

HawkV6: Application-Specific Traffic Steering with Intent-Based Networking and SRv6

Severin Dellsperger and Daniel Politze

Institute for Network and Security

Eastern Switzerland University of Applied Sciences

Rapperswil, Switzerland

{severin.dellsperger, daniel.politze}@ost.ch

Abstract—Existing traffic steering mechanisms often struggle to meet the dynamic requirements of today’s applications. This paper demonstrates a novel approach that leverages Segment Routing over IPv6 (SRv6) and Intent-Based Networking (IBN) principles to enable advanced, application-aware end-to-end traffic engineering between clients and servers. The solution illustrates how high-level, application-specific traffic steering intents are seamlessly translated into dynamic network instructions, thereby eliminating the need for complex, manual network steering policies. The system further incorporates standardized protocols to enable real-time telemetry data collection and analysis, pioneering next-level traffic engineering capabilities that were previously unachievable. This demonstration showcases the practical feasibility of this approach through targeted use cases, leveraging its open-source foundation, and highlights the significant benefits of creating highly dynamic networks that autonomously adapt to changing conditions and application requirements.

Index Terms—Intent-Based Networking, IBN, Segment Routing, SRv6, Telemetry, Traffic Steering, Traffic Engineering

I. INTRODUCTION

Traffic steering mechanisms, such as Multiprotocol Label Switching Traffic Engineering (MPLS-TE) and Segment Routing Traffic Engineering (SR-TE), are widely applied to manage data flow and optimize resource utilization in modern communication networks. These techniques offer valuable advantages, allowing for precise traffic steering decisions. This is achieved using metrics beyond basic Interior Gateway Protocol (IGP) costs, including critical performance indicators like link delay, jitter, and packet loss. Centralized components, such as the Path Computation Element (PCE), have provided a central point for path calculation and control, aiming to enhance network management.

Despite these advances, the application of these standard traffic engineering techniques faces significant challenges in meeting the evolving requirements of future networks. The following limitations, which are addressed by this demonstration, highlight an urgent need for a more flexible, intelligent, and application-aware approach to traffic steering:

- The configuration and maintenance of traffic engineering policies remain inherently complex and cumbersome, demanding specialized expertise.
- The underlying rules are often limited to distinct use cases, preventing the integration of advanced traffic en-

gineering concepts such as dynamic service function chaining.

- Policies are predominantly enforced at the network level, which inherently disregards the specific needs of individual applications and breaks the fundamental end-to-end communication model [1].

A compelling use case is the dynamic optimization of data paths for distinct applications in a network. For instance, while one application requires the lowest possible latency, another needs to send a backup stream over a less-utilized path to optimize for cost and improve network utilization. Rather than relying on complex, static network configurations, a promising approach allows the operator to declare a high-level intent for each application, and the system will autonomously ensure that packets follow the intended path and adapt to real-time network changes.

This paper presents a novel proof of concept addressing these shortcomings, building upon the concepts and system detailed in our previous paper [2]. Our HawkV6 solution leverages Intent-Based Networking (IBN) principles and Segment Routing over IPv6 (SRv6). It translates high-level, application-specific traffic steering intents into dynamic network instructions, which are then embedded directly into the IPv6 packet header within a Segment Routing Header (SRH) as a segment list composed of Segment Identifiers (SIDs). This empowers end-users and applications to influence their data paths without complex, manual policies. The entire HawkV6 system is open-source [3], and its integration of real-time telemetry allows continuous network monitoring for intent compliance, pioneering next-level traffic engineering responsiveness. The subsequent sections outline the system architecture, present compelling demonstration scenarios, and highlight the significant practical benefits of this innovative approach.

II. SYSTEM OVERVIEW

The HawkV6 system architecture, illustrated in Figure 1, comprises several components that translate traffic steering intents into dynamic network instructions and enforce them. The entire codebase for HawkV6 is publicly available on GitHub at [3]. The key components of the HawkV6 system include:

