Шевченко Дмитрий Александрович
студент, Российский государственный университет нефти и газа И.М. Губкина, РФ, г. Москва
Боготов Ислам Сергеевич
студент, Российский государственный университет нефти и газа И.М. Губкина, РФ, г. Москва

НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ VPN СОЕДИНЕНИЯ НА БАЗЕ WIREGUARD

Аннотация. В статье представлены результаты нагрузочного тестирования протокола WireGuard и его модификации RuWireGuard с отечественными криптографическими алгоритмами ГОСТ [1–3, 5, 12]. Проведено сравнение производительности двух реализаций на тестовом стенде из двух виртуальных машин ALT Workstation 11.1 с использованием инструментов iperf3, ITGsend и irtt. Оцениваются пропускная способность, задержки, загрузка CPU и другие метрики при различных нагрузках, выявляются ограничения RuWireGuard и определяются сценарии применения.

Abstract. This article presents the results of load testing of the WireGuard protocol and its modification RuWireGuard with domestic cryptographic algorithms GOST [1–3, 5, 12]. A comparison of the performance of the two implementations was carried out on a test bench consisting of two ALT Workstation 11.1 virtual machines using the iperf3, ITGsend, and irtt tools. Throughput, latency, CPU load, and other metrics were evaluated under various loads, the limitations of RuWireGuard were identified, and application scenarios were determined.

Ключевые слова: нагрузочное тестирование, WireGuard, RuWireGuard, ГОСТ, производительность VPN, iperf3, ITGsend, пропускная способность, задержка, Кузнечик

Keywords: load testing, WireGuard, RuWireGuard, GOST, VPN performance, iperf3, ITGsend, throughput, latency, Kuznechik

Введение

В настоящее время VPN-технологии широко применяются для защиты сетевого трафика, поэтому при выборе решения важно учитывать не только стойкость криптографии, но и производительность при реальных нагрузках. Протокол WireGuard рассматривается как современная альтернатива традиционным VPN-решениям, а также является основой для работ по адаптации под отечественные криптографические алгоритмы. Одним из таких вариантов является RuWireGuard — модификация WireGuard с использованием алгоритмов ГОСТ, для которой в литературе отмечается потенциальное снижение производительности относительно wireguard-go. Цель данной работы — выполнить сравнительное нагрузочное тестирование WireGuard и RuWireGuard на стенде из двух виртуальных машин ALT Workstation 11.1 с применением iperf3, ITGsend и irtt и оценить ключевые показатели (пропускная способность, задержка, джиттер и загрузка CPU) для определения практических сценариев применения RuWireGuard.

Виртуальные частные сети: назначение и требования

Виртуальные частные сети (VPN) — это технология для защиты передачи данных через открытые и потенциально небезопасные каналы связи, расширяющая приватную сеть на открытые каналы. VPN применяются для защиты трафика в корпоративных сетях от перехвата и анализа, безопасного соединения удалённых офисов и филиалов, защиты критических данных и обеспечения конфиденциальности в открытых сетях.

Для эффективного выполнения этих задач VPN-протоколы должны обеспечивать чётко определённый набор свойств безопасности:

- Конфиденциальность гарантирует, что передаваемые данные не могут быть прочитаны перехватившей их стороной, достигается криптографическим шифрованием пакетов.
- Целостность гарантирует, что данные не были изменены в пути передачи, реализуется через коды аутентификации сообщений и хеш-функции. [1–3, 10]

- Аутентификация обеспечивает проверку подлинности партнёров коммуникации и исключает участие неавторизованных узлов, реализуется криптографическими механизмами обмена ключами.
- Высокая производительность — способность обрабатывать большие объёмы данных с минимальными задержками — критична для production-среды и систем с ограниченными ресурсами.

Криптография как основа VPN

Реализация перечисленных требований невозможна без криптографии — методов защиты информации путём преобразования её в недоступную для чтения форму. В VPN-протоколах используются три основных класса криптографических алгоритмов, каждый решает специфичные задачи. [1–3, 6, 10]

Алгоритмы асимметричного шифрования применяются для установления защищённого канала и обмена ключами между сторонами, ранее не имевшими общего секрета. Основаны на математических проблемах, вычислительно сложных в одном направлении (факторизация больших чисел, дискретный логарифм, эллиптические кривые). Позволяют публично распространять открытые ключи без компрометации безопасности, но вычислительно затратны.

