

УДК 004.8:004.7

DOI: 10.31673/2412-9070.2025.050233

Ю. Л. ВЛАДАРЧИК, аспірант;

ORCID: 0009-0008-5271-6285

К. С. НЕСТЕРЕНКО, доктор техн. наук, професор,

ORCID: 0000-0001-7672-7386

Державний університет інформаційно-комунікаційних технологій, Київ

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПОЛЕГШЕНОГО АВТОНОМНОГО АГЕНТА ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ БАГАТОКРИТЕРІАЛЬНОЇ ОПТИМІЗАЦІЇ SRv6 У СЕРЕДОВИЩІ FRROUTING

Сучасні інфокомунікаційні мережі, які все частіше використовують сегментну маршрутизацію на базі IPv6 (SRv6) для гнучкості, стикаються із значними проблемами в динамічній багатокритеріальній оптимізації (БО). Традиційне централізоване управління має труднощі з масштабуванням та часом реакції, тоді як розгортання інтелектуальних систем безпосередньо на мережних пристроях з обмеженими ресурсами залишається складним. У цій статті розглядається можливість розгортання безконтрольного автономного агента штучного інтелекту (ШІ) безпосередньо на маршрутизаторі на базі Linux для виконання БО для інженерії трафіку SRv6. Ми представляємо принципи проєктування та стратегію реалізації полегшеного агента ШІ, призначеного для використання методів навчання з підкріпленням (RL) та/або графових нейронних мереж (GNN), придатних для середовищ з обмеженими ресурсами. Запропонований агент функціонує як окремий програмний процес, локально взаємодіючи із співрозташованим пакетом маршрутизації з відкритим кодом FRRouting через стандартні API (наприклад, YANG/NETCONF або REST через локальну міжпроцесну взаємодію (IPC)) для моніторингу стану мережі та застосування модифікацій політик SRv6.

Приведений детальний опис архітектури системи, концептуальних компонентів процесу прийняття рішень агентом (представлення стану, простір дій, концепції цільової функції), механізм локальної інтеграції на основі API з FRRouting та заплановану методологію реалізації з використанням Python зі стандартними бібліотеками ШІ в контейнеризованому середовищі (Docker) для валідації в Mininet. Основним внеском цієї роботи є демонстрація практичної можливості проєктування та реалізації цієї архітектури автономного агента на пристрої, що демонструє його здатність взаємодіяти із стандартною платформою маршрутизації, такою як FRRouting, для БО SRv6 та відкриває шлях до більш адаптивних та стійких стратегій управління мережею.

Ключові слова: автономне управління мережею; штучний інтелект; навчання з підкріпленням; сегментна маршрутизація; SRv6; багатокритеріальна оптимізація; FRRouting; модель; локальний API; міжпроцесна взаємодія (IPC).

Вступ

Невпинне зростання обсягу мережного трафіку, різноманітності послуг (зумовлене 5G, IoT та граничними обчисленнями) та динамічних вимог до продуктивності вимагає зміни парадигми від традиційного, часто ручного або статично налаштованого, управління мережею до більш автономних та інтелектуальних систем [1, 32, 33]. Сегментна маршрутизація на базі IPv6 (SRv6) забезпечує потужну основу для цієї еволюції, пропонуючи безпрецедентну гнучкість та програмованість шляхом кодування інструкцій маршрутизації безпосередньо у площині даних IPv6 [2, 6]. Однак використання цієї програмованості для ефективного управління шляхами SRv6 — одночасної оптимізації за кількома, часто суперечливими критеріями, такими як затримка, пропускна здатність, надійність та використання ресурсів — залишається значною проблемою, особливо в умовах динамічної зміни стану мережі [7].

Традиційні підходи до управління мережею, включаючи централізовані контролери програмно-визначених мереж (SDN), часто стикаються з обмеженнями масштабованості, потен-

ційними єдиними точками відмови та властивою затримкою, пов'язаною зі збором глобального стану та розподілом команд управління [3]. Вбудовування інтелекту для прийняття рішень безпосередньо у мережні пристрої є привабливою альтернативою, що потенційно дозволяє швидше реагувати на динаміку мережі на локальному рівні та підвищувати загальну стійкість системи. Незважаючи на цей потенціал, розгортання складних агентів ШІ на мережних маршрутизаторах часто ускладнюється властивими пристроям обмеженнями обчислювальних ресурсів (ЦП, пам'ять) [4, 5].

