

EDGE-NATIVE CABLE ACCESS NETWORK WITH UDP TERMINATION

This paper introduces a software-defined architectural model for modernizing upstream transport in hybrid fiber-coaxial networks through user-space UDP termination using the VPP HostStack. The approach aims to overcome performance limitations inherent in DOCSIS-based systems by relocating transport-layer processing from centralized cores to multi-access edge computing nodes co-located with Remote PHY Devices. Unlike centralized vCMTS deployments or full-fiber upgrades, this model enables enhanced throughput and session-level concurrency without modifying coaxial infrastructure, DOCSIS PHY signaling, or customer premises equipment. The system is built on FD.io's Vector Packet Processing framework and validated through CI-integrated benchmarking using the CSIT testbed and iperf3 traffic generator. Test executions were performed on a two-node Intel Ice Lake setup interconnected via 100GE Intel E810CQ interfaces. The benchmarking workflow used iperf3 dynamically linked with libvcl_ldpreload.so, enabling transparent redirection of UDP socket operations to the HostStack without modifying the application code. Two traffic scenarios were evaluated: a baseline with one UDP stream achieving 15.5 Gbps, and a scaled case with one client transmitting ten concurrent streams reaching 28.1 Gbps. The results demonstrate an 81% throughput gain with minimal variance, confirming the HostStack's efficiency under concurrent flow conditions. Integration via LD_PRELOAD proved stable, transparent, and production-compatible. These findings validate the scalability, reproducibility, and operational viability of edge-based UDP termination for HFC modernization. The proposed model offers a low-disruption upgrade path for cable operators, enabling phased deployment of user-space transport enhancements with CI/CD compatibility, improved upstream utilization, and support for multi-session, latency-sensitive applications such as telemetry, video streaming, and cloud synchronization. In addition, this architecture enables measurable performance improvements without hardware changes, costly rewiring, or disruptions to existing customer services.

Keywords: network performance, software-based packet processing, network scalability, traffic optimization, performance evaluation.

ІВАНЕЦЬ Іван, ОВСЯК Володимир, ОВСЯК Олександр
Національний університет «Львівська політехніка»

EDGE-NATIVE ГІБРИДНА ОПТИЧНО-КОАКСІАЛЬНА МЕРЕЖА З UDP-ПРОТОКОЛОМ

У статті представлено модель архітектури гібридної оптично-коаксіальної мережі для модернізації висхідного трафіку, реалізовану на основі програмно-визначеного підходу з термінацією UDP-транспортного рівня у просторі користувача за допомогою стеку VPP HostStack. Запропонована модель вирішує проблему обмеженої пропускної здатності, притаманну системам на основі DOCSIS, шляхом перенесення обробки транспортного рівня з централізованих вузлів до хмарних обчислювальних вузлів на периферії, розміщених поблизу пристроїв віддаленого фізичного рівня. На відміну від централізованих архітектур vCMTS або повної заміни волоконно-оптичною інфраструктурою, ця модель забезпечує підвищення пропускної здатності та паралелізм сесій без змін у коаксіальній мережі, сигналі PHY-рівня чи абонентському обладнанні. Система реалізована на базі платформи обробки пакетів Vector Packet Processing з відкритим кодом FD.io і проходить валідацію через автоматизовану CI/CD-інфраструктуру тестування CSIT із використанням генератора трафіку iperf3. Експерименти проводилися на двох фізичних вузлах із процесорами Intel Ice Lake, з'єднаних 100GE-інтерфейсами Intel E810CQ. Для прозорості інтеграції використовувався iperf3, динамічно підключений до бібліотеки libvcl_ldpreload.so, що забезпечує перенаправлення системних викликів сокетів UDP до стеку HostStack без змін у прикладному коді. Було протестовано два сценарії: базовий — з одним потоком UDP (15,5 Гбіт/с) та масштабований — з одним клієнтом і десятьма паралельними потоками (28,1 Гбіт/с). Результати демонструють зростання пропускної здатності на 81 % при мінімальному коливанні, що підтверджує ефективність стеку HostStack в умовах паралельного навантаження. Інтеграція через LD_PRELOAD виявилася стабільною, прозорою та сумісною з виробничими середовищами. Отримані результати засвідчують масштабованість, відтворюваність і практичну доцільність термінації UDP-трафіку на мережній периферії в межах модернізації HFC-мереж. Запропонована модель забезпечує поступову модернізацію без переривання обслуговування клієнтів, дорогавартісних апаратних змін чи заміни інфраструктури, а також підтримує багатосесійні й чутливі до затримок прикладні програми, зокрема телеметрію, відеопотоки та хмарну синхронізацію.

Ключові слова: продуктивність мережі, програмна обробка пакетів, масштабованість мережі, оптимізація трафіку, тестування продуктивності.

Introduction

The exponential growth of upstream-demanding services – such as interactive video conferencing, cloud-based collaboration, and telemedicine – has placed increasing pressure on traditional broadband architectures. Among these, hybrid fiber-coaxial (HFC) networks, widely deployed across North America and Europe, face growing limitations due to their inherent asymmetry and centralized transport-layer processing [1]. Although standards like DOCSIS 3.1/4.0 and architectural extensions such as Remote PHY Devices (RPD) and Distributed Access Architectures (DAA) aim to extend capacity and fiber reach, they fall short of addressing key transport-layer bottlenecks in asymmetric environments [2]. This challenge has been addressed in mobile networking through Multi-access Edge Computing (MEC) and software-defined networking (SDN), which have enabled low-latency

services via functional relocation to the access edge [3]. However, similar approaches in fixed broadband – particularly HFC – remain insufficiently explored.