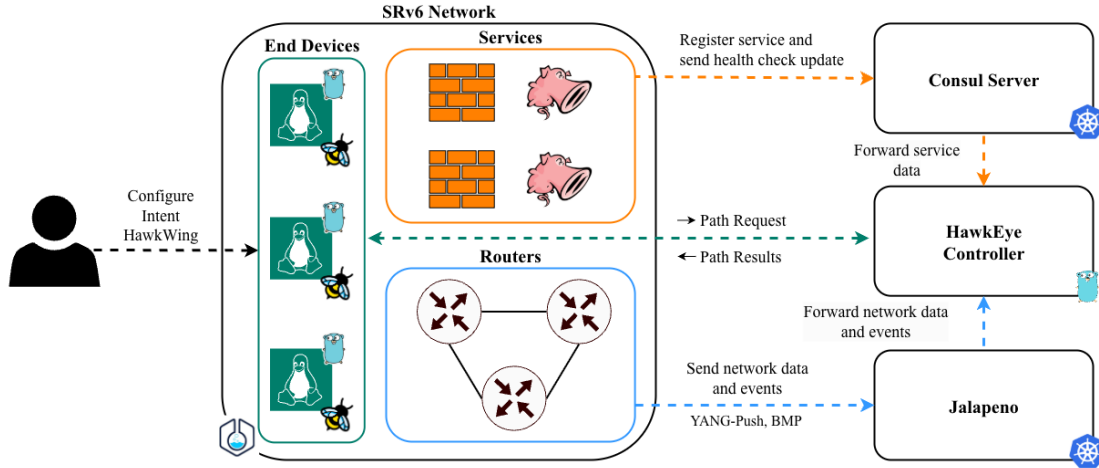


Fig. 1: High-level System Architecture

A. Linux End Devices with HawkWing

Linux end devices act as clients, running the HawkWing Go application. This application enables the declaration of application-specific traffic intents. To obtain the necessary path instructions, HawkWing sends path requests to the central controller, HawkEye. Upon receiving these path results, it performs the SRv6 encapsulation of outgoing IPv6 application packets with the provided SRH to ensure traffic follows the intended path.

B. SRv6 network

The SRv6 network, primarily Cisco IOS XR routers deployed in Containerlab, forwards traffic based on SRH instructions. Through standardized protocols such as YANG-Push and BGP Monitoring Protocol (BMP), routers continuously send real-time network telemetry data to Jalapeno, providing a detailed view of the operational network.

C. Segment Routing Aware Services

SRv6-aware firewalls and intrusion detection system (IDS) services are deployed within the SRv6 network, which is integral for advanced traffic engineering and service function chaining (SFC). Each service registers with the Consul server and reports its health, ensuring only operational services are considered for path computation.

D. Jalapeno

Jalapeno is the central telemetry platform. It ingests, processes, stores, and visualizes network metrics from the SRv6 network, making this critical state information available to the HawkEye controller.

E. Consul server

The Consul server functions as the service registry, receiving service registration and health updates from SRv6-aware services. This data is provided to the HawkEye controller

for dynamic service function chaining, enabling optimal path calculation based on real-time service availability.

F. HawkEye controller

The HawkEye controller, the core of HawkV6, receives path requests from HawkWing and gathers all necessary real-time network telemetry and service data. It calculates the optimal SRv6 segment list for retrieved intents, taking into account application requirements and current network conditions. This segment list is sent back to HawkWing, forcing application packets to take the intended path. HawkEye continuously receives network updates, sending revised path results to the client upon changes to ensure appropriate path adaptation.

III. DEMONSTRATION SCENARIOS AND EXPERIENCE

The demonstration provides an interactive experience, showcasing the advanced concepts and capabilities of Intent-Based Networking and SRv6 for traffic steering. Attendees gain direct insight into how application-aware traffic engineering is achieved and dynamically adapted in a production-like network environment. The demonstration setup uses a virtualized network testbed, allowing for the emulation of diverse network conditions and the deployment of various services. Since this testbed leverages vendor network operating systems and their implementation of standardized protocols, it closely mirrors real-world network environments. The demonstration presents a complete cycle of traffic steering driven by high-level intents, encompassing: 1) defining application-specific intents; 2) observing real-time data gathering and visualization; and 3) ultimately executing instructions embedded in the application packets. To illustrate these capabilities, two distinct scenarios are considered. These collectively provide a comprehensive understanding of HawkV6's technical capabilities, its practical feasibility, and the significant benefits it offers.