Алгоритмы симметричного шифрования используются для защиты основного потока данных после установления соединения. Работают с единственным секретным ключом, известным обеим сторонам. Обеспечивают высокую производительность благодаря оптимизированной реализации и применению специализированных процессорных инструкций. При одинаковом уровне криптографической стойкости симметричные алгоритмы значительно быстрее асимметричных.

Криптографические хеш-функции используются для проверки целостности данных и формирования кодов аутентификации. Преобразуют входные данные произвольной длины в выход фиксированной длины с свойством, что малое

изменение входа приводит к полному изменению выхода. Это позволяет эффективно обнаруживать несанкционированные модификации данных.

Необходимость оценки эффективности VPN-решений

На практике выбор конкретного VPN-решения для организации требует не только наличия криптографической стойкости, но и оценки практических характеристик работы. Различные VPN-решения, даже при использовании криптографических алгоритмов с эквивалентным уровнем безопасности, могут существенно отличаться по производительности, требованиям к ресурсам и стабильности в различных условиях.

Организация должна убедиться, что выбранное решение может справиться с объёмом трафика, поддерживать приемлемые задержки для используемых приложений и работать в рамках доступных аппаратных ресурсов. Кроме того, практическое тестирование имеет образовательную ценность, развивая у специалистов понимание взаимосвязи между архитектурой, выбором алгоритмов и реальной производительностью системы. Такие навыки критичны для профессионалов в области кибербезопасности.

Методы оценки производительности VPN-систем

Для объективной оценки пригодности VPN-решения необходимо измерить несколько ключевых метрик производительности.

- Пропускная способность — максимальный объём данных, передаваемый через туннель за единицу времени (в битах/сек). Определяет пригодность системы для высоконагруженных приложений.
- Задержка (latency) — время, требуемое пакету для прохождения от источника к назначению. Критична для real-time приложений (VoIP, видеоконференции, системы управления критической инфраструктурой).
- Джиттер — вариативность задержки пакетов. Высокий джиттер приводит к деградации качества real-time приложений.

- Использование ресурсов (CPU, память) — объём вычислительных ресурсов, необходимых системе. Важно для систем с ограниченными ресурсами и оценки масштабируемости.
- Потери пакетов — процент пакетов, не достигших пункта назначения. Влияет на надёжность передачи данных.

Измерение этих метрик при различных сценариях нагрузки позволяет составить полную картину производительности системы.

Инструменты нагрузочного тестирования VPN-систем

Для комплексной оценки производительности VPN-решений применяются три основных инструмента:

- iperf3 измеряет пропускную способность при параллельных потоках TCP/UDP, генерируя идеализированный поток данных. Синтаксис: iperf3 -c <IP> -t <время> -P <потоки>, где параметры задают целевой хост, длительность теста в секундах и количество параллельных соединений.
- ITGsend (D-ITG) генерирует реалистичный трафик с определёнными интервалами и размерами пакетов, имитируя приложения (VoIP, видеоконференции, файловая передача). В отличие от iperf3, позволяет оценить поведение VPN при реальных паттернах нагрузки. Команда: /ITGSend -a <IP> -T UDP -C <пак/сек> -c <размер> -t <мс>.
- irtt анализирует задержки и джиттер, фиксируя время прохождения каждого пакета при регулярных интервалах отправки. Вычисляет среднюю задержку, стандартное отклонение и экстремальные значения, критичные для оценки пригодности к real-time приложениям.

Комбинированное использование этих инструментов обеспечивает полную оценку: iperf3 показывает максимальный потенциал, ITGsend демонстрирует реальное поведение, irtt раскрывает характеристики задержек.

