Методы восстановления топологической связности сетевых инфраструктур

Чеканов К.Ю.[1] ORCID: 0009-0007-0688-7342 <checkanov.kyu@phystech.edu>,
Ларин Д.В.[2] ORCID: 0000-0001-8686-8916 <larin.dv@phystech.edu>
Ларин А.В.[2] ORCID: 0009-0000-4415-8836 <avlarin@ispras.ru>
Подлесных Д. А.[1] https://www.researchgate.net/profile/Dmitry-Podlesnykh-2 <podlesnykh.da@mipt.ru>
[1]Московский Физико-Технический Институт, 141700, Россия, Московская Область, г. Долгопрудный, Институтский пер. 9
[2]Московский Государственный Университет им. М.В.Ломоносова, 119991, Россия, г. Москва, Ленинские горы, дом 1

1 Аннотация

Сетевые инфраструктуры сегодня – это основа для любой ИТ-инфраструктуры компаний. Поэтому, обеспечив стабильность их работы, можно гарантировать стабильную работу бизнеса, общественно доступных сервисов и т.д. Одной из сложностей для достижения этого является большое количество различных инструментов мониторинга и управления, которыми пользуются сетевые инженеры. Слабая связанность этих инструментов друг с другом и отсутствие API не дают автоматизировать процессы управления сетями и мешают достижения цели полной автоматизации мониторинга и управления сетевыми инфраструктурами. В данной работе изучаются подходы, которые предлагались общественными организациями (IETF, ONF) и различными компаниями (Google, Microsoft), для решения проблемы автоматизации работы с сетями. В частности, изучена возможность восстановления топологической связности сети на основании различных протоколов маршрутизации (а именно BGP и OSPF). Существующие решения в этой области обладают рядом недостатков, поэтому целью работы стала разработка собственной системы сбора топологии сетевой инфраструктуры. В работе рассматриваются различные подходы к мониторингу сетей включая мониторинг связности сети. А также описан собственный подход к сбору маршрутной информации, на основе которого реализована система, позволяющая в режиме реального времени контролировать изменение топологии внешней и внутренней связности сетей, детектировать нарушения связности, изменения маршрутов и т.д.

2 Введение

Ни один бизнес, государственное ведомство или сервис, доступный широкой аудитории, не может обойтись без использования сложных информационных систем, в основе которых всегда лежит сетевая инфраструктура. Для обеспечения бесперебойной работы ключевых прикладных систем и сервисов, а также обеспечения их доступности для конечных пользователей, всегда необходимо было понимать, как функционирует сетевая инфраструктура и иметь возможность отслеживать ее состояние в любой момент времени. Поэтому на

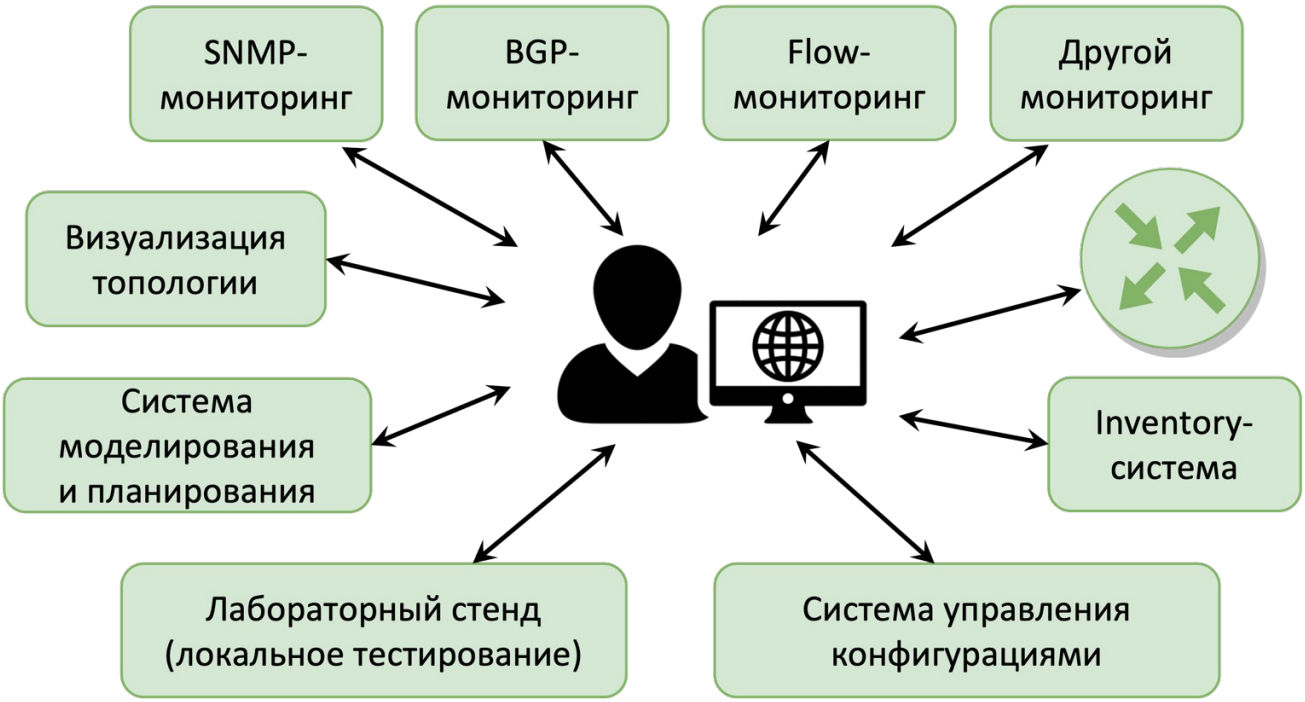

Рис. 1: Разнообразие инструментов мониторинга и управления сетями

протяжении всей истории развития сетей (и, в частности, Интернета) непрерывно развивалось направление по их

мониторингу, управлению и автоматизации. С развитием и усложнением сетей и появлением новых технологий, подходы к мониторингу, управлению и автоматизации тоже менялись. При этом изменялись и инструменты, которыми в своей работе пользовались сетевые инженеры.

Направления мониторинга, управления и автоматизации можно разделить на несколько больших групп:

1) Построение топологической связности сети (Network Topology Discovery)
2) Мониторинг производительности сети (Network Performance Monitoring)
3) Мониторинг маршрутизации и коммутации трафика
4) Анализ сетевого трафика (Network Traffic Analysis)
5) Безопасность. Предупреждение и защита от угроз (DDoS, Route Hijacking)
6) Моделирование и планирование (Network Service Modelling & Planning)
7) Управление конфигурациями сетевых устройств (Network Configuration Management)
8) Инвентаризация технических средств (Inventory или OSS (Operation Support System) системы)

В каждой из этих групп есть множество решений, которые тем или иным образом решают поставленные задачи. Многие из инструментов решают некоторую обособленную задачу – например, предназначены только для мониторинга производительности сети или анализа сетевого трафика. Однако сетевым инженерам необходимо решать сразу весь спектр описанных выше направлений. Что приводит к необходимости иметь в своей сети множество инструментов предназначенных для решения различных задач (см. рис. 1).

3 Построение топологической связности сети

В данной работе рассматривается восстановление связности сети на третьем (сетевом) уровне модели ISO/OSI. Топологическая связность в первую очередь дает базовое понимание структуры сети, которая находится под мониторингом и управлением. Она является «каркасом» или «скелетом» сети – сам по себе он несет мало информации, но зато является точкой, которая объединяет все остальные сервисы. Поэтому построение топологии – одна из основополагающих задач для описания сети.

В различных решениях существует множество способов построить топологическую связность сети. Все они строятся на основе автоматического обнаружения (AutoDiscovery). В данной работе намеренно не рассматриваются способы ручного сбора топологии сети, т.к. они не масштабируются.

