# Титульный лист

# Задание на ВКР

Тема: Автоматизация управления сетевой доступностью в контейнеризированной сети организации

# Аннотация

В работе предлагается решение по разработке и внедрению адаптивного алгоритма для обеспечения увеличения скорости конвергенции протокола граничного шлюза (BGP). Апробация алгоритма производится в открытой платформе для автоматизации развертывания и масштабирования контейнеризированных приложений с использованием модуля управления сетевой политикой.

**Оглавление**

[Титульный лист 1](#_Toc197940106)

[Задание на ВКР 2](#_Toc197940107)

[Аннотация 3](#_Toc197940108)

[Список используемых сокращений 5](#_Toc197940109)

[Введение 6](#_Toc197940110)

[Литературный обзор 8](#_Toc197940111)

[1.1 Исследование различных протоколов динамической маршрутизации 8](#_Toc197940112)

[1.2 Исследование протокола граничного шлюза BGP 9](#_Toc197940113)

[1.3 Исследование Kubernetes и средств обеспечения сетевой доступности в нём 13](#_Toc197940114)

[1.4 Анализ моделей по предсказанию трафика 15](#_Toc197940115)

[1.5 Анализ статей по схожей тематике 15](#_Toc197940116)

[1.6 Вывод 15](#_Toc197940117)

[2. Проектирование 16](#_Toc197940118)

[3. Практическая реализация стенда и внедрение алгоритма 19](#_Toc197940119)

[3.1 Алгоритм решения поставленных задач 19](#_Toc197940120)

[Список использованных источников 21](#_Toc197940121)

[Список использованных источников 22](#_Toc197940122)

# Список используемых сокращений

AS – Autonomous systems/автономные системы

AWS – Amazon Web Servies/Сетевые сервисы Amazon

IGP – Interior gateway protocol/ Протокол внутреннего шлюза

MED - Multi-Exit Discriminator

TCP - Transmission Control Protocol/Протокол управления передачей

RAM – Random access memory/Оперативная память

ЛВС – Локальная вычислительная сеть

ЦОД – Центр обработки данных

BGP – Borderline gateway protocol/Протокол граничного шлюза

EBGP – External borderline gate protocol/Внешний протокол граничного шлюза

IBGP – Internal borderline gate protocol/Внутренний протокол граничного шлюза

OSPF – Open Shortest Path First/Протокол первого кратчайшего пути

BW- Bandwidth/Пропускная способность

CNI – Container networking interface/Сетевой интерфейс контейнера

K8s – Kubernetes

# Введение

В современном мире всё большую роль играют сервисы, их количество постоянно растет, а в связи с использованием микросервисной архитектуры влияние на увеличение масштаба сетей колоссально. Увеличение количества микросервисов, использующихся в центрах обработки данных (ЦОД), обусловливает существенное повышение необходимости к быстрому и качественному решению задач сетевого управления в условиях постоянно растущей нагрузки. Данный рост приводит к необходимости постоянного масштабирования данных сетей, заключающимся в увеличении количества серверов и маршрутизаторов. Существует потребность в простых и эффективных масштабируемых решениях для облегчения выполнения задач автоматизации, маршрутизации и управления постоянно растущими сетями.

Для реализации сервисов, которыми пользуются миллионы людей, используются такие технологии, как Docker и Kubernetes, в локальных или территориально распределённых вычислительных сетях с использованием технологии BGP для обеспечения маршрутизации между кластерами и контейнерами. Однако, BGP с стандартным набором параметров зачастую не обеспечивает наибольшую сходимость и надежность сети.

Цель данной работы – разработать и внедрить метод автоматизации сетевой доступности в контейнеризированную сеть организации.

Цель проекта определяет список задач работы:

Планирование и описание сетевой инфраструктуры организации (стенда).

Рассмотрение и анализ факторов, влияющих на эффективность BGP в описанной ЛВС.

Реализация стенда (развертывание k8s, с плагином Calico)

Разработка правил маршрутизации.

Разработка адаптивного алгоритма.

Использование программных средств для анализа трафика, позволяющих сделать качественную оценку внедренного решения. (снятие статистической картины с использованием инструментов, не влияющих на утилизацию канала связи).

