UDC 004.7
DOI https://doi.org/10.32782/2663-5941/2026.1.2/15

Ivanets I.D.
https://orcid.org/0009-0004-7508-4867
Lviv Polytechnic National University

AN HFC NETWORK ARCHITECTURE WITH TCP TERMINATION AT EDGE NODES

Hybrid fiber–coaxial (HFC) access networks remain a central component of broadband infrastructure, yet core-centric designs with DOCSIS grant-based upstream scheduling increasingly struggle to support upstream-intensive workloads such as video conferencing, real-time collaboration, and large-scale data synchronization. Conventional modernization paths–DOCSIS 4.0 spectrum expansion, full-fiber migration (FTTH), or disruptive distributed access architectures–require substantial capital expenditure, plant re-engineering, and long rollout cycles. This paper proposes and experimentally evaluates an edge-based transport architecture that shifts critical traffic processing from centralized virtual cable modem termination systems (vCMTS) to Multi-access Edge Computing (MEC) nodes co-located with Remote PHY Devices in the HFC access domain. The MEC node embeds the FD.io Vector Packet Processing framework with HostStack (a user-space TCP/IP stack) and uses LD_PRELOAD-based socket interception to terminate unmodified applications transparently in user space, without changes to the coaxial plant, DOCSIS PHY/MAC signaling, or customer premises equipment. The solution is validated on the publicly accessible FD.io CSIT three-node Ice Lake (3n-icx) testbed with 25-GbE links by comparing three scenarios: a kernel-to-kernel TCP baseline, HostStack-to-kernel termination, and symmetric HostStack-to-HostStack termination. Traffic is generated by iperf3 with configurable connection counts and congestion control algorithms, while an automated CI/CD pipeline reports aggregate throughput, per-flow retransmissions, MEC-side CPU utilization, and run-to-run variability. Experimental results show that edge-based HostStack termination increases average TCP throughput by up to 43%, reduces retransmissions by approximately 75%, and lowers MEC-side CPU utilization by about 40% relative to the kernel baseline. These findings indicate that softwarized transport processing at edge nodes can improve the effective use of constrained upstream capacity in HFC networks and offers cable operators a scalable, incrementally deployable modernization path between legacy DOCSIS 3.1 deployments and more disruptive DOCSIS 4.0 or FTTH upgrades.

Keywords: *edge termination, transport-layer processing, hybrid fiber-coaxial networks, vector packet processing, performance evaluation.*

Formulation of the problem. The rapid growth of upstream-intensive services – such as real-time video conferencing, interactive collaboration, and large-scale cloud data synchronization – has exposed structural limitations of asymmetric broadband access, where the upstream segment becomes the primary bottleneck [1], [2]. Hybrid fiber–coaxial (HFC) access networks standardized by the Data Over Cable Service Interface Specification (DOCSIS) family remain widely deployed due to cost efficiency and the longevity of the outside plant, yet grant-oriented upstream scheduling in DOCSIS makes transport behavior sensitive to scheduling cycles and queue dynamics. In contrast, fiber-to-the-home (FTTH) access provides better physical-layer properties and upstream symmetry, but full-fiber migration is often constrained by high deployment cost, civil engineering complexity, regulatory barriers, and operational risks. As a result, operators commonly rely on incremental modernization, including DOCSIS 4.0 upgrades [3] and Distributed Access Architectures (DAA) enabled by Remote PHY Devices (RPDs) [4], [5]. While these steps push fiber deeper, they do not directly eliminate transport-layer bottlenecks because end-to-end transport control remains largely core-centric. This motivates an edge-based approach in which transport processing is moved closer to the access domain. The FD.io Vector Packet Processing (VPP) framework provides a high-throughput packet-processing platform [6], and its HostStack component enables user-space transport termination on Multi-access Edge Computing (MEC) nodes [7].

Application compatibility can be preserved via Portable Operating System Interface (POSIX) socket interception using the LD_PRELOAD mechanism [8]. Therefore, the problem addressed in this work is to define and validate an HFC access architecture with transport termination [9] at MEC edge nodes that measurably improves upstream throughput and retransmission behavior while preserving compatibility with the DOCSIS plant and customer equipment.