По принципу работы системы AutoDiscovery можно разделить на два класса - активные и пассивные:

1) **Активное обнаружение топологии сети**

   Принцип: обнаружение топологии происходит периодическим опросом устройств.

   Опрос может проводиться при помощи:

   a) Утилиты Traceroute [9]
   b) Доступа к консоли (CLI – Command Line Interface) устройств и разбора таблиц маршрутизации/коммутации по протоколу SSH [10]
   c) Доступа к API (в качестве альтернативы API может выступать web-интерфейс, если он допускает автоматизацию сбора информации через curl, python requests и другие библиотеки).
   d) Доступа и разбора MIB (Management Information Base) на сетевых устройствах по протоколу SNMP (Simple Network Management Protocol) с целью получения таблиц маршрутизации/коммутации [11]

   Ограничения вытекают из особенностей данных и работы протоколов:

   1) Traceroute может находить не все узлы сети, если она находится за файрволлом, если в сети настроен MPLS (Multi-Protocol Label Switching), если в сети настроен балансировщик и т.п. (более детальное описание ограничений см. [9]). Строго говоря, узел не обязан отвечать на все ICMP-сообщения.
   2) Разбор таблиц маршрутизации/коммутации из CLI становится непростой задачей, когда необходимо работать в мульти-вендорной сети. В этом случае вывод CLI у всех устройств разный и необходимо поддерживать парсеры для различных ОС.
   3) Аналогичная b) проблема и с SNMP – MIB’ы могут быть вендоро-зависимыми.
   4) Формат работы API, CLI и MIB может изменяться между версиями программного обеспечения активного сетевого оборудования. Обновление версии может стать необходимым, в частности, для закрытия уязвимостей

   Такой способ может оказывать дополнительную нагрузку на сеть – т.к. необходимо на постоянной основе проводить опросы сетевых устройств. Кроме того, требуется принимать дополнительные меры информационной безопасности, в частности, ограничивать адреса, с которых может происходить запрос к командной строке устройства, чтобы предотвратить несанкционированный доступ к нему.

   Однако, зачастую информацию нельзя достать никаким другим способом, кроме как активным опросом устройств – например, информацию о настроенных на устройстве статических маршрутах.

Несмотря на приведенные выше недостатки, способы восстановления топологии сети при помощи опроса по SSH/SNMP широко используются самыми разными решениями: Zabbix [6], IPFabric [5], Ciena BluePlanet [2], Cisco WAE [3], Juniper NorthStar [4], NOC Project [12] и другими.

Отдельно отметим, что есть и другие способы получить таблицы маршрутизации/коммутации от устройств – по протоколам NETCONF/RESTCONF и gNMI (gRPC Network Management Interface). Последний можно частично отнести к категории пассивных, т.к. он позволяет подписываться на обновления данных на устройствах, в том числе и таблиц маршрутизации/коммутации [13]. Однако реального применения этих протоколов для восстановления топологий на практике и в научных трудах авторы данной работы не встречали.

2) ***Пассивное обнаружение топологии сети***

Принцип: обнаружение топологии происходит при помощи пассивного прослушивания.

Ограничения: этот способ подходит только для небольшого подмножества протоколов, которые хранят и передают информацию о топологии сети – так называемые Link-State протоколы. Среди них OSPF (Open Shortest Path First), ISIS (Intermediate System Intermediate System), BGP-LS (Border Gateway Protocol Link-State, его можно отнести к этому множеству лишь косвенно, т.к. он передает информацию, хранящуюся в таблицах маршрутизации IGP, Interior Gateway Protocol, протоколов) [1].

Метод пассивного прослушивания можно использовать совместно с такими протоколами, как NETCONF/RESTCONF, gNMI, но у них есть некоторые ограничения в основном связанные с широтой поддержки этих протоколов на оборудовании.

Данный способ позволяет обнаружить топологию без активного вмешательства в сеть. Соответственно никак не влияет на производительность сетевых устройств. Кроме того, он позволяет отслеживать работу протокола маршрутизации и связность сети. Подробнее об этом рассказывается в разделе 5.

Пассивное прослушивание для восстановления топологической связности используется в решениях: Ciena BluePlanet [2] и Topolograph (в частности OSPFWatcher) [8].

4 Связанные задачи

Как было сказано, топологическая связность - одна из основополагающих задач в построении описания сети. На ее основе можно решать целый комплекс задач в области мониторинга и моделирования сетей. Рассмотрим некоторые из таких задач.

4.1 Задача мониторинга маршрутизации и коммутации трафика

Мониторинг маршрутизации и коммутации трафика позволяет контролировать внутреннюю и внешнюю связность сети. В первую очередь речь идет об отслеживании изменений маршрутов трафика. Так как эти изменения зачастую могут (косвенно или напрямую) говорить о возникших в сети ошибках. Кроме того, зачастую для прохождения трафика необходимо выбирать оптимальные пути – например, проходящие через наиболее «широкие» по пропускной способности порты и соединения. Отслеживание всех маршрутов трафика в сети позволяет оценить, насколько сетевая инфраструктура соответствует требованиям бизнеса, обнаружить узкие места, которые требуют выбора более оптимальных маршрутов и т.д.

Мониторинг маршрутизации и коммутации осуществляется за счет отслеживания одноименных таблиц на сетевом оборудовании. В этих таблицах содержится актуальная информация о том, как сетевой трафик будет перенаправляться с одного устройства на другое до достижения целевого узла. Разберем мониторинг маршрутизации и коммутации более подробно:

1) ***Маршрутизация трафика***

Мониторинг маршрутизации производится путем отслеживания таблиц маршрутизации. Примеры наиболее распространенных протоколов маршрутизации: OSPF, ISIS, BGP, EIGRP, RIP.

Все протоколы основываются на различных принципах распространения маршрутной информации, поэтому единого способа, позволяющего одинаково эффективно мониторить каждый, учитывая их особенности, нет. Например, таблицы маршрутизации OSPF и ISIS (Link-State протоколов) можно отслеживать путем прослушивания служебных сообщений протоколов, для получения таблицы маршрутизации BGP (path-vector протокол) удобнее всего устанавливать соседство с маршрутизаторами по протоколу BGP, а для EIGRP и RIP информацию и путях прохождения трафика можно достать только из CLI или аналогичных интерфейсов (SNMP, Netconf, gNMI и т.д.). Более подробное описание способов получения таблиц маршрутизации для протоколов OSPF и BGP и о том, какую аналитику по полученным из них данным можно проводить, приведено в разделе 5.

Среди решений, которые позволяют отслеживать внутренние маршруты прохождения трафика можно выделить: IPFabric [5], Ciena BluePlanet [2], Cisco WAE [3], Juniper NorthStar [4], Topolograph [8] и другие. Внешние маршруты трафика позволяют отслеживать Ciena BluePlanet [2], Cisco WAE [3], Juniper NorthStar [4], pmacct [7] и другие, работающие с протоколом BGP.

2) ***Коммутация трафика***

Для отслеживания таблиц коммутации необходимо просматривать таблицы непосредственно на сетевом оборудовании. Получить таблицы можно при помощи CLI или аналогичных интерфейсов (SNMP, Netconf, gNMI и т.д.). Помимо классических таблиц коммутации (таблицы протоколов ARP, LLDP, CDP и других [11]), существуют и менее распространенные таблицы меток, которые используются в протоколе MPLS. В этом случаем идет речь о так называемых LSP (Label Switch Path), когда маршруты строятся на основе меток, а не при помощи MAC или IP адресов. MPLS чаще всего встречается в сетях операторов связи и используется для создания сервисов виртуальных частных сетей (Virtual Private Network, VPN) для корпоративных клиентов. Понимание коммутации трафика на основе меток позволяет отслеживать доступность сетей клиентов операторов связи, а также оптимизировать пути прохождения трафика клиентов.

