $P(t)$. Дискретные варианты алгоритма преследования и алгоритм ВРС, обладающий свойством усреднения и монотонности, представлены далее.

Дискретный алгоритм преследования

Дискретный алгоритм преследования обозначается аббревиатурой ДАП и имеет схожесть с очень многими непрерывными аналогичными алгоритмами преследования, т.е. алгоритмом $СР_{ПН}$, за исключением того, что корректировки вероятностей действия для ДАП производятся дискретными шагами. Поэтому уравнения в алгоритме $СР_{ПН}$, которые включают в себя умножение на параметр обучения λ, заменяются на уравнения сложения или вычитания пропорциональные минимальному размеру шага. Как и в случае с алгоритмом $СР_{ПН}$, ДАП выполняется в три шага. Если $\Delta = 1/(rN)$ (где *N* обозначает решение, а *r* - количество действий) обозначает минимальный размер шага, целые кратные множества Δ обозначают размеры шага, в котором вероятности действия корректируются. Как и в случае с непрерывным алгоритмом поощрение-нештраф, где выбранное действие $\alpha(t) = \alpha_i$

штрафуется, а вероятности действия остаются неизменными. Вместе с тем, когда выбранное действие $\alpha(t) = \alpha_i$ поощряется и алгоритм не конвергирован, алгоритм уменьшается на целые кратные множества $\Delta$ вероятностей действия, которые не соответствуют наивысшей оценке поощрения. Ланктот и Ооммен показали, что ДАП обладает свойством усреднения и монотонности, и что, таким образом, он является $\varepsilon$-оптимальным. Кроме того, они доказали опытным путем, что в разных типах сред — от простых до сложных — ДАП по крайней мере на 60% быстрее алгоритма $СР_{ПШ}$.

Дискретный алгоритм ВРС

Ланктот и Ооммен также дискретизировали алгоритм ВРС и назвали его дискретным алгоритмом ВРС (ДВРС). Поскольку данный алгоритм основан на непрерывном варианте алгоритма ВРС, очевидно, что он обладает тем же уровнем сложности, если не большим. Лактот и Ооммен доказали теоретическим путем, что, как и алгоритм оценивания ДАП, этот алгоритм тоже обладает свойством усреднения и монотонности, в то же время имея качества непрерывного алгоритма ВРС. Помимо этого, они привели доказательство конвергенции указанного алгоритма. У алгоритма ДВРС имеются два параметра:

1. $\Delta = 1/(rN\theta)$, где $N$ — это, как и прежде, разрешающий параметр

2. $\theta$ — целое число, обозначающее самое большое значение, на которое любая из вероятностей действия может меняться за одну итерацию.

Дискретный общий алгоритм преследования

Агаче и Ооммен предложили дискретный вариант своего ОАП, представленного ранее. Их алгоритм, именуемый дискретным общим алгоритмом преследования (ДОАП), также в целом обобщает алгоритм преследования Татачара и Састри. Однако в отличие от алгоритма ВРС, он преследует все действия, показатели которых выше показателей выбранного действия. В целом, в любой итерации алгоритм просчитывает количество действий, показатели поощрения у которых выше, чем у выбранного в данный момент времени действия,

обозначаемого K(t), откуда следует, что вероятность всех действий, показатели у которых выше, чем у выбранного действия, увеличивается на величину $\Delta/K(t)$, а вероятности всех других действий повышаются на величину $\Delta/(r - K(t))$, где $\Delta=1/(rN)$ обозначает разрешающий шаг, а $N$ — разрешающий параметр. Было доказано, что ДОАП обладает свойством усреднения и монотонности, и, таким образом, является $\varepsilon$-оптимальным.

Стохастический оценочный алгоритм обучения (СОАО)

СОАО принадлежит к классу АО, и был предложен Василакосом и Пападимитриу. Еще с того момента он используется для решения задач в области компьютернывх сетей. Обладает эргодической схемой, которая способна конвергировать к оптимальному действию вне зависимости от распределения начального состояния.

Как и до этого, пусть $A = \{\alpha_1, \alpha_2,..., \alpha_r\}$ обозначает множество действий, а $B = \{0, 1\}$ обозначает множество ответов, которые могут быть получены от среды, где $\beta(t)$ обозначает обратную связь от среды в соответствии с выбранным действием $\alpha(t)$ в момент времени $t$. Пусть вероятность выбора действия в $k$-ый момент времени будет $p_k(t)$. СОАО корректирует расчетные характеристики среды как вектор $E(t)$, который может быть обозначен как $E(t) = <D(t), M(t), U(t)>$, он объясняется ниже.

$D(t) = \{d_1(t), d_2(t),..., d_r(t)\}$ обозначает вектор показателей поощрения, где

$$d_k(t) = \sum_{i=1}^{W} \beta_k(t)$$

В формуле значение справа обозначает общее количество поощрений, полученных ОА в размере окна, обозначающем последние $W$ раз, когда алгоритм выбрал определенное действие $\alpha_k$. $W$ обозначает окно обучения.

Второй параметр $E(t)$ называется вектором старости, а обозначается как $M(t) = \{m_1(t), m_2(t),..., m_r(t)\}$, где $m_k(t)$ обозначает время, прошедшее (считается за количество итераций) с последнего момента, когда было выбрано действие $\alpha_k(t)$. Последний параметр $U(t)$ называется стохастическим вектором оценивания, и

обозначается $U(t) = \{u_1(t), u_2(t),..., u_r(t)\}$, где стохастическое оценивание $u_i(t)$ действия $\alpha_i$ высчитывается по формуле $u_i(t) = d_i(t) + N[0, \sigma_i^2(t)]$, где $N[0, \sigma_i^2(t)]$ обозначает случайное число, выбранное при нормальном распределении со значением 0 и стандартным отклонением $\sigma_i(t)=\min\{\sigma_{max}, a\ m_i\ (t)\}$, $a$ — параметр, обозначающий скорость, с которой стохастические расчеты становятся независимыми, а $\sigma_{max}$ обозначает максимально возможную норму погрешности, которая может быть у стохастических расчетов. В симметрично распределенных шумных стохастических средах, СОАО проявляется как оптимальный и находит применение в сетях банкоматов с маршрутизацией [12.18, 43].

### 3.6 Модель обучающегося автомата и оценка безопасности узла

Модель автомата может быть описана пятеркой $< Q, A, B, G, T >$, где $Q$ обозначает различные состояния автомата, $A$ – множество действий, $B$ – множество ответов, которые ОА получает от среды, $G$ – функция выхода, а $T$ – корректирующая схема вероятности действия $T : [0, 1]^r \times A \times B \rightarrow [0, 1]^r$ так, чтобы $P(t + 1) = T[P(t), \alpha(t), \beta(t)]$, где $P(t)$ – это вектор вероятности действия

Выбранные параметры автомата:

— стохастический;

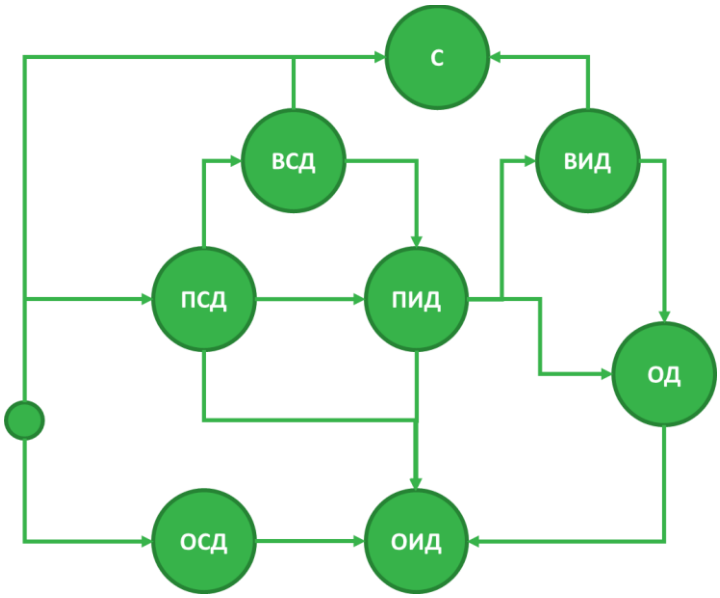
— с переменной структурой;

— Р-модель;

— комбинированная модель обучения $L_{П-\varepsilon Ш}$ (a<<b) и $L_{ПН}$ (b = 0)

*a*–параметрпоощрения *b* – параметр штрафа.

Схема автомата представлена на рисунке 2.




Обозначения

— ПСД/ОСД – прием/отправка служебных данных;

— ПИД/ОИД – прием/отправка информационных данных;

— ВИД/ВСД – верификация информационных/служебных данных;

— ОД – обработка данных;

— RST – сброс соединения.

Использование обучающегося автомата для описания процесса взаимодействия узла с его соседом позволит оценить безопасность этого взаимодействия, основываясь на целесообразности поведения автомата. Иными словами, взаимодействие с узлом будет безопасным, если автомат, которым описывается поведение данного узла, является целесообразным. Таким образом, можно сформулировать теорему о необходимом условии нахождения узла в безопасном для взаимодействия с ним состоянии.

**Теорема** о необходимом условии нахождения узла в безопасном для взаимодействия с ним состоянии

Узел является безопасным для взаимодействия, если математическое ожидание штрафа автомата, которым описывается поведение узла, удовлетворяет условию $M < \frac{1}{r}\sum_{i=1}^{r} c_i$ , где $c_i$ – штраф за определенное действие автомата.

Иными словами, автомат должен обладать целесообразным поведением, что позволяет использовать его для выстраивания процесса взаимодействия с узлом

**Доказательство:** Доказательство теоремы будет основано на методе от противного и использовании того факта, что составленная модель обучающегося автомата является эргодической.

Эргодичность обучающегося автомата следует из структуры ее модели.

Построенную в п.п. модель обучающегося автомата с точки зрения вероятностных переходов можно представить в виде простейших автоматов с двумя состояниями и двумя действиями.

Предположим, что автомат не обладает и каждый переход из одного состояния в другое равновероятный. Автомат работает в среде C = C (a1 a2). В таком случае в соответствии с формулами в предыдущем разделе математическое ожидание штрафа такого автомата будет равно $\frac{1}{r}\sum_{i=1}^{r} c_i$.

Без потери общности можно говорить о таком же равенстве для целесообразного автомата размерности n, обладающего свойством эргодичности.

