TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

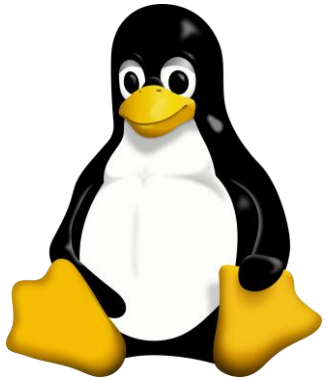
cnit consorzio nazionale interuniversitario per le telecomunicazioni

eBPF and SRv6

Stefano Salsano

University of Rome Tor Vergata / CNIT

Netdev 0x19 - Zagreb, Croatia
March, 11th 2025



why eBPF for SRv6 ?

- Provides high-performance, in-kernel programmability to process SRv6 packets efficiently.
- Allows dynamic policy enforcement and network function injection without modifying kernel code.
- Reduces reliance on vendor-specific hardware by enabling software-based processing.

thanks to ChatGPT

Projects Leveraging eBPF for SRv6

- **Linux Kernel SRv6 eBPF Hooks:** Ongoing efforts to extend eBPF support for SRv6 processing in the Linux networking stack.
- **IOVisor and Cilium:** Open-source projects exploring SRv6 integration with eBPF for advanced routing and network security.
- **FD.io VPP (Vector Packet Processing):** Implementing SRv6 processing using eBPF to improve performance and flexibility.
- **University and Industry Research Projects:** Several academic and industry-led initiatives are studying eBPF-based SRv6 optimization.

thanks to ChatGPT

1. Integration of eBPF with SRv6 in the Linux Kernel: Since Linux kernel version 4.18, there has been support for implementing custom SRv6 network functions using eBPF. This integration allows the execution of specific eBPF code upon receiving SRv6 packets containing local segments, enabling dynamic and programmable packet processing. segment-routing.org arxiv.org

thanks to
ChatGPT

IOVisor Project Overview

- **Mission:**
  - To advance networking and security by leveraging eBPF for dynamic kernel-level programmability.
- **Key Contributions:**
  - Development of tools and libraries facilitating the creation and deployment of eBPF programs.
  - Promotion of an ecosystem for sharing eBPF-based applications and innovations.
- **SRv6 Integration:**
  - Enables programmable network functions by executing eBPF code within the SRv6 framework.

dl.acm.org +1

thanks to ChatGPT

Cilium project overview

- **Mission:**
  - To provide networking, security, and observability for cloud-native environments using eBPF. cilium.io +3
- **Key Features:**
  - High-performance networking for Kubernetes clusters. cilium.io +5
  - Advanced security policies with identity-based enforcement. isovalent.com +2
  - Deep observability into network traffic and application behavior.
- **SRv6 Integration:**
  - Utilizes eBPF to implement SRv6 functionalities, enhancing routing efficiency and flexibility.

thanks to ChatGPT

Cilium Open Components

- **Networking:**
  - eBPF-based datapath for efficient packet processing. isovalent.com
  - Kubernetes CNI plugin for seamless integration. isovalent.com +1
- **Security:**
  - Network policies enforcing Layer 3 to Layer 7 security controls. Wikipedia +1
  - Transparent encryption using IPSec or WireGuard. Wikipedia
- **Observability:**
  - Hubble: Real-time monitoring and visibility into network flows. Wikipedia +5

Sources:

- Cilium Official Website
- Cilium GitHub Repository

thanks to ChatGPT

Isovalent Enterprise for Cilium (propretary)

- **Isovalent Enterprise for Cilium:**
  - Advanced networking capabilities, including high-performance load balancing and multi-cluster connectivity. isovalent.com +1
  - Enhanced security features like zero-trust network policies. isovalent.com
  - Comprehensive observability tools for deep network insights.

Sources:

- Isovalent Enterprise for Cilium
- Isovalent Extends Cilium's Enterprise Networking and Security Features

thanks to ChatGPT

eBPF for SRv6 in FD.io VPP (?)

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

Overview:

- **FD.io VPP (Vector Packet Processing):** An open-source, high-performance packet processing framework operating in user space, designed for flexible and efficient networking solutions.

Current Integration Status:

- As of now, **FD.io VPP does not natively support the execution of eBPF programs within its user-space architecture**. Discussions about integrating eBPF into VPP have occurred in the past, but there are no current implementations or plans to support running eBPF or P4 in VPP. lists.fd.io

thanks to ChatGPT

Challenges in SRv6 and eBPF Integration

- **Fragmented Development Efforts:**
  - Many projects implement custom eBPF programs for SRv6 processing, often duplicating efforts.
- **Lack of Standardized Libraries:**
  - No unified open-source toolkit for performing **common SRv6 operations** using eBPF.
- **Barriers to Adoption:**
  - Developers need to write **low-level eBPF programs** from scratch for each SRv6 use case.

thanks to ChatGPT

The Need for Open-Source Solutions

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

✅ **Reusable eBPF Libraries:**

- Provide **pre-built, optimized functions** for common SRv6 tasks.
- Example: Parsing SRH headers, segment processing, traffic engineering functions.

✅ **Standard APIs & SDKs:**

- Unified interfaces to interact with SRv6 functionalities via eBPF.

✅ **Community Collaboration:**

- Open-source projects should work together to avoid **re-inventing the wheel** every time.

thanks to ChatGPT

The Need for Open-Source Solutions

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

Call to Action

- Encourage collaboration among open-source communities (FD.io, IOVisor, Cilium, Linux SRv6)
- Develop a shared eBPF-based SRv6 processing library
- Push for upstream kernel support for better eBPF-SRv6 integration

🚀 The future of programmable SRv6 depends on shared, open, and reusable eBPF tools! 🚀

thanks to ChatGPT

(asked ChatGPT to give more emphasis…)

⚡🚨 Call to Action: Let's Build This Together! 🚨⚡

💪 Open-source developers, researchers, and network engineers – we need YOU! 💡💻

🌍 Let's create a shared eBPF-based SRv6 processing library & push for upstream kernel support! 🛠️

📡

📢 Join the discussion, contribute code, and make SRv6+eBPF truly programmable & scalable! 💬👨‍💻

👨‍💻

🔥 The Future of Programmable Networking Depends on Us! 🔥

🚀 Let's make SRv6 + eBPF Open, Standardized, and Powerful – Together! 🚀

💬 Who's in? Drop your thoughts & ideas! Let's make this happen! 💡💬🎯

thanks to ChatGPT

Thank you for your attention!

stefano.salsano@uniroma2.it

Today's program

09:00 - 09:15	Stefano Salsano Introduction to the workshop
09:15 - 09:35	Ahmed Abdelsalam Technical intro to SRv6, IETF status, interoperability status
09:35 - 10:05	Andrea Mayer SRv6 in Linux kernel: past, present and future
10:05 - 10:20	Ahmed Abdelsalam SoNIC and SRv6
10:20 - 10:50	Carmine Scarpitta FRR: status and evolution of SRv6 support
10:50 - 11:10	COFFEE BREAK
11:10 - 11:25	Stefano Salsano eBPF and SRv6
11:25 - 11:40	Angelo Tulumello eBPF and SRv6: a use case for RoCEv2 support
11:40 - 12:00	Emilien Wansart Mitigating the Double-Reallocation Issue for IPv6 Lightweight Tunnel Encapsulations
12:00 - 12:30	Stefano Salsano (Moderator) Panel discussion on next steps for SRv6 in Linux networking