Мониторинг классических таблиц коммутации доступно в решениях: IPFabric [5], Ciena BluePlanet [2], Cisco WAE [3]. Таблицы LSP позволяют отслеживать только решения, рассчитанные на сети операторов связи - Ciena BluePlanet [2], Cisco WAE [3], Juniper NorthStar [4].

4.2 Задача моделирования и планирования

Моделирование и планирование на основе данных мониторинга, позволяет решать задачи планирования развития сетевой инфраструктуры, обеспечивая поддержку принятия решений с учетом знания о функционировании сети.

Основой этого направления является матрица трафика, которая содержит в себе информацию о том, сколько трафика течет от каждого сетевого узла к каждому (формальное определение см. [15, 16]). Для построения данной матрицы трафика необходимо иметь точное знание о топологии сети и о потоках трафика, между узлами данной топологии. Матрица трафика позволяет решать задачи «What-If» аналитики (букв. аналитика «Что будет если»), Traffic Engineering (TE, процесс распределения нагрузки по сети), а также планирования [15]. Получение достоверной матрицы трафика сети довольно сложная задача – поскольку данные необходимо брать из разных источников (SNMP, Flow, счетчики LSP и т.п.), которые могут плохо коррелировать друг с другом [15]. В том числе, получение подобного рода телеметрии может нагружать сетевое оборудование, поэтому существуют различные методики, позволяющие оптимально получать данные телеметрии из сети специально для построения матрицы трафика, например [18].

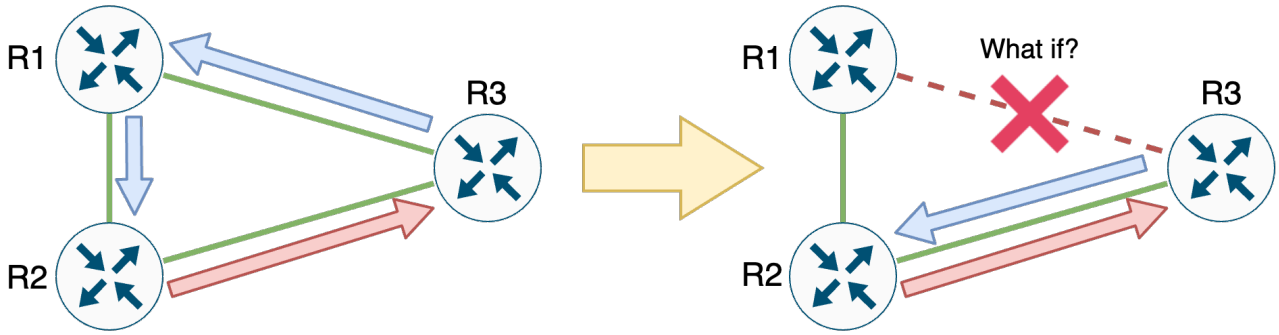

Рис. 2: Применение What-If анализа для оценки последствий сетевых сбоев. Слева: маршруты прохождения трафика до сбоя. Справа: изменение маршрутов прохождения трафика после сбоя на соединении R1-R3

Рассмотрим более подробно идею «What-If» аналитики, которая основывается на работе с матрицей трафика. Воспользуемся примером - предположим, что имеется 3 сетевых узла: R1, R2, R3 (см. рис. 2). Трафик от R2 до R3 проходит по пути R2-R3. А обратный маршрут трафика проходит через узел R1 – R3-R1-R2. Предположим, что сетевое соединение между R1-R3 пропало. В этом случае должно произойти изменение маршрута прохождения трафика от R3 к R2. Какие проблемы могут при этом возникнуть:

1) Может отсутствовать запасной (backup) маршрут между R3 и R2, что приведет к невозможности передавать данные между ними.
2) Пропускной способности соединения R2-R3 может не хватить для того, чтобы без потерь передавать трафик от R2 к R3 и обратно.
3) Требуемая пропускная способность соединения R2-R3 превысит заранее оговоренную и оплаченную, что потребует дополнительной оплаты (на этом основаны типовые договоры между провайдерами).

Для того, чтобы минимизировать риски получения описанных выше проблем, можно воспользоваться What-If анализом для превентивного обнаружения этих проблем еще до того, как они возникнут в сети. Фактически, для этого необходимо смоделировать сеть – ее топологию, загрузку соединений и узлов, маршруты прохождения трафика и т.п. И на основе модели получить ответ на вопрос: «Что будет, если соединение между R1 и R3 оборвется?» с возможными сценариями развития. Подобная модель сети позволяет упростить процесс планирования изменений сети и минимизирует риски возникновения ошибок.

Направление моделирования и планирования является сложным в реализации и востребовано в основном у компаний с большими сетевыми инфраструктурами (сотни тысяч устройств), т.к. в этих условиях ручное планирование невозможно или может приводить к ошибкам из-за невозможности учесть все факторы. Системы, которые позволяют решать эти задачи: Ciena BluePlanet [2], Cisco WAE [3], Juniper NorthStar [4].

4.3 Задача инвентаризации технических средств

Системы хранения информации о техническом оборудовании (Inventory или OSS системы) являются важной частью учета оборудования в сетевой инфраструктуре. Они могут выполнять следующие функции:

1) Хранение общей информации об устройстве. Например, имя устройства, его IP и MAC адреса, серийный номер, версия прошивки, физическое расположение, инвентарный номер, линия питания и т.п.
2) Хранение информации о подключении кабелей. Например, через какие порты соединены два устройства, каким кабелем и т.п.
3) Система управления IP-адресами (IP Address Management, IPAM).
4) Хранения информации о настроенных VLAN (Virtual Local Area Network).

Подобные системы используются повсеместно. В том числе как источник заведомо достоверной информации (Source of Truth) для внедрения автоматизации (например, из Inventory системы можно получить IP-адреса нужных сетевых устройств, а затем использовать эти данные для их автоматизированного обхода). Примеры систем инвентаризации технических средств: Netbox [19], Device42 [20], Solarwinds NCM [21] и множество других. Задача актуализации и выявления неточностей в подобных системах инвентаризации является актуальной проблемой многих компаний, занимающихся эксплуатацией сетей. Для ее решения возможно использование информации из системы пассивного восстановления топологии, в качестве альтернативного источника правды для системы инвентаризации устройств.

5 Реализация системы сбора топологической связности сети

В соответствии с приведенными выше обзором решений, проблематикой и описанием задач, в которых возможно использование системы по восстановлению топологии сети, авторами был сделан вывод о необходимости разработки собственного подобной решения. К этой системе был сформирован ряд следующих требований:

1) Инструменты сбора должны работать в режиме реального времени для отражения всех изменений сети.
2) Преимущественно (там, где это возможно) инструменты должны пассивно получать топологию и ничего не отправлять в сеть.
3) Инструменты должны обладать API для их простой интеграции со сторонними системами.

Как было описано в разделе 2.1.1, существует несколько способов восстановления топологической связности сети, которые делятся на два типа – активные и пассивные. Активные, в основном, предназначены для обнаружения топологий самых нижних уровней (физического L1, канального L2, в том числе для ряда протоколов сетевого уровня L3), а пассивные можно использовать для обнаружения топологии сетевого уровня L3 (и других – ограничения использования пассивного прослушивания связаны с интерфейсами общения с сетевым оборудованием, которые подробно разобраны в разделе 2.2.7).

Для восстановления топологии сетевого уровня L3 было принято решение начать с прослушивания Link-State протокола OSPF и протокола BGP. Однако, в будущем в системе планируется поддерживать и активные методы восстановления топологической связности.