Обновление веростностей автомата описывается следующими системами уравнений

$$P_j(n+1) = \begin{cases} P_j(n) + a\left(1 - P_j(n)\right), \ j = i \\ P_j(n) - aP_j(n), \ \ \forall j, \ j \neq i \end{cases}$$

$$P_j(n+1) = \begin{cases} (1-b)P_j(n), \ \ j = i \\ \dfrac{b}{r-1} + (1-b)P_j(n), \ \forall j, \ j \neq i \end{cases}$$

Первое уравнение обновляет вероятность при выигрыше. Второе уравнение при штрафе за действие.

### 3.7 Выводы

Проанализированы адаптивные системы, используемые для управления информационно техническими среджствами. Выбран необходимсй тип адаптации. В качестве математического аппарата используется концепция обучающегося автомата.

Проанализирован принцип работы, выявлены разновидности, классифицированы по разным критериям. Выбрана конфигурация автомата для

описания принципов взаимодействия узлов беспроводных сенсорных сетей друг с другом Построена конечно автоматная модель. Сформулирована теорема о необходимом условии для взаимодействия с узлом беспроводной сенсорной сети.

С использованием предложенной модели программа принятия решений функционирует в случайной среде и корректирует свою тактику для выбора оптимальных действий на основе полученного ответа.

# 4 РАЗРАБОТКА СТЕНДА И ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА ЭФФЕКТИВНОСТИ

В рамках раздела приводится разработка стенда для экспериментальной оценки эффекnивности предложенного метода защиты. Для определения оптимального средства для имитации БСС были рассмотрены наиболее популярные из подходящих симуляторы. При анализе средств эмуляции, подходящих для БСС сетей, использовались электронные ресурсы [44, 45, 46].

## 4.1 Сравнительный анализ средств имитационного моделирования беспроводных сенсорных сетей

Средство INETMANET для эмуляции БСС не рассматривалось, т. к. согласно информации с репозитория проекта [38], проект более не поддерживается. Автор не рекомендует использовать реализованные MANET протоколы для INET.

**OMNeT++**

OMNeT++ – объектно-ориентированный модульный сетевой симулятор. OMNeT++ бесплатен для академических целей. Он был разработан Андрасом Варгой из Технического университета Будапешта с некоторыми непостоянными участниками. Симулятор был лицензирован по собственной академической общественной лицензии. Он может использоваться в областях, таких как моделирование сетей связи, сетей массового обслуживания, мультипроцессоров и других распределенных систем, моделирование протоколов, проверка аппаратных архитектур и оценка производительности сложной программной системы. OMNeT++ предоставляет компонентную архитектуру для моделей. Компоненты программируются на C ++, а затем собираются в более крупные компоненты с использованием языка высокого уровня, называемого Network Description Language (NED).

Достоинства:

— развитый GUI;

— возможность построения распределенной системы вычислений;

— имеет много различных библиотек.

Недостатки:

— плохая документированность;

— медленная работа в реальном времени;

— предоставляет мало параметров, характеризующих эффективность протоколов/моделей симуляции.

GloMoSim

GloMoSim (Global Mobile Information System Simulator) – симулятор, нацеленный на параллельную работу. Он обеспечивает масштабируемую среду моделирования для беспроводных сетей. Он разработан с использованием фреймворка PARSEC, предназначенного для упрощения параллельной работы в симуляторах. GloMoSim – это комбинация имитатора последовательных и параллельных дискретных событий. Он поставляется с богатым набором моделей. Этот инструмент моделирования может масштабироваться до тысяч узлов. Модульная конструкция GloMoSim позволяет легко расширять и изменять основные функции. Инструменты анализа и визуализации являются базовыми, но их достаточно для общих исследований.

Достоинства:

— поддержка большого количества устройств;

— поддержка параллельных вычислений;

— поддержка асинхронной работы виртуальных устройств.

Недостатки:

— не поддерживает сенсоры;

— плохая документация;

— поддержка продукта прекращена много лет назад;

— необходима отдельная установка фреймворка PARSEC.

**NS-3**

NS-3 – сетевой симулятор с дискретными событиями. В NS-3 имитационные модели реализованы на C ++. NS-3 экспортирует свой API в Python. NS-3 не является обновленной версией NS-2. NS-3 предназначен в основном для интернет-систем, в первую очередь предназначенных для исследовательских и образовательных целей. NS-3 является свободным программным обеспечением и распространяется по лицензии GNU GPLv2. Программное обеспечение NS-3 поощряет разработку имитационных моделей, достаточно реалистичных, чтобы симулятор NS-3 можно было использовать в качестве эмулятора сети реального времени, связанного с реальной окружающей средой.

Достоинства:

— расширяемый/гибкий;

— быстрая скорость работы;

— эффективный по памяти;

— эффективен по времени вычислений.

Недостатки:

— количество встроенных моделей ограниченно;

— ограниченная поддержка визуализации.

**JSim**

JSim – система моделирования на основе Java для построения количественных числовых моделей и их анализа по отношению к экспериментальным справочным данным. Основное внимание JSim уделяет физиологии и биомедицине, однако его вычислительный механизм является довольно общим и применимым к широкому кругу научных областей. Модели JSim могут смешивать дифференциальные уравнения, неявные уравнения, интегралы, суммирования, дискретные события и процедурный код в зависимости от ситуации. Компилятор модели JSim может автоматически вставлять коэффициенты преобразования для совместимых физических единиц, а также обнаруживать и отклонять несбалансированные уравнения.

Достоинства:

— гибкость;

— есть библиотека с GUI.

Недостатки:

— проблемы с производительностью при генерации больших сетей;

— ограниченный функционал.

**MannaSim**

MannaSim – эмулятор WSN сетей. Данный эмулятор разработан на базе NS-2. Цель MannaSim – разработать детальную среду моделирования, которая может точно моделировать различные сенсорные узлы и приложения, обеспечивая при этом универсальный испытательный стенд для алгоритмов и протоколов. MannaSim позволяет настраивать топологию, способ передачи информации, используемые протоколы. Данный фреймворк также учитывает расход энергии устройствами.

Достоинства:

— учет расходы энергии устройствами;

— большая коллекция реализованных моделей.

Недостатки:

— плохая документированность;

— плохая поддержка GUI;

— плохая поддержка отладки;

— построен на базе устаревшего симулятора;

— использует много памяти.

**MobiWan**

MobiWan является расширением NS-2, добавляющим возможность работы с IPv6 сетями. Задачей MobiWan является симуляция мобильных IPv6 сетей. Для работы с фреймворком используются языки C++ и Octl.

Достоинства:

— легкие конфигурация топологий и манипулирование;

— симуляция передвижения объектов сети;

— поддержка IPv6.

Недостатки:

— плохая документированность;

— плохая поддержка GUI;

— плохая поддержка отладки;

— построен на базе устаревшего симулятора;

— использует много памяти.

## 4.2 Сравнительный анализ симуляторов NS-3 и OMNeT++

После сбора информации о существующих бесплатных симуляторах БСС сетей, был сделан вывод, что лучшими из них являются NS-3 и OMNeT++. Остальные варианты были отклонены по ряду причин: устаревание, плохая документированность, недостаточная функциональность, низкая эффективность использования ресурсов ВМ.

Для определения симулятора, который будет использоваться в НИР, был проведен сравнительный анализ, результаты которого отражены в таблице 1 и таблице 2. При проведении сравнительного анализа использовались статьи и электронные ресурсы [40, 41, 42, 43].

Таблица 1 – Сравнение по различным параметрам

| Критерий | NS-3 | OMNeT++ |
|---|---|---|
| Тип лицензии | Открытая | Открытая (для учебных и исследовательских целей) |
| Наличие GUI | Реализован как сторонний модуль | Есть |
| Сложность обучения работе с симулятором | Высокая | Лёгкая |

| Поддержка ОС | GNU/Linux, FreeBSD, Mac OS | Windows, MAC OS, UNIX |
|---|---|---|
| Тип симулятора | Дискретный | Дискретный |
| Интеграция с реальным миром | Поддерживается | Поддерживается |
| Поддержка параллельных вычислений | Есть | Есть |
| Масштабируемость | Лучше | Хуже |
| Документированность | Отличная | Хорошая |
| Время вычислений | Меньше | Больше |
| Используемое количество памяти | Меньше | Больше |
| Использование CPU | Более активное | Менее активное |
| Последняя дата обновления (на 13 июня 2021) | 9 января 2021 | 20 мая 2020 |
| Возможность внесения изменений в существующие модули | Есть | Есть |

Таблица 2 – Сравнение поддерживаемых протоколов ad-hoc сетей

| Протокол | Задача протокола | Является ли протокол официальным модулем | Реализация для NS-3 | Реализация для OMNeT++ |
|---|---|---|---|---|
| Ad-hoc On-demand Distance Vector (AODV) | Маршрутизация | Да | Реализован | Реализован |
| Optimized Link State Routing (OLSR) | Маршрутизация | Да | Реализован | Реализован |
| Destination-Sequenced Distance-Vector | Маршрутизация | Да | Реализован | Реализован |

| | | | | |
|---|---|---|---|---|
| Routing (DSDV) | | | | |
| Dynamic Source Routing (DSR) | Маршрутизация | Да | Реализован | Реализован |
| Babel | Маршрутизация | Нет | Не реализован | Реализован |
| Fisheye State Routing (FSR) | Маршрутизация | Нет | Не реализован | Реализован |
| Greedy Perimeter Stateless (GPS) | Маршрутизация | Нет | Не реализован | Реализован |
| Dynamic MANET On-demand (AODVv2 или DYMO) | Маршрутизация | Нет | Не реализован | Реализован |

### *4.2.1 Сценарий использования симулятора OMNeT++*

OMNeT++ предоставляет возможность разработки симуляции в IDE, основанной на IDE Eclipse. В качестве примера возьмем проект ieee80221 для OMNeT++, который скачивается при установке фреймворка INET для OMNeT++. Данном примере тестируется модель IEEE 802.11 в режиме ad-hoc.

При разработке симуляции необходимо создать файл настроек с расширением .ini (рисунок 3).

