

*Куреева Е.С.*

*Студентка 5 курс*

*факультет «Комплексная безопасность ТЭК» Российский  
государственный университет нефти и газа (НИУ) имени*

*И.М. Губкина*

*Россия, г. Москва*

*Федосова А.С.*

*Студентка 5 курс*

*факультет «Комплексная безопасность ТЭК»*

*Российский государственный университет нефти и газа (НИУ) имени*

*И.М. Губкина*

*Россия, г. Москва*

**ИССЛЕДОВАНИЕ ВРЕМЕНИ ЗАДЕРЖКИ И ПОТЕРЬ ПАКЕТОВ В  
СЕТИ С ИСПОЛЬЗОВАНИЕМ UDP И TCP ПРОТОКОЛОВ С  
ПОМОЩЬЮ СИМУЛЯТОРА СЕТИ NS-3**

*Аннотация:* в статье рассматривается исследование времени задержки и потерь пакетов в сети с использованием протоколов UDP и TCP. Для проведения исследования использовался сетевой симулятор ns-3, который позволяет моделировать различные сценарии работы сети. В работе представлены результаты экспериментов, демонстрирующие влияние указанных параметров на показатели задержки и потерь пакетов. Полученные данные позволяют сделать выводы о применимости каждого из протоколов в различных сетевых условиях.

**Ключевые слова:** транспортные протоколы, UDP, TCP, сетевой симулятор, ns-3, задержка пакетов, потери пакетов, пропускная способность, моделирование сети, производительность сети, экспериментальная симуляция.

**Abstract:** the article examines the study of delay time and packet loss in a network using the UDP and TCP protocols. To conduct the study, the ns-3 network simulator was used, which allows you to simulate various network scenarios. The paper presents the results of experiments demonstrating the influence of these parameters on the delay and packet loss indicators. The data obtained allow us to draw conclusions about the applicability of each of the protocols in different network conditions.

**Keywords:** transport protocols, UDP, TCP, network simulator, ns3, packet delay, packet loss, bandwidth, network simulation, network performance, experimental simulation.

## **Введение**

Для современных компьютерных сетей стало важным не только наличие соединения, но и качество передаваемых данных, которое характеризуется различными метриками, такими как пропускная способность, задержка и потери пакетов. Эти метрики особенно критичны для приложений реального времени, таких как видеоконференции, стриминг и тому подобное, где даже небольшие изменения в параметрах сети могут существенно повлиять на качество обслуживания.

В сетях, основанных на передаче данных, наиболее часто используются два транспортных протокола: UDP (User Datagram Protocol) и TCP (Transmission Control Protocol). Они играют важную роль в обеспечении обмена данными, но различаются по принципам работы и поведению в сети.

## **Принципы работы UDP и TCP протоколов**

Одни из основных протоколов транспортного уровня — User Datagram Protocol (UDP) и Transmission Control Protocol (TCP). Протоколы этого уровня необходимы для адресации пакетов между портами приложения отправителя и приложения получателя.

Протокол UDP характеризуется простотой, что делает его быстрым, но уязвимым к потере пакетов и задержкам. В свою очередь TCP предлагает надёжную доставку данных за счёт установки соединения и контроля потока, что происходит за счет увеличения времени передачи. Эти различия делают оба протокола применимыми для разных задач, в зависимости от требований к передаче данных.

### **User Datagram Protocol**

По информации из официального документа RFC-768, протокол передачи пользовательских дейтаграмм (User Datagram Protocol) нужен для поддержания режима обмена дейтаграммами. Он основан на коммутации пакетов между собой в компьютерной сети. Протокол UDP применяется для связи, не устанавливая соединение, то есть происходит пересылка пакетов от одного узла к другому. При этом при отправке не проверяется информация о получателе, готов ли он к приемке пакетов и вообще существует ли. После отправки этот узел не ждет какого-либо подтверждения о полученных пакетах [13].

Основная особенность, уже описанная ранее, — это отсутствие установки соединения перед передачей данных, что делает UDP быстрым, но мало надёжным, так как нет никаких гарантий достижения назначения, приводящее к потере данных.