Analysis of recent research and publications. Legacy HFC access networks are commonly described as hierarchical systems with centralized control at the headend and radio-frequency distribution over coaxial segments toward customer premises [10, p. 26]. The upstream and downstream directions are asymmetric by design, and upstream performance is sensitive to scheduling cycles and queue dynamics.

A major modernization line in the literature focuses on extending DOCSIS capabilities. DOCSIS 4.0 introduces full-duplex (FDX) and extended-spectrum (ESD) operation modes to increase upstream capacity and move toward upstream–downstream parity. [11] Published guidance also notes stricter requirements on plant conditioning, interference control, and echo cancellation that can translate into plant-wide upgrades and higher operational complexity.

In parallel, fiber-based access systems such as XGS-PON and 10G-EPON are widely discussed as long-term alternatives because they provide native upstream symmetry and favorable physical-layer characteristics [12]. However, full-fiber migration, including FTTH, remains capital intensive and logistically challenging due to outside-plant replacement, customer premises equipment (CPE) upgrades, and permitting constraints in dense urban environments [13].

As an intermediate path, the literature highlights DAA as a practical approach that pushes parts of access processing deeper into the network while keeping higher-layer control largely centralized [14]. A representative realization is the RPD architecture, which relocates digital-to-radio-frequency conversion closer to subscribers and improves signal quality and observability [15].

Nevertheless, the reviewed publications also reveal a persistent limitation: improvements in the physical layer and access distribution do not directly eliminate transport-layer inefficiencies in upstream-intensive scenarios. Studies on broadband cable networks show that queueing dynamics and delay-control mechanisms remain critical for application performance and fairness, which indicates that transport behavior can remain constrained even when access-layer modernization improves signal quality [16]. Under high upstream concurrency, centralized processing and long feedback loops reduce responsiveness to access-edge dynamics and amplify the impact of scheduling and buffering on flow control [17]. Furthermore, while DAA and RPD-based modular access architectures provide important benefits by relocating parts of the access functions deeper into the network, they primarily address access-layer partitioning and do not inherently provide transport termination or flow-level control at the edge [18]. Therefore, despite progress in DOCSIS evolution and DAA/RPD deployments, an unresolved gap remains in access-edge transport processing, particularly in experimentally validated designs that relocate transport termination closer to the access domain while preserving compatibility with existing HFC infrastructure.

Task statement. The purpose of this article is to develop and experimentally validate an HFC access architecture in which TCP termination is performed at MEC edge nodes using VPP HostStack, with the goal of improving upstream TCP efficiency under DOCSIS 3.1 constraints while keeping the existing HFC plant and customer equipment unchanged.

In accordance with the stated purpose, the following tasks are solved.

1. To define the target HFC architecture and the functional placement of transport termination at MEC edge nodes co-located with RPDs, while preserving the conventional DOCSIS control and scheduling model.
2. To implement TCP termination in user space using HostStack and to ensure transparent interoperability with unmodified applications by redirecting standard socket traffic into VPP.
3. To specify the experimental setup and test methodology on the Continuous System Integration and Testing (CSIT) [19] 3n-icx testbed, i.e., a publicly accessible three-node Ice Lake platform interconnected with 25 GbE (25-gigabit Ethernet) links [20].
4. To define the compared scenarios: kernel-to-kernel baseline (both endpoints using the Linux kernel TCP stack), HostStack-to-kernel termination (one endpoint terminated by HostStack), and HostStack-to-HostStack termination (both endpoints terminated by HostStack).
5. To evaluate the architecture under upstream-constrained and high-concurrency conditions using iperf3 traffic generation [21], and to analyze aggregate throughput, retransmission behavior, RTT (round-trip time), and MEC-side CPU utilization as primary outcome metrics [22], [23].

Outline of the main material of the study. To mitigate upstream limitations in DOCSIS-based HFC networks without physical upgrades, this study proposes a software-defined transport architecture that performs TCP termination and flow-level control at the network edge. The key design choice is to co-locate an MEC node with the RPD at the access node, enabling transport-aware packet processing close to the DOCSIS scheduling domain while leaving the coaxial segment and DOCSIS MAC unchanged.

