**Оглавление**

[Введение 3](#_Toc27046362)

[1 Глава – Выбор эмулятора для работы с беспроводными сетями 5](#_Toc27046363)

[1.1 Система контроля версий Git 5](#_Toc27046364)

[1.1.1 Изучение Git 5](#_Toc27046365)

[1.1.2 Основные команды Git 5](#_Toc27046366)

[1.1.3 Выбор инструмента для проведения слияний Git 7](#_Toc27046367)

[1.1.4 Разрешение конфликта при слиянии. 8](#_Toc27046368)

[1.2 Основные сетевые команды для работы с беспроводными сетями 11](#_Toc27046369)

[1.2.1 Утилита iw 11](#_Toc27046370)

[1.2.2 Утилита wpa\_cli 12](#_Toc27046371)

[1.2.3 Утилита ip 14](#_Toc27046372)

[1.3 Сравнение эмуляторов беспроводных сетей 17](#_Toc27046373)

[1.4 Настройка и эксплуатация эмулятора 20](#_Toc27046374)

[Выводы по первой главе 22](#_Toc27046375)

[2 Глава – Возможности программно-конфигурируемых беспроводных сетей 23](#_Toc27046376)

[2.1 Исследование возможностей эмулятора 23](#_Toc27046377)

[2.1.1 Назначение и основные возможности модуля wmediumd 23](#_Toc27046378)

[2.1.2 Исследование взаимодействия беспроводных устройств 25](#_Toc27046379)

[2.2 Построение различных конфигураций при помощи MininetWifi 28](#_Toc27046380)

[2.2.1 WifiDirect 28](#_Toc27046381)

[2.2.2 VANETs 30](#_Toc27046382)

[2.2.3 Mobility 31](#_Toc27046383)

[2.2.4 Связь через сокеты 31](#_Toc27046384)

[2.2.5 Mesh сети 32](#_Toc27046385)

[2.2.6 Модели потерь данных 34](#_Toc27046386)

[2.2.7 Adhoc 38](#_Toc27046387)

[Выводы по второй главе 40](#_Toc27046388)

[Заключение 41](#_Toc27046389)

[Список литературы 42](#_Toc27046390)

# Введение

Целью курсовой работы является совершенствование технологии разработки средств анализа кадров с помощью программно-конфигурируемых WiFi сетей. Данная цель разбивается на следующие задачи:

* поиск подходящего эмулятора;
* имитация атак на точки доступа и защита от них;
* максимальное приближение поведения программного стенда к реальным устройствам;
* изучение работы систем обнаружения вторжений;
* запись файлов трафика формата .pcap с различными протоколами и их анализ.

Структура курсовой работы: в первой главе рассматриваются необходимые для дальнейшей работы с беспроводными сетями программные средства и основные сетевые команды. В первой главе сравниваются эмуляторы беспроводных сетей, выбирается наиболее подходящий эмулятор для решения поставленных задач. Во второй главе исследуются возможности эмулятора и его применимость в различных сферах деятельности.

Mininet - это эмулятор проводной компьютерной сети, который позволяет познакомиться с устройством и функционированием компьютерных сетей без необходимости использования какого-либо сетевого оборудования. Сеть составляют простые компьютеры - хосты, коммутаторы, а так же OpenFlow-контроллеры. С помощью простейшего синтаксиса в примитивном интерпретаторе команд можно разворачивать сети из произвольного количества хостов, коммутаторов в различных топологиях в рамках одной виртуальной машины. На всех хостах можно изменять сетевую конфигурацию, пользоваться стандартными утилитами(ipconfig, ping), получать доступ к терминалу. На коммутаторы можно добавлять различные правила и маршрутизировать трафик. В настоящее время развитие информационных технологий и постоянный рост «интернета вещей» ведет к повсеместному распространению беспроводных сетей. Сети Wi-Fi являются основой высокопроизводительных сетей для дома и офиса. Такие сети имеют ряд преимуществ: совместимость, быстрота построения сети, мобильность рабочих мест. Зачастую Wi-Fi используется в случаях, когда развёртывание кабельной системы невозможно или экономически нецелесообразно. Основными тенденциями развития данной технологии являются: увеличение скорости передачи данных, улучшение качества обмена данными, повышение безопасности, выделение дополнительных полос частот.

На данный момент задача оптимального использования сетевого оборудования с технологией беспроводной передачи данных является актуальной. Взаимодействие устройств и влияющие на него факторы трудно воспроизвести реальном мире. Для решения этих задач было рассмотрены средства эмуляции беспроводных сетей. Mininet WiFi является дополнением проекта Mininet, которое предоставляет возможность работы с беспроводными сетями. Функциональность Mininet была расширена добавлением виртуальных WiFi-станций и точек доступа, основанных на Linux драйвере 80211\_hwsim. Были добавлены классы для поддержки сценариев, имитирующих местоположение, движение и помехи при взаимодействии станций и точек доступа.

# Глава - Выбор эмулятора для работы с беспроводными сетями

## Система контроля версий Git

### Изучение Git

Cистема контроля версий (СКВ) предназначена для отслеживания истории изменений проекта. При помощи СКВ группа разработчиков может комфортно взаимодействовать при работе с одним проектом, вносить новый код, запускать тесты, исправлять ошибки. История проекта содержит пункты:

* разница с предыдущим изменением;
* имя пользователя, который внес изменения;
* время внесения изменений;
* причина внесения изменений.

Git является примером распределенной системы контроля версий. Такая система предоставляет полный доступ к каждому файлу, ветви и итерации проекта, а также предоставляет каждому пользователю доступ к полной и автономной истории всех изменений.

### Основные команды Git

Чтобы использовать Git, разработчики используют специальные команды для копирования, создания, изменения и объединения кода. Эти команды могут быть выполнены непосредственно из командной строки или с помощью графического интерфейса. Ниже представлены некоторые общие команды для использования:

Таблица 1.1 - Список основных команд Git

|  |  |
| --- | --- |
| Команда | Назначение |
| git init | Инициализирует новый Git-репозиторий и начинает отслеживать существующий каталог. Он добавляет скрытую подпапку .git в существующий каталог, в котором находится внутренняя структура данных, необходимая для контроля версий |

Продолжение таблицы 1.1

|  |  |
| --- | --- |
| git clone | Создает локальную копию проекта, который уже существует удаленно. Клон включает в себя все файлы проекта, историю и ветви |
| git add | Фиксирует изменения. Git отслеживает изменения в кодовой базе разработчика, но необходимо подготовить и сделать снимок изменений, чтобы включить их в историю проекта. Эта команда выполняет постановку, первую часть этого двухэтапного процесса. Все внесенные изменения станут частью следующего снимка и истории проекта. Постановка и фиксация по отдельности дает разработчикам полный контроль над историей своего проекта, не меняя при этом код и работу |
| git commit | Сохраняет снимок в историю проекта и завершает процесс отслеживания изменений. Все, что было поставлено git add, станет частью снимка с git commit |
| git status | Показывает статус изменений как неотслеживаемых, так и измененных или промежуточных |
| git branch | Показывает ветви, над которыми работают локально |
| git merge | Объединяет ветки развития вместе. Эта команда обычно используется для объединения изменений, сделанных в двух разных ветвях. Например, разработчик может объединиться, если он хочет объединить изменения из ветви компонентов в главную ветвь для развертывания |
| git pull | Обновляет локальную линию разработки обновлениями от своего удаленного аналога. Разработчики используют эту команду, если партнер по команде сделал коммиты на ветке на удаленном компьютере, и они хотели бы отразить эти изменения в своей локальной среде |
| git push | Обновляет удаленный репозиторий любыми коммитами, сделанными локально в ветке |

### Выбор инструмента для проведения слияний Git

В рамках курсовой работы была поставлена задача - выбор Git инструмента для проведения слияний (merge). Были выделены основные критерии для выбора:

* небольшой вес программы;
* возможность просмотра истории коммитов;
* просмотр изменений в виде графа;
* проведение слияний;
* кроссплатформенность;
* открытый исходный код.

В таблице 1 рассматривается несколько наиболее распостраненных решений. Производится сравнение и оценка достоинств и недостатков каждого из них.

Таблица 1.2 - Сравнение инструментов слияния git

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | KDiff3 | DiffMerge | Meld | Diffuse | Araxis Merge |
| Цена | - | $19 | - | - | $129 |
| Размер | 11Mb | 8Mb | 38Mb | 9Mb | 140Mb |
| Трехстороннее слияние | + | + | + | + | + |
| Зависимости | TortoiseHg | - | Python, GTK+, Glib, GtkSourceView | - | - |
| Поддержка Windows, Mac OS X, Linux | + | + | + | + | - |
| Открытый исходный код | - | - | + | + | - |

Продолжение таблицы 1.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Возможность сравнения директорий | + | + | + | - | + |
| Прочие недостатки | - | Проблемы с кирилицей | - | - | - |

По результатам сравнения было решено использовать открытое решение Diffuse. Оно является кроссплатформенным, имеет небольшой вес и поддерживает основные ОС.

### Разрешение конфликта при слиянии.

Перед слиянием, в котором может возникнуть конфликт, небходимо убедиться в том, чтобы рабочая директория была без локальных изменений.

Если имеются несохраненные наработки, необходимо сохранить их во временной ветке. Таким образом, имеется возможность легко отменить любые изменения, которые сделаны в рабочей директории. Если при выполнении слияния не сохранить изменения сделанные в рабочей директории, то некоторые данные могут быть потеряны.

Далее необходимо рассмотреть разрешение конфликта при помощи diffuse. Для начала необходимо выбрать diffuse инструментом для разрешения конфликтов по-умолчанию. Для этого требуется настроить файл gitconfig. Настройка файли представлена в листинге 1.1.

