

## **МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И КОМПЬЮТЕРНЫХ СЕТЕЙ**

УДК 004.72

### **ИССЛЕДОВАНИЕ И АНАЛИЗ ПРОЦЕССОВ МНОГОПУТЕВОЙ МАРШРУТИЗАЦИИ И БАЛАНСИРОВКИ ПОТОКОВ ДАННЫХ В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ НА ОСНОВЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА**

**Д. А. Перепелкин**, д.т.н., доцент, декан ФВТ РГРТУ, Рязань, Россия;  
orcid.org/0000-0003-4775-5745, e-mail: perepelkin.d.a@rsreu.ru

**В. Т. Нгуен**, магистрант РГРТУ, Рязань, Россия;  
orcid.org/0000-0003-2930-5775, e-mail: nguyenvantinsreu@gmail.com

*В настоящее время для обеспечения качества сетевых сервисов и приложений широкую популярность получила новая технология компьютерных сетей – программно-конфигурируемые сети (ПКС). ПКС позволяют обеспечить гибкость в управлении потоками данных за счет централизованного представления всей сети и возможности программирования сетевых сервисов. Для обеспечения качества сервиса мультимедийного контента, голоса, видео-трафика и других высоконагруженных приложений наиболее эффективным механизмом является использование методов, принципов и алгоритмов многопутевой маршрутизации и балансировки нагрузки в сети. Цель работы – исследование и анализ процессов многопутевой маршрутизации и балансировки нагрузки в программно-конфигурируемых сетях на основе генетического алгоритма. Проведенные исследования показывают, что применение генетического алгоритма в ПКС позволяет повысить эффективность процессов многопутевой маршрутизации и балансировки потоков данных, а также уменьшить джиттер и задержку передачи в сети.*

**Ключевые слова:** программно-конфигурируемые сети, многопутевая маршрутизация, балансировка нагрузки, протокол OpenFlow, генетический алгоритм.

**DOI:** 10.21667/1995-4565-2022-79-31-48

#### **Введение**

Спрос на широкий спектр сетевых услуг является основной движущей силой инноваций и развития различных сетевых технологий. Обеспечение пропускной способности сети и гарантии качества обслуживания (Quality of Service, QoS) являются ключевыми вопросами в удовлетворении этого спроса. Наличие нескольких физических или логических интерфейсов, объединенных в протокол многопутевой маршрутизации и пересылки пакетов данных, позволяет пользователям использовать несколько маршрутов для установления одновременных соединений. Использование нескольких маршрутов направлено на обход сценариев единой точки отказа, а также сосредоточено на облегчении предоставления всей сети, где ее эффективность зависит от обеспечения широкого класса сетевых услуг и гарантии качества сервиса при высоких нагрузках в сети. В данной работе выполнены исследование и анализ процессов многопутевой маршрутизации и балансировки потоков данных в программно-конфигурируемых сетях (ПКС) на основе генетического алгоритма.

### Программно-конфигурируемые сети

Программно-конфигурируемые сети (ПКС) являются современным трендом в построении сетей связи нового поколения. ПКС позволяют обеспечить гибкость в управлении потоками данных за счет разделения плоскости управления и передачи данных. Основная идея ПКС состоит в упрощении сетевых элементов плоскости передачи данных за счет логической централизации управления сетью, контроля и поддержки сетевых потоков с помощью специального программного обеспечения. Концепция ПКС позволяет выполнить переход от отдельных сетевых элементов к платформам в целом. Управление сетевыми потоками происходит с помощью программных приложений, которыми можно оптимизировать потоки, чтобы найти оптимальный маршрут, а также сконфигурировать сеть для максимального использования связей и числа пользователей [1-4].

Архитектура ПКС состоит из трех основных компонентов: плоскости управления, которая может включать один или несколько управляющих контроллеров; плоскости передачи данных, которая характеризуется наличием сетевого оборудования, взаимодействующего для формирования сети передачи данных; и плоскости программных приложений ПКС, который предоставляет возможности для выполнения различных сетевых решений.

### Теоретические исследования

В работах [5-9] предложены усовершенствованные математические модели и алгоритмы многопутевой маршрутизации в телекоммуникационных сетях с балансировкой нагрузки. Вопросы исследования задержки передачи в ПКС и оценки масштабирования контроллера ПКС подробно рассмотрены в работах [10-12]. Анализ метрик качества сервиса в ПКС детально исследован в работе [13]. Поточковые модели многопутевой маршрутизации с поддержкой различных классов качества сервиса рассмотрены в работах [14, 15]. Концептуальный подход динамического формирования трафика и балансировки потоков данных в ПКС с обеспечением качества сервиса подробно рассмотрен в работах [16, 17]. Программная инфраструктура и визуальная среда распределенной обработки потоков данных в ПКС предложена в работе [18]. Задача динамического управления трафиком ПКС в облачной инфраструктуре центров обработки данных (ЦОД) рассмотрена в работах [19-21]. Программная система многопутевой маршрутизации в ПКС распределенных ЦОД предложена в работе [22]. В работе [23] предложены балансировщики нагрузки с поддержкой многопутевой маршрутизации для передачи сетевого трафика в ПКС. В работе [24] для минимизации перегрузки сети предложено использовать схему многопутевой маршрутизации на основе классического подхода. В работе [25] предложен алгоритм многопутевой маршрутизации, позволяющий повысить производительность сети за счет уменьшения объема служебных пакетов. В работе [26] использование многопутевой маршрутизации позволяет быстро выполнить перемаршрутизацию в ПКС при отказе каналов связи или сетевых коммутаторов. В работе [27] предложено использовать генетический алгоритм для решения задачи многопутевой маршрутизации с ограниченной полосой пропускания. В данной статье контроллер ПКС вызывает алгоритм вычисления кратчайших путей, отвечающих требованиям полосы пропускания, только после получения управляющей информации об имеющихся ресурсах в сети. В работе [28] также предложено использовать генетический алгоритм для поиска кратчайших путей от одного источника ко многим узлам назначения в сети. Таким образом, актуальной научной задачей является исследование и анализ процессов многопутевой маршрутизации и балансировки потоков данных в ПКС на основе эволюционных методов и алгоритмов.

### Исследование работы модифицированного генетического алгоритма

Генетический алгоритм (ГА) – это метод, имитирующий эволюционную адаптацию биологических популяций, основанный на теории Дарвина. ГА – это метод случайного оптимального поиска, моделирующий эволюцию человека или организмов. Идея генетического алгоритма заключается в моделировании природных явлений, наследовании и борьбе за выживание. Раз-

ные виды, а также особи одного и того же вида сосуществуют или конкурируют друг с другом за ресурсы. Виды, способные лучше адаптироваться, получают ресурсы и производят потомство на будущее. Виды, недостаточно хорошо адаптировавшиеся, по каким-то причинам постепенно уменьшаются в численности или не имеют потомства. Блок-схема модифицированного генетического алгоритма показана на рисунке 1.

