

(REVIEW ARTICLE)



Evaluating SRv6 architecture, migration paths and operational impacts in service-provider networks

Kwabena Akomea-Agyin *, Carl Amekudzi and Jibriel Adjei

J. Warren McClure School of Emerging Communication Technologies, Ohio University, USA.

World Journal of Advanced Engineering Technology and Sciences, 2026, 18(03), 442-452

Publication history: Received on 11 February 2026; revised on 20 March 2026; accepted on 23 March 2026

Article DOI: <https://doi.org/10.30574/wjaets.2026.18.3.0169>

Abstract

Multiprotocol Label Switching (MPLS) has been the foundation of service-provider networks for many years, supporting traffic engineering and scalable VPN services. However, modern networks now demand greater programmability, automation, and native IPv6 support, which adds complexity to traditional MPLS operations. Segment Routing over IPv6 (SRv6) offers a new approach by embedding forwarding instructions directly into IPv6 packet headers, removing the need for separate label-distribution protocols and enabling more flexible traffic control. This paper evaluates SRv6 as a natural evolution of MPLS by comparing the architectures of MPLS, SR-MPLS, and SRv6, and by examining practical migration strategies for service providers. The aim is to provide a clear and measurable understanding of SRv6 architecture, migration options, and operational impact to support real-world deployment decisions.

Keywords: Segment Routing over IPv6 (SRv6); Multiprotocol Label Switching (MPLS); Segment Routing (SR); Service Provider Networks; Traffic Engineering; Network Programmability; IPv6 Networks; Network Migration

1. Introduction

Multiprotocol Label Switching (MPLS) has been the primary transport technology in service-provider networks for more than twenty years. Since the MPLS architecture was introduced in RFC 3031, it has enabled scalable traffic engineering, VPN services, and differentiated service delivery across large backbone networks [1]. One of its key innovations is label-based forwarding, which separates forwarding decisions from traditional IP routing [2]. This approach allows network operators to implement explicit routing and service overlays while maintaining high performance, making MPLS the preferred transport solution for both carrier networks and enterprise WANs.

Today, network demands are changing rapidly. The growth of cloud services, video traffic, and edge applications is driving the need for greater automation, programmability, and more dynamic traffic control. Traditional MPLS relies on multiple control-plane protocols such as LDP, RSVP-TE, and BGP for service signaling [3], [4]. Managing and troubleshooting these protocols adds operational complexity, especially in large-scale networks with many engineered paths.

Segment Routing (SR) was introduced to address these challenges by simplifying how paths are controlled in the network. Instead of relying on distributed signaling protocols, SR shifts path control to ingress routers or centralized controllers. It represents forwarding instructions as an ordered list of segments carried in packet headers [5]. Early implementations used the MPLS data plane, known as SR-MPLS, which removed the need for LDP while remaining compatible with existing MPLS environments [6]. Although SR-MPLS simplifies the control plane, it still depends on label stacks and does not fully take advantage of IPv6 capabilities.

* Corresponding author: Kwabena Akomea-Agyin.

Segment Routing over IPv6 (SRv6) builds on this concept by embedding forwarding instructions directly into IPv6 packets using the Segment Routing Header (SRH). In this model, segments are represented as IPv6 addresses that define both location and behavior [5], [7]. This creates a network programming approach where transport and service functions are integrated within the IPv6 data plane, enabling more flexible traffic steering, service chaining, and better alignment with software-defined networking.

This paper evaluates SRv6 as a natural evolution of MPLS. It examines the technical foundations of SRv6 and compares its control-plane and data-plane characteristics with those of MPLS and SR-MPLS. It also explores practical migration strategies that allow operators to transition gradually without disrupting existing services. In addition, the paper looks at operational considerations such as hardware readiness, performance impact, and required changes to management systems for large-scale deployment. The overall goal is to provide guidance for network architects who are considering SRv6 as the next generation of transport architecture.

1.1. Problem Statement

Service-provider networks have relied on MPLS for many years because it supports traffic engineering and scalable service delivery. However, as network demands grow with cloud computing, video traffic, and edge applications, MPLS-based architectures are becoming more complex to operate. This is mainly due to their dependence on multiple control-plane protocols such as LDP, RSVP-TE, and BGP, which increase operational overhead and make automation and troubleshooting more difficult in large-scale environments. While Segment Routing and SRv6 offer a simplified and more programmable approach, there is still a lack of clear, practical guidance on how these technologies compare to MPLS in real deployments, how service providers can migrate without disrupting existing services, and what operational challenges they should expect. As a result, network operators face uncertainty when deciding whether and how to transition to SRv6-based architectures.