Далее рассмотрим такую характеристику как отсутствие подтверждения. При получении пакетов отправитель не получает никакого подтверждения об

этом. Такое решение значительно снижает нагрузку на сеть вследствие отсутствия дополнительных пакетов, перемещающихся по сети, но опять же увеличивает риск потери данных.

При работе с UDP отсутствует управление потоком данных между получателем и отправителем. То есть отправитель может отправлять пакеты с большей скоростью, чем может принять и обработать их получатель. Такая ситуация тоже приводит к уже названным основным проблемам UDP в случае превышения скорости передачи над пропускной способностью сети.

### **Transmission Control Protocol**

Протокол TCP является основным элементом при организации стабильной передачи данных, без каких-либо потерь, из-за чего он становится важным для многих сфер. Задача протокола, по RFC-793, обеспечение надежных и потенциально безопасных логических устройств или соединений между парами процессов. Этот протокол обеспечивает гарантированную доставку пакетов без потерь и в правильном порядке, что осуществляется с помощью контролирования потока и перегрузок для эффективного использования сетевых ресурсов [14].

Упорядоченность протокола TCP осуществляется с помощью использования при отправлении нумерации каждого пакета и восстановления получателем данных за счет присвоенных пакетам номеров. Это свойство помогает в случае частичной потери и повторной отправки данных, что позволяет избежать дублирования пакетов.

TCP обеспечивает надежную передачу данных благодаря использованию механизма подтверждения и повторной отправки. Перед тем, как отправляется каждый пакет, устанавливается связь между отправителем и получателем, то есть handshake (рукопожатие). В начале отправитель отправляет запрос на передачу информации, ожидая ответ с готовностью, а получатель, в свою

очередь, отправляет запрос о получении, если получил запрос. Если подтверждение не приходит, данные автоматически отправляются заново, для этого для каждого пакета есть таймер на подтверждение. Этот процесс гарантирует доставку всех пакетов, даже если некоторые из них потеряны по пути и время отправки информации увеличено. При этом сохраняется защита от дублирования пакетов: если один и тот же пакет получен несколько раз, то он игнорируется. Также, существует Sliding Window (скользящее окно), которое используется для подтверждения нескольких пакетов [6].

Управление перегрузками сети в протоколе включает алгоритмы медленного старта и избегания перегрузок (TCP Congestion Control), которые регулируют скорость передачи данных в зависимости от состояния сети. Медленный старт увеличивает объем передаваемых данных постепенно, избегая резких скачков нагрузки на сеть.

Механизм контроля потока в TCP предотвращает перегрузку получателя, регулируя объем передаваемых данных в соответствии с его возможностями. Если сеть перегружена, TCP снижает скорость передачи, а при освобождении ресурсов — увеличивает ее, что оптимизирует использование сетевых возможностей и минимизирует потери.

### **Параметры передачи данных в UDP и TCP протоколах**

Протоколы UDP и TCP зависят от некоторых параметрах, которые определяют эффективность передачи данных. Пропускная способность, потери фреймов, задержка и минимальный размер MTU играют ключевую роль в сетевой производительности. Анализ этих параметров позволяет понять поведение сети и правильно настроить соединения для обеспечения стабильной передачи данных для необходимых условий [2].

Одной из ключевых метрик является пропускная способность, под которой понимается объем данных, который можно передать из одной точки в другую за определенное время.

Также важным показателем является задержка – время, за которое отправленный пакет доходит до конечной точки («end-to-end»). Производительность сети, которая состоит из нескольких сегментов или подсетей, несмотря на преобладания высокой пропускной способности в большинстве сегментов, будет ограничиваться скоростью самого медленного соединения между источником и получателем. Вследствие один слабый участок станет узким местом и снизит общую эффективность всей сети [5].

Производительность — это объем данных, передаваемый через подключение за определённый промежуток времени. В реальности она часто оказывается ниже заявленной пропускной способности из-за множества факторов, влияющих на передачу данных на физическом уровне. Таким образом, реальная производительность обычно уступает теоретически возможной пропускной способности.