ОБЗОР WIREGUARD И RUWIREGUARD

Архитектура и криптографические основы WireGuard

WireGuard — это современный VPN-протокол с фиксированным набором криптографических алгоритмов, выбранных для баланса между безопасностью и производительностью:

Функция	Алгоритм	Описание
Обмен ключами	Curve25519	Эллиптическая кривая для обмена ключами Диффи-Хеллмана
Шифрование	ChaCha20	Потоковый шифр с высокой производительностью
Аутентификация (MAC)	Poly1305	Код аутентификации сообщения
Хеш-функция	BLAKE2s	Криптографическое хеширование
Вспомогательные	SipHash24, HKDF	Для хеш-таблиц и получения подключей

WireGuard обеспечивает криптографическую стойкость, эквивалентную AES-256 и HMAC-SHA256, с высокой производительностью без специализированного оборудования. Алгоритмы разработаны ведущими криптографами и имеют многолетнюю историю анализа.

Модификация RuWireGuard с отечественной криптографией

RuWireGuard — модификация WireGuard, разработанная в контексте требований российского законодательства о применении отечественных криптографических алгоритмов. Реализует весь функционал оригинального протокола, но заменяет криптографические компоненты на ГОСТ-аналоги, обеспечивая соответствие нормативным требованиям при сохранении базовой архитектуры. [1–3, 6, 7, 9]

Соответствие компонентов:

Функция	WireGuard	RuWireGuard
Обмен ключами	Curve25519	ГОСТ Р 34.10-2012 (эллиптические кривые)
Шифрование данных	ChaCha20	ГОСТ Р 34.12-2015 «Кузнечик» (Grasshopper)
Аутентификация	Poly1305	ГОСТ Р 34.13-2015 (режим гаммирования)
Хеширование	BLAKE2s	ГОСТ Р 34.11-2012
Получение подключей	HKDF	ГОСТ Р 34.13-2015

ГОСТ Р 34.10-2012 обеспечивает уровень безопасности 128 бит, сравнимый с Curve25519. Кузнечик — 128-битный блочный шифр с 256-битным ключом, работающий по принципу сетей Фейстеля с 10 раундами. ГОСТ Р 34.11-2012 — хеш-функция с выходом 256 бит, функционально эквивалентная BLAKE2s. Замена сохраняет структуру Noise Protocol Framework.

WireGuard использует алгоритмы, разработанные мировыми криптографами, имеющие многолетнюю историю криптоанализа и оптимизации. ГОСТ-алгоритмы, хотя и прошли проверку ФСБ, имеют меньше публичного криптоанализа. Криптографические операции на эллиптических кривых ГОСТ и Кузнечик требуют больше вычислительных ресурсов в текущих реализациях. [1–3, 6, 7, 9]

Сравнение подходов и архитектурные различия

WireGuard реализован как модуль ядра Linux, исключая контекстные переключения между пользовательским пространством и ядром при обработке каждого пакета. Это обеспечивает минимальные накладные расходы, эффективное использование кеша и прямой доступ к оптимизированным структурам данных.

RuWireGuard реализуется как пользовательское приложение (userspace) на Go. Каждый пакет требует контекстного переключения, вносящего дополнительные накладные расходы на управление памятью и синхронизацию потоков.

Кроме того, WireGuard использует алгоритмы, оптимизированные мировыми криптографами с многолетней историей публичного анализа. ГОСТ-алгоритмы имеют меньше публичного криптоанализа и оптимизированных реализаций, требуя больше вычислительных ресурсов .

RuWireGuard и WireGuard несовместимы на криптографическом уровне и не могут напрямую обмениваться данными. Они совместимы только на уровне протокола управления.

Развертывание и применение

Сравнение развертывания WireGuard и RuWireGuard:

Этап	WireGuard	RuWireGuard
Установка	apt-get install wireguard-tools wireguard-tools-wg-quick	apt-get install git golang make gcc pkg-config
Получение кода	Встроена в систему	git clone https://github.com/bi-zone/ruwireguard-go.git && cd ruwireguard-go
Сборка	Не требуется	make
Генерация ключей	wg genkey, wg pubkey	./wg genkey, ./wg pubkey
Инициализация	wg-quick up wg0 (автоматическая)	./wireguard-go wg0 (явная)
Загрузка конфигурационного файла	Автоматическая в wg-quick	./wg set (явная)

Поднятие интерфейса	Автоматическое	ip link set wg0 up (явное)
Присвоение IP	Автоматическое	ip addr add (явное)

ПРАКТИЧЕСКАЯ ЧАСТЬ И АНАЛИЗ РЕЗУЛЬТАТОВ

Методология и инструменты нагрузочного тестирования

Нагрузочное тестирование проведено на тестовом стенде из двух виртуальных машин ALT Workstation 11.1, соединённых по локальной сети.

