



Московский государственный университет имени М.В.Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра автоматизации систем вычислительных комплексов

Гудов Дмитрий Олегович

**Разработка средства описания конфигурации сети
виртуальных машин для лабораторных работ**

Курсовая работа

Научный руководитель:

К.ф.-м.н. с.н.с.

Балашов Василий Викторович

Научный консультант:

Курячий Георгий Владимирович

Москва, 2026

Аннотация

В данной работе рассматривается задача разработки средства описания конфигурации сети виртуальных машин для лабораторных работ. Изучены существующие решения и выявлены их недостатки. Разработана и реализована модель данных для представления конфигурации сети. Разработаны и реализованы программные инструменты для редактирования, графического представления и вывода в структурированном текстовом формате конфигурации сети, входном для средства развертывания сети ВМ.

Содержание

Введение	4
1 Постановка задачи	5
2 Состав информации о конфигурации сети виртуальных машин	6
3 Обзор графических средств описания сетевых конфигураций	9
3.1 Универсальные диаграммеры	9
3.2 Сетевые эмуляторы	11
3.3 Генераторы диаграмм	12
3.4 Вывод	15
4 Выбор формата табличного представления конфигурации сети	16
5 Описание программной реализации	19
Заключение	23
Список литературы	24

Введение

В лабораторных практикумах по курсам, связанным с компьютерными сетями (например, "Введение в сети ЭВМ"[1]) значительная часть заданий связана с подготовкой сетевых конфигураций на стенде из виртуальных машин, размещённых на ресурсах одного физического компьютера. Для выполнения таких работ требуется не только создать набор виртуальных машин (ВМ), но и задать структуру их взаимодействия, сетевые интерфейсы, связи и параметры адресации.

Подготовка стенда включает клонирование базовых образов ВМ, подключение виртуальных сетевых адаптеров с контролем MAC-адресов, а также первичную настройку IP-адресов внутри гостевых операционных систем. Эти действия повторяются при выполнении лабораторных работ студентами и при демонстрации технологий на лекциях, что занимает время при ручном выполнении.

Существующие средства построения диаграмм позволяют наглядно изобразить структуру сети, однако не всегда удобны для задания формализованного описания с атрибутами элементов и возможностью дальнейшей автоматической обработки. В связи с этим актуальна разработка средства описания конфигурации сети виртуальных машин на основе структурированной модели данных. Такое описание должно быть редактируемым, читаемым и воспроизводимым, а также пригодным для последующего формирования диаграмм топологии или экспорта в формат конфигурации. Под воспроизводимостью будем понимать возможность повторно получить одну и ту же логическую структуру сети из одного и того же описания.

1 Постановка задачи

В рамках курсовой работы требуется разработать инструмент описания сетевых конфигураций виртуальных машин, обеспечивающий задание структуры сети и экспорт результата в формализованное описание.

Разрабатываемое решение должно обеспечивать:

- представление сетевой конфигурации в виде структурированных данных;
- возможность задания известных параметров элементов сети и сохранения незаполненных полей для последующего задания обучающимся при выполнении лабораторной работы;
- ввод и редактирование конфигурации;
- проверяемость и воспроизводимость полученного описания;
- экспорт выбранного представления в структурированный текстовый формат, входной для средства развертывания конфигурации сети ВМ.

Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать существующие подходы к описанию и визуализации сетевых конфигураций и сформулировать требования к разрабатываемому инструменту по поддержке редактирования конфигураций;
- разработать модель представления конфигураций в виде структурированных данных, позволяющую задавать известные параметры и оставлять поля для последующего заполнения;
- реализовать выбранный способ ввода и редактирования конфигурации, обеспечивающий проверяемость и воспроизводимость описания;
- разработать механизм экспорта выбранного представления в структурированный текстовый формат, входной для средства развертывания конфигурации сети ВМ.

2 Состав информации о конфигурации сети виртуальных машин

Под конфигурацией сети виртуальных машин в данной работе понимается формализованное описание набора виртуальных машин и связей между ними, необходимое для развертывания сетевой топологии лабораторного стенда. Такое описание должно фиксировать не только структуру сети ВМ, но и параметры элементов сети, необходимые для настройки ВМ.

В модель конфигурации сети ВМ входят следующие элементы и их атрибуты:

1. Топология сети

Топология представляет собой полное описание рассматриваемой конфигурации. Она объединяет все устройства и сети, входящие в состав лабораторного стенда.

В составе топологии фиксируются:

- список устройств;
- список сетей;

2. Устройство

Устройство соответствует отдельному элементу стенда (виртуальной машине). В зависимости от назначения оно может рассматриваться как хост, маршрутизатор или коммутатор. Такое деление нужно для указания роли устройства в топологии, но все эти элементы имеют одинаковую структуру описания.

Для устройства задаются:

- **имя устройства** — уникальное обозначение элемента в пределах топологии;
- **роль устройства** — назначение элемента в сети, например хост, маршрутизатор или коммутатор;
- **набор сетевых интерфейсов** — перечень интерфейсов, принадлежащих данному устройству.

Одно устройство может иметь один или несколько сетевых интерфейсов. Каждый интерфейс принадлежит ровно одному устройству.

3. Сетевой интерфейс

Сетевой интерфейс описывает точку подключения устройства к сети. Через интерфейсы задаются адресация, принадлежность к сетям, VLAN и отношения между физическими или логическими интерфейсами.