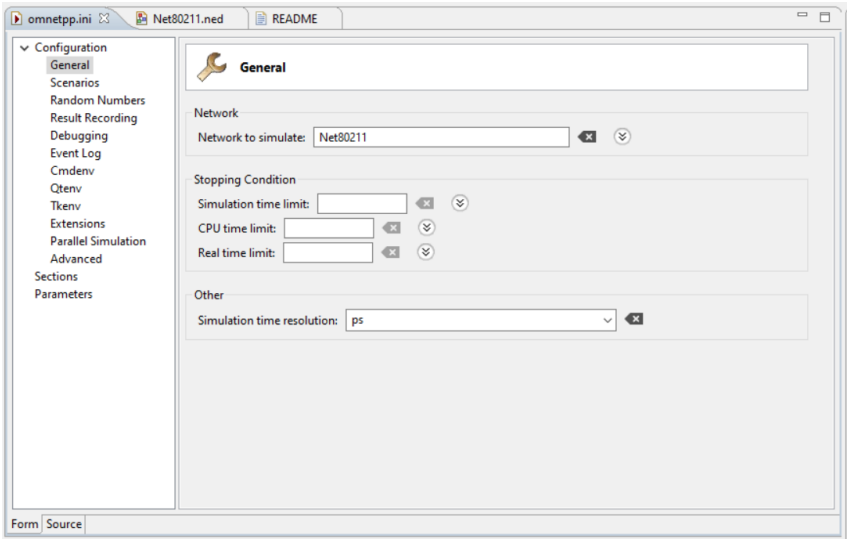



Рисунок 3 – Средство редактирования файла настроек в OMNeT++

Так же необходимо описать топологию сети в файле с расширением .ned. OMNeT++ позволяет задавать топологию с помощью с помощью текстового редактора (рисунок 4) или графического интерфейса (рисунок 5).


Рисунок 4 – .ned файл в текстовом редакторе OMNeT++

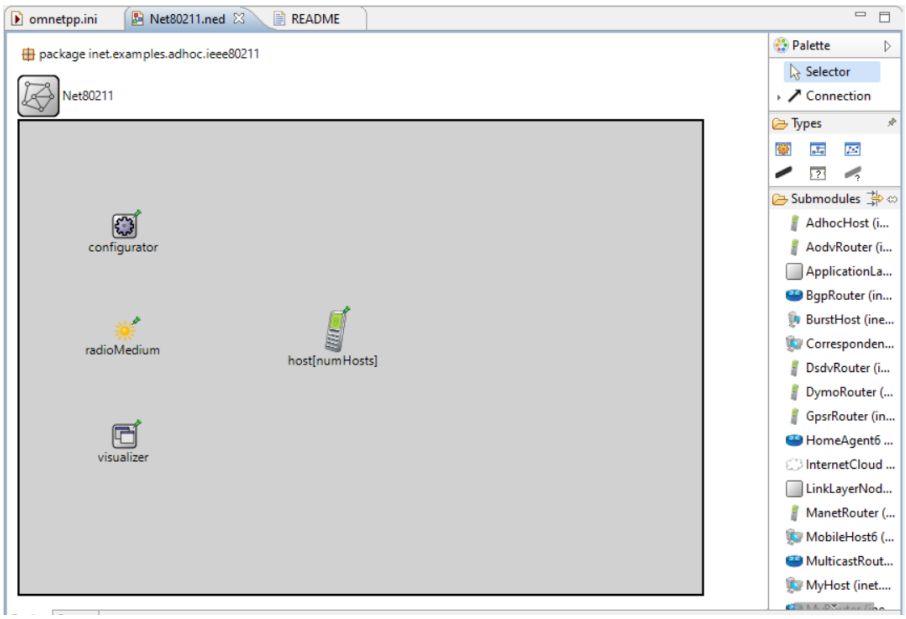



Рисунок 5 – .ned файл в графическом редакторе OMNeT++

В данном примере отсутствуют файлы .h и .cpp C++ кода, т.к. используются уже реализованные модели. Для редактирований C++ кода OMNeT++ предоставляет редактор текстовый редактор.

Запустим симуляцию. Появляется окно симуляции (рисунок 6) с визуализацией узлов сети, анимацией их перемещений и коммуникаций. Так же присутствует окно вывода отладочной и другой информации.

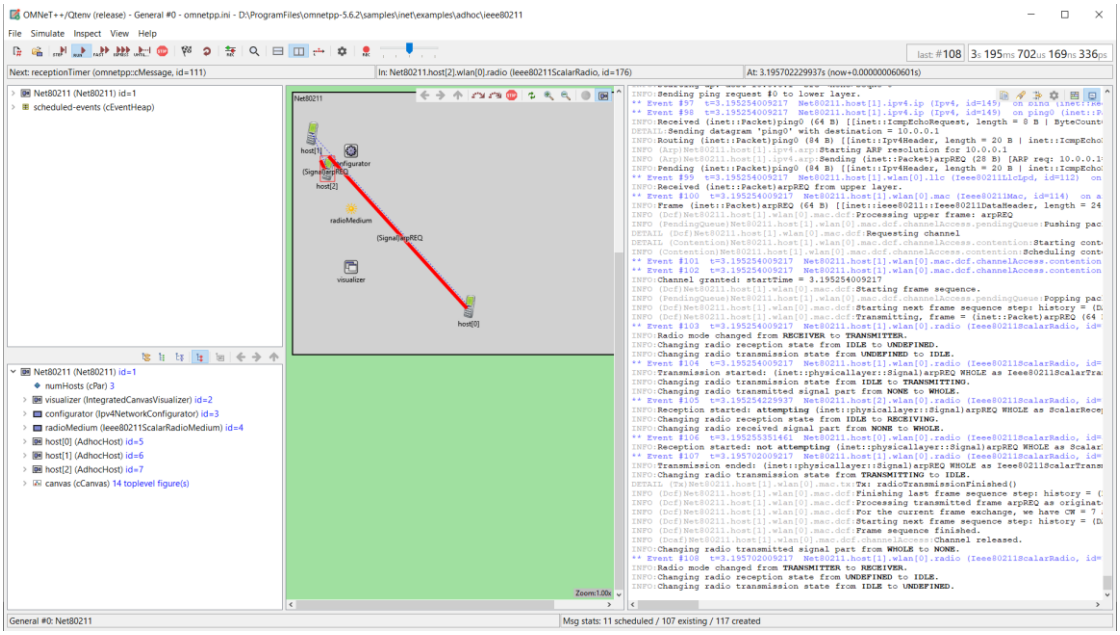


Рисунок 6 – Визуалзация симуляции в OMNeT++

Так же OMNeT++ предоставляет средства сбора информации. В .ned файле задаются параметры, значения которых сохраняются. Формируется база данных симуляции. OMNeT++ предоставляет механизмы для визуализации собранных данных в различном виде. Пример визуализации данных в виде гистограммы на рисунке 7.

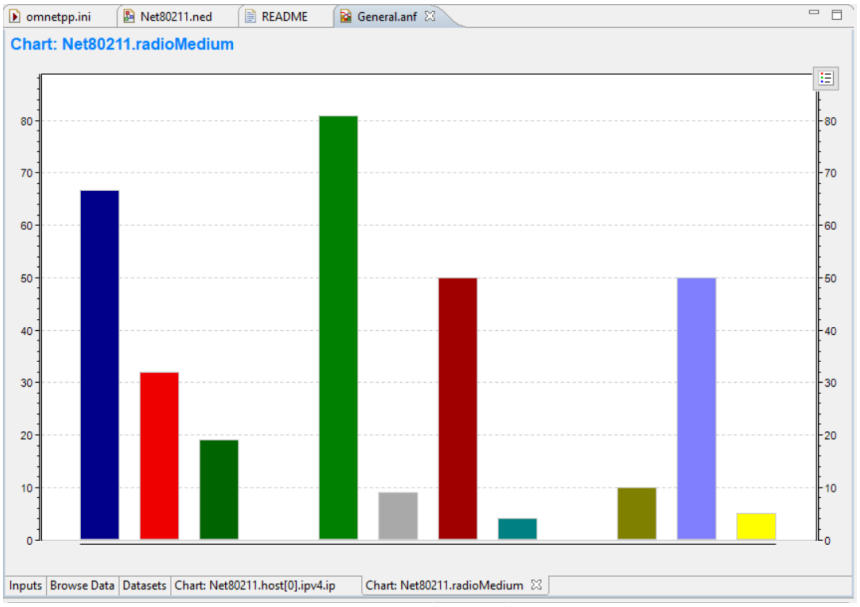



Рисунок 7 – Пример визуализации данных в OMNeT++

### *4.2.2 Сценарий использования NS-3*

NS-3 не предоставляет IDE, редактирование кода симуляции происходит с помощью любого текстового редактора. Сборка и запуск симуляции происходит с помощью системы сборки waf (рисунок 8).

```
user@user-VirtualBox:~/workspace/ns-3-allinone/ns-3.30$ ./waf --run "wifi-si
mple-adhoc"
```

Рисунок 8 – Сборка симуляции с помощью waf

Собираем и запускаем пример простой wifi сети в режиме ad-hoc [44]. Некоторые информационные сообщения выводятся в терминал (рисунок 9). NS-3 не предоставляет встроенных средств анимации.

```
user@user-VirtualBox:~/workspace/ns-3-allinone/ns-3.30$ ./waf --run "wifi-si
mple-adhoc"
Waf: Entering directory `/home/user/workspace/ns-3-allinone/ns-3.30/build'
Waf: Leaving directory `/home/user/workspace/ns-3-allinone/ns-3.30/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.248s)
Testing 1 packets sent with receiver rss -80
Received one packet!
```

Рисунок 9 – Вывод информационных сообщений в терминал во время работы симуляции

В ходе симуляции для каждого узла генерируются .pcap файлы, описывающее трафик узлов. Исследовать .pcap можно с помощью утилиты tcpdump (рисунок 10).

```
user@user-VirtualBox:~/workspace/ns-3-allinone/ns-3.30$ tcpdump -r wifi-simp
le-adhoc-0-0.pcap -nn -ttreading from file wifi-simple-adhoc-0-0.pcap, link-
type IEEE802_11_RADIO (802.11 plus radiotap header)
1.008704 1008704us tsft 1.0 Mb/s 2412 MHz 11b -80dBm signal -94dBm noise use
r@user-VirtualBox:~/workspace/ns-3-allinone/ns-3.30$ tcpdump -r wifi-simple-
adhoc-1-0.pcap -nn -tt
reading from file wifi-simple-adhoc-1-0.pcap, link-type IEEE802_11_RADIO (80
2.11 plus radiotap header)
1.000000 1000000us tsft 1.0 Mb/s 2412 MHz 11b IP 10.1.1.2.49153 > 255.255.25
5.255.80: UDP, length 1000
```

Рисунок 10 – Результат работы tcpdump для сгенерированных .pcap файлов

### *4.2.3 Результаты сравнительного анализа*

Были описаны некоторые из симуляторов WSN сетей (NS-3, OMNeT++, GloMoSim, JSim, MannaSim, MobiWan), их качества, предлагаемые возможности и инструменты создания моделей. Обхват всех возможных симуляторов представляется невозможным, поэтому было принято решение исследовать только наиболее популярные из них. Все исследованные симуляторы имеют определенные достоинства и недостатки. Они имеют разные характеристики и особенности. Разновидности БСС сетей и протоколы, поддерживаемые каждым симулятором, различны. Выбор конкретного симулятора зависит от задачи исследования и возможностей симулятора. Некоторые из симуляторов нацелены на работу только с определенным типом WSN сетей. Однако, наиболее широкий спектр общих возможностей предлагают ns-3 и OMNeT++. Их главными преимуществами, по

сравнению с другими симуляторами, являются: возможность бесплатного использования в академических целях, хорошая документация, большой список готовых к использованию моделей и активная поддержка симуляторов разработчиками.

