МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский Авиационный Институт»

(Национальный Исследовательский Университет)

Институт№8: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

**КУРСОВОЙ ПРОЕКТ**

По курсу «Вычислительные системы»

I семестр

Тема:

«Схема лабораторной вычислительной системы»

|  |  |
| --- | --- |
| Группа: | М8О-106Б-22 |
| Студент: | Каримов А.А. |
| Преподаватель: | Дубинин А.В. |
| Оценка: |  |
| Дата: |  |

Москва, 2022

Оглавление

[Постановка задачи 2](#_Toc123937497)

[Схема локальной сети 3](#_Toc123937498)

[Теоретическая часть 4](#_Toc123937499)

[Виды топологий 4](#_Toc123937500)

[Виды сетевых (промежуточных) устройств 7](#_Toc123937501)

[Сетевая модель OSI 8](#_Toc123937502)

[Сетевая модель TCP/IP 11](#_Toc123937503)

[Domain Name System (DNS) 11](#_Toc123937504)

[Dynamic Host Configuration Protocol (DHCP) 14](#_Toc123937505)

[Network Address Translation (NAT) 15](#_Toc123937506)

[Заключение 16](#_Toc123937507)

[Источники 17](#_Toc123937508)

# Постановка задачи

Этот курсовой проект направлен на изучение сетевого устройства кабинета, где проводятся практические занятия.

Цель: изучить устройство компьютерного кабинета вычислительной системы.

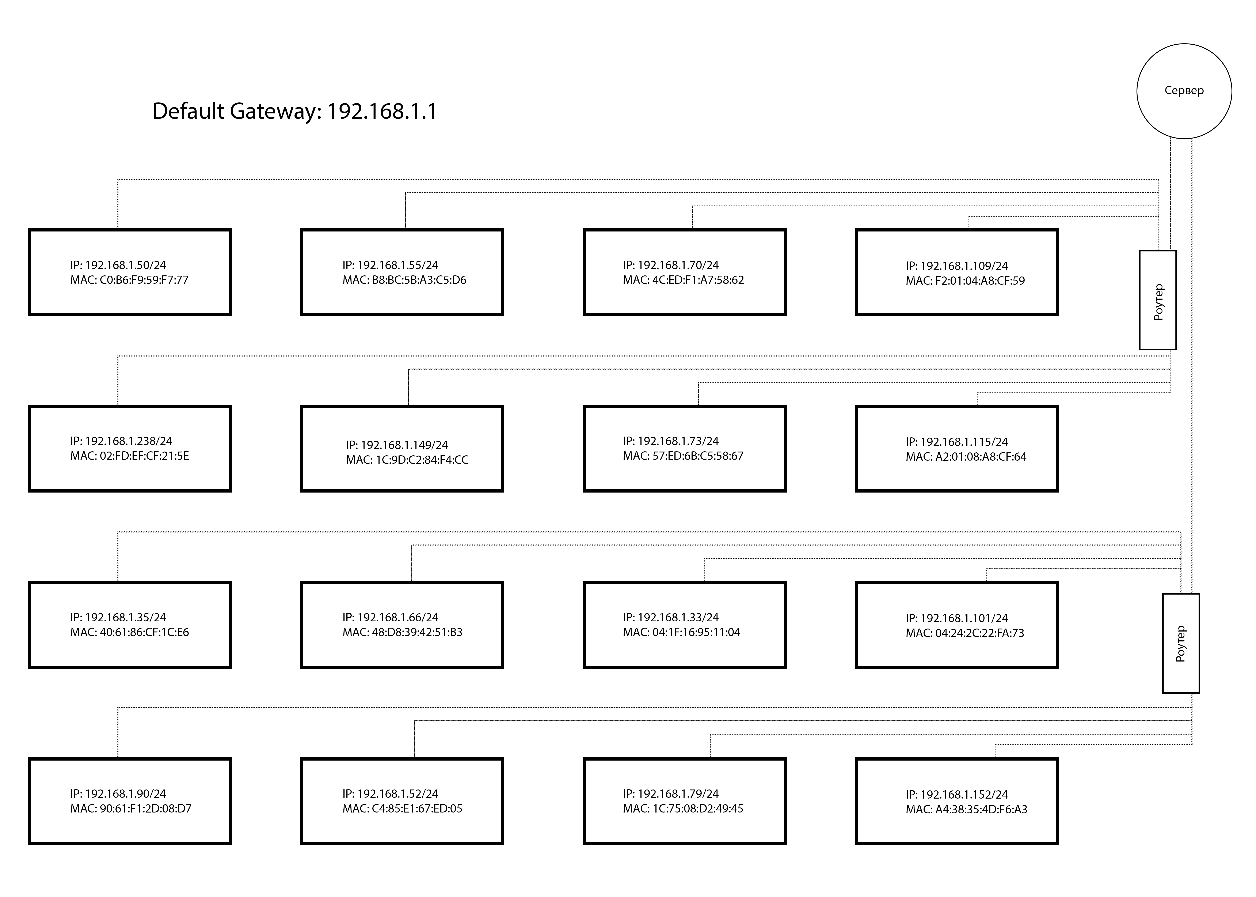
Задачи:

1. Дать определение понятию «компьютерные сети», рассказать, какими они бывают;
2. Изучить схему кабинета;
3. Изучить оборудование, которое используется в кабинете
4. Проанализировать работу компонентов сети;
5. Рассказать, как компоненты взаимодействуют между собой.

# Схема локальной сети

На рисунке 1 представлена схема кабинета, в котором у моей группы проходят практические занятия.

Рисунок 1 - Схема локальной сети



По схеме видно, что кабинет содержит 16 компьютеров, 4 коммутатора и 1 сервер.

Пояснение к рисунку 1:

Шлюз по умолчанию (англ. Default gateway) — в маршрутизируемых протоколах — сетевой шлюз, на который пакет отправляется в том случае, если маршрут к сети назначения пакета не известен (не задан явным образом в таблице маршрутизации хоста)

# Теоретическая часть

Компьютерная сеть – это вычислительные устройства, связанные каналами передачи данных, которые могут обмениваться информацией и совместно использовать ресурсы. Пример таких устройств могут выступать сервера, компьютеры, телефоны, маршрутизаторы и т.д.

Компоненты сети можно классифицировать:

1. Оконечные узлы

Передающие/отправляющие устройства данные в сети, например, телефоны, ноутбуки.

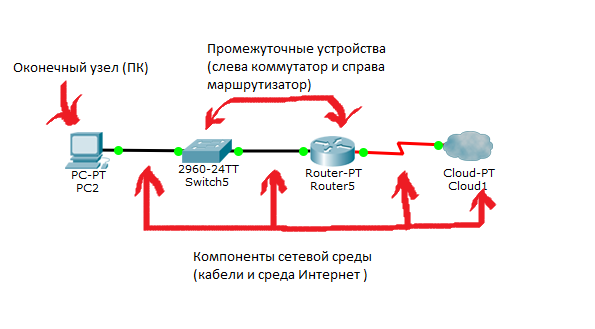
1. Промежуточные устройства

Устройства, служащие для связи оконечных узлов, например, роутеры.

1. Сетевые среды

Среды, которые являются физическими проводниками передачи данных. Сюда относятся кабели, коннекторы, воздушная среда. В медном кабеле – это передача электрических сигналов, в оптоволокне – световых лучей.

Рисунок 2 - Схема расположения компонентов

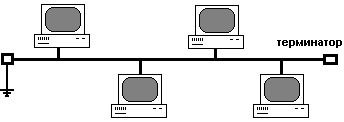


# Виды топологий

1. Топология с общей шиной

К одному кабелю присоединяются все устройства, образуя локальную сеть. На концах присутствуют кабеля-терминаторы. Из достоинств этой топологии можно выделить только простоту установки. Тем не менее, любой разрыв кабеля влечет за собой крах всей сети.