Для интерфейса задаются:

- **имя интерфейса** — обозначение интерфейса внутри устройства;
- **тип интерфейса** — характеристика интерфейса, например физический, виртуальный, VLAN-интерфейс или bridge-интерфейс;
- **адаптер** — сведения о сетевом адаптере, с которым связан интерфейс;
- **подчинённые интерфейсы** — список интерфейсов для bridge-интерфейса;
- **родительский интерфейс** — интерфейс, на основе которого создан vlan-интерфейс;
- **IP-адрес** — адрес интерфейса в соответствующей сети;
- **сеть** — имя сети или сегмента, к которому подключён интерфейс;
- **маска подсети** — параметр, определяющий размер IP-сети;
- **шлюз по умолчанию** — адрес маршрутизатора, используемый для выхода из сети;
- **VLAN** — идентификатор виртуальной локальной сети;
- **устройство-владелец** — устройство, которому принадлежит данный интерфейс.

Интерфейс связан с одним устройством и может быть включён в одну сеть. При этом сеть может содержать несколько интерфейсов разных устройств.

4. Сеть

Сеть описывает общий сетевой сегмент, к которому могут быть подключены интерфейсы устройств. Она используется для группировки интерфейсов и задания их "физической" связи.

Для сети задаются:

- **имя сети** — обозначение сетевого сегмента в пределах топологии;
- **список подключённых интерфейсов** — интерфейсы устройств, входящие в данную сеть.

Одна сеть может объединять один или несколько интерфейсов. Каждый интерфейс в текущей модели относится к одной сети или остаётся без указания сети, если этот параметр ещё не определён.

Так как на этапе описания конфигурации часть параметров может быть неизвестна, необходимо допускать частично заполненные данные. Это позволяет зафиксировать структуру сети и часть параметров, а затем уточнять отдельные атрибуты без изменения общей структуры.

3 Обзор графических средств описания сетевых конфигураций

Существующие графические средства, позволяющие описывать и/или отображать конфигурацию сети виртуальных машин, можно разделить на три группы: универсальные диаграммеры, сетевые эмуляторы и генераторы диаграмм по текстовому описанию. Их применимость целесообразно оценивать по следующим критериям: возможность изобразить структуру сети, возможность сопоставить элементам сети набор параметров, возможность интерпретировать результат как формализованное описание, пригодное для последующей обработки.

3.1 Универсальные диаграммеры

К универсальным диаграммерам относятся такие средства как draw.io [2] (рис. 1), Lucidchart [3] (рис. 2) и Microsoft Visio [4] (рис. 3). Такие средства позволяют строить различные виды схем, в том числе сетевые диаграммы. Например, draw.io позиционируется как онлайн-инструмент для построения блок-схем, UML-, ER- и сетевых диаграмм; Microsoft Visio также содержит шаблоны и фигуры для сетевых диаграмм; Lucidchart поддерживает технические диаграммы и визуальное моделирование систем.

Основное преимущество таких решений состоит в удобстве визуального построения схемы сети. Пользователь может быстро расположить элементы, провести связи между ними и получить наглядное изображение сетевой топологии. Однако для поставленной задачи этого недостаточно. Диаграмма в таких системах в первую очередь является графическим объектом: она хорошо передаёт внешний вид структуры, но не задаёт строгую модель данных сетевой конфигурации. Параметры виртуальных машин, интерфейсов, MAC-адресов, IP-адресов и сетевых сегментов могут быть добавлены только как текстовые подписи или дополнительные свойства объектов, но их последующая автоматическая интерпретация становится нетривиальной.

Следовательно, универсальные диаграммеры могут использоваться для иллюстрации результата, но не являются подходящей основой для хранения и редактирования конфигурации. Они не обеспечивают требуемой проверяемости описания и плохо подходят для ситуации, когда одному элементу сети необходимо сопоставить множество формализованных атрибутов.