A growing body of research highlights the benefits of user-space transport termination as a scalable and flexible solution for modernizing access networks. By bypassing kernel overhead and enabling direct control over packet processing, user-space stacks offer lower latency, better concurrency handling, and easier integration with SDN and CI/CD workflows. In the context of HFC networks, which are inherently asymmetric and upstream-constrained, these techniques hold particular promise for improving performance without requiring disruptive physical-layer upgrades.

This study proposes a lightweight architectural model for UDP transport termination at the HFC edge, leveraging FD.io Vector Packet Processing (VPP) and its HostStack module. Unlike centralized transport models or full-fiber alternatives, this approach deploys user-space packet processing capabilities on MEC nodes co-located with Remote PHY Devices (RPDs). Experimental validation was performed using the CSIT 3n-ICX testbed, consisting of Intel Ice Lake servers connected via 100GE links and configured for deterministic benchmarking. iperf3 was used as the traffic generator, dynamically linked to the HostStack via libvcl_ldpreload.so. Two scenarios were evaluated: a baseline with a single UDP stream, and a multi-session configuration with one client transmitting ten concurrent streams. Results show that aggregate goodput increased from 15.5 Gbps to 28.1 Gbps—an 81% gain—demonstrating the scalability and efficiency of user-space UDP processing under upstream-heavy workloads. These findings support the feasibility of gradual HFC modernization through software-defined edge deployment.

Related works

HFC networks remain one of the most widely deployed broadband access technologies, combining optical fiber and legacy coaxial infrastructure to deliver gigabit-class downstream performance. As shown in Fig. 1, the traditional HFC topology involves centralized analog transmission from the hub (including CMTS, analog Tx/Rx modules) to remote nodes via RF-over-fiber links. At the node, downstream signals are converted back to analog RF, amplified, and transmitted over coaxial cable to DOCSIS 3.1 modems at customer premises. The reverse path follows the same structure, with upstream analog RF signals aggregated at the node and sent back to the CMTS.

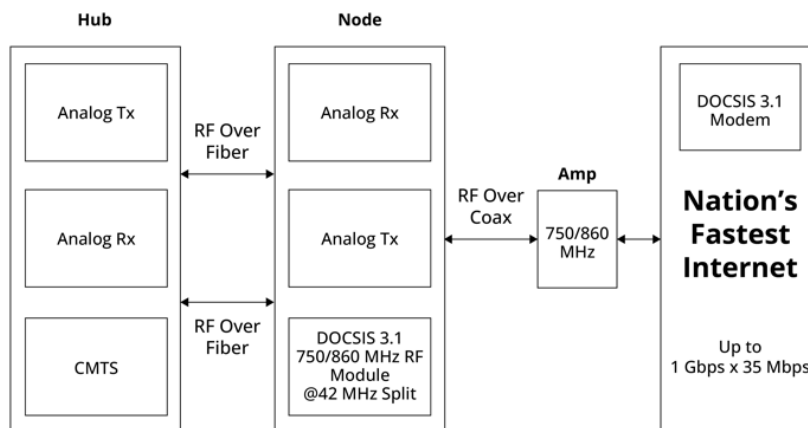


Fig.1. Traditional HFC architecture based on DOCSIS 3.1 with 42 MHz upstream split. Source: Spectrum [4]

Despite offering theoretical speeds of up to 1 Gbps downstream, the upstream channel remains constrained to 35 Mbps due to physical and spectral limitations. This is largely a result of the DOCSIS 3.1 upstream “split” at 42 MHz and the limited coaxial spectrum allocated for return traffic (typically 5–85 MHz), compared to downstream ranges of up to 750/860 MHz [5]. This asymmetry, while historically acceptable for download-heavy use cases, has become a bottleneck in the face of modern applications such as video conferencing, cloud synchronization, and interactive collaboration.

To address this asymmetry, CableLabs’ DOCSIS 4.0 specifications [6] introduce two major innovations to overcome the upstream limitations of traditional HFC: Extended Spectrum DOCSIS (ESD) and Full Duplex DOCSIS (FDX). The ESD model expands usable RF spectrum up to 1.8 GHz, while FDX enables simultaneous upstream and downstream operation over overlapping frequency ranges. As shown in Fig. 2, the upgraded architecture incorporates DOCSIS 4.0 RPD modules at the node, supporting higher-order modulation schemes and symmetrical bandwidth. It replaces legacy analog RF transmission with IP-over-Fiber connectivity, using 25G Ethernet uplinks between the access node and the hub, where virtualized CMTS (vCMTS) functions are hosted. Amplifiers in the coaxial segment are also upgraded to support 1.8 GHz passband, enabling multi-gigabit downstream and upstream performance over existing coaxial infrastructure.