Листинг 1.1

[diff]

tool = dfu

[difftool "dfu"]

cmd = diffusew $LOCAL $REMOTE

[difftool]

prompt = false

keepBackup = false

trustExitCode = false

[merge]

tool = dfu

[mergetool "dfu"]

cmd = diffusew $MERGED $LOCAL $BASE $REMOTE

Проверена работа сравнения командой:

$ git difftool f6e6909 cb950e2

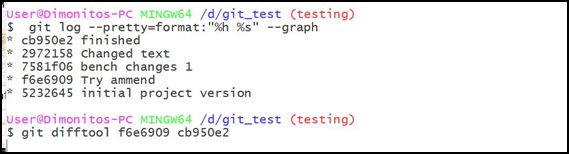


Рисунок 1.1 - Команда сравнения данных

При выполнении команды открывается окно редактора. Отображаются отличия содержимого двух коммитов, сделанных в разное время:

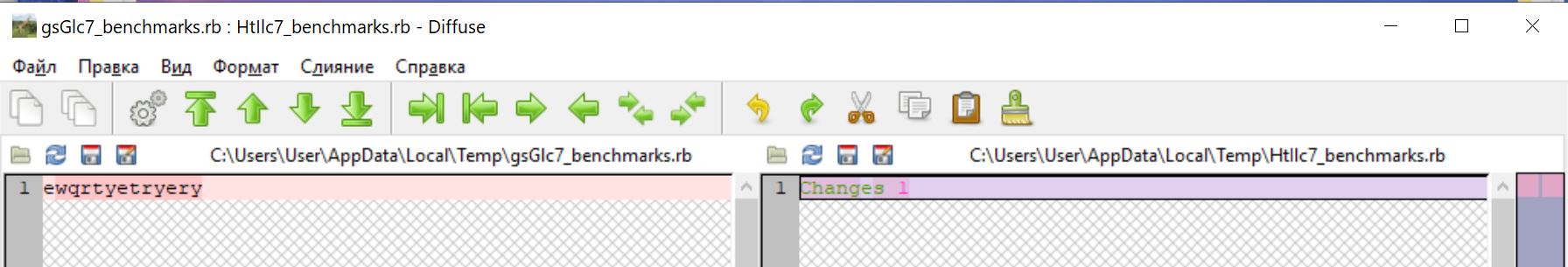


Рисунок 1.2 - Сравнение данных в коммитах

Проверена работа слияния (merging):

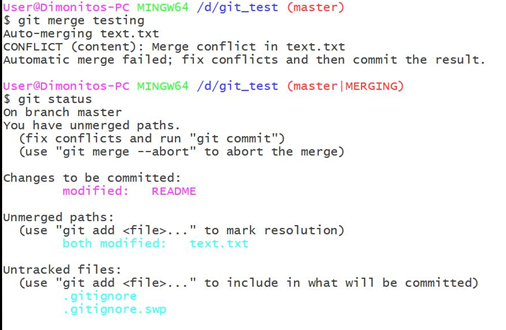


Рисунок 1.3 - Запрос на слияние

Был получен конфликт при слиянии.

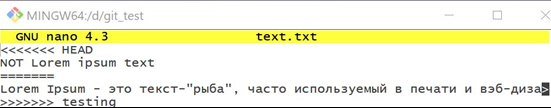


Рисунок 1.4 - Требующий вызова mergetool файл

Конфликт разрешен при помощи Diffuse:

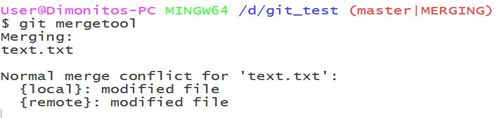


Рисунок 1.5 - Вызов mergetool

Необходимо выбрать один из четырех вариантов содержимого файла для успешного слияния.

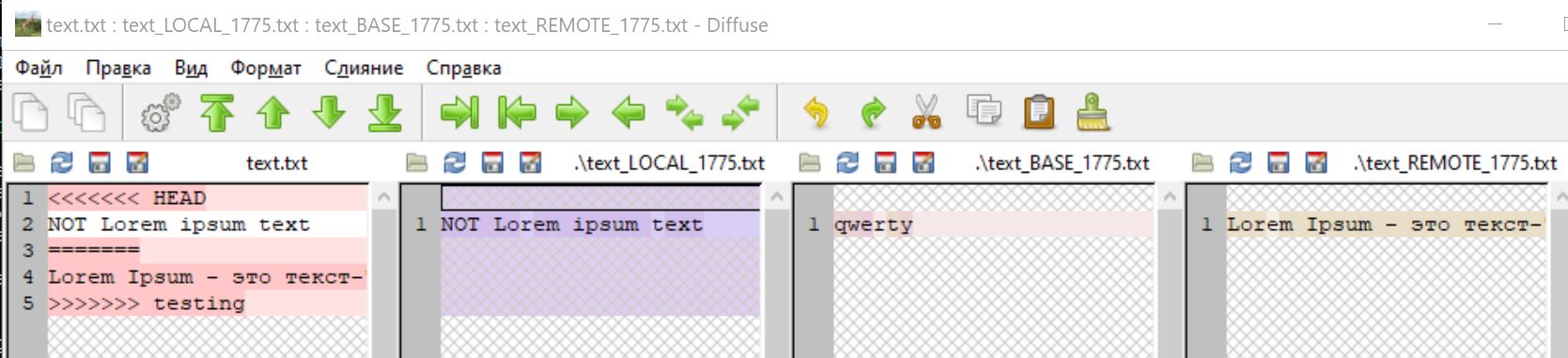


Рисунок 1.6 - Процесс разрешения конфликта

## Основные сетевые команды для работы с беспроводными сетями

ОС Linux предоставляет широкий спектр команд для настройки и отладки беспроводных адаптеров. Производители Linux выпустили дистрибутивы, которые поддерживают значительное количество беспроводных карт. Ubuntu Wi-Fi документация является хорошим, и часто обновляемым руководством, где указана информация о том, какие карты поддерживаются в последних версиях Ubuntu. В этой главе рассмотрены основные инструменты для работы с беспроводными сетевыми адаптерами в ОС Linux.

### Утилита iw

iw - новая утилита командной строки, основанная на nl80211, для беспроводных устройств. Она поддерживает все новые драйверы, которые были добавлены в ядро за последнее время. Рассмотрим основные команды, необходимые для работы с устройствами беспроводной передачи данных:

Таблица 1.3 - Основные команды утилиты iw

|  |  |
| --- | --- |
| Команда | Назначение |
| iw help | Просмотр документации |
| iw event | Ожидание события от ядра |
| iw dev wlan0 link | Просмотр состояния соединения |
| iw wlan0 connect ssid-no-password | Установка подключения без пароля |
| iw wlan0 connect ssid-wep key s:password-ascii | Установка подключения wep |

Продолжение таблицы 1.3

|  |  |
| --- | --- |
| wpa\_passphrase your-essid your-passphrase | sudo tee /etc/wpa\_supplicant.conf  sudo wpa\_supplicant -c /etc/wpa\_supplicant.conf -i wlan0 | Установка подключения wpa |
| iw dev wlan1 station dump | Получение информации о станции |
| iw wlan0 set bitrates legacy-2.4 12 18 24 | Задание скорости передачи данных |
| iw dev <devname> set txpower <auto|fixed|limit> [<tx power in mbm>] | Настройка мощности передатчика |
| sudo iw dev wlan0 set power\_save on | Энергосбережение |
| iw phy phy0 interface add moni0 type monitor | Добавление интерфейсов |
| iw dev moni0 del | Удаление интерфейсов |
| iw dev wlan0 scan | Поиск точек доступа |

### Утилита wpa\_cli

Wpa\_supplicant является реализацией компонента IEEE 802.1X/WPA Supplicant, который используется на клиентских машинах. Он реализует согласование ключей шифрования с аутентификатором WPA, аутентификацию EAP с сервером аутентификации (Authentication Server), а также управляет роумингом и выполняет сопряжение адаптера с беспроводной сетью.

Утилита wpa\_cli представляет собой текстовый пользовательский интерфейс для взаимодействия с wpa\_supplicant. Ее можно использовать для запросов текущего статуса, изменения конфигурации, инициирования события и запроса, интерактивного ввода данных пользователем.

Утилита wpa\_cli может выполняться в интерактивном режиме или в режиме командной строки. В обоих режимах используется один и тот же набор программ. Основное отличие интерактивного режима – обеспечение доступа к незатребованным сообщениям (сообщения о событиях, запросы имени пользователя/пароля).