У цій статті досліджується можливість подолання цієї прогалини шляхом представлення проєкту та стратегії реалізації полегшеного автономного агента ШІ, спеціально розробленого для багатокритеріальної оптимізації (БО) SRv6. Ключовим аспектом нашого підходу є передбачувана робота агента безпосередньо на платформі маршрутизації на базі Linux, використовуючи широко поширений пакет маршрутизації з відкритим кодом FRRouting як платформу для інтеграції та валідації [26]. Запропонована архітектура агента передбачає запуск агента ШІ як окремого програмного процесу на тій самій хост-системі, що й домени FRRouting. Важливо, що взаємодія між агентом та FRRouting відбувається через визначені локальні API або механізми міжпроцесної взаємодії (IPC) [23, 29], що дозволяє агенту (потенційно розробленому на Python) відстежувати стан мережі, отриманий від FRRouting, та автономно застосовувати модифікації політик інженерії трафіку (TE) SRv6 до FRRouting без залежності від віддаленого контролера.

Центральне дослідницьке питання, що розглядається тут: чи можливо спроектувати та реалізувати функціональну, полегшену архітектуру автономного агента ШІ, яка локально інтегрується із стандартним пакетом маршрутизації, таким як FRRouting, для виконання багатокритеріальної оптимізації для SRv6?

Ця стаття робить наступні внески:

- Пропонує архітектуру системи для безконтролерного агента ШІ на пристрої, спеціально розробленого для БО SRv6 у середовищі FRRouting.
- Окреслює методологію вибору та реалізації полегшених моделей ШІ (на основі концепцій RL/GNN), придатних для мережних пристроїв з обмеженими ресурсами.
- Деталізує визначений підхід до інтеграції зовнішнього процесу агента ШІ з пакетом FRRouting за допомогою локальних API/IPC, що забезпечує моніторинг та управління.
- Обговорює стратегію реалізації з використанням стандартних інструментів (Python, бібліотеки ШІ, Docker, Mininet, GCP для навчання), необхідних для побудови системи та підготовки її до функціональної валідації.

Основна увага цього рукопису зосереджена виключно на проєкті, стратегії реалізації та отриманій оцінці можливості реалізації цієї архітектури автономного агента. Подальші дослідження, призначені для майбутніх публікацій, будуть присвячені кількісній оцінці покращень продуктивності та адаптивних можливостей агента на основі експериментальних результатів [31]. Демонструючи практичну можливість реалізації цього проєкту, ми робимо фундаментальний крок до реалізації більш адаптивного, стійкого та інтелектуального управління мережею, розгорнутого безпосередньо у мережній інфраструктурі.

Аналіз літературних даних та постановка проблеми. Проєктування автономного агента ШІ для БО SRv6 у середовищі FRRouting перетинається з кількома ключовими дослідницькими областями: мережі SRv6, багатокритеріальна оптимізація мереж, застосування ШІ/МО в мережах (зокрема, полегшені моделі) та розширюваність програмного забезпечення маршрутизації.

Інженерія та управління трафіком SRv6 пропонує нативні можливості програмування мережі, що дозволяє здійснювати деталізовану інженерію трафіку (TE) [8, 9]. Значна час-

тина існуючих робіт зосереджена на централізованому управлінні через контролери SDN або елементи обчислення шляху (PCE) для управління політиками TE SRv6 на основі глобального уявлення про мережу [10, 11]. Хоча ці централізовані підходи ефективні для певних сценаріїв, вони контрастують з нашою метою децентралізованої автономії на пристрої. Інші дослідження вивчають розподілені механізми SRv6, але часто не мають адаптивних можливостей навчання, властивих агентам на основі ШІ. Наша робота відрізняється тим, що пропонує та оцінює можливість реалізації автономного агента, що навчається, який працює на рівні вузла та приймає рішення на основі локального контексту.

