SC26 Network Research Exhibition: Demonstration Publishable Abstract

International Demonstration of 400 Gbps 8K Uncompressed Video Processing using SRv6-based In-Network Computing

Katsuhiro SEBAYASHI, Kanagawa Institute of Technology,
sebayashi@nw.kanagawa-it.ac.jp

Abstract

This demonstration presents an international 8K uncompressed video processing workflow over a 400 Gbps network, enabled by SRv6-based in-network computing. We use SRv6 End.AN to integrate packet forwarding with computational processing in the network data plane. By processing media along the forwarding path, the approach aims to avoid the traffic hairpinning associated with conventional NFV-based service chaining. High-speed forwarding and SRv6 processing are supported by our open-source, DPDK-based SRv6 software router.

At SC26, we demonstrate this architecture across a network connecting Japan and the United States. Multiple video processing functions (VVFs) for 8K uncompressed streams are dynamically chained on demand to create end-to-end media workflows. A custom-developed monitoring system capable of handling 400 Gbps traffic visualizes transmission conditions and resource utilization in real time. The demonstration explores how tightly integrated networking and computing resources can support next-generation, ultra-high-bandwidth media applications.

Introduction

In the Beyond 5G (B5G) era, more people are expected to create and share ultra-high-quality media in real time. Low-latency networks can enhance the immediacy and immersion of live events and enable new forms of digital culture.

To support these applications, we have developed real-time ultra-high-definition video processing technologies using edge and cloud computing resources, with a particular focus on 8K uncompressed video.

Such streams require approximately 24–48 Gbps of bandwidth. To process them in real time using software alone, we developed the Virtualized Video Handling Function (VVF) framework based on the Data Plane Development Kit (DPDK).

Building on these technologies, we aim to enable stable, low-latency sharing of continuous, high-bandwidth media streams across heterogeneous networks. Our proposed high-speed inline computing architecture supports (i) coordinated network functions through close integration of networking and computing resources, (ii) low-latency processing through time-synchronized cooperation between those resources, and (iii) sharing of network state information with applications and end-user devices.

To realize this architecture, we use SRv6 End.AN, a behavior that has not yet been standardized by the IETF, together with a control plane designed for dynamic resource allocation. End.AN integrates forwarding and computing within network nodes. Our platform uses DPDK across routing and processing components to support high-throughput, low-latency operation. Through application-level demonstrations on wide-area testbeds, we aim to assess its suitability for practical deployment.

Inline Computing Platform and SRv6 End.AN

As illustrated in Fig. 1, the proposed inline computing model places processing functions within intermediate routers and applies them selectively to designated traffic.

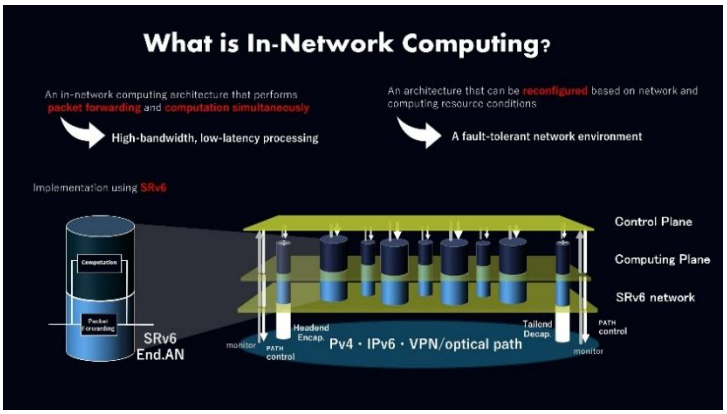


Fig. 1 Inline Computing Platform

To coordinate these functions, we adopt Service Function Chaining (SFC), which steers selected traffic through an ordered set of functions to provide services such as security and media processing. Integrating forwarding and computation within the network can support high-bandwidth, low-latency applications, particularly immersive media workflows.

For example, live video production can use SFC to interconnect geographically distributed equipment and computing resources. SFC is particularly suitable for video workflows in

which processing functions must be executed in a specified order.

We adapt SRv6 because it can operate within existing IPv6 networks. Conventional SFC deployments often use Service Function Forwarders (SFFs) and SFC proxies to steer traffic to processing functions. Depending on their topology, these deployments can introduce detours and complicated service reconfiguration.