Figure 1 presents the logical decomposition of the proposed system. The SDN controller [24] acts as a policy engine that defines per-flow rules and pushes them to the MEC node over a secure control interface. The vCMTS retains DOCSIS MAC responsibilities (service flows, registration, bandwidth allocation and configuration delivery) and communicates with the RPD via standard Remote PHY interfaces. The MEC node embeds VPP with HostStack and terminates TCP sessions in user space, allowing per-flow classification and shaping in software. Unmodified applications on the MEC node are transparently redirected into HostStack using LD_PRELOAD-based socket interception. The RPD remains a PHY-domain device responsible for modulation/demodulation and timing, while the amplifier/coax plant and cable modems remain unchanged and standards-compliant.

The evaluation was executed on the publicly accessible FD.io CSIT 3n-icx infrastructure using CI-driven performance job templates, which provides deterministic bare-metal conditions for repeatable benchmarking. The testbed uses three symmetric servers interconnected with direct 25-GbE links and DPDK-capable NICs [25], enabling controlled comparison of transport stacks under identical physical connectivity. The software environment was kept consistent across runs (Ubuntu 24.04, VPP with HostStack, DPDK, and automation via Jenkins and Robot Framework) [26], and scenarios were parameterized through CI templates to eliminate manual variability [27].

Figure 2 illustrates the benchmarking topology, and Table 1 summarizes the three evaluated scenarios: S1 (kernel-to-kernel) as a baseline, S2 (HostStack-to-kernel) as a partial termination case, and S3 (HostStack-to-HostStack) as a fully user-space termination case. In HostStack modes, iperf3 runs unmodified while its socket calls are intercepted and redirected into HostStack.

Table 2 shows that HostStack-based termination increases aggregate goodput relative to the kernel baseline, with S3 achieving the highest throughput and the lowest variance.

Compared to S1, S3 delivers a throughput gain of up to 43%, indicating that edge termination in user space improves effective utilization of constrained upstream capacity under concurrent flows.

Table 3 reports retransmission behavior, where HostStack termination substantially reduces retransmissions, consistent with shorter transport control loops and reduced sensitivity to centralized buffering and scheduling artifacts.

Table 4 compares MEC-side CPU utilization and shows that HostStack-based processing reduces CPU cost by roughly 40% relative to the kernel baseline, which is consistent with kernel bypass, batching, and user-space execution efficiency.

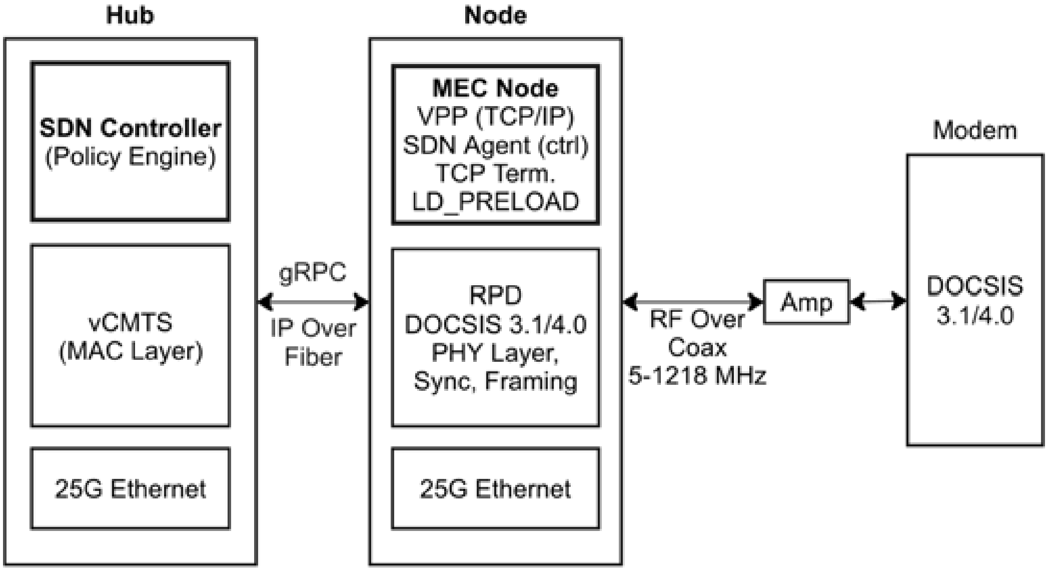

Fig. 1. Software-Defined HFC Architecture with Edge TCP Termination and SDN Policy Control