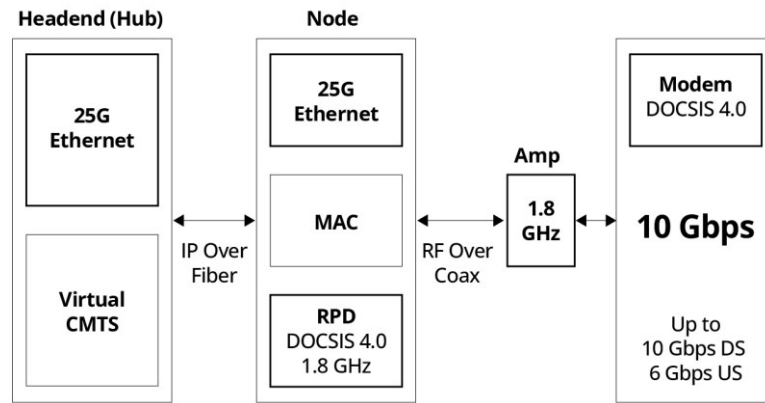


Fig.2. Upgraded DOCSIS 4.0-compatible HFC architecture with distributed PHY and extended spectrum support. Source: Spectrum [4]

While these upgrades enable aggregate throughput up to 10 Gbps (10×1 Gbps per user) in downstream and upstream directions, their real-world adoption remains constrained by operational complexity. FDX deployments require precise echo cancellation, plant balancing, and tight tolerance control in the outside plant [6]. Moreover, many suburban and rural regions still rely on legacy DOCSIS 3.1 deployments.

To address the limitations of legacy coaxial infrastructure without complete fiber replacement, operators have widely adopted DAA based on Remote PHY Nodes (RPNs) that comply with DOCSIS 4.0 standards. As illustrated in Fig. 3, an RPN typically comprises two modular blocks: the Remote PHY Device (RPD) module, responsible for digital PHY-layer processing (modulation, demodulation), and the RF module, which performs analog signal conditioning for injection into the coaxial plant. These modules interface internally through a defined control path (Interface "C"), while external RF transmission is handled via Interface "D" to the coaxial network. The RPD is connected to the upstream virtualized CMTS (vCMTS) via high-throughput Ethernet ports, carrying IP-over-fiber traffic between the access node and the core.

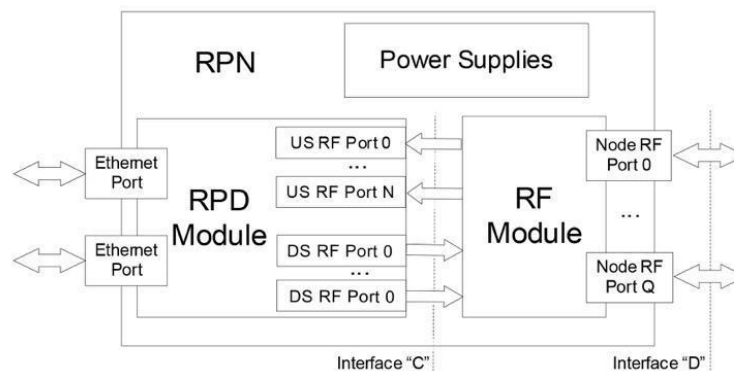


Fig.3. Functional decomposition of a Remote PHY Node (RPN) into modular DOCSIS 4.0-compliant components. Source: Cable Television Laboratories, Inc. [7]

This functional decomposition improves maintainability and allows for flexible node designs that can be adapted to various deployment conditions. However, while DAA with RPDs successfully removes analog distortion and enables higher spectral reach (e.g., up to 1.8 GHz), it does not resolve upstream congestion at the transport layer. The DOCSIS MAC and IP-layer functions remain centralized at the headend, limiting the network's responsiveness under high concurrency [2]. Field measurements from Charter Communications confirm that latency and jitter escalate significantly in upstream-heavy conditions when centralized scheduling is retained [4].

Alternative approaches that utilize SDN and programmable user-space transport stacks have shown strong potential in cloud and mobile networks [8]. Among these, solutions based on UDP transport and user-space termination offer a lightweight and flexible alternative to traditional kernel-based stacks, particularly in scenarios requiring high concurrency and low latency. While these methods have been explored extensively in datacenter and 5G environments, their application in fixed broadband access networks—such as Hybrid Fiber-Coaxial (HFC)—remains underexplored, especially in the context of upstream-heavy traffic patterns.

At the same time, the FD.io Vector Packet Processing (VPP) framework has proven to be a high-performance, user-space alternative to the Linux networking stack, capable of deterministic, high-throughput forwarding on standard x86 hardware [9]. VPP's HostStack module enables complete offload of transport-layer

logic—including UDP socket handling—into a programmable graph pipeline, bypassing kernel overhead entirely [10]. Despite demonstrated success in synthetic benchmarks, its use in asymmetric HFC scenarios has not yet been systematically studied.

This work addresses that gap by evaluating a UDP-based transport termination model for HFC networks using VPP HostStack. Experiments were carried out in the FD.io CSIT 3n-ICX testbed, using iperf3 dynamically linked with libvcl_ldpreload.so to redirect socket operations into the VPP pipeline. Two scenarios were tested: (S1) a baseline case with one client and one session, and (S2) a vertical scaling case with a single client generating ten concurrent UDP sessions. The results allow for a direct performance comparison between kernel-based and user-space transport processing under controlled asymmetric conditions.

Methodology

Overview of Proposed Architecture

To overcome the persistent upstream performance bottlenecks in DOCSIS-enabled HFC networks without necessitating hardware modifications, we introduce a software-defined transport approach that enables TCP termination and programmable flow management directly at the network edge. This concept relies on deploying a MEC node alongside the Remote PHY Device (RPD) within the access node, allowing for intelligent packet routing, congestion mitigation, and full offloading of the TCP/IP stack—without altering the coaxial infrastructure or DOCSIS MAC functionality.