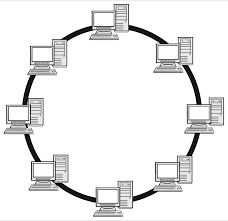
Рисунок 3 - Топология с общей шиной



1. Кольцевая топология

В этой топологии каждое устройство подключается к двум соседним, образуя кольцо. Так, устройство с одного конца только отправляет, а устройство с другого – принимает, а все остальные устройства в сети являются ретрансляторами сигнала. Здесь больше не нужны терминаторы. В этой схеме остается проблема предыдущей: разрыв кабеля рушит всю сеть, поэтому для большей надежности делали двойной кольцо, чтобы при отказе основного кабеля начинал работать резервный.

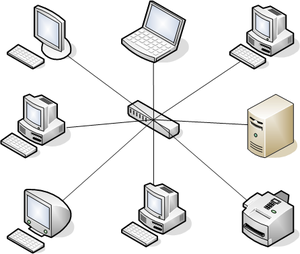
Рисунок 4 - Кольцевая топология



1. Топология звезда

Здесь все устройства подключаются к одному центральному узлу, который является ретранслятором. Ныне эта топология иногда используется в локальных сетях. Надежность этой сети выше, так как при разрыве одного из кабелей, сеть продолжит работать, но, если откажет центральный узел, сеть откажет.

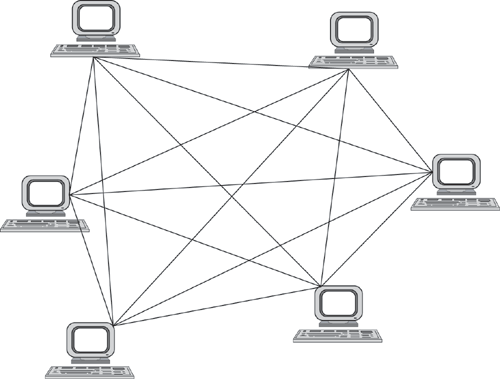
Рисунок 5 - Топология звезда



1. Полносвязная топология

В этой схеме все устройства связаны напрямую друг с другом. Благодаря связи каждого с каждым, надежность этой топологии очень высока. Но такое количество кабелей делает эту сеть сложной и дорогой: для сети из 100 компьютеров потребуется 4950 кабелей.

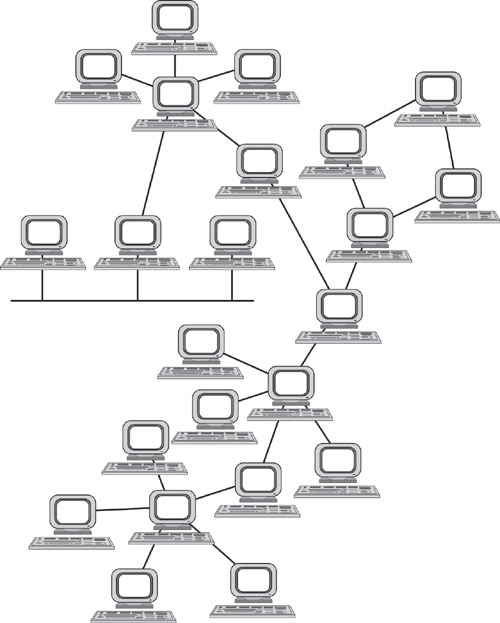
Рисунок 6 - Полносвязная топология



1. Смешанная топология

Наиболее распространенная сейчас. Она объединяет в себе все вышеперечисленные топологии. Такая схема представляет собой древовидную структуру. Она является надежной, так как при обрыве кабеля между соседними узлами связь потеряется только между ними, а все остальные объединенные устройства продолжат работу.

Рисунок 7 - Смешанная топология



# Виды сетевых (промежуточных) устройств

1. Hub (данные по MAC адресу)

Транслирует пакеты, поступающие на один из портов на все остальные. Работает на физическом уровне OSI. Из-за копирования пакетов во все порты работает медленно.

1. Switch (данные по MAC адресу)

Передает данные строго от одного порта к другому на основании информации из пакета. Работает на канальном уровне OSI.

1. Router (данные передает по IP адресу)

Служит для передачи пакетов между устройствами на основе каких-либо правил. Работает на сетевом уровне OSI. Роутер выбирает оптимальный маршрут передачи между нужными устройствами. Имеет таблицы маршрутизации, которые содержат информацию обо всех возможных маршрутах передачи пакетов с дополнительной информацией о состоянии канала, загруженности и т.д.