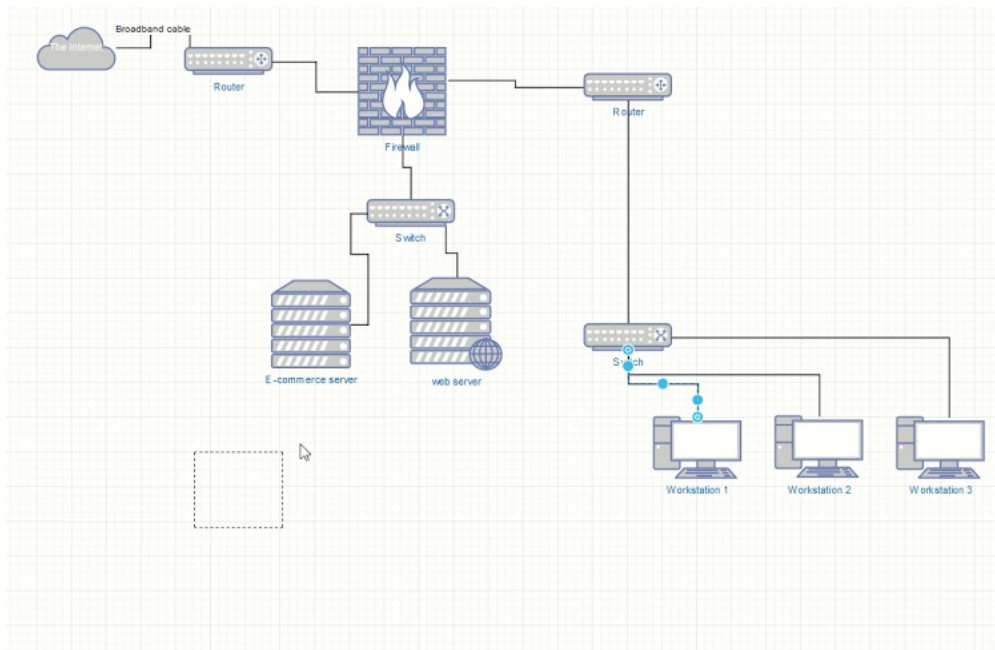


Рис. 1: Диаграмма сети в draw.io

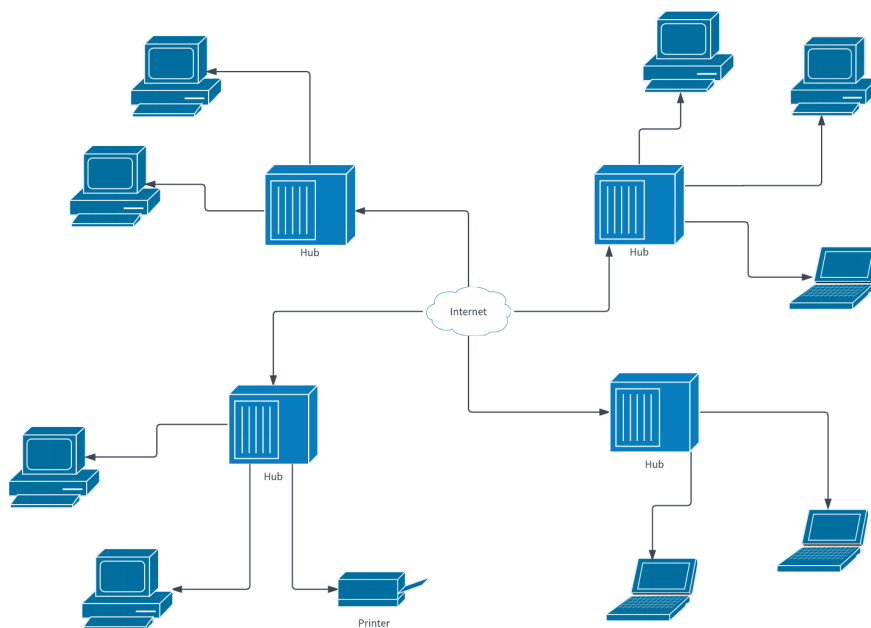


Рис. 2: Диаграмма сети в Lucidchart

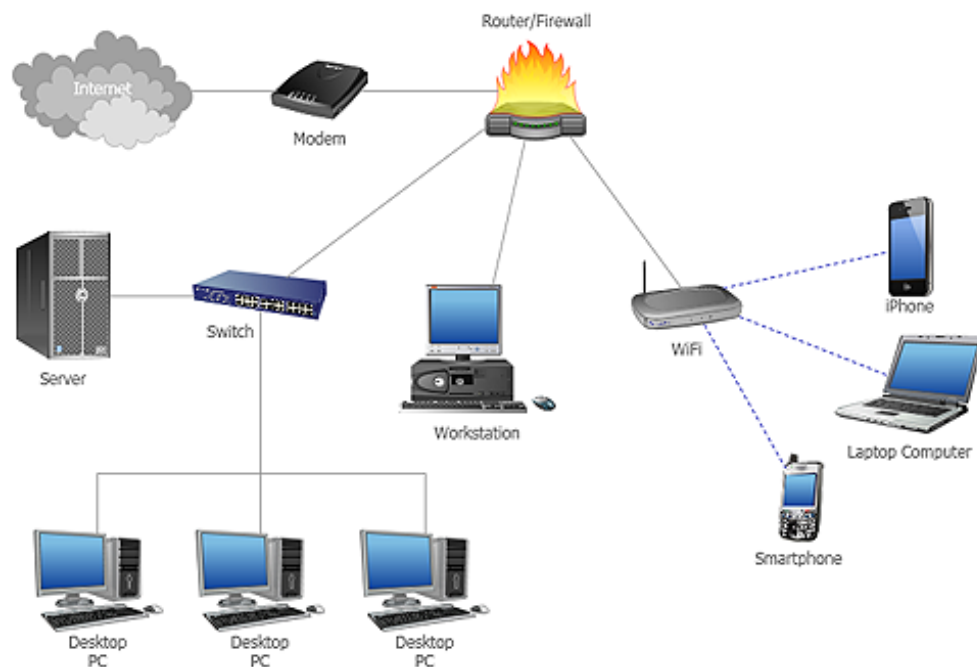


Рис. 3: Диаграмма сети в Visio

3.2 Сетевые эмуляторы

Ко второй группе относятся сетевые эмуляторы, например Cisco Packet Tracer [5] (рис. 4) и Huawei eNSP [6] (рис. 5). Cisco Packet Tracer предназначен для обучения сетевым технологиям и позволяет моделировать работу сетей в виртуальной лабораторной среде; Huawei eNSP также позиционируется как средство моделирования сетей, применяемое для подготовки специалистов в области сетевой передачи данных.

Эти решения ближе к реальной настройке сети, чем универсальные диаграммы: они позволяют нарисовать топологию сетевых устройств, однако их назначение далеко от цели данной работы. Диаграммы в их составе ориентированы на описание сети на основе оборудования конкретного производителя, а не на универсальное формализованное описание конфигурации сети виртуальных машин для лабораторных стендов.

