«РАЗРАБОТКА СИСТЕМЫ МОДЕЛИРОВАНИЯ ВИРТУАЛЬНОЙ СЕТИ»

Автор: Кимов А.М.

Москва 2024 г.

**СОДЕРЖАНИЕ**

[**1. Обзор темы 4**](#_Toc168779487)

[**1.1 Основные концепции виртуальных сетей 4**](#_Toc168779488)

[**1.2 Анализ существующих решений 5**](#_Toc168779489)

[**1.3 Технологии и методы моделирования сетей 8**](#_Toc168779490)

[**2. Теоретические основы 9**](#_Toc168779491)

[**2.1 Основы построения локальных сетей 9**](#_Toc168779492)

[**2.2 Принципы виртуализации сетей 9**](#_Toc168779493)

[**3. Проектирование системы 10**](#_Toc168779494)

[**3.1 Технические требования 10**](#_Toc168779495)

[**3.2 Архитектура системы 11**](#_Toc168779496)

[**3.3 Выбор программных средств и технологий 11**](#_Toc168779497)

[**4. Реализация системы 12**](#_Toc168779498)

[**4.1 Описание функциональных модулей 12**](#_Toc168779499)

[**4.2 Процесс разработки 13**](#_Toc168779500)

[5. Интерфейс программы 14](#_Toc168779501)

[**5.1 Обзор интерфейса 14**](#_Toc168779502)

[**5.2 Разбор кейсов 15**](#_Toc168779503)

[6. ВЫВОДЫ 21](#_Toc168779504)

**ВВЕДЕНИЕ**

#### Актуальность темы

В современном мире виртуальные сети становятся неотъемлемой частью информационной инфраструктуры как малых предприятий, так и крупных корпораций. Они позволяют эффективно управлять ресурсами, обеспечивать безопасность данных и повышать гибкость работы сети. Актуальность исследования обусловлена растущими требованиями к быстроте и точности моделирования сетевых конфигураций, что делает разработку специализированного программного обеспечения в данной области весьма востребованной.

Цель проекта:

Целью данной работы является разработка системы моделирования виртуальной локальной сети, которая позволит пользователям создавать, модифицировать и управлять различными сетевыми конфигурациями с использованием таких устройств, как ПК, Роутер и Сервер.

Задачи исследования:

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить теоретические основы построения локальных сетей и виртуализации.
2. Проанализировать существующие программные решения для моделирования сетей.
3. Разработать архитектуру и функциональные модули системы.
4. Реализовать программное обеспечение на языке C# в среде Microsoft Visual Studio.
5. Провести тестирование системы и оценить её эффективность.

Обоснование выбора инструментов разработки

Язык программирования C# был выбран для реализации данного проекта из-за его высокой производительности, обширных возможностей для работы с сетевыми приложениями и хорошей интеграции с платформой .NET, что обеспечивает надёжность и масштабируемость разрабатываемого решения. Среда разработки Microsoft Visual Studio предоставляет удобные инструменты для написания, отладки и тестирования кода, что значительно ускоряет процесс разработки и повышает качество конечного продукта.

### **1. Обзор темы**

## **1.1 Основные концепции виртуальных сетей**

В данном подразделе будет проведен обзор ключевых концепций, связанных с виртуальными сетями. Рассматриваются основные принципы виртуализации сетей на уровнях 2 и 3.

**Виртуальные сети (VLAN)**: Одним из основных методов виртуализации сетей является создание виртуальных локальных сетей (VLAN). VLAN позволяют разделить физическую сеть на несколько логических сегментов, что обеспечивает повышенную безопасность и эффективное управление трафиком.

**Виртуальные частные сети (VPN)**: VPN представляют собой технологию, которая позволяет создавать безопасные соединения между удаленными сетями через общественную сеть, такую как интернет. VPN обеспечивают защищенную передачу данных и могут использоваться для организации удаленного доступа к корпоративным ресурсам.

**Виртуальная маршрутизация и пересылка (VRF)**: VRF позволяют создавать несколько виртуальных экземпляров маршрутизации внутри одного физического маршрутизатора. Это позволяет разделять сетевые ресурсы между различными группами пользователей или приложений, что повышает безопасность и управляемость сети.