Следующим важным параметром является потеря пакетов – явление, при котором один или несколько пакетов не достигают конечного назначения. Многие протоколы, такие как UDP, не гарантируют доставку пакетов, а просто пересылают данные в нужном направлении. Протоколы с гарантией доставки, например TCP, могут терять в пропускной способности из-за необходимости повторной передачи кадров.

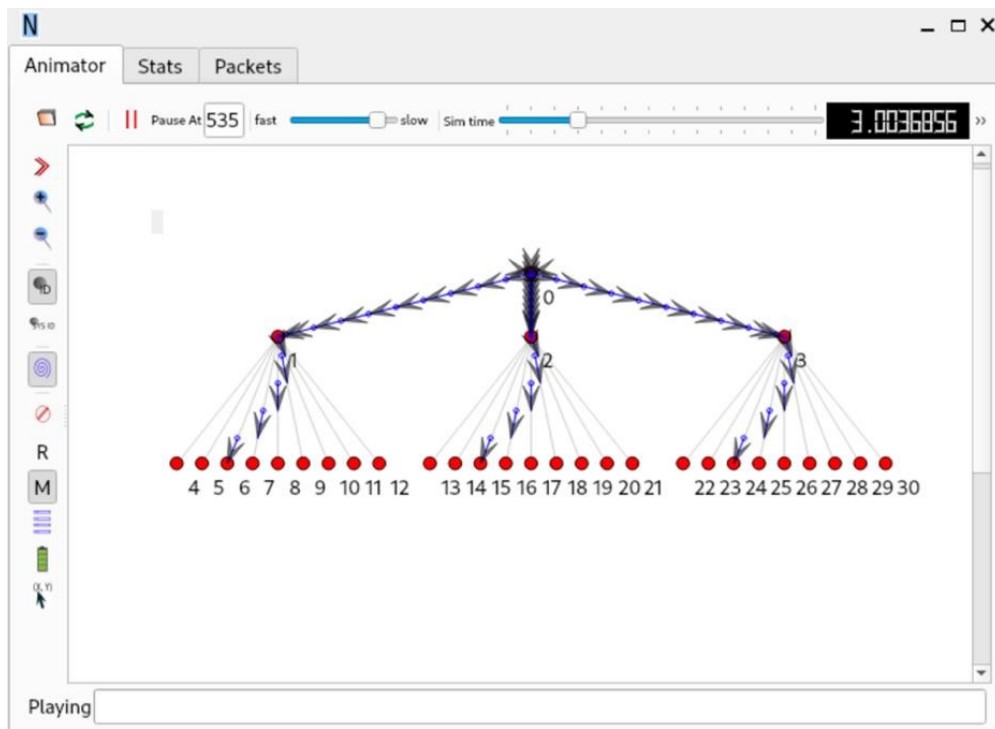
Еще одной метрикой является MTU. Максимальная единица передачи — это размер самого большого пакета, который может передать сеть. Все что будет иметь больший размер, чем установленный MTU, будет разбиваться на более мелкие фрагменты, что значительно замедлит передачу [2].

## **Анализ UDP и TCP протоколов на основе полученных результатов**

В ходе анализа работы протоколов UDP и TCP с использованием симулятора ns-3 в рамках сети, состоящей из 31 узла, представленных в топологии дерево, было исследовано время задержки и потери пакетов для каждого протокола в условиях повышения нагрузки на сеть. В качестве основных метрик использовались следующие значения: допустимый уровень потерь пакетов не должен превышать 1% [1], а задержка передачи данных — 200 мс [7]. Основная идея заключается в измерении времени задержки и количества потерянных пакетов при изменении объема трафика, которая осуществляется за счет изменения числа отправляемых пакетов, увеличивая их количество от 100 до 1000.