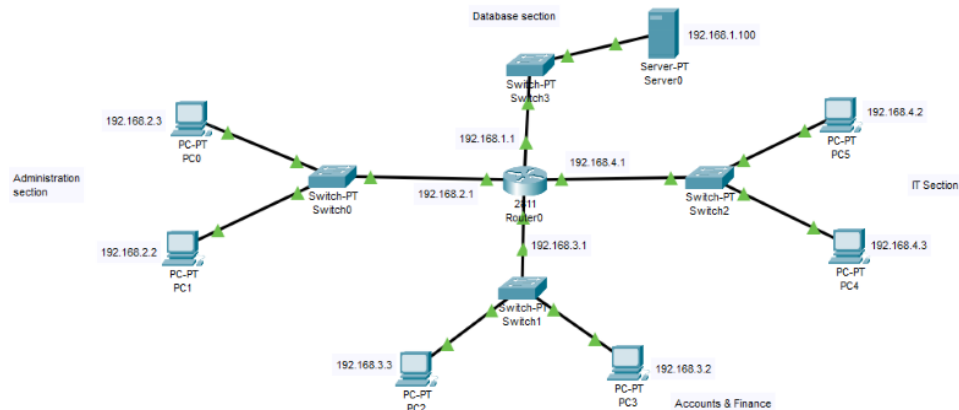


Рис. 4: Диаграмма сети в Cisco Packet Tracer

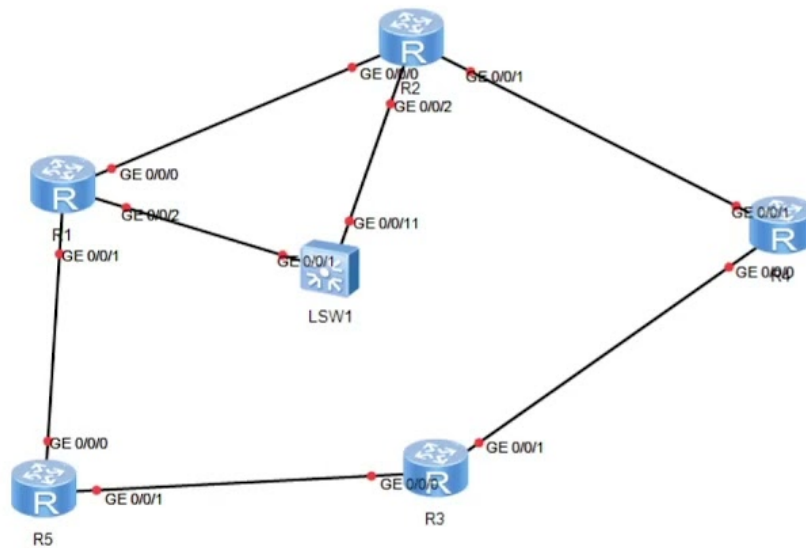


Рис. 5: Диаграмма сети в Huawei eNSP

3.3 Генераторы диаграмм

Третью группу составляют генераторы диаграмм по текстовому описанию: D2 [7] (рис. 6), Graphviz [8] (рис. 7), PlantUML [9] (рис. 8). Все три инструмента используют декларативный язык, позволяющий преобразовывать текстовое описание в диаграмму и выполняют смежные задачи.