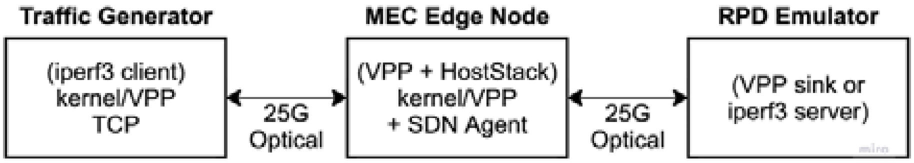

Fig. 2. Experimental topology of the CI-integrated 3n-icx testbed, featuring dynamic role allocation. Nodes are interconnected via direct 25G optical links

Table 1

Evaluated Traffic Scenarios

Scenario ID	Client Stack	Server Stack	TCP Term. Point
S1	Linux Kernel	Linux Kernel	Baseline reference (traditional architecture)
S2	VPP HostStack	Linux Kernel	Measure gains from partial HostStack usage
S3	VPP HostStack	VPP HostStack	Maximize stack-level performance

Table 2

Average Aggregate TCP throughput (Gbps) across S1-S3

Scenario ID	Avg Goodput (Gbps)	Std. Dev. (%)
S1	16.4	±0.9
S2	21.7	±0.6
S3	23.5	±0.4

Table 3

Average TCP retransmission rate (%) per scenario

Scenario ID	Retransmission Rate (%)	Std. Dev. (%)
S1	3.8	±0.7
S2	1.4	±0.5
S3	0.9	±0.2

Table 4

Average CPU utilization on MEC node per scenario

Scenario ID	Avg CPU utilization (%)	Std. Dev. (%)
S1	67.2	±2.1
S2	42.9	±1.6
S3	38.5	±1.3

Figure 3 summarizes the normalized comparison of throughput, retransmissions, and CPU utilization across S1–S3, highlighting that S3 simultaneously achieves the highest throughput, the lowest retransmissions, and the lowest CPU utilization.

The results confirm that relocating TCP termination to MEC edge nodes improves throughput efficiency and reduces retransmissions without modifying the HFC coax plant, DOCSIS PHY/MAC signaling, or cable modems. In contrast to DOCSIS 4.0 FDX upgrades and FTTH migration, the proposed approach is deployable as an incremental modernization step that primarily changes software at the edge. The architecture is suitable for phased rollout (region-by-region) and can be integrated with policy control to support upstream-heavy workloads.

The approach does not increase the physical upstream spectral capacity and remains bounded by DOCSIS PHY constraints. Production deployment requires additional hardening of application-level security and operational integration, and SDN-policy orchestration must be engineered to avoid management overhead.

Conclusions. The study substantiates that upstream performance limitations in DOCSIS-based HFC networks cannot be addressed solely by physical-layer modernization and access distribution changes, and that transport-layer processing placement is a decisive factor for upstream-intensive services. The proposed architecture performs TCP termination at MEC edge nodes using VPP HostStack while keeping the coaxial plant, DOCSIS PHY/MAC operation, and cable modems unchanged, which makes the approach