Параметры стенда:

Операционная система	ALT Workstation 11.1
Версия ядра	Linux 6.12.41-6.12-alt1
Количество ядер CPU	2 ядра
Оперативная память	4 ГБ

Тестирование включает три сценария оценки производительности:

Сценарий 1 — iperf3: Измерение пропускной способности при увеличении параллельных соединений. Позволяет оценить максимальный потенциал и масштабируемость.

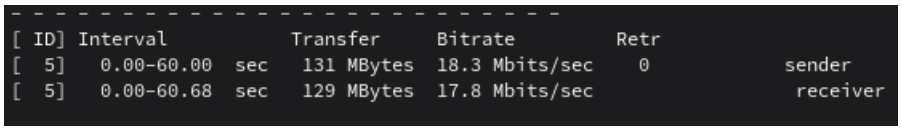

Рисунок 1 – Пример вывода команды iperf –c 10.0.0.2 –t 60 (измерение пропускной способности)

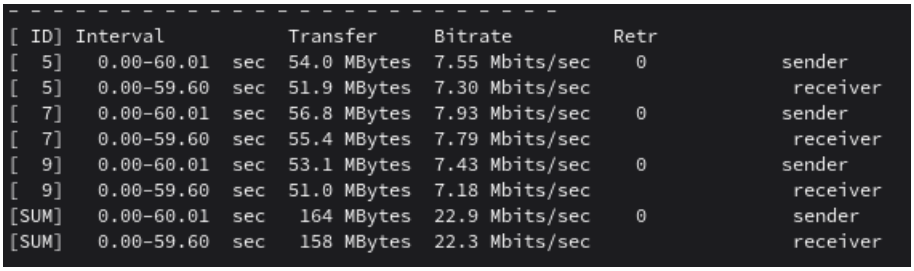

Рисунок 2 – Пример вывода команды iperf –P 3 –c 10.0.0.2 –t 60 (измерение пропускной способности)

Сценарий 2 — D-ITG (ITGsend): Имитация реального трафика с определёнными размерами пакетов и интервалами. Два варианта: малые пакеты (512 байт, 100 пак/сек) для VoIP-подобного трафика и крупные пакеты (1024 байт, 1000 пак/сек) для видео-подобного трафика.

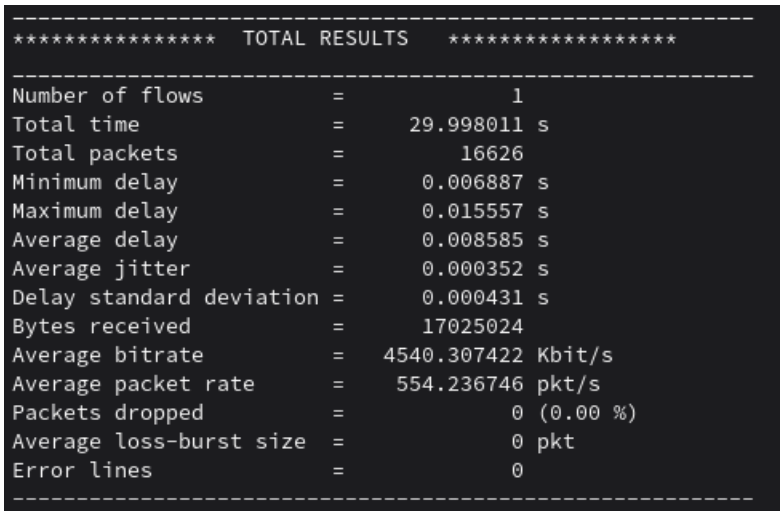

Рисунок 3 – Пример вывода команды /ITGSend -a 192.168.10.20 -T UDP -C 1000 -c 1024 -t 30000 (тестирование соединения в условиях реального трафика)

Сценарий 3 — irtt: Анализ задержек и джиттера для оценки пригодности к real-time приложениям.

