НАУЧНЫЙ ЖУРНАЛ

СИСТЕМЫ СИНХРОНИЗАЦИИ, ФОРМИРОВАНИЯ И ОБРАБОТКИ СИГНАЛОВ

№2-2025 год

Учредитель:
ООО «ИД Медиа Паблишер»

Номер подписан в печать 15.04.2025 г.

СОДЕРЖАНИЕ

ПОВЫШЕНИЕ ТОЧНОСТИ СПЕКТРАЛЬНОГО АНАЛИЗА СИГНАЛА ЗВУКОВОГО ВЕЩАНИЯ

Попов Олег Борисович
Московский технический университет связи и информатики, профессор, Москва, Россия,
olegp45@yandex.ru

Чернышева Татьяна Васильевна
Московский технический университет связи и информатики, доцент, Москва, Россия,
krba2012@yandex.ru

Коростелев Кирилл Андреевич
Московский технический университет связи и информатики, аспирант, Москва, Россия,
kirill19990511@yandex.ru

Волчков Дмитрий Александрович
Московский технический университет связи и информатики, аспирант, Москва, Россия
qwerty.load@yandex.ru

Аннотация
Быстрое преобразование Фурье (БПФ) является общепринятым методом для спектрального анализа звуковых сигналов. Тем не менее, его разрешающая способность ограничена, что затрудняет точную оценку спектральных компонентов, расположенных близко друг к другу. Дискретное косинусное преобразование, хотя и обладает вдвое большей разрешающей способностью, имеет существенный недостаток – широкие боковые лепестки. В данной работе предложен метод, базирующийся на модифицированном дискретном косинусном преобразовании, который позволяет решить указанную проблему.

Ключевые слова
Дискретное преобразование Фурье, быстрое преобразование Фурье, дискретное косинусное преобразование, коэффициенты преобразования, пик-фактор, транспонирование спектра.

Введение

Человеческий слух обладает удивительной способностью различать мельчайшие нюансы звукового сигнала. Однако, существующие методы спектрального анализа, основанные на БПФ, не могут полностью повторить эту точность [1-3, 7, 9-13]. Для повышения качества звуковоспроизведения необходимо разработать методы спектрального анализа, которые бы соответствовали возможностям человеческого слуха.

Исследования показали, что для адекватного представления звукового сигнала необходимо обеспечить следующие характеристики спектрального анализа:

- Высокая точность определения частоты: Ошибка определения частоты должна составлять не более 1,5% от значения частоты во всем слышимом диапазоне.
- Высокая точность определения амплитуды: точность определения уровня сигнала должна быть не хуже 0,4 дБ.
- Высокая разрешающая способность: метод должен позволять различать близко расположенные спектральные компоненты с разностью частот не менее 60 Гц.
- Точная оценка фазы: ошибка определения фазы должна быть не более 12 градусов.

Кроме того, для анализа быстро меняющихся звуковых событий требуется высокая временная разрешающая способность, не превышающая 8 мс.

Существующие методы спектрального анализа, такие как быстрое преобразование Фурье, не позволяют достичь указанных характеристик при анализе коротких фрагментов сигнала.

Результаты исследований

Ранее авторами был разработан алгоритм спектрального анализа на основе БПФ, который обеспечивал необходимую точность формирования оценок амплитуды, фазы и частоты за счет анализа набора транспонированных по частоте сигналов звукового вещания (СЗВ) [4, 5]. Основной трудностью является формирование комплексного СЗВ, при умножении которого на комплексную же низкочастотную составляющую позволяет формировать набор транспонированных спектров. Селекция по амплитуде дает возможность выбрать частотный сдвиг наиболее близкий к искомой частоте СЗВ. В этом случае частота амплитуда и фаза

выявляются с точностью близкой к точности слухового анализатора.

Для формирования ортогонального сигнала, т.е. реализации преобразования Гильберта, выбран алгоритм на основе дискретного преобразования Фурье (*ДПФ*):

$$\begin{aligned} &S(k) = DFT\left[s(n)\right] = \sum_{n=0}^{N-1} S(n)\cdot W_N^{n\cdot k};\\ &s(n) = IDFT\left[S(k)\right] = \frac{1}{N}\sum_{k=0}^{N-1} S(k)\cdot W_N^{-n\cdot k};\\ &W_N^{n\cdot k} = \exp\left(-j\cdot\frac{2\pi}{N}\cdot n\cdot k\right);\\ &n = 0..N-1, k = 0..N-1 \end{aligned} \quad (1)$$

где $DFT\left[s(n)\right]$ – оператор ДПФ, и $IDFT\left[S(k)\right]$ – оператор ОДПФ.

Для поворота фазы СЗВ у всех коэффициентов *S(k)* на 90°, должен быть. изменен знак мнимой части $IDFT\left[S(k)\right]$.

Точность формирования ортогонального сигнала в основном определяется оконной функцией, используемой в ходе БПФ преобразования, и точностью самого представления сигнала [6]. Исследования показали, что при реализации преобразования Гильберта для широкополосного СЗВ минимальные ошибки обеспечивают оконные функции с минимальным уровнем боковых лепестков.

Наименьшим уровнем боковых лепестков, обладает окно Наттолла:

$$\begin{aligned} &w(n) = a_0 - a_1\cdot\cos\left(\frac{2\cdot\pi\cdot n}{N-1}\right) + a_2\cdot\cos\left(\frac{4\cdot\pi\cdot n}{N-1}\right) - a_3\cdot\cos\left(\frac{6\cdot\pi\cdot n}{N-1}\right),\\ &a_0 = 0.355768,\ a_1 = 0.487396,\ a_2 = 0.144232,\ a_3 = 0.012604. \end{aligned} \quad (2)$$

Для компенсации неравномерности частотной характеристики, возникающей после применения БПФ при 50% перекрытии окон, используется дополнительное окно 1/W2(n).

Предложенный алгоритм способствует повышению точности в определении амплитуды, фазы и частоты спектральных составляющих СЗВ, однако он не решает проблему улучшения разрешающей способности самого преобразования Фурье. Ширина спектральной полосы, соответствующей каждому коэффициенту преобразования Фурье, определяется выражением:

$$df = \frac{B\cdot f_D}{N}\ , \quad (3)$$

где B – коэффициент расширения окна, зависящий от типа оконной функции, f_D – частота дискретизации, N – длина отсчетов. Частоты, соответствующие значениям $n\Delta f$, называются спектральными линиями (бинами). На отрезке длительностью 8-9 мс достичь разрешающей способности 60 Гц с помощью БПФ невозможно.

Алгоритм комплексного дискретного косинусного преобразования

ДКП широко применяется для компактного представления звуковых сигналов благодаря вдвое большей разрешающей способности по сравнению с БПФ. Однако наличие широких боковых лепестков в спектре, полученном с помощью ДКП, ограничивает его применение для точной оценки спектральных составляющих.

Разработанный авторами алгоритм нацелен на устранение проблемы боковых лепестков путем суммирования исходного спектра с дополнительным спектром, полученным из ортогонального сигнала. При этом, боковые лепестки в этих спектрах характеризуются противоположной фазой. В результате этой операции формируется комплексное КДКП, обеспечивающее более точную спектральную оценку. Алгоритм КДКП включает следующие этапы:

* Дублирование сигнала: Исходный сигнал дублируется для создания его комплексного представления.
* Применение оконной функции: к сигналу применяется оконная функция для получения ортогонального сигнала.
* БПФ: вычисляется БПФ ортогонального сигнала.
* Поворот фаз: фазы коэффициентов БПФ поворачиваются на 90 градусов.

* Обратное БПФ: вычисляется обратное БПФ.
* Компенсация окна: применяется компенсирующая оконная функция.
* ДКП: вычисляется дискретное косинусное преобразование исходного и ортогонального сигналов.
* Обращение частоты: аналогичные операции выполняются для сигнала с инвертированной частотой.
* Компенсация боковых лепестков: боковые лепестки компенсируются путем зеркального сложения результатов анализа исходного и инвертированного сигналов.

Рисунок 1 иллюстрирует оценку спектральной составляющей с помощью разработанного алгоритма.

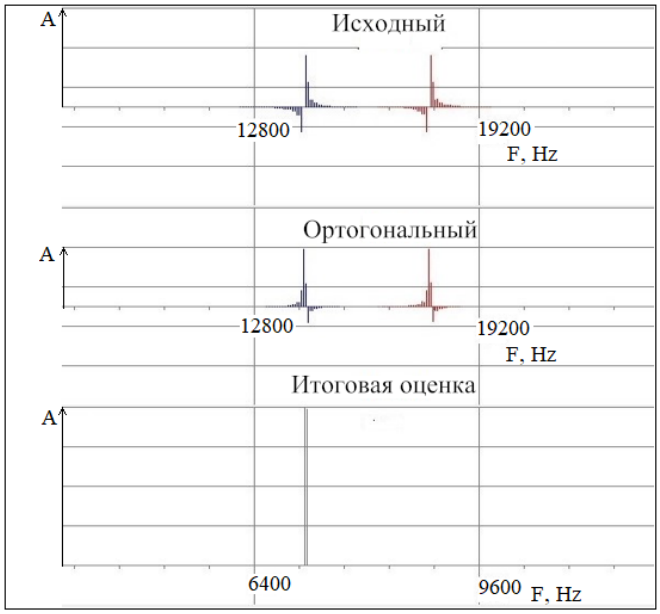

Рис. 1. На верхней шкале FFT исходного сигнала, на средней ортогонального, внизу итоговая оценка, с компенсацией боковых лепестков (А-амплитуда составляющих сигнала)

Перевести на русский

Рисунок 2 демонстрирует процесс компенсации боковых лепестков.

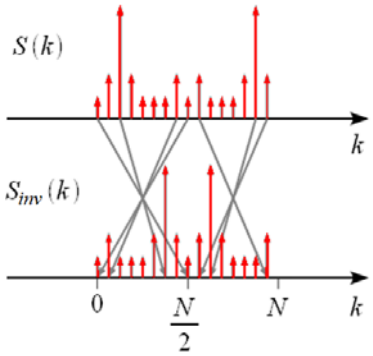

Рис. 2. Инверсия спектра при компенсации боковых лепестков:
k – номер коэффициента, N – длина выборки

Для сохранения полосы исходного сигнала перед применением КДКП производится удвоение частоты дискретизации с использованием метода, не вносящего искажений.

На рисунке 3 представлен пример сложения исходного спектра и спектра компенсирующего сигнала для частоты, соответствующей определенной спектральной линии (бину).

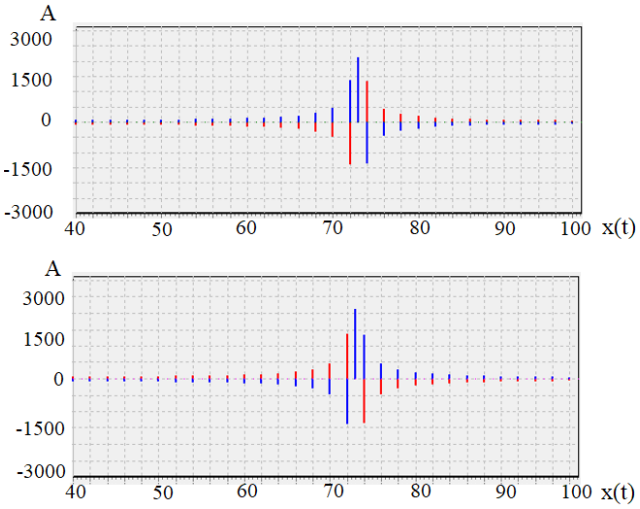

Рис. 3. Коэффициенты оценки КДКП для вещественной и мнимой составляющей (n-амплитуда в шагах квантования, x(k)-номер частотного коэффициента)

После того как спектры КДКП исходного и компенсирующего сигналов складываются, боковые лепестки нивелируются, а значения коэффициентов, которые соответствуют спектральным линиям, суммируются. Амплитудная оценка коэффициента приведена на рисунке 4.

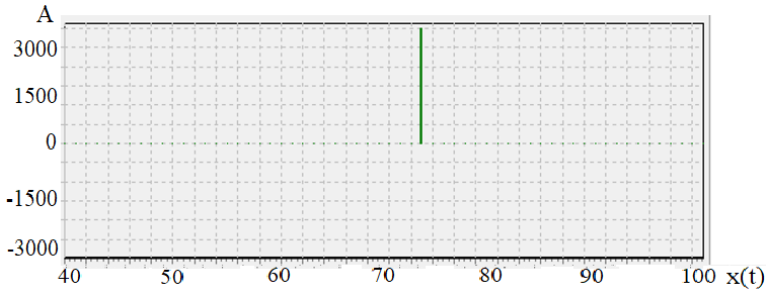

Рис. 4. Амплитудная оценка 73 бина КДКП (n-амплитуда в шагах квантования, x(k)-номер частотного коэффициента)

Согласно оценке погрешности, вычисленной как разница между исходным сигналом и сигналом, восстановленным после обратного КДКП, было установлено, что для частот, не совпадающих со спектральными линиями, ошибка не превышает двух уровней квантования.

Анализ сигнала ошибки, сформированного как разность между исходным и обработанным сигналами, показал, что эта ошибка для частот, соответствующих бинам, не превышает 1 шага квантования (рис. 5) [8, 9].

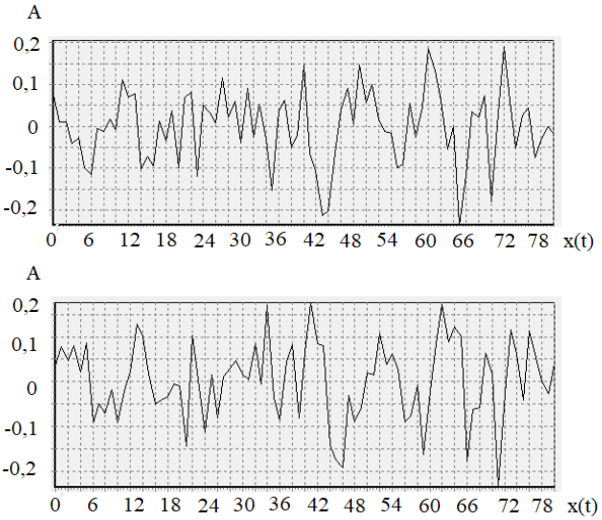

Рис. 5. Ошибка синтеза сигнала после прямого и обратного КДКП, для частоты кратной бину (x(k)-номер отсчета, n-амплитуда в долях шага квантования)

Свойства комплексного дискретного косинусного преобразования

Оценки свойств ДПФ обычно проводятся для наихудшего случая, когда частота сигнала не совпадает с центром спектральной линии (бина). Аналогично, для КДКП мы рассматриваем частоту, не соответствующую центру бина. На рисунке 6 представлена амплитудная оценка КДКП для коэффициента 73.5, которая соответствует такой ситуации.

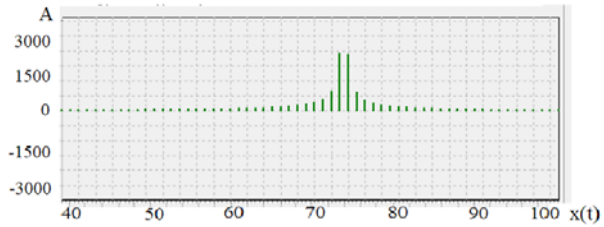

Рис. 6. Амплитудная оценка коэффициента КДКП, расположенного между двумя бинами (n-амплитуда в шагах квантования, x(k)-номер частотного коэффициента)

Эквивалентная шумовая полоса (ЭШП) окна определяется как ширина полосы пропускания идеального фильтра, который пропускает столько же шума, сколько и рассматриваемое окно (рис. 7).

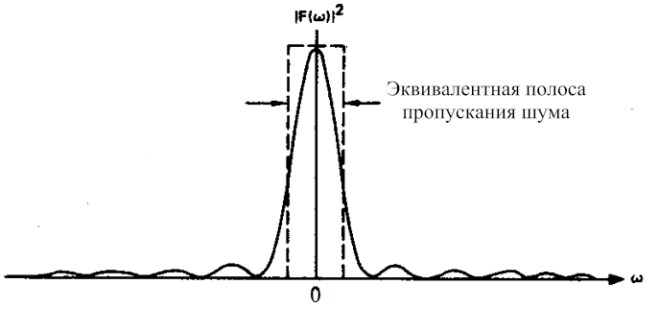

Рис. 7. Эквивалентная шумовая полоса

Нормированная ЭШП окна представляет собой мощность шума на единицу частоты и рассчитывается по формуле

$$\text{ЭШП} = \frac{\sum_n w^2(nT)}{\left[\sum_n w(nT)\right]} \tag{5}$$

где $\text{Макс.Усилиние.Сигнала} = W(0) = \sum_n w(nT)$; $\text{Макс.Усилиние.по.Мощности} = W^2(0) = \left[\sum_n w(nT)\right]^2$.

Для КДКП нормированная ЭШП равна 1 на уровне -3 дБ, 0.4 на уровне -6 дБ. Усиление преобразования характеризует степень усиления сигнала при прохождении через систему. Снижение амплитуды колебаний на краях окна приводит к уменьшению усиления преобразования и, соответственно, к увеличению потерь преобразования.

Если входная последовательность соответствует выражению:

$$f(nT) = A * \exp(+i\omega_k nT) + q(nT), \tag{6}$$

Где $q(nT)$ – последовательность отсчетов белого шума.

Тогда колебание после воздействия оконной функции будет:

$$F(\omega_k)_{\text{сигнал}} = \sum_n w(nT) A * \exp(+i\omega_k nT) \exp(-i\omega_k nT) = A \sum_n w(nT) \tag{7}$$

Некогерентная часть колебания вычисляется как:

$$F(\omega_k)\,|_{\_\text{шум}} = \sum_n w(nT) q(nT) \exp(-i\omega_k nT) \tag{8}$$

А некогерентная мощность:

$$E\{|F(\omega_k)|_{\_шум\_}|^2\} = \sum_n \sum_m w(nT)w(mT)E\{q(nT)q(mT)\}\exp(-i\omega_k nT)\exp(+i\omega_k mT) = \sigma_q^2 \sum_n w^2(nT), \quad (9)$$

где $E\{\ldots\}$ – оператор математического ожидания.

УП вычисляется как:

$$\frac{S_o / N_o}{S_i / N_i} = \frac{A^2[\sum_n w(nT)] / \sigma_q^2 \sum_n w^2(nT)]}{A^2 / \sigma_q^2} * \frac{[\sum_n w(nT)]^2}{\sum_n w^2(nT)} \quad (10)$$

Корреляция между соседними отрезками сигнала, обрабатываемыми ДПФ, зависит от длины отрезка, частоты дискретизации и типа оконной функции. Паразитная амплитудная модуляция возникает из-за того, что амплитуда спектральных составляющих, расположенных между бинами, зависит от их положения относительно центра на 0,5 бина.

$$F(\omega_{(1/2)}|_{\_сигнал} = A\sum_n w(nT)\exp(-i\omega_{(1/2)}nT)$$

$$\omega_{(1/2)} = \frac{1}{2}\frac{\omega_s}{N} = \frac{\pi}{NT} \quad (12)$$

Тогда паразитная АМ, при сдвиге на 0,5 бина (максимальная) будет:

$$\text{ПаразитнаяАМ} = \frac{|\sum_n w(nT)/\exp(-i\frac{\pi}{N}n|}{\sum_n w(nT)} = \frac{|\sum_n W(\frac{1}{2}\frac{\omega_s}{N})|}{W(0)} \quad (13)$$

Величина потерь при преобразовании для рассматриваемых частот напрямую зависит от формы используемой оконной функции и способна оказать значительное воздействие на точность измерения слабых сигналов, особенно в условиях шумовых помех. В таблице 1 представлено сравнительное исследование параметров оценки для прямоугольного окна и окна Хэмминга, применяемых в контексте ДПФ и КДКП.

Таблица 1

Параметры оценок для прямоугольного окна, окна Хэмминга для ДПФ и КДКП

МУБЛ, дБ	КУ	ЭШП, бин	ЭШП -3, бин	ЭШП -6, бин	ПАМ, дБ	КПУ, %
-13	1,0	1,00	0.89	1,21	3,92	50
-43	0,5	1,36	1,30	1,81	1,78	23,5
-10	1,0	1,0	0,4	0,6	3,92	50

МУБЛ – максимальный. уровень боковых лепестков, КУ – когерентное усиление, ЭШП – эквивалентна шумовая полоса по уровню 0,-3,-6 дБ, ПАМ – паразитная амплитудная модуляция, КПУ – корреляция перекрывающихся участков

Минимальная разрешаемая полоса частот (МРПЧ). При анализе двух близко расположенных спектральных составляющих происходит наложение их оценок на соседних спектральных линиях. Чтобы уверенно различить эти составляющие, необходимо, чтобы расстояние между их максимумами превышало ширину спектральной линии на уровне -6 дБ (рис. 8).

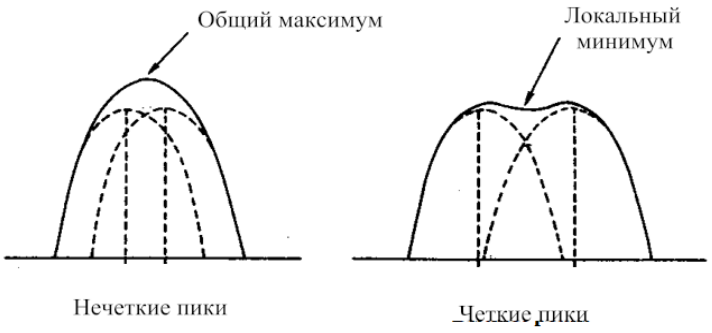

Рис. 8. Спектральное разрешение для двух близко расположенных бинов

Перевести на русский

Для КДКП минимальная разрешаемая полоса частот составляет около 5%. Учитывая, что разрешающая способность КДКП вдвое выше, чем у дискретного преобразования Фурье (ДПФ), такой показатель достаточен для большинства практических задач.

$$\Delta f = \beta(\frac{f_s}{N})$$

Коэффициент корреляции определяется как:

$$c(r) = \frac{\{\sum_{n=0}^{rN-1}(W(n)W(n+[1-r]N))\}}{\{\sum_{n=0}^{N-1}(W^2(n)\}} \quad (11)$$

С целью определения степени воздействия смежных спектральных линий друг на друга, был создан испытательный сигнал, включающий в себя три синусоидальные компоненты с аналогичными амплитудами и отличающимися начальными фазами. Продолжительность данного сигнала была установлена в 256 отсчетов. Визуализация амплитудного спектра полученного сигнала демонстрируется на рисунке 9

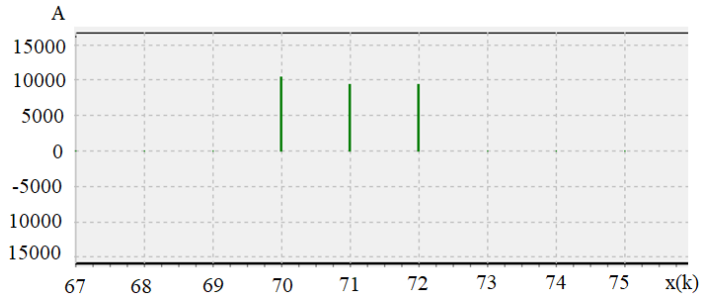

Рис. 9. Амплитудный спектр трех спектральных составляющих, кратных бинам (n-амплитуда в шагах квантования, x(k)-номер частотного коэффициента)

Анализ показал, что при использовании КДКП спектральные составляющие не оказывают существенного влияния друг на друга. Для сравнения, на рисунке 10 приведен амплитудный спектр того же сигнала, полученный с помощью ДПФ. Видно, что при длине сигнала 256 отсчетов ДПФ не позволяет точно определить амплитуды и фазы спектральных составляющих. Для более точного анализа требуется увеличить длину сигнала до примерно 1024 отсчетов.

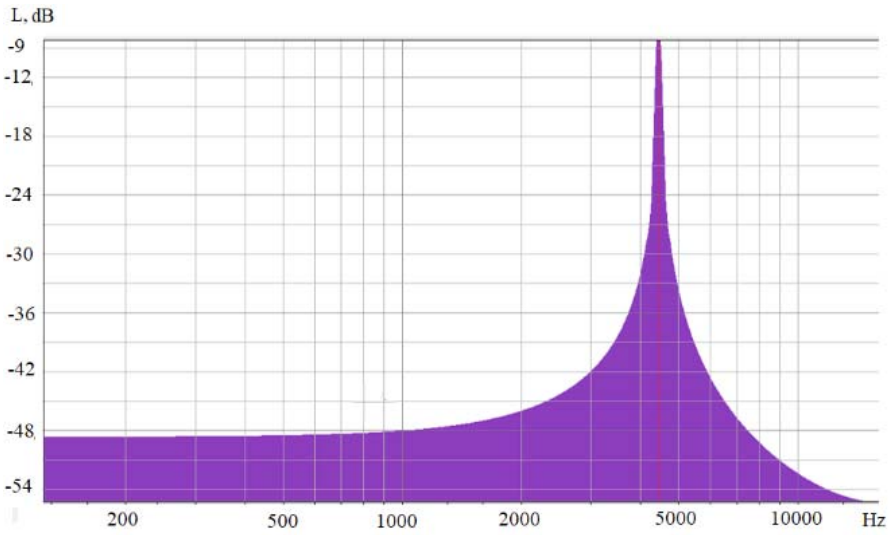

Рис. 10. Амплитудный спектр трехчастотного сигнала, сформированный с помощью ДПФ (L-уровни сигнала)

В современных системах передачи данных часто используются сигналы, спектральные компоненты которых точно соответствуют частотам дискретизации (бинам) ортогонального преобразования. Однако при модуляции таких сигналов, необходимой для передачи информации, возникает проблема боковых лепестков, приводящая к взаимному влиянию спектральных составляющих.

Для оценки степени влияния модуляции на спектр сигнала были проведены исследования с использованием частотной, амплитудной и фазовой модуляций. Спектры полученных сигналов анализировались с помощью КДКП и ДПФ.

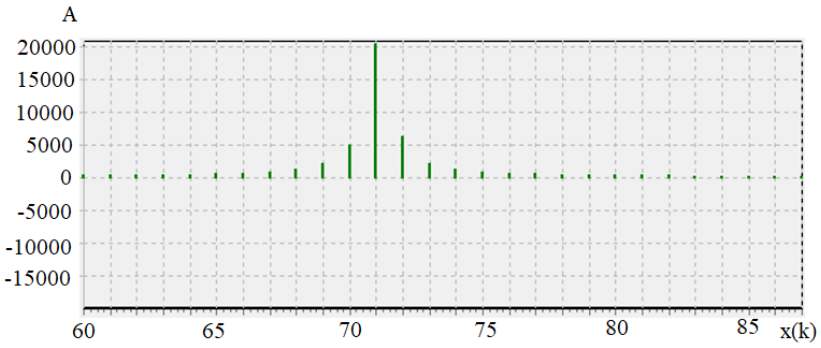

Рис. 11. Амплитудная характеристика частотно модулированного сигнала, сформированного с помощью КДКП (n-амплитуда в шагах квантования, x(k)-номер частотного коэффициента)

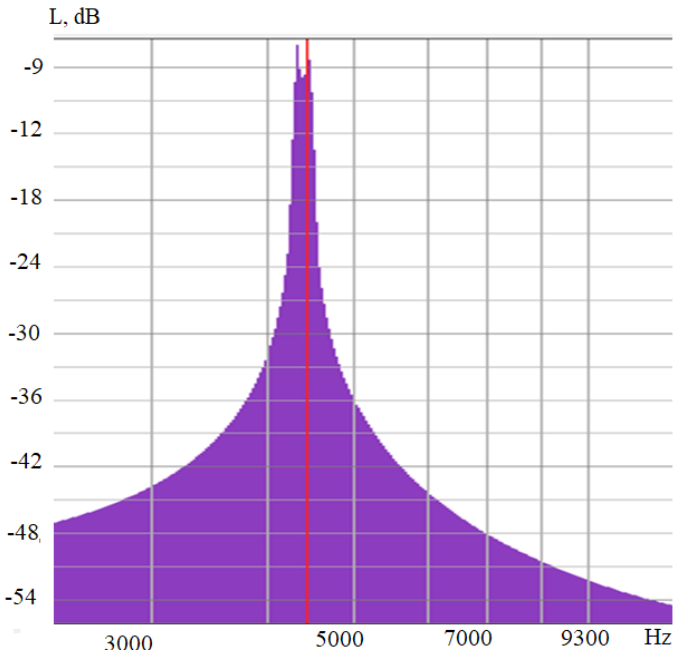

Рис. 12. Амплитудная характеристика частотно модулированного сигнала, сформированного с помощью ДПФ (L-уровни сигнала)

Частотная модуляция. Сигнал с несущей частотой 4372 Гц был модулирован синусоидальным сигналом с частотой 9 Гц и девиацией частоты 62,5 Гц. Результаты спектрального анализа представлены на рисунках 11 (КДКП) и 12 (ДПФ). Видно, что уровень боковых лепестков для КДКП значительно ниже, чем для ДПФ (-4 дБ против -1 дБ).

Амплитудная модуляция. В качестве сигнала с амплитудной модуляцией был использован звуковой сигнал типа тремоло. Осциллограмма этого сигнала приведена на рисунке 13. Результаты спектрального анализа представлены на рисунке 14. Для сравнения с КДКП использовалось ДПФ с прямоугольным окном и длительностью выборки 256 отсчетов.