Our approach instead uses SRv6 End.AN to implement inline service chaining, with processing functions integrated into intermediate routers. In the data plane, End.AN combines SRv6 forwarding with function execution. An orchestration mechanism collects network and computing resource information, identifies available capacity, and dynamically allocates functions and forwarding paths.

We are working toward standardization of the SRv6 End.AN data-plane behavior and associated control mechanisms through the IETF process.

From IETF 121 (November 2024) through IETF 127 (Nov. 2026), we presented our SRv6-based SFC architecture in the IETF SPRING Working Group, including the Internet-Draft "SRv6 SFC Architecture with SR-aware Functions" [1], and collaborated with the community at IETF Hackathons to develop and validate prototype control-plane implementations [2]. We extended open-source tools, including GoBGP, Exa-BGP, PolaPCE, and Wireshark, and released the resulting implementations as open-source software.

SRv6 Software Router

To support high-performance inline computing, we designed and implemented an extensible SRv6 software router. Built from scratch using DPDK and released as open-source software [3], it combines high-speed packet processing with routing functionality and provides a platform for developers to implement and evaluate SRv6 End.AN and associated network functions.

The router supports high-performance data-plane forwarding and integration with existing routing software through reusable networking libraries. Its main components include:

(1) Implementation of a DPDK thread-based scheduling mechanism

(2) Integration of a unified execution environment, DPDK-dock, combining shell interfaces with the DPDK runtime

(3) Implementation of the Linux netlink interface for control-plane integration

The software continues to be developed and improved. It is released under the GNU General Public License (GPL) to encourage adoption and experimentation by network operators, data centers, and research institutions.

Demonstration at SC26

At SC26, we demonstrate an SRv6-based network connecting Japan and the United States with 400 Gbps connectivity. Multiple processing functions for 8K uncompressed video are dynamically chained across SRv6 End.AN nodes to form an end-

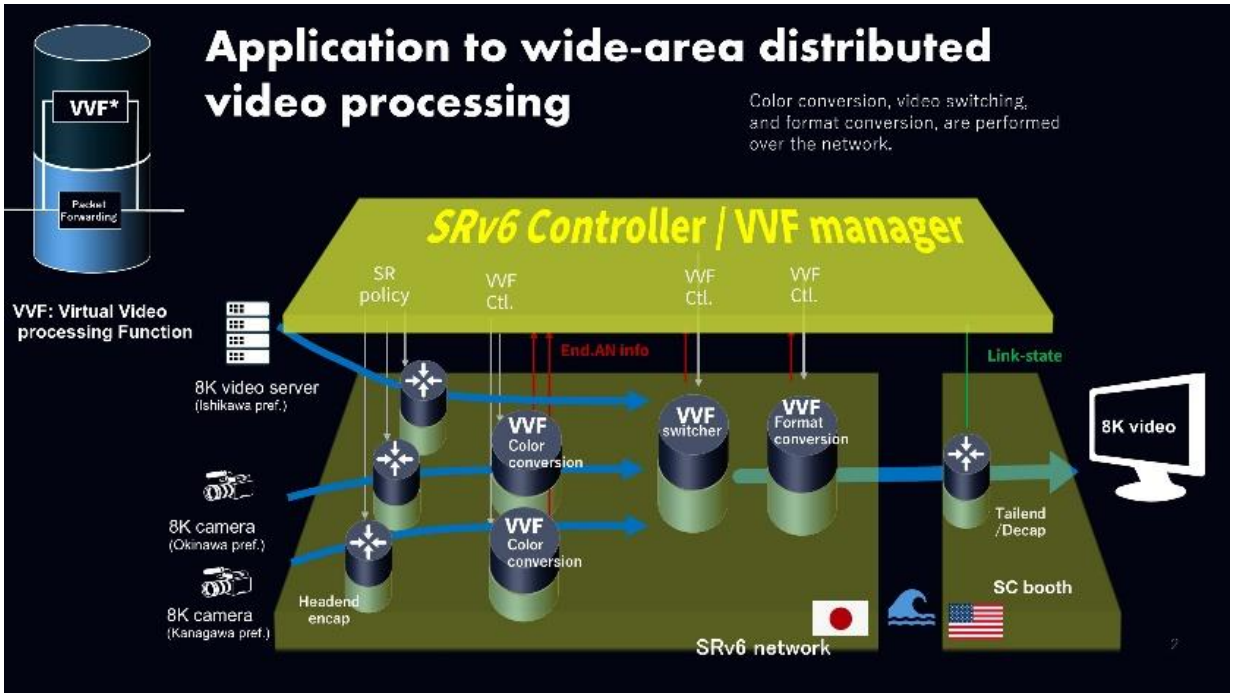


Fig. 2 Experimental Setup

to-end workflow. A custom-developed network monitoring system with 400 Gbps capability visualizes transmission conditions in real time.