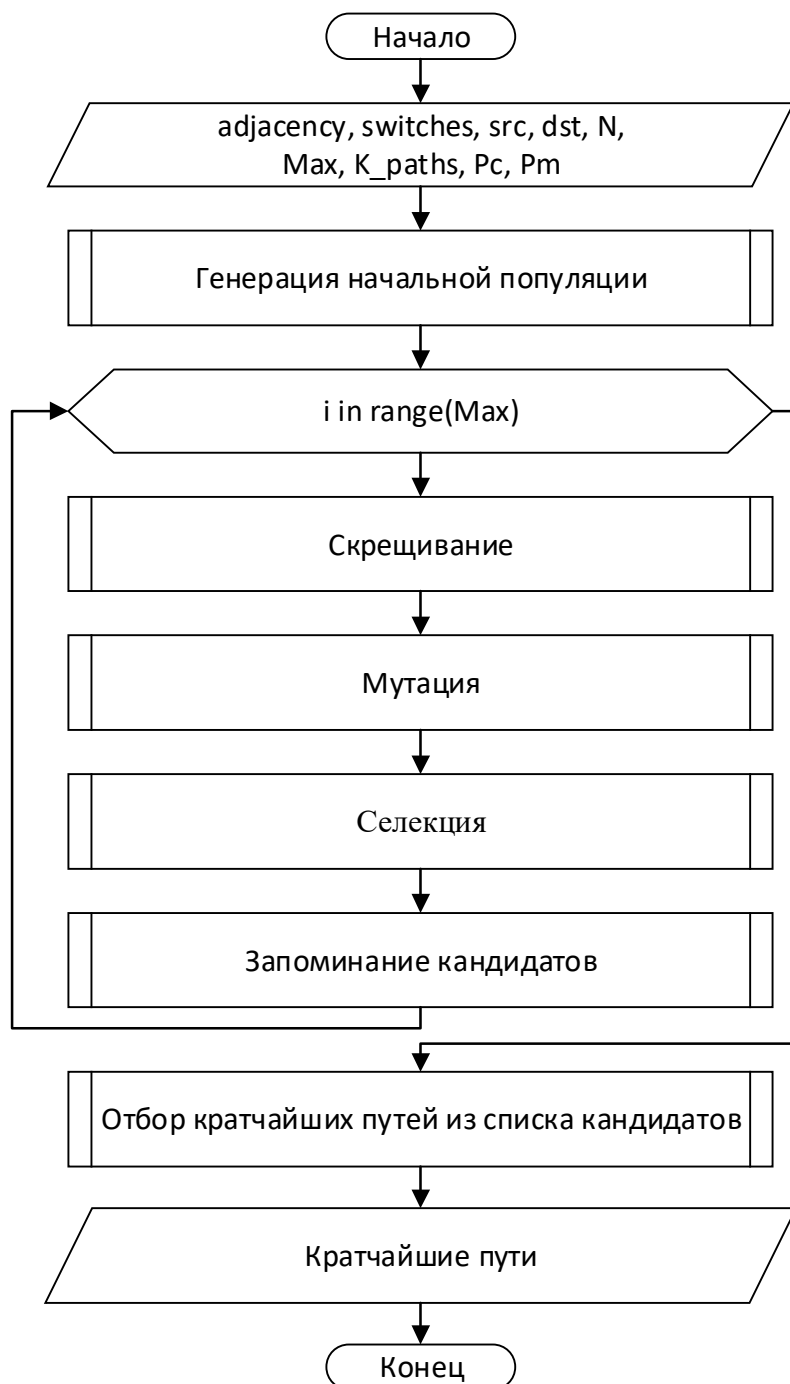


Рисунок 1 – Блок-схема модифицированного генетического алгоритма  
Figure 1 – Flow chart of modified genetic algorithm

а) генерация начальной популяции

Начальная популяция состоит из набора хромосом. Каждая хромосома – это решение проблемы. В задаче маршрутизации каждой хромосоме соответствует маршрут. В данной работе используется метод прямого кодирования пути с нефиксированной длиной. Каждый ген в хромосоме соответствует узлам, через которые проходит маршрут. Первый ген всегда является исходным узлом, а последний ген – целевым узлом.

Например: хромосома  $C$  обозначается следующим образом:

$$[v_8, v_4, v_3, v_7, v_6, v_5],$$

где  $v_8$  – исходный узел и  $v_5$  – целевой узел.

б) оценка приспособленности

Жизнеспособность каждой хромосомы оценивали по приспособленности. Особи с высокой приспособленностью, скорее всего, будут оставлены для потомства, а менее приспособленные выброшены. В задаче маршрутизации, приспособленность – это длина каждого маршрута. Функция приспособленности определяется как:

$$f(C_i) = w(C_i),$$

где  $w(C_i)$  вычисляет значение длины пути соответствующей хромосомы  $C_i$ .

с) скрещивание

Скрещивание – это процесс получения потомства путем обмена частичными генами двух родительских хромосом. Существует несколько различных методов скрещивания, таких как: одноточечный, многоточечный и равномерный. В данной работе используется метод одноточечного скрещивания. Случайно выбранные родители должны удовлетворять условию наличия хотя бы одного общего гена, кроме первого и последнего генов.

Например:

Пусть двумя выбранными родительскими хромосомами для скрещивания будут  $C_f = [v_8, v_4, v_3, v_7, v_6, v_5]$  и  $C_m = [v_8, v_2, v_{10}, v_7, v_9, v_{11}, v_5]$  соответственно. Точка скрещивания –  $v_7$ .

Точка скрещивания делит каждую родительскую хромосому на две части. Скрещивая эти две части между собой, получают потомства  $C_{s1}$  и  $C_{s2}$ .

$$C_{s1} = [v_8, v_4, v_3, v_7, v_9, v_{11}, v_5],$$

$$C_{s2} = [v_8, v_2, v_{10}, v_7, v_6, v_5].$$

В процессе скрещивания могут образовываться нежелательные потомства, соответствующие путям с петлей. В этом случае необходимо удалить петлю для получения желательных подпоколений.

Например:

$$C_s = [v_8, v_4, v_3, v_7, v_9, v_{11}, v_7, v_{10}, v_{13}, v_5],$$

$$\Rightarrow C_s = [v_8, v_4, v_3, v_7, v_{10}, v_{13}, v_5].$$

д) мутация

Мутация – это процесс создания новых особей из потомства, полученного в процессе гибридизации. В генетических алгоритмах мутация направлена на поиск новых решений, позволяющих избежать локальной оптимизации. Также существуют различные методы мутации, такие как: одноточечный и многоточечный. В данной работе используется метод одноточечной мутации. Случайным образом выбирается ген в качестве точки мутации. Эта точка мутации разделяет родительскую хромосому на две части. Первая часть будет сохранена. Вторая часть обновится так же, как и при инициализации хромосомы.

Например:

Пусть выбранной хромосомой для мутации будет  $C_m = [v_8, v_9, v_{13}, v_{10}, v_6, v_5]$ . Точка мутации –  $v_{10}$ . Тогда новая хромосома, созданная оператором мутации, будет:

$$C_s = [v_8, v_9, v_{13}, v_{10}, v_7, v_1, v_{12}, v_5].$$

е) селекция

Селекция – это процесс отбора хромосом, которые хорошо адаптированы для создания следующего поколения. Существует множество различных методов отбора, таких как: турнирная селекция, метод рулетки, метод ранжирования, сигма-отсечение и т.д.

В данной работе используется турнирная селекция. Случайным образом выбирается определенное количество хромосом. Хромосома с лучшей приспособленностью среди них

будет добавлен к следующему поколению. Процесс продолжается до тех пор, пока не будет достигнута желаемая численность популяции.

f) запоминание кандидатов

Для задачи многопутевой маршрутизации, чтобы после завершения алгоритма всегда получались  $k$  различных маршрутов, на каждой итерации необходимо сохранять  $k$  различных лучших хромосом в популяции и добавлять их в список кандидатов.

g) отбор кратчайших путей из списка кандидатов

После выполнения заданного количества итераций получается список кандидатов. Из этого списка выбираются лучшие  $k$  хромосом, соответствующие  $k$  кратчайшим путям, полученным предложенным алгоритмом.

### Экспериментальное исследование

Для подтверждения эффективности применения модифицированного генетического алгоритма для решения задачи многопутевой маршрутизации и балансировки потоков данных в ПКС разработана визуальная программная система SDNLoadBalancer. В программной системе проектируется карта сети и генерируется сценарий на языке Python, запускающий эмулятор MiniNet со спроектированной сетевой топологией. Для исследования процессов многопутевой маршрутизации и балансировки потоков данных в ПКС на основе генетического алгоритма рассматривается экспериментальная топология ПКС (рисунок 2). Исходный узел-отправитель пакетов данных – коммутатор S1, а узел-получатель пакетов данных – коммутатор S19.