1.2. Research Objectives

The main objective of this study is to evaluate Segment Routing over IPv6 (SRv6) as a practical evolution of MPLS for modern service-provider networks. Specifically, the study aims to compare the architectures of MPLS, SR-MPLS, and SRv6 in terms of control-plane design, forwarding behavior, scalability, and operational complexity. It also seeks to examine realistic migration strategies that allow MPLS and SRv6 to coexist during transition, ensuring service continuity and reduced deployment risk. In addition, the research analyzes key operational factors such as hardware readiness, performance impact, monitoring capabilities, and automation requirements for large-scale SRv6 deployment. Overall, the goal is to provide a clear and structured understanding that can guide network architects in making informed decisions about adopting SRv6 in real-world environments.

2. Literature Review

The evolution of service-provider network architectures has been shaped by the need for scalable forwarding, flexible traffic engineering, and efficient service delivery. MPLS was introduced to improve traffic control in IP networks by using label-based forwarding instead of traditional destination-based routing. RFC 3031 defined the MPLS architecture and introduced a model where packets follow label-switched paths across provider networks, making it possible to support scalable traffic engineering and VPN services [1]. Over time, MPLS was enhanced with RSVP-TE to enable advanced traffic engineering and with BGP extensions to support large-scale VPN deployments [3], [4].

While MPLS brought significant benefits, both researchers and network operators identified several operational challenges. RSVP-TE depends on distributed signaling and requires routers along the path to maintain state, which can limit scalability in large networks [8]. In addition, running multiple protocols such as LDP, RSVP-TE, and BGP together increases operational complexity and makes automation and troubleshooting more difficult [9]. These challenges led to the search for new traffic-engineering approaches that could simplify the control plane while still maintaining precise control over network paths.

Segment Routing was introduced as one such solution. It simplifies the network by removing the need for signaling protocols and instead encoding forwarding instructions as segments within packet headers. In this model, ingress routers or centralized controllers define explicit paths, and there is no need to maintain per-flow state across the network [5]. Early implementations used the MPLS data plane, known as SR-MPLS, which allowed operators to adopt segment routing within existing MPLS environments while reducing control-plane complexity and eliminating the need for LDP [6].

Segment Routing over IPv6 (SRv6) further extends this concept by embedding forwarding instructions directly into IPv6 packets using the Segment Routing Header (SRH). In SRv6, segments are represented as IPv6 addresses that encode both location and behavior [5], [7]. This introduces a network programming model where traffic-steering policies and service functions are defined through segment lists. Research has explored several SRv6 use cases, including service-function chaining, programmable traffic engineering, and integration with software-defined networking architectures [10].

Recent studies have also examined SRv6 in emerging environments such as 5G transport networks and cloud data-center fabrics. SRv6 enables tighter integration between transport networks and virtualized network functions by embedding service behaviors directly into segment identifiers [11]. Operational studies have evaluated important factors such as header overhead, hardware support, and how SRv6 can coexist with MPLS during migration phases [12], [13].

Despite these developments, several challenges still exist when transitioning from established MPLS networks to SRv6 architectures. These challenges include defining effective migration strategies, ensuring interoperability between MPLS and SRv6 domains, and preparing monitoring and automation tools for the new architecture [14], [15]. Understanding these issues is critical for network operators considering SRv6 adoption. This study contributes to this area by examining SRv6 as a practical evolution of MPLS and by analyzing its architecture, migration strategies, and operational considerations for real-world deployment.

3. Methodology

This study evaluates Segment Routing over IPv6 (SRv6) as a natural evolution of traditional MPLS-based transport architectures. It uses a combination of comparative architectural analysis and a review of real-world deployment practices. The analytical approach is grounded in established protocol specifications and prior research on MPLS traffic engineering, segment routing design, and IPv6-based service programming [1], [5], [7], [16].

The study begins by comparing three forwarding models: traditional MPLS, SR-MPLS, and SRv6. This comparison focuses on key areas such as control-plane design, forwarding behavior, scalability, traffic-engineering capabilities, and overall operational complexity. The criteria used for this evaluation are based on earlier research on MPLS scalability and segment routing operations [8], [12], [16].

Next, the study explores migration strategies that allow MPLS and SRv6 networks to operate together during the transition period. This includes examining interworking mechanisms and gateway-based translation models that enable traffic to move smoothly between different forwarding environments as deployment progresses in phases [14], [17].

Finally, the study looks at the operational aspects of deploying SRv6. It reviews existing implementation studies that evaluate performance, hardware support, and the scalability of segment routing architectures [10], [11], [18]. Important factors considered include packet overhead, the ability of hardware to support SRv6 forwarding, and how well SRv6 integrates with automation frameworks.

By combining architectural comparisons with practical operational insights, this methodology provides a structured way to evaluate the feasibility and real-world impact of adopting SRv6 in modern service-provider networks.

3.1. SRv6 Architecture and Protocol Mechanics

Segment Routing over IPv6 (SRv6) builds on the segment routing concept by placing forwarding instructions directly inside IPv6 packet headers. Instead of relying on MPLS label stacks, SRv6 uses IPv6 addresses as segment identifiers, known as SIDs. This approach allows both routing information and service functions to operate within a single addressing framework [7], [19]. As a result, SRv6 supports flexible traffic steering and service programming while using standard IPv6 forwarding mechanisms.

