# 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](#_Toc193687529)

[Задание на ВКР 2](#_Toc193687530)

[Аннотация 3](#_Toc193687531)

[Список используемых сокращений 5](#_Toc193687532)

[Введение 6](#_Toc193687533)

[Литературный обзор 8](#_Toc193687534)

[1.1 Исследование протокола граничного шлюза BGP 8](#_Toc193687535)

[1.2 Исследование Kubernetes и средств кластеризации 13](#_Toc193687536)

[Список использованных источников 14](#_Toc193687537)

# 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 данный момент крайне популярной архитектурой для сервисов архитектурой является микросервисный тип планировки. 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 масштабируемых протоколах маршрутизации для облегчения выполнения задач автоматизации 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

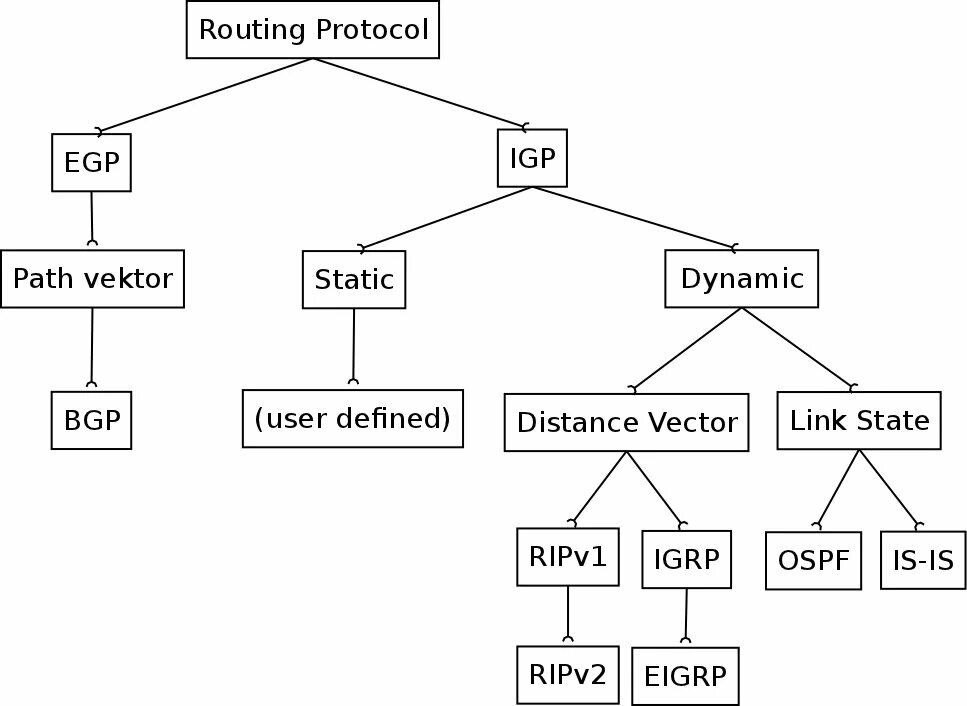
Приведение сравнительной характеристики с решением, и без

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 протокола граничного шлюза BGP

Рисунок 1 - Протоколы маршрутизации

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 [1]. Под «**автономной системой**» (АС) в таких случаях 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.

1.1.1 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 [2]. Протокол используется в маршрутизации сети интернет, а также распространен в ЦОД и сетях с использованием средств кластеризации [3-4].

1.1.2 Принцип работы

Процесс определения наилучшего маршрута в BGP осуществляется последовательно через итеративное сравнение доступных путей по строго заданным критериям [1]. Алгоритм работает по принципу приоритетного выбора: первый путь, удовлетворяющий условиям, объявляется оптимальным, а последующие варианты игнорируются. Основные этапы сравнения включают:

**1. Локальные приоритеты маршрутизатора**

* **WEIGHT**: Наибольший вес пути (внутренний параметр Cisco, не передается между устройствами).
* **LOCAL\_PREF**: Путь с максимальным значением (по умолчанию 100, настраивается командой bgp default local-preference).

**2. Происхождение маршрута**

* Локально сгенерированные пути (через команды network, aggregate или редистрибуцию из IGP) имеют приоритет над полученными от соседей.

**3. Атрибуты пути**

* **AS\_PATH**: Предпочтение кратчайшему списку пройденных автономных систем (AS).
* **MED (Multi-Exit Discriminator)**: Выбор пути с минимальным значением (используется для определения оптимального входа в AS).