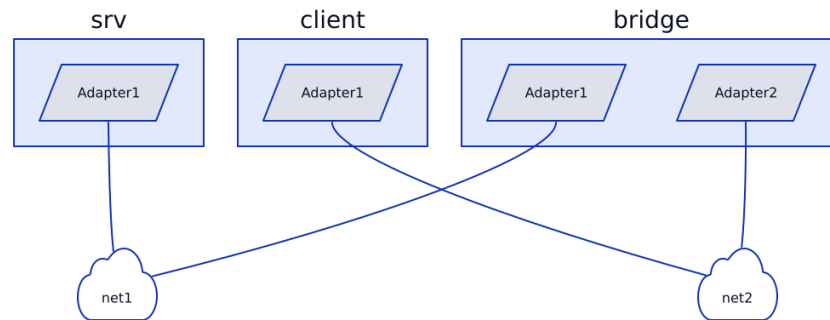


Рис. 6: Диаграмма сети в D2

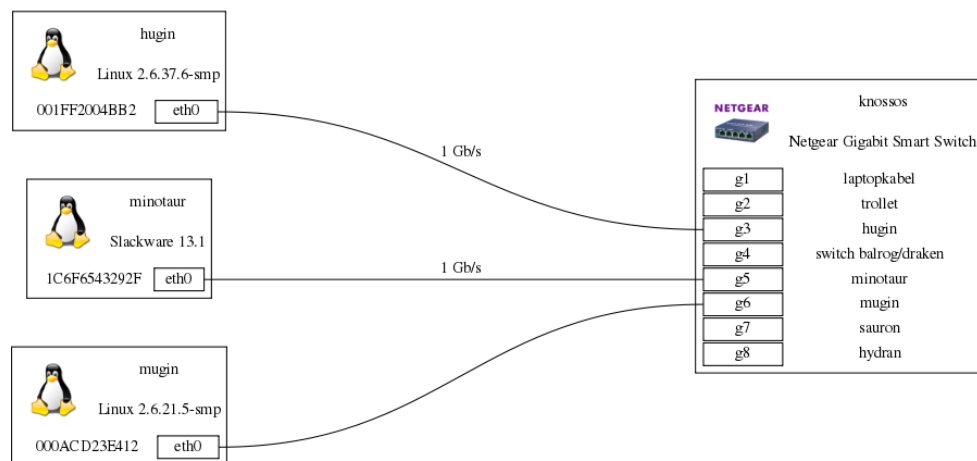


Рис. 7: Диаграмма сети в Graphviz

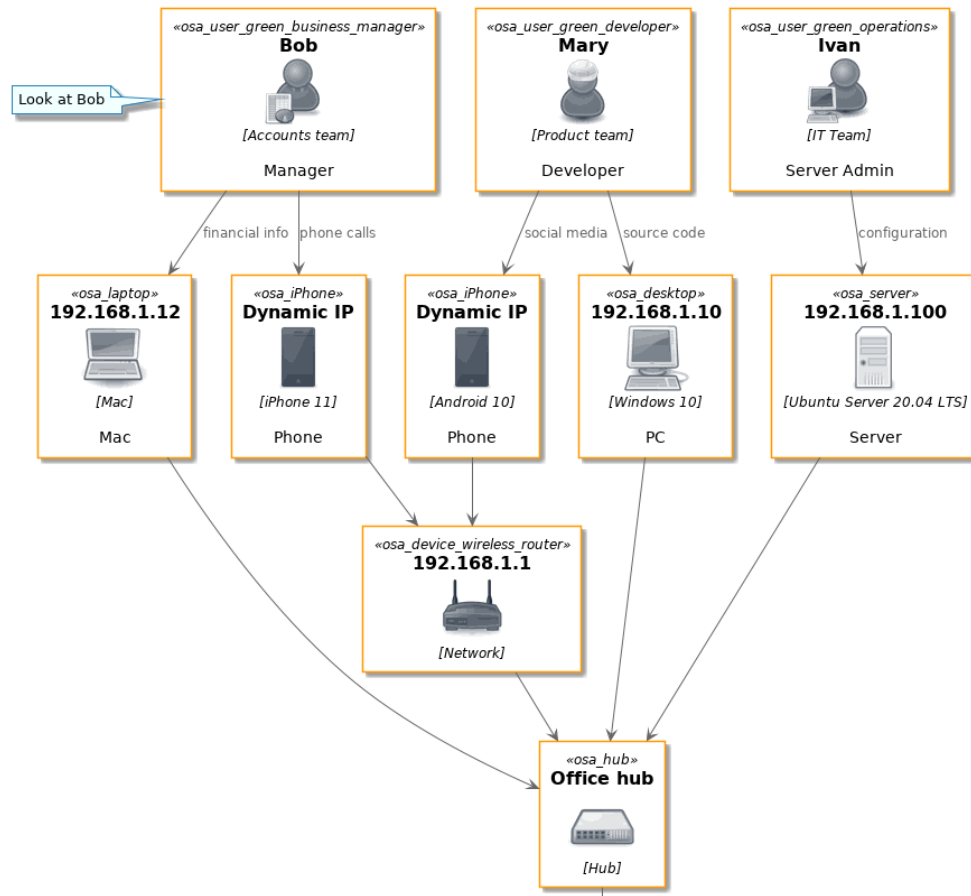


Рис. 8: Диаграмма сети в PlantUML

Такие инструменты упрощают интеграцию с внешними средствами описания конфигурации, поскольку исходное описание диаграммы хранится в текстовом виде. По этой причине генераторы диаграмм могут быть полезны как вспомогательный механизм визуализации сетевой топологии. В частности, в рамках данной работы D2 используется для построения графа L1 топологии сети на основе внутренней модели данных.

Однако эти средства не решают задачу полностью. Они позволяют генерировать диаграмму, но не редактировать ее структуру или атрибуты ее элементов. Узлы и связи во входных языках таких средств можно снабжать дополнительными атрибутами, наращивая необходимые свойства, но такое представление, по мере увеличения сложности топологии, быстро становится трудно читаемым. Генератор диаграмм может отобразить уже подготовленное описание сетевой диаграммы, но не предоставляет язык описания конфигурации сети и способ её редактирования.

3.4 Вывод

Рассмотренные решения покрывают отдельные аспекты поставленной задачи, но не позволяют ее решить полностью. Универсальные диаграммеры удобны для ручного построения схемы сети, однако результат их работы в первую очередь является графическим представлением. Такие средства плохо подходят для хранения набора формализованных параметров устройств, интерфейсов и сетевых сегментов, а также для последующей программной обработки этого описания. Сетевые эмуляторы ориентированы на оборудование конкретного производителя, что ограничивает их применимость для описания лабораторных стендов общего вида. Генераторы диаграмм позволяют получать визуальное представление на основе текстовых данных. Тем не менее их основная функция заключается в построении диаграммы, а не в задании и редактировании конфигурации сети.

Таким образом, в качестве основы разрабатываемого решения целесообразно использовать не формат диаграммы, а структурированную модель данных сетевой конфигурации. Диаграмма в этом случае рассматривается как вспомогательное представление. D2 был выбран в качестве генератора диаграмм, поскольку для него существует лёгкая в использовании библиотека в языке Python [10], позволяющая генерировать текстовое описание диаграммы. Другие генераторы диаграмм также могли бы использоваться.

4 Выбор формата табличного представления конфигурации сети

Исходя из результатов обзора, в качестве пользовательской формы задания конфигурации выбрано табличное представление. Такая форма представления удобна тем, что позволяет явно фиксировать элементы сети и их атрибуты, редактировать отдельные значения и проверять заполненность полей. При этом она допускает частично заданные конфигурации: часть параметров может быть указана сразу, а часть оставлена пустой для последующего уточнения.

При разработке табличного представления конфигурации сети виртуальных машин рассматривались два варианта организации данных: единая таблица (представление аналогично первой нормальной форме в контексте СУБД [11]) и отдельные таблицы (аналогично третьей нормальной форме в контексте СУБД). Оба варианта позволяют структурированно описать сеть, однако различаются сложностью редактирования данных в случае непосредственной работы с таблицами.