SRv6 introduces the Segment Routing Header (SRH), which is an IPv6 extension header that carries a list of segments defining the path a packet should follow. Each segment represents a specific instruction that is executed by a network node. The SRH includes the segment list, a Segments Left counter, and optional policy fields used for traffic engineering. When a packet enters the network, the ingress router adds the SRH and encodes the intended path as an ordered list of SIDs [7].

A key concept in SRv6 is the Segment Identifier itself. Each SID is a 128-bit IPv6 address, structured into different components, as illustrated in Figure 1.

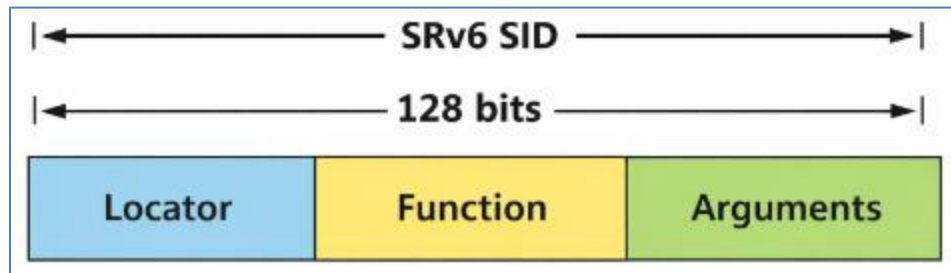


Figure 1 SRv6 SID structure

In SRv6, each segment identifier is structured to carry specific meaning. The locator identifies the node responsible for processing the segment, the function defines the action that should be performed, and the optional arguments field provides additional parameters for that function [7], [19]. This structure supports both simple forwarding and more advanced operations such as service chaining and the execution of network functions.

Packet forwarding in SRv6 follows a predictable and well-defined process. When a packet reaches a node whose locator matches the destination address, that node recognizes itself as the active segment endpoint. It then performs the required function, reduces the Segments Left value, and replaces the IPv6 destination address with the next segment in the list. The packet is then forwarded using standard IPv6 routing toward the next segment [7], [20].

One of the key innovations of SRv6 is its network programming model. In this model, segments represent both physical locations and specific behaviors. These behaviors can range from simple forwarding actions, such as the End function, to more advanced operations like encapsulation, decapsulation, and service insertion [20]. This design allows operators to define service chains and traffic-engineering policies directly within packet headers.

Compared to traditional MPLS architectures, SRv6 offers several important advantages. The use of IPv6 addressing enables hierarchical aggregation of segment identifiers, which improves routing scalability and makes policy enforcement easier through standard IPv6 filtering. In addition, eliminating label-distribution protocols reduces control-plane complexity and supports centralized traffic engineering through software-defined controllers [5], [18].

Overall, SRv6 brings together traffic engineering, service delivery, and network programmability within a single IPv6-based forwarding architecture, providing a flexible and modern foundation for service-provider networks.

3.2. Comparative Analysis: SRv6 versus Traditional MPLS

The architectural differences between MPLS and SRv6 have a strong impact on control-plane complexity, traffic-engineering flexibility, and overall network operations. While both technologies are designed to support scalable transport and service delivery, they take very different approaches to how paths are controlled and how packets are forwarded across the network.

Traditional MPLS networks rely on a combination of control-plane protocols working together. LDP is used to distribute labels for hop-by-hop forwarding, RSVP-TE is used to build traffic-engineered tunnels with resource reservation, and BGP extensions are used to distribute VPN routing information and service labels for multi-tenant environments [3], [4]. Although this multi-protocol approach provides powerful traffic-engineering capabilities, it also increases operational complexity because routers in the network must maintain distributed state and participate in multiple signaling processes.

Segment Routing simplifies this design by shifting path control to the edge of the network. In SRv6 environments, ingress routers encode forwarding instructions as segment lists within IPv6 packet headers. Transit routers then forward packets based on the active segment in the IPv6 destination address, without needing to maintain per-path state [5], [7]. This removes the need for LDP and reduces reliance on RSVP-TE for traffic engineering.

In MPLS, traffic engineering is typically achieved through RSVP-TE tunnels that maintain explicit state across every router along the path. This provides precise and deterministic control, but it also requires each router to store tunnel information and participate in signaling [8]. In contrast, SRv6 uses segment lists that are computed at the ingress router

or by a centralized controller. Since transit routers remain stateless, SRv6 can support a large number of engineered flows without increasing control-plane complexity.

Service delivery also differs between the two approaches. In MPLS Layer 3 VPNs, service delivery is based on stacked labels, where an outer label defines the transport path and an inner label identifies the VPN context at the egress router [3]. SRv6 replaces this model by encoding service behaviors directly into segment identifiers. For example, a segment associated with a VPN function at the egress node can trigger decapsulation and forward the packet into the correct routing context [19]. This approach combines service identification and forwarding logic within the IPv6 addressing structure, making service provisioning more straightforward and efficient.