В данном подразделе будет также проведена оценка преимуществ и недостатков каждой из этих концепций, их применимости и влияния на процесс моделирования виртуальных сетей.

## **1.2 Анализ существующих решений**

Этот подраздел посвящен анализу существующих программных решений для моделирования сетей. Рассматриваются такие инструменты, как Cisco Packet Tracer, GNS3, EVE-NG и другие.

**Cisco Packet Tracer**: Packet Tracer - это инструмент, разработанный компанией Cisco для моделирования сетей на основе оборудования Cisco. Он позволяет создавать сетевые топологии, настраивать устройства и симулировать трафик.

**Функциональность**: Packet Tracer предоставляет широкий набор функций для моделирования сетей, включая создание сетевых топологий, настройку устройств (маршрутизаторов, коммутаторов, ПК и др.), симуляцию трафика и отладку сетевых конфигураций.

**Поддерживаемые устройства и протоколы**: Packet Tracer включает в себя множество устройств Cisco, таких как маршрутизаторы серий 800, 1900, 2900, коммутаторы серий 2960 и 3560, а также ПК и другие устройства. Поддерживаемые протоколы включают IPv4, IPv6, OSPF, EIGRP, VLAN, VPN и многие другие.

**Возможности моделирования различных сетевых сценариев**: Packet Tracer предоставляет возможность моделировать разнообразные сетевые сценарии, начиная от простых домашних сетей до сложных корпоративных инфраструктур с множеством подсетей, VPN и сегментации трафика.

**Преимущества**:

1. Простой и интуитивно понятный интерфейс.
2. Бесплатное распространение.
3. Поддержка широкого спектра устройств и протоколов Cisco.
4. Возможность использования в образовательных целях для изучения сетевых концепций и практических навыков.

**Недостатки**:

1. Ограниченные возможности для моделирования некоторых продвинутых функций и устройств Cisco.
2. Не всегда точное воспроизведение поведения реальной сети.

**GNS3**: GNS3 - это сетевой эмулятор, который позволяет создавать виртуальные сети на основе реальных образов устройств. Он поддерживает широкий спектр маршрутизаторов и коммутаторов различных производителей.

**Функциональность**: GNS3 предоставляет расширенные возможности для моделирования сетей, включая поддержку реальных образов устройств Cisco, Juniper, Huawei и других производителей, интеграцию с внешними виртуальными машинами, а также возможность создания масштабируемых сетевых топологий.

**Поддерживаемые устройства и протоколы**: GNS3 поддерживает широкий спектр маршрутизаторов и коммутаторов различных производителей, включая Cisco, Juniper, Huawei, MikroTik и др. Поддерживаемые протоколы аналогичны Cisco Packet Tracer.

**Возможности моделирования различных сетевых сценариев**: GNS3 предоставляет более расширенные возможности для моделирования сложных сетевых сценариев, включая взаимодействие с внешними сетями, масштабирование сетевых топологий и создание тестовых сред для разработки и тестирования сетевых приложений.

**Преимущества**:

1. Поддержка реальных образов устройств, что позволяет более точно моделировать поведение реальных сетей.
2. Большие возможности для масштабирования сетевых топологий и интеграции с внешними ресурсами.
3. Активное сообщество пользователей и разработчиков.

**Недостатки**:

1. Более сложная настройка и использование по сравнению с Packet Tracer.
2. Требует более мощных ресурсов компьютера.

**EVE-NG**: EVE-NG (Emulated Virtual Environment - Next Generation) - это сетевой эмулятор, основанный на виртуализации, который позволяет создавать сложные сетевые топологии с использованием виртуальных образов устройств.

**Функциональность**: EVE-NG (Emulated Virtual Environment - Next Generation) предоставляет возможность создания сложных виртуальных сетевых топологий с использованием виртуальных образов устройств. Он также поддерживает интеграцию с облачными провайдерами и внешними виртуальными машинами.

**Поддерживаемые устройства и протоколы**: EVE-NG поддерживает образы устройств Cisco, Juniper, Arista, Palo Alto Networks, Fortinet и других производителей. Поддерживаемые протоколы аналогичны GNS3 и Packet Tracer.