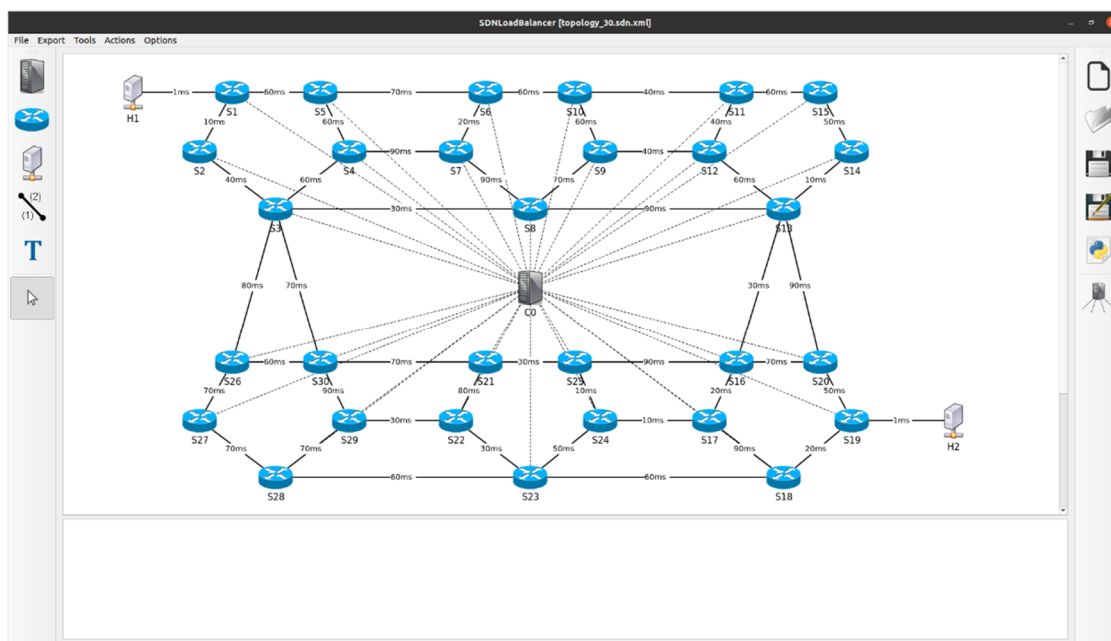


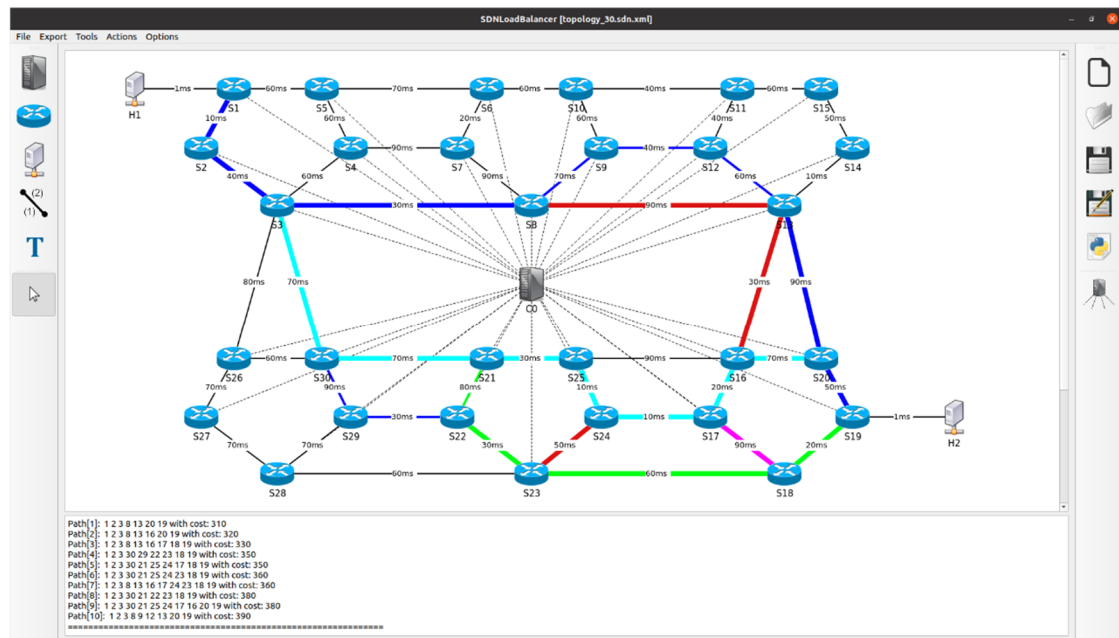
Рисунок 2 – Экспериментальная топология ПКС  
Figure 2 – SDN Experimental topology

Для проверки эффективности работы модифицированного генетического алгоритма при поиске кратчайших путей, полученные результаты будут сравниваться с результатами классических алгоритмов. В данной статье для сравнения используется алгоритм Йена. В таблице 1 показаны кратчайшие пути, найденные алгоритмом Йена.

Результаты работы модифицированного генетического алгоритма в визуальной среде SDNLoadBalancer приведены на рисунке 3. Полученные маршруты отмечены разными цветами.

**Таблица 1 – Результат работы алгоритма Йена**  
**Table 1 – The result of Yen's algorithm**

№	Кратчайшие пути	Стоимость (вес)
1	1 – 2 – 3 – 8 – 13 – 20 – 19	310
2	1 – 2 – 3 – 8 – 13 – 16 – 20 – 19	320
3	1 – 2 – 3 – 8 – 13 – 16 – 17 – 18 – 19	330
4	1 – 2 – 3 – 30 – 21 – 25 – 24 – 17 – 18 – 19	350
5	1 – 2 – 3 – 30 – 29 – 22 – 23 – 18 – 19	350
6	1 – 2 – 3 – 8 – 13 – 16 – 17 – 24 – 23 – 18 – 19	360
7	1 – 2 – 3 – 30 – 21 – 25 – 24 – 23 – 18 – 19	360
8	1 – 2 – 3 – 30 – 21 – 22 – 23 – 18 – 19	380
9	1 – 2 – 3 – 30 – 21 – 25 – 24 – 17 – 16 – 20 – 19	380
10	1 – 2 – 3 – 8 – 9 – 12 – 13 – 20 – 19	390
		<b>Общая стоимость (вес) всех путей = 3530</b>



**Рисунок 3 – Полученные маршруты для рассматриваемой топологии ( $N = 50$ ;  $Max = 25$ )**

**Figure 3 – Obtained routes for the considered topology ( $N = 50$ ;  $Max = 25$ )**

Для проверки работы модифицированного генетического алгоритма необходимо запустить эмулятор MiniNet и контроллер OpenFlow. В результате в командном интерфейсе MiniNet появятся сообщения об успешной доставке пакетов.

```
mininet> h1 ping -c8 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data:
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=632 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=627 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=627 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=627 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=626 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=626 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=627 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=627 ms

--- 10.0.0.2 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7006ms
rtt min/avg/max/mdev = 626.068/627.273/632.141/1.861 ms
mininet>
```

**Рисунок 4 – Сообщения об успешной доставке пакетов в MiniNet**  
**Figure 4 – Messages about the successful delivery of packages in MiniNet**

В рисунках 5-7 и 11-13 приведены результаты работы модифицированного генетического алгоритма для рассматриваемой топологии ПКС.

Parameters	K_paths	CD	Ds	I, %	AV	SD	MxVL	MnVL	J, %
N = 10 Max = 25 Pc = 0.9 Pm = 0.9	10	3610	310	11.6	51.7	29.7	90	10	24.39
			320	11.2	45.7	25	90	10	
			330	10.9	41.3	29.3	90	10	
			350	10.2	43.8	25.5	90	10	
			360	9.9	36	23.7	90	10	
			360	9.9	40	22.6	70	10	
			380	9.4	47.5	24.4	80	10	
			390	9.2	48.8	23.2	90	10	
			400	9	44.4	18.9	70	10	
			410	8.7	51.3	23.7	80	10	
N = 25 Max = 25 Pc = 0.9 Pm = 0.9	10	3550	310	11.4	51.7	29.7	90	10	22.5
			320	11	45.7	25	90	10	
			330	10.7	41.3	29.3	90	10	
			350	10.1	43.8	25.5	90	10	
			350	10.1	38.9	28.8	90	10	
			360	9.8	40	22.6	70	10	
			360	9.8	36	23.7	90	10	
			380	9.3	47.5	24.4	80	10	
			390	9	48.8	23.2	90	10	
			400	8.8	44.4	18.9	70	10	
N = 50 Max = 25 Pc = 0.9 Pm = 0.9	10	3530	310	11.3	51.7	29.7	90	10	20.51
			320	11	45.7	25	90	10	
			330	10.6	41.3	29.3	90	10	
			350	10	38.9	28.8	90	10	
			350	10	43.8	25.5	90	10	
			360	9.8	40	22.6	70	10	
			360	9.8	36	23.7	90	10	
			380	9.2	38	24.4	70	10	
			380	9.2	47.5	24.4	80	10	
			390	9.1	48.8	23.2	90	10	