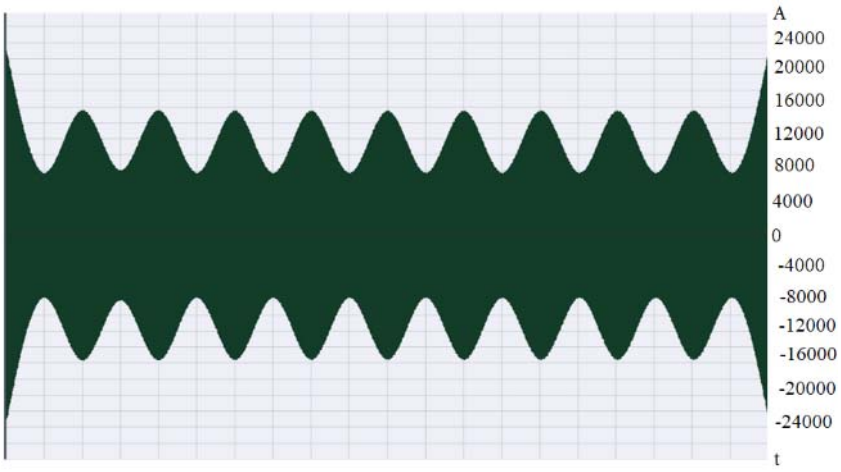

Рис. 13. Осциллограмма амплитудно- модулированного сигнала (тремоло) (n-амплитуда в шагах квантования)

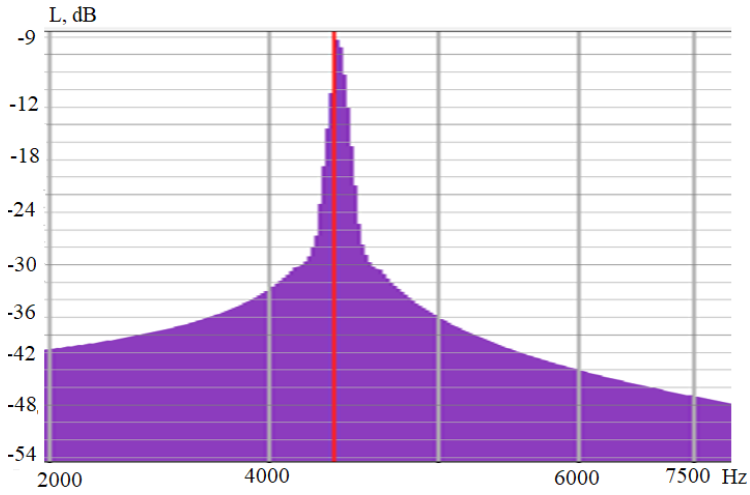

Рис. 14. Амплитудный спектр амплитудно- модулированного сигнала сформированного с использованием ДПФ (тремоло) (L-уровни сигнала)

Рисунок 15 наглядно демонстрирует результаты оценки амплитудного спектра сигнала, подвергнутого амплитудной модуляции и обработанного с использованием КДКП. При этом, для анализа была использована выборка, состоящая из 256 точек.

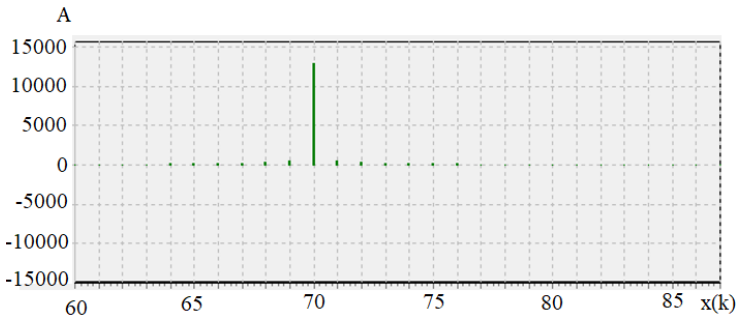

Рис. 15. Амплитудный спектр амплитудно модулированного сигнала сформированного с использованием КДКП (тремоло) (n-амплитуда в шагах квантования, x(k)-номер частотного коэффициента)

Фазовая модуляция. В качестве примера фазовой модуляции был рассмотрен сигнал с фазовой манипуляцией на 180 градусов. Осциллограмма такого сигнала представлена на рисунке 16. Результаты спектрального анализа этого сигнала с помощью ДПФ и КДКП приведены на рисунках 17 и 18 соответственно. Для обоих преобразований использовалась выборка длительностью 256 отсчетов и прямоугольное окно.

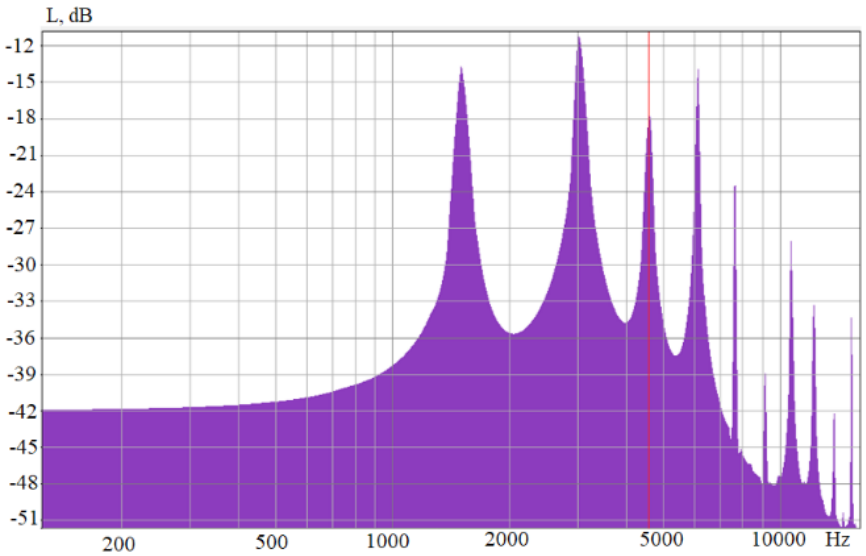

Рис. 17. Амплитудный спектр фазоманипулированного сигнала сформированного с использованием ДПФ (L-уровни сигнала)

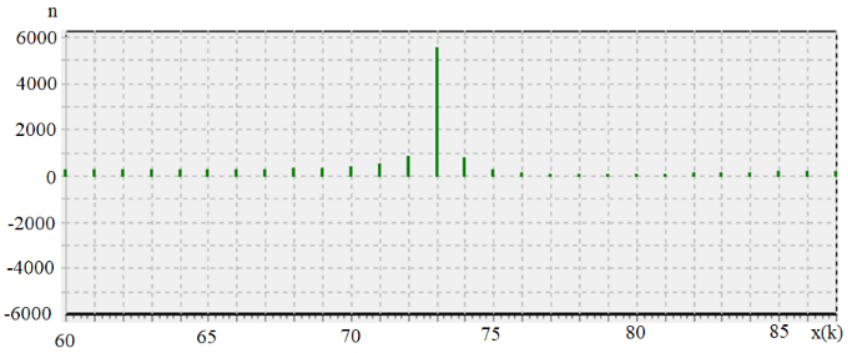

Рис. 18. Амплитудный спектр фазоманипулированного сигнала сформированного с использованием КДКП (n-амплитуда в шагах квантования, x(k)-номер частотного коэффициента)

Максимальный уровень боковых лепестков для КДКП составил -40 дБ, в то время как для ДПФ этот уровень был значительно выше и составил -3 дБ.

Результаты исследований. Проведенные исследования показали, что при анализе спектров модулированных сигналов комплексное дискретное КДКП обеспечивает более точные результаты по сравнению с ДПФ. Кроме того, КДКП обладает вдвое большей разрешающей способностью.

Заключение

Анализ требований, предъявляемых к точности спектрального анализа для адекватного восприятия звуковых образов человеческим слуховым аппаратом, выявил, что традиционные подходы не всегда соответствуют этим требованиям.

В связи с этим, авторами была предложена инновационная методика спектрального анализа, основанная на БПФ и использующая комплекс частотно-транспонированных сигналов. Этот методический подход позволяет оценивать амплитуду, фазу и частоту спектральных составляющих с высокой точностью, которая сопоставима к возможностям человеческого слуха. Однако, необходимо подчеркнуть, что БПФ не обеспечивает в полной мере достаточной разрешающей способности для детального анализа.

Для решения проблемы недостаточной разрешающей способности был разработан новый алгоритм спектрального анализа, базирующийся на комплексном КДКП. Этот метод сочетает в себе высокую разрешающую способность и комплексное представление спектральных коэффициентов, что позволяет проводить более точный анализ звуковых сигналов.

Исследование ключевых свойств КДКП подтвердило его эффективность для анализа звуковых сигналов. Было установлено, что характеристики коэффициентов КДКП сопоставимы с характеристиками ДПФ, но при этом КДКП обеспечивает вдвое большую разрешающую способность. Кроме того, КДКП демонстрирует улучшенную избирательность по частоте.

Анализ спектров модулированных сигналов наглядно показал, что КДКП позволяет более эффективно концентрировать энергию сигнала в меньшем количестве спектральных коэффициентов. Высокая разрешающая способность и точность КДКП делают его многообещающим инструментом для прецизионного спектрального анализа звуковых сигналов, а также для разработки перспективных систем передачи информации с частотным мультиплексированием каналов.

Литература

1. *Абрамов В.А., Попов О.Б., Чернышева Т.В., Борисов А.А.* Патент 2808156 РФ Способ и устройство высокоточного измерения спектра информационных акустических сигналов; заяв.10.03.2023; опубл. 24.11.2023.

2. *Абрамов В.А., Попов О.Б.* Свидетельство о государственной регистрации программы для ЭВМ № 2022 2022616423, Россия. Программа для высокоточного спектрального анализа звукового сигнала. Программное обеспечение "ДКП-Спектр": опуб. 19.04.2022.

3. *Абрамов В.А., Попов О.Б., Чернышева Т.В., Борисов А.А.* Патент 2813684 C1 РФ Способ и устройство измерения спектра и кепстральных параметров информационных акустических сигналов телерадиовещания; заяв.13.07.2023, опубл.15.02.2024.

4. *Абрамов В.А., Попов О.Б., Власюк И.В., Балобанов А.В.* Патент 2756934 C1 РФ, Способ и устройство измерения спектра информационных акустических сигналов с компенсацией искажений; заяв.17.11.2020; опубл.07.10.2021.

5. *Попов О.Б., Рихтер С.Г., Терехов А.Н.* и др. Компандирование сигналов в канале звукового вещания. Учебное пособие для вузов; под ред. С. Г. Рихтера. М.: Горячая линия – Телеком, 2021. 298 с.

6. *Ковалгин Ю.А*. Цифровое радиовещание: системы и технологии. М.: Горячая линия – Телеком. 2021, 580 с.

7. *Záviška P., Rajmic P., Ozerov A., Rencker L.* A Survey and an Extensive Evaluation of Popular Audio Declipping Methods // IEEE Journal of Selected Topics in Signal Processing, vol. 15, no. 1, pp. 5-24, Jan. 2021, doi: 10.1109/JSTSP.2020.3042071.

8. *Schlecht S.J., Fierro L., Välimäki V., Backman J.* Audio Peak Reduction Using a Synced Allpass Filter // ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, Singapore, 2022, pp. 1006-1010, doi: 10.1109/ICASSP43922.2022.9747877.

9. *Ming B., Wu P.* Research on Audio Signal Denoising and Simulation Processing // 2019 International Conference on Communications, Information System and Computer Engineering (CISCE), Haikou, China, 2019, pp. 192-194, doi: 10.1109/CISCE.2019.00051.

10. Recommendation ITU-R BS. I 284-2 (01/2019). General methods for the subjective assessment of sound quality. BS Series. Broadcasting service (sound).

11. *Bai T., Xie L., Li Z., Yang J., Chen Z., Wan P.* A High-Precision Audio Z-Δ D/A Converter // 2020 IEEE 14th International Conference on Anti-counterfeiting, Security, and Identification (ASID), Xiamen, China, 2020, pp. 120-123, doi: 10.1109/ASID50160.2020.9271769.

12. *Попов О.Б., Чернышева Т.В., Сапронов П.С., Коростелев К.А*. Минимизация искажений при аналого-цифровом преобразовании в студии // DSPA: Вопросы применения цифровой обработки сигналов. 2024. Т. 14. № 1. С. 35-42.

13. *Абрамов В.А., Попов О.Б., Чернышева Т.В., Кузнецов П.* Алгоритм комплексного дискретного косинусного преобразования // DSPA: Вопросы применения цифровой обработки сигналов. 2022. Т. 12. № 2. С. 4-12.

ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ СЕТИ ДОСТУПА МУЛЬТИСЕРВИСНОЙ СЕТИ СВЯЗИ

Короткова Виктория Игоревна,
МТУСИ, ассистент, Москва, Россия,
v.i.korotkova@yandex.ru

Новодерёжкин Константин Юрьевич,
МТУСИ, доцент, к.воен.н., Москва, Россия,
k.novoderezhkin@agatrt.ru

Пшеничников Анатолий Павлович,
МТУСИ, профессор, к.т.н., Москва, Россия,
psheni4nikov@yandex.ru

Аннотация
Рассмотрены основные функции сети доступа: пакетизация сообщений голосовых, передачи данных видеоконференцсвязи; способы организации виртуальных сетей связи VLAN; организация очередей; размещение пакетов в очередях в порядке убывания приоритетов в обслуживании; циклическое обслуживание очередей по строгому приоритету. Приведены результаты расчётов основных параметров сети доступа, необходимых для оценки среднего числа пакетов и величины задержки пакетов в очередях с помощью теоремы Дж. Литтла.

Ключевые слова
Сеть доступа, качество обслуживания, пакетизация сообщений, интенсивность пакетов и нагрузки, кодеки, классы приоритетов, система поллинга.

Введение

Почти четверть века системы телекоммуникаций в соответствии с концепцией МСЭ-Т (рекомендации серии *Y*. 2000) «Сети следующего поколения» (*NGN - Next Generation Networks)* [1] успешно переходят с технологии коммутации каналов на технологию коммутации пакетов. За первое десятилетие XX1-го века основные задачи перехода на технологию коммутации пакетов были решены и в 2011 году МСЭ-Т в развитие *NGN* принял рекомендации серии *Y*.3000 по концепции Будущих сетей *FN* (*Future Networks*) [2].

Концепция *FN* ориентирована на применение технологии виртуализации сетевых функций *NFV* (*Network Functions Virtualization*), программно-конфигурируемых сетей *SDN* (*Software Defined Networking),* искусственного интеллекта *AI (Artificial Intelligence*), методов работы с большими данными (*Big Data*), облачных вычислений (*Claude Computing*), Интернета вещей *IoT (Internet of Things)* и других современных технологий.

Для исследования путей преодоления ограничений существующих фиксированных сетей связи, МСЭ-Т в 2018 г. создал фокус-группу технологий *Network* 2030 (*FG NET*-2030) для определения путей создания фиксированных сетей связи на период до 2030 года и в дальнейшей перспективе [3].

Фокус-*группа FG NET*-2030 определила три базовых принципа реализации Сети 2030:

1) Новые сетевые услуги BBE&HPC *(Beyond Best Effort and High-Precision Communications* – качество выше, чем у услуг с негарантированной доставкой *BE (Best Effort)* и высокоточные коммуникации);

2) Новые медиа VLV&TIC (*Very Large Volume & Tiny Instant Communications* - очень большой объем передаваемых данных и крошечные мгновенные сообщения), которые включают следующие медиа:

- расширенная дополненная/виртуальная реальности;
- коммуникации голографического типа;
- коммуникации голографического типа;
- очень высокая полоса пропускания (> Тбит/с);
- голографическая телепортация (задержки < 5мс);
- цифровые сенсоры;
- качественные коммуникации;
- координированная передача потоков сообщений.

3) Новая сетевая архитектура ManyNets *(Many Networks* – множество сетей), которая включает следующие сети и сетевые технологии: спутниковые сети; масштабируемый Интернет; частные сети; граничные вычисления с множественным доступом; сети специального назначения; сверхплотные сети; новые интерфейсы «сеть-сеть», «Оператор-Оператор».

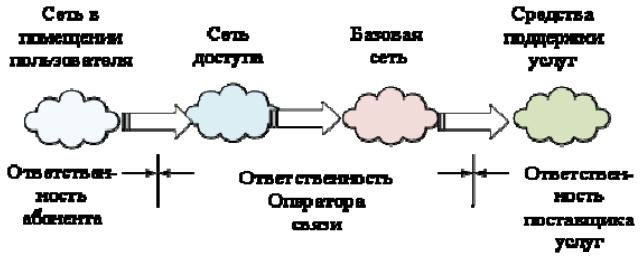

Рис. 1. Рекомендуемая МСЭ-Т модель сети электросвязи

МСЭ-Т рекомендует представлять сеть электросвязи в виде четырёх следующих фрагментов (рис. 1).

Объектом исследования в данной работе является сеть доступа, однако на её функционирование оказывает существенное влияние функциональные возможности конечных устройств пользователей, а также отдельные сегменты транзитной (базовой) сети.

Использование в сети подсистемы передачи мультимедийных сообщений на основе протокола *IP* (*IMS – IP Multimedia Subsystem*) позволило в сети доступа обслуживать сообщения как проводного, так и беспроводного доступа.

В Сети 2030 сеть доступа будет включать:

– сеть фиксированного доступа;

– сеть радиодоступа;

– базовые станции и соответствующие контроллеры сегментов транзитной сети.

В соответствии с Федеральным законом Российской Федерации «О защите прав потребителей» [4] все предоставляемые пользователям услуги телекоммуникационных сетей должны иметь подробную спецификацию, включающую количественную оценку показателей качества.

Эффективность доставки информации через телекоммуникационную сеть измеряется показателем эффективности работы сети (*NP - Network Performance*). В сетях с коммутацией пакетов требования к сетевым показателям качества обслуживания *QoS (Quality of Service)* приведены в рекомендации МСЭ-Т *Y.*1541 [5].

Обслуживание потоков сообщений в сети доступа представляет собой систему циклического опроса очередей, или систему поллинга. На рисунке 2 приведена простейшая модель такой системы.

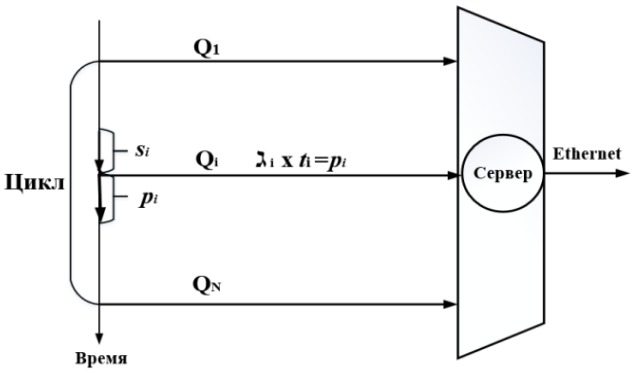

Рис. 2. Простейшая модель системы поллинга

Обозначения в модели системы поллинга:

Q_i – обозначение *i*-ой очереди, в которой: λ_i – параметр потока заявок;

t_i – средняя длительность обслуживания одной заявки;

s_i – среднее время подключения сервера к *i*-ой очереди;

ρ_i – среднее время обслуживания сервером заявок в *i*-ой очереди;

$S = \sum_{i=1}^{N} s_i$ – суммарное время переключения сервера за один цикл;

$\rho = \sum_{i=1}^{N} \rho_i$ – суммарная загрузка системы поллинга.

Система имеет один сервер и $N \geq 2$ очередей. Число мест для ожидания в очереди не ограничено. Среднее время цикла C складывается из времени обслуживания всех очередей (доля времени ρ от C) и времени переключения между очередями S за цикл, то есть $C = \rho C + S$, откуда время цикла $C = S/(1-\rho)$.

Некоторые перспективные телекоммуникационные технологии предъявляют очень жёсткие требования к допустимым задержкам при передаче сообщений. Поэтому анализ способов сокращения задержек в очередях сети является весьма актуальной задачей.

Основной набор функций сети доступа

Рассмотрим простейший численный пример сети доступа, обслуживающей три потока сообщений: голосовые сообщения (ГС), видеоконференцсвязи (ВКС), передачи данных (ПД). Пусть пакеты сообщений в очередях обслуживает сервер фирмы Eltex производительностью $V_{\text{сер}}$. = 12,8 Гбит/с (12,8 $\cdot 10^9$ бит/с).

Пакетизация голосовых сообщений. Линии и терминальное оборудование, не работающие с пакетными технологиями, подключаются к сети доступа с помощью медиашлюзов. Преобразование информации в пакетный вид реализуется с помощью кодеков. Выходы шлюзов и мультимедийные терминалы включаются в пакетный коммутатор Ethernet, на выходе которого используются интерфейсы 1 Гбит/с или 10 Гбит/с.

При использовании голосового кодека G.711 размер речевого пакета с учётом адресной информации составляет $d_{\text{гол. пак}}$ = 218 байт = 1,744⋅103 бит. Длительность одного голосового пакета $t_{\text{гол. пак.}}$ = 20 мс = $20 \cdot 10^{-3}$ с. Средняя длительность одного голосового вызова [6] $t_{ГС}$ = 72 с. Отсюда среднее число голосовых пакетов в одном голосовом сообщении (при обслуживании одного вызова):

$n_{1выз.} = t_{ГС} / t_{гол\ пак.} = 72 / 20 \cdot 10 = 3,6 \cdot 10^3$ пакетов. Среднее число вызовов в час наибольшей нагрузки (ЧНН) от 1 абонента [6] c_1=3. Пусть число источников ГС равно $m_{\text{ист. ГС}}$ =10⋅10^3. Среднее число вызовов в ЧНН от всех источников будет равно:

$$S = \sum_{i=1}^{N} s_i$$

От всех пользователей ГС в ЧНН интенсивность голосовых пакетов составит:

$\lambda_{гол.} = n_{1выз.} \cdot c_{выз.ЧНН} = 3,6 \cdot 10^3 \cdot 30 \cdot 10^3 = 108 \cdot 10^6$.

Объём информации от одного голосового вызова составит:

$d_{1выз} = d_{гол.пак} \cdot n_{1выз} = 1,744 \cdot 10^3 \cdot 3,6 \cdot 10^3 = 6,278 \cdot 10^6$.

От всех пользователей ГС объём информации составит:

$d_{сум} = c_{выз.ЧНН} \cdot d_{1выз} = 30 \cdot 10^3 \cdot 6,278 \cdot 10^6 = 188,35 \cdot 10^9$ бит.

Сервер за 1 секунду обслуживает $V_{сер.}$ = 12,8 $\cdot 10^9$ бит. Для обслуживания заявок от всех пользователей голосового трафика серверу потребуется время

$t_{гол.} = d_{сум} / V_{сер} = 188,35 \cdot 10^9 / 12,8 \cdot 10^9 = 14,72$ с.

Интенсивность нагрузки на обслуживание голосовых сообщений сервером в ЧНН ($t_{\text{ЧНН}}$ = 3600 с)

$\rho_{ГС} = t_{гол} / t_{чнн} = 14,72 / 3600 = 0,00409$ Эрл.

Пакетизация сообщений видеоконференцсвязи.

Видеоконференция – технология, обеспечивающая одновременную двустороннюю передачу, обработку, преобразование и представление видеоинформации на расстоянии в режиме реального времени.

Кодеки для сжатия данных [7]:

– кодеки для сжатия текста: *GZIP*, *Bzip*2, *LZW;*

– кодеки для сжатия изображений: *JPEG*, *PNG*, *GIF;*

– кодеки для сжатия аудио: *MP*3, *AAC*, *Vorbis*, *Opus* -

– кодеки для сжатия видео: *H*.264/*AVC*, *H*.265/*HEVC*, *VP*8, *VP*9. Стандарт сжатия видео кодеками *H*.264 и *H*.265 обеспечивает запись видеопотока по каналам *IP* на скорости 8 Мбит/с.

Существует несколько *степеней сжатия кодека*, которые выбираются в зависимости от требований к качеству и скорости передачи данных:

– без сжатия (0%) - изображение передаётся без изменений, но занимает много места;

– низкая степень сжатия (10–20%) - изображение сжимается незначительно, сохраняя хорошее качество;

– средняя степень сжатия (30–50%) - изображение сжимается сильно, но качество может ухудшиться;

– высокая степень сжатия (60-90%) – изображение сжимается максимально, но качество может быть очень низким.

Рассмотрим ВКС улучшенной чёткости [8] с разрешением 1280x720p 50.

Число пикселей в одном кадре составляет:

$v_{пикс.} = 1280 \cdot 720 = 921600$.

Пусть глубина цвета на 1 пиксель равна $s_{1пикс} = 16$ бит. Информационная нагрузка одного кадра ВКС составит $d_{1кадр} = v_{пикс.} \cdot s_{1пикс} = 921600 \cdot 16 = 14,7456 \cdot 10^6$ бит.

При частоте f=50 кадров в секунду информационная нагрузка составит:

$V_{ВКС} = d_{1кадр} \cdot f = 14,75 \cdot 10^6 \cdot 50 = 737,5 \cdot 10^6$ бит/с.

Пусть длительность одной ВКС равна $t_{ВКС}$ = 1 200с. Объём информации при передаче в ЧНН одной ВКС составит

$d_{1\,ВКС} = V_{ВКС} \cdot t_{ВКС} = 737,5 \cdot 10^6 \cdot 1,2 \cdot 10^3 = 885 \cdot 10^9$ бит.

Интенсивность кадров ВКС в ЧНН составит

$\lambda_{ВКС} = d_{1ВКС} / d_{1\,кадр} = 885 \cdot 10^9 / 14,75 \cdot 10^6 = 60 \cdot 10^3$.

При скорости работы сервера *Vcep.*=12,8 Гбит/с (12,8 · 10^9бит/с) для обслуживания 1 ВКС в ЧНН сервер затратит время: $t_{1\,ВКС} = d_{1ВКС} / Vcep. = 885 \cdot 10^9 / 12,8 \cdot 10^9 = 69,12$ с.

Интенсивность нагрузки сервера на обслуживание 1 ВКС в ЧНН: $\rho_{ВКС} = t_{1\,ВКС} / t_{чнн} = 69,14 / 3600 = 0,0192$ Эрл.

Пакетизация сообщений передачи данных (ПД).

Пусть число пользователей ПД в ЧНН равно $n_{ПД}$ =100. Пусть один пользователь скачивает данные со скоростью $V_{1ПД}$=10·10^6 бит/с в течение $t_{1ПД}$=10 минут (6·10^2с). Объём информации в ЧНН от 100 пользователей ПД за 10 минут составит $d_{ПД} = V_{1ПД} \cdot t_{1ПД} \cdot n_{ПД} = 10 \cdot 10^6 \cdot 6 \cdot 10^2 \cdot 100 = 600 \cdot 10^9$ бит.

Интенсивность заявок ПД $\lambda_{ПД} = d_{ПД} / d_{1\,ПД} = 600 \cdot 10^9 / 10 \cdot 10^6 = 60 \cdot 10^3$.

Для обслуживания 100 ПД в ЧНН сервер затратит время: $t_{100\,ПД} = d_{ПД} / Vcep = 600 \cdot 10^9 / 12,8 \cdot 10^9 = 46,875$ с.

Интенсивность нагрузки сервера на обслуживание 100 пользователей ПД в ЧНН $\rho_{ПД} = t_{100\,ПД} / t_{чнн} = 46,875 / 3600 = 0,013$ Эрл. при обслуживании сервером всех заявок в ЧНН равна $\rho = \rho_{гол.} + \rho_{ВКС} + \rho_{ПД} = 0,00409 + 0,0192 + 0,013 = 0,03629$ Эрл.

Параметры потоков сообщений приведены в таблице 1. Организация VLAN (*Virtual Local Area Network* - виртуальная локальная сеть) это логическое деление пакетного коммутатора на выходе сети доступа на не сообщающиеся между собой сети. Идентификатор *VLAN (VID)* имеет размер поля 12 бит, диапазон возможных значений *VLAN* – от 0 до 4094.Технология *VLAN* описана в стандарте *IEEE* 802.1*Q* [9]. Для идентификации принадлежности трафика к определённой виртуальной локальной сети (*VLAN*) производится тегирование трафика – добавления в заголовок кадра специальной метки (тега). Тег определяет восемь уровней приоритета потоков сообщений.

На канальном уровне *L*2 в технологии Ethernet [10] класс услуг *CoS (Class of Service*) – это 3-битное поле для маркировки кадров в соответствии со стандартом *IEEE* 802.1*p*. Оно расположено в 4-байтовом заголовке *IEEE* 802.1*Q*. Для маркировки пакетов на сетевом уровне *L*3 используется поле *DSCP* (*Differentiated Services Code Point* – кодовая точка дифференцированных услуг), позволяющее классифицировать сетевой трафик в *IP*-сетях (деление потоков сообщений на классы приоритетов по обслуживанию – табл. 2). Разметкой кадра является определение так называемой метки *QoS*. Для кадров, поступающих в плату сети доступа без тега *VLAN*, выполняется начальное маркирование с добавлением в них тега *VLAN*.

Таблица 1

Параметры потоков сообщений

№ п/п	Тип сообщения	Число источников	Число сообщений в ЧНН	Средняя длительность одного сооб. в секундах	Размер одного пакета/ кадра, бит	Интенсивность пакетов/кадров - λ	Интенсивность нагрузки вЧНН, Эрл
1	ГС	$10 \cdot 10^3$	$30 \cdot 10^3$	72	G.711 $1{,}744 \cdot 10^3$	$108 \cdot 10^6$	0,00409
2	ВКС	1	1	1 200	$14, 75 \cdot 10^6$	$60 \cdot 10^3$	0,0192
3	ПД	100	100	600	$10 \cdot 10^6$	$60 \cdot 10^3$	0,013

Таблица 2

Деление потоков сообщений на классы приоритетов на уровне *L*3

Класс приоритета *QoS*	Код приоритета	Тип приоритета	Назначение приоритета
0	000	*Routine*	Самый низкий приоритет
1	001	*Best Effort Service*	Негарантированная доставка
2	*010*	*Immediate*	Данные с высоким приоритетом
3	011	*Flash*	Важные приложения
4	100	*Flash Override*	Передача видео
5	101	*Enterprise Cache Protocol*	Передача голоса
6	110	*Routing*	Межсетевой контроль
7	111	*Network Management Traffic*	Трафик управления сетью. Самый высокий приоритет

Коммутация пакетов осуществляется с помощью пакетных коммутаторов/маршрутизаторов, работающих на базе стека протоколов TCP/IP.Основными мультимедийными широкополосными услугами являются: получение и передача данных, звука, видеоизображения. Сеть доступа должна обеспечить комплексное их предоставление через единую инфраструктуру.