Застосування методів багатокритеріальної оптимізації (БО) в мережах для збалансування суперечливих мережних цілей (затримка, пропускна здатність, вартість тощо) є добре усталеною галуззю [12, 13]. Техніки варіюються від класичних методів оптимізації до передових метаевристик, таких як NSGA-II. Однак багато рішень БО розроблені для офлайн-планування або централізованого виконання. Проблема полягає у виконанні БО динамічно та автономно в межах ресурсних обмежень мережного пристрою. Наш підхід використовує полегшений ШІ спеціально для вирішення цієї проблеми.

Застосування навчання з підкріпленням (RL) та графових нейронних мереж (GNN) для управління мережею швидко зростає [14, 15, 20]. RL дозволяє агентам вивчати оптимальні політики управління через взаємодію з середовищем, тоді як GNN надають потужний засіб для обробки даних мережної топології з графовою структурою [21, 22]. Однак багато запропонованих рішень ШІ вимагають значних обчислювальних ресурсів, що обмежує їх пряме застосування на стандартному мережному обладнанні. Критичним аспектом нашого дослідження є явний фокус на полегшених моделях ШІ, використовуючи такі методи, як скорочення моделі, квантування або ефективні архітектури [16, 17, 18, 19], щоб забезпечити можливість реалізації в середовищах з обмеженими ресурсами.

Інтеграція зовнішньої логіки з програмним забезпеченням маршрутизації промислового рівня є нетривіальним завданням. FRRouting, як пакет з відкритим кодом, пропонує кілька точок розширюваності, включаючи завантажувані модулі [24], внутрішні хуки та все більш стандартизовані північні API на основі YANG (доступні через NETCONF або gRPC) та потенційно REST [25, 29]. Хоча ці інтерфейси існують, дослідження, що деталізують їх конкретне використання для інтеграції автономних, полегшених агентів ШІ, які виконують складні завдання, такі як БО SRv6 через локальну IPC, залишаються відносно рідкісними. Існуючі приклади інтеграції часто зосереджені на моніторингу [27], простіших завданнях конфігурації або специфічних розширеннях протоколів.

Ця робота має на меті заповнити прогалину, надаючи детальний проєкт та аналіз можливості реалізації повної системи, що включає автономного, полегшеного агента ШІ, який виконує БО SRv6 через пряму локальну взаємодію з платформою FRRouting, тим самим демонструючи практичний шлях до вбудовування адаптивного інтелекту у стандартну інфраструктуру маршрутизації [30].

Основна частина

Запропонована архітектура вбудовує автономного агента ШІ у вузол маршрутизації, розташований разом із пакетом FRRouting на хост-системі Linux. Цей проєкт явно уникає залежності від зовнішніх контролерів для основного циклу оптимізації, маючи на меті підвищену стійкість та потенційно швидшу локальну адаптацію. Агент функціонує як незалежний програмний процес, який взаємодіє з демонами FRRouting через локальні API або механізми IPC.