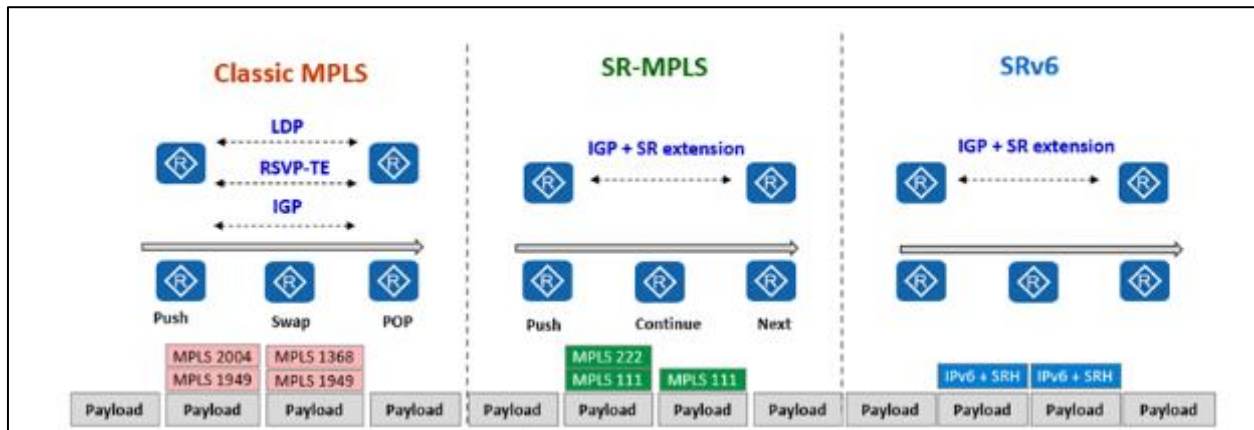


Figure 2 Architectural Comparison of MPLS and SRv6 Transport Networks

Packet overhead is an important difference between MPLS and SRv6. In MPLS, each label adds only four bytes to the packet, which keeps overhead relatively low. In contrast, SRv6 adds a full IPv6 header along with a Segment Routing Header, and each segment is represented as a 128-bit IPv6 address. This means that SRv6 packets can have higher overhead, especially when multiple segments are used in the path [20]. However, techniques such as binding segments and segment-list compression can help reduce this overhead in real-world deployments.

Operational visibility also differs between the two approaches. In MPLS networks, troubleshooting often requires checking multiple protocol states across different routers, including LDP bindings, RSVP-TE tunnels, and BGP service labels. This can make fault isolation more complex. With SRv6, the process is simpler because the intended path is directly encoded in the packet header. By examining the segment list, operators can clearly understand how the packet is being forwarded without needing to compare information from multiple protocol databases.

Overall, SRv6 offers clear advantages in terms of control-plane simplicity, network programmability, and integration with software-defined networking. At the same time, these benefits come with practical considerations, such as increased packet overhead and the need for compatible hardware support. Choosing between MPLS and SRv6 ultimately depends on the operator's priorities, network requirements, and existing infrastructure.

3.3. Migration Strategies and Coexistence Mechanisms

Moving from existing MPLS networks to SRv6 architectures requires careful planning to avoid service disruption. Service-provider networks carry large-scale, live traffic, so maintaining service continuity and minimizing operational risk is critical. For this reason, migration cannot be done all at once. Instead, it must happen gradually, using deployment models that allow MPLS and SRv6 to operate together during the transition period.

One approach is parallel infrastructure deployment. In this model, SRv6 capabilities are introduced alongside the existing MPLS network. Some parts of the network run SRv6, while others continue using MPLS. Over time, services are moved to SRv6 as equipment is upgraded and the network becomes ready. This method reduces risk because changes are controlled and gradual, but it may require additional investment and can increase operational complexity while both systems run at the same time.

A more widely used approach is incremental migration. Here, SRv6 is introduced step by step across different parts of the network. Providers often start with metro networks or newly built segments, while keeping MPLS in the core. At the

boundaries between these domains, interworking mechanisms are used to ensure that traffic can move smoothly between MPLS and SRv6 environments without interruption [14]. These mechanisms rely on gateways that translate between MPLS label stacks and SRv6 segment identifiers.

Interworking gateways play a key role in this process. When traffic moves from an MPLS network into an SRv6 domain, the gateway removes the MPLS labels and adds an IPv6 header with a Segment Routing Header that defines the new path. When traffic moves in the opposite direction, from SRv6 to MPLS, the gateway applies the appropriate MPLS labels. This translation allows different parts of the network to evolve independently while still maintaining end-to-end connectivity.