Имплементация предложенного решения. (интеграция адаптивного алгоритма в сетевой стек).

Сравнение трафика предложенного решения с эталонным.

Объектом исследования является протокол граничного шлюза и принцип работы модулей для обеспечения взаимодействия между контейнерами и кластерами, а в частности IBGP в реализации CNI Calico.

Предметом исследования является методика и разработка адаптивного алгоритма маршрутизации для обеспечения наибольшей сходимости протокола BGP в контейнеризированной сети с использованием k8s и CNI Calico.

Значимость работы

Разработка алгоритма направлена на обеспечение более качественной сетевой доступности в кластеризированных сетях, подобный подход позволит более эффективно использовать доступную маршрутную информацию, а в следствии, повысит качество предоставляемых сервисов.

Использование алгоритма, взаимодействующее с доступными инструментами оркестровки, а не модифицируя данные инструменты, предоставляет выбор конечному пользователю между двумя, не влияя на базовый функционал оригинального программного обеспечения.

Вывод

Таким образом, разработка алгоритма по обеспечению автоматизации сетевой доступности является актуальной задачей, направленной увеличение при

# Литературный обзор

# Исследование различных протоколов динамической маршрутизации



Рисунок 1 - Протоколы маршрутизации

Протоколы динамической маршрутизации семейства IGP использующее алгоритмы поиска кратчайшего пути, в основе работы которых лежат алгоритмы Дейкстры и Беллмана-Форда не применяются в больших масштабируемых сетях из-за:

-Большого количества потребляемых ресурсов узла, в случае алгоритма Дейкстры.

-Невозможности использования циклов отрицательной метрики в сетях, где более 15 переходов, в случае алгоритма Беллмана Форда [1].

Таким образом, одним из наиболее популярных не проприетарных протоколов динамической маршрутизации, применяемых в масштабируемых сетях, является протокол граничного шлюза BGP.

# Исследование протокола граничного шлюза BGP

BGP – border gateway protocol (протокол граничного шлюза) – протокол динамической маршрутизации автономных систем (autonomous systems – AS). Основной функцией протокола является обмен информации о доступности сетей с другими узлами, использующими BGP. Эта информация доступности позволяет составить список автономных систем, через которые она проходит [2].

Под «**автономной системой**» (АС) традиционно понимается условная «зона ответственности» оператора связи с принадлежащими ему маршрутизаторами, находящимися под единым административным управлением и использующими единый согласованный план внутренней маршрутизации, а также согласованную картину адресатов, доступных через данную АС. Имеет большое количество параметров, обеспечивающих быструю и эффективную настройку сетевой политики.

**1.2.1 Применение в различных областях**

Востребованность обусловлена способностью эффективно масштабироваться и адаптироваться к динамичным средам, что критически важно для выполнения задач высокой доступности, балансировки нагрузки и управления трафиком [3].

Протокол используется в маршрутизации сети интернет, а также распространен в ЦОД и сетях с использованием средств кластеризации [4] [5].

BGP крайне востребован в обеспечении взаимодействия территориально распределенных приложений, развёрнутых на базе контейнерных сред под управлением инструментов оркестровки.

**1.2.2 Принцип работы**

Процесс определения наилучшего маршрута в BGP осуществляется последовательно через итеративное сравнение доступных путей по строго заданным критериям [2]. Алгоритм работает по принципу приоритетного выбора: первый путь, удовлетворяющий условиям, объявляется оптимальным, а последующие варианты игнорируются. Основные этапы сравнения, описанные в стандарте RFC и следующие в таком порядке:

**1.Проверка достижимости следующего прыжка.**  
Если следующий прыжок недостижим, маршрут игнорируется.

**2.Наибольший локальный приоритет**  
Маршрут с наибольшим значением Local Preference.

**3.Наименьший AS\_PATH**  
Предпочтение маршрута с наименьшей длиной AS\_PATH.

**4.Lowest Origin Type**  
Приоритет типов origin в порядке: IGP (0)> EGP (1)> Incomplete (2).

**5.Наименьший MED(Multi-exit-descriminator)**  
Маршрут с наименьшим значением MED (только если оба маршрута пришли от одного соседнего AS).