Разберём оба представления на примере топологии, изображённой на рисунке 9

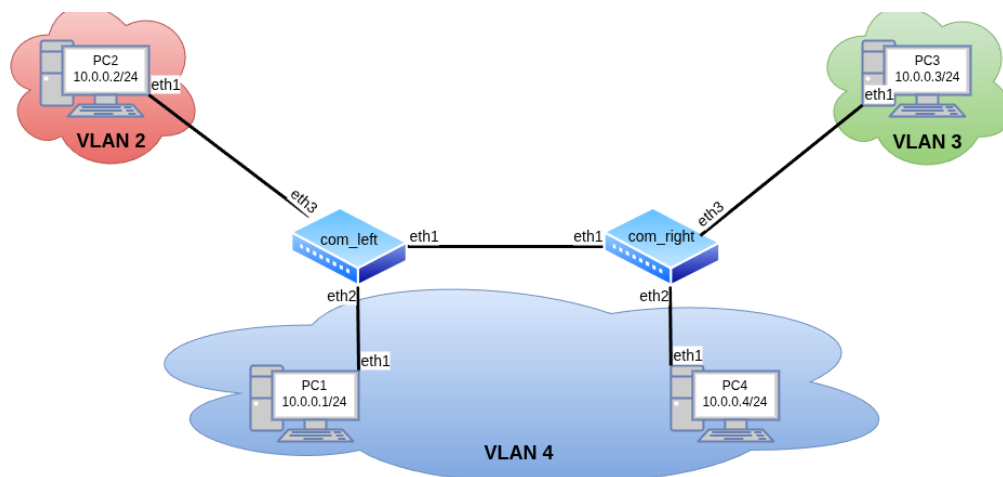


Рис. 9: Пример топологии

Представление в единой таблице

При представлении данных в виде одной таблицы каждая строка содержит сведения об отдельном элементе или связи конфигурации: имя узла, интерфейс, адреса, принад-

лежность сети и т.д. Такое представление остаётся достаточно простым для ручного заполнения и позволяет сразу видеть основную структуру конфигурации.

Пример представления конфигурации сети в 1НФ приведен в таблице 1.

Name	Role	Interface	Network	Device IP
PC1	Host	eth1	A	10.0.0.1/24
PC2	Host	eth1	B	10.0.0.2/24
PC3	Host	eth1	C	10.0.0.3/24
PC4	Host	eth1	D	10.0.0.4/24
com_left	Switch	eth1	tr	
com_left	Switch	eth2	A	
com_left	Switch	eth3	B	
com_right	Switch	eth1	tr	
com_right	Switch	eth2	D	
com_right	Switch	eth3	C	

Таблица 1: Единая таблица конфигурации сетевых устройств

Представление в отдельных таблицах

При представлении данных в виде отдельных таблиц сведения о конфигурации распределяются по основным видам сущностей: узлы (устройства), интерфейсы, сетевые сегменты и связи. Каждой из этих сущностей сопоставляется отдельная таблица. Такой подход уменьшает дублирование данных по сравнению с единой таблицей, однако делает представление более громоздким. Для понимания одной связи приходится обращаться сразу к нескольким таблицам.

Пример представления конфигурации сети в 3НФ приведен в таблицах 2 - 4.

Name	Device
PC1	Host
PC2	Host
PC3	Host
PC4	Host
com_left	Switch
com_right	Switch

Таблица 2: Сетевые устройства

Host Name	Interface	Network
PC1	eth1	A
PC2	eth1	B
PC3	eth1	C
PC4	eth1	D
com_left	eth1	tr
com_left	eth2	A
com_left	eth3	B
com_right	eth1	tr
com_right	eth2	D
com_right	eth3	C

Таблица 3: Интерфейсы

Interface	IP
PC1.eth1	10.0.0.1/24
PC2.eth1	10.0.0.2/24
PC3.eth1	10.0.0.3/24
PC4.eth1	10.0.0.4/24

Таблица 4: IP-адреса интерфейсов

Выбор представления

В качестве основного пользовательского представления была выбрана единая таблица. Она удобнее для ручного заполнения, проще воспринимается при описании лабораторных стендов и не требует постоянного перехода между несколькими связанными таблицами.

Единая таблица используется как форма ввода, а при дальнейшей обработке данные могут преобразовываться во внутреннюю модель, где отдельно выделяются узлы, интерфейсы и связи. Такой подход сохраняет простоту заполнения таблицы и одновременно позволяет использовать данные для экспорта во входные форматы средства развертывания сетевых конфигураций или средства построения диаграмм топологии.