**Рисунок 5 – Результат работы генетического алгоритма**  
 ( $N = [10; 25; 50]$ ;  $Max = 25$ ;  $P_c = 0,9$ ;  $P_m = 0,9$ ;  $K_{paths} = 10$ )

**Figure 5 – The result of genetic algorithm**  
 ( $N = [10; 25; 50]$ ;  $Max = 25$ ;  $P_c = 0,9$ ;  $P_m = 0,9$ ;  $K_{paths} = 10$ )

На данных рисунках используются следующие обозначения: Parameters – параметры алгоритма;  $K_{paths}$  – число кратчайших путей, CD – общая длина, т.е. общая маршрутная метрика всех доступных маршрутов между коммутатором-источником и коммутатором-получателем, Ds – длина маршрута, I – доля информации, проходящей через маршрут, AV – среднее значение каналов, входящих в маршрут, SD – квадратичное отклонение каналов, входящих в маршрут, MxVL – максимальное значение канала в маршруте, MnVL – минимальное значение канала в маршруте, J – отклонение значения длины текущего маршрута от длины оптимального маршрута.

Из рисунка 5 видно, что при поиске 10 кратчайших путей, когда  $N = 50$ , получены следующие результаты:  $J = 20,51\%$  и  $CD = 3530$ . По сравнению с результатами алгоритма Йена (таблица 1) этот результат является оптимальным. При значениях  $N = 10$  и  $N = 25$  полученные результаты оказываются ниже.

Parameters	K_paths	CD	Ds	I, %	AV	SD	MxVL	MnVL	J, %
N = 10 Max = 25 Pc = 0.9 Pm = 0.9	8	2800	310	14	51.7	29.7	90	10	20.51
			320	13.6	45.7	25	90	10	
			330	13.2	41.3	29.3	90	10	
			350	12.4	43.8	25.5	90	10	
			360	12.1	36	23.7	90	10	
			360	12.1	40	22.6	70	10	
			380	11.4	47.5	24.4	80	10	
			390	11.2	48.8	23.2	90	10	
N = 25 Max = 25 Pc = 0.9 Pm = 0.9	8	2760	310	13.9	51.7	29.7	90	10	18.42
			320	13.4	45.7	25	90	10	
			330	13	41.3	29.3	90	10	
			350	12.3	43.8	25.5	90	10	
			350	12.3	38.9	28.8	90	10	
			360	11.9	40	22.6	70	10	
			360	11.9	36	23.7	90	10	
			380	11.3	47.5	24.4	80	10	
N = 50 Max = 25 Pc = 0.9 Pm = 0.9	8	2760	310	13.9	51.7	29.7	90	10	18.42
			320	13.4	45.7	25	90	10	
			330	13	41.3	29.3	90	10	
			350	12.3	38.9	28.8	90	10	
			350	12.3	43.8	25.5	90	10	
			360	11.9	40	22.6	70	10	
			360	11.9	36	23.7	90	10	
			380	11.3	38	24.4	70	10	

K\_paths = 8

Рисунок 6 – Результат работы генетического алгоритма  
( $N = [10; 25; 50]$ ;  $Max = 25$ ;  $P_c = 0,9$ ;  $P_m = 0,9$ ;  $K_{paths} = 8$ )

Figure 6 – The result of genetic algorithm  
( $N = [10; 25; 50]$ ;  $Max = 25$ ;  $P_c = 0,9$ ;  $P_m = 0,9$ ;  $K_{paths} = 8$ )

При поиске 8 кратчайших путей при  $N = 50$  и  $N = 25$  получены одинаковые результаты, которые являются оптимальными:  $J = 18,42\%$  и  $CD = 2760$ . При  $N = 10$ , полученные результаты несколько ниже.



Result of evolutionary algorithm										
Parameters	K_paths	CD	Ds	I, %	AV	SD	MxVL	MnVL	J, %	
N = 10 Max = 25 Pc = 0.9 Pm = 0.9	6	2030	310	18.1	51.7	29.7	90	10	13.89	
			320	17.6	45.7	25	90	10		
			330	17	41.3	29.3	90	10		
			350	16.1	43.8	25.5	90	10		
			360	15.6	36	23.7	90	10		
			360	15.6	40	22.6	70	10		
N = 25 Max = 25 Pc = 0.9 Pm = 0.9	6	2020	310	18	51.7	29.7	90	10	13.89	
			320	17.5	45.7	25	90	10		
			330	17	41.3	29.3	90	10		
			350	16	43.8	25.5	90	10		
			350	16	38.9	28.8	90	10		
			360	15.5	40	22.6	70	10		
N = 50 Max = 25 Pc = 0.9 Pm = 0.9	6	2020	310	18	51.7	29.7	90	10	13.89	
			320	17.5	45.7	25	90	10		
			330	17	41.3	29.3	90	10		
			350	16	38.9	28.8	90	10		
			350	16	43.8	25.5	90	10		
			360	15.5	40	22.6	70	10		

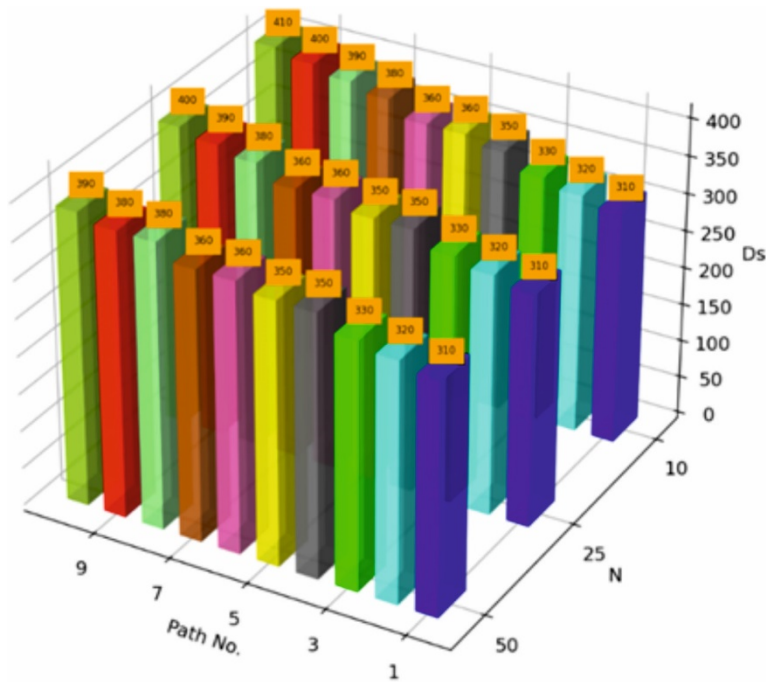
K\_paths =

**Рисунок 7 – Результат работы генетического алгоритма**  
 ( $N = [10; 25; 50]$ ;  $Max = 25$ ;  $P_c = 0,9$ ;  $P_m = 0,9$ ;  $K_{paths} = 6$ )

**Figure 7 – The result of genetic algorithm**  
 ( $N = [10; 25; 50]$ ;  $Max = 25$ ;  $P_c = 0,9$ ;  $P_m = 0,9$ ;  $K_{paths} = 6$ )

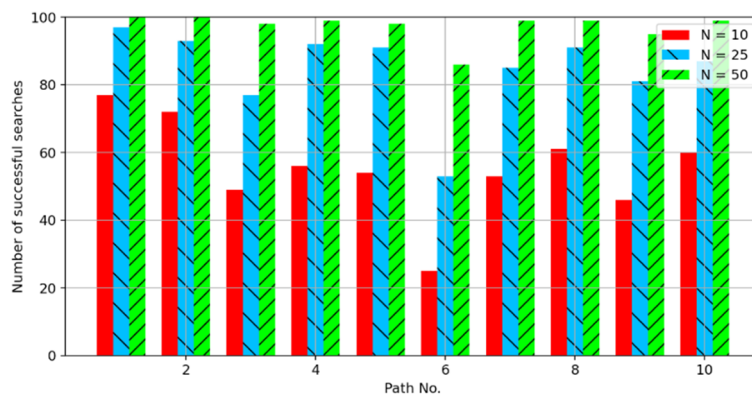
При поиске 6 кратчайших путей, во всех трех случаях, для  $N = 50$ ,  $N = 25$  и  $N = 10$  получаются одинаковые результаты: джиттер  $J = 13,89$ . Однако при  $N = 10$ , полученное значение  $CD$  хуже, чем в двух других случаях ( $2030 > 2020$ ).