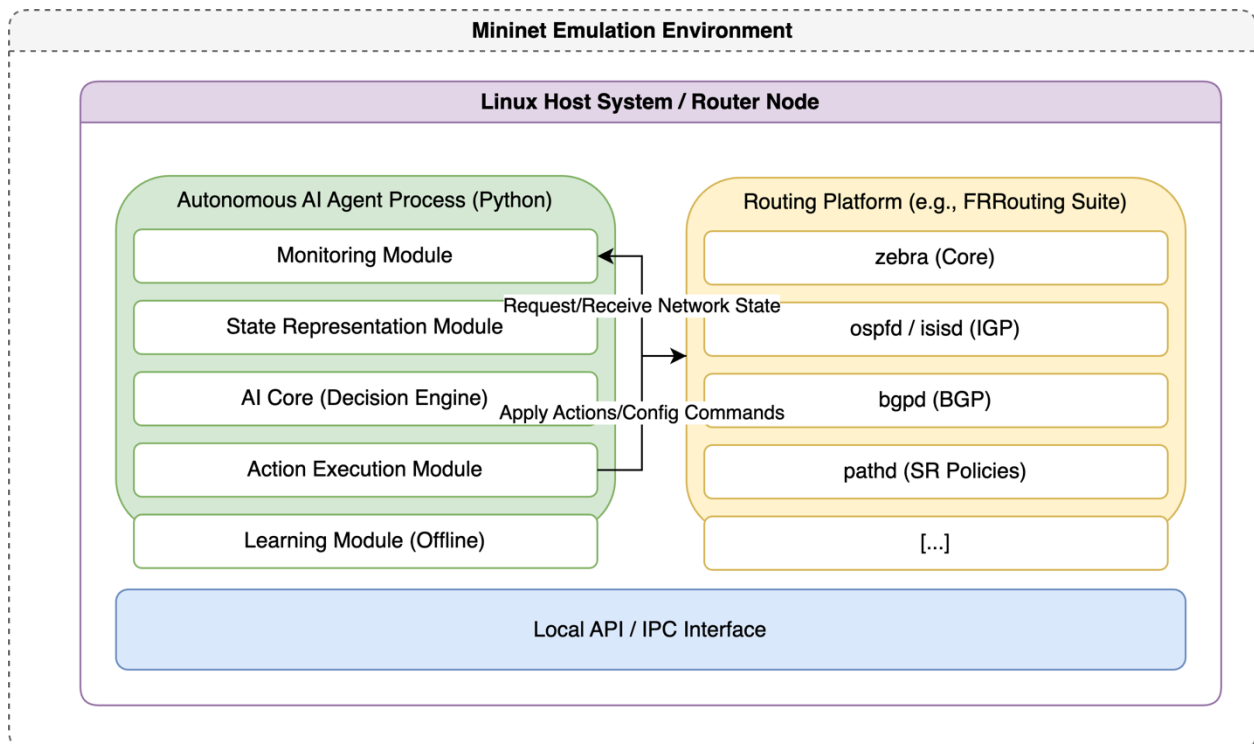


Рис. 1. Діаграма архітектури системи: діаграма, що ілюструє компоненти на хості Linux (демони FRRouting (zebra, bgpd, ospfd/isisd, pathd), процес агента ШІ (з внутрішніми модулями: моніторинг, представлення стану, ядро ШІ, виконання дій, навчання) та зв'язок через локальний API/IPC, що їх з'єднує)

Архітектура складається з наступних ключових компонентів:

- Хост-система Linux: Надає базове середовище операційної системи, мережний стек та контекст виконання.
- Пакет FRRouting: Платформа маршрутизації з відкритим кодом [28], відповідальна за запуск стандартних протоколів маршрутизації та управління функціями та політиками SRv6 [23].
- Процес автономного агента ШІ: Окремий процес на Python, що інкапсулює інтелект для БО SRv6. Включає модулі моніторингу, представлення стану, прийняття рішень (ядро ШІ), виконання дій та навчання (RL).

Модуль моніторингу: Запитує локальні API FRRouting [27] та інші джерела для збору даних про стан мережі. Модуль представлення стану: Перетворює дані у структурований формат для ШІ. Механізм прийняття рішень (ядро ШІ): Містить навчену полегшену модель ШІ для визначення оптимальної дії. Модуль виконання дій: Транслює рішення у команди для FRRouting. Модуль навчання (RL): Використовується для оновлення моделі ШІ.

- Локальний інтерфейс API / IPC: Канал зв'язку між агентом ШІ та FRRouting.

Ця роз'єднана, модульна архітектура сприяє незалежній розробці та оновленню.

Стратегія реалізації пріоритезує полегшені моделі ШІ та стандартні інструменти. **Python** обрано як мову. Ядро - **полегшені моделі ШІ**, щоб зменшити розмір та складність [16]. Кандидати - **RL (PPO, A2C)** [14] та **GNN (GCN, GAT)** [17], з можливим скороченням/квантуванням [18, 19].

Представлення стану буде отримуватися через API FRRouting. **Простір дій** відображатиметься на зміни політики SRv6. **Функція винагороди (RL)** буде інкапсулювати множинні цілі.

Зв'язок через **локальний API або IPC**. Кандидати [23, 24, 25]: YANG/gRPC, REST [29], OSPF API [27] або хуки. Модель **локальної взаємодії** забезпечує автономію.

Використовується **контейнеризоване симуляційне середовище**. Агент та FRRouting – у **контейнерах Docker**. **Docker Compose** для управління. **Mininet** для емуляції топологій. **Робочий процес** включає розробку, пакування, симуляцію, оркестрацію, офлайн-навчання (GCP) та валідацію.