Размещение пакетов в очередях в порядке убывания приоритетов. На входе в устройство кадр проходит процедуру классификации. Для этого устройство разбирает последовательность бит, находящуюся в теге *VLAN*, и в соответствии с ней избирает дальнейший сценарий действий. Эта последовательность может указывать на правила обслуживания *QoS*, действия, которые обработчик должен проделать перед его отправкой на выход, например, измерить скорость кадра, при необходимости назначить кадру новую метку *QoS*, далее кадр проходит механизмы балансировки интенсивностей нагрузки (если требуется) и поступает в очередь на обработку планировщика. Механизм предотвращения перегрузки вступает в действие непосредственно перед назначением кадров выходным очередям по приоритету. Цель реализации этого механизма состоит в предотвращении заполнения очередей кадрами с последующим неконтролируемым "отбрасыванием хвоста" кадров. Используется взвешенное произвольное раннее обнаружение (*WRED - Weighted Random Early Detection*), которое является более совершенным алгоритмом предотвращения перегрузки, поскольку оно обеспечивает возможность заблаговременного выборочного отбрасывания определённых пакетов до заполнения очередей.

Для назначения приоритетов потокам трафика большое значение имеет правильная стратегия дебуферизации выходных очередей. Потоки, для которых необходимо обеспечить малое время задержки и постоянную пропускную способность, должны обслуживаться первыми и, соответственно, первыми передаваться через интерфейс. Для каждой очереди предусмотрен максимальный допустимый размер в диапазоне от 0 до 256 кадров.

Существуют различные методы обслуживания приоритетных очередей:

– *строгий приоритет очереди SP* (*Strict Priority queuing* – пакеты распределяются по очередям в соответствии с классом обслуживания);

– *взвешенный циклический перебор WRR* (*Weighted Round Robin).*

Для этого алгоритма предусмотрен параметр "кредит", связанный с каждой очередью и определяющий количество пакетов, подлежащих отправке из очереди в рамках каждого цикла. При отправке пакета происходит уменьшение значения кредита. Максимальное значение кредита равно весу очереди.

Наиболее продуктивным является алгоритм *LLQ (Low Latency Queuing – организация очередей с низкой задержкой*), который представляет собой смесь *SP* и *WRR*. Передаваемый в реальном времени трафик (такой как *VoIP*), помещается в очереди со строгим приоритетом, а остальной трафик - в очереди *WRR*. В общем случае должна быть одна приоритетная очередь с голосовым трафиком.

Диспетчеризация выходных очередей по приоритету вместе с классификацией на входе представляет собой основной механизм *QoS*.

Блок организации очередей. Данный блок расположен на выходе платы. Базовым компонентом блока являются выходные очереди по приоритету или просто очереди. Перед передачей кадры помещаются в очередь, где они ожидают своей очереди на передачу.

Если интенсивность входящего потока кадров превышает интенсивность исходящего потока кадров, происходит перегрузка очереди. В этом случае цель реализации очередей состоит в разделении кадров на "более важные" и "менее важные". Можно сказать, что каждая очередь представляет собой один поток.

Таким образом, обеспечивается возможность назначения более важным кадрам более высокого приоритета, чем менее важным. Данный блок обеспечивает реализацию следующих функций:

- установление соответствий для очередей: определение очереди, в которую следует поместить кадр в соответствии с меткой *QoS*;
- предотвращение перегрузки: сравнение кадра с данными измерителей по коэффициенту использования очереди. Если при реализации алгоритма предотвращения перегрузки выявляется повышенный коэффициент использования очереди для данного кадра (в соответствии с меткой *QoS*), выполняется отбрасывание кадра. В противном случае кадр помещается в очередь.

Конечным результатом для данного блока является постановка пакета в одну из очередей или отбрасывание пакета. Другие действия на данном этапе не выполняются.

Циклическое обслуживание по строгому приоритету. Обслуживание по строгому приоритету (SP - Strict Priority) представляет собой основной алгоритм диспетчеризации. Каждой очереди назначается так называемый приоритет, определяющий приоритет обслуживания очереди по отношению к другим очередям. Обслуживание очереди с наиболее высоким приоритетом в системе очередей выполняется, пока в этой очереди не заканчиваются кадры.

Главное преимущество обслуживания по строгому приоритету заключается в возможности обеспечения привилегированного обслуживания кадров, для которых необходимо обеспечить абсолютный приоритет по отношению к остальным. Обслуживание по строгому приоритету целесообразно использовать при обработке высокоприоритетного трафика с малой полосой пропускания, для которого необходимо обеспечить абсолютный приоритет обработки по отношению к другим типам трафика. Для оценки задержек сообщений в очередях разработан приближённый метод анализа средних [11].

Расчёт стационарных вероятностей u_i опроса очередей Q_i, $i = \overline{1,N}$ и среднего времени цикла C

В методе анализа средних для расчётов $u_i, i = \overline{1,N}$, и C рекомендована следующая система нелинейных уравнений

$$u_i^{n+1} = 1/(1+ e^{-\lambda_i c^n}),\ i = \overline{1,N}, \tag{1}$$

$$C^{n+1} = [\sum_{i=1}^{N} s_i u_i^{n+1} + \beta \prod_{i=1}^{N} (1-u_i^{n+1})]/(1-\rho),\ n \neq 0. \tag{2}$$

Система уравнений (1) – (2) решается с применением итерационного метода простых итераций. Для расчётов использован калькулятор экспоненты [12]. Для расчёта примера приняты следующие начальные условия среднего времени: цикла $C^0 = 5 \times 10^{-5}$; подключения сервера к очереди $s_i = 10^{-5}$.

Результаты расчётов параметров C, u_1, u_2, u_3 в примере с помощью простого итерационного процесса приведены в таблице 3. Простой итерационный процесс обеспечивает высокую точность сходимости. В приведённом примере при пяти итерациях отличие на соседних шагах значений C составляет $0{,}0004 \cdot 10^{-5}$.

Порядок опроса очередей - циклический, при котором сервер посещает очереди в порядке возрастания их номеров, а затем вновь возвращается к первой очереди: $Q_1, Q_2, \ldots, Q_N, Q_1, Q_2, \ldots, Q_N, \ldots$. Системы поллинга с таким порядком опроса называют *циклическими,* а время обслуживания очередей с первой до последней – *циклом.*

Таблица 3

Результаты расчётов параметров C, u_1, u_2, u_3 с помощью простого итерационного процесса

№ п/п	Среднее время цикла – С	Отличие на соседних шагах $C^n - C^{n-1}$	u_1	u_2	u_3
0	$C^0 = 5 \cdot 10^{-5}$	-	-	-	-
1	$C^1 = 3{,}014 \cdot 10^{-5}$	$1{,}9858 \cdot 10^{-5}$	1	0,952	0,952
2	$C^2 = 2{,}82 \cdot 10^{-5}$	$0{,}1941 \cdot 10^{-5}$	1	0,859	0,859
3	$C^3 = 2{,}79 \cdot 10^{-5}$	$0{,}03 \cdot 10^{-5}$	1	0,844	0,844
4	$C^4 = 2{,}785 \cdot 10^{-5}$	$0{,}005 \cdot 10^{-5}$	1	0,842	0,842
5	$C^5 = 2{,}7846 \cdot 10^{-5}$	$0{,}0004 \cdot 10^{-5}$	1	0,841	0,841

Параметры системы поллинга, приведённые в таблицах 1 и 3, позволяют найти значения среднего числа заявок L_i в каждой очереди Qi, $i = \overline{1,3}$. При шлюзовой дисциплине обслуживания очереди, при которой сервер обслуживает лишь те заявки, которые находились в очереди в момент начала её опроса. Заявки, поступившие в очередь после опроса, обслуживаются в следующем цикле. При шлюзовой дисциплине обслуживания очереди Q_i, условное среднее $L_{i,i}$ определяется суммой двух слагаемых: $\vec{Li}$ – среднее число заявок, обслуживаемых сервером за время посещения очереди Q_i , и $\tilde{L}_{i,j}$ – среднее число заявок, поступивших в очередь после опроса данной очереди. Длина очереди $\tilde{L}_{i,n}$ в период (i, j) – это число заявок, которое поступит в очередь в течение рассматриваемого периода:

$$\sum_{n=i}^{i+j-1}(q_{n,1}/q_{i,j})\tilde{L}_{i,n} = \lambda_i(v_{i,j} + (1-u_i-u)/(1-\rho+\rho_m)\sum_{\substack{m=1\\m\neq i}}^{N} u_m s_m), i,j = \overline{1,N}. \quad (3)$$

При N=3 число уравнений (3) равно 6. Время пребывания сервера у очереди Q_i , $v_i = \rho_I C + s_i u_i, \quad i = \overline{1,N}$. Средняя продолжительность (i,j)-го периода $v_{i,j} = \sum_{n=i}^{i+j-1} v_n$, $i,j = \overline{1,N}, \quad q_{i,j} = v_{i,j}/C, \quad i,j = \overline{1,N}.$ Среднее число заявок $\vec{Li}$, которое будет обслужено сервером за время посещения очереди Q_i, вычисляется по выражению:

$$\vec{L_i} = \lambda_i u_i / \rho_i (v_i + (1-u_i-u)/(1-\rho+\rho m)\sum_{\substack{m=1\\m\neq i}}^{N} u_m s_m) - (1-\rho_i)\, u_i / \rho_i \sum_{n=1}^{N} q_{n,1} L_{i,n}. \quad (4)$$

При N=3 число уравнений (4) равно 3. После вычисления значений $\tilde{L}_{i,n}$ и $\vec{Li}$ по выражениям (3) и (4), среднее число заявок в очереди Q_i вычисляется по выражению:

$$L_i = \sum_{n=1}^{N} \tilde{L}_{i,n} q_{n,1} + \vec{L}_i(\rho_i / u_i), \; i = \overline{1,N}. \quad (5)$$

При N=3 число уравнений (5) равно 3. Среднее время пребывания заявки в очереди (величина задержки) определяется по теореме Дж. Литтла.

$$W_i = L_i / \lambda_i, \; i = \overline{1,N}. \quad (6)$$

Всего при N=3 в системах (3) - (6) необходимо решить 15 уравнений.

Заключение

При анализе функций сети доступа проведена оценка основных параметров обслуживаемых сообщений. Оценки отдельных параметров использовались при расчёте стационарных вероятностей опроса всех очередей и среднего времени цикла.

Полученные оценки параметров сети доступа позволяют с помощью метода средних решить системы уравнений для нахождения среднего числа пакетов в каждой очереди и с помощью теоремы Дж. Литтла рассчитать задержки пакетов в очередях.

Для простейшего примера при N=3 в системах (3) – (6) необходимо решить 15 достаточно сложных уравнений. Решение этих систем уравнений вынесено в отдельную задачу и в настоящей работе не приводится.

Литература

1. Сети следующего поколения *NGN* / под ред. А.В. Рослякова. М.: Эко-Трендз, 2009. 424 с.

2. *Росляков А.В., Ваняшин С.В*. Будущие сети (Future Networks). Самара: ПГУТИ, 2015. 274 с.

3. *Росляков А.В.* Сети фиксированной связи пятого поколения. Учебное пособие. М.: ООО «ИКЦ «Колос-с», 2024. 232 с.

4. Закон Российской Федерации от 07.02.1992 N 2300–1 (ред. от 08.08.2024) «О защите прав потребителей».

5. Рекомендация МСЭ-Т *Y*.1541. Требования к сетевым показателям качества для служб, основанных на протоколе IP/ Международный союз электросвязи – Женева, 2007. 42с.

6. *Пшеничников А.П.* Теория телетрафика. Учебник для вузов. М.: Горячая линия – Телеком, 2019. 212 с.

7. Видеофайлы: разбираемся в кодеках, контейнерах и нюансах воспроизведения. https://club.dns-shop.ru/blog/t-99-videokartyi/111834-videofailyi-razbiraemsya-v-kodekah-konteinerah-i-nuansah vosproi/?ysclid=m63u64o1n5144457166&utm_referrer=https%3A%2F%2Fyandex.ru%2F (дата обращения 12.12.2024).

8. ГОСТ Р 53536–2009. Цифровое телевидение повышенной чёткости. Основные параметры цифровой системы с построчным разложением. Аналоговые и цифровые представления сигналов. Параллельный цифровой интерфейс. М.: Стандартинформ, 2020.

9. IEEE Standards Association – IEEE 802.1Q-2022. http://standards.ieee.org›ieee/802.1Q/10323/ (дата обращения 07.12.2024).

10. *Нетес В.А*. Ethernet операторского класса. Учебное пособие для вузов. М.: Горячая линия – Телеком, 2023. 128 с.

11. *Вишневский В.М., Семёнова О.В.* Системы поллинга: теория и применение в широкополосных беспроводных сетях. М.: Техносфера, 2007. 312 с.

12. Калькулятор экспоненты. smartcalculator.online/ (дата обращения 07.12.2024).

АНАЛИЗ МЕТОДОВ ПРОГНОЗИРОВАНИЯ ОТКАЗОВ В ВЕДОМСТВЕННОЙ СЕТИ СВЯЗИ

Фатхулин Тимур Джалилевич
Московский технический университет связи и информатики, доцент кафедры МК и ИТ, к.т.н., Москва, Россия
t.d.fatkhulin@mtuci.ru

Сазыкин Сергей Владимирович
Московский технический университет связи и информатики, Москва, Россия

Рахматова Азиза Акрамовна
Московский технический университет связи и информатики, Москва, Россия

Аннотация
В статье рассматриваются основные методы машинного обучения для прогнозирования отказов в сети передачи данных. Цель работы – проанализировать существующие методы прогнозирования отказов, которые могут быть использованы в ведомственных сетях связи. Дан краткий обзор применяемых наборов данных и признаков для обучения моделей, а также сопутствующих трудностей в применении методов машинного обучения. В заключении сделаны выводы по проведенному исследованию.

Ключевые слова
Сети связи, машинное обучение, временные ряды, нейросети, прогнозирование трафика

Введение

В последнее время, благодаря быстрому развитию технологий, связанных с интернетом появлению новых услуг и устройств таких как облачные хранилища, потоковое видео высокого разрешения, VR, телефоны и домашние устройства умного дома, объем трафика данных во всем мире резко возрастает. Сети становятся все более разнообразными и запутанными для эффективной обработки трафика и контроля за широким спектром устройств. Типичная производственная сеть включает в себя широкий спектр устройств, работает по многочисленным протоколам (ZigBee, WiMAX, IEEE 802.11 ac/ad, Bluetooth, и LTE) и поддерживает различные приложения. Такое расширение сетевой инфраструктуры увеличивает сложность сети и создает множество проблем в эффективной организации, администрировании и оптимизации сетевых ресурсов [1-4]. С учетом этого прогнозирование ошибок с использованием временных рядов, позволит повысить отказоустойчивость сложных многоуровневых сетей с минимальными вложениями.

Анализ применимости методов машинного обучения в рассматриваемой задаче

Машинное обучение (МО) – это методы искусственного интеллекта, с помощью которых можно создавать самообучающиеся компьютерные системы без использования программирования, использую только шаблоны и логические выводы. По сути, цель МО - обнаружить и использовать скрытые закономерности в "обучающих" данных. Выявленные закономерности затем используются для анализа новых, скрытых данных, их классификации или сопоставления с уже известными группами. Такой подход представляет собой отказ от традиционного программирования, при котором программы для выполнения задач пишутся вручную [12-15].

Машинное обучение может быть применено к четырем основным категориям задач: кластеризация, классификация, регрессия и извлечение правил. Цель кластеризации – сгруппировать похожие точки данных, максимально увеличив расстояние между различными группами. В целом, МО особенно подходит для решения задач, связанных с наличием большого репрезентативного набора данных, позволяя извлекать закономерности и делать прогнозы на основе этих закономерностей [15]. В основе методологий машинного обучения лежит хорошо известная теорема Байеса об условной вероятности, а также фундаментальное правило, связывающее общую вероятность с условной вероятностью [2, 7].

Недавние достижения в области вычислительной техники предоставили необходимую вычислительную мощность и объемы хранения данных, чтобы обучать и тестировать модели машинного обучения на больших массивах данных. Например, облачные вычисления предлагают практически неограниченные вычислительные ресурсы, в то время как графические процессоры (GPU) и тензорные процессоры (TPU) ускоряют как обучение, так и вывод моделей для обработки огромных объемов данных [3-6]. Примечательно, что после того, как МО обучена, ее можно использовать для вывода данных на менее мощных устройствах, таких как смартфоны. Однако, несмотря на технический прогресс, работа СПД и управление ими остаются сложными и подвержены возникновению ошибок, а сбои в работе сети часто вызваны человеческим факто-

ром. Эти сбои могут привести к значительным финансовым потерям и ущербу репутации сетевых провайдеров [20]. В результате все большее внимание уделяется разработке автономных сетей – самонастраивающихся, самовосстанавливающихся, самооптимизирующихся и самозащищающихся систем, – которые являются более устойчивыми и способны свести к минимуму необходимость вмешательства человека.

История машинного обучения (МО) начинается в 1943 году, когда Уоррен Маккалох предложил первую математическую модель нейронных сетей (НН) в компьютерах. В этой модели был представлен искусственный нейрон, который и по сей день остается основой для разработки НН. Однако первоначальная модель требовала ручной настройки весов соединений между нейронами. В 1949 году это ограничение было устранено с помощью Hebbian learning, алгоритма, основанного на правилах, для автоматического обновления весов. Как искусственный нейрон, так и хеббианское обучение оказали глубокое влияние на эволюцию нейронных сетей. в 1950 году [1-5, 6, 7, 15]. В том же году Алан Тьюринг разработал "Тест Тьюринга" – метод для оценки способности компьютера демонстрировать разумное поведение, пытаясь обмануть человека, заставив его поверить, что он человек.

В 1950-х годах для расчетов линейной регрессии на электромеханических настольных калькуляторах использовался Ordinary Least Squares (OLS) – метод, который используется в линейной регрессионной модели для нахождения лучшей подгонки линии для набора точек данных путём минимизации остатков. На основе этого были введены две линейные модели классификации: максимальная энтропия (MaxEnt) и логистическая регрессия. Одновременно с этим акцент на распознавании образов привел к разработке двух непараметрических моделей (не ограниченных предопределенным набором параметров), способных выполнять регрессию и классификацию: k-ближайших соседей (k-NN) и оценку плотности ядра (KDE) [8. 10], а также известна как плотность Парцена. В то время как алгоритм k-NN анализирует данные с использованием показателя расстояния, KDE применяет функцию ядра (Гауссову) для оценки функции плотности вероятности данных.

Применение временных рядов при прогнозировании отказов в ведомственной сети связи

Временной ряд – это последовательность данных, полученных в результате регулярных наблюдения за переменной на протяжении определенного промежутка времени. Эти наблюдения обычно записываются в хронологическом порядке, что облегчает анализ закономерностей [7-15, 20]. Данные временных рядов могут охватывать широкий спектр задач из самых разных областей, таких как финансы, экономика, инженерия, метеорология и социальные науки. Главной особенностью временных рядов является их непостоянство, при котором время появления данных играет решающую роль в понимании изменений, лежащих в основе данных. В отличие от информации, полученной от поперечного сечения, которые собирают информацию в один момент времени о нескольких организациях или отдельных лицах, данные временных рядов фокусируются на отслеживании одной переменной за конкретный период [1, 4-5].

Ключевой характеристикой временных рядов является их зависимость от прошлых наблюдений. Каждая точка данных привязана к определенному временному промежутку, образуя структурированную временную шкалу, которая показывает, как развивается событие [13-15, 17]. Данные временных рядов могут демонстрировать широкий спектр закономерностей, от предсказуемых и регулярных циклов до неустойчивых и случайных событий [1]. Общие закономерности включают тренды, которые отражают долгосрочные изменения в данных, и сезонность, которая определяет повторяющиеся циклы с фиксированными интервалами. Кроме того, данные временных рядов часто содержат случайный шум, отражающий изменчивость или неопределенность.

Существующие программные решения для анализа временных рядов

Для анализа временных рядов существует множество прикладных программ, предлагающие различные средства, включая хранение, визуализацию, анализ данных и прочие. Ниже приведены некоторые из них.

- Grafana [7] – реализованное как веб-приложение программная система визуализации данных для ИТ-мониторинга. Она прекрасно подходит для работы с временными рядами.
- STUMPY [7] – это надежная и масштабируемая библиотека Python, предназначенная для эффективного матричного профиля. В этой библиотеке имеется множество методов для поиска примитивов в данных, таких как мотивы, диссонансы, цепочки и другие элементы.
- Matplotlib [7] – это библиотека для создания графиков и визуализаций данных на Python, которая поддерживает создание статических, анимированных и интерактивных визуализаций. Главная цель Matplotlib – предоставить пользователям разнообразные инструменты для создания графиков различных типов: от простых линейных до сложных трехмерных. Она значительно упрощает анализ данных, помогая визуализировать и лучше понять их. В настоящее время поддерживается большим сообществом разработчиков.

- ClickHouse [7] – высокопроизводительная колонковая база данных с открытым исходным кодом, предназначенная для обработки больших объемов данных и выполнения оперативной аналитической обработки. Она использует все системные ресурсы для максимально быстрой обработки запросов и доступна как в виде открытого исходного кода, так и в виде облачного приложения.
- Graphite [7] – это инструмент для сбора и визуализации метрик и временных рядов, который отлично работает как на недорогом оборудовании, так и в облачной инфраструктуре. Graphite предоставляет мощные средства агрегации данных и построения графиков для мониторинга производительности системы.

Анализ методов прогнозирования трафика без использования временных рядов

В отличие от традиционных методов прогнозирования временных рядов (TSF), для прогнозирования сетевого трафика можно использовать альтернативные подходы. Существует метод, который работает в частотной области для моделирования потоков сетевого трафика, выходя за рамки простого прогнозирования. [1-6]. Их внимание сосредоточено на прогнозировании входящего и исходящего трафика по каналам связи между центрами обработки данных, в которых преобладают "elephant flow"[3].

В предлагаемой модели используется нейронная сеть прямого действия (FHH), обученная с помощью обратного распространения (BP) и простого градиентного спуска, в сочетании с волновыми преобразованиями для получения временных и частотных характеристик временных рядов в трафике [21-25]. Elephant flow рассматриваются как отдельные характеристики в процессе прогнозирования. Однако отслеживание всех elephant flow на высоких частотах сопряжено с большими затратами по сравнению с мониторингом объема трафика по количеству байт [15].

Для решения этой проблемы данные об elephant flow формируются на более низких частотах, а затем интерполируются для заполнения пробелов, что сокращает накладные расходы, связанные с частым сбором данных об elephant flow [3]. Набор данных, используемый для обучения, состоит из общего объема входящего и исходящего трафика, измеренного с интервалом в 30 секунд и собранного с помощью счетчиков SNMP на оконечных маршрутизаторах и центрах обработки данными (DC). Кроме того, комбинированная модель нейронной сети и волновых преобразований позволяет снизить пиковую нагрузку каналов постоянного тока на 9%, что важно для выставления счетов интернет-провайдерами.

Результаты анализа систематизированы и представлены в таблице 1.

Таблица 1

Методы прогнозирования трафика

Способ	Преимущества	Недостатки	Применение
Многослойный персептрон (MLP)	- может моделировать сложные нелинейные зависимости. - подходит для различных типов данных	- требует тщательной настройки параметров. - склонен к переобучению при ограниченном количестве данных.	- достаточно данных. - когда в данных присутствуют нелинейные закономерности.
Рекуррентные нейронные сети (RNN)	- подходит для последовательных данных. - LSTM/GRU хорошо справляются с долгосрочными зависимостями.	- требует больших вычислительных мощностей. - сложен в проектировании и настройке.	- когда временные зависимости имеют решающее значение. - для задач, связанных с долговременной памятью (например, LSTM, GRU).
Сверточные нейронные сети (CNN)	- эффективен при распознавании локальных закономерностей. - эффективен с точки зрения вычислений благодаря параллельной обработке.	- может возникнуть проблема с долгосрочными зависимостями. - требует тщательного проектирования архитектуры.	- когда локальные закономерности имеют большое значение. - когда необходима высокая вычислительная эффективность.
Модели, основанные на деревьях решений	- хорошо справляется с нелинейными зависимостями. - устойчив к переобучению при правильной настройке. - требует меньше усилий при разработке функций.	- менее интерпретируемый. - требует больших вычислительных затрат для больших наборов данных.	- при сложных взаимодействиях признаков. - для надёжных, высокопроизводительных моделей с меньшим количеством признаков.
Трансформерные-ные нейронные сети	- эффективно выявляет долгосрочные зависимости. - позволяет проводить параллельное обучение, ускоряя процесс.	- требуются большие наборы данных. - сложный в реализации и настройке.	- для моделирования долгосрочных зависимостей. - когда параллельная обработка выгодна. - при работе с большими наборами данных.

Заключение

В ходе анализа существующих методов прогнозирования сети выявлены ключевые особенности и проблемы, связанные с точностью и эффективностью текущих подходов. Эти методы зачастую имеют ограничения в доступности данных, сложности учета всех факторов, влияющих на вероятность отказа, и недостаточной адаптивности к изменениям в сети. Прогнозирование отказов в сетях связи требует учета множества переменных, таких как состояние оборудования, нагрузка на каналы связи, технические характеристики устройств и внешние воздействия.

Литература

1. *Chollet F.* Deep Learning with Python – Manning Publications, 2017. 384 p.

2. *Geron Au.* Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems Second Edition (Third Release) – O'Reilly Media, 2019. 856 p.

3. *Kong X., Hu C., Duan Z.* Principal Component Analysis Networks and Algorithms. Springer Singapore, 2017. 323 p.

4. *Liu J.* Radial Basis Function (RBF) Neural Network Control for Mechanical Systems: Design, Analysis and Matlab Simulation. Springer Berlin Heidelberg, 2014. 365 p.

5. *Min F., Zhu W.* A Competition Strategy to Cost-Sensitive Decision Trees // International Conference on Rough Sets and Knowledge Technology – 7th International Conference, RSKT 2012, Chengdu, China, August 17-20, 2012. Proceedings: Rough Sets and Knowledge Technology. Springer-Verlag Berlin Heidelberg 2012, pp. 359-368.

6. *Chen X., Yang R., Xue Y., Huang M., Ferrero R., Wang Z.* Deep Transfer Learning for Bearing Fault Diagnosis: A Systematic Review Since 2016. // IEEE Transactions on Instrumentation and Measurement, 2023. 21 с.

7. *Peng B., Bi Y., Xue B., Zhang M., Wan S.* A Survey on Fault Diagnosis of Rolling Bearings // Algorithms, 2022. 347 с.

8. *Рашка Себастьян, Мирджалили Вахид.* Python и машинное обучение: машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow 2, 3-е изд.: Пер. с англ. СПб.: ООО "Диалектика", 2020. 848 с.

9. *Althubaiti A., Elasha F., Teixeira J. A.* Fault diagnosis and health management of bearings in rotating equipment based on vibration analysis – a review // Journal of Vibroengineering, 2021. С. 46-74.

10. Прогнозирование временных рядов в Python [Электронный ресурс] URL: https://teletype.in/@pythontalk/time_series_forecasting (дата обращения 14.12.2024

11. *Zhirkin Y.V., Puzik E.A., Filatov A.A., Sultanov N.L.* Prolonging the Service Life of the Rolling Bearings of the Work Rolls of the 2000 Tandem Cold-Rolling Mill at the Magnitogorsk Metallurgical Combine // Metallurgist, 2017. С. 1180-1182.

12. *Raouf Boutaba, Mohammad A. Salahuddin, Noura Limam, Sara Ayoubi, Nashid Shahriar, Felipe Estrada-Solano and Oscar M. Caicedo.* A comprehensive survey on machine learning for networking: evolution, applications and research opportunities

13. *Вострикова П.В., Рыбка С.О., Рыжкова У.С., Фатхулин Т.Д.* Анализ нейросетевых технологий, использующихся для улучшения качества изображений // REDS: Телекоммуникационные устройства и системы. 2024. Т. 14, № 1. С. 57-65. EDN WVBDRN

14. *Леохин Ю.Л., Фатхулин Т.Д.* Разработка методов и алгоритма формализации текстового запроса к онлайн-сервисам, генерирующим изображения посредством нейросетевых технологий // Вестник Рязанского государственного радиотехнического университета. 2023. № 85. С. 82-95. DOI 10.21667/1995-4565-2023-85-82-95. EDN PZWYZV

15. *Леохин Ю.Л., Фатхулин Т.Д., Кожанов М.С.* Анализ и исследование применения нейросетевых технологий для генерации программного кода // Вестник Рязанского государственного радиотехнического университета. 2024. № 87. С. 41-53. DOI 10.21667/1995-4565-2024-87-41-53. EDN HKEOFX

16. *Леохин Ю.Л., Фатхулин Т.Д., Ментус М.В.* Разработка и применение методов распознавания зашумленных аудиофайлов посредством нейросетевых технологий // Вестник Рязанского государственного радиотехнического университета. 2024. № 88. С. 65-73. DOI 10.21667/1995-4565-2024-88-65-73. EDN NMXASI

17. *Маслов К.В., Фатхулин Т.Д., Иванов Д.А.* Анализ технологий автоматизации бизнес-процессов и разработки программного обеспечения с использованием low-code платформ // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 1. С. 6-11. EDN HDBOYM

18. *Фатхулин Т.Д., Бойцов К.В.* Анализ функционала программного обеспечения, применяемого для классификации труб на предприятии методами компьютерного зрения // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 6-12. EDN FVQYQA

19. *Мяличева А.А., Фатхулин Т.Д.* Анализ методов машинного обучения для прогнозирования дефектов в исходном коде // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 16-19. EDN IVJCZF

20. *Фатхулин Т.Д., Леонова В.О., Тремасова Л.А.* Анализ нейросетевых технологий, применяемых для web-разработки // REDS: Телекоммуникационные устройства и системы. 2024. Т. 14, № 2. С. 35-41. EDN SDCNKM

21. *Фатхулин Т.Д., Исаев А.В.* Анализ моделей ARIMA и LSTM, используемых для прогнозирования криптовалют и определения портфеля инвестиций // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 20-25. EDN ODWOPA

22. *Фатхулин Т.Д., Смирнов Д.А., Разумов И.В. и др.* Анализ влияния составляемых текстовых запросов (промптов) на качество изображений, генерируемых нейросетевыми технологиями // Системы синхронизации, формирования и обработки сигналов. 2024. Т. 15, № 2. С. 52-57. EDN TSVMSK

23. *Фатхулин Т.Д., Фатхулина Г.Г., Ментус М.В.* Разработка методики формирования запроса к нейросети с целью генерации изображений с учетом рекомендаций компьютерной лингвистики // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2023. № 1. С. 133-139. EDN PPRTOM

24. *Фатхулин Т.Д., Исаев А.В.* Анализ эффективности использования моделей ARIMA для прогнозирования котировок и определения портфеля инвестиций в области криптовалюты // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 26-31. EDN CESRTK

25. *Фатхулин Т.Д., Бойцов К.В.* Оценка эффективности алгоритма на основе YOLO v.8 для классификации труб на предприятии по фото в зависимости от различных условий // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 51-55. EDN KWLMYA

26. *Вишневский В.М., Леохин Ю.Л., Фатхулин Т.Д., Занегин А.В.* Методы машинного обучения в решении задачи прогнозирования спроса на отдельные виды товаров // T-Comm: Телекоммуникации и транспорт. 2024. Т. 18. №10. С. 34-43.