**Возможности моделирования различных сетевых сценариев**: EVE-NG предоставляет возможность создания разнообразных сетевых топологий, включая сети различных масштабов, интеграцию с облачными сервисами и виртуальными машинами, а также возможность создания собственных образов устройств.

**Преимущества**:

1. Поддержка широкого спектра устройств и производителей.
2. Интеграция с облачными сервисами и внешними виртуальными машинами.
3. Гибкие возможности для создания и тестирования сетевых сценариев.

**Недостатки**:

1. Сложная настройка и использование по сравнению с Packet Tracer.
2. Требует высоких требований к ресурсам компьютера.

## **1.3 Технологии и методы моделирования сетей**

В этом подразделе будет проведен обзор основных технологий и методов моделирования сетей.

**Сетевое симулирование**: Сетевое симулирование - это метод моделирования сетей, при котором используются математические модели для анализа поведения сети в различных условиях. С помощью сетевого симулирования можно оценить производительность, надежность и безопасность сети до ее фактической реализации.

**Эмуляция сети**: Эмуляция сети - это метод, при котором реальные сетевые устройства и протоколы заменяются программными аналогами. Это позволяет создавать точные копии реальных сетей и тестируть их в виртуальной среде.

**Виртуализация сети**: Виртуализация сети - это метод, при котором физическая сеть разделяется на несколько виртуальных сетей, каждая из которых работает независимо от остальных. Виртуализация сети позволяет повысить гибкость, масштабируемость и безопасность сети.

### **2. Теоретические основы**

## **2.1 Основы построения локальных сетей**

Локальная сеть (LAN) представляет собой сеть компьютеров и других устройств, объединенных на небольшой территории, такой как офис, школа или дом. Основой любой LAN является среда передачи данных, чаще всего это сетевой кабель (Ethernet), но также могут использоваться беспроводные технологии, такие как Wi-Fi. Каждое устройство в LAN имеет уникальный сетевой адрес (MAC-адрес), который позволяет идентифицировать его в сети.

#### Типы сетевых устройств и их функции

Сетевые устройства выполняют различные функции в LAN и играют важную роль в обеспечении её работоспособности. Среди таких устройств можно выделить следующие:

* **Компьютеры (ПК)**: Являются основными узлами в сети, на которых работают прикладные программы и осуществляется обмен данными.
* **Роутеры**: Обеспечивают маршрутизацию данных между различными сегментами сети и между LAN и внешними сетями, такими как Интернет.
* **Серверы**: Предоставляют различные сервисы и ресурсы для клиентских устройств, такие как файловые хранилища, электронная почта, веб-серверы и базы данных.

## **2.2 Принципы виртуализации сетей**

Виртуализация сети — это процесс создания виртуальной абстракции сетевых ресурсов, таких как сетевые устройства, каналы передачи данных и сетевые сервисы. Основные принципы виртуализации сетей включают:

* **Изоляция ресурсов**: Обеспечение независимости виртуальных сетей друг от друга и от реальных физических ресурсов.
* **Абстракция**: Предоставление удобного интерфейса для управления виртуальными сетевыми ресурсами без необходимости знания о их физической реализации.
* **Эффективное использование ресурсов**: Минимизация издержек на создание и поддержание виртуальных сетей путем их мультиплексирования на общих физических ресурсах.

Включение данных теоретических основ в проект позволит эффективно проектировать и реализовывать систему моделирования виртуальных сетей, учитывая основные принципы функционирования сетевых систем.

### **3. Проектирование системы**

## **3.1 Технические требования**

Перед разработкой системы были определены следующие технические требования:

1. **Поддержка различных типов сетевых устройств**: Система должна обеспечивать возможность создания и модификации различных типов устройств, таких как ПК, Роутер и Сервер, с возможностью настройки их параметров и связей между собой.
2. **Интерактивный пользовательский интерфейс**: Важным требованием является наличие удобного и интуитивно понятного пользовательского интерфейса, позволяющего пользователям легко создавать, изменять и управлять сетевыми конфигурациями.
3. **Гибкость конфигураций**: Система должна обеспечивать гибкость в создании различных конфигураций сети, включая возможность добавления новых устройств, изменения их параметров и настройки сетевых соединений.
4. **Элементарное взаимодействие с системой**: Важным требованием является возможность взаимодействовать с построенной системой посредством аналогии командной строки. Данный пункт позволит добавить приложению дополнительный слой взаимодействия с системой, приближающий виртуальную модель к реальной сети.