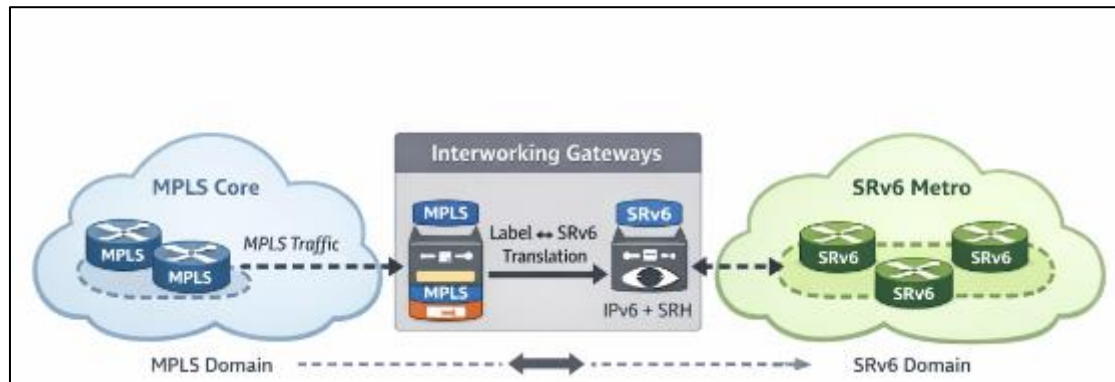


Figure 3 Incremental Migration Strategy from MPLS to SRv6

Service migration also requires careful coordination at the control-plane level. In hybrid networks where MPLS and SRv6 operate together, BGP can be used to advertise both MPLS labels and SRv6 segment identifiers for VPN services. This allows edge routers to choose the appropriate forwarding method depending on the network domain the packet will traverse [3], [19]. By supporting both signaling methods at the same time, operators can transition services gradually without needing to upgrade all provider-edge routers at once.

Operational planning is equally important during this process. Service providers need to evaluate whether their hardware can support SRv6, update management systems, and adjust operational procedures to handle the new architecture. In many cases, migration is aligned with existing equipment refresh cycles, which helps reduce costs and minimizes service disruption [21].

By following a phased migration approach and using interworking mechanisms, service providers can adopt SRv6 step by step while continuing to support existing MPLS services. This allows them to benefit from the advantages of SRv6 while keeping risks under control and maintaining reliable network operations.

3.4. Operational Considerations and Implementation Best Practices

Deploying SRv6 successfully requires careful attention to several operational factors, including hardware capability, monitoring tools, security policies, and automation frameworks. While SRv6 simplifies the control plane compared to MPLS, it introduces new requirements that must be considered during planning and implementation.

One of the most important factors is hardware support. SRv6 forwarding relies on processing IPv6 extension headers and handling the Segment Routing Header at line rate. Modern network processors and ASICs are designed to support these functions, including segment-list processing and endpoint behaviors. However, older hardware, especially systems built before SRv6 was introduced, may not support these features. In such cases, operators may need to upgrade equipment or limit SRv6 deployment to newer parts of the network [23].

Monitoring and troubleshooting are also key considerations. In MPLS networks, operators often need to reconstruct forwarding paths by analyzing multiple protocol states across different routers. In contrast, SRv6 encodes the intended path directly in the packet header. By inspecting the segment list in the SRH, operators can clearly see how a packet is expected to move through the network. Tools such as SRv6-aware traceroute, flow telemetry, and packet capture provide better visibility and make fault isolation easier compared to traditional MPLS environments [20].

Security must also be carefully managed. Since SRv6 carries forwarding instructions within IPv6 headers, networks must ensure that only trusted sources can send SRv6 traffic. Service providers typically implement strict ingress filtering at network boundaries to block or sanitize unauthorized SRv6 packets. This ensures that only trusted routers or authorized controllers can influence forwarding behavior and helps prevent potential misuse [24].

Another major advantage of SRv6 is its support for automation and orchestration. Because forwarding paths are defined using segment lists, they integrate well with software-defined networking and intent-based systems. Automation platforms can calculate optimal paths based on real-time network conditions and program ingress routers accordingly. This enables more flexible traffic engineering and supports closed-loop optimization using live telemetry data [22].

Capacity planning is also important when deploying SRv6. Each segment identifier is a 128-bit IPv6 address, which increases packet header size compared to MPLS label stacks. While modern high-capacity networks can typically handle this overhead, operators still need to ensure that link capacity and MTU settings are properly configured. Techniques such as segment-list compression and binding segments can help reduce the overhead in practical deployments [23].

By addressing these operational factors during the planning stage, service providers can successfully integrate SRv6 into their existing networks while maintaining reliability, performance, and operational efficiency.

3.5. Industry Adoption and Real-World Deployments

The increasing interest in SRv6 can be seen in real-world deployments and trials carried out by major telecommunications operators and equipment vendors. These efforts show that SRv6 is not just a theoretical concept but a practical transport technology that can meet modern network demands such as programmable traffic engineering, cloud connectivity, and 5G integration.

Many service providers have tested SRv6 as part of their network modernization efforts, with the goal of simplifying existing transport architectures. By removing the need for label-distribution protocols and enabling source-based traffic engineering, SRv6 helps reduce control-plane complexity and gives operators greater flexibility in how they select paths and deliver services. Early deployments have mainly focused on backbone and metro networks, where dynamic traffic steering can improve resource utilization and overall service performance [25].