ОПТИМИЗАЦИИ ИНТЕРНЕТ-ТРАФИКА ПОСРЕДСТВОМ ПРОТОКОЛОВ HTTP/3 И QUIC

Гадасин Денис Вадимович
МТУСИ, доцент кафедры СИТиС, к.т.н., Москва, Россия
dengadiplom@mail.ru

Кобелькова Александра Дмитриевна
МТУСИ, Москва, Россия
kad08.02@mail.ru

Родина Алина Андреевна
МТУСИ, Москва, Россия
alina_rodina2004@mail.ru

Сурова Мария Андреевна
МТУСИ, Москва, Россия
surrovamma@mail.ru

Аннотация
Современные пользователи Интернета сталкиваются с растущими требованиями к скорости, надёжности и безопасности передачи данных. В ответ на эти вызовы разрабатываются различные эффективные решения, такие как усовершенствование имеющихся и создание новых транспортных и веб-протоколов. В целях оптимизации передачи данных была разработана новейшая версия протокола HTTP/3, который представляет собой значительный шаг вперёд по сравнению с предыдущими версиями HTTP за счет своих собственных характеристик и за счет своей реализации поверх современного и эффективного транспортного протокола QUIC. В данной работе рассматриваются ключевые особенности HTTP/3, включая его работу поверх QUIC. Рассматривается развитие протоколов HTTP, начиная с самой первой версии. Проводится анализ трех транспортных протоколов: TCP, UDP и QUIC и осуществляется сравнение их по множеству характеристик. Особое внимание уделено преимуществам протокола QUIC, как принципиально нового для работы в сочетании с протоколами HTTP. На основе анализа характеристик протокола HTTP/3 в сочетании с протоколом QUIC делаются выводы об эффективности их применения в современном Интернете.

Ключевые слова.
Транспортный протокол, веб-протокол, Интернет, протоколы HTTP, протокол UDP, протокол TCP, протокол QUIC, задержка, оптимизация, шифрование, соединение, передача данных.

Введение

За последние годы Интернет превратился в неотъемлемую часть жизни миллиардов людей по всему миру. Согласно отчетам Cisco Annual Internet Report количество пользователей Интернета к 2023 году возросло на 36 % с 2018 года и составило 5,3 миллиарда пользователей. Такой рост обусловлен развитием и внедрением сетевых технологий во все сферы жизни современного человека и расширением доступа к Интернет-услугам. Быстрое увеличение числа пользователей и постоянное развитие всевозможных онлайн-сервисов неизбежно сопровождаются стремительным увеличением объемов Интернет-трафика. В связи с этим возникает необходимость в создании новых эффективных средств для оптимизации передачи данных, что рассмотрено в работах [1-4].

HTTP/3 – это современная версия протокола прикладного уровня HTTP (HyperText Transfer Protocol), используемого для передачи гипертекстовых файлов, медиафайлов, документов и других ресурсов. HTTP/3 предлагает усовершенствованный механизм передачи HTTP-запросов и ответов, минимизируя блокировки и улучшая производительность веб-приложений. Протокол HTTP/3 реализован поверх протокола QUIC, инновационного для веб-приложений за счет своей способности объединять функции транспортного и криптографического уровней, обеспечивая высокую безопасность, обеспечивать высокую производительность, устранять задержки при установлении соединений и повышать устойчивость к потерям пакетов за счет сочетания преимуществ различных протоколов. Совместная работа этих технологий позволяет разработчикам создавать более отзывчивые и эффективные веб-приложения, обеспечивая более эффективный и безопасный опыт использования Интернета.

Для того чтобы сделать выводы об эффективности передачи данных с помощью современного протокола HTTP/3, использующего QUIC в качестве протокола транспортного уровня, необходимо рассмотреть различные аспекты протокола HTTP, включая предшествующие версии, а также подробно рассмотреть ар-

хитектуру и принципы работы протоколов QUIC и работающего в предшествующих версиях HTTP транспортного протокола TCP. На основании рассмотренных сведений необходимо провести сравнение характеристик транспортных протоколов, а также проиллюстрировать различия в их эффективности на примере сценариев передачи данных.

Развитие протоколов HTTP

Для определения преимуществ и нововведений протокола HTTP/3, работающего поверх QUIC, рассмотрим предшествующие этапы развития основного веб-протокола, организацию их взаимодействия с протоколами стека и проведем анализ транспортных протоколов, являющихся неотъемлемой частью функционирования HTTP.

История развития протокола HTTP (также известного как веб-протокол) начинается с 1991 года, когда британский информатик Тим Бернерс-Ли, который вместе с группой ученых, разрабатывающих концепцию всемирной паутины, создал первую версию протокола HTTP, работающего на прикладном уровне модели OSI. Изначально он использовался исключительно для получения методом GET с сервера гипертекстовых документов в формате HTML с использование протокола TCP (Transmission Control Protocol) транспортного уровня. Первая версия HTTP не имела технических возможностей для удовлетворения потребностей пользователей в передаче изображений и аудио и имел ряд других недостатков, таких, как ограничение использования пределами одного сегмента сети. Обращение клиента к серверу представляло из себя подключение и отправление единственного запроса на получение гипертекстового документа. В ответ сервер предоставлял доступ к запрашиваемому документу, и соединение прекращалось. В 1999 году IETF представила стандарт HTTP/1.1, который улучшил предыдущие версии, введя постоянные соединения, потоковую передачу данных, кэширование, коды состояния и новые методы (PUT, PATCH, HEAD, OPTIONS, DELETE). Это расширило возможности работы с веб-ресурсами, такими как HTML-документы, изображения, аудио и видео, идентифицируемыми через URI [5]. В 2000 году появился HTTPS, обеспечивающий безопасную передачу данных за счет шифрования и аутентификации с использованием TLS. Это сделало HTTPS стандартом для защиты информации в интернете [6]. В 2015 году был разработан HTTP/2, который ввел мультиплексирование для передачи нескольких запросов через одно TCP-соединение, бинарный формат данных и приоритезацию потоков. Основными элементами стали фреймы и потоки, что ускорило загрузку страниц и повысило производительность веб-приложений. В 2018 году началась разработка HTTP/3, основанного на протоколе QUIC. В отличие от HTTP/2, HTTP/3 использует QUIC для передачи данных через UDP, что уменьшает задержки и улучшает производительность, особенно в условиях нестабильных сетей. HTTP/3 поддерживает только защищенные HTTPS-соединения.

Стек протоколов, обеспечивающий работу HTTP, представляет собой многоуровневую структуру, состоящую из сетевых протоколов, которые совместно обеспечивают передачу данных в глобальной сети Интернет. С точки зрения модели OSI, HTTP, как протокол прикладного уровня, предназначен для обмена гипертекстовыми документами и другими ресурсами между клиентскими приложениями, такими как веб-браузеры, и серверными системами. Клиент отправляет HTTP-запрос, который направляется на сервер для обработки. Сервер, в свою очередь, формирует HTTP-ответ, содержащий запрашиваемые данные или результат выполнения операции, и отправляет его обратно клиенту. Этот процесс может повторяться многократно при последующих запросах. Запросы и ответы, сформированные на прикладном уровне, содержат информацию о методах и идентификаторах обрабатываемых ресурсов. Они передаются на транспортный уровень и обрабатываются с помощью соответствующих протоколов: во всех версиях, предшествующих HTTP/3, применяется TCP, в HTTP/3 используется протокол QUIC, работающий поверх UDP. На транспортном уровне происходит установление соединения между отправителем и получателем, разбиение данных на сегменты и осуществление передачи в соответствии с принципами работы каждого протокола. На сетевом уровне задействуется протокол IP (Internet Protocol), отвечающий за маршрутизацию пакетов между узлами сети. На канальном уровне происходит работа с MAC-адресами (Media Access Control), которые управляют доступом устройств к физическим средам передачи данных, например, Ethernet или Wi-Fi. Эти технологии отвечают за кодирование, модуляцию и передачу сигналов между устройствами.

Стеки протоколов, реализующие работу каждой из описанных версий протокола HTTP представлены на рисунке 1.

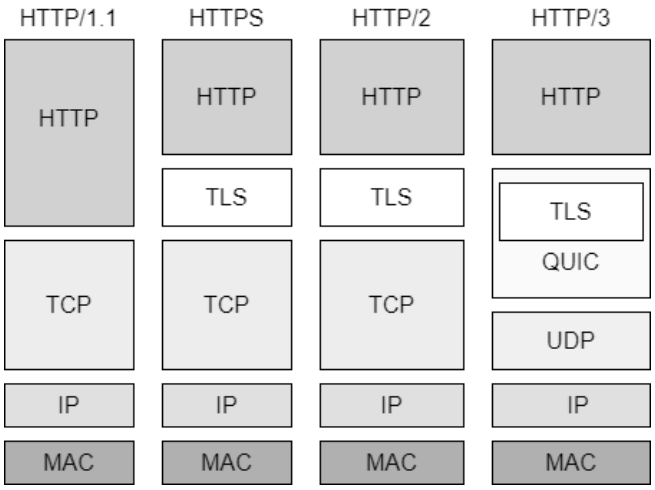

Рис. 1. Стеки протоколов, обеспечивающие работу версий HTTP

Анализ характеристик транспортных протоколов QUIC, TCP, UDP

Протокол TCP (Transmission Control Protocol) обеспечивает надежную передачу данных, разбивая информацию приложения на сегменты оптимального размера для отправки. Размер сегмента определяется MTU (Maximum Transmission Unit) сети. Процесс установления выделенного сквозного соединения в протоколе TCP известен как «трёхстороннее рукопожатие» (three-way handshake) и происходит по следующему алгоритму:

Клиент отправляет сегмент SYN (Synchronize), указывая свой порядковый номер (ISN (Initial Sequence Number)).

Сервер отвечает сегментом SYN ACK, в котором содержится его собственный ISN и подтверждение SYN клиента.

Клиент подтверждает сообщение сервера, отправляя ACK сегмент с ISN сервера, что завершает процесс установления соединения.

Когда TCP используется в сочетании с TLS (Transport Layer Security), процесс установления соединения включает дополнительные шаги для обеспечения безопасности. После завершения трёхстороннего рукопожатия TCP начинается процесс TLS-рукопожатия, который включает следующие этапы:

Клиент отправляет серверу сообщение "Client Hello", содержащее информацию о поддерживаемых версиях TLS, шифрах и случайных данных.

Сервер отвечает сообщением "Server Hello", выбирая шифр и версию TLS, которые будут использоваться, а также отправляет свой сертификат для аутентификации.

Происходит обмен данными для генерации сеансового ключа, который будет использоваться для шифрования данных.

Клиент и сервер подтверждают, что дальнейшие сообщения будут зашифрованы, и устанавливают защищённое соединение, что рассмотрено в работах [7-10].

Процесс завершения соединения включает обмен сегментами FIN и ACK, где один конец отправляет сегмент FIN, чтобы сообщить о завершении передачи данных, а другой конец подтверждает это, отправляя ACK. При отправке сегмента TCP устанавливает таймер, ожидая подтверждение от получателя, если подтверждение не поступает вовремя, сегмент отправляется повторно. TCP гарантирует, что данные будут доставлены получателю в том же порядке, в котором они были отправлены, благодаря использованию последовательных номеров для каждого сегмента. Если сегменты приходят в неправильном порядке, они сохраняются в буфере до тех пор, пока не будет получен недостающий сегмент. Для отслеживания целостности данных протокол использует контрольную сумму. Если сегмент имеет недопустимую контрольную сумму, он отбрасывается, и подтверждение не отправляется. Каждое окончание TCP-соединения имеет ограниченное буферное пространство, и принимающий сервер позволяет отправлять столько данных, сколько может быть обработано, чтобы предотвратить переполнение буферов. Протокол TCP поддерживает управление потоком с помощью механизма скользящего окна (sliding window), который позволяет получателю сообщать отправителю, сколько данных он может принять без риска переполнения буфера. Благодаря своей надежности протокол TCP представляет собой основу всех версий протокола HTTP, предшествующих HTTP/3. Однако принципы работы протокола TCP имеют значительные недостатки, влияющие на его скорость и эффективность.

Из-за того, что протокол TCP не реализует мультиплексирование потоков на транспортном уровне, при передаче данных часто возникает так называемая «блокировка начала очереди». Хотя, начиная с HTTP/2 на прикладном уровне происходит разделение HTTP-сообщений на отдельные потоки, на транспортном уровне, где работает TCP, отправка сообщений происходит посредством передачи одного потока байтов. Получая пакеты, протокол TCP разбивает их на сегменты. При потере пакета данных TCP повторно отправляет весь сегмент, то есть этот пакет и все последующие, и даже если последующие байты успешно доставлены и относятся к независимому HTTP-запросу, они не могут быть переданы приложению до восстановления потерянных данных. Это приводит к значительной задержке из-за неспособности протокола определить, возможна ли обработка полученных данных без недостающих байтов. Соединение по протоколу TCP происходит посредством «трехэтапного рукопожатия», а в современном мире, где безопасность передачи данных является одной из ключевых потребностей каждого пользователя Интернета и защищенный протокол HTTPS используется в подавляющем большинстве веб-приложений, чаще всего после TCP-рукопожатие происходит подключение криптографического протокола TLS, которое требует времени для отдельного «TLS-рукопожатия». Такой процесс создает проблему значительного увеличения задержки при установлении соединения.

UDP (User Datagram Protocol) — это простой протокол транспортного уровня, ориентированный на дейтаграммы – единицы данных, передаваемые по протоколу UDP, содержащие заголовок и полезную нагрузку. Каждый запрос создает одну дейтаграмму UDP, которая приводит к отправке одной IP-дейтаграммы. Протокол ориентирован на транзакции и не гарантирует доставку, защиту от дублирования, исправление ошибок, упорядочивание и предотвращение перегрузок. В отличие от TCP, UDP не устанавливает выделенного соединения между отправителем и получателем. Дейтаграммы отправляются без подтверждения состояния получателя, что делает протокол уязвимым к потере данных, дублированию пакетов и нарушению порядка доставки. При превышении MTU сети происходит фрагментация. Кроме того, UDP не имеет встроенных механизмов безопасности.

TCP продолжает оставаться доминирующим протоколом для большинства веб-приложений, использующих протоколы HTTP/2 и HTTPS, обеспечивая надежную передачу данных. UDP ввиду своей низкой надежности и быстрой скорости, используется в приложениях, низкая задержка для которых является приоритетным качеством передачи: сервисы потокового видео, онлайн-игры, технологии Интернета вещей, формирования кластеров [11-14, 23-31].

В целях решения технических проблем основного транспортного протокола Web-приложений – TCP, компания Google в 2012 году начала работу над экспериментальным протоколом QUIC (Quick UDP Internet Connections) для внедрения в HTTP, работающим поверх протокола UDP. В 2021 протокол был стандартизирован IETF. Решая проблемы низкой надежности UDP, QUIC представляет механизмы, работающие на основе UDP и гарантирующие соединение, получение данных, безопасность, контроль перегрузки и обнаружение потерь.

Протокол QUIC, в отличие от TCP, который использует единую последовательную передачу байтов, мультиплексирует запросы и ответы по одному соединению, предоставляя каждому из них собственный поток, пронумерованный уникальным идентификатором, что исключает взаимную блокировку потоков. Технология мультиплексирования QUIC, которая подразумевает независимую передачу потоков на транспортном уровне, решает еще одну значимую проблему протокола TCP – «блокировку начала очереди». Благодаря тому, что потоки QUIC передаются независимо друг от друга и идентифицируются с помощью уникальных номеров, потеря пакетов в одном потоке не влияет на обработку данных в других потоках [14].

Для формирования постоянного соединения QUIC использует пару 64-битных идентификаторов каждой из сторон. Использование этих идентификаторов служит гарантией сохранения соединения даже в случае изменения IP-адресации и номеров портов. Этот механизм называется миграцией соединения [15]. В целях обеспечения безопасной передачи данных QUIC по умолчанию обеспечивает сквозное шифрование с помощью TLS, однако в отличие от TCP-соединения, которое требует отдельного «рукопожатия» TLS после завершения TCP-«рукопожатия», процесс соединения QUIC сочетает в себе согласование криптографических и транспортных параметров, что позволяет клиентам отправлять данные сразу после установления соединения благодаря возможности 0-RTT (zero round trip time resumption, нулевое время приёма-передачи).

Передача данных между конечными точками происходит с помощью пакетов QUIC, содержащих последовательность фреймов, которые передают контрольную информацию и данные приложений. Заголовок пакета QUIC содержит метаданные, необходимые для управления соединением: Connection ID – идентификатор соединения, Packet Number – номер пакета, Flags – флаги, указывающие, например, тип пакета. QUIC инкапсулирует свои пакеты в дейтаграммы UDP, то есть добавляется в пакет UDP в качестве полезной нагрузки.

Пакеты QUIC шифруются отдельно, что позволяет их расшифровывать без ожидания доставки всех пакетов, в отличие от протокола TCP, у которого такая возможность отсутствует.

QUIC обеспечивает надёжную доставку и контроль перегрузки, используя алгоритмы обнаружения потерь и восстановления. Восстановление потерянных данных в TCP основано на порядковых номерах байтов, что может привести к неоднозначности при повторной передаче сегментов, так как получатель не в состоянии отличить оригинальные пакеты от повторных. QUIC решает эту проблему, присваивая каждому пакету уникальный номер, даже при повторной передаче, что упрощает обработку потерь. Восстановление потерянных пакетов осуществляется путем помещения всех фреймов пакета в новый пакет, которому присвоен новый идентификатор [16-18].

Для того, чтобы проиллюстрировать эффективность мультиплексирования независимых потоков и «составного рукопожатия» QUIC в сравнении с процессом подключения по протоколу TCP и передачей данных путем одного потока байтов, сравним работу обоих протоколов на примере следующего сценария: подключение между клиентом и сервером происходит впервые, после подключения сервер осуществляет передачу трех пакетов данных, в ходе которой первый пакет оказывается потерянным. В качестве стандартного Round-Trip Time (времени, за которое запрос доходит от клиента до сервера) возьмем примем 50 мс – усредненное значение, которое считается приемлемым для большинства пользователей.

Алгоритм соединения и передачи пакетов данных по протоколу TCP (рис. 2).

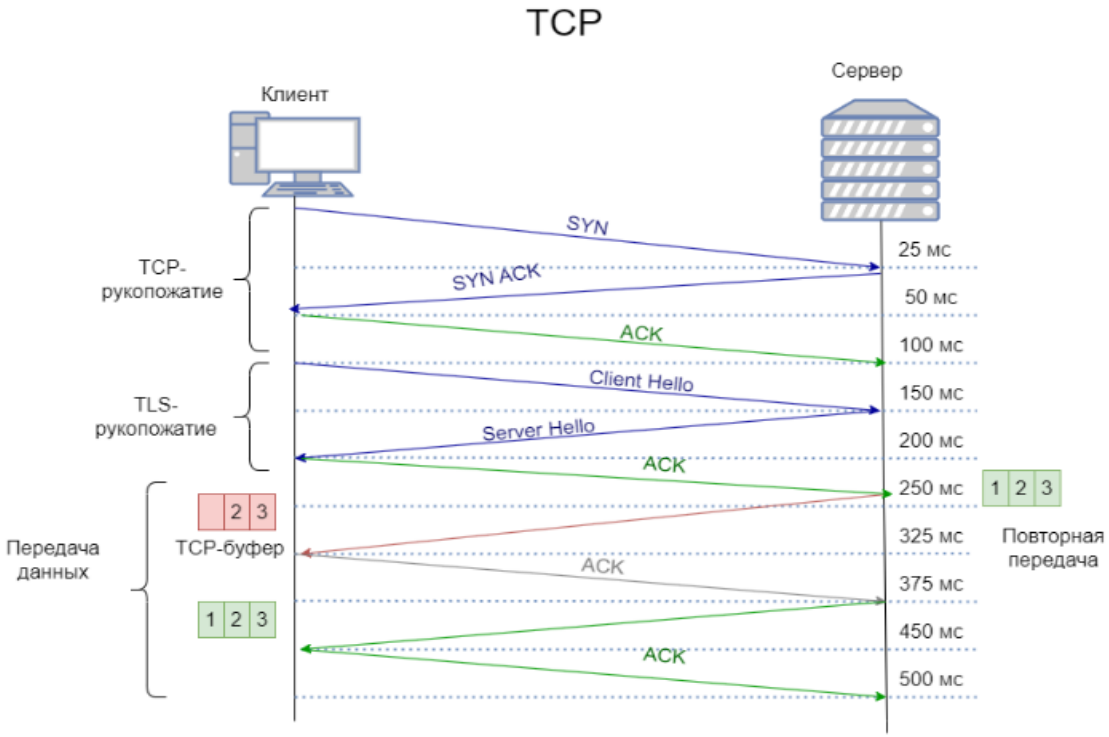

Рис. 2. Схема передачи данных по протоколу TCP

1. Клиент → Сервер (SYN). Клиент отправляет пакет с флагом SYN (synchronize), чтобы инициировать соединение. В пакете указывается начальный номер последовательности (ISN). Время: 25 мс.

2. Сервер → Клиент (SYN-ACK). Сервер отвечает пакетом с флагами SYN и ACK (acknowledge). SYN подтверждает получение запроса клиента, а ACK подтверждает готовность сервера к соединению. Сервер также указывает свой ISN. Время: 50 мс.

3. Клиент → Сервер (ACK). Клиент отправляет пакет с флагом ACK, подтверждая получение ответа сервера. Соединение TCP установлено. Время: 100 мс.

4. Клиент → Сервер (ClientHello). Клиент отправляет сообщение ClientHello, указывая поддерживаемые версии TLS, алгоритмы шифрования и случайное число. Время: 150 мс.

5. Сервер → Клиент (ServerHello, Certificate, ServerHelloDone). Сервер отправляет клиенту версию TLS, сертификат сервера для аутентификации. Время: 200 мс.

6. Клиент → Сервер (ACK). Клиент отправляет серверу подтверждение. TLS-рукопожатие завершается. Время: 250 мс.

После успешного процесса соединения начинается процесс передачи данных. Сервер последовательно отправляет три пакета данных в TCP-сегменте. Время, затраченное на передачу трех пакетов данных: 25 мс * 3 = 75 мс..

7. Сервер → Клиент (Пакеты 1,2,3). Сервер передает три пакета данных клиенту. Время: 325 мс.

В ходе передачи происходит потеря первого пакета данных. Клиент ожидает пакет 1, но получает пакет 2 и пакет 3. Поскольку пакет 1 отсутствует, клиент не может подтвердить получение пакетов 2 и 3. Происходит «блокировка начала очереди».

8. Клиент → Сервер (ACK). Клиент отправляет дублирующий ACK для пакета 1, указывая, что ожидает именно его. Время: 375 мс..

9. Сервер → Клиент (Повторная передача пакетов 1,2,3). Сервер повторно отправляет пакет 1, так как он был потерян, отправляет пакеты 2 и 3, так как они были отправлены после потерянного пакета 1. Время: 450 мс.

10. Клиент → Сервер (ACK). Клиент отправляет подтверждение получения пакетов. Время: 500 мс.

Алгоритм соединения и передачи данных по протоколу QUIC (рис. 3).

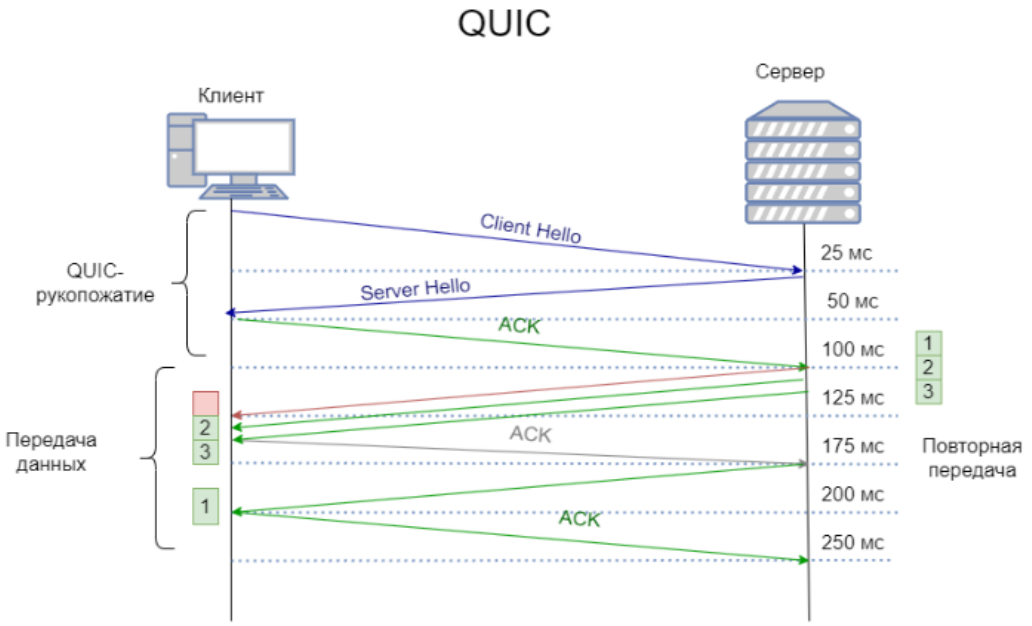

Рис. 3. Схема передачи данных по протоколу QUIC

1. Клиент → Сервер (ClientHello). Клиент отправляет начальный пакет с информацией для установления соединения (идентификатор соединения) и данные для TLS-рукопожатия (версию TLS и случайное число). Время: 25 мс.

2. Сервер → Клиент (ServerHello). Сервер отправляет идентификатор соединения, версию TLS, сертификат сервера для аутентификации. Время: 50 мс.

3. Клиент → Сервер (ACK). Клиент отправляет подтверждение получения ответа от сервера. Соединение установлено. Время: 100 мс. (Впоследствии, повторное соединение будет происходить мгновенно за счет сохраненных параметров сессии благодаря технологии Zero RTT).

После успешного соединения начинается процесс передачи данных. Сервер отправляет пакеты 1, 2 и 3 данных с помощью технологии мультиплексирования по трём независимым потокам.

4. Сервер → Клиент (Пакеты 1, 2 и 3). Сервер передает три пакета данных клиенту. Каждый пакет находится в отдельном независимом потоке. Время: 125 мс.

В ходе передачи происходит потеря первого пакета данных. Клиент получает пакеты 2 и 3, которые находятся в независимых потоках. Поскольку потоки независимы, клиент может обработать пакеты 2 и 3, даже если пакет 1 потерян.

5. Клиент → Сервер (ACK). Клиент отправляет подтверждение с уведомлением об утере пакета 1. Время: 175 мс.

6. Сервер → Клиент (Повторная передача пакета 1) Сервер повторно отправляет пакет 1. Время: 200 мс.

7. Клиент → Сервер (ACK). Клиент подтверждает получение пакета 1. Время: 250 мс.

На основании приведенных сведений о рассмотренных протоколах проведем анализ ключевых характеристик протоколов, сопоставив их в таблице 1.