# Сетевая модель OSI

Для решения проблемы отсутствия единых стандартов международная организация стандартизации (ISO) разработала сетевую модель OSI к 1984 году. Проблемой уже на старте стала долгая разработка – сетевую модель разрабатывали около 7 лет, пока остальной мир занимался разработкой и совершенствованием других сетевых моделей.

Сама модель состоит из 7 уровней. Каждый уровень отвечает за определенные задачи.

1. Физический уровень

Физический уровень отвечает за методы передачи данных, за используемую среду (оптоволокно, витая пара, коаксиальный кабель), уровень напряжения, методы кодирования двоичных сигналов

1. Канальный уровень

Этот уровень отвечает за адресацию в пределах локальной сети, обнаруживает ошибки, проверяет целостность данных. Здесь работают MAC-адреса и протокол Ethernet. Также полученные с физического уровня данные пакуются в фреймы, проверяется их целостность и исправляются ошибки путем повторного запроса на нецелый фрейм. Канальный уровень подразделяется на 2 подуровня:

1. Media Access Control (MAC)

Сам MAC-адрес позволяет уникально идентифицировать каждый узел сети и доставлять данные только этому узлу.

1. LLC (Logical Link Control)

Он осуществляет управление передачей данных и обеспечивает проверку и правильность передачи информации по соединению.

1. Сетевой уровень

Основная задача этого уровня – доставка пакетов от одного узла-отправителя к узлу-получателю независимо от принадлежности узлов к конкретной локальной сети. Если на канальном уровне адресация осуществляется за счет MAC-адресов, то на сетевом появляются IP адреса. IP адреса устройств одной сети имеют общую часть, которая называется адресом или номером этой IP-сети и специфическую для каждого устройства часть, называемую адресом, или номером, данного устройства в данной IP-сети.

Так как IP адрес содержит в себе как адрес узла, так и адрес сети, необходимо иметь алгоритм получения из IP адреса устройства адрес сети, к которой принадлежит устройство и сам номер устройства в этой сети. Для этого существует маска сети. Она используется для отделения битов для номера сети и битов, которые являются частью идентификатора хоста.

1. Транспортный уровень

Этот уровень обеспечивает функции доставки пакетов. Здесь работают протоколы TCP и UDP. Первый устанавливает соединения между хостами и гарантирует целостность переданных данных (если какая-то часть утеряна, то протокол запрашивает её повторно).

User Datagram Protocol (UDP) не устанавливает соединения, а передает автономные датаграммы. В процессе работы потерянные данные игнорируются, то есть контроля целостности не производится, благодаря чему достигается высокая скорость работы в ущерб надежности.

Хосты при установке соединения по этим протоколам идентифицируются согласно номерам портов. Количество портов ограничено с учётом 16-битной адресации: 216 = 65536, начало – «0». Порты разделяются на три диапазона:

1. Общеизвестные (системные, well-known): 0 – 1023. Эти порты используются самыми популярными проколами и приложениями. Например, на порту 25 работает почтовый протокол SMTP, SSH на 22ом.
2. Зарегистрированные (пользовательские, registered): 1024 – 49151. Эти порты используются менее популярными протоколами/приложениями, например, сетевыми играми или, например, Skype. Однако в этом блоке есть можно сказать известные порты, например, 8080 – альтернатива 80 для HTTP сервера.
3. Динамические (частные, private) 49152 – 65535. Эти порты используются короткоживущими соединениями, например, клиент – сервер.

Первоначально из-за использования полудуплексного соединения портов требовалось два, теперь же хватает одного.

1. Сеансовый уровень

Уровень предназначен для установки, поддержания и завершения соединения между хостами. Сюда входит периодическая проверка соединения, его восстановление, либо перезагрузка, если восстановление невозможно. Выделяется 3 основных вида соединений:

1. Симплексное соединение

Оно предусматривает одностороннюю передачу данных, без отправки ответных сообщений, например, отправка информации с датчиков температуры и т.п.