5.1 Протокол OSPF

5.1.1 Описание протокола OSPF

OSPF (Open Shortest Path First) – это протокол динамической маршрутизации, широко используемый в крупных IP-сетях. Он является протоколом внутренней шлюзовой маршрутизации (Interior Gateway Protocol, IGP) и предназначен для эффективного распределения маршрутной информации в пределах автономной системы (Autonomous System, AS).

Приведем некоторые особенности протокола OSPF:

1) ***OSPF является протоколом на основе состояния связи (Link-State)***. Это означает, что на каждом устройстве в топологии OSPF поддерживается информация обо всей топологии – так называемая база

данных состояний связи (Link-State Database, LSDB). Для поддержания LSDB в протоколе OSPF используются специальные системные сообщения – Link-State Update (LSU), внутри каждого из которых содержится несколько Link-State Advertisement (LSA) сообщений. Последние несут в себе информацию обо всех связях в топологии – каждый маршрутизатор OSPF собирает информацию о своих связях и делится ей со своими соседями путем передачи сообщений LSA.

2) ***Алгоритм поиска кратчайшего пути***. Протокол OSPF использует алгоритм Дейкстры для расчета кратчайшего пути до подсетей назначения в пределах своей топологии. Для определения наиболее оптимального пути, алгоритм учитывает стоимость (метрику), связанную с каждым соединением.
3) ***Области (Areas)***. Топология OSPF может быть разделена на несколько областей для улучшения масштабируемости и снижения нагрузки на маршрутизаторы (из-за большого количества маршрутизаторов, LSDB может значительно вырастать в размерах. Из-за чего перерасчет путей может занимать продолжительное время). Маршрутизаторы, находящиеся в одной OSPF Area, обладают знанием о топологии всей Area. Маршрутизация между смежными областями происходит на основе сводной информации об анонсированных подсетях. Существует множество способов объединения разных Area друг с другом – подробнее о них см. RFC 2328 [66].
4) ***Hello сообщения***. Для установления и поддержания соседства маршрутизаторы OSPF используют Hello сообщения. Пакеты Hello с заданной периодичностью (Hello Interval, настраиваемый параметр) отправляются путем многоадресной рассылки (multicast), позволяя поддерживать связность топологии OSPF. Отметим, что кроме Hello Interval, который регулирует частоту отправки Hello сообщений, есть Dead Interval – если в течение заданного Dead Interval, маршрутизатором не будет получено Hello сообщение от его соседа, то такое соединение будет считаться разорванным, и маршрутизатор уведомит об этом всех своих соседей. Для корректной работы Hello и Dead интервалы должны быть синхронизированы между соседями (иметь одинаковые значения), иначе соседство не будет установлено [66].
5) ***Аутентификация***. Протокол OSPF поддерживает механизмы аутентификации для обеспечения целостности и безопасности обмена маршрутной информацией. Аутентификация может быть основана на аутентификации паролем или с помощью криптографических хэш-функций (например, MD5 или более криптостойких).
6) ***Быстрая сходимость***. OSPF был разработан для быстрой сходимости сети. В случае сбоев он позволяет быстро перерасчитать пути маршрутизации трафика и перенаправить трафик по новым оптимальным путям.

Таким образом, в соответствии с особенностями протокола OSPF, для восстановления топологии сети достаточно получить базу данных связности LSDB маршрутизаторов из всех областей (Area) топологии.

5.1.2 Инструмент сбора топологии и телеметрии OSPF

Рассмотрим различные способы получения LSDB маршрутизаторов OSPF:

1) **Интерфейс командной строки (CLI).** Данный способ заключается в использовании CLI для получения LSDB маршрутизатора. Таким образом, необходимо подключиться к одному маршрутизатору в каждой Area OSPF и выполнить команду в консоли устройства для получения LSDB.
   Плюсы:
   1. Сравнительно быстро можно получить топологию.
   2. Небольшая нагрузка на маршрутизатор.

   Минусы:
   1. Вендорно-зависимый вывод (CLI на оборудовании различных вендоров отличается – необходимо поддерживать оборудование разных вендоров).
   2. Нет возможности отслеживать сообщения OSPF напрямую, из-за чего невозможен мониторинг изменений топологии в режиме реального времени.
   3. Нужны данные для авторизации на устройстве.
2) **Simple Network Management Protocol (SNMP**). Данный способ заключается в использовании протокола SNMP для получения LSDB. Необходимо подключиться по протоколу SNMP к маршрутизаторам в каждой Area OSPF и запросить LSDB из SNMP MIB (Management Information Database).
   Плюсы:
   1. Нужен доступ на чтение (read-only) к маршрутизатору в сети.

   Минусы:
   1. Использование SNMP может оказывать нагрузку на роутер [22].
   2. Медленное получение топологии.

3. Нет возможности отслеживать сообщения OSPF напрямую, из-за чего невозможен мониторинг изменений топологии в режиме реального времени.

3) **NETCONF/RESTCONF**. Данный способ заключается в использовании протоколов NETCONF/RESTCONF для получения LSDB. Необходимо подключиться по протоколам NETCONF/RESTCONF к маршрутизаторам в каждой Area OSPF и запросить LSDB.

   Плюсы:
   1. Сравнительно быстро.
   2. Небольшая нагрузка на маршрутизатор.

   Минусы:
   1. Вендорно-зависимый вывод. NETCONF/RESTCONF работают с форматом YANG. YANG-описание LSDB у разных вендоров разные.
   2. Нет возможности отслеживать сообщения OSPF напрямую, из-за чего невозможен мониторинг изменений топологии в режиме реального времени.
   3. Нужны данные для авторизации на устройстве.

4) **gRPC Network Management Interface (gNMI)**. Данный способ заключается в использовании протокола gNMI для получения LSDB. Необходимо подключиться по протоколу gNMI к маршрутизаторам в каждой Area OSPF и запросить LSDB.

   Плюсы:
   1. Сравнительно быстро.
   2. Небольшая нагрузка на маршрутизатор.
   3. Возможность подписаться (push-уведомления) на изменения топологии.

   Минусы:
   1. gNMI в настоящий момент поддерживается на небольшом количестве оборудования, что не позволяет назвать его универсальным.
   2. Нужны данные для авторизации на устройстве.

5) **Установление соседства**. Данный способ заключается в установлении соседства (с помощью демона маршрутизации, например FRR [24] или BIRD [25]) с маршрутизаторами OSPF в нескольких Area. Это позволит получить на демоне маршрутизации актуальную LSDB.

   Плюсы:
   1. Сравнительно быстро.
   2. Небольшая нагрузка на маршрутизатор.
   3. Возможность отслеживать сообщения OSPF, мониторинг в режиме реального времени.

   Минусы:
   1. Сложно управлять, т.к. включение демона маршрутизации в сеть влияет на топологию OSPF (т.о. изменяется предмет мониторинга – топология сети). Необходимо дополнительно убеждаться в том, что демон маршрутизации ничего не отправит в сеть.

6) **Прослушивание трафика**. Данный способ заключается в пассивном прослушивании трафика с маршрутизаторов OSPF. Получая все системные сообщения OSPF (а именно Link State Update сообщения), можно восстановить топологию. Данный способ возможен благодаря тому, что один раз в заданный временной интервал (OSPF MaxAge, по умолчанию 60 минут) все маршрутизаторы OSPF отправляют свою таблицу LSDB соседям. Получение сообщений можно обеспечить за счет использования SPAN/TAP зеркалирования (mirroring) трафика с маршрутизаторов.

   Плюсы:
   1. Сравнительно быстро.
   2. Полностью пассивное прослушивание.
   3. Возможность отслеживать сообщения OSPF, мониторинг в режиме реального времени.