Таблица 1

Сравнение характеристик протоколов TCP, UDP и QUIC

Характеристика	**TCP**	**UDP**	**QUIC**
Протокол нижнего уровня	IP	IP	UDP
Надёжность	Гарантирует доставку данных (подтверждение получения, повторная передача)	Не гарантирует доставку данных (нет подтверждения получения)	Гарантирует доставку данных (подтверждение получения, повторная передача)
Мультиплексирование	Нет	Нет	Да. Мультиплексирует запросы и ответы по независимым потокам
Блокировка начала очереди	Да. Потеря одного пакета блокирует обработку последующих	Нет	Нет. Потеря пакета в одном потоке не влияет на другие потоки
Идентификация соединения	Использует IP-адрес и порт	Нет соединения	Использует 64-битные идентификаторы соединения
Миграция соединения	Нет. Изменение IP или порта разрывает соединение.	Нет соединения	Да. Поддерживает миграцию соединения при изменении IP или порта.
Шифрование	Опционально. TLS требует отдельного «рукопожатия»	Нет	Обязательное сквозное шифрование
Аутентификация	Опционально	Нет	Обязательная аутентификация
Установление соединения	Требует трёхэтапного TCP-«рукопожатия» и отдельного TLS-«рукопожатия» (опционально)	Нет соединения	Объединённое «рукопожатие»
Задержка соединения	Высокая из-за раздельного TCP и TLS «рукопожатия»	Нет	Низкая благодаря 0-RTT и объединённому «рукопожатию»
Формат пакетов	Сегменты TCP с порядковыми номерах байтов	Дейтаграммы UDP без порядковых номеров	Пакеты QUIC с уникальными номерами пакетов и фреймами
Обнаружение и восстановление потерь	Основано на порядковых номерах байтов, что может вызывать неоднозначность	Нет	Каждый пакет имеет уникальный номер, даже при повторной передаче
Совместимость с HTTP	HTTP/1.1 и HTTP/2	Нет	HTTP/3
Производительность в нестабильных сетях	Низкая из-за блокировки потоков и строгой последовательности	Высокая, но без гарантии доставки	Высокая благодаря независимым потокам и устойчивости к потерям
Использование в реальных приложениях	Веб-сайты, электронная почта, файлообмен	IoT, видеотрансляции, онлайн-игры	HTTP/3, современные CDN, Google-сервисы

Заключение

На основе анализа процессов установления соединения и передачи данных по протоколам TCP и QUIC можно сделать следующие выводы. TCP требует отдельного выполнения TCP-рукопожатия и TLS-рукопожатия, что занимает 250 мс, тогда как QUIC объединяет установление соединения и TLS-рукопожатие в один этап, сокращая время до 100 мс. При повторном соединении QUIC может использовать 0-RTT, что делает процесс практически мгновенным. В TCP данные передаются в рамках одного потока байтов, что приводит к «блокировке начала очереди» при потере пакета.

В данном сценарии потеря первого пакета вынуждает сервер повторно передавать все три пакета, увеличивая общее время передачи до 500 мс. QUIC поддерживает мультиплексирование независимых потоков, что позволяет клиенту обрабатывать пакеты 2 и 3, обрабатываемые в разных потоках, даже если пакет 1 потерян. Потеря пакета в одном потоке не блокирует обработку данных в других потоках. Время передачи и восстановления данных в QUIC составляет всего 250 мс.

Таким образом, протокол QUIC обеспечивает более быстрое установление соединения, эффективную передачу данных и устойчивость к потерям пакетов благодаря своей современной архитектуре. В то время как TCP страдает от блокировки очереди и более длительных задержек.

HTTP/3 имеет относительно немного отличий от предшествующей ему версии. Основной причиной его создания стала необходимость реализации главного веб-протокола поверх принципиально нового QUIC. Одной из важнейших причин создания новой версии HTTP стала проблема схемы сжатия заголовков HTTP/2 – HPACK. Этот механизм использует последовательно обновляющуюся таблицу заголовков и требует строгого порядка доставки запросов и ответов. В HTTP/2 он контролируется протоколом TCP, который передает единственный поток последовательных байтов и гарантирует их последовательное получение. В отличие от TCP, протокол QUIC гарантирует последовательность доставки байтов только в пределах одного потока. Потоки передаются независимо друг от друга и контроль за их последовательностью не осуществляется.

Для решения этой проблемы в HTTP/3 была реализована новая схема сжатия заголовков – QPACK, в которой присутствует механизм разделения процессов сжатия заголовков и обновления динамической таблицы за счет передачи обновлений через отдельные потоки управления. Это позволяет избежать зависимости динамической таблицы заголовков от порядка доставки потоков. [19] Помимо описанной, существует также следующая причина необходимости создания протокола HTTP/3: использование HTTP/2 было бы нецелесообразным по причине того, что QUIC переносит с прикладного уровня на транспортный значительную часть функций, которые до этого были реализованы непосредственно в HTTP/2. QUIC интегрирует большинство функций протокола безопасности TLS, включая шифрование и аутентификацию, что устраняет необходимость отдельного «рукопожатия» TLS после установления транспортного соединения.

QUIC реализует мультиплексирование соединения на транспортном уровне, что позволяет передавать несколько независимых потоков данных в рамках одного соединения без взаимной блокировки. В отличие от HTTP/2, где мультиплексирование реализовано на прикладном уровне, QUIC обеспечивает более эффективную обработку потоков, устраняя проблему блокировки начала очереди (Head-of-Line Blocking) на транспортном уровне. Таким образом можно заключить, что HTTP/3 оптимизирован для работы с QUIC, обеспечивая более высокую производительность, устраняя ограничения предыдущих версий и повышая надежность соединения [20-22].

Литература

1. *Гадасин Д.В., Шведов А.В*. Применение транспортной задачи для балансировки нагрузки в условиях нечеткости исходных данных // T-Comm: Телекоммуникации и транспорт. 2024. Т. 18, № 1. С. 13-20. DOI 10.36724/2072-8735-2024-18-1-13-20. EDN: WKNPIX

2. *Гадасин Д.В., Шустов С.А*. Исследование эффективности протоколов маршрутизации в условиях сетей с высокой нагрузкой // Теория и практика экономики и предпринимательства: Труды XXI Международной научно-практической конференции, Симферополь – Гурзуф, 18-20 апреля 2024 года. Симферополь: ИП Зуева Т. В., 2024. С. 236-237. EDN: WNVEOC

3. Tremasova L.A., Andriyanova A.K., Gadasin D.V., Gadasin D.D. Modeling and Solving the Problem of Load Balancing in Data Transmission Networks Using the Stepping Stone Method // 2024 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, Russia, 2024, pp. 1-7, doi: 10.1109/IEEECONF60226.2024.10496718.

4. *Shvedov A.V., Gadasin D.V., Klygina O.G., Tremasova L.A*. Optimization of Network Routing Using the Markov Decision Process and Hamiltonian Cycle // 2023 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, Russian Federation, 2023, pp. 1-4, doi: 10.1109/IEEECONF56737.2023.10091989.

5. RFC 2616: Hypertext Transfer Protocol – HTTP/1.1. 1999, URL: https://datatracker.ietf.org/doc/html/rfc2616

6. RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3. 2018, URL: https://datatracker.ietf.org/doc/html/rfc8446

7. *Гадасин Д.В., Веденеев П.С., Шведов А.В*. Уязвимости системы маршрутизации глобальной сети Интернет и возможные пути их преодоления // Перспективные технологии в средствах передачи информации – ПТСПИ-2019: Материалы XIII международной научно-технической конференции. В 2-х томах, Владимир, 03-05 июля 2019 г. Том 1. Владимир: Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых, 2019. С. 94-96. EDN: YFEIAH

8. *Кузин И.А., Гадасин Д.В*. Модель контейнера данных для минимизации трафика при передаче субъективных характеристик объектов на изображении трехмерной сцены // Телекоммуникации и информационные технологии. – 2021. Т. 8, № 2. С. 96-100. EDN: TYFFBH

9. *Шведов А.В., Гадасин Д.В., Клыгина О.Г., Тремасова Л.А*. Оптимизация маршрутизации в сети при помощи гамильтонова цикла и марковского процесса принятия решений // DSPA: Вопросы применения цифровой обработки сигналов. 2023. Т. 13, № 3. С. 42-49. EDN: BSWDEQ

10. *Золотарева П.Ю., Гадасин Д.В., Маклачков К.А*. Методы обработки информации в распределенных информационных системах // Тенденции развития Интернет и цифровой экономики: Труды VI Международной научно-практической конференции, Симферополь-Алушта, 01-03 июня 2023 г. Симферополь: ИП Зуева, 2023. С. 187-189. EDN: LGONZK

11. *Мелькова Е.К., Шведов А.В., Тремасова Л.А., Гадасин Д.В.* Организация кластера исходя из функции принадлежности // Системы синхронизации, формирования и обработки сигналов. 2023. Т. 14, № 1. С. 30-39. EDN: CNVIJU

12. *Гадасин Д.В., Золотарева П.Ю., Тремасова Л.А*. Влияние кластеризации при обработке сырых данных // Системы синхронизации, формирования и обработки сигналов. 2024. Т. 15, № 3. С. 10-19. EDN: JQIPHX

13. *Тремасова Л.А., Первухина А.А., Гадасин Д.В*. Использование методов Косарайю и k-средних для формирования кластеров // Электросвязь. 2024. № 9. С. 47-55. DOI 10.34832/ELSV.2024.58.9.007. EDN: DOZTZK

14. *Гадасин Д.В.* Построение бинарного дерева минимальной цены // T-Comm: Телекоммуникации и транспорт. – 2024. Т. 18, № 11. С. 38-44. DOI 10.36724/2072-8735-2024-18-11-38-44. EDN: GMCEWG

15. Докучаев В.А., Лопатина Е.В., Павлов С.В., Гадасин Д.В. Качество передачи информации в корпоративных IP-сетях (часть 1). М.: Московский технический университет связи и информатики, Инсвязьиздат, 2010. 36 с. EDN: ZGJSJH

16. *Гадасин Д.В., Юдина А.А*. Сложные сети как симбиоз современных сетевых технологий и искусственного интеллекта // Технологии информационного общества: Сборник трудов XIV Международной отраслевой научно-технической конференции, Москва, 18-19 марта 2020 г. М.: Издательский дом Медиа Паблишер, 2020. С. 270-272. EDN: FHRMNZ

17. *Гадасин Д.В., Пантелеева К.А., Маклачков К.А*. Разработка единой точки входа сообщений о пользовательском негативном опыте взаимодействия с web-сервисами // Искусственный интеллект в автоматизированных системах управления и обработки данных: Сборник статей II Всероссийской научной конференции. В 5-ти томах, Москва, 27-28 апреля 2023 г. М.: Издательский дом КДУ, "Добросвет", 2024. С. 413-417. EDN: ADRGFV

18. *Пантелеева К.А., Палибза С.А., Гадасин Д.В*. Принципы построения системы управления при возникновении сбоев в ит-инфраструктуре // REDS: Телекоммуникационные устройства и системы. 2024. Т. 14, № 2. С. 24-34. EDN: MOYCNG

19. A QUICK Introduction to HTTP/3. 2023, URL: https://www.akamai.com/blog/developers/a-quick-introduction-http3 (дата обращения: 22.12.2024).

20. *Гадасин В.А., Гадасин Д.В.* Надежность крупномасштабных сетей связи с аддитивной структурой // Автоматика и телемеханика. 1997. № 1. С. 160.

21. *Гадасин В.А., Гадасин Д.В.* Надежность двухполюсных сетей с аддитивной структурой II. Финальная вероятность связи // Автоматика и телемеханика. 1999. № 10. С. 164-179. EDN: OKEMTZ

22. *Яковенко Н.В., Гадасин Д.В., Коцич Л*. Повышение точности коэффициента влияния ошибок в информационных системах с применением метода обратного распространения ошибки // Системы синхронизации, формирования и обработки сигналов. 2024. Т. 15, № 4. С. 35-42. EDN: CMFVNH

23. *Гадасин Д.В., Шведов А.В., Кузин И.А.* Трехмерная реконструкции объекта по одному изображению с использованием глубоких свёрточных нейронных сетей // T-Comm: Телекоммуникации и транспорт. 2022. Т. 16, № 7. С. 29-35. DOI: 10.36724/2072-8735-2022-16-7-29-35. EDN: YTLCNW

24. *Shvedov A.V., Gadasin D.V., Alyoshintsev A.V.* Segment routing in data transmission networks // T-Comm. 2022. Vol. 16. No. 5, pp. 56-62. DOI: 10.36724/2072-8735-2022-16-5-56-62. EDN: VAYLJQ

25. *Назаров М.Д., Шведов А.В.* Корреляция атрибутов соглашения об уровне обслуживания с основными параметрами QoS в корпоративных сетях // Телекоммуникации и информационные технологии. 2020. Т. 7. № 2. С. 73-79. EDN: VQHDTJ

26. *Kalmykov N.S., Dokuchaev V.A.* Segment routing as a basis for software defined network // T-Comm. 2021. Т. 15. № 7. С. 50-54. EDN: LYVZCV

27. *Dokuchaev V.A., Maklachkova V.V., Statev V.Yu.* Classification of personal data security threats in information systems // T-Comm. 2020. Т. 14. № 1. С. 56-60. EDN: QOGYHH

28. *Докучаев В.А., Маклачкова В.В., Статьев В.Ю.* Цифровизация субъекта персональных данных // T-Comm: Телекоммуникации и транспорт. 2020. Т. 14. № 6. С. 27-32. EDN: XVWYJP

29. *Pavlov S.V., Dokuchaev V.A., Mytenkov S.S.* Model of a fuzzy dynamic decision support system // T-Comm. 2020. Т. 14. № 9. С. 43-47. EDN: VYFNLB

30. *Гадасин Д.В*., *Тремасова Л.А., Гадасин Д.Д.* Распределение поступающей нагрузки с применением SS-метода // I-methods. 2023. Т. 15, № 3. EDN: HQEYTW

31. *Гадасин Д.В., Бессолицын А.Д., Гадасин Д.Д*. Оценка качества данных информационных систем // DSPA: Вопросы применения цифровой обработки сигналов. 2024. Т. 14, № 2. С. 4-12. EDN: GYIWJU.

ЭТАПЫ ПЕРЕХОДА ОРГАНИЗАЦИИ С ЛОКАЛЬНЫХ СЕРВЕРОВ НА ОБЛАЧНЫЕ РЕШЕНИЯ

Левин Алексей Михайлович
Московский технический университет связи и информатики, Москва, Россия
a.m.levin@edu.mtuci.ru

Ванина Маргарита Федоровна
Московский технический университет связи и информатики, доцент, к.т.н., доцент, Москва, Россия
margo.vanina2012@yandex.ru

Аннотация
В условиях стремительного развития цифровых технологий и растущей зависимости бизнеса от IT-инфраструктуры, переход с локальных серверов на облачные решения становится для компаний стратегически важным шагом. Компании, работающие в сфере высокотехнологичных разработок, сталкивается с вызовами, связанными с ограничениями локальной инфраструктуры, необходимостью повышения производительности и оптимизации затрат. Современные облачные решения предлагают встроенные механизмы резервного копирования, репликации данных и защиты от сбоев. Это минимизирует риски потери данных и гарантирует бесперебойную работу системы. Это особенно важно для организаций, работающих в сфере высокотехнологичных разработок, где нагрузка на инфраструктуру может значительно варьироваться. ООО "Нано Лабс", как компания, работающая в сфере высокотехнологичных разработок, сталкивается с постоянным ростом требований к своей IT-инфраструктуре. В данной работе рассматривается алгоритм перехода организации с собственных локальных серверов на облачные решения.

Ключевые слова
Облачные серверы, cloud, серверы, виртуальные машины, дата-центр, ЦОД, центр обработки данных

Введение

Использование локальных серверов, которые до недавнего времени считались стандартным решением, больше не отвечает текущим требованиям компании. Эти системы обладают рядом ограничений, которые негативно влияют на эффективность работы и развитие бизнеса. Локальные серверы имеют фиксированный объём ресурсов, увеличение вычислительных мощностей требует покупки нового оборудования, что приводит к значительным финансовым затратам и увеличению времени на интеграцию новых серверов. Решением является внедрение облачных технологий. Облачные технологии предлагают инновационную альтернативу традиционным локальным системам [1].

Основная цель перехода ООО "Нано Лабс" на облачные серверы – модернизация IT-инфраструктуры для соответствия требованиям современной цифровой экономики. В рамках этого процесса преследуются следующие цели:

- Оптимизация ресурсов. Облачная инфраструктура позволяет оптимально использовать вычислительные мощности без необходимости содержать избыточные ресурсы.
- Улучшение управляемости системы. Централизованное управление облачной средой упрощает контроль над данными, распределение ресурсов и настройку приложений.
- Снижение затрат. Отказ от покупки оборудования и затрат на его обслуживание позволяет значительно снизить расходы компании.
- Обеспечение гибкости и скорости развития. Возможность быстро внедрять новые сервисы и расширять инфраструктуру под новые задачи.

Алгоритм перехода на облачные решения

ООО "Нано Лабс", как компания, работающая в сфере высокотехнологичных разработок, сталкивается с постоянным ростом требований к своей IT-инфраструктуре. В данной работе рассматривается алгоритм перехода организации с собственных локальных серверов на облачные решения (рис. 1).

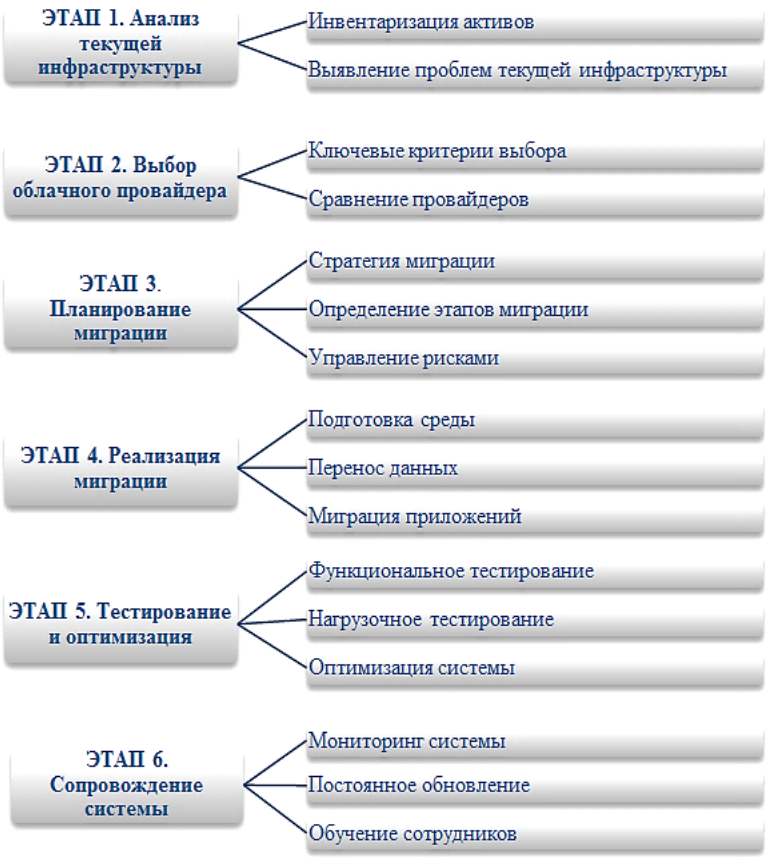

Рис. 1. Алгоритм перехода организации на облачные решения

Этап 1. Анализ текущей инфраструктуры

На этом этапе формируются технические и бизнес-требования, которым должна соответствовать новая IT-инфраструктура. Анализ существующей IT-инфраструктуры является важнейшим шагом в процессе подготовки к переходу с локальных серверов на облачные. Этот этап помогает компании ООО "Нано Лабс" не только понять текущее состояние своих IT-ресурсов, но и выявить их слабые стороны, которые препятствуют масштабированию и повышению эффективности [2].

Инвентаризация активов – это процесс, позволяющий определить все ресурсы, которые используются компанией. Он включает как физическое оборудование, так и программное обеспечение. Цель инвентаризации заключается в создании полного и точного списка ресурсов, который позволит оценить их производительность и соответствие текущим требованиям компании. Грамотно проведённый анализ становится фундаментом для последующих этапов миграции.

Для анализа аппаратного обеспечения необходимо составить детальный список всех физических устройств, которые составляют IT-инфраструктуру компании: серверы, системы хранения данных (NAS, SAN), тип носителей (SSD, HDD), сетевое оборудование.

В таблице 1 показан состав текущего аппаратного обеспечения организации.

Таблица 1

Анализ аппаратного обеспечения

Модуль	Описание	Кол-во
Серверный корпус	Корпус для размещения серверных компонентов, совместимый с серверными стойками	1
Материнская плата	Основная плата, поддерживающая процессоры серверного класса и память ECC	1
Процессор (CPU)	Вычислительный модуль, обеспечивающий выполнение задач	2
Оперативная память (RAM)	Память для обработки данных, с поддержкой технологий коррекции ошибок (ECC)	8
Системы хранения данных (HDD/SSD)	Хранилища данных для системных файлов и данных пользователей	4

Сетевые интерфейсные карты (NIC)	Устройства для подключения серверов к локальной сети или интернету	2
Системы охлаждения	Вентиляторы или жидкостные системы для поддержания температуры компонентов	5
Блок питания	Обеспечивает энергоснабжение всех компонентов, с запасом по мощности	1
RAID-контроллер	Контроллер для управления дисковыми массивами и конфигурациями RAID	1
Мониторинг оборудования (IPMI/DRAC)	Модули для удалённого управления сервером и мониторинга состояния	1
Источник бесперебойного питания (UPS)	Устройство для предотвращения отключения при перебоях электроэнергии	1
Резервные накопители (NAS/DAS)	Резервные хранилища для создания копий данных	2
Коммутаторы для локальной сети	Устройства для объединения серверов в локальную сеть	1
Кабели (питание, сеть)	Кабели для подключения питания и сетевых интерфейсов	16

Не менее актуальным является аудит всех используемых программных решений: операционные системы; СУБД (MySQL, PostgreSQL, Microsoft SQL Server), их использование и производительность; бизнес-приложения, включая CRM, ERP и системы аналитики; резервное копирование, например, технологии и инструменты Veeam, Acronis.

Важно собрать информацию о текущих затратах на обслуживание инфраструктуры. В таблице 2 представлены финансовые затраты на поддержание локальных серверов организации.

Таблица 2

Финансовые затраты на поддержание локальных серверов

Категория затрат	**Годовые затраты, руб.**
Покупка серверного оборудования	1 200 000
Обновление систем хранения данных	400 000
Сетевое оборудование	250 000
Лицензии на программное обеспечение	300 000
Электроэнергия	600 000
Охлаждение серверов	200 000
Резервное копирование	150 000
Зарплаты IT-персонала	900 000
Техническое обслуживание оборудования	500 000
Итог	4 500 000

Анализ инфраструктуры также включает выявление её слабых мест. Это необходимо для понимания текущих ограничений, которые следует устранить при переходе на облачные серверы:

- Ограничения масштабируемости. Локальные серверы имеют ограниченные ресурсы, что не позволяет быстро реагировать на растущие потребности бизнеса. Масштабирование возможно только путём покупки нового оборудования, что требует значительных финансовых затрат и времени.
- Высокие эксплуатационные затраты. Затраты на электроэнергию и охлаждение составляют значительную часть бюджета IT-инфраструктуры. Обслуживание серверов требует участия квалифицированных специалистов, что увеличивает расходы.
- Риски физического повреждения оборудования. Локальная инфраструктура подвержена рискам, связанным с природными катаклизмами, пожарами, отключением электроэнергии.
- Ограниченная отказоустойчивость. Большинство локальных серверов не поддерживают автоматическое переключение на резервные системы в случае сбоя. Это может привести к длительным простоям и потере данных.
- Проблемы с безопасностью. Устаревшие протоколы и механизмы защиты увеличивают риск несанкционированного доступа к данным. Отсутствие современных инструментов мониторинга делает невозможным своевременное обнаружение угроз.

Анализ текущей инфраструктуры является основой для успешной реализации перехода на облачные серверы. Он позволяет компании ООО "Нано Лабс" определить, какие ресурсы используются, выявить их слабые стороны и сформулировать требования к новой системе. Грамотно проведённый анализ минимизи-

рует риски и обеспечивает эффективное использование ресурсов в дальнейшем. Результаты анализа формируют базу для выбора подходящего облачного провайдера и разработки плана миграции.

Этап 2. Выбор облачного провайдера

Выбор подходящего облачного провайдера является стратегическим шагом, от которого зависит успех перехода ООО "Нано Лабс" на облачные серверы. Провайдер должен не только удовлетворять текущие потребности компании, но и обеспечивать гибкость для будущего роста, соответствие стандартам безопасности и экономическую эффективность [3].

Ключевые критерии выбора. При выборе провайдера важно учитывать несколько основных факторов, которые определяют, насколько решение будет соответствовать техническим и бизнес-требованиям компании.

• *Облака* предоставляются крупными провайдерами, которые предлагают широкий спектр услуг, включая хранилище данных, вычислительные ресурсы и аналитические инструменты. Пример публичного облака: Yandex Cloud, ЦОД Ростелеком, Mail Cloud.

• Стоимость (сравнение тарифных планов). Провайдеры предлагают разные модели оплаты: Pay-as-you-go: (оплата только за используемые ресурсы); фиксированная стоимость (Пакеты с включёнными объёмами ресурсов). Если компания активно работает с большими объёмами данных, стоит учитывать стоимость хранения и передачи информации (CPU, RAM, объём хранилища, использование сетевых ресурсов). Скрытые расходы включают стоимость обучения сотрудников, интеграции и переноса данных.Например, переход на Yandex Cloud может потребовать перенастройки текущих приложений, что увеличивает затраты.

• *Безопасность*. Крупные провайдеры соответствуют государственным стандартам. Инструменты защиты данных включают многофакторную аутентификацию, мониторинг активности, автоматическое обнаружение угроз. Например, ЦОД Ростелеком обеспечивает анализ и предотвращение угроз в режиме реального времени. Важно учитывать, где физически будут храниться данные. Например, в России требуется соблюдение закона о персональных данных (ФЗ-152), что делает критичным выбор дата-центров в соответствующих регионах.

• *Уровень поддержки.* Техническая поддержка во многом связана с возможностью получения консультаций 24/7. Например, все рассмотренные поставщики предлагает уровни поддержки, включая базовый (для малого бизнеса) и корпоративный (с индивидуальным консультантом). Кастомизация решений связана с тем, что провайдеры предоставляют API для интеграции с собственными приложениями компании, например, использование API Yandex для настройки систем аналитики или автоматизации.

Сравнение провайдеров. После определения ключевых критериев важно провести сравнительный анализ различных облачных платформ, чтобы выбрать подходящее решение. (табл. 3).

Таблица 3

Сравнение поставщиков

Критерии	Ростелеком	Yandex Cloud	Mail Cloud
Стоимость	Высокая, гибкая модель	Средняя, адаптивные пакеты	Высокая, выгодная для больших объёмов данных
Масштабируемость	Очень высокая	Высокая	Высокая
Инструменты безопасности	Передовые технологии	Передовые технологии	Расширенные шифровальные механизмы
Поддержка разработчиков	Широкий выбор инструмента	Фокус на аналитике и машинном обучении	Гибкость для предприятий

• Ростелеком: подходит для крупных компаний с высокими требованиями к масштабируемости и доступу к современным технологиям. Однако стоимость может быть выше среднего.

• Yandex Cloud: универсальное решение, которое обеспечивает хорошую интеграцию с корпоративными системами и поддержку российских клиентов. Преимущество в обработке больших данных и аналитике

• Mail Cloud: универсальное решение, которое обеспечивает хорошую интеграцию с корпоративными системами и поддержку российских клиентов.

Выбор облачного провайдера для ООО "Нано Лабс" должен быть основан на тщательном анализе бизнес-потребностей и технических требований. Чёткое понимание критериев выбора и детальный анализ возможностей провайдеров позволяют минимизировать риски и обеспечить успешный переход на облачные серверы.

Этап 3. Планирование миграции

Планирование миграции IT-инфраструктуры – это стратегически важный этап, от которого зависит успешность перехода компании на облачные технологии. Для ООО "Нано Лабс" этот процесс включает разработку стратегии миграции, определение этапов её выполнения и управление рисками. Комплексный подход на стадии планирования позволяет минимизировать простои, избежать потери данных и обеспечить плавное функционирование всех бизнес-процессов в период перехода.

Выбор стратегии миграции зависит от ее целей, текущего состояния инфраструктуры и доступных ресурсов. Существует три основные стратегии, каждая из которых подходит для определённых сценариев [4].

• Lift-and-Shift (перенос без изменений). Эта стратегия предполагает перемещение существующих приложений и данных из локальной инфраструктуры в облако без внесения изменений в их архитектуру. Преимущества: быстрое выполнение миграции; минимальные изменения в существующих системах; снижение первоначальных затрат. Недостатки: возможные проблемы с производительностью, если приложения не оптимизированы для облака и ограниченные возможности использования облачных функций (например, автоматического масштабирования). Подходит для организаций, стремящихся быстро начать использовать облачные ресурсы, сохраняя при этом привычные рабочие процессы.

• Рефакторинг. Включает внесение изменений в приложения с целью их адаптации к облачной архитектуре. Это может включать обновление кода для использования микросервисов, контейнеров или серверлесс-технологий. Преимущества: улучшенная производительность в облачной среде и возможность использования передовых облачных функций (например, автоматизации и анализа данных). Недостатки: более длительный процесс миграции и необходимость наличия квалифицированных разработчиков. Подходит для приложений, которые будут интенсивно использовать возможности облака.

• Реархитектура. Предполагает полный пересмотр инфраструктуры компании с учётом возможностей облачных технологий. Это наиболее комплексный подход, требующий глубокого анализа и проектирования. Преимущества: максимальная эффективность и гибкость и полное соответствие требованиям бизнеса и возможностей облака. Недостатки: высокие начальные затраты и временные ресурсы; требует тщательной подготовки и управления проектом. Подходит для компаний, желающих построить IT-инфраструктуру «с нуля», максимально используя потенциал облаков.

Определение этапов миграции. Миграция – это пошаговый процесс, который должен быть структурирован и разбит на этапы для обеспечения прозрачности и управляемости. На начальном этапе миграции переносятся данные, которые не влияют на операционные процессы компании (например, архивы, резервные копии, второстепенные системы). Этот процесс позволяет протестировать процесс миграции и минимизирует риски, связанные с потерей данных [5].

Миграция рабочих нагрузок. После успешного переноса некритичных данных начинается перенос рабочих нагрузок, таких как приложения, которые поддерживают основные бизнес-процессы (например, внутренние порталы сотрудников, системы аналитики).

Перенос систем с высокими требованиями к доступности. На последнем этапе мигрируются системы, которые требуют высокой доступности и надёжности (например, базы данных; бизнес-приложения, такие как CRM и ERP).