## **3.2 Архитектура системы**

Система разделена на следующие основные компоненты:

1. **Модель устройства**: Каждое сетевое устройство представлено в системе в виде отдельного объекта, который содержит информацию о его типе, параметрах и текущем состоянии.
2. **Модель соединения**: Для описания связей между устройствами используется модель соединения, которая определяет тип и параметры соединения между устройствами.
3. **Пользовательский интерфейс**: Для взаимодействия с системой пользователи используют графический интерфейс, который позволяет создавать, изменять и управлять сетевыми конфигурациями.
4. **Логика приложения**: Этот компонент отвечает за обработку действий пользователя, управление моделью данных и обеспечение связи между пользовательским интерфейсом и внутренней моделью системы.

## **3.3 Выбор программных средств и технологий**

Для реализации системы были выбраны следующие программные средства и технологии:

1. **Язык программирования C#**: Для написания основной логики приложения был выбран язык C#, который обеспечивает высокую производительность, удобство разработки и интеграцию с платформой .NET.
2. **Среда разработки Microsoft Visual Studio**: Для разработки и отладки приложения была использована среда разработки Visual Studio, которая предоставляет широкие возможности для написания, отладки и тестирования кода.
3. **Windows Forms**: Для создания графического пользовательского интерфейса была выбрана технология Windows Forms, которая обеспечивает быструю и простую разработку интерфейса приложения.
4. **.NET Framework**: Для обеспечения совместимости и интеграции с другими компонентами системы был использован .NET Framework, который предоставляет широкий набор библиотек и инструментов для разработки приложений под Windows.

### **4. Реализация системы**

## **4.1 Описание функциональных модулей**

1. **Модуль управления устройствами**
   * Реализует функции добавления, удаления и модификации устройств (ПК, Роутер, Сервер) в виртуальной сети.
   * Обеспечивает возможность изменения параметров устройств, таких как IP-адреса, название устройства, тип соединения и другие.
2. **Модуль управления соединениями**
   * Позволяет пользователю устанавливать соединения между устройствами в сети.
3. **Модуль визуализации**
   * Отображает текущую конфигурацию виртуальной сети в удобном графическом виде.
   * Обеспечивает возможность интерактивного взаимодействия с устройствами и соединениями, включая перемещение и изменение свойств.
4. Модуль командной строки
   * Позволяет использовать стандартные команды для взаимодействия устройств между собой внутри сети. Например, команда ping.

## **4.2 Процесс разработки**

1. **Проектирование интерфейса**
   * Были разработаны макеты пользовательского интерфейса для каждого функционального модуля с учетом простоты использования и интуитивной навигации.
2. **Написание кода**
   * Вся логика приложения была реализована с использованием языка программирования C# и платформы .NET Framework.
   * Каждый модуль был разбит на отдельные классы и методы для обеспечения четкой структуры кода и возможности повторного использования.
3. **Тестирование**
   * Проводились различные виды тестирования, включая модульное тестирование каждого функционального модуля, интеграционное тестирование и тестирование пользовательского интерфейса.
   * Выявленные ошибки и недочеты исправлялись незамедлительно для обеспечения стабильной работы приложения.

Реализация системы моделирования виртуальной сети была успешно завершена, обеспечивая пользователям интуитивно понятный интерфейс и широкие возможности по управлению сетевыми конфигурациями. Процесс разработки был осуществлен с соблюдением всех технических требований и стандартов качества, что позволяет гарантировать эффективную и надежную работу приложения.