Основна мета - встановити **практичну можливість реалізації**. Оцінка ґрунтується на успішному проектуванні та реалізації. Можливість реалізації спирається на зрілі технології. Можливість інтеграції залежить від взаємодії через API. Загальна архітектурна можливість встановлюється демонстрацією у симуляції.

Висновки

У цій статті детально описано обґрунтування проекту, архітектуру системи та стратегію реалізації полегшеного автономного агента III для багатокритеріальної оптимізації SRv6. Ключовою особливістю є робота агента як окремого процесу, інтегрованого локально з FRRouting через стандартні API. Представлена робота встановлює практичну можливість побудови такої архітектури. Використовуючи стандартні інструменти та IPC, ми демонструємо життєздатний шлях для вбудовування логіки III у мережні елементи з FRRouting. Це основа для майбутніх досліджень адаптивних, стійких та децентралізованих систем управління. Подальші дослідження зосередяться на кількісній оцінці, бенчмаркінгу, оцінці адаптивності, аналізі ресурсів та дослідженні вдосконалень. Ця робота є значним кроком до реалізації потенціалу автономних III-агентів для управління мережами SRv6.

Список літератури

1. NGMN Alliance. Automation and Autonomous System Architecture Framework. Accessed April 3, 2025. (https://www.ngmn.org/wp-content/uploads/Automation_and_Autonomous_System_Architecture_Framework.pdf).
2. Huawei Technical Support. What Is SRv6? How Does SRv6 Work? Accessed April 3, 2025. (<https://info.support.huawei.com/info-finder/encyclopedia/en/SRv6.html>).
3. Bharatiya, P. Understanding AI Agents and Agentic AI: A Comprehensive Guide. Pronod Bharatiya's Blog. Accessed April 3, 2025. (<https://data-intelligence.hashnode.dev/understanding-ai-agents-agentic-ai-guide>).
4. YouTube. What is Agentic AI? Create your Own Autonomous AI Agent with Python | GenAI | Machine Learning | NLP. Accessed April 3, 2025. (https://www.youtube.com/watch?v=dsbjMbK_9Gw).
5. Arsénio, A. M., et al. Research Challenges in Coupling Artificial Intelligence and Network Management. IETF Draft. Accessed April 3, 2025. (<https://www.ietf.org/id/draft-irtf-nmrq-ai-challenges-05.html>).
6. Cisco Press. Chapter 3. What Is Segment Routing over IPv6 (SRv6)? Accessed April 3, 2025. (<https://www.ciscopress.com/articles/article.asp?p=3203556&seqNum=2>).
7. Jakab, F., et al. Segment Routing: A Comprehensive Survey of Research Activities, Standardization Efforts, and Implementation Results. Accessed April 3, 2025. (http://www.hit.bme.hu/~jakab/edu/litr/SegRou/Segment_Routing_A_Comprehensive_Survey_of_Research_Activities_Standardization_Efforts_and_Implementation_Results.pdf).
8. Filsfils, C., et al. RFC 8986: Segment Routing over IPv6 (SRv6) Network Programming. IETF Datatracker. Accessed April 3, 2025. (<https://datatracker.ietf.org/doc/rfc8986/>).
9. APNIC Blog. What is SRv6 network programming? Accessed April 3, 2025. (<https://blog.apnic.net/2020/05/01/what-is-srv6-network-programming/>).
10. Li, Z., et al. Routing Optimization Mechanism for SRv6 Based Power Data Network. ResearchGate. Accessed April 3, 2025. (https://www.researchgate.net/publication/376828109_Routing_Optimization_Mechanism_for_SRv6_Based_Power_Data_Network).
11. Filsfils, C., et al. RFC 9256: Segment Routing Policy Architecture. RFC Editor. Accessed April 3, 2025. (<https://www.rfc-editor.org/rfc/rfc9256.html>).
12. Wikipedia. Multiple-criteria decision analysis. Accessed April 3, 2025. (https://en.wikipedia.org/wiki/Multiple-criteria_decision_analysis).
13. Wikipedia. Multi-objective optimization. Accessed April 3, 2025. (https://en.wikipedia.org/wiki/Multi-objective_optimization).