Управление рисками. Переход на облачные технологии сопряжён с различными рисками, которые необходимо учитывать на этапе планирования. Эффективное управление рисками помогает минимизировать потенциальные проблемы и снизить затраты. Для этого важно обеспечить:

• Создание резервных копий. Перед началом миграции необходимо создать резервные копии всех данных.

• Разработать план действий в случае сбоев. Определить роли и обязанностей команды в случае возникновения чрезвычайных ситуаций.

• Постоянный мониторинг выполнения задач.

• Использование инструментов управления проектами для отслеживания статуса миграции.

• Управление зависимостями. Убедиться, что все зависимости приложений учтены, чтобы предотвратить сбои в работе после миграции. Например, приложения, зависящие от специфических конфигураций серверов, должны быть протестированы перед переходом.

Планирование миграции – это основополагающий этап, который требует тщательного анализа, чёткой структуры и внимательного управления. ООО "Нано Лабс" может использовать подход «Lift-and-Shift» для быстрого перехода или выбрать более сложные стратегии, такие как рефакторинг или реархитектура, для повышения эффективности и производительности в долгосрочной перспективе [6, 10-12].

Этап 4. Реализация миграции

Реализация миграции IT-инфраструктуры ООО "Нано Лабс" в облачную среду – это комплексный процесс, требующий чёткой организации. Этап включает подготовку среды, перенос данных и приложений, а также их интеграцию. На этом этапе важно соблюдать порядок действий, чтобы минимизировать риски и исключить потери данных.

Подготовка среды включает настройку инфраструктуры в облаке, чтобы обеспечить её соответствие требованиям компании. Это этап, где создаётся техническая база для размещения данных и приложений.

Настройка сетевого подключения включает*:*

- Конфигурация виртуальных сетей. Создание виртуальной частной сети (VPC), обеспечивающей изоляцию и безопасность. Определение подсетей, маршрутов и правил межсетевого взаимодействия.
- Управление доступом. Использование инструментов управления доступом, таких как AWS Identity and Access Management (IAM) или Azure Active Directory.
- Мониторинг и безопасность сети. Настройка брандмауэров, VPN и шифрования трафика для защиты данных. Подключение инструментов мониторинга сетевого трафика, таких как Azure Network Watcher или GCP Cloud Armor.

Создание виртуальных машин и контейнеров означает:

- Выбор подходящей платформы. Виртуальные машины: используются для приложений, которые требуют отдельной операционной системы (например, AWS EC2, Azure Virtual Machines). Контейнеры: Оптимальны для приложений, которые можно разделить на микросервисы (например, Docker, Kubernetes).
- Конфигурация ресурсов. Настройка количества процессоров, объёма оперативной памяти и хранилища для каждой виртуальной машины. Например, для интенсивных вычислений выделяются ресурсы с высокой производительностью CPU и большим объёмом RAM.
- Оптимизация стоимости. Использование предоплаченных или спотовых инстансов для экономии затрат.

Настройка сервисов в облаке означает интеграцию с облачными сервисами:

- Настройка баз данных (например, AWS RDS, Azure SQL Database).
- Подключение систем хранения данных (например, S3, Azure Blob Storage).
- Автоматизация процессов. Использование скриптов для автоматического развёртывания инфраструктуры. Например, использование Terraform для создания и управления облачными ресурсами.

Перенос данных – это ключевой этап, который включает перемещение всех файлов, баз данных и других объектов из локальной инфраструктуры в облако.

Использование инструментов миграции данных. Популярные инструменты для миграции:

- EaseUS Todo PCTrans. Простой в использовании инструмент для переноса данных, включая перенос файлов, приложений и настроек с одного компьютера на другой.
- Liquibase. Продукт на основе CLI, написанный на Java. В нём миграции схем организованы в виде наборов изменений и изменений журнала.
- AWS Migration Hub: позволяет управлять процессом миграции и отслеживать его статус.
- Azure Migrate: Инструмент для переноса данных и приложений в Azure.
- Google Transfer Service: используется для больших объёмов данных.

Проверка целостности данных. Контрольные проверки включают:

- Сравнение хэшей данных до и после переноса.
- Использование автоматизированных тестов для проверки целостности.
- Восстановление данных в случае ошибок
- Хранение резервных копий данных до завершения миграции.
- Проверка доступности и корректности всех файлов.

Миграция приложений требует особого внимания, так как от их корректной работы зависит эффективность бизнес-процессов компании, а именно:

Тестирование работоспособности приложений

- Проверка функциональности. Тестирование основных функций каждого приложения в облаке, например, Проверка работы CRM-системы после переноса.
- Оценка производительности. Измерение времени отклика и скорости обработки запросов.
- Настройка параметров виртуальных машин для повышения производительности.

Настройка интеграции между компонентами

- Связь между модулями. Настройка взаимодействия приложений с базами данных и другими сервисами. Например, подключение аналитической платформы к облачному хранилищу.

• Оптимизация работы. Использование API для автоматизации взаимодействия между компонентами.
• Настройка очередей сообщений (например, Amazon SQS или Google Pub/Sub) для передачи данных.

Этап 5. Тестирование и оптимизация

После завершения миграции IT-инфраструктуры в облачную среду, ООО "Нано Лабс" должна выполнить комплексное тестирование и оптимизацию системы. Эти действия являются ключевыми для обеспечения бесперебойной работы всех процессов, достижения требуемой производительности и снижения затрат.

Функциональное тестирование направлено на проверку корректности работы всех элементов системы в облачной среде. Этот этап помогает выявить несовместимости, ошибки конфигурации и проверить, насколько система соответствует бизнес-требованиям.

Проверка корректности работы всех систем включает:

• Тестирование процессов и операций. Проверяются ключевые процессы, такие как хранение данных, обработка запросов и их передача между системами. Например, в ERP-системе проверяется корректность формирования отчётов, учётных записей и данных о транзакциях.

• Тестирование интеграции. Проверяется взаимодействие между различными компонентами: базами данных, приложениями и аналитическими инструментами. Например, CRM должна корректно взаимодействовать с облачным хранилищем для загрузки и обновления данных.

• Проверка безопасности. Убеждаются, что все политики безопасности работают корректно, включая доступ к данным, шифрование и управление учетными записями.

Обнаружение ошибок включает:

• Идентификацию проблем. Проводится поиск ошибок, связанных с неправильной конфигурацией ресурсов или несовместимостью приложений. Например, ошибка в конфигурации виртуальной машины может привести к медленной работе приложения.

• Исправление и повторное тестирование. После устранения ошибок проводятся повторные тесты.

Нагрузочное тестирование проверяет способность системы выдерживать большие объёмы данных и одновременные запросы. Этот этап помогает понять пределы производительности системы и определить узкие места. Нагрузочное тестирование включает:

Проверку работы при высокой нагрузке

• Создание сценариев нагрузки. Определяются сценарии, которые имитируют пиковые нагрузки, например, симуляция 10,000 запросов к базе данных за короткий период времени.

• Использование инструментов тестирования. Применяются инструменты, такие как Apache JMeter, LoadRunner или Gatling, для создания искусственной нагрузки.

• Мониторинг системы в условиях нагрузки. Отслеживаются показатели производительности, такие как время отклика, использование CPU и памяти.

Выявление узких мест

• Анализ производительности компонентов. Изучается работа каждого компонента, чтобы выявить элементы, которые создают задержки или сбои.

• Оптимизация конфигурации. После выявления проблем производится настройка параметров системы, например, увеличение объёма оперативной памяти на сервере баз данных для улучшения скорости обработки запросов.

Оптимизация системы направлена на улучшение её производительности, автоматизацию управления ресурсами и снижение затрат. Оптимизация системы включает:

Автоматизацию управления ресурсами

• Функции автошкалирования. Включаются функции автоматического масштабирования ресурсов в зависимости от нагрузки.

• Балансировку нагрузки. Настраиваются системы балансировки нагрузки для равномерного распределения запросов между серверами.

• Мониторинг и автоматизация. Настраиваются инструменты мониторинга, такие как CloudWatch, Azure Monitor или Stackdriver, для автоматического реагирования на изменения нагрузки.

Минимизация затрат подразумевает:

• Анализ использования ресурсов. Используются аналитические инструменты для оценки использования ресурсов. Например, если серверы работают менее чем на 50% своей мощности, их количество можно уменьшить.

• Оптимизация расходов. Применяются экономичные модели оплаты, такие как предоплаченные инстансы или спотовые цены.

• Планирование резервирования. Долгосрочные задачи можно переместить на менее дорогие серверы, оптимизируя затраты.

Этап тестирования и оптимизации является критически важным для достижения полной готовности системы после миграции в облако. ООО "Нано Лабс" должна:

a) Провести детальное функциональное тестирование, чтобы убедиться в корректности работы всех процессов.

b) Выполнить нагрузочные тесты, чтобы определить пределы производительности системы и устранить узкие места.

c) Настроить автоматическое управление ресурсами и оптимизировать расходы.

Успешное выполнение всех задач на этом этапе гарантирует, что система будет не только надёжной и производительной, но и экономически эффективной.

Этап 6. Сопровождение системы

После успешного завершения миграции и тестирования системы ООО "Нано Лабс" необходимо организовать постоянное сопровождение облачной инфраструктуры. Этот процесс включает мониторинг работы системы, её регулярное обновление, а также обучение сотрудников для эффективного взаимодействия с новой средой. Сопровождение системы обеспечивает её стабильность, производительность и соответствие требованиям бизнеса.

***Мониторинг* системы** является ключевым инструментом для обеспечения стабильной работы облачной инфраструктуры. В табл. 4 представлены популярные инструменты мониторинга

Таблица 4

Популярные инструменты мониторинга

Инструменты мониторинга	Возможности
CloudWatch (AWS)	Предоставляет детализированные метрики использования ресурсов, включая CPU, память, сетевую активность и дисковое пространство. Позволяет настраивать алерты для уведомлений о превышении установленных порогов.
Azure Monitor	Собирает данные о производительности виртуальных машин, баз данных и приложений. Включает встроенные инструменты для анализа логов и создания отчётов.
Google Operations Suite (Stackdriver)	Предоставляет метрики, журналы и трассировку для мониторинга производительности и доступности. Настройка метрик и алертов.

Анализ метрик производительности и доступности

• Сбор и обработка данных. Ежедневный сбор метрик для анализа производительности и хранение данных для долгосрочного анализа и прогнозирования.

• Анализ тенденций. Выявление паттернов использования системы, например, пиковые нагрузки наблюдаются в определённые дни недели или часы суток.

• Оптимизация конфигурации ресурсов на основе данных анализа.

• Предотвращение проблем. Использование предиктивной аналитики для прогнозирования возможных сбоев, например, если утилизация CPU постоянно превышает 80%, необходимо увеличить мощности или перераспределить нагрузки.

Постоянное обновление системы гарантирует её соответствие новым требованиям бизнеса и техническим стандартам. Этот процесс включает внедрение новых функций, исправление ошибок и актуализацию базы знаний.

Внедрение новых функций

• Анализ потребностей бизнеса. Оценка текущих потребностей компании и определение областей, где можно внедрить улучшения, например, добавление функции автоматического формирования отчётов в CRM-системе.

• Использование возможностей облачных провайдеров. Интеграция новых сервисов, предлагаемых провайдером, например, AWS Lambda для выполнения серверлесс-функций, и тестирование новых функций перед их внедрением.

• Обновление программного обеспечения. Регулярная проверка обновлений для виртуальных машин, баз данных и приложений, например, установка новой версии MySQL для повышения производительности и безопасности.

Актуализация базы знаний

• Создание базы знаний. Разработка документации, включающей описание всех процессов и конфигураций системы.

• Регулярное обновление информации. Например, обновление раздела документации после внедрения нового API.

• Доступность для сотрудников. Обеспечение доступа сотрудников к базе знаний через корпоративный портал. Настройка удобного поиска по ключевым словам.

Обучение сотрудников – это важная часть сопровождения системы, обеспечивающая её эффективное использование и минимизацию ошибок.

Проведение тренингов для пользователей

• Обучение конечных пользователей Проведение тренингов для сотрудников, которые будут использовать облачную систему в повседневной работе. Например, обучение сотрудников отдела продаж работе с обновлённой CRM-системой.

• Обучение новым функциям. После внедрения новых функций проводятся дополнительные тренинги для пользователей, например, обучение работе с аналитическими инструментами, встроенными в облачную платформу.

Обучение администраторов

• Техническое обучение IT-специалистов настройке и администрированию облачной инфраструктуры. Пример: курсы по управлению виртуальными машинами и настройке сети в Azure.

• Обучение безопасности. Подготовка администраторов по управлению доступом, настройке шифрования и реагированию на угрозы. Пример: тренинг по настройке многофакторной аутентификации и мониторинга логов.

Мониторинг, обновление и обучение сотрудников являются основными элементами успешного сопровождения системы. Для ООО "Нано Лабс" это означает:

a) Постоянное отслеживание производительности и доступности системы для предотвращения проблем.

b) Регулярное внедрение новых функций и актуализация базы знаний для повышения эффективности.

c) Проведение тренингов и предоставление обучающих материалов, чтобы сотрудники могли эффективно использовать систему.

d) Эффективное сопровождение позволяет компании максимально использовать возможности облачной инфраструктуры и поддерживать её актуальность в условиях быстро меняющихся бизнес-требований.

Заключение

Переход компании ООО "Нано Лабс" на облачную инфраструктуру представляет собой сложный, многоэтапный процесс, который требует детального анализа, планирования, тестирования и оптимизации. Каждый этап этого перехода играет ключевую роль в достижении конечной цели – создания надёжной, масштабируемой и производительной IT-системы, которая соответствует современным бизнес-требованиям и соответствует всем требованиям действующего законодательства [7-9].

ПАО Ростелеком был определён как наиболее подходящий провайдер для компании, благодаря высокой доступности в России, поддержке бизнес-приложений и гибким тарифам.

На этапе планирования были определены стратегии миграции:

- "Lift-and-Shift" для быстрого переноса систем.
- Рефакторинг и реархитектура для оптимизации и полной модернизации приложений.

Каждый из этих подходов нацелен на решение специфических задач компании.

Миграция компании ООО "Нано Лабс" на облачные серверы стала успешным шагом в направлении цифровой трансформации. Грамотное выполнение всех этапов – от анализа текущей инфраструктуры до сопровождения системы – позволило создать современную IT-среду, способную поддерживать рост компании и отвечать вызовам цифровой экономики.

Литература

1. *Смирнов А.В., Иванова Т.Н*. Облачные вычисления: теоретические основы и практическое применение. М.: Инфра-М, 2020. 456 с.

2. *Петров И.Н., Сидоров В.А.* Модернизация IT-инфраструктуры: от локальных серверов к облачным технологиям. Санкт-Петербург: Наука, 2021. 368 с.

3. *Васильев А.М.* Технологии облачных вычислений в бизнесе. 3-е изд. Екатеринбург: Уральское издательство, 2022. 312 с.

4. *Кузнецов В.П*. Основы корпоративных IT-систем: переход на облачные решения. Новосибирск: Сибирское университетское издательство, 2020. 420 с.

5. *Кузнецов В.И., Петров С.М*. Переход на облачные технологии: преимущества и вызовы для бизнеса // Информационные технологии и системы. 2022. Т. 38. № 4. С. 35-42.

6. *Сидоров М.А.* Использование облачных технологий в корпоративной среде [Электронный ресурс]. М.: Российская академия наук, 2021. URL: https://cloudsolutions.ru/articles (дата обращения: 14.01.2025).

7. Федеральный закон от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации» // Собрание законодательства Российской Федерации. 2006. № 31. Ст. 3448.

8. Федеральный закон от 27 июля 2006 г. № 152-ФЗ «О персональных данных» // Российская газета. 2006. № 165.

9. Федеральный закон от 29 июля 2017 г. № 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации» // Российская газета. 2017. № 169.

10. *Kirov D.E., Toutova N.V., Vorozhtsov A.S., Andreev I.A.* Feature selection for predicting live migration characteristics of virtual machines // T-Comm: Телекоммуникации и транспорт. 2021. Т. 15. № 7. С. 62-70. EDN: AGGBDW

11. *Тутов А.В., Тутова Н.В., Ворожцов А.С., Андреев И.А.* Многокритериальная оптимизация размещения виртуальных машин по физическим серверам в облачных центрах обработки данных // T-Comm: Телекоммуникации и транспорт. 2021. Т. 15. № 1. С. 28-34. EDN: IOFQSS

12. *Губин А.С., Тутова Н.В.* Анализ подхода к разработке приложений с "чистой" архитектурой // Телекоммуникации и информационные технологии. 2022. Т. 9. № 1. С. 28-37. EDN: NOZMKG

ПРИМЕНЕНИЕ АЛГОРИТМОВ УСИЛЕННОГО ОБУЧЕНИЯ В ТЕСТИРОВАНИИ БЕЗОПАСНОСТИ WI-FI СЕТЕЙ

Кривченко Ольга Сергеевна
МТУСИ, магистрант, Москва, Россия
o.s.krivchenko@edu.mtuci.ru

Антонычева Ольга Леонидовна
МТУСИ, старший преподаватель кафедры «Интеллектуальные системы в управлении и автоматизации» факультета «Кибернетики и информационной безопасности», Москва, Россия
o.l.antonycheva@mtuci.ru

Аннотация
В статье рассмотрены некоторые алгоритмы обучения с подкреплением для тестирования безопасности Wi-Fi сетей с целью анализа современных подходов, основанных на машинном обучении. Также обоснована необходимость использования таких алгоритмов для выявления уязвимостей беспроводных сетей. В процессе исследования изучен проект с открытым исходным кодом Pwnagotchi, используемый для тестирования безопасности с применением алгоритма Q-Learning и его модификаций. Для подтверждения представленных теоретических аспектов алгоритмов проведена серия экспериментов, направленных на исследование влияния различных факторов (а именно: количество эпох, функция расчета награды и методы нормализации данных) на точность и качество модели. Результаты работы демонстрируют преимущества использования алгоритмов усиленного обучения в области тестирования безопасности информационных систем и открывают широкие перспективы их применения в будущем.

Ключевые слова
алгоритмы усиленного обучения, Q-Learning, pwnagotchi, Rasberry Pi, хэндшейк, обучение с подкреплением

Введение

Проблема защиты Wi-Fi сетей в условиях стремительного развития беспроводных технологий становится все более актуальной. Угроза атак, которые направлены на эксплуатацию уязвимостей Wi-Fi сетей, возрастает, а потому их защита становится одной из важнейших задач для исследователей и специалистов в области ИТ [1].

Применяемые в настоящее время методы тестирования безопасности Wi-Fi сетей в основном предполагают ручной анализ и использование специализированного ПО, но такой подход требует не только значимых временных и человеческих ресурсов, но и финансовых. Потому технологии автоматизации тестирования безопасности при помощи алгоритмов машинного обучения становятся все более востребованными. Данные технологии предполагают не только способность адаптироваться к изменениям среды и обнаруживать скрытые угрозы, но также, при помощи таких алгоритмов, как алгоритмы с усиленным обучением, можно анализировать динамические и сложные системы. Таким образом, усиленное обучение является перспективным инструментом для тестирования безопасности беспроводных сетей Wi-Fi.

Результаты исследований

Область машинного обучения содержит в себе большое количество алгоритмов, направленных на решение различных задач. Например, алгоритмы классификации, регрессии, кластеризации позволяют обрабатывать большие объемы данных, выделяя заданные закономерности [3]. Усиленное обучение же применяется для создания моделей, которые обучаются путем взаимодействия с окружающей средой. Это полезно для задач, в которых предполагается последовательное принятие решений. Такие алгоритмы как Q-Learning, Deep-Q-Learning показали свою эффективность в робототехнике, оптимизации процессов и тестировании безопасности [2].

К преимуществам алгоритмов усиленного обучения можно отнести:

– динамическое принятие решений, при котором агент обучается взаимодействовать с реальной средой, принимая последовательные решения и адаптируясь к новым сценариям;

– самостоятельное исследование среды, где алгоритмы балансируют между изучением новых возможностей и использованием уже известных;

– адаптивность к новым угрозам и оптимизацию действий.

Характерная особенность Wi-Fi-сетей – изменчивость, то есть постоянное изменение конфигураций, различие в уровнях сигнала и высокая плотность подключений. Кроме того, разнообразие атак создаёт до-

полнительные сложности для анализа и защиты. Именно поэтому среда Wi-Fi является идеальным объектом для применения алгоритмов усиленного обучения.

За основу исследования взят проект с открытым исходным кодом Pwnagotchi. Он представляет собой не только теоретическое, но и практическое применение алгоритмов машинного обучения для тестирования Wi-Fi сетей. Его архитектура основана на алгоритмах усиленного обучения, что позволяет эффективно взаимодействовать с окружающей средой, адаптироваться к изменениям и выявлять уязвимости. Так как данный проект является открытым, он доступен для различных модификаций под конкретные задачи, для тестирования различных алгоритмов, а также содержит богатый набор инструментов для анализа безопасности.

Алгоритмы усиленного обучения

Наиболее распространенными алгоритмами усиленного обучения в сфере тестирования безопасности являются Q-Learning, Deep Q-Learning (DQN) и SARSA.

Q-Learning – это классический метод RL (Reinforcement Learning), в котором формируется Q-таблица для оценки качества выполнения действия при заданном состоянии. Алгоритм обновляет значения Q-таблицы на основе полученного опыта и затем обеспечивает эффективное обучение в условиях неизвестной среды. Популярным его делают простота и эффективность при решении задач с ограниченным числом состояний и действий.

Deep Q-Learning (DQN) – усовершенствованный Q-Learning, в котором вместо Q-таблицы используются нейронные сети. DQN обычно применяют в задачах с большим количеством состояний и действий, таких как анализ сложных сетевых сред. DQN показывает высокую эффективность в кейсах с непрерывными пространствами и высокой степенью неопределённости.

SARSA отличается от Q-Learning тем, что учитывает текущее действие агента при обновлении оценок. Благодаря этому, алгоритм становится более предсказуемым и устойчивым, что обеспечивает безопасность и стабильность стратегии.

Алгоритм Q-Learning

Алгоритм Q-Learning относится к числу методов обучения с подкреплением, которые не требуют предварительной модели среды. Для успешной реализации важны всего несколько ключевых компонентов, такие как состояния, действия, политика, награды и функция Q-значений.

Состояния – это различные ситуации, в которых может оказаться агент. Действия – варианты поведения агента в каждом из состояний. Политика определяет, какое именно и по какому принципу выбрать действие, а награды определяют результативность действий агента с помощью числовых оценок. Функция Q-значений прогнозирует будущие вознаграждения, определяя качество выполнения действия в конкретном состоянии [6].

Математическая формулировка Q-Learning выглядит следующим образом:

$$(Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R_{t+1} + \gamma \max_{a} Q(s_{t+1}, a') -$$
$$Q(s_t, a_t)])$$
$$Q(s_t, a_t)])$$

где:

$Q(s_t, a_t)$ – текущее Q-значение для состояния $s_t s_t$ и действия a_t,

R_{t+1} – награда, полученная за выполнение действия a_t в состоянии s_t,

γ – коэффициент дисконтирования (обычно $0 \leq \gamma < 1$), который определяет, насколько важны будущие награды по сравнению с текущими,

α – коэффициент обучения, который определяет скорость обновления Q-значений,

$\max_{a} Q(s_{t+1}, a')$ – максимальное Q-значение для следующего состояния s_{t+1} по всем возможным действиям a' a'.

Крайне важно правильно настроить гиперпараметры модели для обеспечения стабильности и эффективности алгоритма. К примеру, α определяет, насколько быстро агент адаптируется к изменениям, а γ (коэффициент дисконтирования) влияет на баланс между краткосрочными и долгосрочными вознаграждениями и т.д. Критерии останова также играют немаловажную роль, что позволяет предотвратить переобучение модели при избыточном использовании ресурсов.

Проект Pwnagotchi выбран основой для исследования благодаря своим уникальным характеристикам, главная из которых – инновационный подход, раскрывающий, как алгоритмы искусственного интеллекта могут решать задачи, которые всего десять дет назад требовали значительных человеческих ресурсов. В

контексте тестирования безопасности Pwnagotchi является крайне значимым еще и потому, что открытый исходный код позволяет детально изучить архитектуру и адаптировать ее под специфические цели исследования. Это способствует не только более глубокому пониманию подобных систем, но и созданию новых решений на базе уже существующих наработок. Pwnagotchi легко разворачивается на платформе Raspberry Pi и обладает возможностями тонкой настройки при работе с гиперпараметрами модели [5].

Практическая ценность заключается в том, что Pwnagotchi легко применять не только на тестовых данных, но и в условиях реальных бизнес-процессов.

Структура AI в Pwnagotchi: краткий обзор

Рассматриваемый проект содержит в себе сложную структуру, состоящую из нескольких взаимозависимых файлов [5]:

Таблица 1

Файловая структура AI в Pwnagotchi

Файл *.py	**Описание**
__init__	Инициализирует старт работы pwnagotchi в режиме искусственного интеллекта (ИИ), является ключевым в настройке алгоритма обучения. Модуль работает на базе библиотеки Stable Baselines для работы с алгоритмами обучения с подкреплением, такими как A2C и др. При необходимости есть возможность загрузки модели обучения с диска, используя указанные конфигурации. При успешной инициализации объект модели доступен для обучения и анализа.
epoch	Отвечает за состояние системы в течение определенного промежутка времени. Этот промежуток назввается "эпохой". В классе «Epoch» происходит наблюдение за изменениями среды, обработка данных о сетевых метриках, таких как количество захваченных пакетов или количество пиров, отслеживания событий (ассоциации или деауты). Тут же вычисляются вознаграждения и передаются в ИИ для дальнейшего обучения. После каждой эпохи файл выводит статистику, отображающую метрики активности устройства и рассчитанное вознаграждение.
featurizer	Извлекает признаки из состояния системы. Признаки – это данные, которые отображают текущее состояние среды (информация о точках доступа, клиентах, пирах, активности устройства и др.). Извлеченные данные преобразуются в вектор, ипользуемый в модели для обучения или оценки.
gym	Определяет интерфейс Environment, связывающий агента и среду.
parameter	В данном файле присутствует класс Parameter, который обеспечивает управление параметрами, которые могут быть настроены агентом для оптимизации стратегии.
reward	Содержит функцию RewardFunction, в которой рассчитываются вознаграждения агента после анализа состояния системы.
train	В данном файле сохраняется обученная модель и обрабатывается полученная в ходе обучения статистика, а также отображается прогресс тренировки.

Вместе данные модули формируют структуру, которая обеспечивает обучение агента в среде, имитирующей реальное сетевое окружение. При взаимодействии с системой и оптимизации на основе вознаграждений, Pwnagotchi демонстрирует возможности алгоритмов обучения с подкреплением в автоматизации анализа безопасности Wi-Fi сетей.

Датасет и предобработка данных

При запуске и тестировании проекта Pwnagotchi важную роль также играют качество и объем данных, используемых для обучения модели. Но сбор таких данных в реальных условиях сопряжен с целым рядом сложностей: недостаточное количество доступных Wi-Fi сетей, ограниченные технические возможности оборудования, немалое количество времени, затрачиваемое на сбор данных. Также серьезным препятствием является законодательство, так как нелегитимная эксплуатация уязвимостей Wi-Fi сетей влечет за собой уголовную ответственность.

Однако, для обучения модели и последующих исследований можно использовать, например, метод генерации случайных данных, который позволяет имитировать параметры, характерные для реальной среды. В формируемый файл включены такие метрики, как количество деаутов, ассоциаций, хэндшейков, длительность взаимодействий и другие показатели. Такой подход помогает создать синтетическую среду и ускорить эксперименты, сосредоточившись на настройке гиперпараметров, а также на разработке критериев оценки эффективности модели.

Также при искусственном синтезировании данных открывается возможность тестирования алгоритма в различных сценариях, в т.ч. таких, которые редко возникают в реальных условиях. Таким образом, исследовательский процесс становится более гибким, появляется основа для применения реальных данных в будущем и проверки разработанных решений в практических задачах.

Входные данные должны содержать различные состояния агента, такие как количество деаутов, отправленных за эпоху (num_deauth), количество перехваченных хендшейков (num_handshakes), количество успешных ассоциаций(num_assotiations) и другие.

Для генерации таких данных можно использовать, например, Python-скрипт, который будет генерировать значения для каждой эпохи и сохранять их в формате JSON. У JSON есть ряд преимуществ, из-за которых стоит остановиться именно на данном формате (рис. 1):

– синтаксис напоминает словари Python, является интуитивно понятным и легким для чтения;
– данные легко редактируются вручную;
– формат поддерживают такие языки программирования как Python и JavaScript;
– из-за компактности оптимален в обработке больших объемов данных.


```
{
    "num_deauths": 16,
    "num_associations": 12,
    "num_handshakes": 4,
    "num_hops": 0,
    "active_for_epochs": 0.03,
    "inactive_for_epochs": 0.56,
    "missed_interactions": 0.86,
    "sad_for_epochs": 6,
    "bored_for_epochs": 0,
    "duration_secs": 169,
    "blind_for_epochs": 0.73
},
```


Рис. 1. Пример сгенерированных случайных данных в формате JSON

Результат первого обучения модели на сгенерированных данных в виде графика (рис. 2).