**6.Приоритет EGBP над IBGP**  
EBGP-маршруты выбираются перед IBGP-маршрутами.

**7.Наименьшая метрика IGP до следующего прыжка**  
Маршрут с наименьшей метрикой IGP (например, OSPF, EIGRP) до Next Hop.

**8.Самый старый маршрут (только для EBGP)**  
Если маршруты пришли через EBGP, выбирается самый «старый» (устойчивый) маршрут.

**9.Наименьший идентификатор соседа**  
Маршрут от BGP-соседа с наименьшим Router ID.

**10.Shortest Cluster List (для отраженных маршрутов)**  
Если используется Route Reflection (отражение маршрутов), предпочитается маршрут с наименьшей длиной Cluster List.

**11.Наименьший адрес соседа**  
Если все предыдущие шаги равны, выбирается маршрут от соседа с меньшим IP-адресом.

**1.2.3 Недостатки протокола**

Несмотря на широкое применение, BGP обладает рядом фундаментальных ограничений, которые осложняют его использование в динамичных и высоконагруженных средах:

1. **Неустойчивость маршрутов (Route Flapping)**  
   BGP подвержен частым изменениям маршрутов из-за нестабильности каналов связи или ошибок конфигурации. Это явление, называемое **«флаппингом»**, приводит к постоянному перерасчету таблиц маршрутизации и генерации избыточного трафика обновлений. Например, кратковременный сбой канала между автономными системами может вызвать каскадное распространение UPDATE-сообщений по всей сети, стандарт RFC2439 стремится решить эту проблему [6].
2. **Медленная конвергенция**  
   BGP характеризуется **длительным временем конвергенции**, обусловленное:
   * Использованием TCP для надежной доставки сообщений.
   * Последовательным применением 11-шагового алгоритма выбора лучшего пути, а в случае некоторых вендоров, 12 или 13. [2] [7]
   * Отсутствием механизмов мгновенного оповещения всех участников AS об изменениях топологии (в отличие от IGP, например, OSPF).
3. **Зависимость от префиксной гранулярности**  
   BGP оперирует префиксами IP-сетей, а не отдельными хостами или каналами. В результате сбой одного физического интерфейса может вызвать массовое обновление тысяч префиксов. Например, отказ магистрального канала между ЦОД Amazon в 2021 г. привел к перерасчету 150 тыс. маршрутов, вызвав частичную недоступность AWS [8].

**Следует отметить еще одно ограничение:**

* + **Сложность конфигурации**: Требуется ручная настройка фильтров, атрибутов и политик, что повышает риск ошибок (около 34% инцидентов связаны с человеческим фактором) [9].

**1.2.4 Перспективы развития**

Перечисленные недостатки и большая востребованность мотивируют создавать решения, стремящиеся уменьшить участие человека в создании конфигурационных параметров, а также повысить надежность при использовании протокола, например – нейронечёткие сети с применением алгоритмов машинного обучения [ссылка на Солодовника, и тд(посмотреть папку)].

1.3 Микросервисы

# Исследование Kubernetes и средств обеспечения сетевой доступности в нём

В связи с особенностями микросервисной архитектуры, где каждый узел в сети выполняет одну или несколько схожих задач, возникает необходимость в применении средств контейнеризации и оркестровки этих контейнеров. Kubernetes — платформа с открытым исходным кодом, созданная для автоматизации управления контейнеризированными приложениями. Первоначально разработанная компанией Google, поддерживается фондом **Cloud Native Computing Foundation (CNCF)** и является отраслевым стандартом для оркестрации контейнеров. Ее ключевая задача — упрощение развертывания, масштабирования и поддержки распределенных приложений в кластерных средах [10].