**Поддерживаемые команды**

В настоящее время утилита wpa\_cli поддерживает следующие команды:

Таблица 1.4 - Основные команды wpa\_cli

|  |  |
| --- | --- |
| Команда | Назначение |
| add\_network | Добавление сети |
| bssid идентификатор\_сети BSSID | Указание предпочтительного идентификатора BSSID для SSID |
| disable\_network идентификатор\_сети | Деактивация сети |
| disconnect | Разрыв соединения до выполнения команды reassociate |
| get\_capability eap/pairwise/group/key\_mgmt/proto/auth\_alg | Получение возможностей для аутентификации |
| get\_network | Идентификатор\_сети переменная - получение переменных сети |
| help | Вывод информации по возможностям использования утилиты |
| identity идентификатор\_сети принадлежность | Настройка принадлежности для SSID |
| interface [имя\_интерфейса] | Выбор интерфейса; |
| new\_password идентификатор\_сети пароль | Изменение пароля для SSID |
| quit | Завершение выполнения wpa\_cli |
| reassociate | Принудительное выполние повторного установления связи |
| reconfigure | Повторное выполнение чтения конфигурационного файла wpa\_supplicant |
| remove\_network идентификатор\_сети | Удаление сети |

Продолжение таблицы 1.4

|  |  |
| --- | --- |
| save\_config | Сохранение текущей конфигурации |
| scan | Запрос нового сканирования BSS |
| scan\_results | Получение результатов последнего сканирования |
| select\_network идентификатор\_сети | Выбор сети (деактивировать другие сети) |
| status [verbose] | Получение текущего статуса WPA/EAPOL/EAP |
| terminate | Прерывание выполнения wpa\_supplicant. |

### Утилита ip

Утилита ip объединяет в себе возможности команд ifconfig, arp и route. Теперь рассмотрим доступные команды, с помощью которых может быть выполнена настройка сети linux:

Таблица 1.5 - Основные команды ip

|  |  |
| --- | --- |
| Команда | Назначение |
| ip link show | Отображение состояния всех сетевых интерфейсов |
| ip neigh show | Показ всех записей ARP |
| ip address show | Показ всех ip адресов и их интерфейсов |
| ip route show | Показ все маршрутов в таблице маршрутизации |
| ip addr add ADDRESS dev IFNAME | Назначение IP-адреса интерфейсу |
| ip link set dev {DEVICE} {up|down} | Включение или выключение интерфейсов |
| ip route flush cache | Очистка кэша таблицы маршрутизации |

Продолжение таблицы 1.5

|  |  |
| --- | --- |
| ip route add 10.0.0/24 via 193.233.7.65 | Добавление маршрута к сети 10.0.0/24 через 193.233.7.65; |
| ip dev {DEVICE} arp on или arp off | Изменение значение флага NOARP на устройстве |
| ip dev {DEVICE} multicast on или multicast off | Изменение флага а MULTICAST на устройстве |
| ip dev {DEVICE} dynamic on или dynamic off | Изменение флага DYNAMIC на устройстве |
| ip dev {DEVICE} name | Изменение имени устройства |
| ip dev {DEVICE} txqueuelen Число или txqlen Число | Изменение длины передаваемой очереди |
| ip dev {DEVICE} mtu Число | Изменение значения MTU на устройстве |
| ip dev {DEVICE} address Адрес | Изменение адреса на устройстве |
| ip dev {DEVICE} broadcast | Адрес или brd Адрес |

* + 1. Рассмотрение iperf

Iperf - кроссплатформенная консольная клиент-серверная программа -генератор TCP и UDP трафика для тестирования пропускной способности сети.

Для тестирования производительности сети на сервере выполняется команда:

iperf -s

На клиенте:

iperf -c server\_host

Параметры рассмотрены в таблицах 1.6-1.8:

Таблица 1.6 – Команды iperf общие для сервера и клиента

|  |  |
| --- | --- |
| Ключ | Описание |
| -p | Определить порт, на котором будет слушать сервер или отправлять запросы клиент |
| -i | Интервал между выводами результатов |
| -F | Указать файл, из которого будут взяты входные данные для запуска |
| -A | Степень нагрузки на процессор |
| -B | Указать, через какой сетевой интерфейс работать |
| -h | Вызвать справку по работе с программой |

Таблица 1.7 – Команды iperf для сервера

|  |  |
| --- | --- |
| Ключ | Описание |
| -s | Запустить iperf в режиме сервера |
| -D | Запустить как демона (как службу) |
| -I | Указать pid-файл |
| -1 | Принять запрос от одного клиента и завершить работу |

Таблица 1.8 - Команды iperf для клиента

|  |  |
| --- | --- |
| Ключ | Описание |
| -c | Запустить iPerf в режиме клиента |
| -u | Отправлять UDP-пакеты |
| -t | Количество секунд, в течение которых будет идти проверка скорости |
| -n | Объем данных для проверки (применяется вместо времени -t) |
| -k | Количество пакетов для проверки (вместо -t или -n) |
| -l | Длина буфера записи/чтения |
| -P | Число параллельных запросов |

Продолжение таблицы 1.8

|  |  |
| --- | --- |
| -R | Обратный режим - сервер отправляет, клиент принимает |
| -w | Размер сетевого окна |
| -M | Задать максимальный размер MTU |

**Примеры:**

1. Запуск сервера в UDP-режиме с выводом результатов раз в секунду, на порту 65005:

iperf -u -s -i1 -p 65005

1. Тестирование UDP c длиной 100 байт со скоростью 20 Кбит/с в течение 180 секунд (имитация VoIP трафика, запускать нужно в обе стороны):

iperf -u -c server\_host -l100 -b20k -t180

1. Передать 10 мегабайт данных на порт 65005

iperf -n 10m -p 65005 -c server\_host

## Сравнение эмуляторов беспроводных сетей

В рамках курсовой работы были рассмотрены эмуляторы, позволяющие работать с беспроводными сетями. Частью задачи является нахождение эмулятора, позволяющего воспроизводить поведение Wi-Fi сетей программно. Для этого в таблице приведено сравнение программных средств, реализующих данную функцию.

Основние требования к эмулятору:

* создание большого количества программных точек доступа и клиентов;
* запись трафика в файл формата .pcap;
* создание интерфейсов в различных режимах;
* задание среды для передачи данных и ее настройка;
* возможность использования основных сетевых утилит.

Таблица 1.9 - Рассмотрение эмуляторов беспроводных сетей

|  |  |
| --- | --- |
| Название эмулятора | Краткая характеристика |
| Mininet Wifi | Работают утилиты Linux. Поддерживает работу до 100 точек доступа. Отображение сети в виде интерактивного графа. Запись трафика в файл .pcap. Работа нескольких интерфейсов в разных режимах. |
| Seawind | Программное обеспечение эмуляции Seawind использует существующие реализации протоколов в смоделированных сетевых средах. Программное обеспечение Seawind предоставляет большой набор способов определения характеристик передачи, включая задержки и ошибки. Устарел, нет исполняемых файлов. |
| OFNet | OFNet - это новый программно-определяемый сетевой эмулятор (SDN), который предлагает функциональность, аналогичную сетевому эмулятору Mininet, и добавляет некоторые полезные инструменты для генерации трафика и мониторинга сообщений OpenFlow и оценки производительности контроллера SDN. Устарел, нет исполняемых файлов. |
| iTrinegy | Сетевые эмуляторы используются организациями, которым необходимо проверять производительность приложений в среде, которая точно копирует реальные сети. Сетевые эмуляторы рассчитывают помехи для сигнала, чтобы имитировать опыт реальной сети. Платное решение. |

Продолжение таблицы 1.9

|  |  |
| --- | --- |
| NetSlim | Сетевой симулятор для обучения / сетевых лабораторных экспериментов в университетах.  TCP / IP, Ethernet, маршрутизация, Wi-Fi, MANET, IOT, LTE, WSN.  Пакетная анимация для визуального понимания. Платный продукт. |
| NS-2 | Ns - это симулятор дискретных событий, предназначенный для сетевых исследований. Ns обеспечивает существенную поддержку для моделирования протоколов TCP, маршрутизации и многоадресной передачи по проводным и беспроводным (локальным и спутниковым) сетям. Не подходит для целей курсовой работы. |
| OMNET++ | OMNeT ++ - это расширяемая, модульная, основанная на C ++ бмблиотека, используемая для построения сетевых симуляторов. Симулятор дискретных событий, включающий: IDE для создания моделей, и сам симулятор (GUI). Нет реализации точек доступа. |
| Qualnet | Qualnet Simulator - это коммерческий симулятор, используемый для проектирования и анализа производительности сетей и сетевых протоколов. |
| Opnet | Решение для проводных сетей. Не поддерживает беспроводные устройства передачи данных |

Исходя из содержимого таблицы 1.9 было решено использовать эмулятор Mininet Wifi из за открытого исходного кода с возможностью доработки скриптов, классов и библиотек, работоспособности команд linux, графического интерфейса и интуитивно понятного способа работы с узлами сети.

## Настройка и эксплуатация эмулятора

#### Установка среды MininetWifi

Действия выполняются на OS Ubuntu 18.04 LTS.

Установка осуществляется командами указанными в листинге 1.2:

Листинг 1.2

sudo apt-get install git

git clone https://github.com/intrig-unicamp/mininet-wiﬁ

cd mininet-wiﬁ

sudo util/install.sh -Wnfv

#### Построение заданной топологии

Для создания простейшей сети (1 точка доступа и 2 клиента) используется команда sudo mn --wifi. При указании параметра --wifi активируется модуль Mininet WIFI в среде Mininet.

Ниже рассмотрена та же команда с использованием нескольких дополнительных флагов.

sudo mn --wifi --topo linear,5 --position --plot

Команда имеет следующие параметры:

* topo linear,5 топология сети линейная (каждой точке доступа соответствует один подключенный к ней клиент) 5 - количество клиентов. При использовании топологии single создается одна точка доступа и 5 клиентов;
* position автоматически устанавливает позиции узлов;
* plot строит граф узлов и их областью видимости и относительным расположением.

Команда очистки и сброса настроек и восстановления настроек сети в ОС:

sudo mn -c

#### Конфигурация топологии

**При помощи терминальных команд.**

После построения модели необходимо провести ее дальнейшую настройку. При указании узла сети (sta1), возможно использование консольных команд Ubuntu. Была рассмотрена часть из них:

* sta1 iw dev sta1-wlan0 link;
* sta1 iw dev sta1-wlan0 disconnect;
* sta1 iw dev sta1-wlan0 connect ap2-wlan1;
* sta1 iw dev sta1-wlan0 info.

При помощи команды link определяется к какой точке доступа подключен клиент на данный момент. Connect - подключение к точке доступа, disconnect - отключение от нее, info - вывод информации.

В командах используются следующие параметры:

* sta1 - идентификатор клиента;
* sta1-wlan0 - беспроводной интерфейс клиента;
* ap2-wlan1 - беспроводной интерфейс точки доступа.

