**Задание к практическому занятию 6**

1. **Сделать краткий обзор программных пакетов, реализующих беспроводную самоорганизующуюся сеть.**

Беспроводные самоорганизующиеся сети (Wireless Ad Hoc Networks) — это сети, где узлы могут взаимодействовать друг с другом напрямую без централизованной инфраструктуры. Для их реализации и симуляции используются различные программные пакеты, которые обеспечивают моделирование, анализ и разработку таких сетей. Вот краткий обзор популярных решений:

**1. Briar**

* **Описание**: Приложение для безопасной, децентрализованной связи.
* **Технология**: Использует Bluetooth, Wi-Fi и Tor (при наличии интернета).
* **Особенности**:
  + Прямое соединение между устройствами.
  + Полностью децентрализованная архитектура (без серверов).
  + Сквозное шифрование для конфиденциальности.
* **Применение**: Общение в условиях отключения интернета, стихийных бедствий или для безопасной связи активистов и журналистов.

**2. Meshtastic**

* **Описание**: Программно-аппаратное решение для связи через LoRa-устройства.
* **Технология**: Радиомодули LoRa создают Ad Hoc сети с большой дальностью связи.
* **Особенности**:
  + Радиус действия одного узла — до 10 км.
  + Сообщения отправляются через смартфон, подключенный к устройству по Bluetooth.
  + Независимость от интернета и высокая энергоэффективность.
* **Применение**: Экспедиции, походы, аварийные ситуации, связь в удаленных районах.

**3. Serval Mesh**

* **Описание**: Приложение для создания mesh-сетей на основе Wi-Fi.
* **Технология**: Использует Wi-Fi Direct для прямого подключения устройств.
* **Особенности**:
  + Поддержка передачи голосовых сообщений и текстовых чатов.
  + Децентрализованная структура без серверов.
* **Применение**: Экстренные ситуации, протесты, удаленные районы.

**4. FireChat**

* **Описание**: Приложение для обмена сообщениями в режиме оффлайн.
* **Технология**: Bluetooth и Wi-Fi для соединения устройств в локальной сети.
* **Особенности**:
  + Поддержка групповых чатов.
  + Ограниченный радиус действия (до 100 м между узлами).
* **Применение**: Для общения на фестивалях, протестах и в зонах с плохой связью.

**5. Althea**

* **Описание**: Программное обеспечение для построения mesh-сетей с доступом в интернет.
* **Технология**: Ad Hoc маршрутизация через Wi-Fi или Ethernet.
* **Особенности**:
  + Интеграция с криптовалютами для оплаты услуг связи.
  + Ориентировано на создание локальных сетей для доступа в интернет.
* **Применение**: Сообщества в сельских и удаленных регионах.

**6. CJDNS**

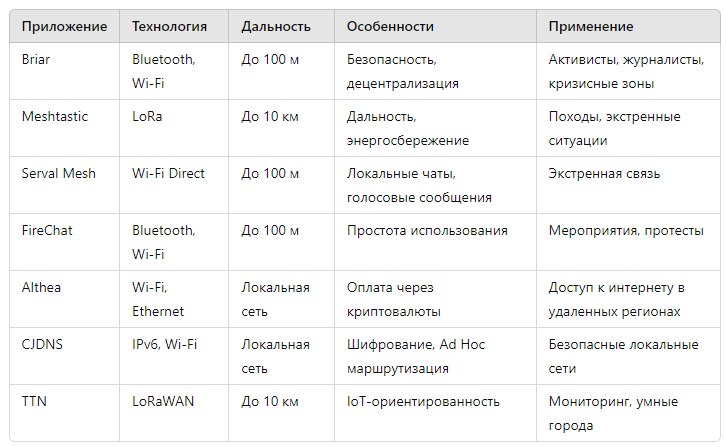
* **Описание**: Протокол для создания децентрализованных сетей.
* **Технология**: Использует IPv6-адресацию и криптографию для безопасной маршрутизации.
* **Особенности**:
  + Поддержка создания Ad Hoc сетей через Wi-Fi.
  + Шифрование всех соединений.
* **Применение**: Создание безопасных локальных сетей.

**7. The Things Network (TTN)**

* **Описание**: Платформа для создания IoT-сетей с использованием LoRaWAN.
* **Технология**: Подключение устройств через LoRaWAN для обмена данными.
* **Особенности**:
  + Упор на передачу данных между IoT-устройствами.
  + Большой радиус действия и низкое энергопотребление.
* **Применение**: Умные города, сельское хозяйство, удаленный мониторинг.

**8. RetroShare**

* **Описание**: Децентрализованное приложение для обмена файлами и сообщений.
* **Технология**: P2P-сеть с поддержкой оффлайн-режима.
* **Особенности**:
  + Поддержка Ad Hoc подключения через Wi-Fi или Bluetooth.
  + Высокий уровень конфиденциальности за счет шифрования.
* **Применение**: Общение и обмен данными в автономных условиях.