# 5. Интерфейс программы

## **5.1 Обзор интерфейса**

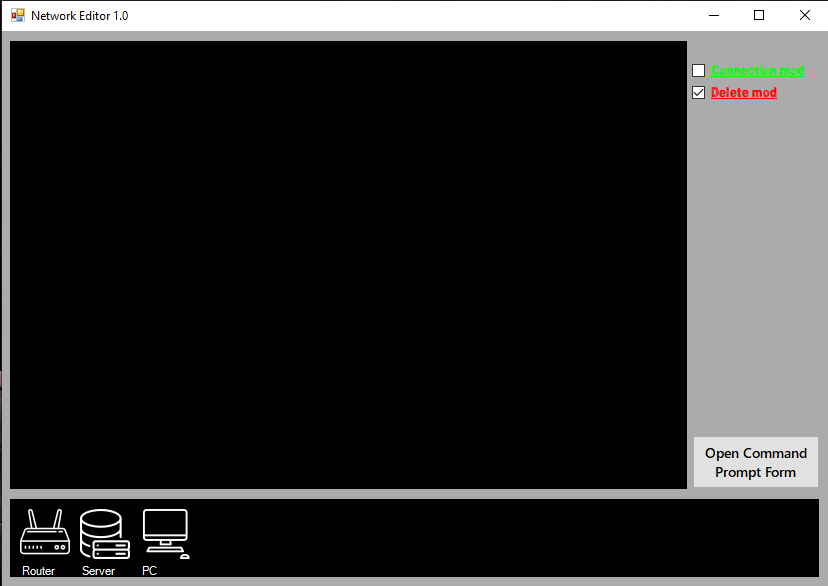
****

Рисунок 1 Интерфейс программы

Главный экран приложения состоит из трёх частей:

1. Рабочее пространство (основная площадь интерфейса). Поле, в котором пользователь размещает устройства, устанавливает между ними соединения и их расположение относительно друг друга. Перемещение внутри рабочего поля осуществляется банальным перетаскиванием выбранного устройства при помощи ЛКМ (левая кнопка мыши). Дополнительным функционалом является нажатие на любое устройство с помощью ПКМ (правая кнопка мыши), данное действие вызовет панель настроек выбранного устройства.
2. Панель инструментов (внизу). На панели инструментов располагаются устройства, доступные для добавления в рабочее пространство. Добавление нового элемента осуществляется системой двух кликов: первый – в панели инструментов для выбора устройства, которое планируется разместить в рабочей области; второй – в рабочей области для выбора места добавления выбранного устройства.
3. Панель дополнительных настроек (правая часть интерфейса). На данном этапе разработки в нее входят три элемента:
   1. Включение режима соединения и его выключение;
   2. Включение режима удаления и его выключение;
   3. Кнопка вызова командной строки.

## **5.2 Разбор кейсов**

В данном разделе на примере кейсов мы рассмотрим различные способы использования функционала приложения одновременно с демонстрацией других форм проекта.

Кейс 1.

В данном кейсе демонстрируется стандартное построение простой локальной сети, связывающей два конечных устройства в виде ПК «User 1» и «User 2».

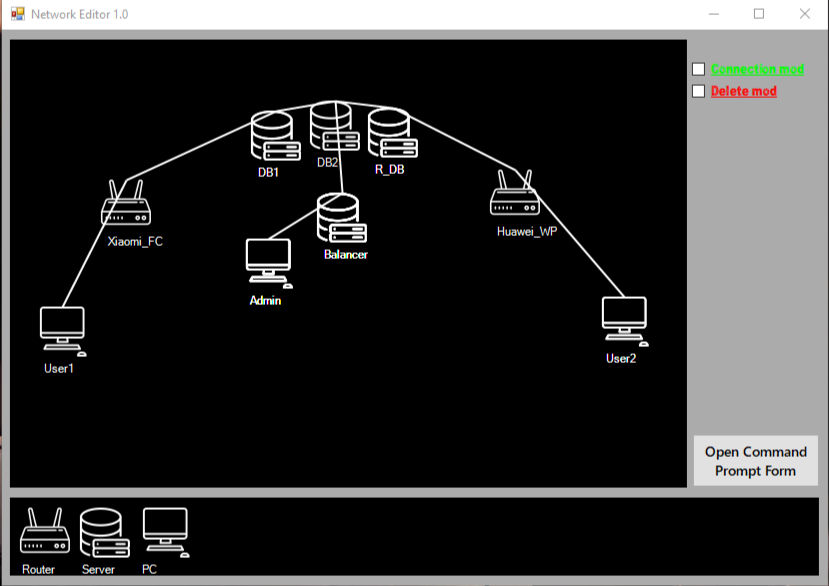


Рисунок 2 Пример сети, построенной в приложении

Кейс 2.