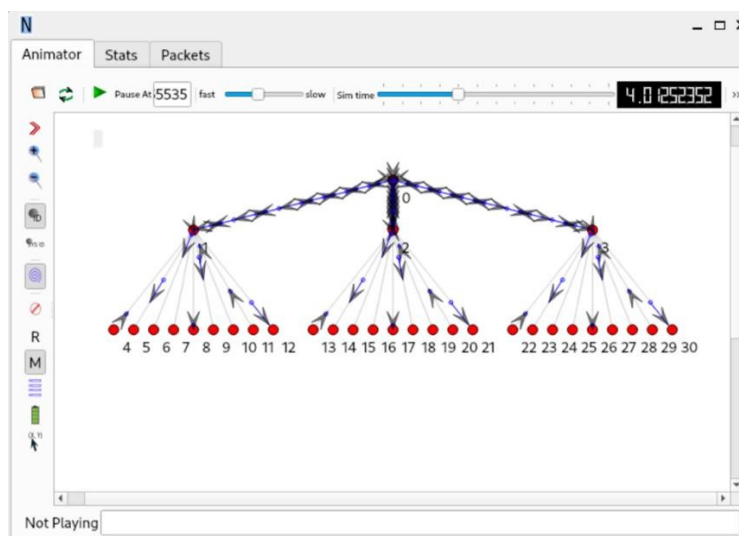
В результате проведения двух экспериментов на основе разных протоколов были получены результаты в виде визуализации созданной симуляции, данных мониторинга трафика и графиков, исходя из которых можно провести анализ поведения UDP и TCP в условиях одинаковой топологии.

При симуляции кода с настроенным UDP протоколом, анимируя работу сети с использованием пакета NetAnim (Рисунок 1), можно проследить, что при отправке данных пакеты следуют от отправителя к получателю и по достижению конечного узла, общение отправителя и получателя завершается.



**Рисунок 1. Анимация работы сети с использованием UDP протокола**

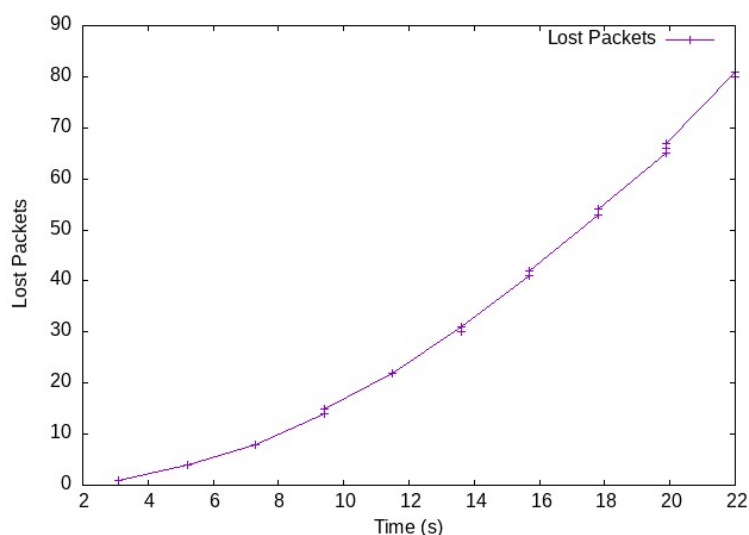
Иным образом происходит симуляция сети при настроенном TCP протоколе. Для того, чтобы начать отправлять необходимые данные, между узлами устанавливается соединение, далее пакеты отправляются от отправителя к получателю и после достижения пакетов своей цели, отправляется подтверждение об этом. На Рисунке 2 можно заметить, что пакеты идут не только от корневого узла к конечным, но и в обратном направлении.



## ***Рисунок 2. Анимация работы сети с использованием TCP протокола***

На основе данных анимации можно было наблюдать, что UDP — это протокол без установления соединения, ориентированный на доставку пакетов без подтверждений, тогда как TCP — протокол с установлением соединения, обеспечивающий доставку данных с подтверждением.