Рассмотрены команды ifconfig и iwconfig и их возможности в рамках стенда. Последовательность команд для перевода точки доступа в режим мониторинга:

1. ap1 ifconfig ap1-wlan1 down
2. ap1 iwconfig ap1-wlan1 mode monitor
3. ap1 ifconfig ap1-wlan1 up
4. ap1 iwconfig
5. ap1 tcpdump -i ap1-wlan1 -n -s 16 -w file3.cap

**При помощи Python скриптов**

Получить информацию о позиции узла:

py ap1.params['position']

Изменить положение узла:

py sta3.setPosition('70,40,0')

Изменить зону действия покрытия узла:

py sta1.setRange(100)

Настроить свойства канала беспроводной передачи данных: ширина пропускания, потери, задержки и латентность:

py net.setChannelEquation(bw='value.rate \* (1.1 \*\* -dist)', loss='(dist \* 2)/100', delay='(dist / 10) + 1', latency='2 + dist')

**При помощи специальных команд**

Отправка ICMP пакетов между узлами:

sta1 ping ap2

Нахождение расстояния между узлами:

distance sta1 ap1

Установка TCP соединения с последующей передачей данных:

1. xterm sta1 ap1
2. Терминал ap1: sta1 iperf --server
3. Терминал sta1: iperf --client 10.0.0.101 --time 60 --interval 2

## Выводы по первой главе

В первой главе были рассмотрены основные способы взаимодействия с беспроводными устройствами передачи данных, необходимых для решения задачи - совершенствования технологии разработки средств анализа кадров с помощью программно-конфигурируемых Wi-Fi сетей. Перечень команд ОС Linux предоставляет универсальный инструмент, позволяющий произвести базовую настройку и проверку оборудования. Помимо этого он может быть расширен пакетами и утилитами представленными в ОС Kali Linux. Это позволит использовать интерфейсы в разных режимах. Осуществлять захват и инжектирование кадров.

# Глава – Возможности программно-конфигурируемых беспроводных сетей

## Исследование возможностей эмулятора

### Назначение и основные возможности модуля wmediumd

Модуль wmediumd работает в отдельном процессе и позволяет эмулировать потери кадров и задержки при их приёме. Его настройка осуществляется с помощью конфигурационного файла вида: /tmp/mn\_wmd\_\*.config.

В конфигурационном файле wmediumd можно выбрать один из трёх типов моделирования: snr, prob, path\_loss.

Тип моделирования **snr** позволяет задавать для каждой пары устройств соотношение сигнал-шум (см. листинг 2.1), исходя из которого вычисляются:

* максимальная скорость передачи данных канального уровня;
* packet loss error probability - вероятность ошибки (потери кадров).

Частота появления ошибки (потери кадров) задаётся в виде таблицы в зависимости от скорости канального уровня. Под показателем "частоты" потери кадров понимается количество потерянных кадров из каждой тысячи отправленных.

Листинг 2.1

ifaces : {

ids = [

"02:00:00:00:00:00",

"02:00:00:00:01:00",

"02:00:00:00:02:00",

"02:00:00:00:03:00"

];

links = (

(0, 1, 0),

(0, 2, 0),

(2, 0, 10),

(0, 3, 0),

(1, 2, 30),

(1, 3, 10),

(2, 3, 20)); };

Если задана одна вероятность (одна строчка в links) для пары устройств, то канал связи будет считаться симметричным по качеству приёма.

Тип моделирования **prob** позволяет задавать вероятности потерь кадров(см. листинг 2.2).

Листинг 2.2

ifaces :

{

ids = [

"02:00:00:00:00:00",

"02:00:00:00:01:00",

"02:00:00:00:02:00",

"02:00:00:00:03:00"

];

};

model:

{

type = "prob";

default\_prob = 1.0;

links = (

(0, 2, 0.000000),

(2, 3, 0.000000)

);

};

При таком типе моделирования не учитывается, что потери кадров зависят от канальной скорости передачи кадров, а также от соотношения сигнал-шум. Правила задания симметричного канала - те же. Значение 0% - это идеальная среда.

Тип моделирования **path\_loss** позволяет выводить соотношение сигнал/шум и вероятности потерь кадров исходя из координат беспроводных устройств в 3-х мерном пространстве. Для типа моделирования path\_loss можно указать конкретную модель:

* ITU;
* logDistance;
* twoRayGround;
* logNormalShadowing;
* free\_space.

Указанные конкретные модели доступны через класс PropagationModels. И рассмотрены в п. 2.3.6. Наиболее интересны: LogDistance (для помещений) и Two-Ray-Ground (для улицы).

Пример конфигурационного файла для типа моделирования: path\_loss и конкретной модели: log\_distance (см. листинг 2.3)

Листинг 2.3

ifaces:

{

ids = [

"00:00:00:00:00:02",

"00:00:00:00:00:03",

"02:00:00:00:00:01",

"02:00:00:00:00:02"

];

enable\_interference = true;

};

model:

{

type = "path\_loss";

positions = (

(10.0, 20.0, 0.0),

(15.0, 25.0, 0.0),

(10.0, 30.0, 0.0),

(24.0, 55.0, 0.0)

);

fading\_coefficient = 0;

noise\_threshold = -91;

isnodeaps = (0, 0, 1, 1);

tx\_powers = (14.0, 14.0, 14.0, 14.0);

model\_name = "log\_distance";

path\_loss\_exp = 4.0;

xg = 0.0;

};

### Исследование взаимодействия беспроводных устройств

*Исследование возможности приема кадров с соседних каналов*

Была осуществлена проверка возможности приема кадров с соседних каналов с модуляцией CCK. Было выяснено, что станции не получают данных с соседних каналов. Этот функционал реализован в wmediumd, но не используется эмулятором. Эта поблема требует дальнейшего рассмотрения. На данном этапе было решено в дипазоне 2.4ГГц:

*Исследование поддержки механизма RTS\_CTS+ACK в демоне wmediumd*

Для эмуляции ретрансмиссии кадров была поставлена задача - проверка механизма RTS\_CTS+ACK в демоне wmediumd. Выяснено, что ретрансмиссия не поддерживается. Для ее реализации требуется модификация библиотеки mac80211\_hwsim.

* добавлять в название сети (SSID) информацию об эмулируемом частотном канале ТД;
* размещать точки доступа только на непересекающихся каналах: 1,6,11 и 14.

*Исследование максимального количества ТД в Mininet-WiFi*

Для решения данной задачи была зафиксирована конфигурация стенда:

* версия Mininet-WiFi ­- 2.4.1;
* версия модуля hwsim80211.ko - mac80211\_hwsim\_mgmt v0.1;
* версия ядра Linux - 5.0.0-36-generic.

ресурсы: 12 Гб ОЗУ, 8 процессоров.

Определено:

* максимально возможное количество одновременно работающих точек доступа WPA/WPA2-PSK - 100.
* максимально возможное количество одновременно работающих точек доступа WPA/WPA2-PSK с одним клиентом - 80.

При большем количестве точек доступа операционная система «зависает», дальнейшая работа невозможна.

*Исследование способа задания периодической активности клиента*

Для задания периодической активности клиента было решено использовать iperf (см. п 1.2.4). Утилита позволяет генерировать трафик различного типа для анализа пропускной способности сети. Поддерживается многопоточная работа.

*Проверка возможности инжектирования кадров в режиме мониторинга*

Возможность инжектирования кадров проверена на отправке пакетов деаутентификации утилиты aireplay-ng:

aireplay-ng -0 0 -a 02:00:00:00:00:01 sta2-mon0

Со стороны клиента был записан файл с трафиком. На основании содержимого файла сделан вывод что инжекция кадров в сеть возможна.