**1.3.1 Выбор модуля сетевой доступности**

Существует множество решений для обеспечения сетевой доступности в Kubernetes. В статье “ СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ СЕТЕВЫХ ПЛАГИНОВ ОРКЕСТРАТОРА KUBERNETES” [11] автором произведен статистический анализ влияния наиболее популярных CNI (Calico, Flannel, Cilium, Canal) на производительность системы, тестирование проходило по ряду протоколов TCP, UDP, HTTP, FTP и SCP в 10 Гбитной сети. За параметры, между которыми производилось сравнение, взяты:

1. Производительность в Мбит/с при использовании вышеуказанных протоколов
2. Потребление RAM
3. Потребление CPU

Как эталон взято базовое оборудование, без использования специализированных программных решений.

По результатам анализа наибольшую производительность показали Flannel и Calico. Автор предполагает, что совместимость Flannel связана с его компактностью и совместимостью с большим количеством архитектур, а также возможность автоматического определения MTU.

В случае Calico высокая производительность связана с тем, что модуль пропускает только установленный администратором трафик. Неоспоримым преимуществом Calico является безопасность в связи с вышеуказанной особенностью.

Flannel не подходит для целей данной работы, так как он использует оверлей сети (VXLAN).

Calico, который использует BGP для обеспечения между подами с возможностью применения IP/IP для построения туннелей между узлами территориально распределенных кластеров, выбран в роли CNI.

**1.3.2 Модуль(плагин) сетевого интерфейса Calico**

**Calico** — сетевое решение для Kubernetes, обеспечивающее безопасное взаимодействие контейнерных приложений и управление политиками доступа. В отличие от оверлейных сетей, Calico использует **подход уровня 3**, основанный на IP-маршрутизации, что минимизирует сложность инфраструктуры и снижает производительные накладные расходы. Использует BGP для распределения маршрутов.

# Анализ моделей по предсказанию трафика

Обзор статей по тематике разработке алгоритмов маршрутизации и тестирования пропускной способности.

# Анализ статей по схожей тематике

Минакова, Н. Н. Решение для обнаружения и блокировки распространения аномальной маршрутной информации протокола BGP-4 / Н. Н. Минакова, А. В. Мансуров // Высокопроизводительные вычислительные системы и технологии. – 2022. – Т. 6, № 1. – С. 26-31. – EDN ZQRPUG.

https://elibrary.ru/download/elibrary\_49373653\_53770577.pdf

Абросимов, Л. И. Разработка и моделирование гибридного протокола динамической маршрутизации / Л. И. Абросимов, М. А. Орлова, Х. Хаю // Программные продукты и системы. – 2023. – № 1. – С. 071-082. – DOI 10.15827/0236-235X.141.071-082. – EDN QGOXNM.

<https://elibrary.ru/download/elibrary_54261860_45203988.pdf>

# Вывод

Предлагается разработка алгоритма для выбора более эффективного маршрута путем динамического варьирования конфигурации протокола BGP по результатам периодического статического среза пропускной способности узла. Таким образом, при обнаружении ухудшающегося тренда, трафик будет направляться через альтернативный маршрут.

# Проектирование

Экспериментальный стенд спроектирован с применением средств виртуализации, а именно Hyper-V, для исследования устойчивости территориально распределенного кластера **Kubernetes** в условиях деградации каналов связи. Архитектура (рис. 2) реализует сценарии отказоустойчивости, где динамическое перераспределение трафика между географически удаленными узлами обеспечивается за счет интеграции **BGP** (Border Gateway Protocol) и **CNI Calico**.



Рисунок 2 - Архитектура сети

**2.1 Цели и функциональные требования**

1. **Имитация распределенного кластера Kubernetes**:
   * Control-node — хост с компонентами управления кластером (kube-apiserver, etcd, kube-controller-manager, kube-scheduler).
   * Worker-node — узел для развертывания рабочих нагрузок (kubelet, kube-proxy), принимающий деплойменты из системы оркестрации.
   * Использование CNI Calico обеспечивает сетевую политику, распределение IP-адресов и интеграцию с BGP для объявления маршрутов между узлами.
2. **Моделирование сбоев каналов связи**:
   * Сегменты **10.10.10.0/30** (R1–R2) и **10.10.20.0/30** (R3–R4) эмулируют каналы двух независимых ISP с возможностью искусственного введения задержек.
   * Деградация канала связи эмулируется путем замедления виртуальной сети средствами гипервизора Hyper-V
3. **Динамическая адаптация маршрутов**:
   * При обнаружении деградации канала алгоритм автоматически корректирует BGP-конфигурацию на хостах Ubuntu, перенаправляя трафик через альтернативный маршрутизатор (например, с R1–R2 на R3–R4).
   * **OPNsense** на маршрутизаторах R1–R4 обеспечивает транзитную маршрутизацию между автономными системами **AS64513** (Control-node, R1, R3) и **AS64514** (Worker-node, R2, R4).