   Минусы:
   1. Необходима реализация стека протоколов для того, чтобы производить разбор (парсинг) сообщений OSPF.
   2. Сравнительно долгое восстановление топологии при первом включении (OSPF MaxAge в худшем случае – 60 минут).
   3. Зеркалирование трафика с высокоскоростных маршрутизаторов требует относительно дорогих интерфейсов с такой же пропускной способностью.

В данной работе был выбран способ получения топологии OSPF через пассивное прослушивание трафика, как наиболее простой в управлении и позволяющий отслеживать сообщения OSPF в режиме реального времени. Отметим, что долгое восстановление топологии при первом включении не является существенным минусом, т.к. ее необходимо собрать только один раз.

Прежде чем переходить к реализации инструмента сбора топологии OSPF обсудим еще несколько способов получения топологии OSPF, которые могут стать дальнейшим развитием системы:

1) **Установление пассивного соседства**. В отличие от способа установления соседства, описанного выше, «пассивное» соседство подразумевает использование упрощенной версии демона маршрутизации OSPF. Этот демон должен поддерживать соединение с соседом, путем отправки Hello сообщений. Кроме того, при получении LSU сообщения от соседа, демон должен отправлять LS Acknowledge сообщение с подтверждением того, что LSU сообщение было получено. Больше никаких сообщений отправлять не требуется. Таким образом, упрощенный демон никак не сможет повлиять на топологию OSPF.
   Используя такой демон маршрутизации можно получить все плюсы пассивного прослушивания, а также простую установку системы в сеть. На текущий момент времени похожий демон используется решением Ciena BluePlanet [2], а Open-Source альтернативы авторы данной работы не встречали.
2) **Interface To the Routing System (I2RS).** В 2016 году было выпущено RFC 7921 [23], которое стандартизировало интерфейс маршрутизаторов, позволяющий им отдавать LSDB (и в целом любую из таблиц маршрутизации) по специализированному протоколу. Стандарт также подразумевал возможность подписки на обновления таблиц. Однако, на практике он до сих пор не был реализован.

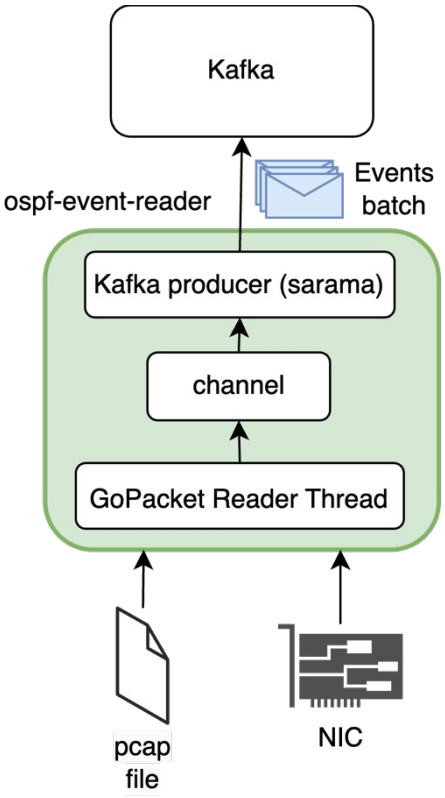

Рис. 3: Архитектура инструмента сбора топологии OSPF

Перейдем к реализации инструмента сбора топологии OSPF при помощи пассивного прослушивания трафика. На рис. 3 изображена его архитектура (ospf-event-reader). Инструмент сбора топологии передает все считанные и разобранные сообщения OSPF в брокер сообщений Kafka.
Сам процесс сбора топологии реализован следующим образом:

1) В отдельной горутине (goroutine, усовершенствованные легковесные потоки в языке Go, требующие небольшого размера стека – 2-4КБ. Не зависят от операционной системы и существуют в виртуальном пространстве выполнения Go). запускается процесс чтения сетевых пакетов OSPF с сетевого интерфейса докер-контейнера или сервера (GoPacket Reader Thread на рис. 8). Также поддерживается чтение пакетов из pcap-дампа трафика OSPF.
2) Полученные пакеты проходят разбор, который реализован при помощи доработанной версии библиотеки gopacket [26]. Из пакетов достается информация о OSPF LSA.
3) Далее, LSA объединяются в группу (батч), реализованную через простейший буфер фиксированного размера. По заполнении буфера или по таймеру, батч сообщений передается во внутренний канал (очередь), а буфер очищается.
4) Далее, батч сообщений из канала читается процессом Kafka Producer. Он сериализует батч сообщений и передает его в Kafka.

Данный инструмент достаточно прост в реализации и обладает рядом достоинств:

1) Легковесный. Может работать на любом оборудовании и не требует большого количества ресурсов.
2) Позволяет собирать топологию OSPF в режиме реального времени. При этом полностью в пассивном режиме.
3) Масштабируемый. За счет использования внешнего брокера сообщений Kafka можно использовать несколько копий инструмента, расположенных в разных участках сети.

5.2 Протокол BGP

5.2.1 Описание протокола BGP

BGP (Border Gateway Protocol) – это протокол динамической маршрутизации, который используется для обмена маршрутной информацией между автономными системами (Autonomous System, AS). Он относится к категории протоколов внешней шлюзовой маршрутизации (Exterior Gateway Protocol, EGP) и изначально создавался для обеспечения связности и оптимального выбора маршрутов в глобальной сети Интернет. Сейчас BGP

применяется для решения множества задач – в числе которых сигнализация маршрутов VPN, маршрутизация в Центрах Обработки Данных (ЦОД) [27] и т.д.

Приведем некоторые особенности протокола BGP:

1) ***Path-Vector Protocol***. BGP является протоколом вектора пути – он передает информацию о маршрутах в виде векторов, которые состоят из набора переходов между AS (т.н. AS-path). Каждый маршрут включает в себя информацию об автономных системах, через которые он проходит до конечного пункта назначения.
2) ***Надежность и масштабируемость***. Протокол BGP предназначен для обеспечения высокой надежности и позволяет масштабироваться на сети масштаба всего Интернета. Благодаря ему автономные системы способны устанавливать и поддерживать пиринговые отношения (peering) и обмениваться информацией о маршрутах трафика.
3) ***Policy-Based Routing (PBR).*** BGP позволяет управлять маршрутами трафика при помощи специальных политик, с помощью которых автономные системы могут самостоятельно определять предпочтительные маршруты на основе различных критериев: пропускная способность каналов, задержки и т.п.
4) ***eBGP и iBGP***. Протокол BGP поддерживает два типа пиринговых отношений - внешние (eBGP) и внутренние (iBGP). Соединение eBGP устанавливается между различными автономными системами и обеспечивает внешнюю связность или используется для создания конфедераций BGP – когда внутри одной публичной автономной системы есть несколько внутренних автономных систем. В свою очередь iBGP обеспечивает распространение маршрутной информации между маршрутизаторами внутри одной автономной системы.
5) ***Стратегия выбора маршрутов***. BGP поддерживает гибкость в выборе маршрутов путем использования различных атрибутов и политик. Это позволяет провайдерам интернет-услуг и корпоративным организациям настраивать маршрутизацию в соответствии с их требованиями и предпочтениями.
6) ***Расширения – Multi-Protocol BGP (MP-BGP).*** MP-BGP – это расширение протокола BGP, которое позволяет ему передавать не только информацию о маршрутах протокола IPv4, но и о других протоколах. Среди которых: IPv6, MPLS (Multi-Protocol Label Switching), маршруты VPN (IPv4/v6 VPN, EVPN, L2VPN VPLS и др.).