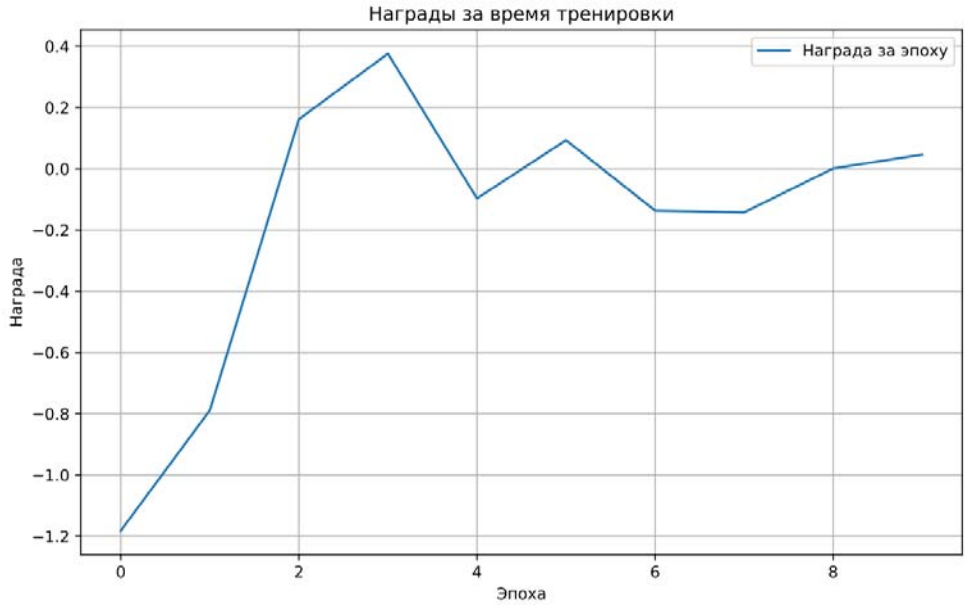

Рис. 2. Результаты обучение модели с отображением наград

Экспериментальная часть

Как описывалось выше, проект Pwnagotchi достаточно гибок, поэтому есть возможность протестировать некоторые изменения и отследить влияние гиперпараметров на поведение модели. В ходе простых экспериментов, таких как изменение количества эпох (рис. 3), корректировка функции расчета наград (рис. 4) и улучшение методов нормализации и денормализации данных (рис. 5) практически определена важность тщательной настройки параметров модели, а также учета множества факторов, влияющих на конечный результат.

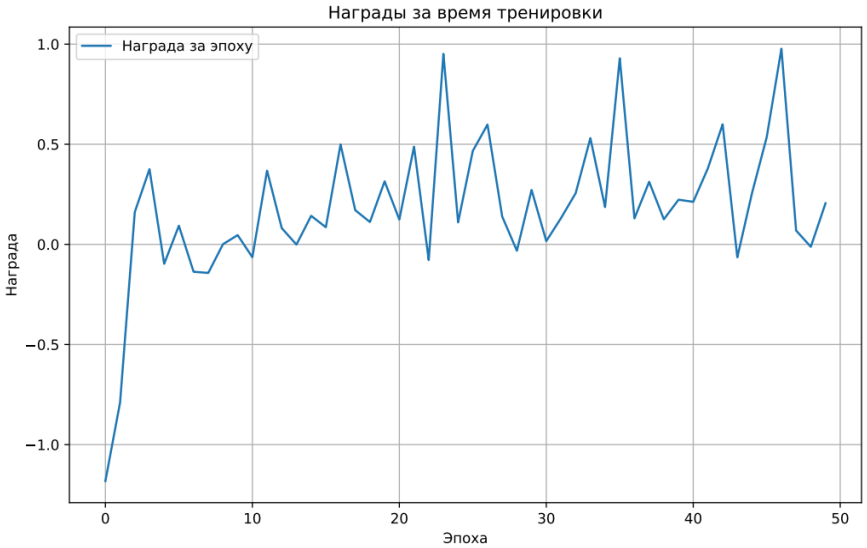

Рис. 3. Результат эксперимента после изменения количества эпох с 10 до 50

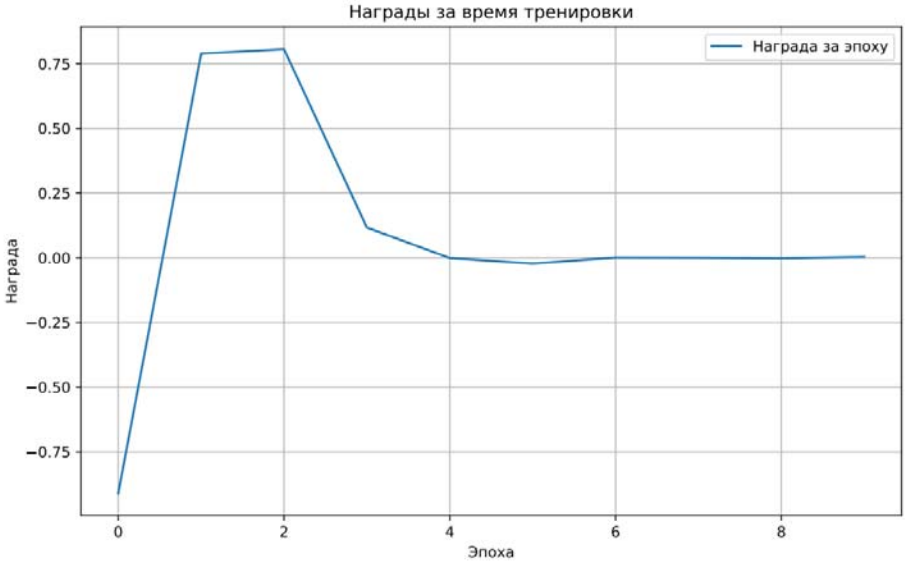

Рис. 4. Результат эксперимента после корректировки функции расчета наград

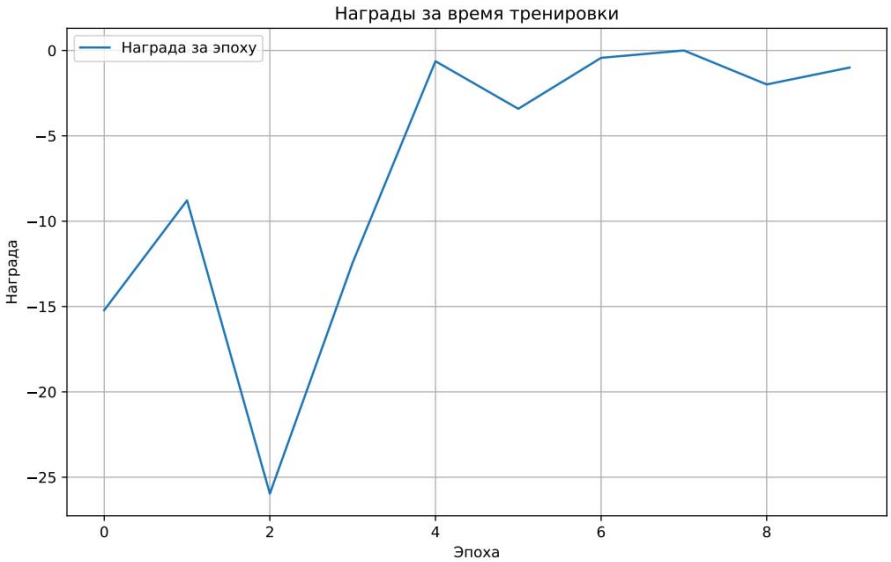

Рис. 5. Результат эксперимента по улучшению методов нормализации и денормализации данных

Возможности улучшений в проекте Pwnagotchi

Благодаря специфичной архитектуре Pwnagotchi и открытому исходному коду, есть возможность непрерывно развивать его в различных направлениях.

В настоящее время в проекте используются такие алгоритмы как Q-Learning и A2C, однако можно заменить их на такие алгоритмы как Deep-Q-Learning или PPO (Proximal Policy Optimization), адаптировав технологию под определенные задачи, что также позволит более подробно рассмотреть новые методы обучения. Также можно доработать систему в части физической реализации и выполнить настройки под под более производительные машины, чем Rasberry Pi, благодаря чему получится увеличить объемы обрабатываемых данных, а следовательно, повысить точность и качество обучения. Благодаря такой адаптации откроется возможность работать на альтернативных сложных конфигурациях и улучшать производительность системы. Pwnagotchi также активно поддерживает систему плагинов, потому доступна собственная разработка нового функционала без необходимости внесения изменений в основной код, например, для интеграции с облачными сервисами, автоматического создания отчетов и т.д. Если изменить источники данных в файле featurizer, можно перенастроить систему для анализа других типов беспроводных соединений таких как Bluetooth или IoT. Наконец, Pwnagotchi можно интегрировать с другими популярными инстурментами для анализа безопасности, такими как WireShark, Metasploit, Kismet. Это позволит еще больше автоматизировать систему и адаптировать ее под более широкий спектр задач.

Таким образом, открытость проекта Pwnagotchi предоставляет широкие возможности как для научных исследований, так и для практического использования в анализе безопасности. Проект может стать основной для мощного автоматизированного инструмента по тестированию безопасности не только Wi-Fi сетей, но и других информационных систем.

Заключение

На основе проекта Pwnagotchi рассмотрены основные принципы работы алгоритмов усиленного обучения в контексте информационной безопасности. Уникальные особенности алгоритмов обучения с подкреплением позволяют заявлять, что подход, предполагающий автоматизацию тестирования безопасности беспроводных сетей, является весьма перспективным и в будущем может быть использован для увеличения эффективности решения задач в области информационной безопасности.

Pwnagotchi предоставляет мощную платформу для исследования различных методов обучения с подкреплением. Рассмотренные алгоритмы используются для автоматического анализа сетей, симуляции реальных угроз и оценки их воздействия на инфраструктуру. Наглядная демонстрация применения синтетических данных для обучения модели и тестирования в различных сценариях показывает, что в процессе исследования можно ускорять разработку и учитывать редко встречающиеся в реальной среде случаи.

Такие алгоритмы как Q-Learning, Deep Q-Learning (DQN) и SARSA позволяют автоматизировать выявление уязвимостей, предсказывать возможные угрозы и разрабатывать эффективные стратегии защиты беспроводных сетей.

Анализ влияния ключевых аспектов настройки моделей, включая выбор гиперпараметров и системы вознаграждений, продемонстрировал высокую гибкость и эффективность алгоритмов обучения с подкреплением в условиях изменяющейся сетевой среды. В дальнейшем проект можно развивать в сторону разработки более сложных моделей, улучшения алгоритмов обучения и расширения функционала системы для работы с новыми типами сетей, включая IoT-устройства и мобильные сети.

Литература

1. *Westcott David A., Coleman David D., Harkins Bryan E.* CWSP Certified Wireless Security Professional Study Guide. Exam CWSP-205. США: Хобокен (Нью-Джерси): John Wiley & Sons Limited, 2018. 701 с. ISBN 9781119244134

2. *Чио Кларенс, Фримэн Дэвид.* Машинное обучение и безопасность. Защита систем с помощью данных и алгоритмов: Чио, Фримэн. М.: ДМК-Пресс, 2020. 388 с. ISBN 978-5-97060-713-8

3. *Куренная В.О.* Искусственный интеллект в информационной безопасности // StudNet. 2022. №6. URL: https://cyberleninka.ru/article/n/iskusstvennyy-intellekt-v-informatsionnoy-bezopasnosti (дата обращения: 22.01.2025).

4. *Кухта А.И.* Анализ методов защиты беспроводной сети Wi-Fi // Молодой исследователь Дона. 2020. №2 (23). URL: https://cyberleninka.ru/article/n/analiz-metodov-zaschity-besprovodnoy-seti-wi-fi (дата обращения: 22.01.2025).

5. Pwny wiki : сайт. URL: https://pwnagotchi.org/getting-started/index.html (дата обращения: 22.01.2025).

6. *Шарибаев А.Н., Шарибаев Р.Н., Абдулазизов Б.Т., Тохиржонова М.Р.* Алгоритмы раннего обучения с подкреплением // Экономика и социум. 2023. №6-2 (109). URL: https://cyberleninka.ru/article/n/algoritmy-rannego-obucheniya-s-podkrepleniem (дата обращения: 22.01.2025).

ОЦЕНКА ВРЕМЕНИ ЗАПАЗДЫВАНИЯ В ИНФОРМАЦИОННЫХ СЕТЯХ

Тымченко Никита
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина), инженер каф. РС, г. Санкт-Петербург, Россия
ntymchenko@etu.ru

Маркелов Олег Александрович
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина), к.т.н., доцент каф. РС, г. Санкт-Петербург, Россия
oamarkelov@etu.ru

Аннотация
В этой статье рассматриваются современные информационно-телекоммуникационные системы как сложные сети с долговременной зависимостью. Традиционные модели теории массового обслуживания недооценивают задержки в сети из-за нестационарности потоков. Авторы предлагают использовать суперстатистический подход, который учитывает взаимозависимость активности узлов сети. Разработана аналитическая коррекция формулы Кингмана на основе вычисления коэффициентов вариации интенсивности поступления и взаимных корреляций между узлами. Результаты подтверждены компьютерным моделированием и анализом трафика в крупной академической сети за 2017-2021 года.

Ключевые слова
Время запаздывания, взаимно коррелированные интенсивности поступления, суперстатический подход, агрегированный трафик, нестационарный трафик.

Введение

В современном мире многопользовательские информационные системы играют ключевую роль в повседневной жизни человека и функционировании различных организаций. Однако одной из важных проблем, с которыми сталкиваются пользователи и разработчики таких систем, является время запаздывания [1]. Задержки в передаче данных, возникающие из-за различных факторов, могут существенно снижать эффективность работы с системой и вызывать недовольство пользователей. Задержки пакетов имеют как постоянную, так и переменную составляющую, к которой относится интересующая нас сетевая задержка (задержка маршрутизации) [2]. Для лучшего понимания динамики трафика и повышения качества обслуживания за счет предсказания поведения информационных сетей моделируют задержки и потери, возникающие в них [3]. Одним из математических методов исследования сложных стохастических систем является теория массового обслуживания, занимающаяся анализом эффективности функционирования систем массового обслуживания (СМО).

Упрощённые математические модели СМО, основанные на аналитических приближениях и предположении об асимптотической независимости, используются уже более века, например в работах Эрланга [4], Поллачека [5], Хинчина [6], Кингмана [7], Маршала [8], Крамера и Лангенбах-Бельца [9]. Однако последние исследования показывают, что предположение о независимости является слишком упрощённым для современных сложных сетей [10-13]. Это приводит к серьёзным погрешностям в оценке их производительности [14-18].

Взаимосвязанные активности на различных узлах приводят к формированию кластеров с повышенными локальными интенсивностями требований в агрегированном трафике, что и показано на рисунке 1.

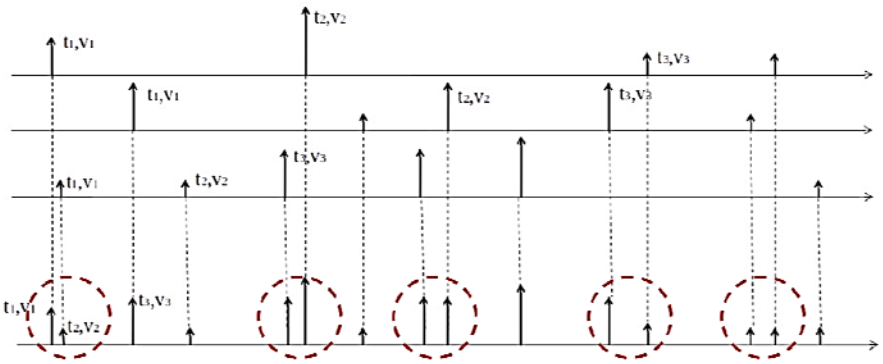

Рис. 1. Схема агрегирование трафика

Недавно нами был предложен суперстатистический подход для численного моделирования и оценки эффективности СМО, который учитывает изменчивость скорости передачи данных между узлами и взаимные корреляции в схемах трафика [10-13]. На основе этого подхода мы оценили влияние автокорреляций и предложили поправки к оценкам длины очереди и времени ожидания на основе анализа интервальных распределений интенсивности поступления [19].

В дальнейшем мы расширили подход для учёта влияния взаимных корреляций между потоками интенсивностей в различных узлах и демонстрации того, как вариации в интенсивностях поступления отражаются на параметрах времени ожидания в агрегированном трафике [20].

В этой же работе будет подробно рассмотрено изменение структуры трафика в крупной академической сети за 2017-2021 год.

Аналитическое моделирование

Основной характеристикой производительности СМО является среднее время пребывания запроса в системе, представляющее собой сумму среднего времени ожидания обработки T_W и среднего времени обслуживания T_S. Традиционный подход к оцениванию среднего времени ожидания обработки T_W предполагает использование формулы Кингмана [7]:

$$T_W = \frac{\bar{v}}{c_0}\left[\frac{U^2}{1-U}\right]\left[\frac{\rho_t^2+\rho_v^2}{2}\right], \quad (1)$$

где $c_0 = \frac{\sum_i v_i}{T_\Sigma}$ – минимальная пропускная способность, необходимая для передачи всего объема трафика за все время анализа; $U = \frac{c}{c_0}$ – коэффициент использования системы; $\rho_\tau = \frac{\sigma_\tau}{\bar{\tau}}$ – коэффициент вариации интервалов времени между запросами; $\rho_v = \frac{\sigma_v}{\bar{v}}$ – коэффициент вариации интенсивностей поступления.

Исследования [14-18] показывают, что данная формула недооценивает эмпирическое среднее время пребывания запроса в СМО на 1-2 порядка, что наиболее ярко выражено в высоконагруженных режимах при значениях коэффициента использования системы $U > 0{,}5\ldots0{,}7$ $U > 0{,}5 \ldots 0{,}7$.

Для повышения точности оценки мы использовали суперстатистический подход, предложенный авторами статьи [21].

Сущность суперстатистического подхода заключается в том, что агрегированный трафик аппроксимируется последовательностью коротких временных фрагментов, в пределах которых данные считаются стационарными. Каждый фрагмент характеризуется интенсивностью запросов пользователей $\beta = 1/\bar{\tau}$, где $\bar{\tau}$ – средний интервал времени между поступлениями отдельных запросов. Распределение временных интервалов между запросами в каждом фрагменте определяется законом Пуассона:

$$P(\tau) = \int_0^\infty P(\beta)\beta^2 e^{-\beta\tau} d\beta. \quad (2)$$

Наилучшим образом интенсивность поступления запросов описывается гамма-распределением:

$$P(\beta) = \frac{\lambda^\alpha}{\Gamma(\alpha)}\beta^{\alpha-1}\exp(-\lambda\beta), \quad (3)$$

где $\lambda = \alpha/\bar{\beta}$ – коэффициент интенсивности, α – коэффициент формы, Γ(α) – гамма-функция. Коэффициент формы обратно пропорционален квадрату коэффициента вариации $\rho = \sigma(\beta)/\bar{\beta} : \alpha = 1/\rho^2$.

Подстановка плотности вероятности $P(\beta)$ в выражение для $P(\tau)$ приводит, как показано в [11], к q-экспоненциальному распределению:

$$P(\tau) = C_0[1 + b(q-1)\tau]^{-1/(q-1)}, \quad (4)$$

$$q = \frac{\alpha+3}{\alpha+2}, \quad (5)$$

$$b = \frac{\alpha+2}{\lambda}, \quad (6)$$

$$C_0 = \frac{\alpha(\alpha+1)}{\lambda^2}. \quad (7)$$

Исследование [11] показало, что q-экспоненциальное распределение хорошо описывает функции распределения вероятностей интервалов времени между запросами как в стационарном, так и в нестационарном случаях.

Математическое ожидание для к q-экспоненциального распределения равно:

$$E(\tau) = \frac{1}{b(3-2q)},$$

а дисперсия определяется выражением:

$$D(\tau) = \frac{q-2}{(2q-3)^2(3q-4)b^2}.$$

Коэффициент вариации агрегированного трафика будет определяться выражением:

$$\rho^2 = \frac{D(S)}{E^2(S)} = \frac{ND(X_1)(1-R)}{N^2E^2(X_1)} + \frac{D(X_1)RN^2}{N^2E^2(X_1)} = \frac{D(X_1)(NR+1-R)}{NE^2(X_1)}.$$

Поскольку используемый в расчетах и при последующем моделировании коэффициент ρ^2 должен быть меньше 1, мы можем определить требования к коэффициенту вариации одного слагаемого (трафика индивидуального пользователя):

$$\rho^2 < 1 \Rightarrow \frac{D(X_1)(NR+1-R)}{NE^2(X_1)} < 1 \Rightarrow \frac{D(X_1)}{E^2(X_1)} < \frac{N}{NR+1-R}. \quad (8)$$

Нами использовались следующие обозначения: $\beta = C$ и так как у нас N узлов β и λ были заменены на $\beta_N = C * N$ и $\lambda_N = \frac{\alpha}{\beta_N}$.

В результате было получено, что

$$\rho^2 = \frac{\rho_1^2(NR_0+1-R_0)}{N}, \quad (9)$$

где $\rho_1 = \frac{D(X_1)}{E^2(X_1)}$ – коэффициент вариации трафика одного пользователя.

На основании оценки квадрата коэффициента вариации ρ^2 (9) был найден квадрат коэффициента вариации интервалов ρ_τ^2 аналитически (10):

$$\rho_\tau^2 = \frac{-\rho^2-1}{1+2\rho^2} \cdot \frac{1+2\rho^2}{\rho^2-1} = \frac{-\rho^2-1}{\rho^2-1}. \quad (10)$$

Подстановка ρ_τ^2 (10) в формулу Кингмана (1) позволяет ввести коррекцию, учитывающую корреляции во взаимной активности различных узлов сети, и позволяет производить адекватную оценку для высоконагруженного режима, что будет показано далее.

Формула Кингмана показывает, что происхождение дисперсии не имеет значения. Она может возникать из-за нестабильности как интенсивности пользователей, так и качества обслуживания, но если эти два фактора независимы, то для задержки обслуживания важна только общая дисперсия [20].

Математическое моделирование

Для проверки точности нашей аналитической модели в разных ситуациях была разработана математическая модель, генерирующая потоки интенсивностей для некоррелированных и долгосрочно коррелированных записей интенсивности, как показано на рисунке 2.

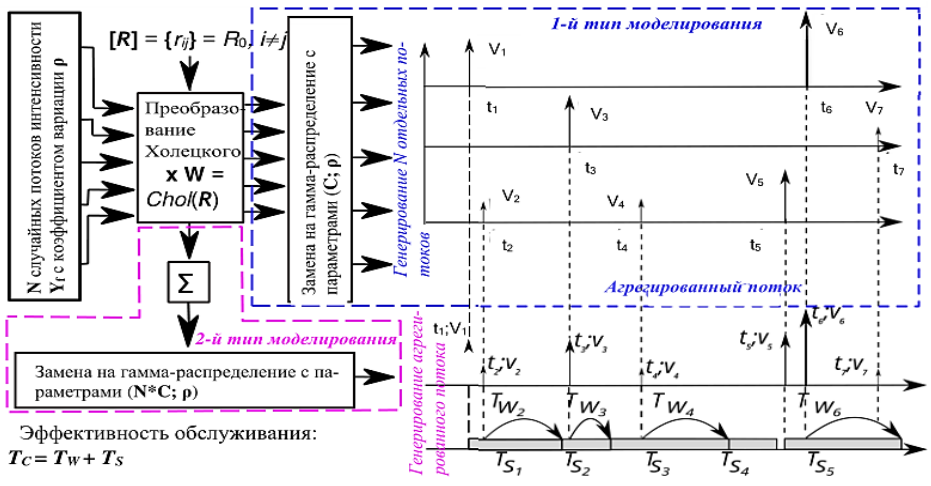

Рис. 2. Схема математического моделирования

Сначала мы построили ряд интенсивности $Y = Y_0 + Y_f$, где $Y_0 = C$ – простая константа, а Y_f – флуктуирующий временной ряд с нулевым средним. Для разных комбинаций параметров модели (N – количества узлов, C – средней интенсивности, ρ – коэффициентов вариации и R_0 – R_0 – взаимных корреляций) мы сгенерировали V случайных конфигураций для разных L длительностей сегментов. Чтобы сравнение симуляционных данных с наблюдениями было более релевантным, мы выбрали длительность сегментов таким образом, чтобы общая статистика оставалась такой же, как и для полной суточной записи из сети с таким же количеством активных узлов, т.е. $L * V \approx 86400$.

Затем мы моделировали систему очередей для каждого сегмента интенсивности, генерируя локальные пуассоновские потоки, состоящие из Y_k – элементов со средним интервалом $1/Y_k$, и конкатенировали их для всех сегментов $k = 1 \dots L$ $K = 1 \dots L$. С помощью алгоритма разложения Холецкого

1) генерируется необходимое количество векторов нормально распределенных случайных величин с заданными параметрами: L, N и ρ;

2) рассчитывается треугольная матрица корреляции с заданным параметром R_0, полученная с помощью преобразования Холецкого;

3) произведение данной матрицы корреляции и исходного набора векторов, полученного на шаге 1, образует набор взаимно коррелированных рядов с долговременной зависимостью;

4) производится замена нормального распределения на гамма распределения с параметром формы $a = 1/\rho$ и масштабным коэффициентом $b = C/a$.

Рекомбинация рядов по Холецкому корректно работает для центрированных нормальных случайных рядов, а для смещенных возникают проблемы с взвешиванием, поэтому введение корреляций нужно производить до замены распределения.

В предыдущем разделе было показано, что лучше всего интенсивность пользовательских запросов описывает гамма распределение, поэтому на четвертом шаге были получены ряды с гамма распределением.

Затем объединили все локальные потоки и получили итоговые последовательности прибытия [19, 22].

С учетом физически реализуемых задержек провели два типа моделирования:

1) с потоками Пуассона, созданными независимо для каждого узла N, затем агрегированными и размещёнными в соответствии с временем прибытия sim1.

2) с объединением локальных интенсивностей трафика из N узлов и моделированием одного агрегированного потока Пуассона sim2.

Важно учитывать, что модель создаёт потоки, а не сам трафик, потому что нам важна маршрутизация, а не структура трафика. В ходе математического моделирования мы построили графики плотности вероятности оценки времени запаздывания $P(\tau)$, учитывая физическое ограничение на время запаздывания от $45 * 10^{-6}$ до 2,7.

На рисунке 3 показаны распределения $P(\tau)$ времени между прибытиями τ в сети с $N = 16$ $N = 16$ активными узлами со средней интенсивностью $C = 100$ $C = 100$ запросов в единицу времени, длительностью $L = 86$ $L = 86$ единиц времени, со статистикой, собранной для $V = 1000$ $V = 1000$ случайных конфигураций, причем боксплоты указывают на результаты компьютерного моделирования в сравнении с аналитической аппроксимацией по уравнению (4).

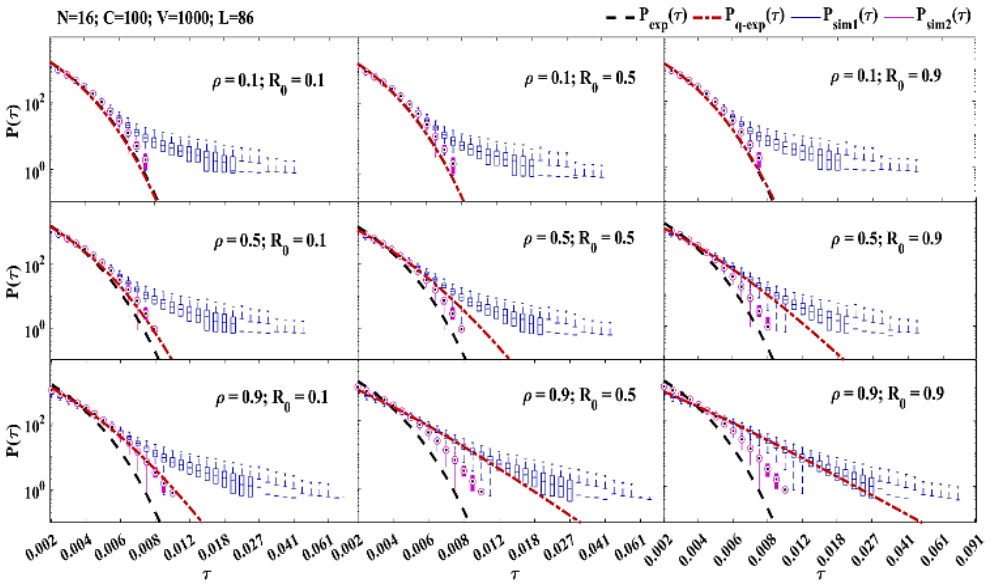

Рис. 3. Семейство графиков зависимости плотности вероятности оценки времени запаздывания от ρ и R_0

На рисунке видно, что, хотя все варианты дискретного моделирования подвержены влиянию эффектов дискретности и конечного размера, причем вариант sim1 $sim1$представляет собой верхнюю, а вариант sim2 $-sim2-$ нижнюю границы, соответственно, аналитическая кривая обычно лежит между ними, с более выраженным отклонением от единичной экспоненты при увеличении ρ и R_0, которое в большей степени зависит от ρ.

Для моделирования требовалось решить задачу подбора набора оптимальных параметров, которые бы минимизировали влияние артефактов. В ходе исследования было установлено, что математическая модель демонстрирует инерционность в пределах одного интервала, и уменьшение длины интервала LL приводит к более прямому графику, при этом значение τ также уменьшается, так как тяжесть хвоста гамма-распределения зависит от LL. Это устраняет провал в графике $P_{sim1}(\tau)$. Увеличение числа пользователей приводит к смещению графиков вправо по оси τ и лучшему описанию с помощью q-экспоненты вместо обычной экспоненты, что соответствует теории, так как модель разрабатывается для многопользовательской высоконагруженной сети.

В работе [20] показано, что с ростом N экспонента недооценивает время на порядок. Графики $P_{sim2}(\tau)$, напротив, смещаются влево при увеличении NN и почти совпадают с экспонентой. При малых N оба метода агрегирования дают схожие результаты.

Далее было построено семейство графиков оценки времени запаздывания по уточненной формуле Кингмана при тех же параметрах моделирования, которое показано на рисунке 4.

Открытые и полные боксплоты представляют результаты компьютерного моделирования по вариантам sim1 и sim2 соответственно, а пунктирные линии соответствуют аппроксимации по формуле Кингмана (1) с соответствующими аналитическими поправками к коэффициенту вариации интервалов времен ρ_t^2 по формуле (10) в зависимости от взаимных корреляций интенсивностей из разных узлов, определяемых аналитической обработкой.

Видно, что формула Кингмана хорошо работает для малых ρ, а при высоких значительно недооценивает уже при $U = 0{,}4$, а при меньших U переоценивает.