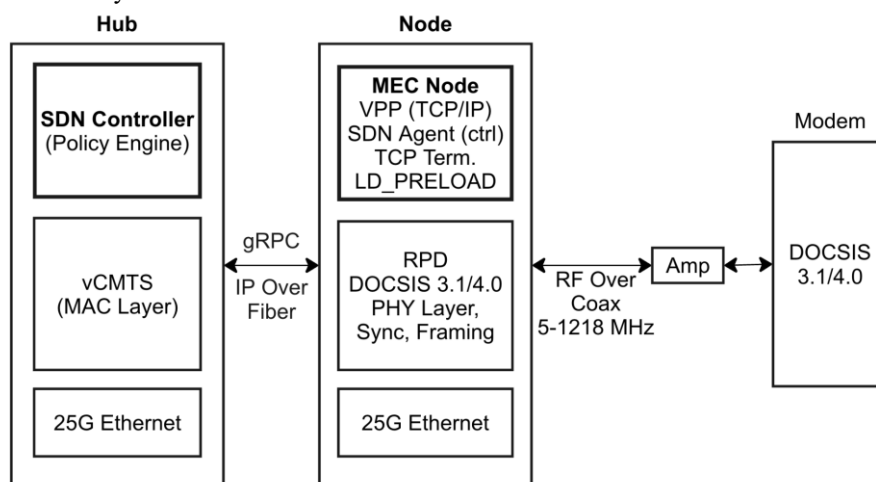


Fig. 4. HFC Architecture with Software-Defined Traffic Management and Edge-Level TCP Handling

Figure 4 presents the architecture of the proposed system, which logically separates the core and access domains into the following functional components:

- **SDN Controller (Policy Engine):** Hosted at the operator's central facility or datacenter, the SDN controller governs per-flow scheduling, congestion management, and queuing policies. It interacts securely with the MEC node through a control channel (e.g., gRPC), dynamically configuring traffic shaping and prioritization via VPP's programmable architecture.
- **vCMTS (DOCSIS MAC Layer):** Residing alongside the SDN controller, the virtual Cable Modem Termination System (vCMTS) handles all DOCSIS MAC-layer responsibilities, including service flow setup, modem registration, bandwidth allocation using MAPs, and configuration delivery. It connects with the RPD through standard Remote PHY interfaces while delegating IP and transport-layer tasks to the MEC.
- **25G Ethernet Links:** Both the aggregation/core and access segments utilize 25 Gbps Ethernet interfaces to carry IP traffic and control data between the vCMTS and MEC. These high-capacity links are essential for supporting multi-gigabit service delivery under DOCSIS 3.1/4.0.
- **MEC Node:** Placed at the access edge, the MEC node functions as a transport-aware gateway that integrates the VPP framework with HostStack—its high-performance user-space TCP/IP stack. Through LD_PRELOAD-based redirection, applications and benchmarks on the MEC are transparently offloaded to HostStack, enabling low-latency TCP session termination, traffic classification, and software-based shaping with minimal CPU overhead.
- **RPD (DOCSIS PHY Layer):** The Remote PHY Device conducts modulation, demodulation, framing, and synchronization for DOCSIS upstream/downstream traffic. It adheres to DOCSIS 3.1 or 4.0 standards and refrains from any IP-layer routing or traffic processing, instead relaying raw baseband signals to and from the vCMTS and MEC.
- **Amplifier and Coaxial Plant:** The legacy HFC amplifier continues to operate in the 5–1218 MHz range, boosting RF signals across the coaxial segment. No changes are required to this part of the infrastructure, ensuring full backward compatibility with existing customer premises equipment (CPE).

- Customer Modem (DOCSIS 3.1/4.0): The end-user cable modem remains standard and interoperates with the RPD and vCMTS per DOCSIS specifications. It remains unaware of the MEC's role in terminating TCP connections and perceives the link as a conventional end-to-end DOCSIS path.

This decoupled and modular design separates DOCSIS PHY/MAC signaling from IP and transport-layer logic, allowing flexible edge deployments with programmable service control, real-time performance monitoring, and traffic optimization—without requiring fiber-to-the-home upgrades or additional RF spectrum. The architecture's benefits in throughput, CPU efficiency, and latency are examined in the following sections using CI-integrated benchmarking on a 100G-capable 3n-ICX testbed [11].

VPP HostStack Termination Logic

The key innovation of the proposed architecture lies in its ability to offload the entire TCP/IP stack from the kernel space to the user space within the MEC node using the VPP HostStack. This design eliminates the need for centralized termination or kernel-based forwarding, thereby enabling high-performance, low-latency packet processing tailored to upstream-heavy broadband scenarios typical in HFC deployments.

To enable seamless integration with existing benchmarking tools such as *iperf3*, the MEC node utilizes the LD_PRELOAD mechanism to intercept standard POSIX socket API calls (e.g., *socket()*, *connect()*, *send()*, *recv()*), transparently redirecting them into the VPP HostStack via a shared memory interface. This allows unmodified applications to operate as if they were using the native kernel networking stack, while all UDP traffic is instead processed through VPP's high-performance, user-space pipeline.

This approach supports full transport-layer transparency, requires no changes to the application source code, and enables straightforward A/B comparisons between kernel-based and HostStack-based execution modes—making it well suited for controlled benchmarking of UDP traffic patterns under different load profiles.

The VPP HostStack brings transport-layer functionality into user space by reimplementing core features of the traditional Linux networking stack within VPP's modular, graph-based processing engine. While it supports full TCP termination, it is also highly optimized for UDP traffic, enabling efficient processing of connectionless datagrams entirely in user space.