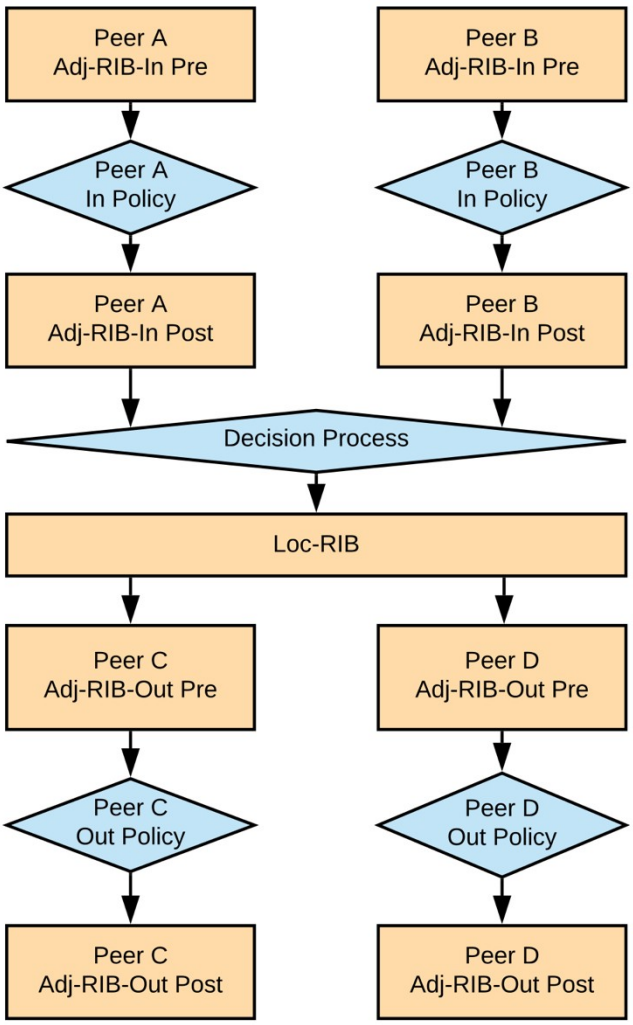

Рис. 4: BGP RIB и политики маршрутизации, использующиеся в процессе BGP [28]

Информация о маршрутах хранится в специализированных in-memory таблицах маршрутной информации (Routing Information Bases, RIBs) на сетевом устройстве. Сам процесс BGP имеет несколько RIB, как показано на рис. 4. RIB входящего соединения (Adjacency RIB Inbound, Adj-RIB-In) содержит информацию о маршрутах, которые были анонсированы соседями маршрутизатора. Adj-RIB-In может быть разделен на две части: до применения политик маршрутизации (pre-policy) и после (post-policy). Для процесса BGP можно задать набор входных политик (in-policy) для каждого соседа (пира, peer), которые позволяют изменять или отклонять входящие маршруты. Только те маршруты, которые проходят проверку in-policy, попадают в post-policy Adj-RIB-In. Еще один фильтр применяется в процессе принятия решения об установке маршрута BGP в локальную таблицу маршрутизации. Локальная RIB (Local RIB, Loc-RIB) содержит маршруты, выбранные этим процессом. RIB исходящего соединения Adjacency RIB Outbound (Adj-RIB-Out) существует для каждого пира и содержащиеся в ней маршруты обычно соответствуют тому, что находится в Loc-RIB. Аналогично Adj-RIB-In, Adj-RIB-Out делится на две части – pre-policy и post-policy, которые регулируются при помощи исходящих политик маршрутизации (out-policy).

5.2.2 Инструмент сбора топологии и телеметрии BGP

Отметим, что для протокола BGP под топологией в первую очередь подразумевается топология внешней связности – то есть соединение всех AS Интернета друг с другом. Данную топологию невозможно однозначно точно определить при наблюдении из одной автономной системы – некоторые соединения между AS не будут обнаружены. Для точного определения связности Интернета необходимо иметь несколько точек мониторинга, установленных в разных сетях по всему миру. Таким образом, например, поступают такие решения, как Kentik [14] и Cisco ThousandEyes [29].

Однако, для протокола BGP существует и внутренняя топология (при использовании протокола iBGP), основанная на TCP сессиях между маршрутизаторами BGP. Ее можно получить при помощи специализированного протокола BGP Monitoring Protocol (BMP). Он позволяет точно определять и отслеживать топологию внутренней связности, основываясь на состоянии BGP сессий [28].

Рассмотрим различные способы мониторинга RIB BGP:

1) **Интерфейс командной строки (CLI).** Данный способ заключается в подключении через CLI к маршрутизаторам BGP и выполнении команд в консоли для получения RIB.
   Плюсы:
   1. Сравнительно быстро можно получить RIB.
   2. Небольшая нагрузка на маршрутизатор.

   Минусы:
   1. Вендорно-зависимый вывод (CLI на оборудовании различных вендоров отличается – необходимо поддерживать оборудование разных вендоров).
   2. Нет возможности отслеживать сообщения BGP напрямую, из-за чего невозможен мониторинг изменений топологии в режиме реального времени.
   3. Нужны данные для авторизации на устройстве.
2) **Simple Network Management Protocol (SNMP**). Данный способ заключается в использовании протокола SNMP для получения RIB. Необходимо подключиться по протоколу SNMP к маршрутизаторам BGP и запросить RIB из SNMP MIB.
   Плюсы:
   1. Нужен доступ на чтение (read-only) к маршрутизатору в сети.

   Минусы:
   1. Использование SNMP может оказывать нагрузку на роутер [22].
   2. Медленное получение топологии.
   3. Нет возможности отслеживать сообщения BGP напрямую, из-за чего невозможен мониторинг изменений топологии в режиме реального времени.
3) **Установление соседства**. Данный способ заключается в установлении соседства (с помощью демона маршрутизации, например FRR [24], BIRD [25], goBGP [30]) с маршрутизаторами BGP. Это позволит получить на демоне маршрутизации актуальную RIB наблюдаемых маршрутизаторов и отслеживать состояние внешней связности сети.
   Плюсы:
   1. Сравнительно быстро.
   2. Небольшая нагрузка на маршрутизатор.
   3. Возможность отслеживать сообщения BGP, мониторинг в режиме реального времени.

   Минусы:

1. Необходима реализация стека протоколов для того, чтобы производить разбор (парсинг) сообщений BGP

4) **Прослушивание трафика**. Данный способ заключается в пассивном прослушивании трафика с маршрутизаторов BGP. Получая все системные сообщения BGP (а именно BGP Update сообщения), можно восстановить топологию. Получение сообщений можно обеспечить за счет использования SPAN/TAP зеркалирования (mirroring) трафика с маршрутизаторов.

   Плюсы:
   1. Сравнительно быстро.
   2. Полностью пассивное прослушивание.
   3. Возможность отслеживать сообщения BGP, мониторинг в режиме реального времени.

   Минусы:
   1. Необходима реализация стека протоколов для того, чтобы производить разбор (парсинг) сообщений BGP.
   2. Нет гарантий, что соберем всю RIB за время прослушивания – в отличие от OSPF, у маршрутов BGP нет времени жизни и они не переанонсируются.
   3. В сообщения BGP Withdraw не содержатся атрибуты Next-Hop и AS-Path, таким образом нет возможности понять, какой именно маршрут был удален из RIB.
   4. Невозможно отследить выставленные политики маршрутизации на прослушиваемых устройствах – при пассивном прослушивании будут получены все маршруты, а какие из них будут отброшены маршрутизатором после прохождения in-policy неизвестно.

5) **BGP Monitoring Protocol (BMP)**. BMP позволяет отслеживать состояние RIB и BGP сессий. Для этого необходимо настроить стриминг телеметрии BMP с маршрутизаторов BGP.

   Плюсы:
   1. Мониторинг RIB (Adj-RIB-In, Loc-RIB, Adj-RIB-Out) в режиме реального времени.
   2. Мониторинг сессий BGP.
   3. Маркирование путей.

