# 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](#_Toc193694669)

[Задание на ВКР 2](#_Toc193694670)

[Аннотация 3](#_Toc193694671)

[Список используемых сокращений 5](#_Toc193694672)

[Введение 6](#_Toc193694673)

[Литературный обзор 8](#_Toc193694674)

[1.1 Исследование протокола граничного шлюза BGP 8](#_Toc193694675)

[1.2 Кратко о Docker? 13](#_Toc193694676)

[1.3 Исследование Kubernetes и средств обеспечения сетевой доступности в нём 14](#_Toc193694677)

[1.4 Анализ статей по схожей тематике(разработка алгоритмов с целью улучшения bgp) 16](#_Toc193694678)

[НИР(по выводам анализа статей в прошлой главе) 18](#_Toc193694679)

[Окр(реализация алгоритма, составленного в нир, практическая часть) 19](#_Toc193694680)

[Список использованных источников 20](#_Toc193694681)

# 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).

Анализ трафика

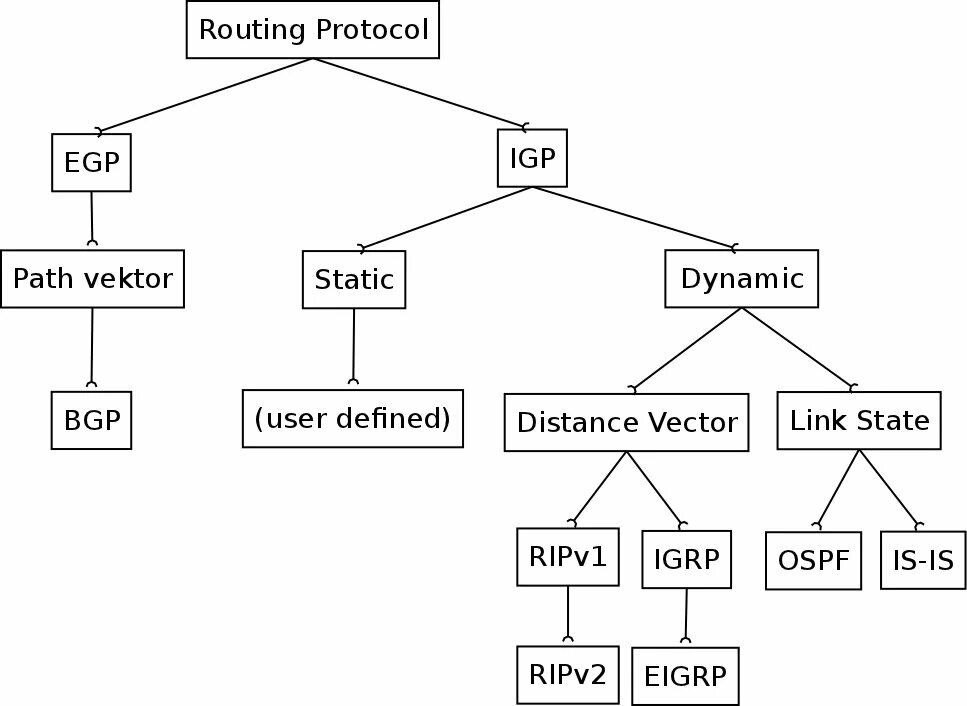
Приведение сравнительной характеристики с решением, и без

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

# Литературный обзор

# Исследование различных протоколов динамической маршрутизации

1. 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.

Среди протоколов динамической маршрутизации следует выделить

# 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]. Использование протокола также применяется 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, включают:

Проверка достижимости следующего прыжка.  
Если следующий прыжок недостижим, маршрут игнорируется (это предварительное условие, а не шаг сравнения).

Наибольший локальный приоритет  
Маршрут с наибольшим значением Local Preference (по умолчанию = 100).

Наименьший AS\_PATH  
Предпочтение маршрута с наименьшей длиной AS\_PATH (кроме случаев с AS\_SET или сегментами конфедерации).

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

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

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

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

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

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

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

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 [ссылка на RFC].
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. [ссылка на rfc, циску и джунипер]
   * 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].

**Дополнительные ограничения**

* + **Отсутствие встроенной безопасности**: Атаки типа **BGP Hijacking** (перенаправление трафика через злонамеренные AS) остаются распространенными из-за отсутствия аутентификации анонсов.
  + **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)].

Исходя из от протокола выбирать более желаемый по качественным параметрам маршрут предлагается рассмотрение динамического варьирования параметра Local\_Pref по результатам периодического статического среза пропускной способности узла.

# 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].

Существует множество решений для обеспечения сетевой доступности в Kubernetes, однако по результатом исследования, проведенного в статье “ СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ СЕТЕВЫХ ПЛАГИНОВ ОРКЕСТРАТОРА KUBERNETES” [11], в роли CNI был выбран Calico.

В работе 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), в то время как Calico, который использует BGP для обеспечения между подами с возможностью применения IP/IP для построения туннелей между узлами территориально распределенных кластеров.

1.3.1 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 и поддерживает несколько сетевых режимов.   
  
  
  
Добавить: Принцип работы, где применяется.

# Анализ статей по схожей тематике (

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

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>

# Практическая часть

# 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. Dutt D.G. BGP in the Data Center. O'Reilly Media Publ., 2017
4. Abhashkumar A., Subramanian K., Andreyev A., 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/>