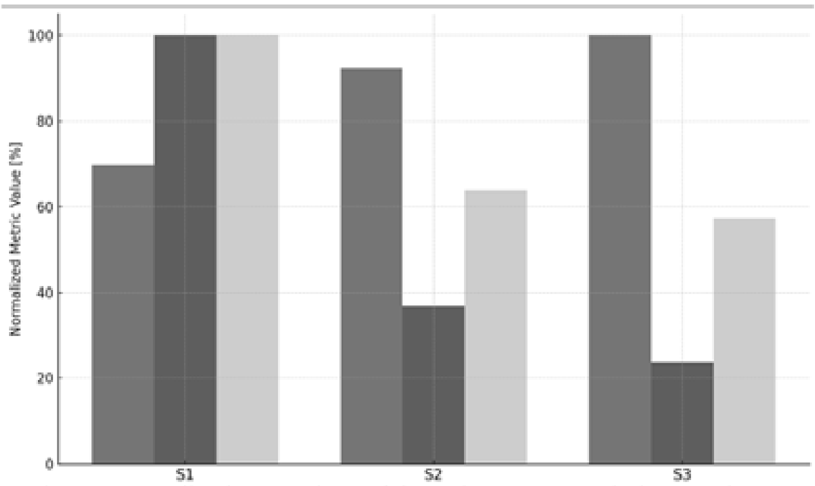

Fig. 3. Normalized comparison of throughput, retransmissions, and CPU utilization across TCP termination scenarios S1–S3

suitable for incremental deployment in existing operator networks.

Experimental validation on the CI-integrated FD.io CSIT 3n-icx testbed with 25-gigabit Ethernet connectivity confirmed stable and reproducible gains relative to a kernel-based baseline. Edge termination with HostStack increased average aggregate TCP throughput by up to 43%, reduced retransmission rates by approximately 75%, and lowered MEC-side CPU utilization by about 40%. These results demonstrate that user-space termination and flow-aware processing at the access edge improves effective utilization of constrained upstream capacity and provides a practical alternative between legacy DOCSIS 3.1 deployments and more disruptive DOCSIS 4.0 or full-fiber upgrades.

Further work will focus on extending edge termination beyond the transport layer toward application-level processing, including protocol-aware handling of encrypted and latency-sensitive traffic patterns. A second direction is to broaden the experimental campaign by evaluating additional traffic mixes, congestion-control configurations, and policy-control strategies to quantify robustness under diverse operational conditions and to refine deployment recommendations for cable operators.

Bibliography:

1. Cisco Systems. Cisco Annual Internet Report: Mobile, Wi-Fi, and Fixed Networks : white paper. 2020. URL: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (дата звернення: 13.01.2026).

2. World Bank. Digital Progress and Trends Report 2023. Washington, DC : World Bank, 2024. DOI: https://doi.org/10.1596/978-1-4648-2049-6 (дата звернення: 13.01.2026).

3. Cable Television Laboratories, Inc. DOCSIS® 4.0 Physical Layer Specification (CM-SP-PHYv4.0). 2022. URL: https://www.cablelabs.com/specifications/CM-SP-PHYv4.0 (дата звернення: 13.01.2026).

4. Cable Television Laboratories, Inc. Remote PHY Specification (CM-SP-R-PHY-I04-160512). 2016. URL: https://account.cablelabs.com/server/alfresco/305841fa-63a7-442d-b08c-c619e10bdd1b (дата звернення: 13.01.2026).

5. Cisco Systems. Cable DAA Network Readiness: A New Operational Model to Drive Efficiency and Business Value. 2019. URL: https://www.cisco.com/c/dam/en/us/solutions/service-provider/industry/cable/pdfs/daa-network-readiness.pdf (дата звернення: 13.01.2026).

6. FD.io Project. Vector Packet Processing (VPP): One Terabit Software Router on Intel® Xeon® Scalable Processor Family Server. 2017. URL: https://fd.io/docs/whitepapers/FDioVPPwhitepaperJuly2017.pdf (дата звернення: 13.01.2026).

7. ETSI MEC ISG. Multi-access Edge Computing (MEC); Framework and Reference Architecture (ETSI GS MEC 003 V2.1.1). 2019. URL: https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/02.01.01_60/gs_mec003v020101p.pdf (дата звернення: 13.01.2026).

8. Kerrisk M. ld.so(8) – Linux Programmer's Manual. URL: https://man7.org/linux/man-pages/man8/ld.so.8.html (дата звернення: 13.01.2026).

9. Mhadhbi A. R., Steger L. Literature Survey: Performance Enhancing Proxies for TCP and QUIC // Proceedings of the Seminar Innovative Internet Technologies and Mobile Communications (IITM). Winter Semester 2023. Munich, Germany : Chair of Network Architectures and Services, Technical University of Munich, NET 2024-04-1, 2024. DOI: https://doi.org/10.2313/NET-2024-04-1_15 (дата звернення: 14.01.2026).