## 4.3 Разработка моделей атак в средстве моделирования работы беспроводной сенсорной сети

Важной частью моделирования БСС является моделирование атак. Для разработки моделей атак был выбран симулятор ns-3.

Были реализованы две атаки: hello флуд и черная дыра для протокола AODV. В ходе исследования работы реализованных моделей были выявлены недочеты системы учета энергии симулятора, предложены варианты исправления.

### Реализация атаки hello флуд для протокола AODV

Была создана топология из 5 узлов (рисунок 11). Узел с номером 5 является вредоносным, на рисунке также изображена зона распространения его сигнала.

Для узлов были созданы источники энергии с запасом энергии в 1000 Джоулей. Для вредоносного узла был установлен период отправки hello пакета каждые 20 миллисекунд, вместо стандартного периода в одну секунду.

Для того, чтобы проверить, была ли осуществлена hello флуд атака, проверим сгенерированные .pcap файлы. Откроем .pcap файлы с помощью программы Wireshark. На рисунке 12 видно, что узел отправляет hello пакеты с высокой частотой. Также можно заметить общее число обработанных на

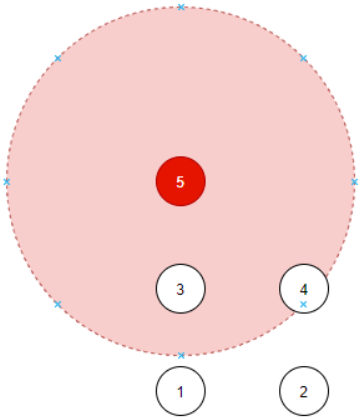



Рисунок 11 - Топология сети

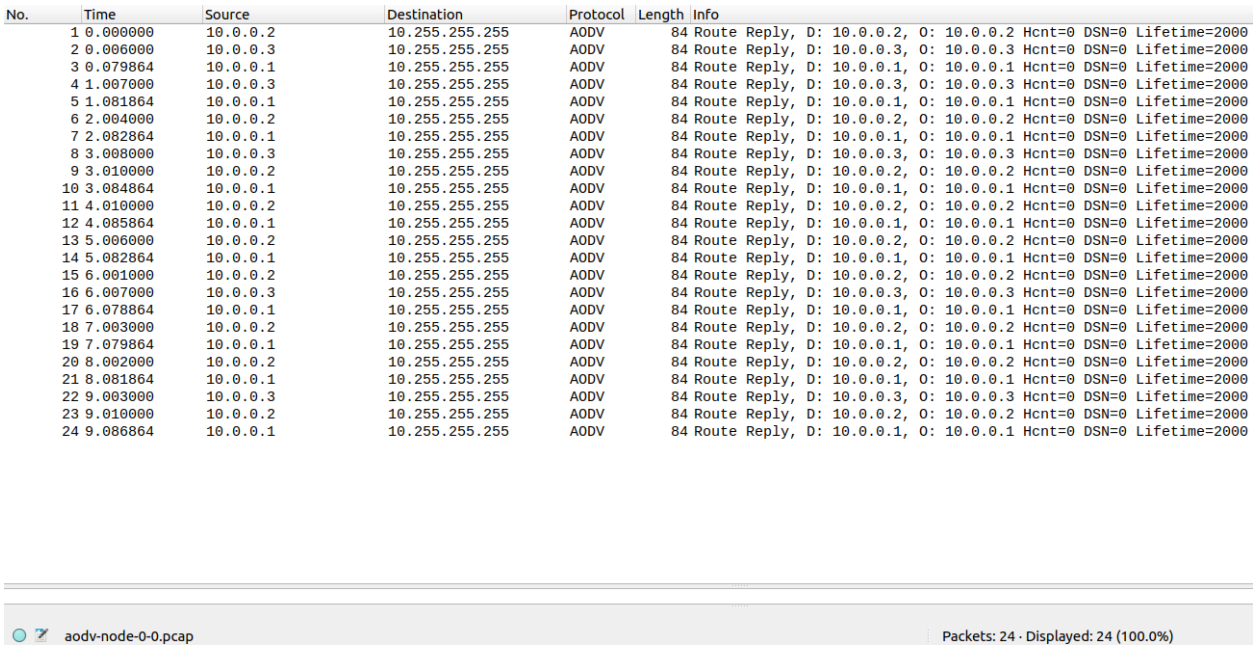
| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 4 | 0.073864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 5 | 0.086864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 6 | 0.106864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 7 | 0.134864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 8 | 0.147864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 9 | 0.173864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 10 | 0.188864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 11 | 0.215864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 12 | 0.234864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 13 | 0.252864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 14 | 0.273864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 15 | 0.288864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 16 | 0.311864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 17 | 0.331864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 18 | 0.351864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 19 | 0.365864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 20 | 0.392864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 21 | 0.415864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 22 | 0.435864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 23 | 0.455864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 24 | 0.470864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 25 | 0.490864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 26 | 0.512864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 27 | 0.527864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 28 | 0.551864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 29 | 0.574864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 30 | 0.590864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 31 | 0.608864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 32 | 0.629864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 33 | 0.651864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 34 | 0.672864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 35 | 0.693864 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |

Frame 49: 84 bytes on wire (672 bits), 84 bytes captured (672 bits)

aodv-malicious-node-0-0.pcap — Packets: 507 · Displayed: 507 (100.0%)

Рисунок 12 – Содержимое .pcap файла вредоносного узла

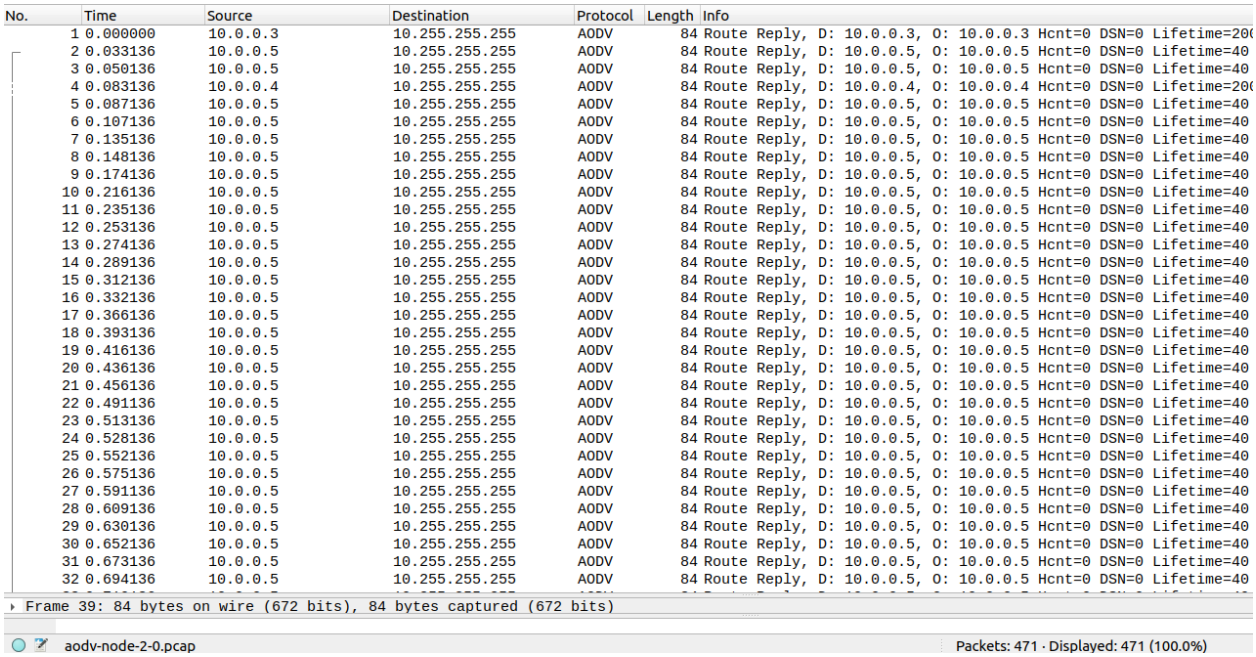
На узле пакетов: 507. На рисунке 13 отображено содержимое .pcap файла для узла вне зоны поражения. Количество обработанных пакетов равно 24. На рисунке 14 отображено содержимое .pcap файла для узла в зоне поражения.



| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1 | 0.000000 | 10.0.0.2 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.2, O: 10.0.0.2 Hcnt=0 DSN=0 Lifetime=2000 |
| 2 | 0.006000 | 10.0.0.3 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.3, O: 10.0.0.3 Hcnt=0 DSN=0 Lifetime=2000 |
| 3 | 0.079864 | 10.0.0.1 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.1, O: 10.0.0.1 Hcnt=0 DSN=0 Lifetime=2000 |
| 4 | 1.007000 | 10.0.0.3 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.3, O: 10.0.0.3 Hcnt=0 DSN=0 Lifetime=2000 |
| 5 | 1.081864 | 10.0.0.1 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.1, O: 10.0.0.1 Hcnt=0 DSN=0 Lifetime=2000 |
| 6 | 2.004000 | 10.0.0.2 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.2, O: 10.0.0.2 Hcnt=0 DSN=0 Lifetime=2000 |
| 7 | 2.082864 | 10.0.0.1 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.1, O: 10.0.0.1 Hcnt=0 DSN=0 Lifetime=2000 |
| 8 | 3.008000 | 10.0.0.3 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.3, O: 10.0.0.3 Hcnt=0 DSN=0 Lifetime=2000 |
| 9 | 3.010000 | 10.0.0.2 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.2, O: 10.0.0.2 Hcnt=0 DSN=0 Lifetime=2000 |
| 10 | 3.084864 | 10.0.0.1 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.1, O: 10.0.0.1 Hcnt=0 DSN=0 Lifetime=2000 |
| 11 | 4.010000 | 10.0.0.2 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.2, O: 10.0.0.2 Hcnt=0 DSN=0 Lifetime=2000 |
| 12 | 4.085864 | 10.0.0.1 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.1, O: 10.0.0.1 Hcnt=0 DSN=0 Lifetime=2000 |
| 13 | 5.006000 | 10.0.0.2 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.2, O: 10.0.0.2 Hcnt=0 DSN=0 Lifetime=2000 |
| 14 | 5.082864 | 10.0.0.1 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.1, O: 10.0.0.1 Hcnt=0 DSN=0 Lifetime=2000 |
| 15 | 6.001000 | 10.0.0.2 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.2, O: 10.0.0.2 Hcnt=0 DSN=0 Lifetime=2000 |
| 16 | 6.007000 | 10.0.0.3 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.3, O: 10.0.0.3 Hcnt=0 DSN=0 Lifetime=2000 |
| 17 | 6.078864 | 10.0.0.1 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.1, O: 10.0.0.1 Hcnt=0 DSN=0 Lifetime=2000 |
| 18 | 7.003000 | 10.0.0.2 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.2, O: 10.0.0.2 Hcnt=0 DSN=0 Lifetime=2000 |
| 19 | 7.079864 | 10.0.0.1 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.1, O: 10.0.0.1 Hcnt=0 DSN=0 Lifetime=2000 |
| 20 | 8.002000 | 10.0.0.2 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.2, O: 10.0.0.2 Hcnt=0 DSN=0 Lifetime=2000 |
| 21 | 8.081864 | 10.0.0.1 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.1, O: 10.0.0.1 Hcnt=0 DSN=0 Lifetime=2000 |
| 22 | 9.003000 | 10.0.0.3 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.3, O: 10.0.0.3 Hcnt=0 DSN=0 Lifetime=2000 |
| 23 | 9.010000 | 10.0.0.2 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.2, O: 10.0.0.2 Hcnt=0 DSN=0 Lifetime=2000 |
| 24 | 9.086864 | 10.0.0.1 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.1, O: 10.0.0.1 Hcnt=0 DSN=0 Lifetime=2000 |