14. Liu, Y., et al. *Lightweight Reinforcement Learning for Energy Efficient Communications in Wireless Sensor Networks*. ResearchGate. Accessed April 3, 2025. (https://www.researchgate.net/publication/331512412_Lightweight_Reinforcement_Learning_for_Energy_Efficient_Communications_in_Wireless_Sensor_Networks).
15. Nemoto, K., & Matsutani, H. *A Lightweight Reinforcement Learning Based Packet Routing Method Using Online Sequential Learning*. Keio University. Accessed April 3, 2025. (https://www.arc.ics.keio.ac.jp/~matutani/papers/nemoto_ieicej2023.pdf).
16. Li, J., et al. *Towards Lightweight and Robust Machine Learning for CDN Caching*. Carnegie Mellon University. Accessed April 3, 2025. (https://pdl.cmu.edu/PDL-FTP/BigLearning/2018MachineLearningCDNcache_HOTNETS.pdf).
17. Liu, X., et al. *Lightweight Graph Neural Network Search with Graph Sparsification*. OpenReview. Accessed April 3, 2025. (<https://openreview.net/forum?id=IefMMX12yk>).
18. Liu, Z., et al. *ALGNN: Auto-Designed Lightweight Graph Neural Network*. ResearchGate. Accessed April 3, 2025. (https://www.researchgate.net/publication/355539525_ALGNN_Auto-Designed_Lightweight_Graph_Neural_Network).
19. Wang, Y., et al. *A Lightweight Method for Graph Neural Networks Based on Knowledge Distillation and Graph Contrastive Learning*. MDPI. Accessed April 3, 2025. (<https://www.mdpi.com/2076-3417/14/11/4805>).
20. Chowdhury, S. R., et al. *Achieving Network Resilience through Graph Neural Network-enabled Deep Reinforcement Learning*. arXiv. Accessed April 3, 2025. (<https://arxiv.org/html/2501.11074v1>).
21. Zhang, Y., et al. *Resource Allocation Based on Graph Neural Networks in Vehicular Communications*. NSF-PAR. Accessed April 3, 2025. (<https://par.nsf.gov/servlets/purl/10286411>).
22. Yu, M., et al. *GNN-RL Compression: Topology-Aware Network Pruning using Multi-stage Graph Embedding and Reinforcement Learning*. Iowa State University. Accessed April 3, 2025. (https://swapp.cs.iastate.edu/files/inline-files/yu_ea_gnn_compression_arxiv_feb_2021.pdf).
23. FRRouting. *Process Architecture — FRR latest documentation*. Accessed April 3, 2025. (<https://docs.frrouting.org/projects/dev-guide/en/latest/process-architecture.html>).
24. FRRouting. *Modules — FRR latest documentation*. Accessed April 3, 2025. (<https://docs.frrouting.org/projects/dev-guide/en/latest/modules.html>).
25. FRRouting. *Architecture — FRR latest documentation (Northbound API)*. Accessed April 3, 2025. (<https://docs.frrouting.org/projects/dev-guide/en/latest/northbound/architecture.html>).
26. NVIDIA Developer. *FRRouting*. Accessed April 3, 2025. (<https://developer.nvidia.com/networking/ethernet-switches/frrouting>).
27. FRRouting. *OSPF API Documentation — FRR latest documentation*. Accessed April 3, 2025. (<https://docs.frrouting.org/projects/dev-guide/en/latest/ospf-api.html>).
28. FRRouting. *The FRRouting Protocol Suite*. GitHub. Accessed April 3, 2025. (<https://github.com/FRRouting/frr>).
29. Omidznlp. *REST-API-FRRouting*. GitHub. Accessed April 3, 2025. (<https://github.com/Omidznlp/REST-API-FRRouting>).
30. Boucadair, M., & Jacquenet, C. *Research Challenges in Coupling Artificial Intelligence and Network Management*. IETF. Accessed April 3, 2025. (<https://www.ietf.org/id/draft-irtf-nmrg-ai-challenges-05.html>).
31. Bondan, L., et al. *SRPerf: a Performance Evaluation Framework for IPv6 Segment Routing*. arXiv. Accessed April 3, 2025. (<https://arxiv.org/abs/2001.06182>).
32. YouTube. *Introduction to AI Agents - Theory and Code*. Accessed April 3, 2025. (<https://www.youtube.com/watch?v=UMYKjT9exb4>).
33. Athina AI. *Understanding Autonomous AI Agents: Components, Challenges, and Future Prospects*. Medium. Accessed April 3, 2025. (<https://medium.com/athina-ai/understanding-autonomous-ai-agents-components-challenges-and-future-prospects-51db55fe0adb>).