Диаграмма длин всех маршрутов для всех трех рассмотренных случаев приведена на рисунке 8.

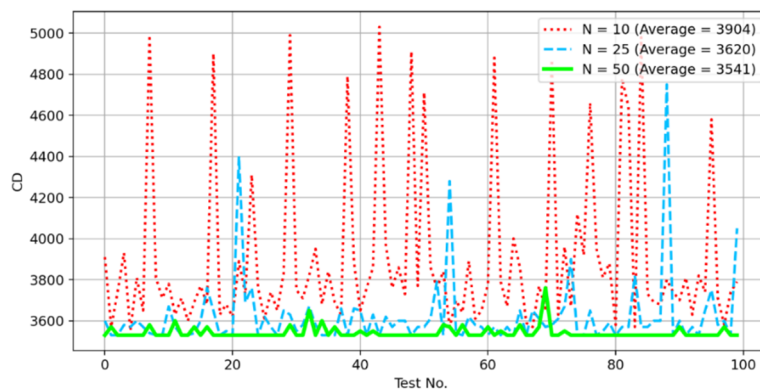


**Рисунок 8 – Диаграмма отображения длины каждого пути**  
**Figure 8 – Chart displaying the length of each path**

Из-за случайности генетического алгоритма для оценки влияния параметров на полученные результаты предложенный модифицированный алгоритм запускался 100 раз. Далее проводились сравнения, сколько раз каждый маршрут из таблицы 1 был найден предлагаемым алгоритмом.



**Рисунок 9 – Диаграмма сравнения количества успешного поиска для каждого маршрута**  
**Figure 9 – Chart comparing the number of successful searches for each route**



**Рисунок 10 – График отображения значения CD для каждого запуска**  
**Figure 10 – Plot showing the value of CD for each run**

Рисунок 9 показывает, что при  $N = 50$  количество правильных вычислений каждого маршрута больше, чем при  $N = 10$  и  $N = 25$ . Из рисунка 10 видно, что при  $N = 50$  полученные результаты стабильны и почти оптимальны во всех запусках.

Parameters	K_paths	CD	Ds	I, %	AV	SD	MxVL	MnVL	J, %
N = 50 Max = 25 Pc = 0.9 Pm = 0.1	10	3800	310	12.1	51.7	29.7	90	10	32.61
			320	11.7	45.7	25	90	10	
			330	11.3	41.3	29.3	90	10	
			350	10.7	43.8	25.5	90	10	
			380	9.8	47.5	24.4	80	10	
			390	9.6	48.8	23.2	90	10	
			410	9.1	51.3	23.7	80	10	
			420	8.9	46.7	25.8	90	10	
			430	8.7	53.8	24.5	90	10	
			460	8.1	51.1	21.3	70	10	
N = 50 Max = 25 Pc = 0.9 Pm = 0.5	10	3670	310	11.7	51.7	29.7	90	10	26.19
			320	11.4	45.7	25	90	10	
			330	11	41.3	29.3	90	10	
			350	10.4	38.9	28.8	90	10	
			360	10.1	36	23.7	90	10	
			380	9.6	47.5	24.4	80	10	
			390	9.3	48.8	23.2	90	10	
			400	9.1	44.4	18.9	70	10	
			410	8.9	51.3	23.7	80	10	
			420	8.5	42	29.3	90	10	
N = 50 Max = 25 Pc = 0.9 Pm = 0.9	10	3530	310	11.3	51.7	29.7	90	10	20.51
			320	11	45.7	25	90	10	
			330	10.6	41.3	29.3	90	10	
			350	10	43.8	25.5	90	10	
			350	10	38.9	28.8	90	10	
			360	9.8	36	23.7	90	10	
			360	9.8	40	22.6	70	10	
			380	9.2	38	24.4	70	10	
			380	9.2	47.5	24.4	80	10	
			390	9.1	48.8	23.2	90	10	

K\_paths =

**Рисунок 11 – Результат работы генетического алгоритма**  
 ( $N = 50; Max = 25; P_c = 0,9; P_m = [0, 1; 0, 5; 0, 9]; K_{paths} = 10$ )

**Figure 11 – The result of genetic algorithm**  
 ( $N = 50; Max = 25; P_c = 0,9; P_m = [0, 1; 0, 5; 0, 9]; K_{paths} = 10$ )

Из приведенного рисунка видно, что при необходимости вычисления 10 кратчайших маршрутов, когда  $P_m = 0,9$  алгоритм получал более оптимальные результаты:  $J = 20,51\%$  и  $CD = 3530$ . А при показателях  $P_m = 0,1$  и  $P_m = 0,5$  алгоритм работал менее эффективно.

Result of evolutionary algorithm										
Parameters	K_paths	CD	Ds	I, %	AV	SD	MxVL	MnVL	J, %	
N = 50 Max = 25 Pc = 0.9 Pm = 0.1	8	2910	310	14.5	51.7	29.7	90	10	26.19	
			320	14	45.7	25	90	10		
			330	13.6	41.3	29.3	90	10		
			350	12.8	43.8	25.5	90	10		
			380	11.8	47.5	24.4	80	10		
			390	11.5	48.8	23.2	90	10		
			410	11	51.3	23.7	80	10		
			420	10.8	46.7	25.8	90	10		
N = 50 Max = 25 Pc = 0.9 Pm = 0.5	8	2840	310	14.2	51.7	29.7	90	10	22.5	
			320	13.8	45.7	25	90	10		
			330	13.3	41.3	29.3	90	10		
			350	12.6	38.9	28.8	90	10		
			360	12.2	36	23.7	90	10		
			380	11.6	47.5	24.4	80	10		
			390	11.3	48.8	23.2	90	10		
			400	11	44.4	18.9	70	10		
N = 50 Max = 25 Pc = 0.9 Pm = 0.9	8	2760	310	13.9	51.7	29.7	90	10	18.42	
			320	13.4	45.7	25	90	10		
			330	13	41.3	29.3	90	10		
			350	12.3	43.8	25.5	90	10		
			350	12.3	38.9	28.8	90	10		
			360	11.9	36	23.7	90	10		
			360	11.9	40	22.6	70	10		
			380	11.3	38	24.4	70	10		

K\_paths =

**Рисунок 12 – Результат работы генетического алгоритма**  
 ( $N = 50; Max = 25; P_c = 0,9; P_m = [0,1;0,5;0,9]; K_{paths} = 8$ )

**Figure 12 – The result of genetic algorithm**  
 ( $N = 50; Max = 25; P_c = 0,9; P_m = [0,1;0,5;0,9]; K_{paths} = 8$ )

Аналогично при необходимости вычисления 8 кратчайших путей, когда  $P_m = 0,9$  полученные результаты были оптимальны:  $J = 18,42\%$  и  $CD = 2760$ . Также стоит отметить, что при  $P_m = 0,5$  алгоритм показывал лучшие результаты, чем при  $P_m = 0,1$  ( $J = 22,5\% < J = 26,19\%$ ;  $CD = 2840 < CD = 2910$ ).