5 Описание программной реализации

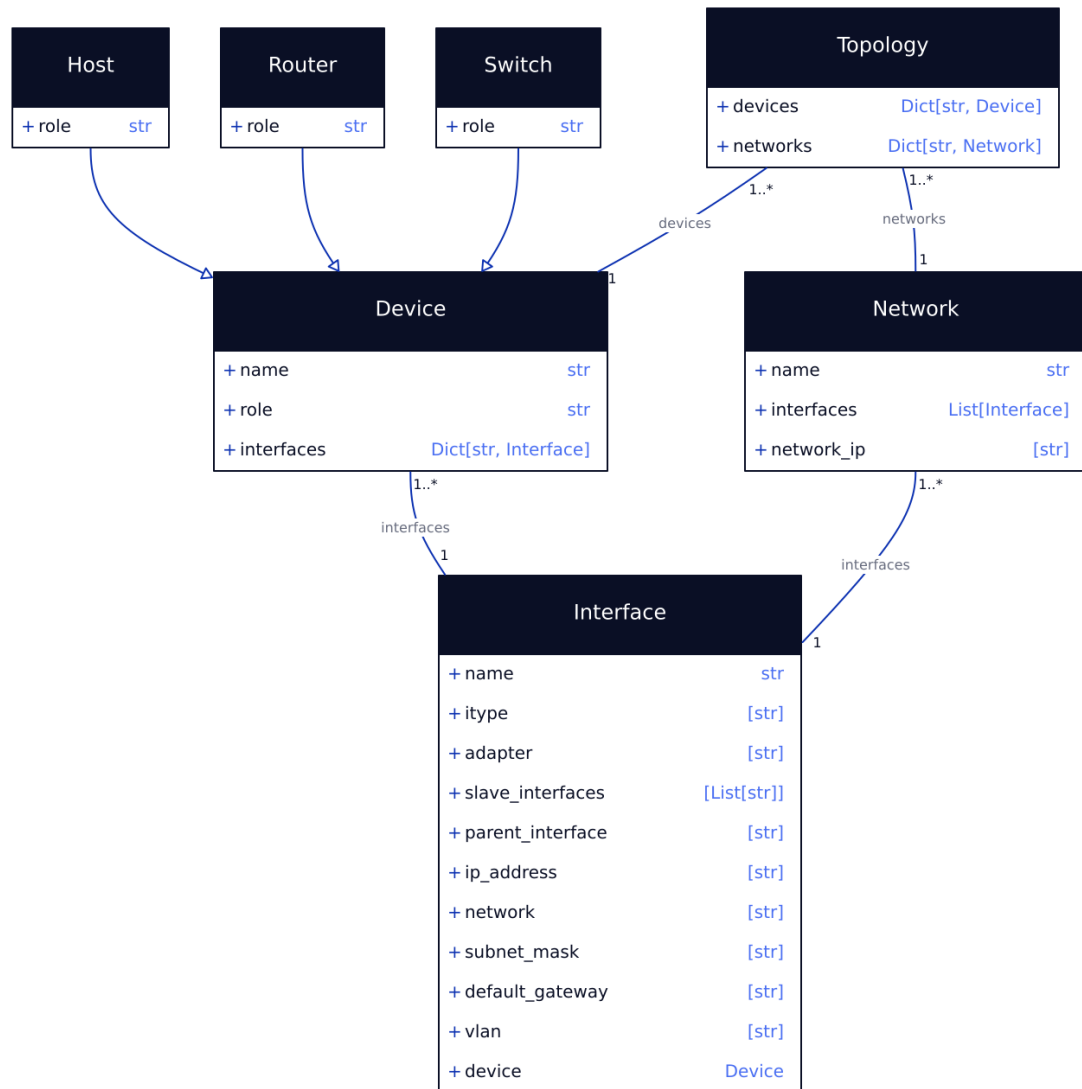


Рис. 10: UML-диаграмма классов Python

Табличное представление используется для ввода, но для дальнейшей обработки требуется программная структура, в которой связи между объектами выражены явно. Поэтому данные таблиц должны преобразовываться во внутреннее представление, которое

было реализовано в виде набора объектов Python (рис. 10), соответствующих основным сущностям конфигурации. Такая модель служит центральным представлением, из которого данные могут быть преобразованы в другие форматы.

Также необходима разработка механизмов преобразования модели. Основными преобразованиями являются импорт данных из табличного описания во внутреннюю модель, экспорт во входные языки средства развертывания сетевых конфигураций [12] и средства построения диаграмм (D2).

Программная реализация выполнена на языке Python. Объём написанного кода: 710 строк, 48 килобайт. Код выложен на веб-сервисе Github [13]. На рисунке 11 представлен основной процесс работы с утилитой. Построение диаграмм происходит с помощью модуля ru-d2 (пример на рис 12).