10. Charter Communications. Charter Communications Investor Meeting. 2022. URL: https://ir.charter.com/static-files/3943ccd3-dd07-4dfa-9946-a9afd1bef557 (дата звернення: 13.01.2026).

11. CableLabs. Cable Operator Preparations for DOCSIS 4.0 Technology Deployment (CM-TR-TAR-4.0-V01-230629). 2023. URL: https://www.cablelabs.com/specifications/CM-TR-TAR-4.0 (дата звернення: 13.01.2026).

12. Houtsma V., Mahadevan A., Kaneda N., van Veen D. Transceiver technologies for passive optical networks: past, present, and future [Invited Tutorial]. *Journal of Optical Communications and Networking*. 2021. Vol. 13, no. 1. P. A44–A55. DOI: https://doi.org/10.1364/JOCN.403500 (дата звернення: 13.01.2026).

13. Nyarko-Boateng O., Xedagbui F. E. B., Adekoya A. F., Weyori B. A. Fiber optic deployment challenges and their management in a developing country: A tutorial and case study in Ghana. *Engineering Reports*. 2020. Vol. 2, no. 2. Art. e12121. DOI: https://doi.org/10.1002/eng2.12121 (дата звернення: 13.01.2026).

14. Salinger J., Sigman S. Lessons from Operating Tens of Thousands of Remote PHY Devices. SCTE and NCTA, 2021. URL: https://nctatechnicalpapers.com/wp-content/uploads/2025/09/2021-lessons-from-operating-tens-of-thousands-of-remote-phy-devices.pdf (дата звернення: 13.01.2026).

15. Sowinski P., Smith A., Liu T. Remote PHY 2.0: The Next Steps for Remote PHY Technology. SCTE and NCTA, 2019. URL: https://nctatechnicalpapers.com/wp-content/uploads/2025/09/2019-remote-phy-2-0-the-next-steps-for-remote-phy-technology.pdf (дата звернення: 13.01.2026).

16. Hong G., Martin J., Westall J. M. On fairness and application performance of active queue management in broadband cable networks. *Computer Networks.* 2015. Vol. 91. P. 390–406. DOI: https://doi.org/10.1016/j.comnet.2015.08.018 (дата звернення: 13.01.2026).

17. Hong G., Martin J., Westall J. Adaptive bandwidth binning for bandwidth management. *Computer Networks*. 2019. Vol. 150. P. 150–169. DOI: https://doi.org/10.1016/j.comnet.2018.12.019 (дата звернення: 13.01.2026).

18. Alharbi Z., Thyagaturu A. S., Reisslein M., ElBakoury H., Zheng R. Performance Comparison of R-PHY and R-MACPHY Modular Cable Access Network Architectures. *IEEE Transactions on Broadcasting.* 2018. Vol. 64, no. 1. P. 128–145. DOI: https://doi.org/10.1109/TBC.2017.2711145 (дата звернення: 13.01.2026).

19. Frank T., Konstantynowicz M., Mikus P., Polak V. FD.io CSIT Performance Dashboard : IETF 118 BMWG Working Group slides. 2023. URL: https://datatracker.ietf.org/meeting/118/materials/slides-118-bmwg-9-fdio-csit-performance-dashboard-00 (дата звернення: 13.01.2026).

20. FD.io CSIT Documentation. FD.io Data Center Testbed Specifications (The Linux Foundation). URL: https://csit.fd.io/cdocs/infrastructure/fdio_dc_testbed_specifications/#3-node-icelake-3n-icx (дата звернення: 13.01.2026).

21. ESnet; Lawrence Berkeley National Laboratory. iperf3 : documentation and project homepage. 2025. URL: https://software.es.net/iperf (дата звернення: 14.01.2026).

22. Sasson V., Babiarz J., Morton B. Framework for TCP Throughput Testing : RFC 6349. RFC Editor, 2011. DOI: https://doi.org/10.17487/RFC6349. (дата звернення: 13.01.2026).

23. Nguyen T. A. N., Gangadhar S., Sterbenz J. P. G. Performance Evaluation of TCP Congestion Control Algorithms in Data Center Networks. Матеріали конференції CFI'16. Nanjing, China, 2016. P. 21–29. DOI: https://doi.org/10.1145/2935663.2935669.