Result of evolutionary algorithm										
Parameters	K_paths	CD	Ds	I, %	AV	SD	MxVL	MnVL	J, %	
N = 50 Max = 25 Pc = 0.9 Pm = 0.1	6	2080	310	18.5	51.7	29.7	90	10	20.51	
			320	17.9	45.7	25	90	10		
			330	17.4	41.3	29.3	90	10		
			350	16.4	43.8	25.5	90	10		
			380	15.1	47.5	24.4	80	10		
			390	14.7	48.8	23.2	90	10		
N = 50 Max = 25 Pc = 0.9 Pm = 0.5	6	2050	310	18.3	51.7	29.7	90	10	18.42	
			320	17.7	45.7	25	90	10		
			330	17.2	41.3	29.3	90	10		
			350	16.2	38.9	28.8	90	10		
			360	15.7	36	23.7	90	10		
			380	14.9	47.5	24.4	80	10		
N = 50 Max = 25 Pc = 0.9 Pm = 0.9	6	2020	310	18	51.7	29.7	90	10	13.89	
			320	17.5	45.7	25	90	10		
			330	17	41.3	29.3	90	10		
			350	16	43.8	25.5	90	10		
			350	16	38.9	28.8	90	10		
			360	15.5	36	23.7	90	10		

K\_paths =

**Рисунок 13 – Результат работы генетического алгоритма**  
 ( $N = 50; Max = 25; P_c = 0,9; P_m = [0,1;0,5;0,9]; K_{paths} = 6$ )

**Figure 13 – The result of genetic algorithm**  
 ( $N = 50; Max = 25; P_c = 0,9; P_m = [0,1;0,5;0,9]; K_{paths} = 6$ )

При вычислении 6 кратчайших путей, показатель  $P_m = 0,9$  алгоритм также показывал лучшие результаты, чем  $P_m = 0,1$  и  $P_m = 0,5$  ( $J = 13,89\% < J = 18,42\% < J = 20,51\%$  и  $CD = 2020 < CD = 2050 < CD = 2080$ ).

Диаграмма длин всех маршрутов для всех трех рассмотренных случаев приведена на рисунке 14.

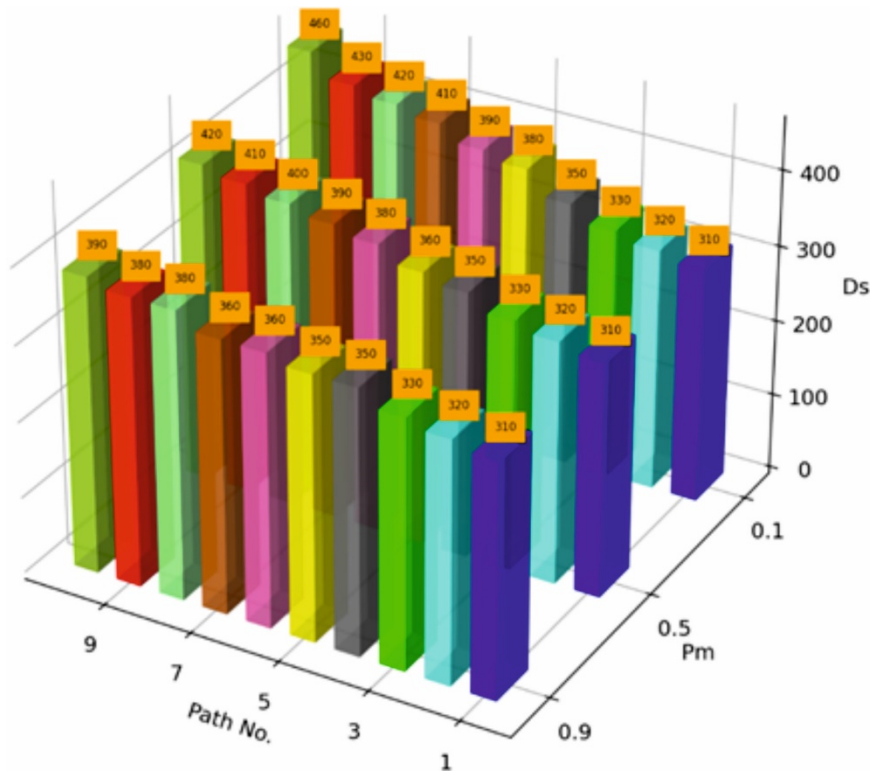


Рисунок 14 – Диаграмма отображения длины каждого маршрута  
Figure 14 – Chart displaying the length of each path

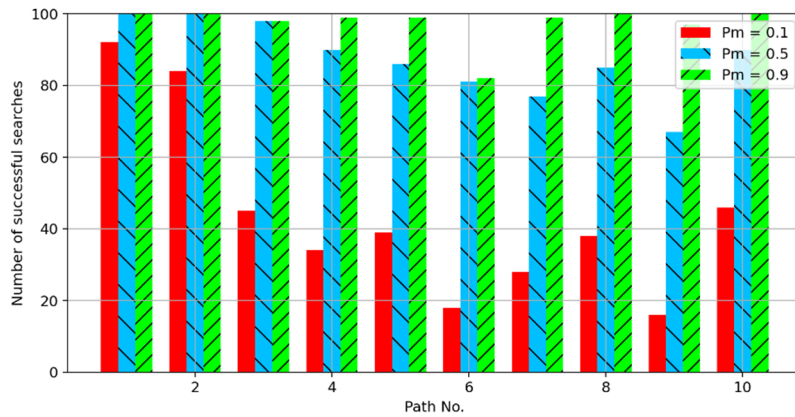


Рисунок 15 – Диаграмма сравнения количества успешного поиска для каждого маршрута  
Figure 15 – Chart comparing the number of successful searches for each route

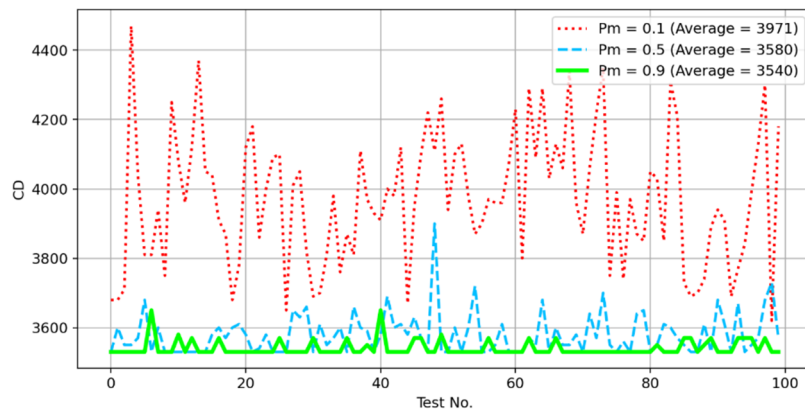


Рисунок 16 – График отображения значения CD для каждого запуска  
Figure 16 – Plot showing the value of CD for each run

Рисунок 15 показывает, что при значении  $P_m = 0,9$  количество успешных результатов поиска маршрутов больше, чем при  $P_m = 0,1$  и  $P_m = 0,5$ . Из рисунка 16 видно, что при  $P_m = 0,9$  полученные результаты более стабильны и почти оптимальны во всех запусках.

Экспериментальные исследования показали, что чем больше размер популяции или выше вероятность мутации, тем лучше результаты работы модифицированного генетического алгоритма. Это объясняется тем, что чем больше размер популяции, тем выше вероятность получения хороших особей. Высокая вероятность мутации повышает разнообразие популяции, что необходимо для решения задачи многопутевой маршрутизации и балансировки потоков данных, а также не влияет на сходимость, как в классическом генетическом алгоритме.

### Заключение

В работе проведены исследование и анализ процессов многопутевой маршрутизации и балансировки потоков данных в программно-конфигурируемых сетях на основе модифицированного генетического алгоритма. Для представления хромосом в работе использовался метод прямого кодирования. Каждая хромосома представляла собой последовательность целых чисел, соответствующих узлам, через которые проходит маршрут. В работе также выполнено сравнение предложенного модифицированного генетического алгоритма с классическим алгоритмом Йена для решения задачи многопутевой маршрутизации и балансировки потоков данных в ПКС. Результаты проведенного исследования показали, что модифицированный генетический алгоритм успешно решает задачу поиска -кратчайших путей при подходящем подборе параметров генетического алгоритма. Для подтверждения эффективности предложенного подхода разработана программная система SDNLoadBalancer, позволяющая детально исследовать и проектировать процессы многопутевой маршрутизации и балансировки нагрузки в ПКС.