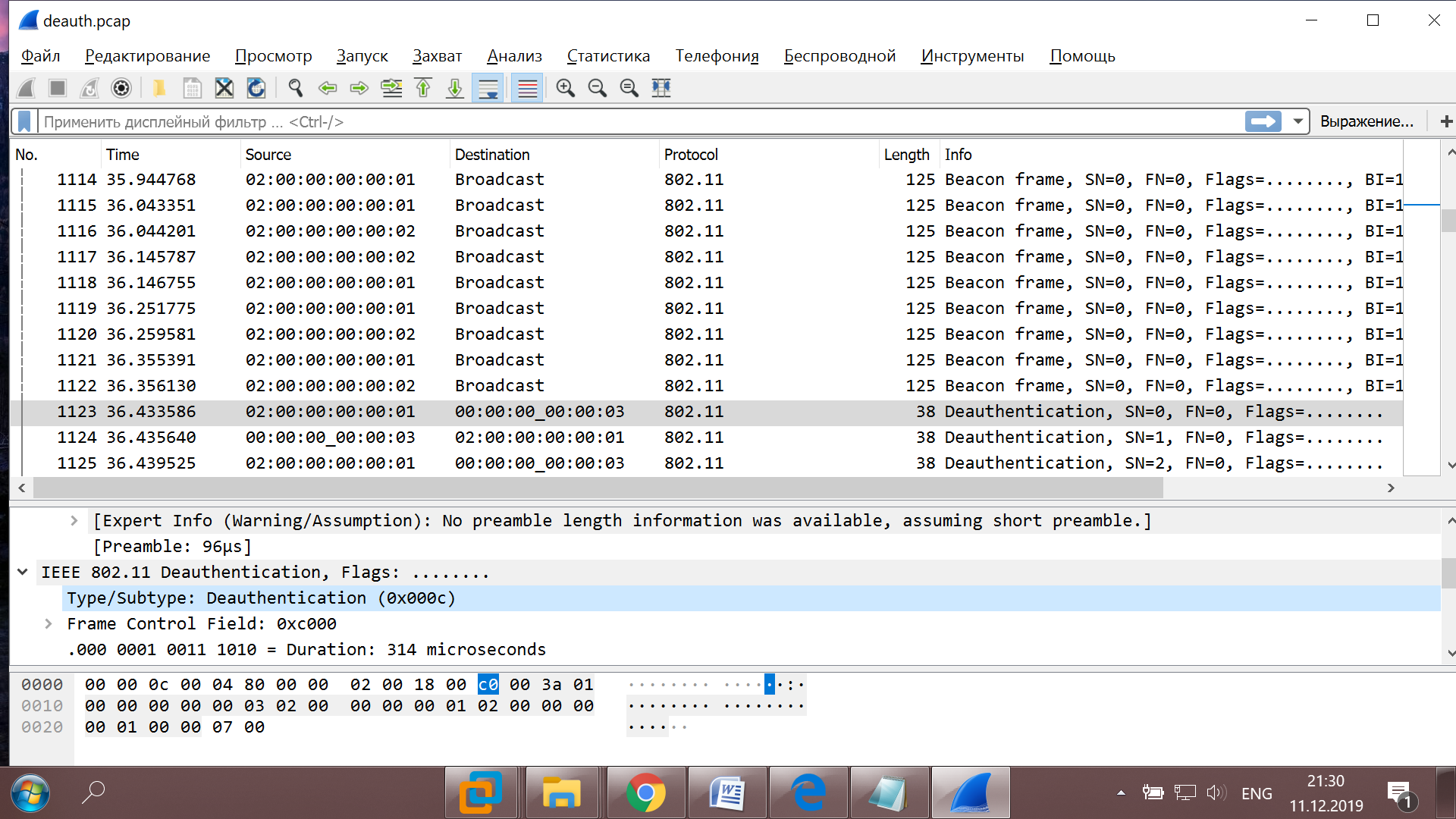


Рисунок 2.1 - Инжектирование кадров деаутентификации

*Исследование работы режима мониторинга*

Целью является написание скрипта для перевода точки доступа в режим мониторинга, а также проверка корректной работы устройства в качестве сниффера. Смоделирована точка доступа и подключенный к ней клиент. Узлы расположены на одном частотном канале. Задана активность между ними при помощи утилиты ping.

Создан сетевой интерфейс в режиме беспроводной станции и переведен в режим мониторинга. Настроен на частотный канал с точкой и клиентом. Записан трафик взаимодействия точки доступа и клиента при помощи утилиты tcpdump.

На основании содержимого записи трафика выяснено, что интерфейс мониторинга получает только широковещательные beacon кадры от точки доступа. Поэтому для дальнейшей работы с режимом мониторинга необходима модификация библиотеки mac80211\_hwsim.

Проверена возможность задания координат для устройства с интерфейсом мониторинга. Координаты назначаются так же, как любому другому узлу сети, при помощи метода setPosition. Исследована возможность задания вероятности потерь кадров от точек доступа и клиентов к устройству с интерфейсом в режиме мониторинга. Для этого необходимо использовать модели потерь данных (см. п. 2.3.6)

## Построение различных конфигураций при помощи MininetWifi

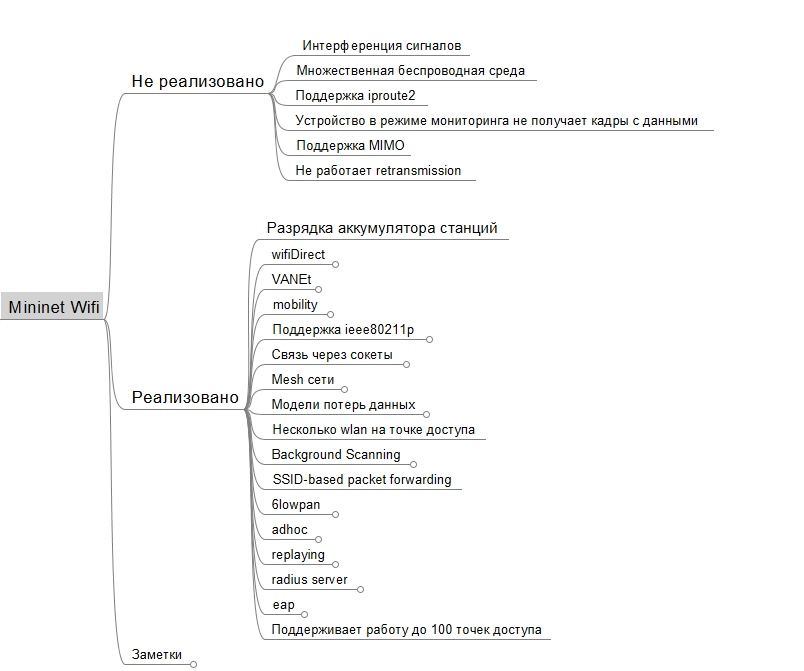


Рисунок 2.2 – Основные возможности Mininet Wi-Fi

### WifiDirect

Wi-Fi Direct - новый стандарт беспроводной передачи данных, позволяющий устройствам соединяться между собой напрямую без дополнительного промежуточного звена в виде роутера.

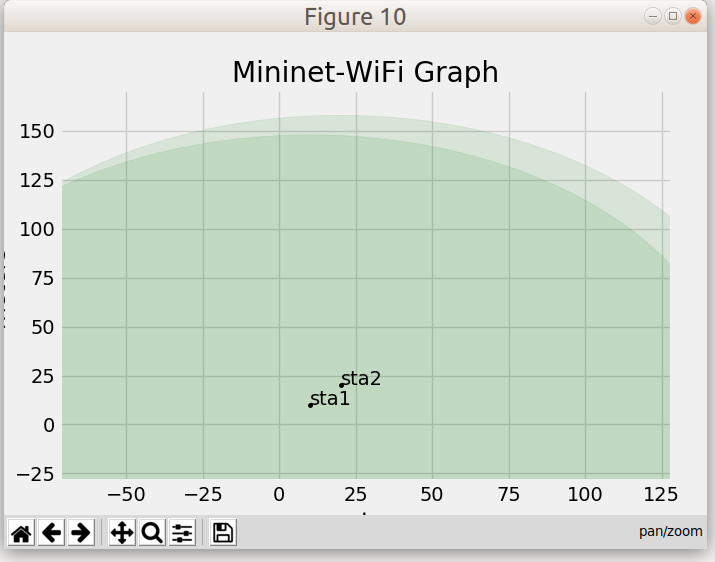


Рисунок 2.3 - Расположение устройств



Рисунок 2.4 - Поиск устройств Wi-Fi Direct



Рисунок 2.5 - Подключение устройств Wi-Fi Direct

Последовательность команд описана в листинге 2.4:

Листинг 2.4

xterm sta1 sta2

wpa\_cli -i sta2-wlan0

p2p\_find

p2p\_stop\_find

p2p\_peers

p2p\_connect 02:00:00:00:01:00 pin auth

p2p\_connect 02:00:00:00:00:00 60631480 auth

### VANETs

Автомобильные самоорганизующиеся сети (VANETs) как и мобильные самоорганизующиеся сети возникают путем спонтанного создания беспроводной сети для обмена данными между транспортными средствами. VANETs используются в архитектурах связи: машина-машина, машина-дорога. Такие сети обеспечивают безопасность дорожного движения, навигацию и другие придорожные услуги. VANETs являются ключевой частью системы интеллектуальных транспортных систем.

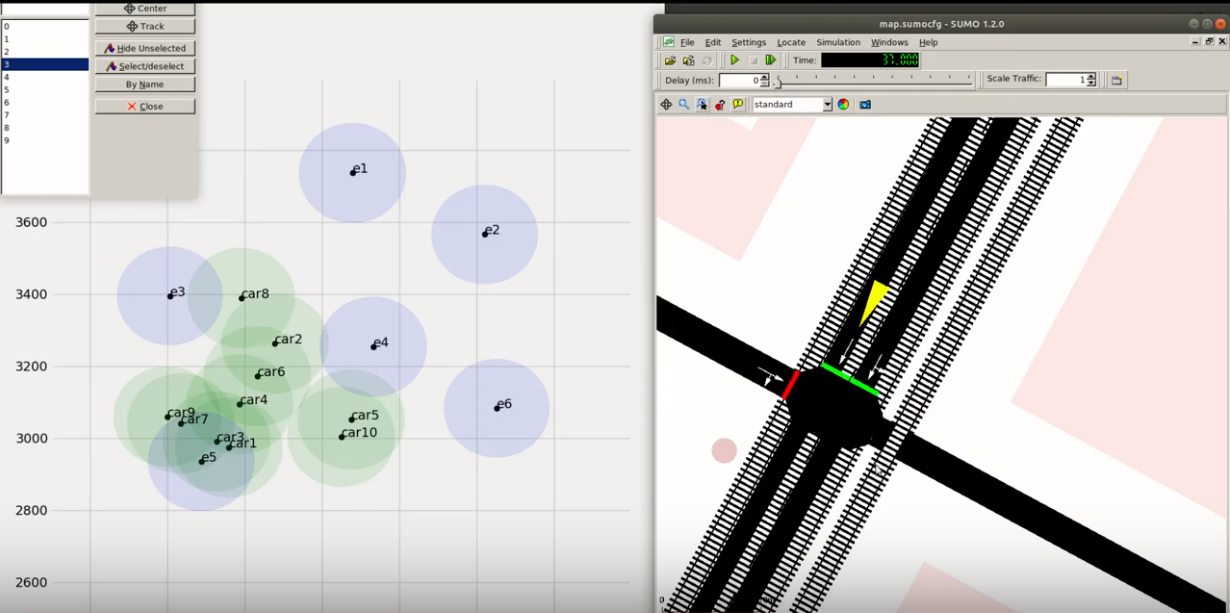
Рисунок 2.6 - Запущено sumo-gui

Рисунок 2.7 - Движение автомобилей и объединение их в сети

### Mobility

Благодаря использованию mobility стала возможна имитация движущихся узлов - клиентов точек доступа. Появилась возможность движения станций как в заданном, так и в случайных направлениях.