Второй метод агрегирования (sim2) хорошо согласуется с теорией при больших значениях ρ и высоких значениях N вплоть до больших нагрузок.

Первый метод начинает расходиться при высоких значениях NN и малых значениях U из-за конечности выборки и дискретности.

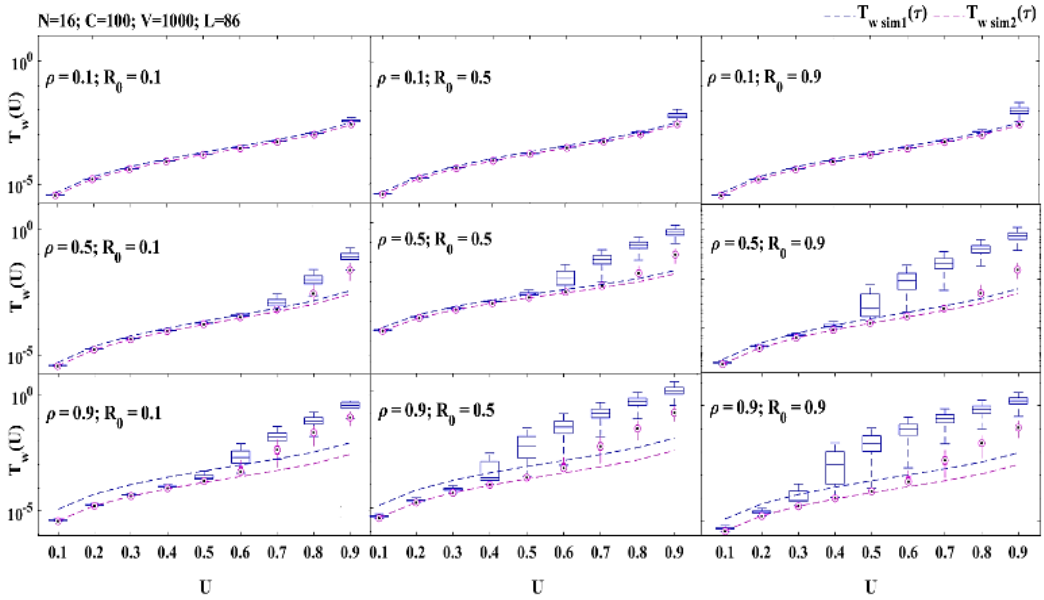

Рис. 4. Семейство графиков зависимости оценки времени запаздывания по уточненной формуле Кингмана от ρ и R_0

При $L \to 1$ оба метода хорошо согласуются с теорией, но разница в оценке времени ожидания составляет порядок.

Результаты моделирования равнокоррелированных моделей трафика с постоянными интенсивностями и вариациями внутри пакета могут показаться нереалистичными, так как в реальности распределения интенсивности в узлах следуют тяжёлым хвостовым функциям [12], а корреляционные модели эволюционируют сложным образом с течением времени [19].

Анализ трафика

Для того чтобы оценить, насколько вышеуказанные аналитические приближения применимы для анализа трафика в реальных сетях, ниже мы проанализировали ту же статистику для эмпирических данных трафика за 2017-2021 год, собранных из выборочной точки на магистрали академической сети MAWI, соединяющей несколько университетов и исследовательских центров в Японии.

Следует учесть, что в реальных условиях трафик подвержен как автокорреляции, так и взаимокорреляции. В реальной сети малое количество узлов в течение суток работает без простоев, поэтому какие-то интервалы времени заполнены 0-ми интенсивностями. Для корректной оценки взаимной корреляции исключили нулевые интенсивности и автокорреляции, так как их влияние уже было рассмотрено в статье [22], и отсортировали их в порядке убывания суммарной интенсивности.

Для избавления от автокорреляций, вызванных суточными циклами, была произведена сортировка столбцов случайным образом с сохранением их взаимного расположения, тем самым избавились от автокорреляций, сохранив прежние корреляции между узлами.

В результате выделили три категории узлов:

1) высокоинтенсивные – с высокой интенсивностью в течение суток (N_{hi});

2) среднеинтенсивные – со средней интенсивностью в течение суток или с высокой в течение 12 часов (N_{mid})N_{mid});

3) низкоинтенсивные – с низкой интенсивностью в течение суток (N_{low}).

На рисунке 5 приведены зависимости средней величины и СКО интенсивности поступления от категории узла в агрегированном трафике за 2017-2021 год.

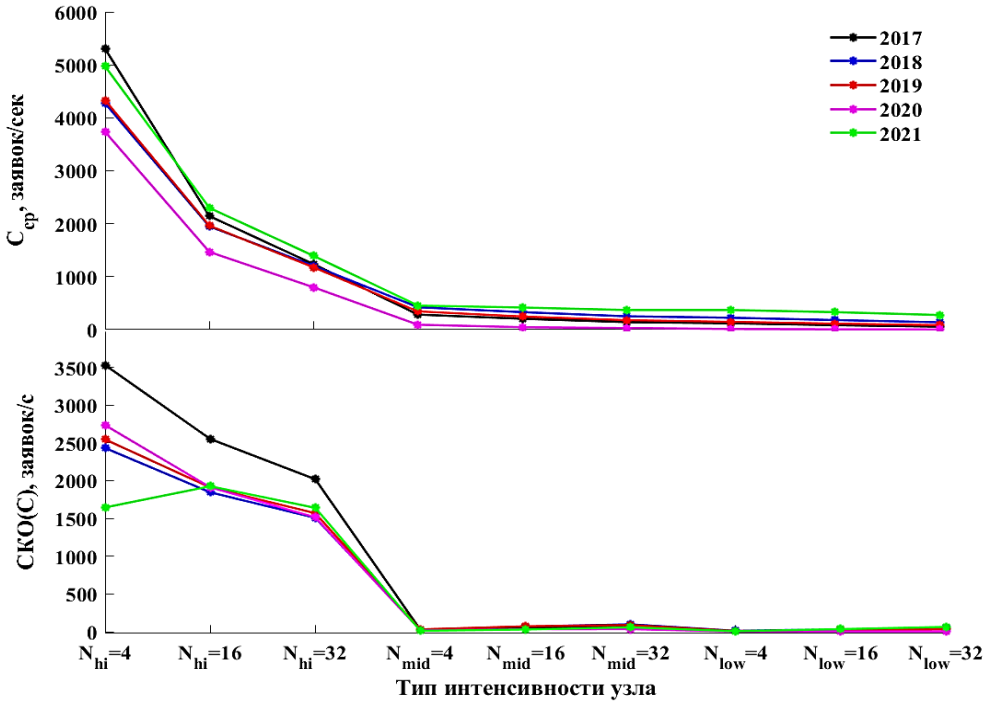

Рис. 5. График зависимости средней величины и СКО интенсивности поступления от категории узла в агрегированном трафике

Затем привели среднесуточную интенсивность поступления на узел к 100 заявкам в секунду.

На рисунке 6 показаны типичные средние коэффициенты взаимной корреляции R_0 и средние коэффициенты вариации ρ интенсивности двунаправленного (т.е. исходящего и/или адресованного) трафика нескольких кластеров $N = 4, 16, 32$ репрезентативных узлов сети с высокой, средней и низкой интенсивностью трафика соответственно при разных интервалах наблюдения T для эмпирических данных трафика, собранных из выборочной точки на магистрали академической сети MAWI.

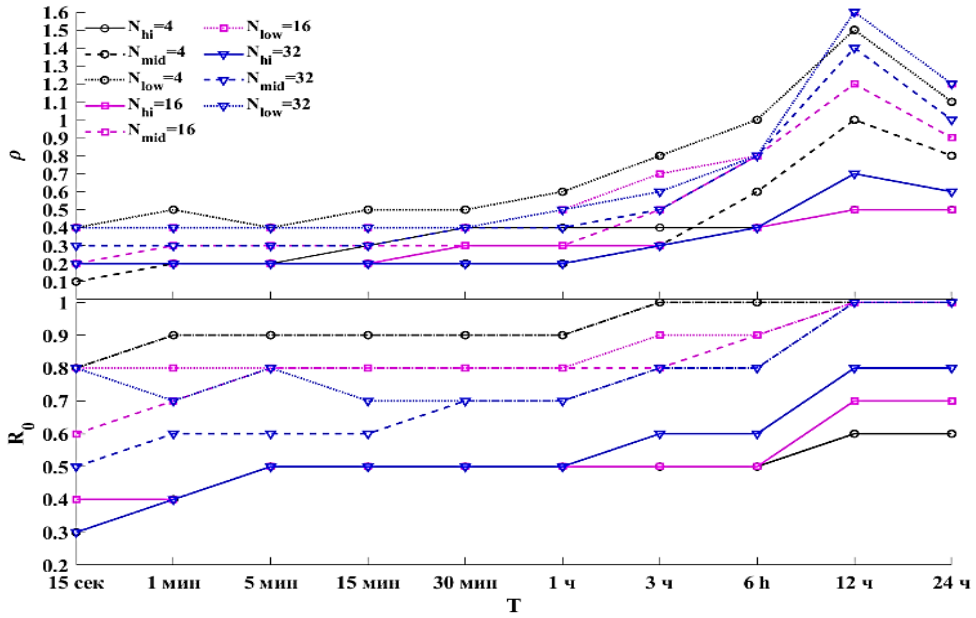

Рис. 6. Зависимость средних значений ρ и R_0 от T, N и типа интенсивности

Из рисунка 6 видно, что ρ для высокоинтенсивных узлов в два раза меньше, чем для всех остальных, а узлы с $\rho \geqslant 1$ уже являются экзотическим случаем и описываются логнормальным распределением интенсивностей, а не гамма-распределением. При этом краткосрочная динамика в масштабах до нескольких часов представлена $\rho \ll 1$, что говорит о том, что указанные параметры модели актуальны для реалистичного примера сети, по крайней мере, в среднем. Также узлы с $R_0 = 1$ $R_0 = 1$при моделировании просто суммируются, поэтому не представляют для нас особого интереса.

Высокоинтенсивные пользователи, наоборот, вызывают интерес для рассмотрения, так как в настоящие время высокоинтенсивный режим не обеспечен адекватными моделями.

На рисунке 7 представлена зависимость среднесуточных значений ρ и R_0 от года, N и типа интенсивности. Видно, что до прихода пандемии и введения локдауна, т.е. до начала 2020 год, коэффициент вариации внутри пакета и взаимная корреляция между узлами линейно возрастали с уменьшением средней интенсивности поступления от каждого узла в агрегируемом трафике.

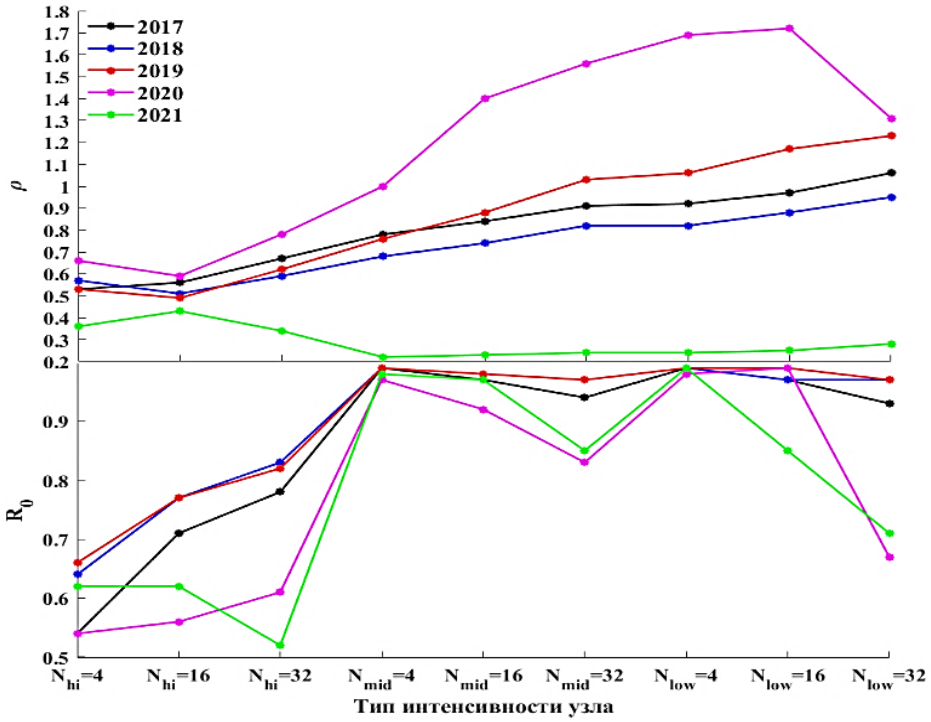

Рис. 7. Зависимость среднесуточных значений ρ и R_0 от года, N и типа интенсивности

Ковидные ограничения в 2020 году внесли свои изменения в том числе и в структуру трафика, так как переход на дистанционную работу и обучение привел к увеличению трафика от видеоконференций и от Интернета-вещей, но снизил его в кампусах университетов и привел к уменьшению рабочего и учебного времени, в связи с этим наблюдается резкое увеличение среднесуточного значения коэффициента вариации для средне- и низкоинтенсивных узлов.

В 2021 году ковидные ограничения начали смягчать и стали переходить на многосменную сокращенную учебу и работу, поэтому среднесуточный коэффициент вариации резко упал для средне- и низкоинтенсивных узлов. Высокоинтенсивные узлы оказались менее чувствительны к этому, так как среднесуточный трафик практически не изменился, и они продолжили использоваться без простоев.

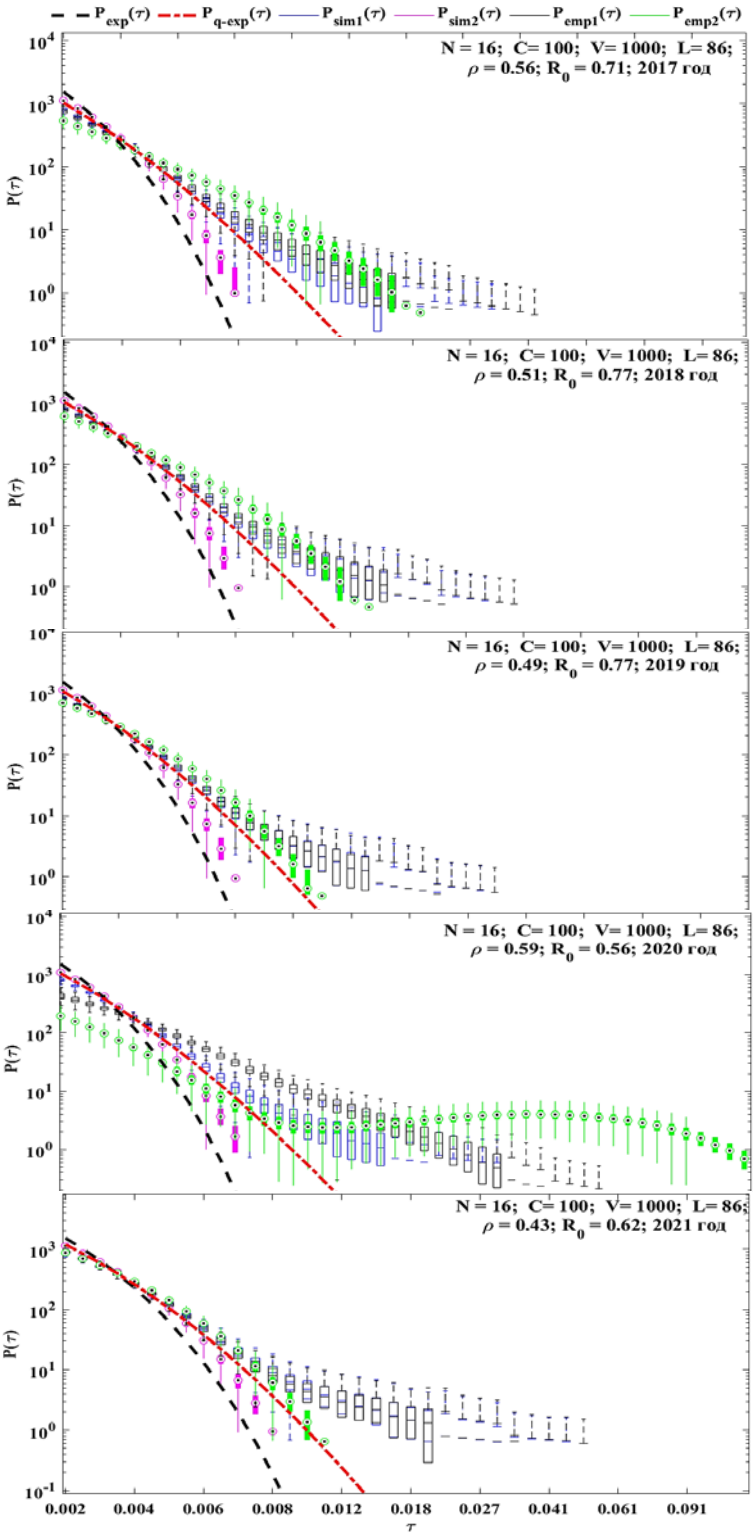

Рис. 8. Семейство графиков зависимости плотности вероятности оценки времени запаздывания от года

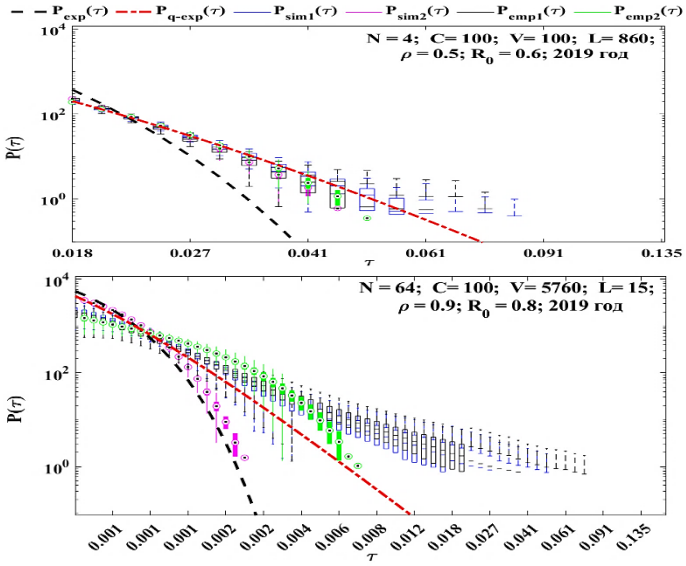

Рис. 9. Семейство графиков зависимости плотности вероятности оценки времени запаздывания от количества узлов в агрегируемом трафике

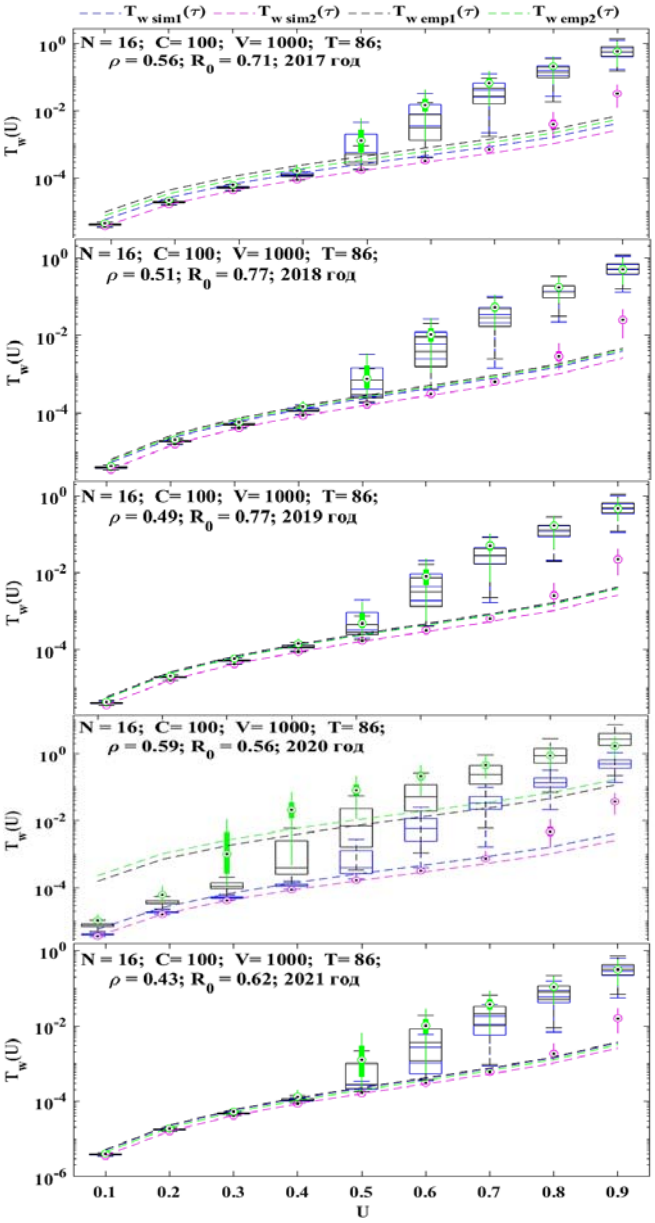

Рис. 10. Семейство графиков зависимости оценки времени запаздывания по уточненной формуле Кингмана от года

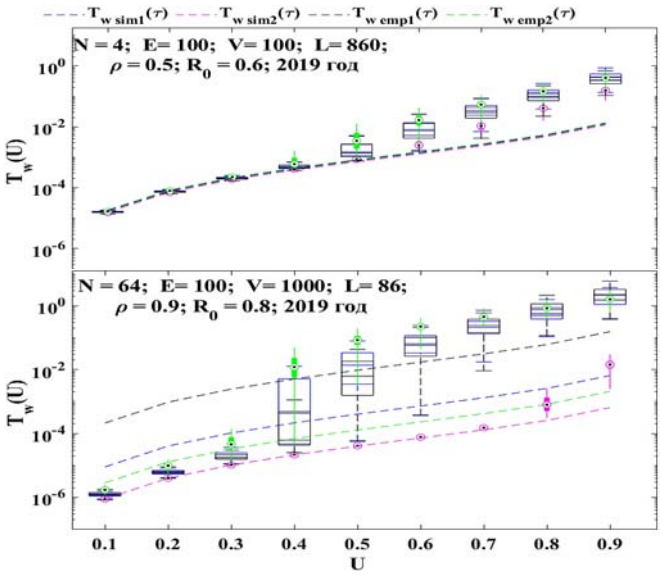

Рис. 11. Семейство графиков зависимости оценки времени запаздывания по уточненной формуле Кингмана от количества узлов в агрегируемом трафике

Затем мы сопоставили результаты прямых оценок и моделирования очередей с ранее представленными аналитическими решениями.

Результаты сравнения представлены на рисунках 8-11.

На рисунках 8-11 открытые и полные боксплоты соответствуют двум вариантам моделирования, в которых интенсивности либо моделируются (sim), либо полученными из эмпирических записей сети MAWI (emp).

На рисунках 8 и 9 красные кривые со штриховыми точками соответствуют аналитическим приближениям для ρ_τ в соответствии с уравнением (10).

На рисунках 10 и 11 пунктирные линии соответствуют аппроксимации по формуле Кингмана (1) с соответствующими аналитическими поправками коэффициенту вариации интервалов времени между запросами.

На рисунках 8 и 9 видно, что формы функций распределения часто совпадают при сравнении результатов имитаций и реальных данных, хотя коэффициенты вариации ρ и взаимные корреляции $R_0 R_0$ в эмпирических моделях соответствуют параметрам аналитической и математической модели только в среднем.

Рисунки 10 и 11 демонстрируют, что для рассматриваемой ранее комбинации параметров модели, предложенные аналитические приближения к коэффициентам вариации времени между прибытиями, введенными в формулу Кингмана (1) обеспечивают разумные приближения, хотя коэффициенты вариации ρ и взаимные корреляции R_0 в эмпирических моделях соответствуют параметрам аналитической и математической модели только в среднем.

Отклонения от результатов моделирования заметны при высоких значениях ρ и R_0, близких к единице, и только для некоторых сценариев моделирования (см. рис. 9 и 11). Это может быть связано с эффектами тяжелых «хвостов» эмпирических распределений и нелинейных взаимодействий, которые не учитываются линейным корреляционным анализом.

Аналитические приближения соответствуют результатам обработки эмпирических данных по алгоритмам 1 и 2, учитывая влияние дискретности и конечного размера. Это подтверждает обоснованность и точность предлагаемых аналитических поправок, которые учитывают взаимную корреляцию в динамике нескольких узлов сети.

Заключение

Возрастающая роль устройств IoT и сложные паттерны трафика приводят к сложному наложению слоёв в разных масштабах сети. Суперстатистические модели, в отличие от классических, оказались более адекватными, сократив недооценку времени ожидания с 3-4 десятков до менее чем десятка [23].

Потоки трафика характеризуются коэффициентами вариации и показателем Херста H. Взаимная динамика описывается матрицами взаимной корреляции $R = r_{ij}$. Узкие места возникают на определённых уровнях агрегирования, и для выявления задержек нужно связывать их с вышеуказанными показателями.

В ходе работы было выявлено, что аналитическое приближение, первоначально полученное при допущении равнозначных моделей интенсивности трафика, также демонстрирует разумную точность, когда вышеуказанные требования удовлетворяются только в среднем.

Аналитическое приближение демонстрирует разумную точность, подтверждённую компьютерным моделированием и анализом реальных данных, собранных с точки доступа на магистральной сети MAWI. Предлагаемая модель обеспечивает аппроксимацию, занимающую промежуточное положение между двумя вариантами моделирования, ни один из которых не является точным из-за неизбежной дискретности и эффектов конечного размера как в сценариях моделирования, так и в реальных сценариях анализа данных.

Благодарности

Данное исследование было поддержано Министерством науки и высшего образования (задание № FSEE-2025-0006).

Литература

1. *Олифер В., Олифер Н*. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 5-е изд. Санкт-Петербург: Питер, 2016. 992 с.

2. *Данилов А.Н., Пантелеев В.Е., Максимов С.П*. Математическая модель для оценки качества обслуживания на участке программно-конфигурируемой корпоративной сети // REDS: Телекоммуникационные устройства и системы. 2024. № 1. С. 12-17.

3. *Шелухин О.И., Тенякшев А.М., Осин А.В.* Моделирование информационных систем: учебное пособие. Москва: САЙНС-ПРЕСС, 2005. 367 с.

4. *Erlang A*. Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges // Elektrotheknikeren. 1975. № 13. С. 5-13.

5. *Pollaczek F.* Über Eine Aufgabe der Wahrscheinlichkeitstheorie // Math Z. 1930. Т. 1. № 32. С. 64-100.

6. *Khintchine A*. Mathematical theory of stationary queues // Matem. Sbornik. 1932. № 39. С. 73-84.

7. *Kingman J*. The single server queue in heavy traffic // Mathematical Proceedings of the Cambridge Philosophical Society. 1961. Т. 04. № 57. С. 902-904.

8. *Marchal M*. An approximate formula for waiting time in single server queues // AIIE Trans. 1976. Т. 4. № 8. С. 473-474.

9. *Kramer W., Langenbach-Belz M.* Approximate formula for the delay in the queueing system GI/G/1 // Congressbook, 8th ITC. 1976. Т. 1. № 235. С. 1-8.

10. *Tamazian A., Nguyen V., Markelov O., Bogachev M*. Universal model for collective access patterns in the Internet traffic dynamics: A superstatistical approach // EPL. 2016. Т. 1. № 115.

11. *Markelov O., Duc V.N., Bogachev M*. Statistical modeling of the Internet traffic dynamics: To which extent do we need long-term correlations? // Physica A. 2017. № 485. С. 48-60.

12. *Nguyen V., Markelov O., Serdyuk A., Vasenev A., Bogachev M.* Universal rank-size statistics in network traffic: Modeling collective access patterns by Zipf's law with long-term correlations // EPL. 2018. Т. 5.

13. *Bogachev M., Pyko N., Pyko S., Vasenev A*. Service delays in strongly linked network communities // Journal of Physics: Conference Series. 2019. Т. 1. № 1352.

14. *Liu Y., Whaitt W.* Stabilizing performance in networks of queues with time-varying arrival rates // Probab. Engrg. Inform. Sci. 2014. Т. 4. № 28. С. 419-449.

15. *Pender J., Rand R.H., Wesson E.* An analysis of queues with delayed information and time-varying arrival rates // Nonlinear Dyn. 2018. Т. 4. № 91. С. 2411-2427.

16. *Whitt W*. Time-varying queues // Queueing Models Serv. Manag. 2018. Т. 2. № 1.

17. *Dudin A., Klimenok V.I., Vishnevsky V.M.* The Theory of Queuing Systems with Correlated Flows// Springer. 2020.

18. *Zhang J., Lee T.T., Ye T., Huang L.* An approximate mean queue length formula for queueing systems with varying rate // J. Ind. Manag. Optim. 2021. Т. 1. № 17.

19. *Bogachev M.I., Kuzmenko A.V., Markelov O.A., Pyko N.S., Pyko S.A*. Approximate waiting times for queueing system with variable long-term correlated arrival rates // Physica A. 2023. № 614.

20. *Bogachev M.I., Pyko N.S., Tymchenko N., Pyko S.A., Markelov O.A*. Approximate waiting times for queuing systems with variable cross-correlated arrival rates // Physica A. 2024. № 654.

21. *Tamazian A., Nguyen V.D., Markelov O.A., Bogachev M.I.* Universal model for collective access patterns in the Internet traffic dynamics: A superstatistical approach // EPL. 2016. Т. 1. № 115.

22. *Тымченко Н., Маркелов О.А.* Оценка времени запаздывания в информационных сетях // Современные проблемы радиоэлектронники и телекоммуникаций: сборник научных трудов. 2024. С. 63.

23. *Markelov O.A., Duc V.N., Bogachev M.I.* Statistical modeling of the Internet traffic dynamics: To which extent do we need long-term correlations? // Physica A. 2017. № 485.