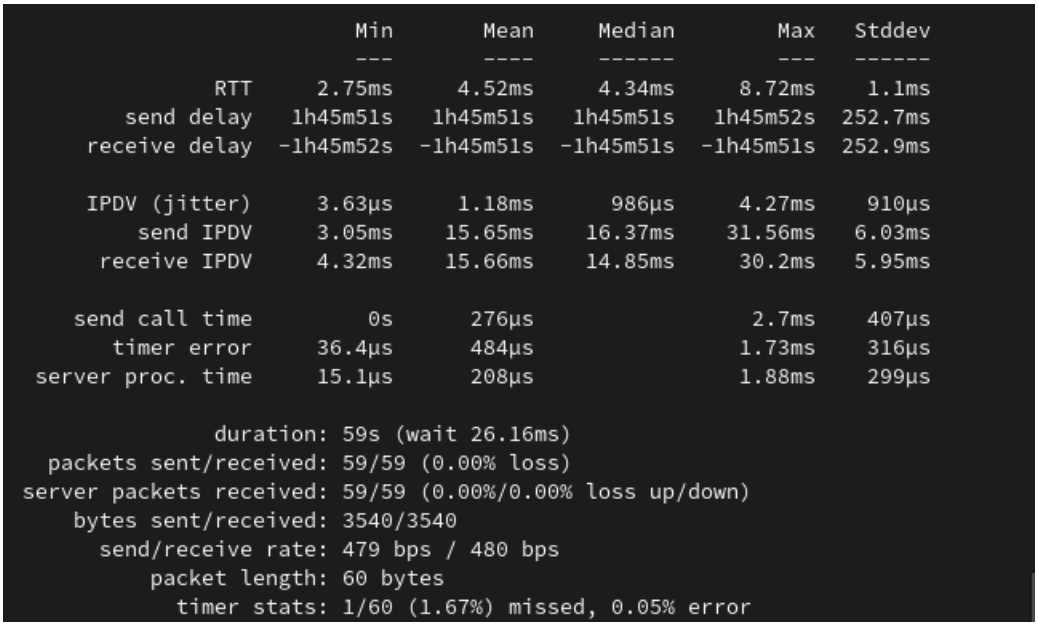

Рисунок 4 – Пример вывода команды ./irtt client 10.0.0.1 (определение задержки и джиттера)

Результаты нагрузочного тестирования:

Пропускная способность (iperf3)

Тест	Native WireGuard	RuWireGuard	Разница
30 сек (1 поток)	21,4 Мбит/с	5,4 Мбит/с	4× медленнее
60 сек (1 поток)	18,0 Мбит/с	6,27 Мбит/с	2,9× медленнее
3 потока (60 сек)	23,1 Мбит/с	10,13 Мбит/с	2,3× медленнее
6 потоков (60 сек)	36,5 Мбит/с	8,8 Мбит/с	4,1× медленнее

Native WireGuard демонстрирует более высокую и стабильную пропускную способность, особенно при увеличении числа параллельных потоков. RuWireGuard показывает значительное снижение производительности: при 6 потоках его пропускная способность падает вместо роста, что указывает на узкие места в обработке.

Объём переданных данных (iperf3)

Тест	Native WireGuard	RuWireGuard	Разница
30 сек (1 поток)	76 МБ	18,5 МБ	4,1× меньше
60 сек (1 поток)	130 МБ	44,6 МБ	2,9× меньше
3 потока (60 сек)	161 МБ	73,1 МБ	2,2× меньше
6 потоков (60 сек)	262 МБ	63 МБ	4,2× меньше

Native WireGuard обрабатывает в 2,2–4,2 раза больше данных в аналогичных условиях. Интересно, что RuWireGuard передал менее данных при 6 потоках (63 МБ) чем при 3 потоках (73,1 МБ), что свидетельствует о деградации производительности при увеличении нагрузки.

Потери пакетов (D-ITG)

Сценарий	Native Wireguard	RuWireGuard
VoIP (100 пак/сек, 512 б)	0%	0%
Видео (1000 пак/сек, 1024 б)	0%	6,19 %

Потери пакетов в обоих тестов не наблюдается для Native Wireguard. Однако RuWireGuard теряет 6% пакетов, что приводит к переотправки пакетов и как следствие увеличивает общую задержку.