На данном кейсе демонстрируется командная строка, которая позволяет с помощью команды «ping» проверить наличие соединения между вышеупомянутыми конечными устройствами. В блоке **Input** вводится команда, общий вид которой представляет из себя следующее:

**ping name\_of\_device\_1 name\_of\_device\_2**

В блоке **Output** получаем ответ, подтверждающий наличие соединения между указанными устройствами.

Важное уточнение: стрелки в рабочем поле и рамка вокруг слова **Success** в блоке **Output** являются дополнительной графикой для демонстрации логики функциональности данного кейса. При реальном использовании приложения они отсутствуют. Данное уточнение применимо с к последующим кейсам.

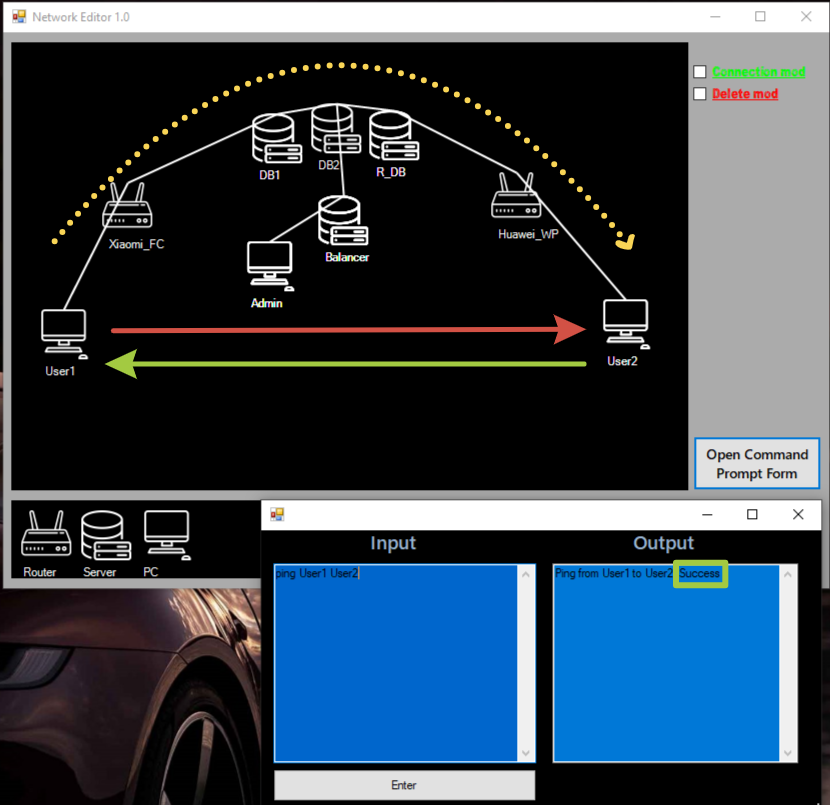


Рисунок 3 Демонстрация командной строки и использования команды ping

Кейс 3.

В данном случае на том же примере демонстрируется реакция команды **ping** на проверку соединения устройств, не находящихся в одной локальной сети.  
Важное уточнение: крестик, рамки вокруг устройства «Imposter», стрелка и подчеркивания красным маркером в блоке командной строки так же являются дополнительной графикой.