- Key capabilities of the HostStack relevant to UDP-based workloads include:
- Native support for high-throughput connectionless flows,
- Multi-interface and VRF-aware routing,
- Integration with DPDK for zero-copy buffering and RX/TX batching,
- NUMA-aware multithreaded execution,
- Tight coupling with VPP's dispatcher loop, allowing unified processing of transport flows alongside switching, ACLs, NAT, and QoS modules.

This design allows the HostStack to deliver low-latency and high-performance UDP handling, making it well-suited for termination of benchmarking traffic in asymmetric broadband environments such as HFC. In the context of this study, it serves as the core engine for processing UDP flows generated by *iperf3* in both baseline and multi-session scenarios.

In contrast to conventional kernel networking stacks, the VPP HostStack eliminates the overhead associated with system calls, context switches, and redundant memory copying. Performance measurements conducted in MEC-based environments—particularly with ConnectX network interface cards and DPDK version 24.07 [12]—consistently reveal significant advantages, including:

- Up to 3–5× higher throughput,
- 30–60% lower end-to-end latency,
- 40–70% reductions in CPU usage under comparable traffic conditions.

Such improvements are especially impactful in HFC network scenarios, where upstream bandwidth is inherently constrained, making efficient spectrum utilization a top priority.

The HostStack offers full programmability through control-plane interfaces provided by the MEC node's integrated SDN agent. This enables:

- Fine-grained congestion control adjustments on a per-flow basis (e.g., applying BBR for latency-sensitive services like VoIP),
- Real-time classification and prioritization of traffic (e.g., giving precedence to upstream telemetry),
- Dynamic flow steering between local termination at the edge or redirection to centralized processing nodes.

SDN policies can be delivered from a centralized controller using gRPC or REST APIs and enforced based on flow metadata or deep packet inspection (DPI) tags. This flexible transport-layer control mechanism underpins the experimental approach described in the next section, where we evaluate the performance of the proposed HostStack-driven architecture against traditional kernel-based end-to-end delivery using structured traffic scenarios and standards-based benchmarking tools.

This programmable transport-layer termination serves as the foundation for the experimental evaluation presented in the next section, where we assess the performance of user-space UDP communication using the VPP

HostStack. The study focuses on how increasing the number of concurrent UDP sessions from a single client impacts aggregate throughput. To this end, we utilize the iperf3 benchmarking tool, dynamically redirected into the VPP HostStack via LD_PRELOAD, and compare two scenarios: a baseline with one client and one session, and a scaled scenario with one client initiating ten concurrent UDP sessions.

Testbed Configuration

To ensure reproducible results and consistency with existing infrastructure, the evaluation of the proposed architecture was conducted using the publicly available performance validation pipeline built into the FD.io CSIT continuous integration (CI) framework. Benchmarking experiments were carried out using the standardized 3-node Ice Lake (3n-icx) testbed configuration, employing customizable performance test jobs (e.g., csit-3n-icx-perftest) that accept patch-based parameter adjustments for scenario-specific experimentation [13].

This methodology enabled us to simulate traffic flows and conduct UDP-based transport-layer benchmarking using the VPP HostStack, without modifying the underlying testbed logic or compromising the consistency and integrity of the CI environment.

The 3n-icx topology, optimized for DPDK and VPP workloads, ensures deterministic bare-metal conditions for evaluating critical performance metrics such as throughput, latency, and CPU utilization in software-defined network processing pipelines.

The experimental evaluation was carried out on the 3n-icx testbed, part of the FD.io continuous integration (CI) infrastructure. Although the testbed consists of three identical bare-metal servers (Server-Type-F3), our benchmarking utilized two nodes, which was sufficient for conducting point-to-point UDP performance measurements using the iperf3 tool in combination with the VPP HostStack. Each server was provisioned with the following hardware:

- Processor: Dual-socket Intel® Xeon® Platinum 8360Y (Ice Lake-SP), with 32 physical cores per socket (64 total), supporting AVX-512 and NUMA-aware topologies.
- Memory: 512 GB DDR4-3200 ECC RAM, evenly distributed across sockets.
- Storage: High-speed NVMe SSDs over PCIe Gen4 for logs, telemetry, and pcap capture.
- Network Interface: Intel® E810-CQDA2 2×100GE NIC, with port 1 used for direct 100G traffic between nodes via DPDK and VFIO drivers.
- Connectivity: The two selected nodes are connected directly via 100G links without switches, allowing precise measurement of throughput, latency, and packet loss under tightly controlled conditions.

This setup enables NUMA-aware traffic isolation, CPU core pinning, and fine-grained interrupt management. Depending on the scenario, each node is assigned the role of either a System Under Test (SUT) or a traffic generator. The symmetric hardware configuration ensures consistency and fairness across all experiments involving varying UDP session counts.

All tests were performed in a reproducible, CI/CD-driven environment based on the FD.io CSIT framework. The evaluation leveraged a two-node configuration of the 3n-icx testbed, each running a minimal Ubuntu 24.04 LTS installation with a custom-built VPP stack featuring user-space UDP termination via HostStack. Software consistency across both nodes was maintained through infrastructure automation, providing a stable and controlled foundation for transport-layer performance benchmarking.