Mobile network operators have also explored the use of SRv6 in 5G transport networks. SRv6 supports service-function chaining and flexible routing policies, which are important for handling mobile-core traffic. These capabilities make it possible to dynamically insert services and efficiently steer traffic between distributed network functions. This is especially valuable in 5G environments, where technologies like network slicing and edge computing require highly programmable and adaptable transport networks [26].

Beyond telecom networks, SRv6 is also gaining attention in environments that require programmable networking for large-scale distributed applications. Its network programming model allows operators to define complex forwarding behaviors directly within the network infrastructure. This reduces the need for external service-chaining solutions and simplifies overall network orchestration.

Despite this growing adoption, the success of SRv6 deployment still depends on factors such as hardware readiness and operational maturity. Most providers introduce SRv6 gradually, starting with specific network segments or service types. This step-by-step approach allows them to test processes, validate monitoring tools, and refine automation workflows while minimizing the risks associated with large-scale changes.

Overall, industry experience shows that SRv6 is emerging as a strong alternative to MPLS, particularly in environments where programmability, automation, and flexible traffic engineering are critical requirements.

3.6. Challenges and Limitations of SRv6 Deployment

Although Segment Routing over IPv6 (SRv6) offers clear architectural benefits, there are still important technical and operational challenges that must be addressed before it can be deployed widely in large service-provider networks. These limitations are especially important for operators that are planning to move from long-established MPLS environments.

One of the main concerns is packet-header overhead. In SRv6, each segment identifier is a 128-bit IPv6 address carried in the Segment Routing Header. This means that packets with several segments can become much larger than MPLS packets, where each label adds only four bytes. While techniques such as segment-list compression and binding

segments can help reduce this overhead, packet size still remains a key consideration in high-throughput networks [20], [27].

Hardware support is another major factor. For SRv6 to work efficiently, routers must be able to process IPv6 extension headers and perform SRH operations at line rate. Many newer platforms include ASICs that are built to support SRv6, but older hardware often does not. Because of this, operators may need to introduce SRv6 during hardware refresh cycles or limit deployment to parts of the network that already have compatible equipment [23], [27].

Migration also brings its own operational challenges. In hybrid MPLS-SRv6 networks, interworking gateways are needed to translate between MPLS label stacks and SRv6 segment identifiers. These mixed environments add extra configuration, coordination, and monitoring requirements until the migration is fully completed [14].

Security is another important issue. Since forwarding instructions are carried directly in IPv6 headers, poorly filtered SRv6 packets from untrusted sources could affect forwarding behavior. To prevent this, operators apply strict ingress filtering and policy controls so that SRv6 traffic is accepted only from trusted network devices [28].

Finally, operational tools and the overall SRv6 ecosystem are still developing. Many monitoring platforms already support SRv6 telemetry and diagnostics, but experience with large-scale production deployments is still growing. As adoption continues, operators will gain better guidance and stronger best practices for monitoring, troubleshooting, and managing SRv6 networks.

4. Discussion

The results of this study show that SRv6 represents a meaningful shift in how service-provider networks can be designed and operated. Compared to traditional MPLS, SRv6 simplifies the control plane by removing the need for multiple signaling protocols while still maintaining strong traffic-engineering capabilities. This shift is important because modern networks are becoming more complex, and operators are looking for solutions that support automation, programmability, and easier management.

One key observation is that SRv6 changes where intelligence sits in the network. Instead of distributing control across many routers, SRv6 moves decision-making to the network edge or to centralized controllers. This makes the network easier to scale and aligns well with software-defined networking approaches. At the same time, it allows operators to define traffic paths and service behaviors directly in packet headers, which improves visibility and simplifies troubleshooting.

However, the study also highlights that SRv6 is not a direct replacement for MPLS in all situations. While it offers advantages in flexibility and programmability, it introduces new considerations such as higher packet overhead and the need for hardware that can support IPv6 extension-header processing at high speed. These factors mean that adoption decisions must be based on each operator's specific network conditions and performance requirements.

Another important finding is that migration is a gradual and carefully managed process. Most service providers are unlikely to replace MPLS entirely in a short period. Instead, they will adopt SRv6 in phases, using hybrid environments where MPLS and SRv6 coexist. This approach reduces risk and allows operators to validate performance, tools, and operational procedures before full deployment.

The study also shows that SRv6 fits well with emerging technologies such as 5G, cloud computing, and edge services. Its ability to support service-function chaining and programmable traffic steering makes it suitable for environments that require dynamic and flexible network behavior. This positions SRv6 as a strong candidate for next-generation transport networks.

Overall, the discussion suggests that SRv6 is not just an incremental improvement but a shift toward a more programmable and automated networking model. While challenges remain, especially around hardware readiness and operational maturity, SRv6 provides a solid foundation for building future network infrastructures that can meet evolving service demands.