**Архитектурные компоненты**

1. **Сетевой уровень**:
   * **Транспортные каналы ISP**:
     + Сегмент **10.10.10.0/30** (AS64513 ↔ AS64514) — основной канал, подвергаемый нагрузочному тестированию.
     + Сегмент **10.10.20.0/30** — резервный канал, активируемый при деградации основного.
   * **Внутрикластерная связь**:
     + Подсети **192.168.10.0/24** (Control Plane) и **192.168.20.0/24** (Data Plane) изолированы для минимизации коллизий управляющего и пользовательского трафика.
2. **Уровень оркестрации**:
   * **Kubernetes Control Plane**: Развернут на Control-node, координирует состояние кластера, распределяет поды на Worker-node.
   * **Calico BGP Peering**: Каждый узел кластера (Control-node, Worker-node) выступает как BGP-пир, анонсируя свои подсети через маршрутизаторы R1–R4. Это позволяет динамически обновлять таблицы маршрутизации при изменении топологии.

Схема уровня оркестрации предоставлена на рисунке 3.



Рисунок 3 – Схема оркестрации

Ниже предоставлены таблицы IP адресов и интерфейсов на каждом узле (таблица 1), а так же таблица BGPpeer-ов(Таблица 2)

|  |  |  |  |
| --- | --- | --- | --- |
| Имя узла | Интерфейс | Ip адрес | Маска |
| Control-node | Eth0 |  | /24 |
|  | Control-IPpool |  | /24 |
| Worker-node | Eth0 |  | /24 |
|  | Worker-IPpool |  | /24 |
| R1 | Eth0 |  | /24 |
|  | Eth1 |  | /30 |
| R2 | Eth0 |  | /24 |
|  | Eth1 |  | /30 |
| R3 | Eth0 |  | /24 |
|  | Eth1 |  | /30 |
| R4 | Eth0 |  | /24 |
|  | Eth1 |  | /30 |

Таблица 1 – IP адреса узлов

|  |  |  |
| --- | --- | --- |
| Узел | BGPpeer | ASN |
| Control-node | R1 | 64513 |
|  | R3 | 64513 |
| Worker-node | R2 | 64514 |
|  | R4 | 64514 |
| R1 | Control-Node | 64513 |
|  | R2 | 64514 |
| R2 | Worker-node | 64514 |
|  | R1 | 64513 |
| R3 | Control-Node | 64513 |
|  | R4 | 64514 |
| R4 | Worker-Node | 64514 |
|  | R3 | 64513 |

Таблица 2 – BGPpeers

# 2.2 Проектирование алгоритма

# 2.2.2 Цели и задачи

Алгоритм разработан для решения проблемы динамической оптимизации BGP-маршрутизации в программно-конфигурируемых сетях (SDN) на основе прогнозирования трафика. Ключевые задачи:

1. **Непрерывный мониторинг** сетевой нагрузки с частотой 0.2 Гц (5-секундные интервалы)
2. **Прогнозирование трафика** на горизонте 5-60 секунд с точностью ±15% (MAPE)
3. **Автоматическая корректировка** B-пиринговых отношений при критических отклонениях
4. **Сохранение целостности** сетевой конфигурации при изменениях

# 2.2.3 Функциональные требования

Система должна обеспечивать:

* Детерминированную реакцию на аномалии трафика (Δt < 120 сек)
* Гарантированную работоспособность при потере до 40% входных данных
* Совместимость с стандартными протоколами управления (BGP, REST API)
* Логирование всех состояний для пост-анализа

# Практическая реализация стенда и внедрение алгоритма

# Алгоритм решения поставленных задач

В рамках разработки адаптивного алгоритма обеспечения сетевой доступностей основной задачей стали создание логики выбора нового маршрута и его применение, без прерывания уже существующих пользовательских сессий.