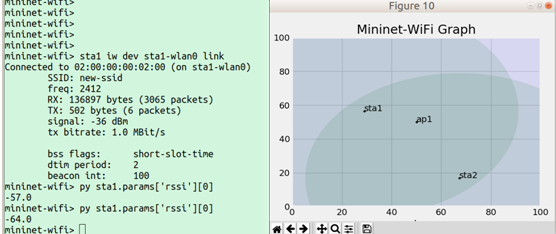


Рисунок 2.8 - Изменение характеристик сигнала при движении клиента

### Связь через сокеты

Предоставляет возможность управления стендом удаленно. При помощи подключения через сокет реализованы команды (см. листинг 2.5)

Листинг 2.5

get.{УЗЕЛ}.{ПАРАМЕТР}

get.sta2.position

get.sta2.rssi

get.sta2.range

get.{УЗЕЛ}.{ПАРАМЕТР}[.ЗНАЧЕНИЕ]

set.sta1.setPosition.30,30,0

set.sta1.setRange.300

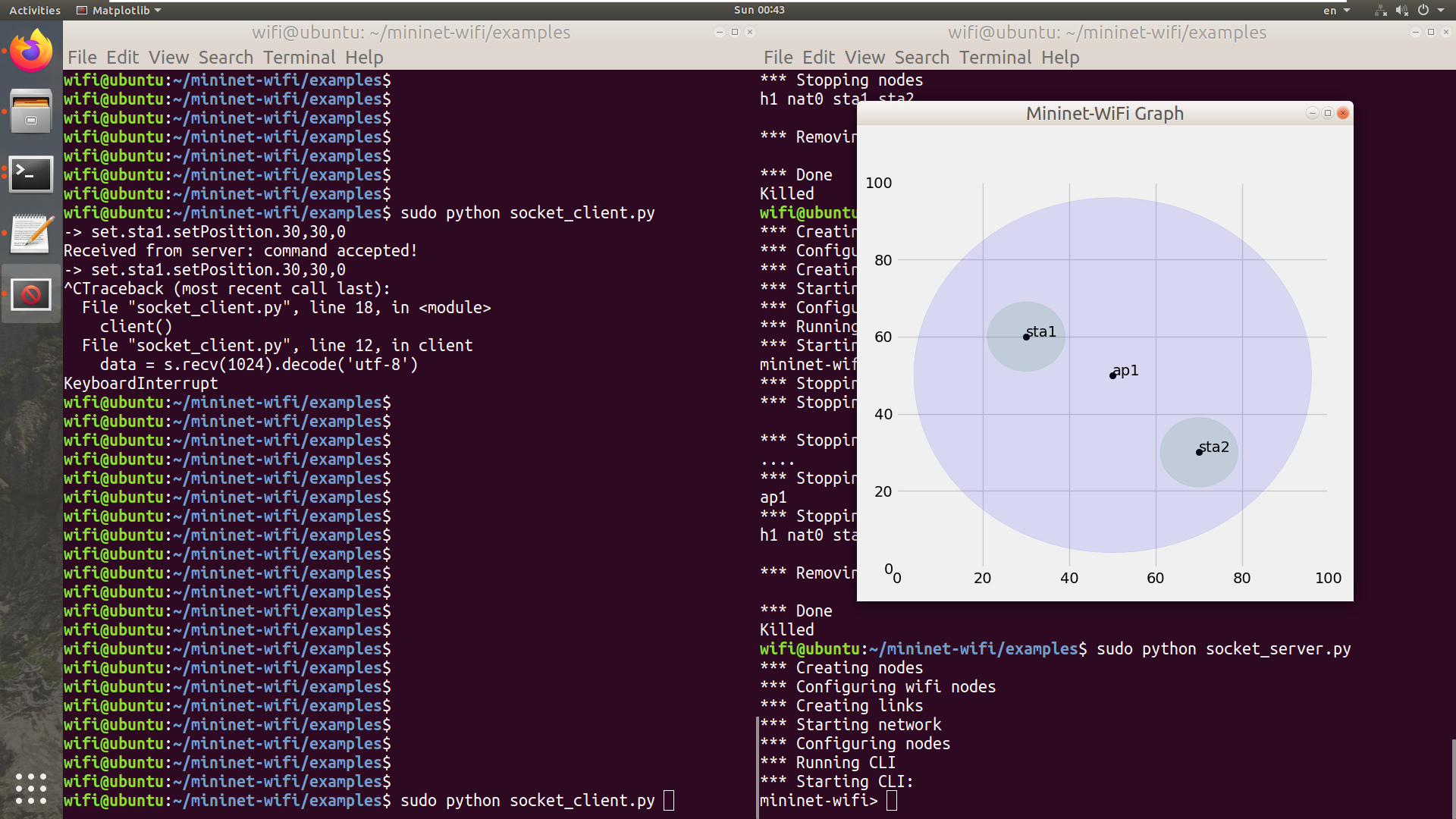


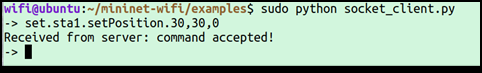
Рисунок 2.9 - Первоначальное состояние узлов

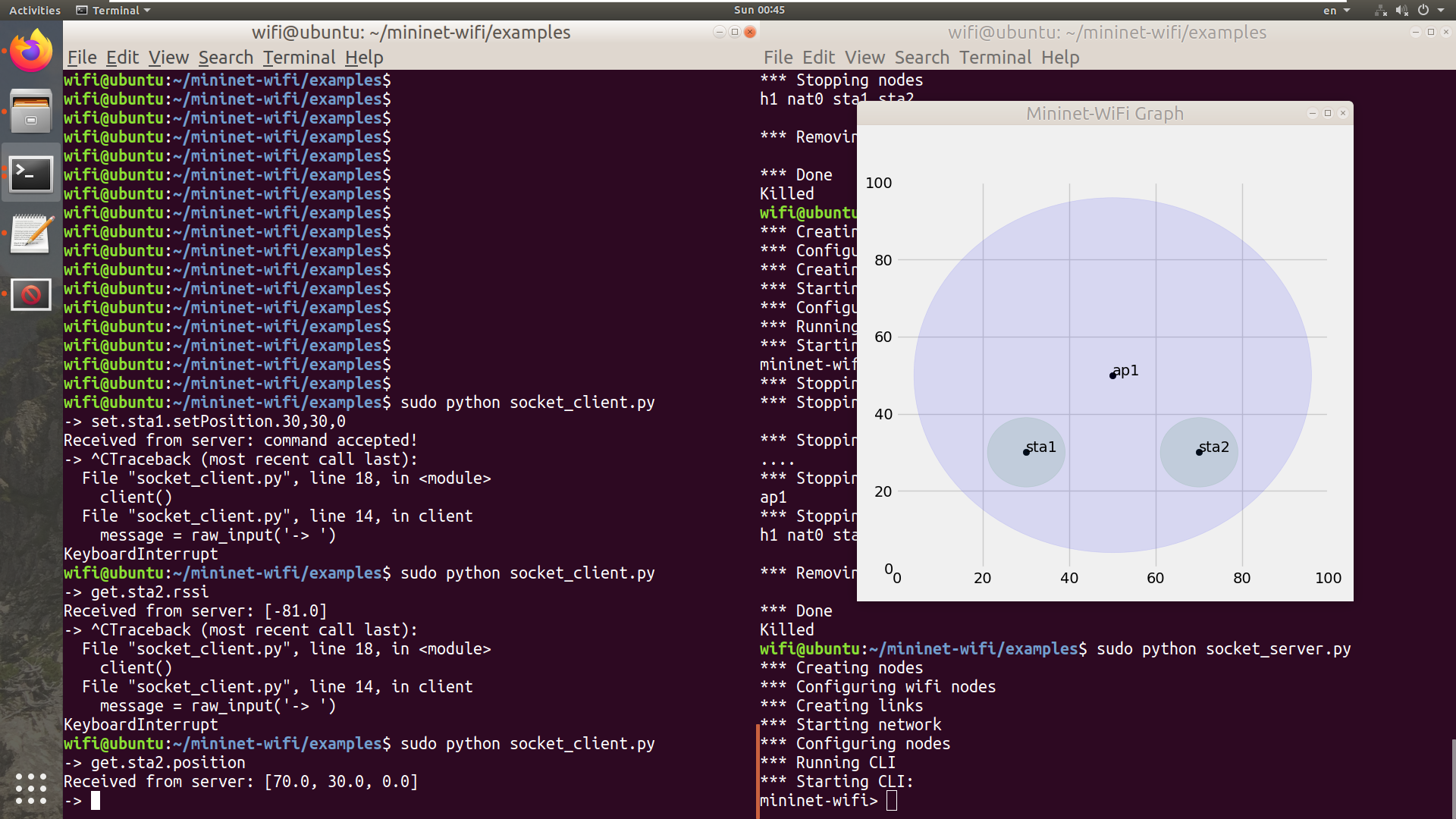
Рисунок 2.10 - Перемещение узла со стороны клиента

Рисунок 2.11 - Результат перемещения

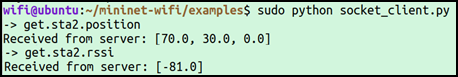


Рисунок 2.12 - Прочие команды

### Mesh сети

Mesh-сеть - сетевая топология компьютерной сети, построенная на принципе ячеек, в которой рабочие станции сети соединяются друг с другом и способны принимать на себя роль коммутатора для остальных участников. При такой топологии реализуется высокая отказоустойчивость. Узлы соединяются по принципу «каждый с каждым». Большое количество связей обеспечивает широкий выбор маршрута трафика внутри сети. Обрыв одного соединения не нарушит функционирования сети в целом.