5. Conclusion

Multiprotocol Label Switching (MPLS) has served as the backbone of service-provider transport networks for many years, providing reliable service delivery and strong traffic-engineering capabilities. However, as networks evolve, the increasing need for programmability, automation, and native IPv6 support has exposed limitations in MPLS, particularly due to its reliance on multiple interacting control-plane protocols.

Segment Routing over IPv6 (SRv6) offers a different approach by embedding forwarding instructions directly into IPv6 packet headers. This study evaluated SRv6 as a natural evolution of MPLS by examining its architecture, simplified control plane, and network programming capabilities. The comparison between MPLS and SRv6 highlighted key differences in how paths are controlled, how services are delivered, and how network operations are managed.

The paper also explored practical migration strategies that enable service providers to transition gradually from MPLS to SRv6 without disrupting existing services. In addition, it reviewed important operational considerations, real-world deployment experiences, and the challenges that still need to be addressed.

Overall, SRv6 represents a significant step toward more programmable and automation-ready transport networks, offering a flexible and scalable foundation for the next generation of service-provider infrastructures.

5.1. Future Research Directions

Although SRv6 has emerged as a strong evolution of MPLS-based transport architectures, there are still important areas that need further research and development to improve its scalability, performance, and operational maturity. Continued progress in these areas will help service providers fully take advantage of SRv6, especially in large-scale production networks.

One major area of focus is header compression. In SRv6, each segment identifier is a 128-bit IPv6 address, which means packets with many segments can become quite large. This creates more overhead compared to MPLS, where each label adds only four bytes. New approaches, such as compressed SRv6 segment identifiers, are being developed to reduce this overhead while still keeping the flexibility of SRv6's network programming model [29]. However, more work is needed to understand how well these solutions scale and how they perform across different network environments.

Another important area is hardware performance. For SRv6 to work efficiently, routers must be able to process IPv6 extension headers and execute segment-based functions at high speed. Ongoing research is focused on improving forwarding performance and designing specialized hardware, such as advanced ASICs, that can support SRv6 operations in high-capacity networks [30].

Integration with software-defined networking and intent-based frameworks is also a key direction. Since SRv6 encodes forwarding paths directly in packet headers, centralized controllers can dynamically compute optimal segment lists based on real-time network conditions. This makes it possible to build automated systems that continuously adjust network behavior using live telemetry data.

At the same time, large-scale operational validation is still needed. While early deployments have shown that SRv6 can work in practice, more studies are required to evaluate its performance and reliability in different real-world environments. Long-term operational data from service-provider networks will help improve best practices for deployment, monitoring, and troubleshooting.

As these areas continue to improve, they will help address current limitations and strengthen SRv6 as a scalable and programmable transport architecture for next-generation service-provider networks.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

- [1] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, IETF, Jan. 2001. Available: <https://www.rfc-editor.org/rfc/rfc3031>
- [2] B. Davie and Y. Rekhter, *MPLS: Technology and Applications*. San Francisco, CA, USA: Morgan Kaufmann, 2000. Available: <https://dl.acm.org/doi/abs/10.5555/374224>
- [3] E. Rosen and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)," RFC 4364, IETF, Feb. 2006. Available: <https://www.rfc-editor.org/rfc/rfc4364>
- [4] D. Awduche et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels," RFC 3209, IETF, Dec. 2001. Available: <https://www.rfc-editor.org/rfc/rfc3209>
- [5] C. Filsfil et al., "Segment Routing Architecture," RFC 8402, IETF, Jul. 2018. Available: <https://www.rfc-editor.org/rfc/rfc8402>
- [6] S. Previdi et al., "Segment Routing with MPLS Data Plane," RFC 8660, IETF, Dec. 2019. Available: <https://www.rfc-editor.org/rfc/rfc8660>
- [7] C. Filsfil et al., "SRv6 Network Programming," RFC 8986, IETF, Feb. 2021. Available: <https://www.rfc-editor.org/rfc/rfc8986>
- [8] D. O. Awduche, "MPLS and traffic engineering in IP networks," *IEEE Communications Magazine*, vol. 37, no. 12, pp. 42–47, 1999, doi: <https://doi.org/10.1109/35.809383>.
- [9] J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network recovery: protection and restoration of optical, SONET-SDH, IP, and MPLS*. San Francisco, Calif.: Morgan Kaufmann, 2004.
- [10] Y. Wang, X. Zhang, L. Fan, S. Yu, and R. Lin, "Segment Routing Optimization for VNF Chaining," *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–7, May 2019, doi: <https://doi.org/10.1109/icc.2019.8761103>.
- [11] Y. Tian et al., "Traffic Engineering in Partially Deployed Segment Routing Over IPv6 Network With Deep Reinforcement Learning," vol. 28, no. 4, pp. 1573–1586, Aug. 2020, doi: <https://doi.org/10.1109/tnet.2020.2987866>.
- [12] Yves Vanaubel, Pascal Mérindol, Jean-Jacques Pansiot, and Benoit Donnet, "MPLS Under the Microscope," *ORBi (University of Liège)*, Oct. 2015, doi: <https://doi.org/10.1145/2815675.2815687>.
- [13] P. L. Ventre, M. M. Tajiki, S. Salsano, and C. Filsfil, "SDN Architecture and Southbound APIs for IPv6 Segment Routing Enabled Wide Area Networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1378–1392, Dec. 2018, doi: <https://doi.org/10.1109/tnsm.2018.2876251>.
- [14] G. D. Salazar, E. F. Naranjo, and L. Marrone, "SDN-Ready WAN networks: Segment Routing in MPLS-Based Environments," Nov. 2018, doi: <https://doi.org/10.1109/uemcon.2018.8796613>.
- [15] S. Vissicchio, L. Vanbever, and O. Bonaventure, "Opportunities and research challenges of hybrid software defined networks," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 2, pp. 70–75, Apr. 2014, doi: <https://doi.org/10.1145/2602204.2602216>.
- [16] G. Ajiardiawan and Y. Suryanto, "Performance Analysis of Segment Routing on MPLS L3VPN using PNETLAB," *2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST)*, pp. 185–190, Jun. 2021, doi: <https://doi.org/10.1109/icaicst53116.2021.9497844>.
- [17] Takuya Miyasaka, Y. Hei, and T. Kitahara, "NetworkAPI," Aug. 2020, doi: <https://doi.org/10.1145/3405672.3405805>.
- [18] Z. N. Abdullah, I. Ahmad, and I. Hussain, "Segment Routing in Software Defined Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 464–486, 2019, doi: <https://doi.org/10.1109/comst.2018.2869754>.
- [19] S. Previdi, D. Voyer, S. Matsushima, C. Filsfil, J. Leddy, and D. Dukes, "IPv6 Segment Routing Header (SRH)," Mar. 2020, doi: <https://doi.org/10.17487/rfc8754>.
- [20] P. Loreti et al., "SRv6-PM: A Cloud-Native Architecture for Performance Monitoring of SRv6 Networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 611–626, Mar. 2021, doi: <https://doi.org/10.1109/tnsm.2021.3052603>.