24. Open Networking Foundation. SDN Architecture – A Primer. 2016. URL: https://opennetworking.org/wp-content/uploads/2013/05/7-26%20SDN%20Arch%20Glossy.pdf (дата звернення: 13.01.2026).

25. AvidThink; The Linux Foundation. Myth-busting DPDK in 2020 : research brief (Rev. B). 2020. URL: https://nextgeninfra.io/wp-content/uploads/2020/07/AvidThink-Linux-Foundation-Myth-busting-DPDK-in-2020-Research-Brief-REV-B.pdf (дата звернення: 13.01.2026).

26. Robot Framework Foundation. Robot Framework – Generic Test Automation Framework for Acceptance Testing and RPA. URL: https://robotframework.org (дата звернення: 13.01.2026).

27. FD.io. Jenkins (CSIT view): список CI-завдань та результати збірок. URL: https://jenkins.fd.io/view/csit (дата звернення: 14.01.2026).

Іванець І.Д. АРХІТЕКТУРА HFC-МЕРЕЖІ З ТЕРМІНАЦІЄЮ TCP НА ПЕРИФЕРІЙНИХ ВУЗЛАХ

У статті представлено результати дослідження, спрямованого на створення та експериментальну перевірку архітектурного підходу до модернізації висхідного транспорту в гібридних оптично-коаксіальних (HFC) мережах. Зростання частки апстрім-трафіку та поширення сервісів реального часу (відеоконференції, синхронізація великих обсягів даних із хмарою) потребують вищої пропускної здатності та меншої варіативності затримки (джитера), тоді як централізована модель керування апстрімом у DOCSIS із grant-орієнтованим плануванням часто стає обмежувальним чинником. Традиційні стратегії оновлення – розширення частотного ресурсу та можливостей DOCSIS 4.0, повний перехід на волоконно-оптичний доступ (FTTH) або впровадження Distributed Access Architecture – потребують значних капіталовкладень і тривалого циклу розгортання. Запропоновано периферійну архітектуру транспортного рівня, у якій критичні функції обробки трафіку канального, мережевого та транспортного рівнів переносяться з virtual Cable Modem Termination System (vCMTS) на вузли Multi-access Edge Computing (MEC), розміщені поруч із Remote PHY Device (RPD) у зоні доступу HFC. На вузлі MEC інтегровано платформу FD.io Vector Packet Processing із HostStack для термінації TCP у просторі користувача; сумісність із прикладними програмами забезпечено перехопленням POSIX-сокетів на основі LD_PRELOAD без змін у коаксіальній інфраструктурі, DOCSIS PHY/MAC та абонентському обладнанні. Валідацію виконано на тестовому стенді FD.io CSIT Ice Lake (3n-icx) із 25-гігабітними Ethernet-інтерфейсами шляхом порівняння трьох сценаріїв: TCP kernel-to-kernel, HostStack-to-kernel та HostStack-to-HostStack; трафік генерувався інструментом iperf3 із керованою кількістю з'єднань і вибором алгоритмів керування перевантаженням. Експериментальні результати показують, що термінація TCP на периферії з використанням HostStack забезпечує зростання середньої пропускної здатності до 43%, зменшення частки повторних передавань приблизно на 75% і скорочення завантаженості процесора MEC орієнтовно на 40% порівняно з базовою конфігурацією ядра. Отримані результати підтверджують, що програмно реалізована периферійна обробка транспортного рівня підвищує ефективність використання обмеженої висхідної смуги в HFC-мережах і надає операторам масштабований, поетапно впроваджуваний шлях модернізації без негайної заміни коаксіальної ділянки та без повного переходу на DOCSIS 4.0 або FTTH.

Ключові слова: *периферійна термінація, обробка транспортного рівня, гібридні оптично-коаксіальні мережі, векторне опрацювання пакетів, оцінювання продуктивності.*

Дата першого надходження статті до видання: 13.01.2026
Дата прийняття статті до друку після рецензування: 05.02.2026
Дата публікації (оприлюднення) статті: 08.04.2026