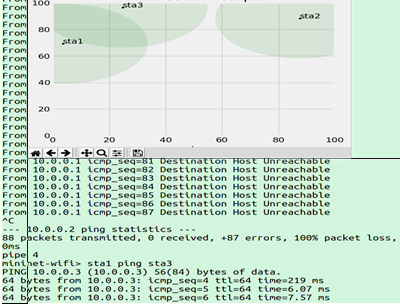


Рисунок 2.13 - Достижимое состояние

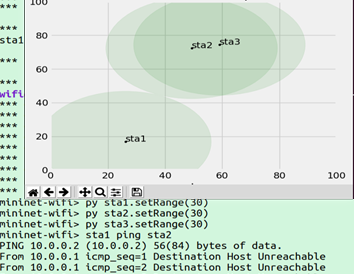


Рисунок 2.14 - Недостижимое состояние

### Модели потерь данных

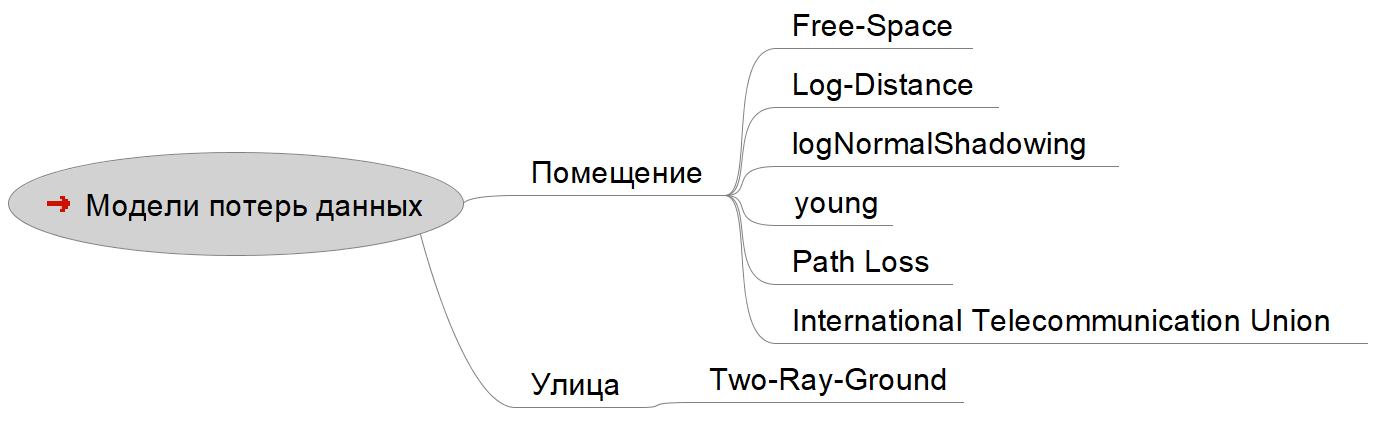


Рисунок 2.15 - Модели потерь данных

За определение отношения сигнал - шум между узлами отвечает класс SNRLink расположенный в mininet-wifi/mn\_wifi/wmediumdConnector.py

Основные параметры рассмотрены в таблице 2.1:

Таблица 2.1 - Параметры влияющие на сигнал

|  |  |
| --- | --- |
| Параметр | Назначение |
| rssi | (Received Signal Strength Indicator) - мощность радиосигнала. В стандарте IEEE 802.11 RSSI это индикатор уровня мощности, принимаемого антенной, выраженный абстрактных единицах, выбираемых производителем беспроводного оборудования. Чем выше число RSSI, тем сильнее сигнал. |
| model | Модели распространения радиоволн рассматриваются в п.п. 2.3.6.1-2.3.6.5. |
| exp | Exponent - показатель затухания волны. Чем выше, тем хуже качество связи. |
| sL | System Loss - системные потери. Чем выше коэффициент, тем хуже качество сигнала |

Продолжение таблицы 2.1

|  |  |
| --- | --- |
| lF | Floor penetration loss factor - коэффициент потерь при проникновении является эмпирической константой. Он зависит от количества этажей, в которые должны проникнуть волны. |
| pL | Power Loss Coefficient - коэффициент потери мощности - число, которое представляет собой баланс между увеличением полезной тепловой энергии и уменьшением вырабатываемой электрической энергии. |
| nFloors | Number of floors - количество этажей при передаче данных в здании. |
| gRandom | Gaussian random variable - гауссовская случайная величина |
| variance | Дисперсия |
| noise\_threshold | Порог шума, при достижении которого кадры начинают отбрасываться без проверок. |
| cca\_threshold | Порог, использующий cпособность приемника обнаруживать отличную от wifi энергию в рабочем канале и отменять передачу данных. |

#### International Telecommunication Union

Модель распространения радиоволн, которая оценивает потери в комнате или закрытой области внутри здания, ограниченными стенами любой формы. Подходит для приборов, предназначенных для использования внутри помещений. Эта модель приблизительно рассчитывает общие потери в пути, которые может возникнуть в помещении.

#### Free-Space (friis loss)

Потери в свободном пространстве - это потеря мощности сигнала электромагнитной волны, которая может возникнуть в результате прохождения через свободное пространство, без каких-либо препятствий поблизости, которые могли бы вызвать отражение или дифракцию. Используется для прогнозирования потерь при распространении волны в широком диапазоне сред, с беспрепятственным прохождением между передатчиком и приемником. Так волны распространяются в вакууме и в идеальных условиях, например, радиосвязь между спутниками.

#### logNormalShadowing

Если на пути прохождения сигнала есть какие-либо объекты (например, здания или деревья), некоторая часть передаваемого сигнала теряется из-за поглощения, отражения, рассеяния и дифракции. Этот эффект называется затенением.

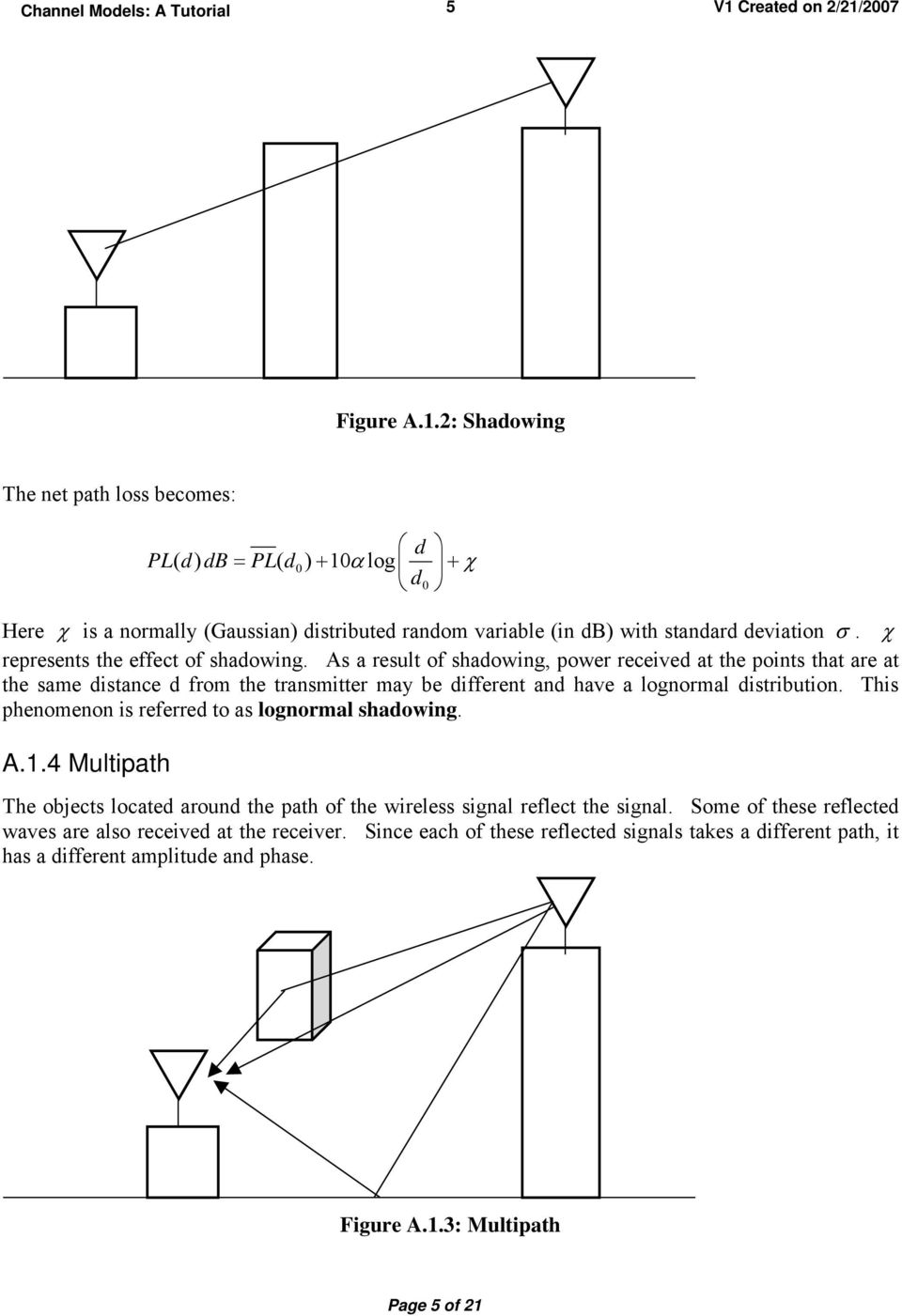


Рисунок 2.16 - Эффект затенения

Как показано на рисунке 2.16, если бы базовая антенна была источником света, среднее здание отбрасывало бы тень на абонентскую антенну. Отсюда и название - затенение. Рассмотрены показатели rssi при различных положениях узлов.