*Работа выполнена при финансовой поддержке гранта Президента РФ МД-3201.2022.1.6.*

### Библиографический список

1. McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., Turner J. Openflow: Enabling Innovation in Campus Networks. Proc. ACM SIGCOMM Computer Communication Review. 2008, vol. 38, no. 2, pp. 69-74.
2. Hilmi E. E., Adaptive Video Streaming over OpenFlow Networks with Quality of Service. Thesis for Degree of Master Science in Electrical and Electronics Engineering, Koc University, July 2012. 91 p.
3. Смелянский Р. Л. Программно-конфигурируемые сети // Открытые системы. СУБД. 2012. Т. 9. С. 23-26.
4. Корячко В. П., Перепелкин Д. А. Программно-конфигурируемые сети. Учебник для вузов. М.: Горячая линия – Телеком, 2020. 288 с.
5. Лемешко А. В., Вавенко Т. В. Усовершенствование потоковой модели многопутевой маршрутизации на основе балансировки нагрузки // Проблемы телекоммуникаций. 2012. № 1 (6). С. 12-29.
6. Корячко В. П., Перепелкин Д. А. Разработка и исследование математической модели многопутевой адаптивной маршрутизации в сетях связи с балансировкой нагрузки // Электросвязь. 2014. № 12. С. 27-31.
7. Перепелкин Д. А. Математическая модель многопутевой адаптивной маршрутизации с балансировкой неоднородной нагрузки в условиях динамических подключений узлов и линий связи в телекоммуникационных сетях // Радиотехника. 2015. № 5. С. 46-54.
8. Koryachko V. P., Perepelkin D. A., Byshov V. S. Improved Multipath Adaptive Routing Model in Computer Networks with Load Balancing. Proceedings SIBCON 2016. IEEE 2016 International Siberian Conference on Control and Communications, 2016, pp. 1-4. DOI: 10.1109/SIBCON.2016.7491721.
9. Koryachko V.P., Perepelkin D.A., Byshov V.S. Development and Research of Improved Model of Multipath Adaptive Routing in Computer Networks with Load Balancing. Automatic Control and Computer Sciences. 2017, vol. 51, no. 1, pp. 63-73. DOI: 10.3103/S0146411617010047.
10. Shalimov A. et al. Advanced study of SDN/OpenFlow controllers. Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia. ACM, 2013.



11. **Карташевский В. Г., Галич С. В., Семёнов Е. С., Кирьянова Н. И.** Оценка масштабируемости задержки ПКС-контроллера на параллельной вычислительной системе // Инфокоммуникационные технологии. 2017. Т. 15. № 2. С. 163-170.
12. **Малахов С. В., Тарасов В. Н., Карташевский И. В.** Теоретическое и экспериментальное исследование задержки в программно-конфигурируемых сетях // Инфокоммуникационные технологии. 2015. Т. 13. № 4. С. 409-413.
13. **Koryachko V. P., Perepelkin D. A., Ivanchikova M. A., Byshov V. S., Tsyganov I. Yu.** Analysis QoS Metrics in Software Defined Networks. Proceedings MECO 2017 – IEEE 6th Mediterranean Conference on Embedded Computing (MECO-2017), 2017, pp. 374-378. DOI: 10.1109/MECO.2017.7977240.
14. **Лемешко А. В., Вавенко Т. В.** Разработка и исследование потоковой модели адаптивной маршрутизации в программно-конфигурируемых сетях с балансировкой нагрузки // Доклады Томского государственного университета систем управления и радиоэлектроники. 2013. № 3 (29). С. 100-108.
15. **Lemeshko O. V., and Yeremenko O. S.** Dynamics analysis of multipath QoS-routing tensor model with support of different flows classes. Proc. of the IEEE International Smart Systems and Technologies (SST-2016), 2016, pp. 225-230.
16. **Перепелкин Д. А.** Концептуальный подход динамического формирования трафика программно-конфигурируемых телекоммуникационных сетей с балансировкой нагрузки // Информационные технологии. 2015. Т. 21. № 8. С. 602-610.
17. **Перепелкин Д. А., Бышов В. С.** Балансировка потоков данных в программно-конфигурируемых сетях с обеспечением качества обслуживания сетевых сервисов // Радиотехника. 2016. № 11. С. 111-119.
18. **Корячко В. П., Перепелкин Д. А., Иванчикова М. А., Бышов В. С., Цыганов И. Ю.** Программная инфраструктура и визуальная среда распределенной обработки потоков данных в программно-конфигурируемых сетях // Вестник Рязанского государственного радиотехнического университета. 2018. № 65. С. 44-54. DOI: 10.21667/1995-4565-2018-65-3-44-54.
19. **Никульчев Е. В., Паяин С. В., Плужник Е. В.** Динамическое управление трафиком программно-конфигурируемых сетей в облачной инфраструктуре // Вестник Рязанского государственного радиотехнического университета. 2013. № 3 (45). С. 54-57.
20. **Леохин Ю. Л., Фатхулин Т. Д.** Оценка возможности предоставления гарантированной скорости передачи данных в программно-конфигурируемой оптической сети // Вестник Рязанского государственного радиотехнического университета. 2020. № 71. С. 45-59. DOI: 10.21667/1995-4565-2020-71-45-59.
21. **Ушакова М. В., Ушаков Ю. А.** Исследование сети виртуальной инфраструктуры центра обработки данных с гибридной программно-конфигурируемой коммутацией // Вестник Рязанского государственного радиотехнического университета. 2021. № 75. С. 34-43. DOI: 10.21667/1995-4565-2021-75-34-43.
22. **Koryachko V., Perepelkin D., Ivanchikova M.** Software System of Multipath Adaptive Routing in Networks of Distributed Data Centers, Proceedings MECO 2019 – 8<sup>th</sup> Mediterranean Conference on Embedded Computing (MECO), 2019, DOI: 10.1109/MECO.2019.8759999.
23. **E. Jo, Deng Pan, J. Liu, and L. Butler.** A Simulation and Emulation Study of SDN-Based Multipath routing for Fat-Tree Data Center Networks, in Proceeding Winter Simulation Conference, 2014.
24. **Ron Banner and Ariel Orda.** Multipath routing algorithms for congestion minimization, in IEEE/ACM Transactions on Networking, 2007, pp. 413-424.
25. **K. Rajasekaran, Kannan Balasubramanian.** Energy Conscious based Multipath Routing Algorithm in WSN, in International Journal of Computer Network and Information Security (IJCNIS), vol. 8, no.1, pp.27-34, 2016.DOI: 10.5815/ijcnis.2016.01.04.
26. **Meenakshi Moza, Suresh Kumar.** Analyzing Multiple Routing Configuration, in International Journal of Computer Network and Information Security (IJCNIS), vol.8, no.5, pp.48-54, 2016.DOI: 10.5815/ijcnis.2016.05.07.
27. **Yilan Liu, Yun Pan, Muxi Yang, Wenqing Wang, Chi Fang, Ruijuan Jiang.** The Multi-Path Routing Problem in the Software Defined Network, in 11th International Conference on Natural Computation (ICNC), 2015, DOI: 10.1109/ICNC.2015.7377999.
28. **Ahmed Younes Hamed.** A Genetic Algorithm for Finding the k Shortest Paths in a Network, in Egyptian Informatics Journal, 2010, DOI: 10.1016/j.eij.2010.10.004.



UDC 004.72

## RESEARCH AND ANALYSIS OF MULTIPATH ROUTING AND DATA FLOWS LOAD BALANCING PROCESSES IN SOFTWARE DEFINED NETWORKS BASED ON GENETIC ALGORITHM

**D. A. Perepelkin**, Dr. Sc. (Tech.), associate professor, Dean of FCE, RSREU, Ryazan, Russia; orcid.org/0000-0003-4775-5745, e-mail: perepelkin.d.a@rsreu.ru

**V. T. Nguyen**, master student, RSREU, Russia; orcid.org/0000-0003-2930-5775, e-mail: nguyenvantinsreu@gmail.com

Currently to ensure the quality of network services and applications, a new technology of computer networks has gained wide popularity – software defined networks (SDN). SDN provide flexibility in managing data flows through a centralized view of the entire network and the ability to program network services. To ensure the quality of service of multimedia content, voice, video traffic and other highly loaded applications, the most effective mechanism is the use of methods, principles and algorithms of multipath routing and load balancing in the network. **The aim of the work** is research and analysis of the processes of multipath routing and data flows load balancing in software defined networks based on a genetic algorithm. The conducted studies show that the use of a genetic algorithm in the SDN makes it possible to increase the efficiency of multipath routing and data flows load balancing processes, as well as to reduce jitter and transmission delay in the network.