Yu. Vladarchyk, K. Nesterenko

DESIGN AND IMPLEMENTATION OF A LIGHTWEIGHT AUTONOMOUS AI AGENT FOR SRv6 MULTI-CRITERIA OPTIMIZATION WITHIN FRRROUTING

Modern info-communication networks, increasingly leveraging Segment Routing over IPv6 (SRv6) for enhanced flexibility and programmability, face significant challenges in achieving dynamic and effective multi-criteria optimization (MCO). The relentless growth in network traffic volume, the escalating diversity of services driven by 5G, IoT, and edge computing, and the highly dynamic nature of performance demands necessitate a paradigm shift from traditional, often manual or statically configured, network management towards more autonomous and intelligent control systems. SRv6 provides a powerful architectural foundation for this evolution by encoding routing instructions directly within the IPv6 data plane, yet harnessing this programmability to simultaneously optimize multiple, often conflicting, criteria—such as latency, throughput, reliability, and resource utilization—remains a complex undertaking, particularly under fluctuating network conditions and diverse application requirements.

Traditional centralized network management approaches, including those based on Software-Defined Networking (SDN) controllers, often encounter limitations related to scalability, potential single points of failure, and the inherent latency involved in collecting global network state and distributing control commands. Conversely, deploying sophisticated decision-making intelligence directly onto network devices, while offering the promise of faster localized responses and enhanced resilience, is frequently hindered by the inherent constraints in computational resources (CPU, memory) typical of standard routing hardware. This paper specifically addresses the critical feasibility of designing and deploying a controller-less, autonomous Artificial Intelligence (AI) agent directly on a Linux-based routing platform to perform MCO for SRv6 traffic engineering. A core aspect of our investigation is the agent's seamless integration with the widely adopted FRRouting open-source routing suite, which serves as the operational platform for both monitoring network state and enacting SRv6 policy modifications.

We present the detailed design principles and a comprehensive implementation strategy for a lightweight AI agent. This agent is specifically architected to utilize resource-efficient Reinforcement Learning (RL) and/or Graph Neural Network (GNN) techniques, which are particularly well-suited for operation within such constrained environments. The proposed agent functions as a distinct software process, running independently yet interacting locally with the co-located FRRouting daemons (e.g., zebra, bgpd, ospfd/isisd, pathd). This interaction is facilitated through standard, well-defined Application Programming Interfaces (APIs), such as YANG/NETCONF, gRPC, or REST, operating over local Inter-Process Communication (IPC) mechanisms. This local API-driven approach enables the agent to continuously monitor relevant network state parameters derived from FRRouting and autonomously apply SRv6 Traffic Engineering (TE) policy modifications back to the FRRouting suite without reliance on any remote controller.

The paper meticulously details the overall system architecture, the conceptual components underpinning the agent's intelligent decision-making process (including state representation derived from network telemetry, the defined action space corresponding to SRv6 policy controls, and the formulation of multi-criteria objective functions), the specific local API-based integration mechanisms with FRRouting and the planned implementation methodology. This methodology leverages Python as the primary development language, augmented by standard AI libraries (e.g., TensorFlow, PyTorch), and employs a containerized environment (Docker) for consistent deployment and rigorous validation within the Mininet network emulator. The offline training of the AI models is envisioned to utilize scalable cloud platforms like Google Cloud Platform (GCP) to handle computational demands. The primary scientific and practical contribution of this work lies in the thorough demonstration of the feasibility of designing and implementing this novel autonomous, on-device agent architecture. By showcasing its capability to effectively interact with a standard, production-grade routing platform like FRRouting for the purpose of SRv6 MCO, this research paves the way for the development of more adaptive, resilient, and intelligent network control strategies deployed directly within the network infrastructure, thereby fostering a new generation of decentralized and autonomous network management solutions.

Keywords: autonomous network management; artificial intelligence; reinforcement learning; segment routing; SRv6; multi-criteria optimization; FRRouting; model; local API; inter-process communication (IPC).