Video sources include 8K camera streams captured in Ginoza, Okinawa; outdoor cameras at Kanagawa Institute of Technology connected through a local 5G network; and streams supplied by an NICT video server. SRv6 End.AN processing nodes are deployed on distributed servers in Japan, including edge systems at the SINET6 Sagamihara data center with two 400 Gbps processing interfaces.

VVFs perform video switching, format conversion, color adjustment, and caption insertion. The control plane dynamically assigns SRv6 End.AN functions to nodes with available resources and specifies SRv6 segment lists to establish the processing workflow. The resulting 8K video is displayed at the SC26 exhibition venue.

A control-plane graphical user interface (GUI) visualizes co-ordination among distributed SRv6 End.AN functions in real time. SRv6 software routers forward traffic across intermediate network segments, while the monitoring system provides visibility into packet transmission conditions at points along the end-to-end path.

Conclusion

Following this demonstration, we plan to further develop the high-performance SRv6 End.AN platform, tightly integrating computing functions and SRv6 packet forwarfing. Using high-resolution video as a demanding use case, we will continue to investigate coordinated control of networking and computing resources.

We also plan to extend experiments across wide-area testbeds, explore additional applications, and assess the practical deployment of the proposed network architecture.

Goals

1. Validate 400 Gbps International Connectivity: Establish a high-bandwidth trans-pacific path using SCinet to rigorously test our custom DPDK-based SRv6 software router at line rate.
2. Demonstrate SRv6 End.AN Efficiency: Prove the architectural superiority of our "SRv6 SFC Architecture with SR-aware Functions" by eliminating traffic hairpinning, enabling direct, low-latency in-network media processing.
3. Orchestrate Dynamic Media Workflows: Showcase real-time, on-demand service chaining of 8K uncompressed video processing functions (VVF).

Resources

- This demonstration will be held at the NII booth (Booth #1459).
- We will conduct the experiment by sharing the 400G line provided at the NII booth with other NRE teams.
- We will use four L2VPNs to connect sites (therefore, we will use four VLANs within the venue).
- These four VLANs are connected with SINET6.
  - VLAN #1: 50 Gbps, 8K uncompressed video (SRv6).
  - VLAN #2: 50 Gbps, 8K uncompressed video (SMPTE ST-2110).
  - VLAN #3: <=1 Gbps, Control & Management (IPv4).
  - VLAN #4: <=1 Gbps, Monitoring (IPv4).

Involved Parties

- Mitsuru MARUYAMA, Kanagawa Institute of Technology
- Yasuhiro OHARA, Kanagawa Institute of Technology
- Wataru MISHIMA, Kanagawa Institute of Technology
- Yuta FUKAGAWA, Kanagawa Institute of Technology
- Takeshi KURIMOTO, National Institute of Informatics
- Taisei TANABE, National Institute of Informatics
- Kenjiro YAMANAKA, National Institute of Informatic
- Atsuko TAKEFUSA, National Institute of Informatics

References

[1] Wataru Mishima, Mai Nakao, “SRv6 SFC Architecture with SR-aware Functions,” https://datatracker.ietf.org/doc/draft-watal-spring-srv6-sfc-sr-aware-functions/, IETF Hackathon, IETF 126, July 18-19, 2026.

[2] Wataru Mishima, Yuta Fukagawa, “SRv6 SFC Architecture with SR-aware Functions,” https://datatracker.ietf.org/meeting/126/materials/slides-126-hackathon-sessd-srv6-sfc-architecture-with-sr-aware-functions-04, SPRING-WG, IETF 126, July 18-19, 2026.

[3] KAIT-CRONOS, https://github.com/kait-cronos/.github/blob/main/profile/README_en.md

Acknowledgment

This work was partially supported by JST CRONOS (Grant No. JPMJCS24N9).