   Минусы:
   1. Поддерживается только на оборудовании крупных вендоров [28].
   2. Множество RFC, регулирующих телеметрию BMP – на разных устройствах разный BMP [28].

В данной работе был выбран способ получение топологии и мониторинга BGP через установление соседства с маршрутизаторами, как наиболее простой в управлении и позволяющий отслеживать сообщения BGP в режиме реального времени.

Перейдем к описанию реализации инструмента. На рис. 5 изображена его архитектура (bgp-event-reader). Инструмент сбора топологии и мониторинга передает все считанные и разобранные сообщения BGP в брокер сообщений Kafka. На вход bgp-event-reader передается gRPC-stream событий BGP, который получен от BGP демона goBGP [30]. GoBGP подключается к маршрутизаторам в сети, устанавливает с ними соседство и передает все полученные сообщения (из таблицы Adj-RIB-In демона goBGP) в gRPC-stream.

Сам процесс сбора топологии и мониторинга реализован следующим образом:

1) В отдельной нити исполнения (горутине) запускается процесс чтения сетевых пакетов BGP из gRPC-stream goBGP. GoBGP используется в том числе и в качестве библиотеки для разбора сообщений BGP.
2) Далее, сообщения BGP (в сыром виде) объединяются в группу (батч), реализованную через буфер фиксированного размера. По заполнении буфера или по таймеру, батч сообщений передается во внутренний канал (очередь), а буфер очищается.
3) Далее, батч сообщений из канала читается процессом Kafka Producer. Он сериализует батч сообщений и передает его в Kafka.

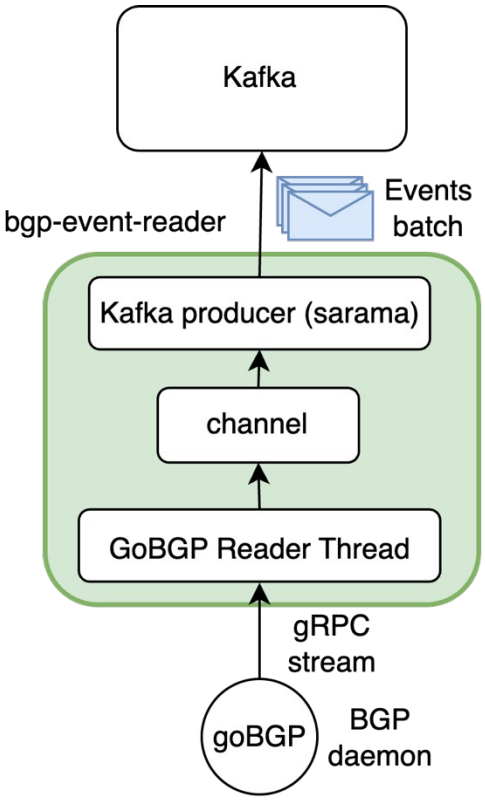

Рис. 5: Архитектура инструмента сбора топологии BGP

Данный инструмент, аналогично инструменту сбора топологии OSPF, прост в реализации, позволяет получать все сообщения BGP в режиме реального времени и легко масштабируется.

В рамках создания данного инструмента, в goBGP авторами данной работы был добавлен разбор дополнительных типов Address Family протокола MP-BGP (в частности, добавлена поддержка Address Family L2VPN VPLS). Эти изменения были одобрены мейнтейнерами goBGP и добавлены в основную ветку проекта.
В дальнейшем планируется добавление поддержки сбора телеметрии по протоколу BMP для мониторинга внутренней связности BGP.

6 Результаты тестирования

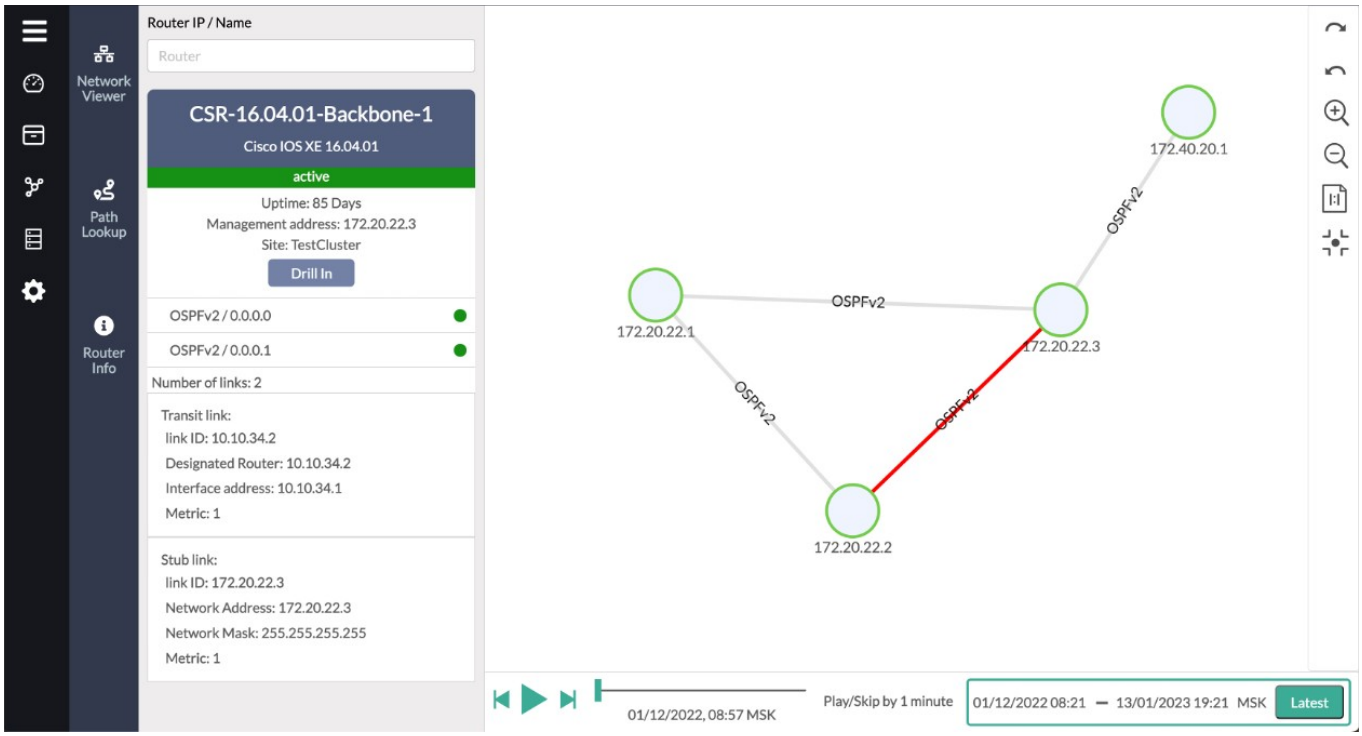

Рис. 6: Результат восстановления топологии OSPF и реакция на отключение интерфейса маршрутизатора router-3