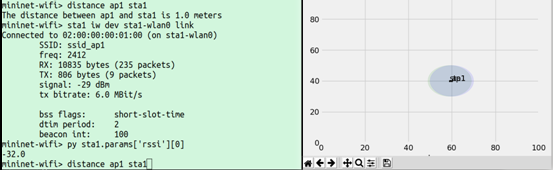


Рисунок 2.17 - Хороший уровень сигнала

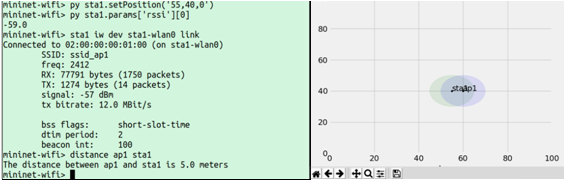


Рисунок 2.18 - Средний уровень сигнала

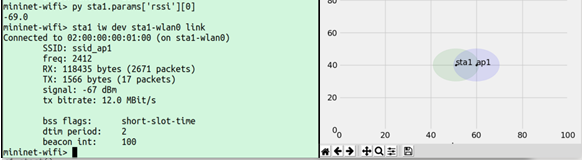


Рисунок 2.19 - Плохой уровень сигнала

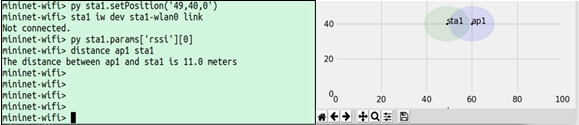


Рисунок 2.20 - Разрыв связи

#### Log-Distance Path Loss

Модель затухания волны с логарифмическим расстоянием - это модель распространения радиоволн, которая прогнозирует потери, образованные столкновениями с препятствиями (зданиями, густонаселенные районами) при передаче волны на большие расстояния.

#### Two-Ray-Ground

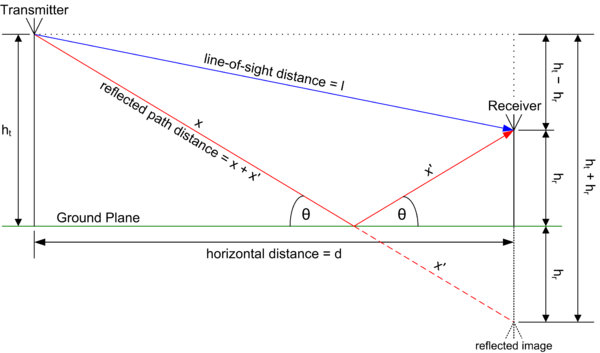


Рисунок 2.21 - Схема передачи данных

Модель двухлучевого отражения от земли - это модель распространения радиоволн, которая предсказывает потери на трассе между передающей антенной и приемной антенной, когда они находятся в пределах линии видимости. Как правило, две антенны имеют разную высоту. Принятый сигнал, имеющий два компонента, компонент линии видимости и компонент многолучевого распространения, образован преимущественно одной отраженной от земли волной.

### Adhoc

Беспроводная ad-hoc-сеть (беспроводная динамическая сеть, беспроводная самоорганизующаяся сеть) - децентрализованная беспроводная сеть, не имеющая постоянной структуры. Клиентские устройства соединяются «на лету», образуя собой сеть. Каждый узел сети пытается переслать данные, предназначенные другим узлам. При этом определение того, какому узлу пересылать данные, производится динамически, на основании связности сети.

Был рассмотрен протокол маршрутизации batman. Основной особенностью B.A.T.M.A.N. является децентрализация сведений о лучшем маршруте в сети - ни один узел не владеет всеми данными. С использованием этой техники отпадает необходимость в распространении информации об изменениях в сети на все узлы. Каждый узел хранит информацию только о «направлении», из которого поступают данные, и так же их отправляет. Таким образом, узлы передают друг другу пакеты по динамически создаваемым маршрутам.



Рисунок 2.22 - Создание сети

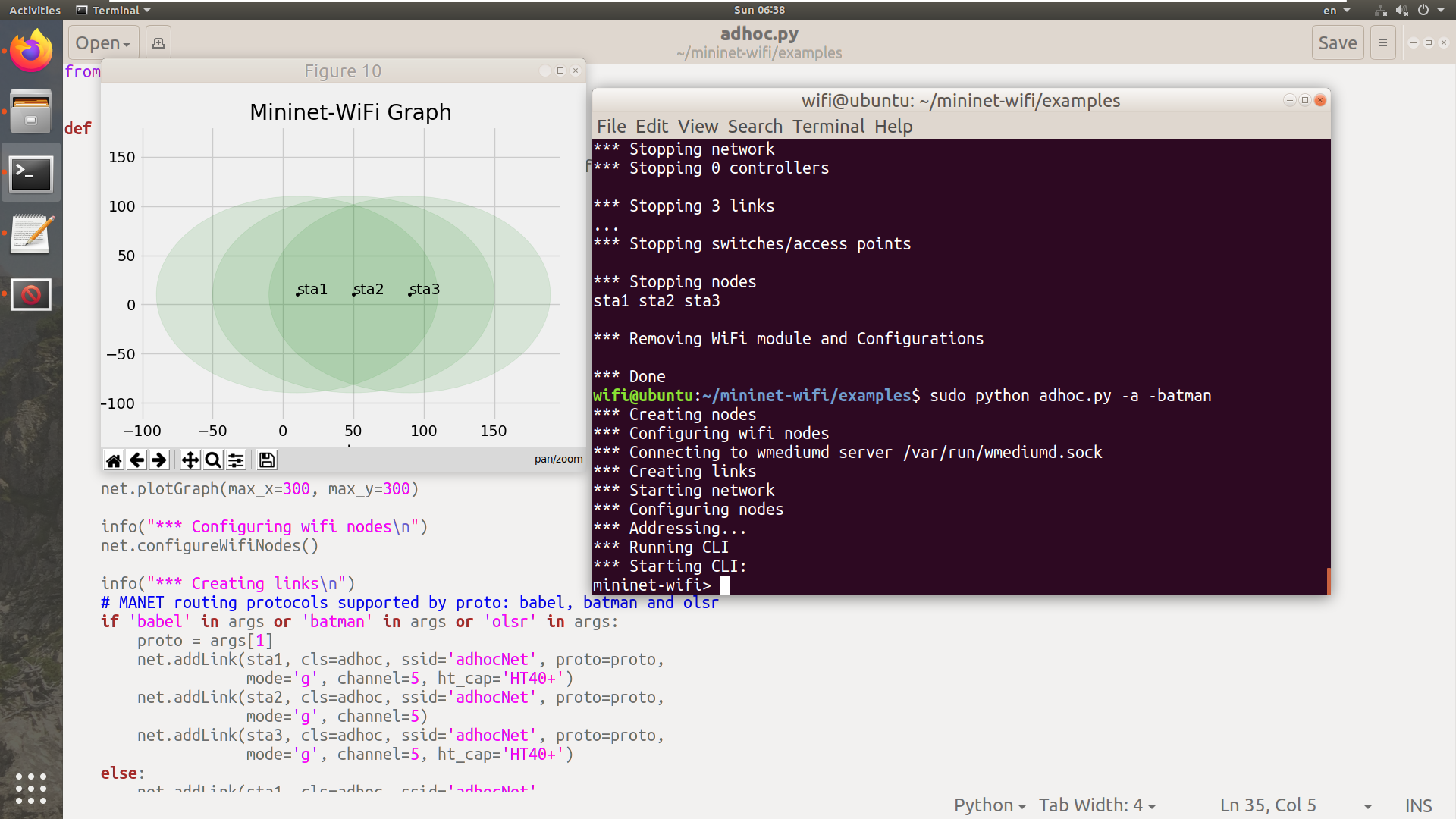


Рисунок 2.23 - Расположение узлов

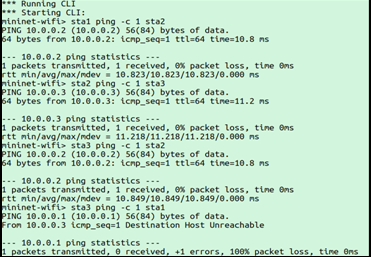


Рисунок 2.24 - Коммуникация узлов

## Выводы по второй главе

В результате исследования возможностей, написания скриптов и построения сложных конфигураций, стало возможно использование всего функционала беспроводных устройств, работающих в реальном мире.

Также данное программное средство позволяет воспроизводить множество экспериментов, трудно реализуемых в реальном мире в условиях отсутствия специализированного оборудования. Можно сделать вывод о том, что на базе данного эмулятора удастся выполнить поставленные задачи, такие как воспроизведение поведения сложных устройств и сведение работы эмулятора к результатам в реальной жизни.Некоторые возможности, такие как перехват всех пакетов, интерференция точек доступа, повторные передачи кадров - требуют дальнейшей доработки.

Заключение

После ознакомления с документацией и изучения эмулятора стало возможно воспроизведение различных сложных топологий и конфигураций узлов. Надстройка над mac80211\_hwsim и поддержка wmediumd позволяет использовать практически весь функционал, реализуемый в ОС Linux.

Дальнейшей задачей будет проведение экспериментов с реальными устройствами и сведение работы эмулятора к результатам в реальной жизни. Для этого потребуется изменение параметров моделей потерь данных, а также использование пользовательских таблиц с показателями rssi для настройки точек доступа. Это позволит в дальнейшем проводить эксперименты без участия аппаратной составляющей.

Список литературы

1. Ramon dos Reis Fontes, Christian Esteve Rothenberg The User Manual [Электронный ресурс] URL: https://github.com/ramonfontes/mn-wifi-book-pt (дата обращения: 04.12.2019)
2. Ramon dos Reis Fontes, Christian Rodolfo Esteve Rothenberg Mininet-WiFi project [Электронный ресурс] URL: https://github.com/intrig-unicamp/mininet-wifi (дата обращения: 14.11.2019)
3. Ramon dos Reis Fontes Emulator for Software-Defined Wireless Networks [Электронный ресурс] URL: https://mininet-wifi.github.io/ (дата обращения: 21.11. 2019)
4. Fontes, Ramon R., et al. ”Mininet-WiFi: Emulating software-deﬁned wireless networks.” Network and Service Management (CNSM), 2015 11th International Conference on. IEEE, 2015.