aodv-node-0-0.pcap — Packets: 24 · Displayed: 24 (100.0%)

Рисунок 13 – Содержимое .pcap файла узла вне зоны поражения



| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 1 | 0.000000 | 10.0.0.3 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.3, O: 10.0.0.3 Hcnt=0 DSN=0 Lifetime=200 |
| 2 | 0.033136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 3 | 0.050136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 4 | 0.083136 | 10.0.0.4 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.4, O: 10.0.0.4 Hcnt=0 DSN=0 Lifetime=200 |
| 5 | 0.087136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 6 | 0.107136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 7 | 0.135136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 8 | 0.148136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 9 | 0.174136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 10 | 0.216136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 11 | 0.235136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 12 | 0.253136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 13 | 0.274136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 14 | 0.289136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 15 | 0.312136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 16 | 0.332136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 17 | 0.366136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 18 | 0.393136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 19 | 0.416136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 20 | 0.436136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 21 | 0.456136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 22 | 0.491136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 23 | 0.513136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 24 | 0.528136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 25 | 0.552136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 26 | 0.575136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 27 | 0.591136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 28 | 0.609136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 29 | 0.630136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 30 | 0.652136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 31 | 0.673136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |
| 32 | 0.694136 | 10.0.0.5 | 10.255.255.255 | AODV | 84 | Route Reply, D: 10.0.0.5, O: 10.0.0.5 Hcnt=0 DSN=0 Lifetime=40 |

▸ Frame 39: 84 bytes on wire (672 bits), 84 bytes captured (672 bits)

aodv-node-2-0.pcap — Packets: 471 · Displayed: 471 (100.0%)

Рисунок 14 – Содержимое .pcap файла узла в зоне поражения

Число обработанных узлом пакетов равно 471. Видим, что большое число пакетов – это широковещательные hello пакеты от вредоносного узла.

На основании анализа траффика можно сделать вывод, что симуляция hello флуд атаки работает корректно. Исследуем воздействие hello флуд атаки на количество потребляемой узлами энергии.

Перед исследованием также уменьшим период отправки hello пакетов вредоносным узлом до 1 миллисекунды. На рисунке 15 видно, что расход энергии в узлах, подверженных атаке, увеличился незначительно. Из описания моделей [45], связанных с системой потребления энергии, можно понять, что в ns-3 не ведется учет энергии для вычислительных процессов. Из описания класса WifiRadioEnergyModel [46] следует, что устройство Wi-Fi потребляет энергию равномерно во времени и количество потребленной энергии зависит только от количества переключений между режимами и от самого режима: получения данных, отправки данных, сон, бездействие. Несмотря на то, что количество обрабатываемых пакетов влияет на число переключений между состояниями, это влияние недостаточно для значительного увеличения потребления энергии.

```
Energy consumption of common nodes:
End of simulation (10s) Total energy consumed by radio = 8.19059J
End of simulation (10s) Total energy consumed by radio = 8.19067J
End of simulation (10s) Total energy consumed by radio = 8.35275J
End of simulation (10s) Total energy consumed by radio = 8.35258J

Energy consumption of malicious nodes:
End of simulation (10s) Total energy consumed by radio = 8.62433J
```

Рисунок 15 – Расход энергии в узлах

На текущий момент функциональных возможностей инструментов симулятора ns-3 недостаточно для получения корректных результатов моделей атак, связанных с энергопотреблением.

**Реализация атаки черная дыра для протокола AODV**

Была создана топология из 5 узлов (рисунок 16). Узел с номером 5 является вредоносным, на рисунке также изображена зона распространения его сигнала. В созданной симуляции узел 4 отправляет ICMP эхо-запрос узлу 1. При запуске атаки узел 5 будет сообщать узлу 4, что он имеет кратчайший путь до узла 1. При получении пакета от узла 4 узел 5 будет отбрасывать пакет.

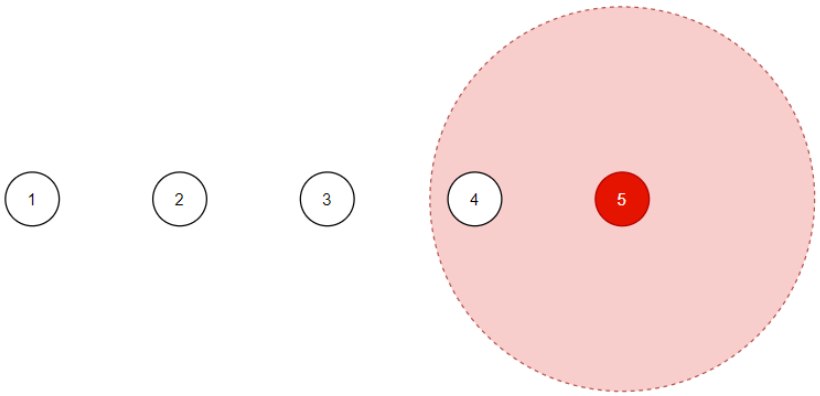



Рисунок 16 – Топология сети

Сначала проверим работу симуляции без атаки. Для этого посмотрим, получает ли узел 1 эхо-запрос. На рисунке 17 видно, что узел 1 получает ICMP эхо-запрос от узла 4, следовательно симуляция работает корректно.

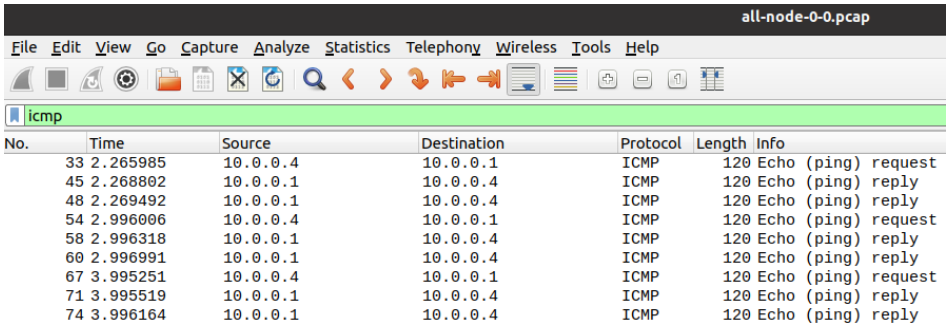



Рисунок 17 – Отфильтрованное содержимое .pcap файла для узла 1

Запустим атаку и снова проверим траффик на узле 1. На рисунке 18 видно, что атака сработала корректно.

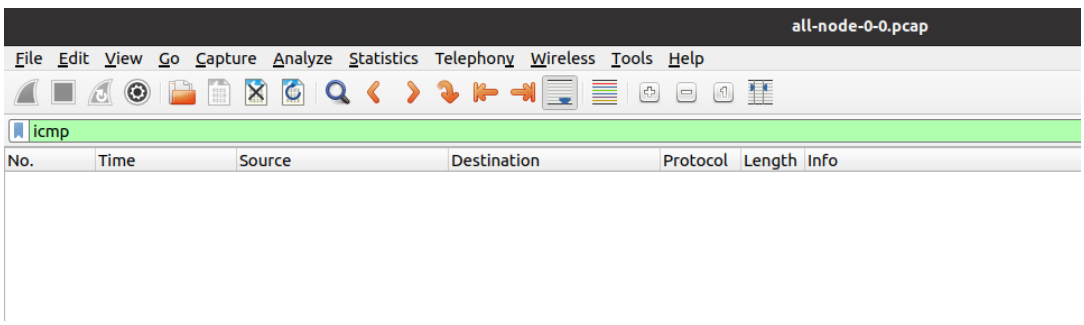


Рисунок 18 – Отфильтрованное содержимое. pcap файла для узла 1

На рисунке 19 отображена блок-схема работы алгоритма черной дыры для протокола AODV.

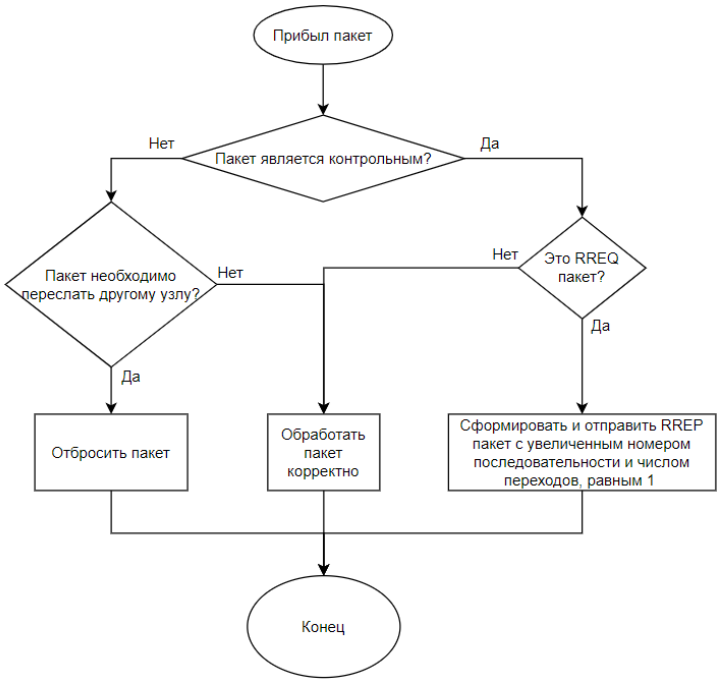



Рисунок 19 – Блок-схема алгоритма работы черной дыры для протокола AODV

В исходный код симулятора были внесены изменения в соответствии с блок-схемой для реализации атаки.