Использование ресурсов (D-ITG)

Первый тест:

Метрика	Native Wireguard	RuWireGuard

CPU user	5%	15%
CPU system	22%	45%
CPU idle	77%	39%
RAM free	486904 Кбайт	1826680 Кбайт
RAM buff	55936 Кбайт	36516 Кбайт
RAM cache	2026248 Кбайт	1168788 Кбайт

Второй тест:

Метрика	Native Wireguard	RuWireGuard
CPU user	4%	7%
CPU system	21%	21%
CPU idle	78%	71%
RAM free	479100 Кбайт	1610972 Кбайт
RAM buff	56164 Кбайт	37608 Кбайт
RAM cache	2026532 Кбайт	1399352 Кбайт

В обоих тестах загруженность CPU у RuWireGuard выше, особенно в первом тесте почти вдвое больше. Однако RuWireGuard использует меньше RAM, что свидетельствует о том, что Native Wireguard использует память под кэширование.

Задержка и джиттер (irtt)

Параметр	Native WireGuard	RuWireGuard
Min RTT	2,75 мс	2,57 мс
Mean RTT	4,52 мс	6,02 мс
Max RTT	8,72 мс	9,43 мс
Jitter (StdDev)	1,1 мс	1,24 мс
Jitter (Min)	3,63 µс	61,7 µс
Jitter (Mean)	1,18 мс	1,57 мс
Jitter (Max)	4,27 мс	1,26 мс

Задержки обоих решений находятся в приемлемом диапазоне для большинства приложений (2–9 мс). Однако Native WireGuard имеет более низкую среднюю задержку (4,52 мс против 6,02 мс). Джиттер RuWireGuard немного выше, что может быть критично для реального видео и VoIP.

Сводная сравнительная таблица:

Показатель	Native WireGuard	RuWireGuard	Разница
Максимальная пропускная способность	36,5 Мбит/с	10,13 Мбит/с	3,6× медленнее
Средняя пропускная способность	24,5 Мбит/с	7,6 Мбит/с	3,2× медленнее
Средняя задержка (RTT)	4,52 мс	6,02 мс	на 33% выше
Максимальная задержка	8,72 мс	9,43 мс	на 8% выше
Джиттер (StdDev)	1,1 мс	1,24 мс	на 13% выше
Стабильность под нагрузкой	растёт с потоками	деградирует	Native лучше

Анализ причин различий и перспективы развития

WireGuard благодаря интеграции в ядро Linux исключает контекстные переключения, обеспечивая минимальные накладные расходы и эффективное использование ресурсов. Данные тестирования подтверждают эффективность этого подхода: пропускная способность растёт с увеличением числа потоков (21,4 → 36,5 Мбит/с при переходе с 1 на 6 потоков).

RuWireGuard, реализованная на Go как userspace приложение, требует контекстного переключения для каждого пакета. Результаты тестов показывают, что система не масштабируется линейно: при 6 потоках пропускная способность снижается до 8,8 Мбит/с против 10,13 Мбит/с при 3 потоках. Документация проекта указывает: «Performance: elliptic curve and MGM implementations are extremely slow»

. ГОСТ-операции требуют больше вычислений в текущих реализациях, а userspace архитектура вносит значительные накладные расходы. [1–3, 6, 7, 9]

Направления оптимизации RuWireGuard:

1. SIMD оптимизация — ускорение криптографических функций ГОСТ в 2–4 раза
2. Ассемблерная реализация — для операций с эллиптическими кривыми (ГОСТ Р 34.10-2012)
3. Интеграция в ядро Linux — исключит контекстные переключения, повысит производительность на порядок
4. Переписание критических путей на C/Rust — вместо Go для критичных криптографических операций

Практические рекомендации

WireGuard — оптимальный выбор для задач без требований к отечественной криптографии. Обеспечивает стабильную производительность (20–36 Мбит/с в тестах), приемлемые задержки и масштабируется с числом параллельных потоков. Production-ready статус и простота развёртывания делают его предпочтительным решением для коммерческого использования.