**Key words:** software defined networks, multipath routing, load balancing, OpenFlow protocol, genetic algorithm.

**DOI:** 10.21667/1995-4565-2022-79-31-48

### References

1. **McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., Turner J.** Openflow: Enabling Innovation in Campus Networks. *Proc. ACM SIGCOMM Computer Communication Review*. 2008, vol. 38, no. 2. pp. 69-74.
2. **Hilmi E. E.**, Adaptive Video Streaming over OpenFlow Networks with Quality of Service. *Thesis for Degree of Master Science in Electrical and Electronics Engineering*, Koc University, July 2012. 91 p.
3. **Smeljanskij R. L.** *Programmno-konfiguriruemye seti. Otkrytye sistemy*. SUBD. 2012, vol. 9, pp. 23-26 (in Russian).
4. **Korjachko V. P., Perepelkin D. A.** *Programmno-konfiguriruemye seti*. Uchebnik dlja vuzov. Moscow: Gorjachaja linija – Telekom, 2020. 288 pp. (in Russian).
5. **Lemeshko A. V., Vavenko T. V.** Uovershenstvovanie potokovoj modeli mnogoputevoj marshrutizacii na osnove balansirovki nagruzki. *Problemy telekommunikacij*. 2012, no. 1 (6), pp. 12-29 (in Russian).
6. **Korjachko V. P., Perepelkin D. A.** Razrabotka i issledovanie matematicheskoi modeli mnogoputevoj adaptivnoj marshrutizacii v setjah svjazi s balansirovkoj nagruzki. *Jelektrosvjaz'*. 2014, no. 12, pp. 27-31 (in Russian).
7. **Perepelkin D. A.** Matematicheskaja model' mnogoputevoj adaptivnoj marshrutizacii s balansirovkoj neodnorodnoj nagruzki v uslovijah dinamicheskikh podkljuchenij uzlov i linij svjazi v telekommunikacionnyh setjah. *Radiotekhnika*. 2015, no. 5, pp. 46-54 (in Russian).
8. **Koryachko V. P., Perepelkin D. A., Byshov V. S.** Improved Multipath Adaptive Routing Model in Computer Networks with Load Balancing. *Proceedings SIBCON 2016 – IEEE 2016 International Siberian Conference on Control and Communications*, 2016, pp. 1-4. DOI: 10.1109/SIBCON.2016.7491721.
9. **Koryachko V. P., Perepelkin D. A., Byshov V. S.** Development and Research of Improved Model of Multipath Adaptive Routing in Computer Networks with Load Balancing. *Automatic Control and Computer Sciences*. 2017, vol. 51, no. 1, pp. 63-73. DOI: 10.3103/S0146411617010047.
10. **Shalimov A. et al.** Advanced study of SDN/OpenFlow controllers. *Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia*. ACM, 2013.

11. **Kartashevskij V. G., Galich S. V., Semjonov E. S., Kir'janova N. I.** Ocenka masshtabiruемости zaderzhki PKS-kontrollera na parallel'noj vychislitel'noj sisteme. *Infokommunikacionnye tehnologii*. 2017, vol. 15, no. 2, pp. 163-170 (in Russian).
12. **Malahov S. V., Tarasov V. N., Kartashevskij I. V.** Teoreticheskoe i jeksperimental'noe issledovanie zaderzhki v programmno-konfiguriruemyh setjah. *Infokommunikacionnye tehnologii*. 2015, vol. 13, no. 4, pp. 409-413 (in Russian).
13. **Koryachko V. P., Perepelkin D. A., Ivanchikova M. A., Byshov V. S., Tsyganov I. Yu.** Analysis QoS Metrics in Software Defined Networks. *Proceedings MECO 2017 – IEEE 6<sup>th</sup> Mediterranean Conference on Embedded Computing (MECO-2017)*, 2017, pp. 374-378. DOI: 10.1109/MECO.2017.7977240.
14. **Lemeshko A. V., Vavenko T. V.** Razrabotka i issledovanie potokovoj modeli adaptivnoj marshrutizacii v programmno-konfiguriruemyh setjah s balansirovkoj nagruzki. *Doklady Tomskogo gosudarstvennogo universiteta sistem upravlenija i radioelektroniki*. 2013, no. 3 (29), pp. 100-108 (in Russian).
15. **Lemeshko O. V., and Yeremenko O. S.** Dynamics analysis of multipath QoS-routing tensor model with support of different flows classes. *Proc. of the IEEE International Smart Systems and Technologies (SST-2016)*, 2016, pp. 225-230.
16. **Perepelkin D. A.** Konceptual'nyj podhod dinamicheskogo formirovanija trafika programmno-konfiguriruemyh telekommunikacionnyh setej s balansirovkoj nagruzki. *Informacionnye tehnologii*. 2015, vol. 21, no. 8, pp. 602-610 (in Russian).
17. **Perepelkin D. A., Byshov V. S.** Balansirovka potokov dannyh v programmno-konfiguriruemyh setjah s obespecheniem kachestva obsluzhivaniya setevyh servisov. *Radiotekhnika*. 2016, no. 11, pp. 111-119 (in Russian).
18. **Korjachko V. P., Perepelkin D. A., Ivanchikova M. A., Byshov V. S., Cyganov I. Ju.** Programmnaja infrastruktura i vizual'naja sreda raspredelennoj obrabotki potokov dannyh v programmno-konfiguriruemyh setjah. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2018, no. 65, pp. 44-54. DOI: 10.21667/1995-4565-2018-65-3-44-54 (in Russian).
19. **Nikul'chev E. V., Pajain S. V., Pluzhnik E. V.** Dinamicheskoe upravlenie trafikom programmno-konfiguriruemyh setej v oblachnoj infrastrukture. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2013, no. 3 (45), pp. 54-57 (in Russian).
20. **Leohin Ju. L., Fathulin T. D.** Ocenka vozmozhnosti predostavlenija garantirovannoj skorosti peredachi dannyh v programmno-konfiguriruemoj opticheskoj seti. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2020, no. 71, pp. 45-59. DOI: 10.21667/1995-4565-2020-71-45-59 (in Russian).
21. **Ushakova M. V., Ushakov Ju. A.** Issledovanie seti virtual'noj infrastruktury centra obrabotki dannyh s gibridnoj programmno-konfiguriruemoj kommutaciej. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2021, no. 75, pp. 34-43. DOI: 10.21667/1995-4565-2021-75-34-43 (in Russian).
22. **Koryachko V., Perepelkin D., Ivanchikova M.** Software System of Multipath Adaptive Routing in Networks of Distributed Data Centers, *Proceedings MECO 2019 – 8<sup>th</sup> Mediterranean Conference on Embedded Computing (MECO)*, 2019, DOI: 10.1109/MECO.2019.8759999.
23. **E. Jo, Deng Pan, J. Liu, and L. Butler.** A Simulation and Emulation Study of SDN-Based Multipath routing for Fat-Tree Data Center Networks, in *Proceeding Winter Simulation Conference*, 2014.
24. **Ron Banner and Ariel Orda.** Multipath routing algorithms for congestion minimization, in *IEEE/ACM Transactions on Networking*, 2007, pp. 413-424.
25. **K. Rajasekaran, Kannan Balasubramanian.** Energy Conscious based Multipath Routing Algorithm in WSN, in *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 8, no.1, pp. 27-34, 2016. DOI: 10.5815/ijcnis.2016.01.04.
26. **Meenakshi Moza, Suresh Kumar.** Analyzing Multiple Routing Configuration, in *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 8, no. 5, pp. 48-54, 2016. DOI: 10.5815/ijcnis.2016.05.07.
27. **Yilan Liu, Yun Pan, Muxi Yang, Wenqing Wang, Chi Fang, Ruijuan Jiang.** The Multi-Path Routing Problem in the Software Defined Network, in *11<sup>th</sup> International Conference on Natural Computation (ICNC)*, 2015, DOI: 10.1109/ICNC.2015.7377999.
28. **Ahmed Younes Hamed.** A Genetic Algorithm for Finding the k Shortest Paths in a Network, in *Egyptian Informatics Journal*, 2010, DOI: 10.1016/j.eij.2010.10.004.