### 4.4 Улучшение модели Energy Framework симулятора ns-3

В результате реализации hello флуд атаки возникла следующая проблема: невозможно корректно подсчитать потребление энергии узлом. В ходе hello флуд атаки наибольшая часть потребления энергии уходит на вычисления. В ns-3 не реализована модель потребления энергии, позволяющая учесть затраты энергии на вычисления, связанные с протоколом маршрутизации AODV.

В качестве улучшения Energy Framework была разработана модель потребления энергии для вычислений. Разработанная модель была интегрирована в реализацию протокола AODV.

Далее описаны подробности реализации. Были реализованы методы интерфейса DeviceEnergyModel. Архитектура симулятора ns-3 не предусматривает симуляцию работы процессоров, поэтому разработанная модель имеет ряд ограничений и недостатков:

— количество потребляемой узлами энергии зависит от времени выполнения симуляции на физическом устройстве пользователя;

— внедрение разработанной модели в другие протоколы требует модификации исходного кода протокола.

Первая проблема возникает вследствие способа замера потребляемой энергии. При исполнении функций протокола AODV происходит замер времени их работы, после чего полученное время умножается на ток потребления (определяется пользователем) и подаваемое напряжение:

$$\frac{(duration * m\_calculationCurrentA * supplyVoltage)}{10^7}$$

Вторая проблема возникает, по причине отсутствия в реализациях протоколов обратных вызовов или других механизмов, позволяющих осуществить замеры потребления энергии.

Модель потребления энергии имеет 2 состояния: бездействия и вычислений. Расчет потребляемой энергии ведется только для состояния вычислений. Для перевода модели в состояние вычислений ее необходимо уведомить вызовом метода NotifyCalculating, для перевода в состояние бездействия вызывается метод NotifyIdle.

Исходный код модели содержится в файлах aodv-energy-model.h, aodv-energy-model.cc. Код хэлпера, осуществляющего установку модели потребления энергии в протокол, а также модернизированный код протокола находятся в файлах aodv-energy-model-helper.h, aodv-energy-model-helper.cc, aodv-routing-protocol.h, aodv-routing-protocol.cc.

Модернизация протокола маршрутизации заключается в добавлении в начало и конец каждой функции обратных вызовов, уведомляющих о начале и завершении вычисления функции.

## 4.5 Реализация средства защиты для симулятора и оценки эффективности

Наиболее важным вопросом, с точки зрения информационной безопасности, является возможность реализации средств защиты, используя средства симулятора. Для ответа на этот вопрос разработаем средство защиты и экспериментальным путем выясним возможности, предоставляемые симулятором.

**Защита на основе конечно-автоматной модели**

В качестве средства защиты для БСС сетей предлагается реализовать 2 конечных автомата, содержащих логику для обнаружения атак. Далее опишем конечные автоматы в контексте протокола маршрутизации AODV.

Конечный автомат 1 следит за тем, пересылает ли соседний узел RREQ пакеты текущего узла. В случае, если соседний узел не пересылает RREQ пакеты и не отправляет RREP пакет, то предполагаем, что соседний узел реализует атаку червоточина.

Конечный автомат 1 определяет, не присылает ли нам соседний узел 1 RREP пакеты с путем до другого нашего соседнего узла 2 и информацией о том, что путь через узел 1 ближайший. Если не присылает, то пытаемся отправить эхо-запрос узлу 2 в соответствии со старой таблицей маршрутизации. В случае, если мы получаем ответ на эхо-запрос, можно говорить о попытке изменения таблицы маршрутизации.

Конечный автомат 2 следит за тем, чтобы соседний узел функционировал согласно протоколу, например, отправляет ли соседний узел hello пакеты. Если не отправляет, то узел можно считать вредоносным, так как стандартная реализация протокола не предполагает скрытого присутствия.

Конечный автомат 2 определяет частоту контрольных пакетов, отправляемых соседним узлом текущему узлу. Если частота hello пакетов превышает пороговое значение, то можно говорить о hello флуд атаке. Если частота RREP пакетов с путем до некоторого конкретного узла превышает пороговое

значение, то можно предположить, что соседний узел пытается изменить таблицу маршрутизации.

Конечный автомат 1 контролирует, пересылает ли соседний узел пакеты, предназначенные другим узлам, для которых соседний узел является частью маршрута.

Конечные автоматы представлены на рисунках 21–22. Конечные автоматы инициализируются для каждого соседа в момент первого его обнаружения. Предполагается, что отдельно взятый узел БСС имеет небольшое количество соседей из-за чего потребление ресурсов для работы конечных автоматов будет низким. Конечные автоматы в отдельно взятый момент времени следят за выполнением не более чем двух атак, что также снижает расход ресурсов.

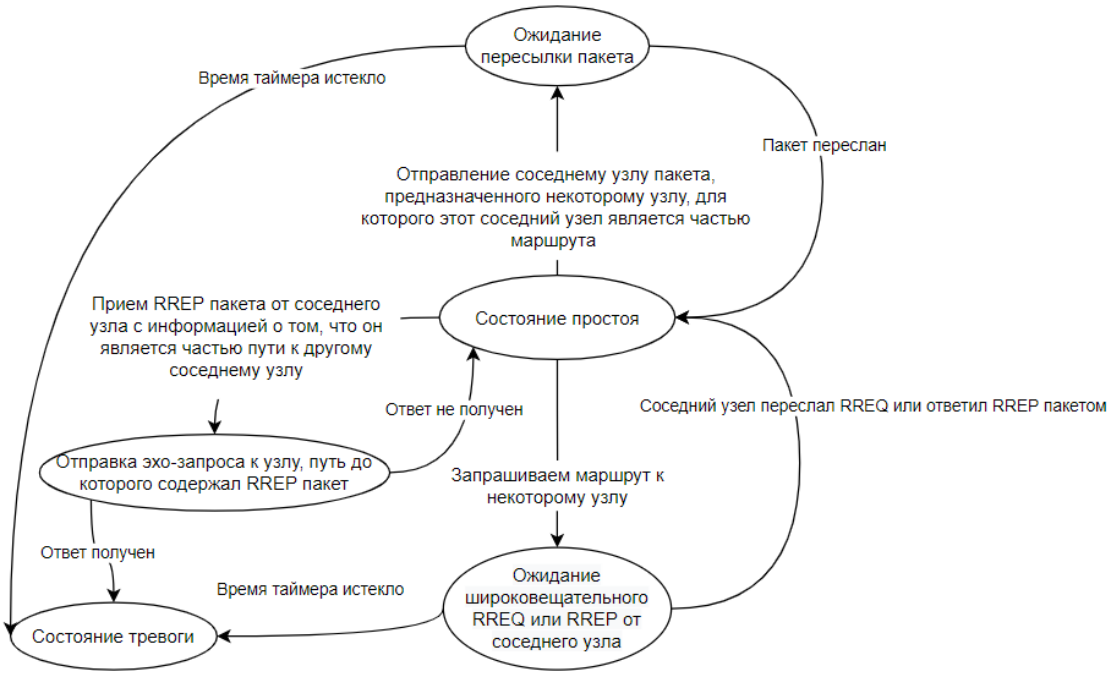



Рисунок 21 – Конечный автомат 1


Рисунок 22 – Конечный автомат 2

**Реализация средства защиты**

Реализация средства защиты предполагает изменение поведения исходных модулей симулятора. NS-3 предлагает несколько вариантов внедрения функциональных возможностей:

— механизм обратных вызовов [47];

— изменение исходного кода симулятора.

Для внедрения конечных автоматов мне было необходимо изменить поведение протокола маршрутизации, по этой причине для внедрения дополнительных функциональных возможностей я изменял исходных код модулей симулятора.

Средство защиты для каждого соседа считает число полученных hello пакетов за заданный период времени (определяется пользователем), если число таких пакетов превышает пороговое значение (определяется пользователем), то сосед помечается как вредоносный.

Для интеграции средства защиты был изменен исходный код протокола маршрутизации AODV. В метод RouteInput (вызывается для обработки каждого входящего пакета) добавлено обращение к средству защиты, для проверки отправителя. Если отправитель вредоносный, то пакет отбрасывается. В метод

ProcessHello (вызывается для обработки входящих hello пакетов) добавлено уведомление средства защиты о входящем hello пакете.

Для симуляции использовалось 5 узлов, топология сети отображена на рисунке 23. Узел 4 осуществляет hello флуд.

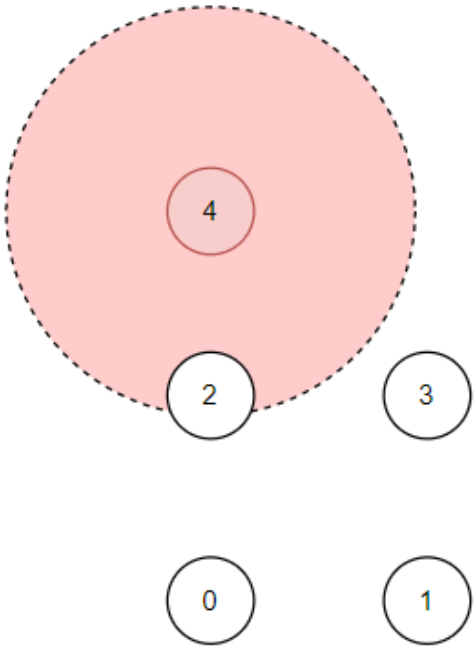



Рисунок 23 – Топология сети

На рисунке 24 отображены результаты эксперимента.