- [21] Ryusei Shiiba, S. Kobayashi, O. Akashi, and K. Fukuda, "Verifying Network Properties in SRv6 based Service Function Chaining," Asian Internet Engineering Conference, pp. 18–26, Dec. 2021, doi: <https://doi.org/10.1145/3497777.3498546>.
- [22] A. Abdelsalam et al., "SRPerf: A Performance Evaluation Framework for IPv6 Segment Routing," IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 2320–2333, Jun. 2021, doi: <https://doi.org/10.1109/tnsm.2020.3048328>.
- [23] F. Duchene, D. Lebrun, and O. Bonaventure, "SRv6Pipes: Enabling in-network bytestream functions," Computer Communications, vol. 145, pp. 223–233, Sep. 2019, doi: <https://doi.org/10.1016/j.comcom.2019.05.009>.
- [24] F. Duchene, M. Jadin, and O. Bonaventure, "Exploring various use cases for IPv6 Segment Routing," pp. 129–131, Aug. 2018, doi: <https://doi.org/10.1145/3234200.3234213>.
- [25] X. Li, B. Liu, X. Zheng, H. Duan, Q. Li, and Y. Huang, "Fast IPv6 Network Periphery Discovery and Security Implications," Jun. 2021, doi: <https://doi.org/10.1109/dsn48987.2021.00025>.
- [26] Z. Li, D. Zheng, and Y. Ma, "Tree, Segment Table, and Route Bucket: A Multi-Stage Algorithm for IPv6 Routing Table Lookup," IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications, pp. 2426–2430, 2007, doi: <https://doi.org/10.1109/infcom.2007.294>.
- [27] P. L. Ventre et al., "Segment Routing: a Comprehensive Survey of Research Activities, Standardization Efforts and Implementation Results," IEEE Communications Surveys & Tutorials, pp. 1–1, 2020, doi: <https://doi.org/10.1109/comst.2020.3036826>.
- [28] J. Zhou, H. Li, Q. Wu, Z. Lai and J. Liu, "SR-TPP: Extending IPv6 Segment Routing to enable Trusted and Private Network Paths," 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 2020, pp. 1–6, doi: [10.1109/ISCC50000.2020.9219705](https://doi.org/10.1109/ISCC50000.2020.9219705).
- [29] G. D. Salazar and A. R. Reinoso, "Segment-Routing Analysis: Proof-of-Concept Emulation in IPv4 and IPv6 Service Provider Infrastructures," 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Apr. 2021, doi: <https://doi.org/10.1109/iemtronics52119.2021.9422559>.
- [30] Bidkar, Sarvesh, et al. "Scalable segment routing—A new paradigm for efficient service provider networking using carrier ethernet advances." Journal of Optical Communications and Networking 7.5 (2015): 445-460. <https://opg.optica.org/jocn/abstract.cfm?uri=jocn-7-5-445>