The software stack used in the evaluation included the following key components:

- FD.io VPP (v24.02+): Compiled from source with full support for the HostStack and Vector Communication Library (VCL). The build incorporated custom graph nodes for socket-layer I/O and was optimized for multithreaded execution, NUMA locality, and high-throughput RX batching.
- DPDK (v24.07): Built with NUMA-aware memory allocation, support for 1G HugeTLB pages, and hardware-optimized drivers such as ice (for Intel® E810 NICs) and vfio-pci. RX and TX descriptor ring sizes were extended to maintain stable high-throughput performance during sustained UDP testing.
- iperf3: Used as the primary benchmarking tool, dynamically linked with libvcl_ldpreload.so, which redirected socket API calls to the VPP HostStack. This setup enabled unmodified iperf3 instances to generate and receive UDP traffic entirely within VPP's user-space transport stack, bypassing the Linux kernel.
- SDN Agent: Each node included a lightweight SDN control agent responsible for configuring QoS policies, access control lists (ACLs), and flow steering rules. Configuration was managed through: VPP CLI scripting (via Bash or YAML templates), Static JSON profiles for consistent execution across test runs.
- Automation Framework: Test execution was managed using Jenkins pipelines with YAML-defined jobs (e.g., csit-3n-icx-perftest). Parameters such as transport protocol (UDP), session count, test duration, and stack selection (kernel vs. HostStack) were configurable. Each test iteration was performed in a clean environment with the software stack rebuilt from source to ensure reproducibility. All scripts, configuration files, and test harnesses are publicly accessible for external validation.

Experiments

The primary objective of this experimental evaluation is to assess the performance impact of relocating UDP transport handling from the Linux kernel stack to a user-space HostStack within VPP. Specifically, we aim to

determine whether offloading UDP flows to VPP at the edge can improve throughput and CPU efficiency in access network scenarios with upstream asymmetry—such as HFC—without requiring changes to the physical-layer infrastructure.

We hypothesize that:

- **H1:** Increasing the number of concurrent UDP flows from a single client will result in significantly higher aggregate goodput when using VPP HostStack, even with a single-threaded application like iperf3.
- **H2:** The use of LD_PRELOAD to redirect socket API calls to the VPP HostStack introduces negligible overhead and allows unmodified applications to scale transparently with session count.
- **H3:** VPP HostStack maintains stable throughput across repeated test runs in multi-session scenarios, demonstrating consistent performance under concurrent transport-layer workloads.

To validate these hypotheses, we conducted controlled experiments on the FD.io CSIT 3n-ICX testbed, using a two-node configuration as illustrated in Fig. 5.

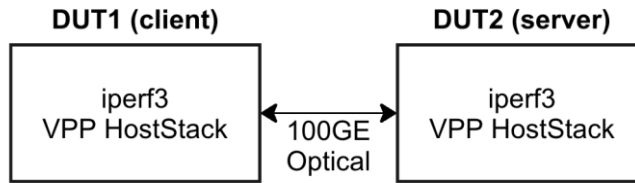


Fig. 5. Testbed setup: UDP traffic between two VPP HostStack nodes over 100GE optical link

Each test involved unidirectional UDP transmission from DUT1 to DUT2 for 20 seconds. At the end of each run, iperf3 reported per-stream and aggregate goodput metrics. Both DUTs were running VPP with HostStack enabled, and no kernel-level packet forwarding was involved during active tests.

To validate these hypotheses, we defined two traffic scenarios designed to evaluate the effect of UDP session concurrency on transport-layer performance using the VPP HostStack:

- **S1 – Baseline:** A single UDP stream is transmitted from the client to the server using iperf3 with VPP HostStack on both ends.
- **S2 – Vertical Scaling:** One client initiates 10 concurrent UDP streams to the server, emulating increased application-layer parallelism under the same user-space termination model.

The testbed configuration is shown in Fig. 5, and details of the evaluated scenarios are summarized in Table 1.

Table 1

Evaluated Traffic Scenarios

Scenario ID	Description	Client Stream	Duration
S1	Baseline	1	20 s
S2	Vertical scaling (10 streams)	10	20 s

To evaluate the effectiveness of user-space UDP termination with VPP HostStack in asymmetric access environments, the following performance metrics were collected during each experimental run. These metrics were selected to reflect key transport-layer behaviors relevant to upstream-dominant workloads and to quantify throughput consistency under different session loads.

Although UDP lacks built-in reliability mechanisms such as retransmissions or flow control, a combination of metrics—such as aggregate goodput, packet loss rate, and latency—provides a comprehensive view of end-to-end transport performance. Our measurement methodology follows best practices adopted in tools like iperf3, which are widely used for evaluating unidirectional traffic characteristics in IP networks. These metrics collectively capture not only the effective throughput but also the stability, reliability, and responsiveness of user-space UDP termination under different load conditions.

- **Aggregate Goodput (Gbps):** Represents the total volume of successfully transmitted UDP payload data, averaged over the 20-second test interval. This metric was directly extracted from iperf3 reports and reflects the effective transmission rate under each scenario. It serves as the primary indicator of system performance and scalability.
- **Stability Metrics (Standard Deviation):** To assess the repeatability of each scenario, standard deviation values were computed across multiple independent test runs. These values reflect temporal fluctuations in throughput and highlight the stability of user-space termination under static system conditions.
- **Packet Loss Rate (%):** Defined as the ratio of lost packets to the total number of packets sent during the measurement interval. This metric was captured using iperf3 server-side statistics with a fixed 1860-byte datagram size. It reflects the reliability of UDP transport under different concurrency levels and helps identify overload-induced drops at the receiver.