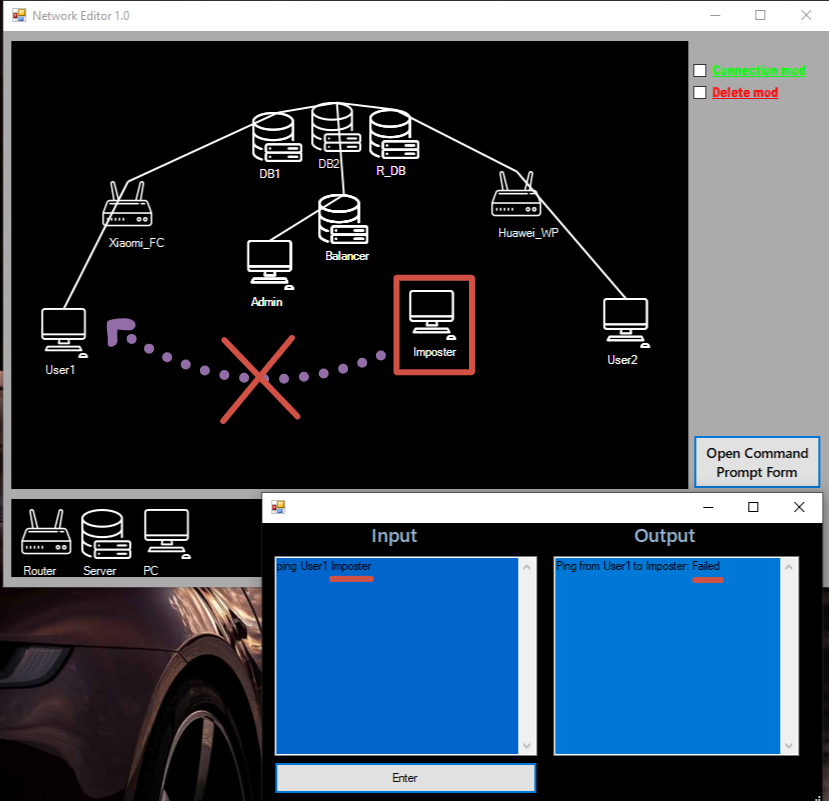


Рисунок 4 Демонстрация отрицательного результата ввода команды ping

Кейс 3.

Рассмотрим настройки трёх устройств:

1. Настройки устройства «User1»
2. Настройки устройства «Use2»
3. Настройки устройства «Imposter»

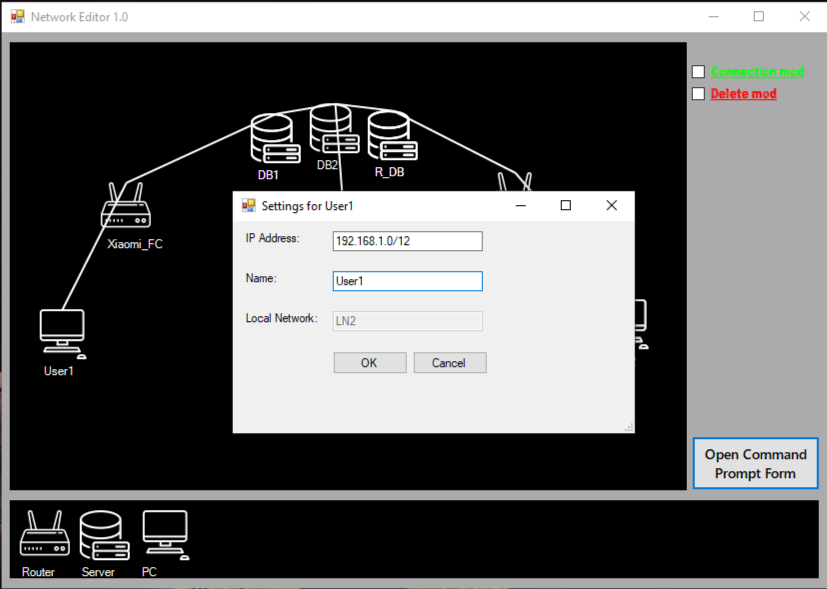


Рисунок 5 Настройки устройства "User1"