A. Scenario 1: Low-Latency Traffic Steering

The first scenario introduces the audience to the HawkV6 framework, beginning with an overview of the virtualized network environment and the real-time distribution and visualization of network metrics. The audience observes key performance indicators, such as link delays, on a dedicated dashboard, as depicted in Figure 2. This gives insights into the network’s operational state and establishes the foundation for the subsequent demonstration of dynamic traffic steering. The audience learns how to use the HawkWing client and how to define a declarative intent for a specific application’s traffic. This demonstration uses a low-latency intent. Building on this foundation, the demonstration highlights the system’s interaction with real-time data collection and network monitoring. When application traffic, such as audio streams, flows between client and server, the audience observes dynamic path changes triggered by emulated network events, including an increased link delay or even a complete link outage. The HawkEye controller, powered by real-time telemetry from Jalapeno, dynamically calculates and sends new SRv6 instructions to the client. Thus, the audience gains a firsthand experience of the system’s responsiveness by observing real-time changes in both the packet header’s segment list and the dashboard metrics, which demonstrates how the system autonomously adapts to maintain the low-latency intent.



Fig. 2: Visualization of Real-time Delay Metrics

B. Scenario 2: Dynamic Service Function Chaining with Health-Aware Routing

The second scenario delves into a more advanced use case: dynamically steering traffic through an optimal service function chain based on application intent. This is particularly important for modern cloud and telecommunications networks where services need to be deployed and scaled on demand. In this demonstration, the SFC intent directs application traffic to pass through a firewall first and then through an IDS. The HawkEye controller actively searches for the best combination of available service instances to construct this optimal service chain, considering both their strategic placement within the

network and the primary optimization metric, which is packet loss in this scenario. This part further elaborates on the role of the Consul server, which acts as the service registry, and how SRv6-aware services continuously report their health status. The scenario starts with traffic being steered through an optimized service chain. Next, a critical event is simulated where a service instance within the chain fails, triggering a health check failure. This health check failure is propagated to the Consul web dashboard and can be observed live by the audience. An example is shown in Figure 3. The Consul server instantly registers this failure and propagates it to the HawkEye controller. HawkEye promptly recalculates the path to bypass the unhealthy service instance and selects a valid and optimal service chain that meets the application’s requirements. The audience observes this seamless re-routing and the continuous optimization of the service path, thus demonstrating HawkV6’s ability to ensure service resilience and performance in the face of dynamic service availability.

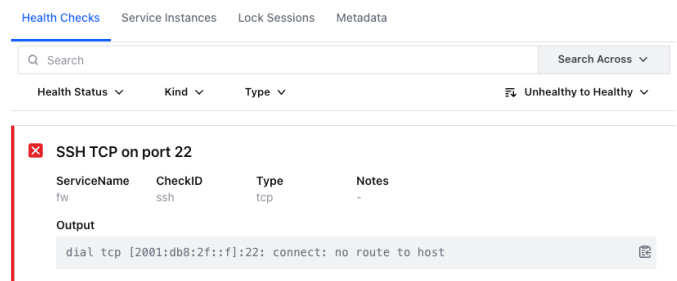


Fig. 3: Failing Service Health Check Consul Web Dashboard

IV. CONCLUSION

This proposed demonstration showcases an innovative approach to end-to-end traffic engineering, combining Intent-Based Networking and Segment Routing over IPv6. We directly addressed limitations of traditional traffic steering by presenting how high-level per-application intents are dynamically translated into network instructions, enabling application-aware traffic steering and restoring the end-to-end communication model. Our work highlights the practical feasibility of autonomous network adaptation to changing conditions and service states, including dynamic service function chaining, critically facilitated by standardized real-time telemetry protocols and open-source components. The underlying conceptual approach paves the way for highly dynamic and responsive networks and represents a crucial step in ensuring future communication infrastructure resilience and optimal performance.

REFERENCES

- [1] L. He, S. Wang, Y. Xu, P. Kuang, J. Cao, Y. Liu, X. Li, and S. Peng, “Enabling Application-Aware Traffic Engineering in IPv6 Networks,” *IEEE Network*, vol. 36, no. 2, pp. 42–49, 2022.
- [2] S. Dellsperger and D. Politze, “Leveraging Intent-Based Networking and SRv6 for Dynamic End-to-End Traffic Steering,” in *NOMS 2025-2025 IEEE Network Operations and Management Symposium*, 2025, pp. 1–9.
- [3] “hawkv6 — github.com,” <https://github.com/hawkv6>, [Accessed 07-08-2025].