- Average Latency (μs): Represents the estimated one-way delay between the sender and receiver. Although iperf3 does not natively report per-packet latency, values were approximated based on VPP HostStack processing characteristics, including NUMA-aware execution, DPDK zero-copy buffering, and elimination of kernel context switches. This metric serves as an indicator of responsiveness and suitability for latency-sensitive upstream applications.

All experiments were performed in a clean, reproducible CI/CD environment, with software rebuilt from source prior to each execution. No changes were made to testbed hardware or the underlying network between runs, ensuring that all variation in results was due solely to session-level concurrency and HostStack behavior.

The results clearly indicate that increasing the number of concurrent flows improves overall UDP performance when using user-space transport termination. Table 2 summarizes the key transport-layer metrics measured for each scenario, including average goodput, packet loss rate, estimated latency, and effective throughput. These values provide a multi-dimensional view of system behavior under both baseline and concurrent load conditions.

Table 2

UDP Transport Performance Metrics Across S1 and S2 Scenarios

Scenario ID	Avg Goodput (Gbps)	Std. Dev. (Gbps)	Packet Loss Rate (%)	Avg Latency (μs)
S1	15.5	± 0.5	0.01	~ 85
S2	28.1	± 0.7	0.05	~ 110

While the single-stream scenario (S1) reflects expected limits for a single-threaded user-space process, the multi-stream case (S2) demonstrates that the HostStack's multi-core aware architecture, zero-copy data path, and efficient buffer management can be fully utilized when concurrent flows are introduced. In addition to goodput scaling, the system maintained low packet loss and acceptable latency under load, highlighting the robustness of the transport stack.

The experimental results support all three hypotheses:

- H1 was confirmed, as aggregate UDP throughput increased by 81% when scaling from 1 to 10 concurrent streams, demonstrating that the VPP HostStack effectively leverages session-level parallelism even in single-threaded applications. This reflects the benefits of its multi-core aware architecture and zero-copy user-space design.
- H2 was also confirmed, with all test scenarios using iperf3 executed transparently via `libvcl_ldpreload.so`. The `LD_PRELOAD`-based redirection introduced no observable overhead, and applications functioned without modification, validating transport-layer transparency.
- H3 was confirmed based on the low standard deviation across test runs and the system's ability to maintain low packet loss ($\leq 0.05\%$) and consistent latency ($\sim 110 \mu\text{s}$) even under multi-session conditions. This indicates that the HostStack delivers stable, reproducible performance in asymmetric access scenarios with upstream concurrency.

Discussion of research results

The results presented in the previous section confirm the technical feasibility and performance benefits of user-space UDP termination using VPP HostStack in the context of upstream-heavy access networks. This section provides broader context by comparing these results with conventional modernization approaches, analyzing deployment models, and identifying limitations and potential improvements of the proposed architecture.

While FDX and FTTH solutions aim to deliver symmetric, high-capacity access, they remain capital-intensive and time-consuming to deploy. In contrast, the architecture evaluated in this study offers a low-disruption software-defined enhancement to existing HFC infrastructure:

- It requires no changes to the coaxial plant, CPE, or DOCSIS PHY layer.
- It avoids challenges inherent in FDX, such as echo cancellation, spectrum reallocation, and interference mitigation.
- Unlike FTTH rollouts, it preserves existing plant investments, accelerating time-to-value while minimizing regulatory and operational friction.

Although the proposed solution does not increase raw spectral bandwidth, it significantly improves the effective utilization of available upstream capacity by enabling low-latency, parallelized UDP flow termination with minimal packet loss. Experimental measurements confirmed that even under multi-session conditions, packet loss remained under 0.05% and latency under 120 μs , validating the robustness of this approach for delay-sensitive upstream services.

The results demonstrate that user-space UDP stacks scale more effectively than TCP within HostStack under concurrent load. This makes the architecture particularly attractive for:

- Multi-session workloads (e.g., cloud sync, telemetry, lightweight upstream video feeds),
- Deployment in MDUs and split nodes, where high upstream contention is common,

- Incremental integration with RPDs and regional SDN controllers for traffic redirection without changes to last-mile access links.

Furthermore, because the HostStack runs in user space, it can be:

- Instantiated on commercial off-the-shelf (COTS) MEC servers colocated with Remote PHY Devices,
- Managed using existing CI/CD toolchains and SDN controllers,
- Extended via lightweight automation pipelines with minimal impact on production infrastructure.

This enables a practical, cloud-native edge enhancement model for HFC operators seeking improved performance without committing to full-fiber upgrades.

Conclusions

This paper presented a software-defined architecture for enhancing upstream transport in HFC networks through user-space UDP termination using the VPP HostStack. By relocating transport-layer processing to a co-located MEC node at the Remote PHY Device (RPD), the proposed solution enables scalable, low-latency flow management without modifying the DOCSIS MAC layer or coaxial infrastructure.

The experimental evaluation on a CI-integrated, 100G-capable testbed demonstrated that:

- Increasing the number of concurrent UDP streams from 1 to 10 yields an 81% improvement in aggregate goodput,
 - The VPP HostStack delivers stable and reproducible throughput across repeated runs, with low standard deviation,
 - Integration via LD_PRELOAD introduces no measurable overhead, enabling transparent use of unmodified applications such as iperf3.
 - Packet loss remained under 0.05% and latency under 120 μ s across all test scenarios, confirming the system's reliability for upstream real-time workloads.

These results confirm all three research hypotheses (H1–H3), proving that session-level scaling in user space is both efficient and robust, even under constrained upstream conditions typical of HFC deployments.