RuWireGuard — необходимое решение для систем, подпадающих под требования российского законодательства. Несмотря на 3,6-кратное снижение производительности по сравнению с WireGuard, обязателен для критической инфраструктуры и государственных ИС. Проект находится в активной разработке и имеет перспективу развития через оптимизацию криптографических функций и переход на интеграцию в ядро.

Практическое применение и тестирование в киберполигоне позволяет подготовить специалистов, компетентных в выборе и развёртывании VPN-инфраструктуры в соответствии с национальными требованиями безопасности

Заключение

Анализ RuWireGuard показал, что проект представляет корректную реализацию WireGuard с заменой криптографических компонентов на ГОСТ-аналоги. Снижение производительности на 3–4 раза обусловлено двумя факторами: особенностями ГОСТ-алгоритмов, требующих больше вычислений, и архитектурой userspace против kernel-mode WireGuard. [1–3, 6, 7, 9, 10]

Результаты нагрузочного тестирования подтверждают практическую применимость обоих решений в разных сценариях. WireGuard обеспечивает высокую производительность и отлично масштабируется, идеален для commercial VPN и высоконагруженных систем. RuWireGuard, несмотря на ограничения производительности (10 Мбит/с при 6 потоках), остаётся пригодным для задач критической инфраструктуры и государственных ИС, где соответствие национальному законодательству о криптографии критично. [1–3, 6, 7, 9]

Выбор между подходами определяется требованиями регуляторной среды и параметрами нагрузки. Дальнейшее развитие RuWireGuard должно ориентироваться на повышение производительности через SIMD оптимизацию, интеграцию в ядро, независимый криптографический аудит и расширение поддержки платформ.

Список использованной литературы

[1] ГОСТ Р 34.10-2012. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. – М.: Стандартинформ, 2012.

[2] ГОСТ Р 34.11-2012. Информационная технология. Криптографическая защита информации. Функция хеширования. – М.: Стандартинформ, 2012.

[3] ГОСТ Р 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры. – М.: Стандартинформ, 2015.

[4] Греков В. С., Уймин А. Г. Стратегии создания модели сети предприятия в рамках киберполигона для эффективной подготовки кадров в области кибербезопасности // Актуальные проблемы комплексной безопасности критически важных объектов топливно-энергетического комплекса: тезисы докладов 78-й Международной молодежной научной конференции. – М., 2025. – С. 17–18.

[5] Исчерпывающее руководство по WireGuard: быстро, безопасно и просто [Электронный ресурс] // XVPN. – Режим доступа: https://xvpn.io/ru/blog/wireguard (дата обращения: 24.11.2025).

[6] Колегов Д. Н., Халниязова Ю. Р. Использование российских криптографических алгоритмов в протоколе безопасности сетевого уровня WireGuard // Вестник ТГУ. Серия «Фундаментальные и прикладные исследования». – 2021. – № 1. – С. 82–96.

[7] Манашев И. Н. Реализация и исследование эталонной версии протокола Ru-WireGuard: дипломная работа. – Томск: Томский государственный университет, 2021. – 27 с.

[8] Организация каналов между офисами через WireGuard VPN на платформе Linux [Электронный ресурс] // Interface31.ru. – Режим доступа: https://interface31.ru/tech_it/2022/03/organizaciya-kanalov-mezhdu-ofisami-cherez-wireguard-vpn-na-platforme-linux.html (дата обращения: 24.11.2025).

[9] BI.ZONE. Ru-WireGuard reference implementation: WireGuard protocol with GOST crypto algorithms [Электронный ресурс]. – Режим доступа: https://github.com/bi-zone/ruwireguard-go (дата обращения: 24.11.2025).

[10] Donenfeld J. A. WireGuard: Next Generation Kernel Network Tunnel // Proceedings of the Network and Distributed System Security Symposium (NDSS). – 2017.

[11] Ejafa Protocol: A custom INC secure protocol [Электронный ресурс] // arXiv. – 2024. – Режим доступа: https://arxiv.org/pdf/2401.02787.pdf (дата обращения: 24.11.2025)

WireGuard VPN: что это и стоит ли использовать? [Электронный ресурс] // Surfshark Blog. – Режим доступа: https://surfshark.com/ru/blog/wireguard-vpn (дата обращения: 24.11.2025).