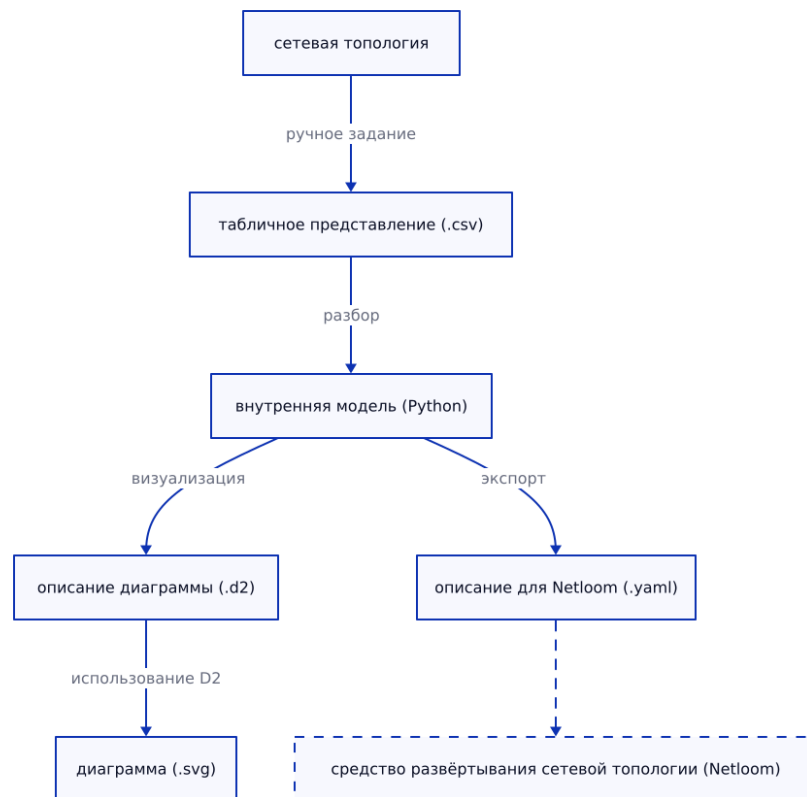


Рис. 11: Схема работы программной реализации

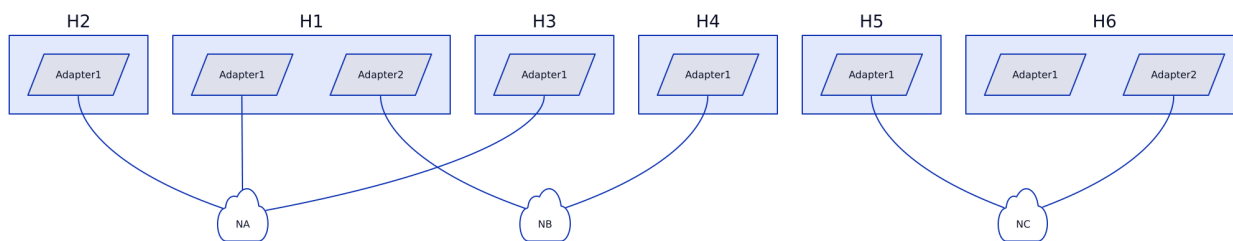


Рис. 12: Пример диаграммы, построенной утилитой

Воспроизводимость описания конфигурации обеспечивается тем, что результат хранится в виде структурированных данных. Одна и та же таблица при повторной обработке приводит к построению той же внутренней модели и, соответственно, к одинаковому описанию конфигурации. А преобразование в формат, пригодный для использования в средстве развёртывания сети виртуальных машин, обеспечивает возможность дальнейшего применения описания при подготовке лабораторных стендов из виртуальных машин.

Пример экспортированного YAML-описания конфигурации сети приведён в листинге 1.

Листинг 1: Фрагмент YAML-описания конфигурации сети виртуальных машин

```

1 meta:
2   id: topology.yaml
3   name: topology.yaml
4 networks:
5 - name: A
6 nodes:
7 - role: host
8   name: PC1
9   interfaces:
10    eth1:
11      ip: 10.0.0.1/24
12      network: A
13      gateway: null
14      index: 1
15   bridges: []
16   vlans: []
17 - role: host
18   name: PC2
19   interfaces:
20    eth1:
21      ip: 10.0.0.2/24
22      network: A

```

```
23     gateway: null
24     index: 1
25 bridges: []
26 vlans: []
```

Заключение

В работе получены следующие результаты:

- выполнен анализ существующих подходов к описанию и визуализации сетевых конфигураций; на его основе выбран способ редактирования конфигурации с использованием табличного представления во внешнем редакторе;
- разработана модель для представления конфигурации сети виртуальных машин в виде структурированных данных, позволяющая описывать устройства, интерфейсы, сети и связи между ними, а также оставлять отдельные параметры незаполненными;
- выполнено сопряжение табличного представления с разработанной моделью данных, что обеспечивает ввод, редактирование и последующую обработку описания конфигурации;
- разработана программа преобразования табличного описания во входной язык средства визуализации конфигурации сети виртуальных машин и средства развертывания конфигурации.

В качестве направлений развития работы можно предложить разработку отдельного редактора конфигурации сети, использующего созданную модель данных. Также перспективным является встраивание в редактор функций проверки корректности конфигурации, а также поддержка диаграмм для отображения конфигурации сети на разных уровнях стека протоколов.

Список литературы

- [1] БМК Кафедра АСВК. Введение в сети ЭВМ. — Available at <https://asvk.cs.msu.ru/uchebnyj-process/chitaemye-kursy/vvedenie-v-seti-evm/> (accessed May 19, 2026).
- [2] draw.io. Configurable diagramming application, 2026. Available at <https://github.com/jgraph/drawio/wiki> (accessed May 19, 2026).
- [3] Lucidchart. Visual workspace for diagramming, 2026. Available at <https://www.lucidchart.com/pages> (accessed May 19, 2026).
- [4] Microsoft Visio. Visual workspace for diagramming, 2026. Available at <https://support.microsoft.com/en-US/Visio/beginner-tutorial-for-visio> (accessed May 19, 2026).
- [5] Cisco Packet Tracer. Network simulation tool, 2026. Available at <https://www.netacad.com/cisco-packet-tracer> (accessed May 19, 2026).
- [6] Huawei eNSP. Enterprise network simulation platform, 2026. Available at <https://support.huawei.com/enterprise/en/nce-data-communication/ensp-pid-9017384> (accessed May 19, 2026).
- [7] D2. Diagram scripting language, 2026. Available at <https://d2lang.com/tour/intro/> (accessed May 19, 2026).
- [8] Graphviz. Open source graph visualization software, 2026. Available at <https://graphviz.org/documentation/> (accessed May 19, 2026).
- [9] PlantUML. Tool for creation of a wide array of diagrams, 2026. Available at <https://plantuml.com/guide> (accessed May 19, 2026).
- [10] py-d2. Fully typed python interface for building .d2 diagram files in python. <https://github.com/MrBlenny/py-d2> (accessed May 23, 2026)
- [11] **Дейт К. Дж.** Введение в системы баз данных (8-е изд.) // Издательский дом "Вильямс". — 2005. — 1327 с.

- [12] Netloom. A CLI tool for creating and managing network lab topologies. <https://github.com/wlix13/NetLoom/> (accessed May 23, 2026)
- [13] Программная реализация инструмента. <https://github.com/kr0sh512/network-diagrams-tool> (accessed May 19, 2026)