1. Полудуплексное соединение

Предусматривает двусторонний обмен данными, но данные в текущей момент могут передаваться только в одну сторону, поэтому требуется разграничивать доступ к каналу связи по какому-либо правилу. Примером может выступать радиосвязь: один говорит, другой слушает.

1. Полный дуплекс

Предусматривает установление двух независимых каналов связи между участниками обмена. Это позволяет вести прием и передачу одновременно. Примером может выступать телефонная связь.

1. Уровень представления

Этот уровень служит для распознавания и форматирования данных перед их отправкой прикладному уровню для дальнейшей обработки. Некоторые функции представительского уровня: изменение кодировки данных, сжатие и расширение данных.

1. Прикладной уровень

Прикладной уровень является узлом между приложением и средой передачи, то есть уровнями 1-6. Сам 7ой уровень не содержит никаких приложений, он лишь является интерфейсом к среде передачи для приложения. Прикладной уровень зависит от типа приложения, как и физический уровень от используемой среды. Некоторые функции: идентификация участников, определение занятости приложений и ресурсов сети. Здесь работают протоколы HTTP, SMTP, FTP и др.

# Сетевая модель TCP/IP

Пока велась разработка OSI, модель TCP/IP набирала всё большую популярность. Эта модель была проще, имела 4 уровня. По своей сути модель очень похожа на OSI, но последние 3 уровня OSI объединены в один прикладной уровень TCP/IP. Также в этой модели физический и канальный уровни объединены в сетевой.

Эта модель вытеснила OSI, потому что OSI имеет строгую иерархичность. Каждый уровень должен быть составлен исключительно в соответствии с международным стандартом, поэтому, например, протокол QUIC*,* призванный заменить TCP+TLS, работающий на транспортном уровне, не может использоваться в этой модели. Этот протокол основан поверх протокола UDP. QUIC в отличие от TCP шифрует данные на транспортном уровне, а не передает для шифровки выше. Этот протокол является протоколом, который имеет множество улучшений как для ускорения трафика, так и для повышения уровня безопасности, тем не менее, его переход к его использованию повсеместно затрудняется из-за того, что NAT-роутеры ориентированы на работу с TCP, а не QUIC, то есть в данный момент большинство современных NAT-роутеров даже не знают о существовании QUIC, поэтому они обычно делают «даунгрейд» к обработке обычного UDP, что делает работу протокола более долгой.Также этот протокол имеет ряд проблем, которые могут быть исправлены переработкой некоторых и введением новых протоколов на разных уровнях. Всё это делает его невозможным для использования в OSI, ведь ради одного протокола придется переработать весь стандарт. Таким образом, стандартизация OSI не представляет возможности для добавления или улучшения её частей, поэтому OSI нигде и не используется.

# Domain Name System (DNS)

Когда пользователь загружает веб-страницу, должен произойти перевод между тем, что пользователь вводит в браузере, и нужным компьютеру IP-адресом. За это и отвечает DNS.

Это преобразование влечет за собой прохождение нескольких этапов обработки.

1. Локальные данные

Для максимально быстрого отклика, запущенная на компьютере служба DNS кэширует и сохраняет на уровне ОС все ответы на запросы к вышестоящему серверу, чтобы при повторном обращении к тому же ресурсу быстро вернуть нужны IP-адрес.

Если послать запрос с компьютера к DNS из терминала, то первым делом он отправился бы в кэш ОС. Если же говорить про браузеры, то большинство из них ныне имеют свои внутренние хранилища, где дополнительно кэшируются все запросы. Это делается, чтобы ускорить время возврата ответа еще больше, а также отказа от лишних запросов к операционной системе.

Так как IP-адреса у серверов могут меняться, при запросе адреса он отправляется не только к локально сохраненной информации, но и к вышестоящему серверу, полученный ответ от которого обновляет старую локальную запись.

Параметр, указывающий как часто нужно обращаться к серверу, игнорируя локальную запись, называется TTL (Time to live). Он указывается для каждой DNS записи и устанавливается авторитетным сервером имен. Обычно значение TTL устанавливается по умолчанию равным 24 часа, тем не менее, это значение может порой изменяться от секунд до недель.