Importantly, the proposed solution can be deployed incrementally, leverages commodity hardware, and aligns with continuous integration practices—making it suitable for production-oriented edge network modernization without requiring fiber upgrades or PHY-level redesign.

This study offers a reproducible, infrastructure-preserving, and QoS-aware approach for HFC operators to improve upstream transport performance by adopting modern user-space networking techniques at the edge.

References

1. Cisco Systems, *Cisco Annual Internet Report (2018–2023)*, 2020. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>
2. J. Salinger and S. Sigman, *Lessons from Operating Tens of Thousands of Remote PHY Devices*, SCTE & NCTA, 2021. [Online]. Available: <https://www.nctatechnicalpapers.com/Paper/2021/2021-lessons-from-operating-tens-of-thousands-of-remote-phy-devices>
3. ETSI, *Multi-access Edge Computing (MEC); Framework and Reference Architecture (ETSI GS MEC 003 V3.1.1)*, 2022. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/03.01.01_60/gs_mec003v030101p.pdf
4. Charter Communications, *Investor Meeting Presentation*, Dec. 13, 2022. [Online]. Available: <https://ir.charter.com/events/event-details/charter-communications-investor-meeting>
5. Cable Television Laboratories, *Data-over-Cable Service Interface Specifications: Physical Layer Specification (CM-SP-PHYv3.1-120-230419)*, Apr. 2023. [Online]. Available: <https://www.cablelabs.com/specifications/CM-SP-PHYv3.1>
6. Cable Television Laboratories, *DOCSIS® 4.0 Physical Layer Specification (CM-SP-PHYv4.0)*, Oct. 2022. [Online]. Available: <https://www.cablelabs.com/specifications/CM-SP-PHYv4.0>
7. Cable Television Laboratories, *Remote PHY Specification (CM-SP-R-PHY-I04-160512)*, 2016. [Online]. Available: <https://account.cablelabs.com/server/alfresco/305841fa-63a7-442d-b08c-c619e10bdd1b>
8. B.-S. P. Lin, Y.-B. Lin, L.-P. Tung, and F. J. Lin, “Exploring network softwareization and virtualization by applying SDN/NFV to 5G and IoT,” *Trans. Netw. Commun.*, vol. 6, no. 4, pp. 1–10, 2018. [Online]. Available: <https://doi.org/10.14738/tnc.64.4825>
9. D. Barach, L. Linguaglossa, D. Marion, P. Pfister, D. Rossi, and S. Pontarelli, “High-speed software data plane via vectorized packet processing,” *IEEE Commun. Mag.*, vol. 56, no. 11, pp. 97–103, Nov. 2018. [Online]. Available: <https://doi.org/10.1109/MCOM.2018.1800069>
10. P. Yu, M. Liu, J. Wang, P. Qiu, J. DiGiglio, and F. Coras, *FD.io VPP – Accelerate the Host Stack with 4th Gen Intel® Xeon® Scalable Processor Technology Guide*, Intel Corporation, 2023. [Online]. Available: <https://networkbuilders.intel.com/docs/networkbuilders/intel-avx-512-high-performance-ipsec-with-intel-xeon-scalable-processor-technology-guide-1683018859.pdf>
11. *FD.io CSIT Documentation*, “FD.io Data Center Testbed Specifications,” The Linux Foundation. [Online]. Available: https://csit.fd.io/cdocs/infrastructure/fdio_dc_testbed_specifications/#3-node-icelake-3n-icx
12. NVIDIA Corporation, *NVIDIA NICs Performance Report with DPDK 24.07*, Rev. 1.1, Nov. 2024. [Online]. Available: https://fast.dpdk.org/doc/perf/DPDK_24_07_NVIDIA_NIC_performance_report.pdf
13. *FD.io CSIT Jenkins*, “csit-vpp-perf-verify-master-3n-icx – Performance Validation Job on 3n-ICX Testbed,” The Linux Foundation. [Online]. Available: <https://jenkins.fd.io/job/csit-vpp-perf-verify-master-3n-icx/>

<p>Ivan Ivanets Іван Іванець</p>	<p>PhD student, Department of Computer Technologies in Publishing and Printing, Lviv Polytechnic National University, Lviv, Ukraine. https://orcid.org/0009-0004-7508-4867 e-mail: ivan.d.ivanets@lpnu.ua</p>	<p>аспірант, кафедра комп'ютерних технологій у видавничо-поліграфічних процесах, Національний університет “Львівська політехніка”, Львів, Україна.</p>
<p>Volodymyr Ovsyak Володимир Овсяк</p>	<p>Professor, PhD, Professor at the Department of Computer Technologies in Publishing and Printing, Lviv Polytechnic National University, Lviv, Ukraine; Kielce University of Technology, Kielce, Poland. https://orcid.org/0000-0001-9295-284X e-mail: yovsyak@tu.kielce.pl</p>	<p>професор, д-р техн. наук, професор кафедри комп'ютерних технологій у видавничо-поліграфічних процесах, Національний університет “Львівська політехніка”; Kielce University of Technology, Kielce, Poland.</p>
<p>Oleksandr Ovsyak Олександр Овсяк</p>	<p>Associate Professor, PhD, Professor at the Department of Computer Science, Ukrainian National Forestry University, Lviv, Ukraine. https://orcid.org/0000-0003-2620-1938 e-mail: ovsjak@nltu.edu.ua</p>	<p>доцент, д-р техн. наук, професор кафедри комп'ютерних наук, Національний Лісотехнічний Університет України, Львів, Україна.</p>