**4. Тип BGP-сессии**

* eBGP-маршруты (между разными AS) приоритетнее iBGP-маршрутов (внутри AS).

**5. Метрики и стабильность**

* **Метрика IGP**: Путь с наименьшей стоимостью до NEXT\_HOP (например, OSPF cost).



Рисунок 2 – расчет стоимости маршрута в OSPF(BW – пропускная способность интерфейса, измеренная в бит/с)

* **Старшинство маршрута**: Стабильность важнее новизны — ранее выбранный eBGP-путь сохраняет приоритет.

**6. Идентификаторы устройств**

* **Router-ID**: Маршрут от устройства с минимальным идентификатором.
* **Cluster List**: При совпадении Router-ID выбирается путь с кратчайшим списком кластеров.
* **IP-адрес соседа**: В случае равенства всех параметров — предпочтение наименьшему адресу отправителя [5].

**7. Атрибут Origin**

Пути сортируются по приоритету источника:

* **IGP** (внутренний, например, через network) > **EGP** (устаревший протокол) > **INCOMPLETE** (импортирован через редистрибуцию).

Данный механизм обеспечивает детерминированность выбора, но требует точной настройки атрибутов для избежания субоптимальных решений [1].

1.1.3 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 [6].
2. **Медленная конвергенция**  
   BGP характеризуется **длительным временем конвергенции** (до нескольких минут в глобальных сетях), что обусловлено:
   * Использованием TCP для надежной доставки сообщений.
   * Последовательным применением 12-шагового алгоритма выбора лучшего пути. [5]
   * Отсутствием механизмов мгновенного оповещения о 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 [7].
4. Проблема усугубляется:
   * **Отсутствием агрегации префиксов** у многих операторов.
   * **Ростом размера глобальной таблицы маршрутизации**.
5. **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) [8].

Перечисленные недостатки мотивируют создавать решения, стремящиеся уменьшить участие человека в создании конфигурационных параметров, а также повысить безопасность при использовании протокола.

# Кратко о Docker?

# Исследование Kubernetes и средств обеспечения сетевой доступности в нём

Kubernetes — платформа с открытым исходным кодом, созданная для автоматизации управления контейнеризированными приложениями. Первоначально разработанная компанией Google, сегодня она поддерживается фондом **Cloud Native Computing Foundation (CNCF)** и стала отраслевым стандартом для оркестрации контейнеров. Ее ключевая задача — упрощение развертывания, масштабирования и поддержки распределенных приложений в кластерных средах [9].

Существует множество решений для обеспечения сетевой доступности в Kubernetes, но после рассмотрения статьи “ СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ СЕТЕВЫХ ПЛАГИНОВ ОРКЕСТРАТОРА KUBERNETES” [10], в роли CNI был выбран Calico.

В статье произведен статистический анализ влияния различных CNI на производительность системы, тестирование проходило по ряду протоколов TCP,UDP,HTTP,FTP и SCP в 10Гбитной 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 “голое железо”

По результатам анализа наибольшую производительность показали Flannel и Calico. Автор предполагает, что совместимость Flannel

1.2.1 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.

# Список использованных источников

1. Border Gateway Protocol 4 (BGP-4) - https://datatracker.ietf.org/doc/html/rfc4271
2. Dutt D.G. BGP in the Data Center. O'Reilly Media Publ., 2017
3. Abhashkumar A., Subramanian K., Andreyev A., et al. Running BGP in data centers at scale. Proc. XVIII USENIX Symposium on NSDI, 2021
4. Use of BGP for Routing in Large-Scale Data Centers - <https://datatracker.ietf.org/doc/html/rfc7938>
5. BGP Best Path Selection Algorithm(2023) - <https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13753-25.html>
6. BGP Route Flap Damping - <https://datatracker.ietf.org/doc/html/rfc2439>
7. AWS Outage Analysis: December 7 & 10, 2021 - https://www.thousandeyes.com/blog/aws-outage-analysis-dec-7-2021
8. Verizon DIBR (2022) - <https://www.verizon.com/business/en-gb/resources/2022-data-breach-investigations-report-dbir.pdf>
9. Официальный сайт Kubernetes - <https://kubernetes.io/docs/concepts/overview/>
10. 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/>