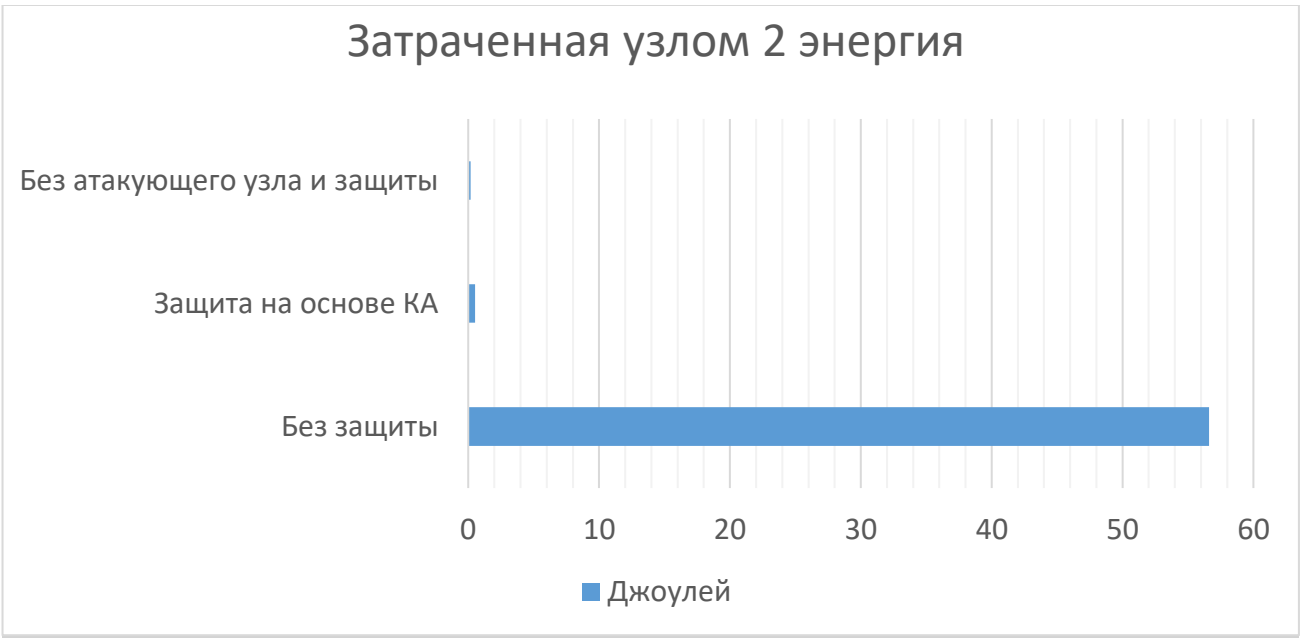



Рисунок 24 – Затраченная узлом 4 энергия

Эксперимент имел продолжительность 5 секунд. эксперименте для узлов были созданы источники энергии с запасом энергии равным 1000 Джоулей. Узел 2 успешно детектировал hello флуд атаку.

Как можно заметить средство защиты снизило затраты энергии при атаке на узел, однако также способствовало повышенному потреблению энергии (в 2-3 раза) в сети без вредоносных узлов.

В ходе эксперимента удалось успешно реализовать модель средства защиты. На основании полученных результатов можно сделать вывод, что архитектура и встроенные средства симулятора NS-3 позволяют успешно модифицировать симулятор с целью внедрения предлагаемого средства защиты.

### 4.6 Выводы

В ходе работы над разделом были рассмотрены принципы работы БСС, возможные топологии таких сетей и характеристики типичных узлов. Особое внимание было уделено угрозам и атакам на БСС сети.

Были проанализированы существующие симуляторы БСС. Среди рассмотренных симуляторов наилучшими в сравнении себя показали OMNeT++ и NS-3.

Для симулятора NS-3 были реализованы модели атак hello флуд и черная дыра. Симулятор ns-3 не позволяет в полной мере тестировать атаки, направленные на увеличение потребляемой энергии. Был предложен способ модернизации симулятора с целью предоставления функциональных возможностей для корректного расчета потребления энергии.

В ходе исследования и анализа существующих средств защиты была выработана и описана модель средства защиты на основе конечных автоматов. Были проведены дополнительные эксперименты, с целью выяснения возможности реализации модели средства защиты с помощью средств симулятора NS-3. На основании полученных результатов можно сделать вывод, что архитектура и встроенные средства симулятора ns-3 позволяют успешно модифицировать симулятор с целью внедрения предлагаемого средства защиты.

# ЗАКЛЮЧЕНИЕ

В ходе научно-исследовательской работы были рассмотрены принципы работы БСС, возможные топологии таких сетей и характеристики типичных узлов. Особое внимание было уделено угрозам и атакам на БСС сети.

В результате проведенных исследований предложен подход для обеспечения безопасности беспроводных сенсорных сетей с помощью адаптивного поведения, позволяющий выявлять вредоносные узлы, сохранить работоспособность сети в условиях совершения атак и наличия в сети неисправных узлов.

Были проанализированы существующие симуляторы БСС. Среди рассмотренных симуляторов наилучшими в сравнении себя показали OMNeT++ и NS-3.

Для симулятора NS-3 были реализованы модели атак hello флуд и черная дыра. Симулятор ns-3 не позволяет в полной мере тестировать атаки, направленные на увеличение потребляемой энергии. Был предложен способ модернизации симулятора с целью предоставления функциональных возможностей для корректного расчета потребления энергии.

Были проведены дополнительные эксперименты, с целью выяснения возможности реализации модели средства защиты с помощью средств симулятора NS-3. На основании полученных результатов можно сделать вывод, что архитектура и встроенные средства симулятора ns-3 позволяют успешно модифицировать симулятор с целью внедрения предлагаемого средства защиты.

В работе получены следующие основные результаты:

1. Исследованы особенности функционирования беспроводных сенсорных сетей, выделены актуальные угрозы безопасности, выявлены показатели поведения и функционирования узлов, позволяющие оценить их безопасность

2. Разработан подход к проверке защищенности узлов беспроводной сенсорной сети на основе интеллектуального анализа показателей функционирования. В качестве аппарата для проверки используются методы машинного обучения (kNN, RF, SVM)

3. Построена конечно-автоматна модель, описывающая взаимодействие узлов и позволяющая узлам изменять принципы взаимодействия со своими соседями

4. Создан метод обеспечения защиты на основе адаптивного поведения узлов с применением обучающегося автомата и анализа показателей поведения и функционирования узлов. Адаптивное поведения узлов достигается за счет изменения вероятностей между состояниями в автомате. Вероятности действий в автомате обновляется на основе полученного сигнала подкрепления (сигнал подкрепления формируется в зависимости от полученных показателей соседних узлов в процессе работы сети).

5. Проведена экспериментальная оценка эффективности разработанного метода защиты путем моделирования его работы в системе симуляции беспроводной сенсорной сети.

# СПИСОК ИСТОЧНИКОВ

1. Akyildiz I. F. et al. Wireless sensor networks: a survey //Computer networks. – 2002. – Т. 38. – №. 4. – С. 393-422.

2. Yick J., Mukherjee B., Ghosal D. Wireless sensor network survey //Computer networks. – 2008. – Т. 52. – №. 12. – С. 2292-2330.

3. Warneke B. et al. Smart dust: Communicating with a cubic-millimeter computer //Computer. – 2001. – Т. 34. – №. 1. – С. 44-51.

4. Scanlon M., Reiff C., Solomon L. Aerostat acoustic payload for transient and helicopter detection //Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense VI. – International Society for Optics and Photonics, 2007. – Т. 6538. – С. 65380H.

5. Ramya C. M., Shanmugaraj M., Prabakaran R. Study on ZigBee technology //2011 3rd International Conference on Electronics Computer Technology. – IEEE, 2011. – Т. 6. – С. 297-301.

6. Suzuki T. et al. Deployment of wireless sensor network using mobile robots to construct an intelligent environment in a multi-robot sensor network. – IntechOpen, 2008.

7. Culman M. et al. Using agrometeorological data to assist irrigation management in oil palm crops: A decision support method and results from crop model simulation //Agricultural water management. – 2019. – Т. 213. – С. 1047-1062.

8. Финогеев А. Г. и др. Оперативный дистанционный мониторинг в системе городского теплоснабжения на основе беспроводных сенсорных сетей //Известия высших учебных заведений. Поволжский регион. Технические науки. – 2010. – №. 3.

9. Тарик А. и др. Последние тенденции в развитии подводной беспроводной сенсорной сети: систематический обзор литературы //Труды Института системного программирования РАН. – 2021. – Т. 33. – №. 1. – С. 97-110.

10. Лямин А. Т., Барабанова Е. А. Применение беспроводных сенсорных сетей в нефтегазовой промышленности //XLV Огарёвские чтения. – 2017. – С. 340-344.

11. Овасапян Т.Д., Иванов Д.В. Подходы к сканированию адресного пространства IPv6 при анализе безопасности узлов сети интернет // Методы и технические средства обеспечения безопасности информации: Материалы 25-ой научно-технической конференции. – СПб: Изд-во Политехн. ун-та, 2016.

12. Lazarescu M. T. Design of a WSN platform for long-term environmental monitoring for IoT applications //IEEE Journal on emerging and selected topics in circuits and systems. – 2013. – Т. 3. – №. 1. – С. 45-54.

13. Awasthi A., Reddy S. R. N. Monitoring for precision agriculture using wireless sensor network-a review //Global Journal of Computer Science and Technology. – 2013.

14. Овасапян Т.Д., Иванов Д.В., Москвин Д. А. Подходы к обнаружению активных сетевых узлов в адресном пространстве IPv6 // Журнал «Проблемы информационной безопасности. Компьютерные системы». – СПб.: Изд-во Политехн. ун-та, 2016. - №4. - С. 68-73.

15. Yick J., Mukherjee B., Ghosal D. Wireless sensor network survey //Computer networks. – 2008. – Т. 52. – №. 12. – С. 2292-2330.

16. Овасапян Т.Д., Иванов Д.В., Москвин Д. А. Исследование методов аудита безопасности сети интернет на основе обнаружения активных узлов IPv6-адресного пространства // Региональная информатика (РИ-2016). Юбилейная XV Санкт-Петербургская международная конференция «Региональная информатика (РИ-2016)»: Материалы конференции. \ СПОИСУ. – СПб, 2016.

17. Овасапян Т.Д., Иванов Д.В. Обеспечение безопасности WSN-сетей на основе модели доверия // Методы и технические средства обеспечения безопасности информации: Материалы 26-ой научно-технической конференции. – СПб: Изд-во Политехн. ун-та, 2017.

18. Atanasov S. An overview of wireless communication technologies used in wireless sensor networks //International Scientific Conference eRA-8. – 2013. – С. 11-18.

19. Овасапян Т.Д., Иванов Д.В. Обеспечение безопасности WSN-сетей на основе модели доверия // Журнал «Проблемы информационной безопасности. Компьютерные системы». – СПб.: Изд-во Политехн. ун-та, 2017. - №4. - С. 64-72.

20. Овасапян Т.Д., Иванов Д.В., Зегжда Д.П. Обеспечение безопасности WSN-сетей на основе модели доверия // Информационная безопасность регионов России (ИБРР-2017). И 74 Юбилейная X Санкт-Петербургская межрегиональная конференция. Санкт-Петербург, 1-3 ноября 2017 г.: Материалы конференции/СПОИСУ. – СПб., 2017.