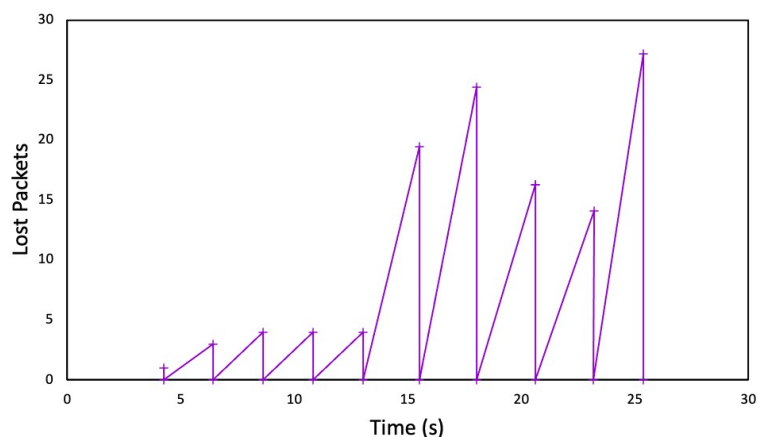
По результатам данных мониторинга был построен график зависимости потерянных пакетов (Lost Packets) с течением времени (Times) (Рисунок 3). По данным графика видно, что потери пакетов относительно равномерно увеличиваются со временем, что также соотносится с равномерным увеличением количества отправляемых пакетов.



***Рисунок 3. График зависимости потери пактов со временем в сети с UDP протоколом***

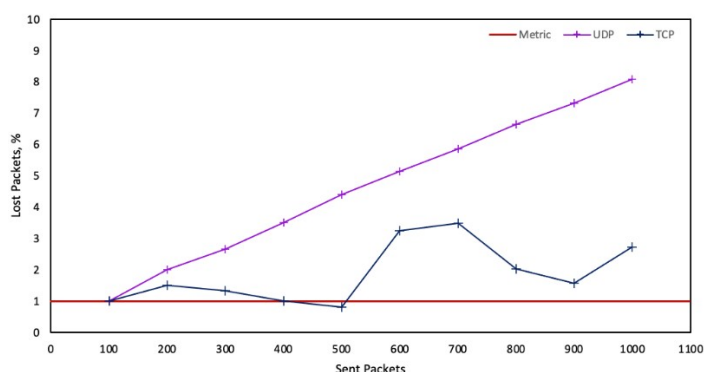
При идентичных параметрах сети с настроенным TCP протоколом отправка пакетов несколько отличается. Изначально, при отправке данных, пакеты отправляются не фиксированным объемом, а с учетом балансировки нагрузки на сеть, вследствие и количество потерянных пактов не всегда равномерно. Также необходимо учитывать, что в случае неполной отправки пакетов они отправляются повторно. На графике зависимости потерянных пакетов (Lost Packets) с течением времени (Times) (Рисунок 4) Видно, что

потеря пакетов значительно ниже, чем при использовании UDP протокола, но не исключена из-за специальных настроек сети.



**Рисунок 4. График зависимости потери пакетов со временем в сети с TCP протоколом**

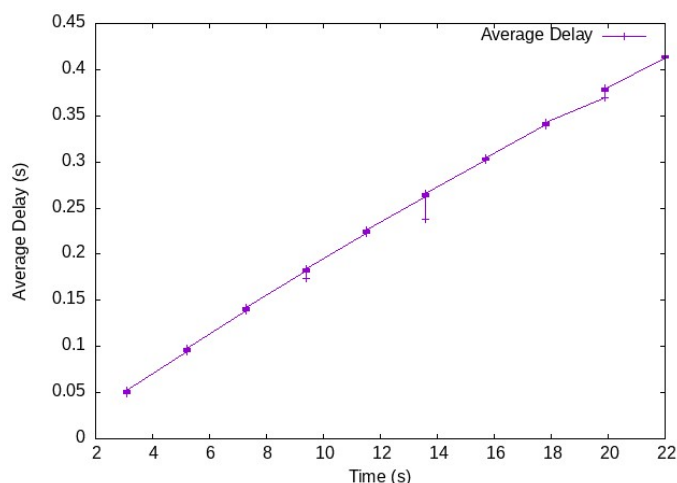
Представим графики выше в виде общего графика зависимости потери пакетов, сделав их более наглядными, относительно проведенного исследования с увеличением числа пакетов (Рисунок 5).



**Рисунок 5. График зависимости потери пакетов от увеличения отправляемых пакетов**

На графике представлена процентное соотношение потери пакетов от общего числа отправленных пакетов в каждый этап отправки, а также условно общепринятое значение максимальной потери пакетов. В рамках исследуемой топологии потери пакетов превышали метрику, что связано с увеличением нагрузки на сеть.