* 1. алгоритм, анализ результатов



Рисунок 4 – Схема процессов  
  
Рисунок логики алгоритма?

Картинки с примерами yamlов, ,ifstat.log и state.json?  
Описание?

1. **Расчет экономической эффективности разработанного решения**

Для расчета эффективности разработанного решения и его внедрения учтено следующее:

1. Заработная плата разработчика
2. Стоимость оборудования
3. Накладные расходы
4. Время на проектирование, разработку и внедрение решения
5. Обучение персонала
6. Стоимость простоя руб/час

За показатели экономической эффективности взяты:

1. Срок окупаемости
2. Возврат вклада
3. Чистая приведённая стоимость

# Список использованных источников

1. Барыбин, Д. А. Сравнение алгоритмов Дейкстры и Беллмана-Форда при решении задачи о поиске кратчайшего пути в протоколах маршрутизации / Д. А. Барыбин, Е. Ю. Кофман, М. С. Шульгин // Символ науки: международный научный журнал. – 2021. – № 6. – С. 27-31. – EDN UIGLRD.  
   URL: https://www.elibrary.ru/download/elibrary\_46179512\_49205758.pdf
2. Border Gateway Protocol 4 (BGP-4) - https://datatracker.ietf.org/doc/html/rfc4271
3. BGP in the Data Center. O'Reilly Media Publ., 2017
4. , et al. Running BGP in data centers at scale. Proc. XVIII USENIX Symposium on NSDI, 2021
5. Use of BGP for Routing in Large-Scale Data Centers - <https://datatracker.ietf.org/doc/html/rfc7938>
6. BGP Best Path Selection Algorithm(2023) - <https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13753-25.html>
7. BGP Route Flap Damping - <https://datatracker.ietf.org/doc/html/rfc2439>
8. AWS Outage Analysis: December 7 & 10, 2021 - https://www.thousandeyes.com/blog/aws-outage-analysis-dec-7-2021
9. Verizon DIBR (2022) - <https://www.verizon.com/business/en-gb/resources/2022-data-breach-investigations-report-dbir.pdf>
10. Официальный сайт Kubernetes - <https://kubernetes.io/docs/concepts/overview/>
11. Бойдадаев, М. Н. Сравнительный анализ производительности сетевых плагинов оркестратора Kubernetes / М. Н. Бойдадаев // Молодой ученый. – 2024. – № 44(543). – С. 4-12. – EDN FQMZNL. <https://moluch.ru/archive/543/118838/>

# Список использованных источников

|  |  |
| --- | --- |
| [1] | Д. А. Барыбин, *Сравнение алгоритмов Дейкстры и Беллмана-Форда при решении задачи о поиске кратчайшего пути в протоколах маршрутизации,* Д. А. Барыбин, Е. Ю. Кофман, М. С. Шульгин // Символ науки: международный научный журнал. – 2021. – № 6. – С. 27-31. – EDN UIGLRD.. |
| [2] | RFC, Border Gateway Protocol 4 (BGP-4). |
| [3] | D. D.G., BGP in the Data Center, O'Relly Media Publ., 2017. |
| [4] | Abhashkumar A., Subramanian K., Andreyev A., Running BGP in data centers at scale., Proc. XVIII USENIX Symposium on NSDI, 2021. |
| [5] | RFC, Use of BGP for Routing in Large-Scale Data Centers. |
| [6] | Cisco, BGP Best Path Selection Algorithm, 2023. |
| [7] | RFC, BGP Route Flap Damping. |
| [8] | AWS, «Сводка перебоя,» декабрь 7 & 10, 2021. |
| [9] | Verizon, «DIBR,» 2022. |
| [10] | Kubernetes, Официальная документация Kubernetes. |
| [11] | Б. М. Н., «Сравнительный анализ производительности сетевых плагинов оркестратора Kubernetes,» *Молодой ученый,* № 44(543), pp. 4-12, 2024. |