21. Ivanov D. V., Moskvin D. A., Ovasapyan T. D. Approaches to detecting active network nodes in IPv6 address space //Automatic Control and Computer Sciences. – 2017. – Т. 51. – №. 8. – С. 902-906.

22. Овасапян Т.Д., Иванов Д.В. Применение распределенной модели доверия для обеспечения безопасности беспроводных сенсорных сетей // Неделя науки СПбПУ. Санкт-Петербург, 13-19 ноября 2017г.: Материалы научной конференции с международным участием. – СПб.: Издательство Политехнического университета, 2018.

23. Овасапян Т.Д., Москвин Д.А., Калинин М.О. Применение нейронных сетей для выявления внутренних нарушителей в VANET-сетях // Журнал «Проблемы информационной безопасности. Компьютерные системы». – СПб.: Изд-во Политехн. ун-та, 2018. - №1. - С. 68-73.

24. Овасапян Т.Д., Москвин Д.А. Выявление внутренних нарушителей в VANET-сетях с использованием нейронных сетей // IV Межрегиональная научно-практическая конференция «Перспективные направления развития отечественных информационных технологий». Севастополь, 18-22 сентября 2018 г.

25. Овасапян Т. Д., Москвин Д. А., Иванов Д. В. Использование аппарата нейронных сетей для выявления внутренних нарушителей в VANET-сетях //27-Й НАУЧНО-ТЕХНИЧЕСКОЙ КОНФЕРЕНЦИИ МЕТОДЫ И ТЕХНИЧЕСКИЕ СРЕДСТВА ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ИНФОРМАЦИИ 24-27 СЕНТЯБРЯ 2018 ГОДА. – 2018. – С. 22.

26. Иванов Д. В., Москвин Д. А., Овасапян Т. Д. Подход к обеспечению кибербезопасности VANET-сетей с использованием теории фрактальных графов //27-Й НАУЧНО-ТЕХНИЧЕСКОЙ КОНФЕРЕНЦИИ МЕТОДЫ И ТЕХНИЧЕСКИЕ СРЕДСТВА ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ИНФОРМАЦИИ 24-27 СЕНТЯБРЯ 2018 ГОДА. – 2018. – С. 107.

27. Ovasapyan T. D., Moskvin D. A., Kalinin M. O. Using Neural Networks to Detect Internal Intruders in VANETs //Automatic Control and Computer Sciences. – 2018. – Т. 52. – №. 8. – С. 954-958.

28. Ovasapyan T. D., Ivanov D. V. Security Provision in Wireless Sensor Networks on the Basis of the Trust Model //Automatic Control and Computer Sciences. – 2018. – Т. 52. – №. 8. – С. 1042-1048.

29. Овасапян Т. Д. Применение аппарата нечеткой логики для противодействия атакам внутренних нарушителей в WSN-сетях //Проблемы информационной безопасности. Компьютерные системы. – 2019. – №. 2. – С. 65-72.

30. Овасапян Т. Д., Князев П. В., Москвин Д. А. Применение taint-анализа для исследования безопасности программного обеспечения устройств интернета вещей, построенных на базе архитектуры ARM //Проблемы информационной безопасности. Компьютерные системы. – 2019. – №. 4. – С. 84-91.

31. Овасапян Т. Д., Москвин Д. А., Иванов Д. В. Применение адаптивной системы управления для противодействия атакам внутренних нарушителей в беспроводных сенсорных сетях //28-ая научнотехническая конференция Методы и технические средства обеспечения безопасности информации. – 2019. – С. 40.

32. Ovasapyan T., Moskvin D. Security Provision in WSN on the Basis of the Adaptive Behavior of Nodes //2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4). – IEEE, 2020. – С. 81-85.

33. Овасапян Т. Д., Данилов В. Д., Иванов Д. В. Исследование эффективности применения алгоритмов генерации синтетических данных для увеличения точности выявления сетевых атак на интернет вещей //Методы и технические средства обеспечения безопасности информации. – 2020. – №. 29. – С. 24-25.

34. Мясников А. В., Москвин Д. А., Овасапян Т. Д. Оптимизация тестирования на проникновение с применением методов обучения с подкреплением //Региональная информатика (РИ-2020). XVII Санкт-Петербургская международная конференция «Региональная информатика (РИ-2020)». Санкт-Петербург, 28-30 октября Р32 2020 г.: Материалы конференции. Часть 1.\СПОИСУ.–СПб, 2020.–393 с. – 2020. – С. 165.

35. Овасапян Т. Д., Князев П. В., Москвин Д. А. АВТОМАТИЗИРОВАННЫЙ ПОИСК УЯЗВИМОСТЕЙ В ПРОГРАММНОМ ОБЕСПЕЧЕНИИ АРХИТЕКТУРЫ ARM С ПРИМЕНЕНИЕМ ДИНАМИЧЕСКОГО СИМВОЛЬНОГО ВЫПОЛНЕНИЯ //Проблемы информационной безопасности. Компьютерные системы. – 2020. – №. 3. – С. 103-113.

36. Овасапян Т. Д., Москвин Д. А., Князев П. В. Применение динамического символьного выполнения для автоматизированного поиска уязвимостей в программном обеспечении архитектуры arm //Методы и технические средства обеспечения безопасности информации. – 2020. – №. 29. – С. 74-75.

37. Овасапян Т. Д., Данилов В. Д., Москвин Д. А. ПРИМЕНЕНИЕ МЕТОДОВ ГЕНЕРАЦИИ СИНТЕТИЧЕСКИХ ДАННЫХ В ЗАДАЧАХ ВЫЯВЛЕНИЯ СЕТЕВЫХ АТАК НА УСТРОЙСТВА ИНТЕРНЕТА ВЕЩЕЙ //Проблемы информационной безопасности. Компьютерные системы. – 2020. – №. 4. – С. 26-34.

38. Mizuhara Y., Hayashi A., Suematsu N. Embedding of time series data by using dynamic time warping distances //Systems and computers in Japan. – 2006. – Т. 37. – №. 3. – С. 1-9.

39. Ovasapyan T. D., Knyazev P. V., Moskvin D. A. Application of Taint Analysis to Study the Safety of Software of the Internet of Things Devices Based on the ARM Architecture //Automatic Control and Computer Sciences. – 2020. – Т. 54. – №. 8. – С. 834-840.

40. Ovasapyan T., Moskvin D., Tsvetkov A. Detection of attacks on the Internet of Things based on intelligent analysis of devices functioning indicators //13th International Conference on Security of Information and Networks. – 2020. – С. 1-7.

41. Овасапян Т. Д., Москвин Д. А. ПРИМЕНЕНИЕ ОБУЧАЮЩИХСЯ АВТОМАТОВ ДЛЯ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ WSN-СЕТЕЙ //Методы и технические средства обеспечения безопасности информации. – 2021. – №. 30. – С. 99-100.

42. Овасапян Т. Д., Никулкин В. А., Москвин Д. А. ПРИМЕНЕНИЕ ТЕХНОЛОГИИ HONEYPOT С АДАПТИВНЫМ ПОВЕДЕНИЕМ ДЛЯ СЕТЕЙ ИНТЕРНЕТА ВЕЩЕЙ //Проблемы информационной безопасности. Компьютерные системы. – 2021. – №. 2. – С. 135-144.

43. Sarika S, Pravin A, Vijayakumar A, Selvamani K «Security Issues In Mobile Ad Hoc Networks» 2016 *2nd International Conference on Intelligent Computing, Communication & Convergence (ICCC-2016)*, Paris, pp. 329-335 doi: 10.1016/j.procs.2016.07.363

44. Sweta Dixit, Krishna Kumar Joshi, Neelam Joshi «A Review: Black Hole & Gray Hole Attack in MANET» International Journal of Future Generation Communication and Networking, Vol. 8, No. 4 (2015), pp. 287-294, doi: 10.14257/ijfgcn.2015.8.4.28

45. Irin Dorathy, M. Chandrasekaran «Simulation tools for mobile ad hoc networks: a survey» Journal of applied research and technology, vol.16 no.5 2018, pp. 70-186 ISSN 1665-6423

46. manna sim [электронный ресурс], URL: http://www.mannasim.dcc.ufmg.br/, дата обращения: 08.12.2020

47. MobiWan: NS-2 extensions to study mobility in Wide-Area IPv6 Networks [электронный ресурс], URL: http://www.inrialpes.fr/planete/mobiwan/t-4.x, дата обращения: 17.09.2020

48. inetmanet [электронный ресурс], URL: https://github.com/aarizaq/inetmanet-4.x, дата обращения: 27.12.2020

49. Amanjot Singh Toor, A. K. Jain «A Survey on Wireless Network Simulators» Bulletin of Electrical Engineering and Informatics, Vol. 6, No. 1, March 2017, pp. 62-69, doi: 10.11591/eei.v6i1.568

50. H.S. Ham, M.J. Choi. Analysis of Android Malware Detection Performance using Machine Learning Classifiers //Information and Communication Technology Convergence (ICTC). – 2016. – P. 490-495.

51. Azmoodeh A. et al. Detecting crypto-ransomware in IoT networks based on energy consumption footprint //Journal of Ambient Intelligence and Humanized Computing. – 2018. – Т. 9. – №. 4. – С. 1141-1152.

52. Anis Zarrad, Izzat Alsmadi «Evaluating network test scenarios for network simulators systems» International Journal of DistributedSensor Networks 2017, Vol. 13(10), doi 10.1177/2F1550147717738216

53. OMNeT++ 5.6 Released [электронный ресурс], URL: https://omnetpp.org/software/2020/01/13/omnet-5-6-released , дата обращения: 27.08.2021

54. ns-3.30 [электронный ресурс], URL: https://www.nsnam.org/releases/ns-3-30/, дата обращения: 28.12.2020

55. ns-3 Model Library [электронный ресурс], URL: https://www.nsnam.org/docs/release/3.30/models/ns-3-model-library.pdf, дата обращения: 03.01.2021

56. Model Catalog [электронный ресурс], URL: https://inet.omnetpp.org/Protocols.html, дата обращения: 04.08.2021

57. wifi-simple-adhoc.cc [электронный ресурс], URL: https://www.nsnam.org/doxygen/wifi-simple-adhoc_8cc_source.html, дата обращения: 12.03.2021

58. Energy Models [электронный ресурс], URL: https://www.nsnam.org/docs/release/3.33/doxygen/group__energy.html, дата обращения:25.05.2021

59. ns3::WifiRadioEnergyModel Class Reference [электронный ресурс], URL:

https://www.nsnam.org/docs/release/3.33/doxygen/classns3_1_1_wifi_radio_energy_model.html, дата обращения:25.08.2021