При исследовании задержки сети в проведенном эксперименте с UDP протоколом, на графике зависимости задержки (Average Delay) от времени (Time) (Рисунок 6) можно заметить, что ее значение также равномерно увеличивается, как и количество потерянных пакетов в этой симуляции. Чем больше становится отправленных пакетов, тем сильнее увеличивается нагрузка на сеть.

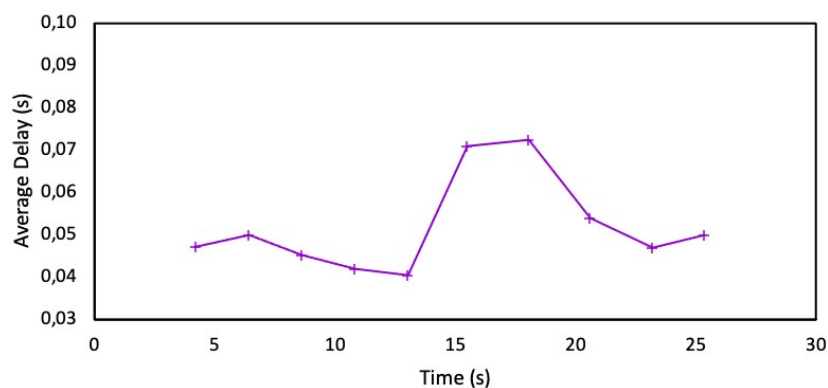


***Рисунок 6. График зависимости задержки со временем в сети с UDP протоколом***

В случае исследования графика зависимости задержки (Average Delay) от времени (Time) (Рисунок 7) для TCP протокола, можно заметить, что значения задержки хоть и не так стабильны, но имеют гораздо меньшую величину, чем при использовании UDP. Это связано с тем, что TCP применяет контроль потока и управление перегрузкой, регулируя скорость отправки данных и адаптируясь к состоянию сети, что помогает предотвратить перегрузки и снижает вероятность потерь пакетов, а также минимизирует колебания в задержке.

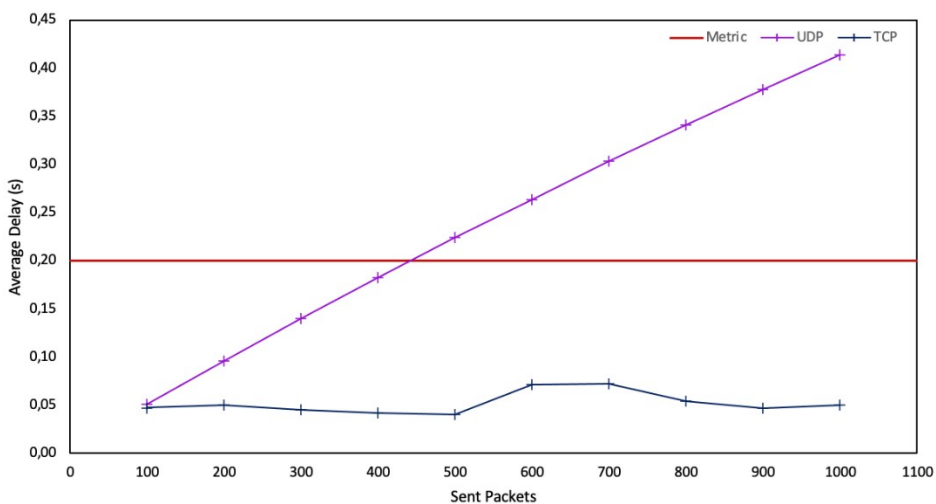
Кроме того, TCP гарантирует доставку данных, включая повторную передачу утерянных сегментов, что, с одной стороны, увеличивает время самой симуляции из-за временных затрат на проверки и пересылки, но с другой

– позволяет поддерживать стабильное соединение и минимизировать задержку.



**Рисунок 7. График зависимости задержки со временем в сети с ТСР протоколом**

Представим графики выше в виде общего графика зависимости задержки, сделав их более наглядными, относительно проведенного исследования с увеличением числа пакетов (Рисунок 8).