Стоит отметить, что сохраненные DNS записи никогда автоматически не удаляются с компьютера. Они остаются лежать в кэше с давно истекшим TTL.

Время, на которое DNS запись будет сохраняться в конечном итоге обозначается разработчиками браузера, а не значением TTL, этот параметр обычно варьируется от 15 секунд до 30 минут.

1. Файл hosts

Файл hosts – это следующее место, где DNS пытается найти нужный IP-адрес. Если же данный запрос производится впервые и кэш ещё пуст, этот файл – первое место, куда попадет запрос.

Этот файл часто используется для перенаправления запросов на внутренние ресурсы сети

1. DNS-рекурсор

Если нужные данные не удалось получить локально, то запрос отправится на указанный в настройках сети DNS-сервер. Как правило каждый провайдер имеет собственный DNS-сервер.

Этот сервер, получая запрос, проверяет собственный серверный кэш и, если он смог его найти, то сразу возвращает искомый адрес. Если адрес не найдет, то DNS-сервер обращается от нашего имени к корневому серверу имён. Процесс работы DNS-сервера от нашего имени и дал ему название DNS-рекурсор.

Существует ещё один путь работы DNS-сервера. Если же он не нашел нужного у себя в кэше, то он не обращается выше, а возвращает список других серверов, где может быть окажется нужная запись, оставляя всю остальную работу на пользователя.

1. Корневой сервер имён

Корневой DNS-сервер находится на последнем уровне иерархии DNS. Он хранит адреса доменных серверов первого уровня (.com, .ru, .net). DNS пакет может содержать только 13 адресов корневых серверов, поэтому во всём мире существует только 13 таких DNS-серверов.

Благодаря созданию зеркал корневого сервера, коих сейчас больше 200, удалось существенно увеличить надежность всей системы и уменьшить время ответа, ускорив работу DNS.

1. Сервер имён TLD

Tot Level Domain (TLD) – домен верхнего уровня. Сервер TLD не хранит информацию о конкретных ресурсах, но хранит данные о серверах, на которых есть эта информация. DNS-рекурсор, посылая запрос TLD серверу, в ответ получает адрес сервера, на котором находится запрошенная запись.

1. Авторитетный сервер имён

Авторитетный сервер имён содержит всю информацию о запрашиваемом домене и может вернуть её на запрос DNS-рекурсора. После этого ответ будет перенаправлен браузеру, и пользователь сможет получить доступ к нужному сайту.

# Dynamic Host Configuration Protocol (DHCP)

IP-адрес в сети можно назначить вручную каждому устройству, но если речь идет о больших, то назначать каждому адрес – довольно трудоемкая задача. Для автоматизации этого процесса был введён протокол прикладного уровня модели TCP/IP – DHCP.

Discovery

Назначение адреса происходит так: изначально устройство находится в состоянии инициализации (INIT) и не имеет своего IP-адреса, поэтому оно отправляет широковещательное (broadcast) сообщение DHCPDISCOVER на все устройства в локальной сети, где присутствующий DHCP-сервер (в локальной сети DHCP-сервером может выступать роутер) ловит на 67 порту сообщение, а после получения отправляет DHCPOFFER на 68 порт устройству.

Offer

DHCP-сервер сообщает в DHCPOFFER IP-адрес, который может подойти устройству. Этот IP выделяется из области (Scope) доступных адресов, которую назначает администратор. Можно задавать доступные области адресов, можно и исключать нужные адреса. DHCP-сервер назначает адрес из scope временно, поэтому нет никакой гарантии, что при подключении к сети в следующий раз, устройство будет иметь тот же адрес. Тем не менее, имеется возможность забронировать некоторый адрес для конкретного устройства. Такое сохранение называется резерваций.

Request

Устройство, получая DHCPOFFER, отправляет обратно broadcast сообщение, которое почти не отличается от DHCPDISCOVER, но в нем содержится выделенный сервером IP-адрес. Таким образом, устройство подтверждает, что берет этот адрес. Сервер, получая DHCPREQUEST, окончательно подтверждает передачу IP-адреса клиенту сообщением DHCPACK. Это широковещательное или прямое сообщение утверждает не только владельца IP, но и срок работы этого адреса.