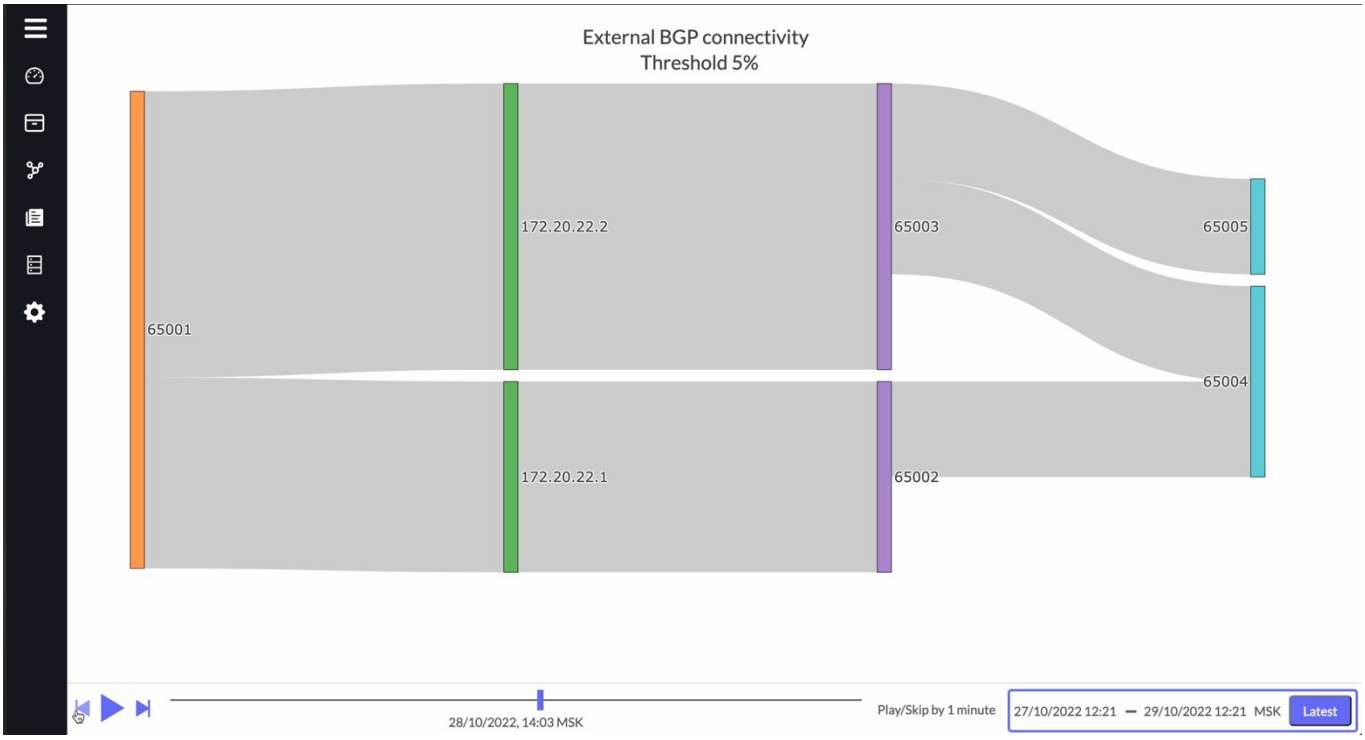

Рис. 7: Результат восстановления топологии BGP на диаграмме SanKey

В рамках тестирования разработанной системы по восстановлению топологии и мониторингу протоколов OSPF и BGP на лабораторном стенде отслеживались следующие параметры:

1) Корректное восстановление топологической связности сети. Проверка корректности осуществлялась визуально и путем чтения таблиц маршрутизации на роутерах.

2) Корректное отслеживание состояния топологии при нарушениях связности. Нарушения связности моделировались при помощи отключения интерфейсов на маршрутизаторах. Для тестирования OSPF использовалось отключение одного из внутренних интерфейсов на маршрутизаторе, а для тестирования изменения внешних маршрутов BGP использовалось отключение интерфейса, подключенного к соседу из внешней автономной системы.

Результаты тестирования восстановления топологии OSPF и реакция на отключение интерфейса изображены на рисунке 6. Красным цветом отмечено то соединение, которое в выбранный момент времени (задан при помощи слайдера внизу страницы) было неактивно. Восстановленная топология сети полностью совпала с лабораторным стендом.

Результаты тестирования восстановления топологии внешней связности для BGP изображены на рисунке 7. Для отображения использовалась диаграмма SanKey – на ней в процентном соотношении показано, какое количество префиксов выходит через граничные маршрутизаторы в сети.

7 Заключение

В рамках данной работы был проведен анализ существующих инструментов по мониторингу и управлению сетевой инфраструктурой. Особое внимание уделано проблеме восстановления топологической связности сети и ряда смежных задач, обосновывающих актуальность и важность основной темы работы. Также был детально рассмотрен процесс сбора информации из протоколов маршрутизации BGP и OSPF, как источника данных для построения топологии сети. Полученные результаты, при тестировании системы сбора маршрутной информации, показывают корректность описанных методов для решения поставленной задачи. В качестве дальнейшего вектора развития работы предполагается построение системы по восстановлению топологической связности сети на уровнях L1/L2, обнаружения точек повреждения и интеграции данной системы для решения задач управления трафиком.

8 Список использованной литературы

[1] Wong T., Jacobson V., Alaettinoglu C. Internet routing anomaly detection and visualization //2005 International Conference on Dependable Systems and Networks (DSN'05). – IEEE, 2005. – С. 172-181.

[2] Ciena BluePlanet https://www.blueplanet.com/

[3] Cisco WAE (WAN Automation Engine) https://www.cisco.com/c/en/us/products/routers/wan-automation-engine/index.html

[4] Juniper NorthStar Controller https://www.juniper.net/us/en/products/network-automation/northstar-controller.html

[5] IPFabric https://ipfabric.io/

[6] Zabbix https://www.zabbix.com/

[7] pmacct https://github.com/pmacct/pmacct

[8] Topolograph https://topolograph.com/

[9] Donnet B., Friedman T. Internet topology discovery: a survey //IEEE Communications Surveys & Tutorials. – 2007. – Т. 9. – №. 4. – С. 56-69.

[10] Srivatsa M. et al. Scalable topology discovery and link state detection using routing events //2008 Symposium on Reliable Distributed Systems. – IEEE, 2008. – С. 165-174.

[11] Andreev A., Bogoiavlenskii I. An algorithm for building an enterprise network topology using widespread data sources //2017 21st Conference of Open Innovations Association (FRUCT). – IEEE, 2017. – С. 34-43.

[12] NOC Project https://getnoc.com/

[13] Arista gNMI documentation https://aristanetworks.github.io/openmgmt

[14] Kentik https://www.kentik.com/

[15] Blili R., Maghbouleh A. Best Practices for Determining Traffic Matrices in IP Networks V 4.0.

[16] Medina A.. Traffic matrix estimation: Existing techniques and new directions //ACM SIGCOMM Computer Communication Review. – 2002. – Т. 32. – №. 4. – С. 161-174.

[17] Anikin A., Dvurechensky P., Gasnikov A., Golov A., Gornov A., Maximov Y., Mendel M., Spokoiny V.. Modern efficient numerical approaches to regularized regression problems in application to traffic demands matrix calculation from link loads //Proceedings of International conference ITAS-2015. Russia, Sochi. – 2015.

[18] Bouhtou M., Gaubert S., Sagnol G. Optimizing the deployment of Netflow to infer traffic matrices in large networks.

[19] Netbox https://netbox.dev/

[20] Device42 https://www.device42.com/

[21] Solarwinds Network Configuration Management (NCM) https://www.solarwinds.com/

[22] Kamiya K., Kurakami H. Visualizing Traffic on Network Topology//NTT, 2012 https://resources.sei.cmu.edu/asset_files/Presentation/2012_017_101_50785.pdf

[23] Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", RFC 7921, DOI 10.17487/RFC7921, June 2016, https://www.rfc-editor.org/info/rfc7921.
[24] FRRouting https://frrouting.org/
[25] BIRD https://bird.network.cz/
[26] gopacket https://github.com/google/gopacket
[27] Lapukhov, P., Premji, A., and J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", RFC 7938, DOI 10.17487/RFC7938, August 2016, https://www.rfc-editor.org/info/rfc7938.
[28] Sgier L. Visualizing BGP RIB changes into forwarding plane by leveraging BMP and IPFIX: дис. – ETH Zurich, 2020.
[29] Cisco ThousandEyes https://www.thousandeyes.com/
[30] goBGP Routing Daemon https://github.com/osrg/gobgp