**Рисунок 8. График зависимости потери пакетов от увеличения отправляемых пакетов**

На графике представлена процентное соотношение задержки от общего числа отправленных пакетов в каждый этап отправки, а также условно общепринятое значение задержки. В рамках исследуемой топологии задержка при протоколе UDP сильно превышает метрику, что связано с увеличением

нагрузки на сеть и отсутствием какой-либо балансировки нагрузки в сети, которая есть у TCP, что мы и видим на графике, задержка в данном случае минимальная.

### **Заключение**

В результате проведенных экспериментов можно сделать вывод, что UDP и TCP протоколы демонстрируют существенные различия в поведении и эффективности передачи данных при одинаковой топологии сети. UDP, как протокол без установления соединения, ориентирован на быструю отправку пакетов, но при этом не гарантирует их доставку, что приводит к потере данных с нарастанием сетевой нагрузки. Это подтверждается результатами графиков, где количество потерянных пакетов и задержка увеличиваются с течением времени.

С другой стороны, TCP, за счет своих механизмов контроля потока, управления перегрузкой и подтверждения передачи, демонстрирует более стабильную и надежную передачу данных, даже несмотря на временные затраты на установление соединения и подтверждение доставки. Эти особенности TCP позволяют ему снижать величину задержки и минимизировать потери пакетов, что особенно важно в условиях высокой сетевой нагрузки.

Таким образом, на основании проведенного исследования можно сделать вывод, что выбор протокола передачи данных должен основываться на требованиях к качеству обслуживания. Для приложений, критичных к задержке, целесообразно использовать UDP, тогда как для обеспечения надежности и целостности данных предпочтительнее TCP.

### **Использованные источники:**

1. Качество обслуживания в операторских сетях // OpenNET URL: [https://www.opennet.ru/docs/RUS/qos\\_oper/](https://www.opennet.ru/docs/RUS/qos_oper/) (дата обращения: 01.10.2024).
2. Качество сетей передачи данных. Программные и аппаратные измерения // Хабр URL: <https://habr.com/ru/articles/250821/> (дата обращения: 01.10.2024).
3. Моделирование телекоммуникационной сети средствами сетевых инструментов Linux: инструменты создания цифровых двойников / А.Г. Уймин, О. Р. Никитин // I-methods. – 2023. – Т. 15, № 2. – EDN NFJDVH
4. Отличия TCP- и UDP-протоколов — определяем разницу на примерах // Selectel URL: <https://selectel.ru/blog/tcp-vsudp/?ysclid=m2v2d924hu815014337> (дата обращения: 01.10.2024).
5. Пропускная способность передачи данных // NOCIP URL: <https://nocip.ru/network/propusknaya-sposobnost-peredachidannyh/?ysclid=m2nf5svipe888354436> (дата обращения: 01.10.2024).
6. Транспортные протоколы TCP и UDP // HackWare.ru URL: <https://hackware.ru/?p=12935> (дата обращения: 01.10.2024).
7. Что такое задержка сети и как уменьшить её? // FS URL: <https://community.fs.com/ru/article/understanding-network-latency-inethernet-switches.html?ysclid=m34t28mvre380021229> (дата обращения: 01.10.2024).
8. Documentation // ns-3 URL: <https://www.nsnam.org/documentation/> (дата обращения: 01.10.2024).
9. How To Implement TCP Protocols In Ns3 // Ns 3 Simulation URL: <https://ns3simulation.com/how-to-implement-tcp-protocols-in-ns3/> (дата обращения: 01.10.2024).

10. How To Implement Tree Topology In Ns3 // Ns 3 Simulation URL: <https://ns3simulation.com/how-to-implement-tree-topology-in-ns3/> (дата обращения: 01.10.2024).
11. Lai Y. C. et al. Performance modeling and analysis of TCP and UDP flows over software defined networks //Journal of Network and Computer Applications. – 2019. – Т. 130. – С. 76-88.
12. TCP и UDP, или Два столпа Интернета // Хабр URL: <https://habr.com/ru/articles/711578/> (дата обращения: 01.10.2024).