# Network Address Translation (NAT)

В локальных сетях IP-адреса обычно имеют вид 192.168.0.0/16, 10.0.0.0/8. Чтобы устройство с таким адресом могло обратиться к другим ресурсам вне этой сети нужно иметь механизм преобразования частного адреса в публичный. За это преобразование отвечает NAT.

Преобразования адресов производится следующим образом: принимая пакет от локального устройства, роутер обрабатывает IP-адрес назначения. Если это локальный адрес, то пакет отправляется другому локальному устройству. Иначе, роутер подменяет обратный IP-адрес пакета на свой внешний IP-адрес и меняет номер порта (чтобы различать ответные пакеты). Комбинацию, нужную для обратной подстановки, роутер сохраняет у себя во временной таблице. По истечению обмена между клиентом и сервером, роутер стирают в таблице информацию об определенном номере порта.

Типы NAT

1. Static NAT

В этом случае единственный внутренний адрес преобразуется в единственный внешний. Все запросы, приходящие на внешний адрес, транслируются на внутренний.

1. Dynamic NAT

Здесь производится работа с пулом публичных адресов, которые назначаются по номеру в очереди. При запросе от внутреннего устройство производится подмена IP-адреса из доступного пула публичных адресов.

1. Port Address Translation (PAT)

Наиболее распространенный тип NAT. Здесь подключение осуществляет трансляцию нескольких частных адресов на один или несколько общедоступных. Сопоставление нескольких частных адресов одному или нескольким общедоступным возможно благодаря отслеживанию каждого локального адреса по номеру порта. Таким образом, адресом источника будет локальный адрес устройства с добавлением номера порта, а адресом назначения будет внешний локальный адрес с добавлением номера служебного порта.

Введение этого механизма позволило: решить проблему нехватки IP-адресов и отсрочить переход к IPv6 от IPv4, обеспечить безопасность, когда нужно скрыть внутренний IP-адрес от внешней сети, или скрыть номер порта.

# Заключение

В процессе выполнения курсового проекта я изучил устройство кабинета, в котором проходят практические занятия у моей группы. Я разобрался с устройством сетевых моделей OSI и TCP/IP, немного изучил работу протокола транспортного уровня QUIC, который является более совершенной заменой TCP вместе с TLS. Также я рассмотрел DNS и NAT, которые показывали мне какое огромное количество операций происходит после вписывания имени сайта в адресную строку браузера.

# Источники

1. Интернет-магазин, статья «Что такое компьютерная сеть?»

<https://aws.amazon.com/ru/what-is/computer-networking/>

1. Информационный портал, статья «Основы компьютерных сетей. Тема №1. Основные сетевые термины и сетевые модели»

<https://habr.com/ru/post/307252/>

1. Информационный портал, статья «MAC-адрес (Media Access Control — управление доступом к носителю)»

<https://www.kipis.ru/info/index.php?ELEMENT_ID=42901>

1. Информационный портал, статья «Сетевой уровень модели OSI. Общие понятия»

<https://www.oslogic.ru/knowledge/809/setevoj-uroven-modeli-osi-obshhie-ponyatiya/>

1. Информационный портал, статья «По пути к QUIC: что лежит в основе HTTP/3»

<https://habr.com/ru/company/Voximplant/blog/430436/>

1. Информационный портал, серия статей об уровнях модели OSI

<http://celnet.ru/OSI5.php>

<http://celnet.ru/OSI6.php>

<http://celnet.ru/OSI7.php>

1. Блог Михаила Шпакова, статья «Всё, что нужно знать про DNS»

<https://shpakov.dev/blog/all-you-need-to-know-about-dns>

1. Информационный портал, статья «Принципы работы протокола DHCP»

<https://selectel.ru/blog/dhcp-protocol/>

1. Блог, «Технология преобразования сетевых адресов (NAT)»

<https://www.smart-soft.ru/blog/tehnologija_preobrazovanija_setevyh_adresov_nat/>