1. **Выбрать один из этих пакетов и установить у себя на телефоне, попробовать связаться с соседями из группы** (**Тестирование мессенджера Briar):**

*Регистрация и добавление контактов*

Я и мой одногруппник установили приложение Briar на наши телефоны.

Для добавления друг друга в список контактов мы использовали функцию сканирования QR-кодов. Это позволило быстро и безопасно обменяться контактами.

*Тестирование без доступа к интернету*

После успешного добавления контактов мы отключили интернет на наших устройствах, чтобы проверить, как приложение работает в автономном режиме.

*Использование Bluetooth*

Мы попробовали общаться через чат и обнаружили, что приложение продолжает работать, используя Bluetooth.

Мы без проблем обменивались текстовыми сообщениями, что подтвердило способность Briar обеспечивать связь в условиях отсутствия традиционной сети.

Примечание: приложение блокирует возможность создания скриншотов.

1. **Сделать обзор API устройств Интернета вещей для операционной системы на вашем рабочем месте.**

API для IoT — это интерфейсы, которые позволяют интегрировать устройства с платформами и системами, предоставляя разработчикам инструменты для управления устройствами, сбора данных и выполнения действий. Рассмотрим несколько примеров популярных API для работы с IoT, их возможности и сферы применения:

**1. AWS IoT Core**

* **Описание**: Платформа от Amazon Web Services для подключения и управления IoT-устройствами. Поддерживает масштабируемую и безопасную интеграцию.
* **Функции**:
  + Двусторонняя связь между устройствами и облаком через MQTT, HTTP и WebSocket.
  + Управление устройствами (аутентификация, авторизация, обновление прошивок).
  + Сбор и анализ данных через встроенные аналитические инструменты.
* **Применение**:
  + Умные дома (контроль освещения, температуры).
  + Промышленный IoT (мониторинг производственного оборудования).
  + Сельское хозяйство (контроль влажности почвы, полив).

**2. Google Cloud IoT**

* **Описание**: Облачная платформа Google для управления IoT-устройствами и анализа данных.
* **Функции**:
  + Подключение устройств через MQTT и HTTP.
  + Хранение и обработка данных в реальном времени с использованием BigQuery.
  + Интеграция с машинным обучением для построения прогнозных моделей.
* **Применение**:
  + Управление городскими IoT-системами (например, управление уличным освещением или трафиком).
  + Энергосбережение (оптимизация энергопотребления в умных зданиях).

**3. IBM Watson IoT**

* **Описание**: Платформа IBM для управления IoT-устройствами и обработки данных с использованием аналитических возможностей.
* **Функции**:
  + Подключение и управление устройствами через MQTT и REST API.
  + Встроенная аналитика для выявления закономерностей и автоматизации процессов.
  + Интеграция с искусственным интеллектом (Watson AI) для обработки данных.
* **Применение**:
  + Здравоохранение (мониторинг медицинских приборов).
  + Автономные транспортные средства (анализ данных с датчиков).

**4. Azure IoT Hub**

* **Описание**: Платформа от Microsoft для подключения, мониторинга и управления IoT-устройствами.
* **Функции**:
  + Безопасная двусторонняя связь между устройствами и облаком через протоколы MQTT, AMQP и HTTPS.
  + Поддержка массового управления устройствами (например, обновление прошивок).
  + Интеграция с другими сервисами Azure для анализа данных.
* **Применение**:
  + Смарт-грины (управление энергосетями).
  + Производство (управление производственными линиями).

**5. CoAP (Constrained Application Protocol)**

* **Описание**: Протокол прикладного уровня для ресурсов с низким энергопотреблением и ограниченными возможностями.
* **Функции**:
  + Упрощённая REST-модель, оптимизированная для передачи данных в IoT-сетях.
  + Используется в устройствах с ограниченными вычислительными ресурсами.
* **Применение**:
  + Управление сенсорными сетями.
  + Устройства умного дома, например, умные замки или термостаты.

1. **Попытаться инсталлировать один из API на своем рабочем месте.**

В этом примере я продемонстрировал использование OpenWeatherMap One Call API 3.0 на Python для получения текущей погоды и прогноза.

1. **Установка библиотеки**: установил requests для выполнения HTTP-запросов (pip install requests).
2. **Получение API-ключа**: зарегистрировался на сайте OpenWeatherMap для получения уникального ключа.
3. **Написание скрипта**: представил пример кода, включающий:
   * Импорт requests.
   * Функцию get\_weather\_data для запроса данных по координатам и ключу.
   * Функцию print\_weather для отображения текущей погоды из JSON-ответа.
   * Основной блок с координатами, API-ключом и вызовом функций.
4. **Запуск скрипта**: сохранить код в api.py, заменил ключ, и запустил скрипт в терминале.