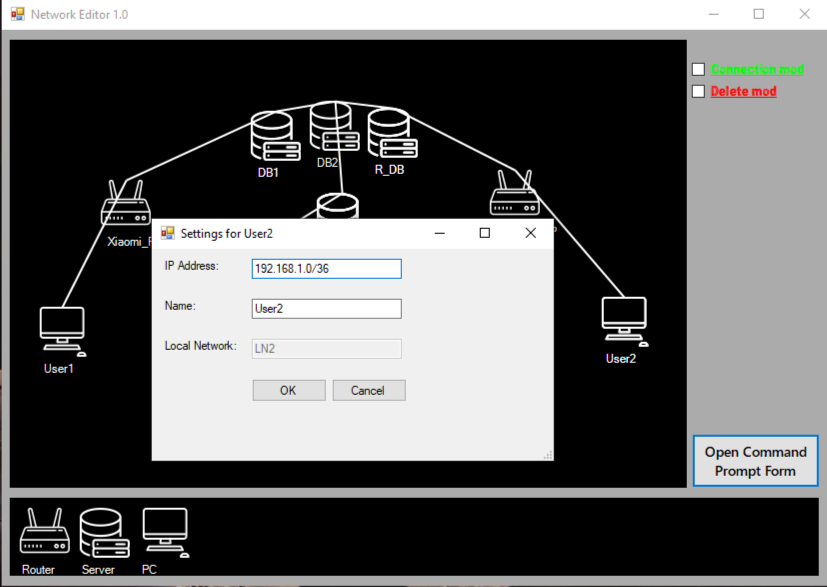


Рисунок 6 Настройки устройства "User2"

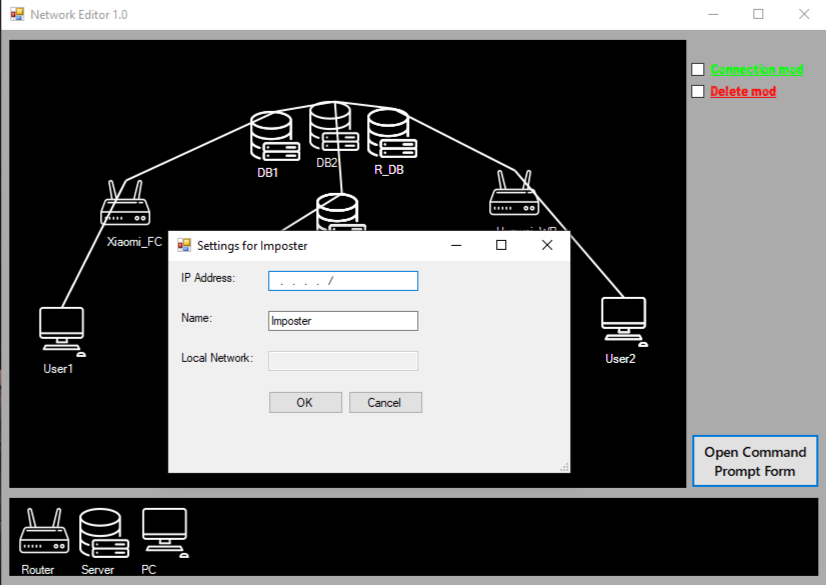


Рисунок 7 Настройки устройства "Imposter"

При детальном изучении представленных скриншотов можно сделать вывод, что идентификатор локальной сети у устройства «Imposter» отсутствует полностью, что логично, так как оно не подключено ни к одному устройству, ни внутри нашей основной локальной сети, ни к никакому другому. Данный кейс подтверждает правильность результата ввода команды ping в предыдущем кейсе.

# ВЫВОДЫ

На данный момент функционал приложения, разработанного в процессе написания данной работы, является плотной базой с потенциалом к расширению, комплексированию и оптимизации. Из первостепенных задач, которые следует реализовать, можно назвать следующие:

1. **Файловая система.** Добавление функционала сохранения текущих сборок для возможности при последующих сессиях продолжать работу с ними.
2. **Личный кабинет.** Позволит пользователям взаимодействовать со своими проектами не только локально на одном устройстве, но и позволит расширить функционал до включения облачного хранилища.
3. **Навигация.** Реализация главного меню для взаимодействия с проектами, хранящимися на локальном диске и/или в облаке.
4. **Углубление логики и добавление гибкости.** 
   1. Переход от абстрактных устройств, представленных в данном варианте, к более реалистичным девайсам, реально использующимся в сетевых коммуникациях в данный момент.
   2. Добавление возможности создания собственных устройств по параметрам пользователя.
   3. Переход от взаимодействия объектов посредством их названия на реальные протоколы коммуникации IPv4/IPv6 с учетом 802.11, TCP и UDP IP. Данный пункт так же ускорит переход от абстрактной сети к реальному построению виртуальной, что позволит в дальнейшем разработать функционал для отладки и тестирования моделей сети и обучения персонала в данной сфере.