МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЧЕРНІГІВСЬКА ПОЛІТЕХНІКА»
WIN-WIN EDIH



Co-funded by
the European Union



Співфінансується
Європейським Союзом



ВІЛЬНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ В ОСВІТІ, НАУЦІ ТА БІЗНЕСІ

FREE/OPEN SOURCE SOFTWARE IN EDUCATION, SCIENCE & BUSINESS (FOSS 2026)

Матеріали II Міжнародної науково-практичної конференції
01–02 жовтня 2026 р.
м. Чернігів

Чернігів 2026

УДК 004.49:[37+001+334.722]
В41

Рекомендовано до друку вченою радою Національного університету «Чернігівська політехніка» (протокол № 12 від 02.11.2026 р.)

Вільне програмне забезпечення в освіті, науці та бізнесі : матеріали ІІ Міжнар. В41 наук.-практ. конф. (м. Чернігів, 01–02 жовтня 2026 р.) / відп. за вип. Д. Б. Мехед. Чернігів : НУ «Чернігівська політехніка», 2026. 156 с.

ISBN 978-617-8584-09-2

Збірник містить матеріали ІІ Міжнародної науково-практичної конференції «Вільне програмне забезпечення в освіті, науці та бізнесі» (FOSS 2026), що відбулася онлайн 01–02 жовтня 2026 року в Національному університеті «Чернігівська політехніка». Розглянуто питання використання вільного та відкритого програмного забезпечення в освіті, науці, бізнесі, ІТ-інфраструктурі, кібербезпеці, аналізі даних і штучному інтелекті. Матеріали подано українською та англійською мовами.

Для науковців, викладачів, здобувачів освіти, фахівців ІТ-галузі та всіх, хто цікавиться вільним програмним забезпеченням.

Матеріали подано в авторській редакції. Відповідальність за зміст, достовірність фактів, цитат і посилань несуть автори.

Конференцію проведено в межах проєкту WIN-WIN EDIH (грантова угода № 101191647), співфінансованого Європейським Союзом у межах програми Digital Europe. Висловлені погляди та думки належать виключно автору(ам) і не обов'язково відображають позицію Європейського Союзу чи Європейської Комісії.

Відповідальний за випуск: Мехед Дмитро Борисович

ISBN 978-617-8584-09-2

© НУ «Чернігівська політехніка», 2026

Чопорова О. В., Чопоров С. В. ВИКОРИСТАННЯ 3D-МОДЕЛЮВАННЯ ТА АЛГОРИТМІВ НЕЙРОННИХ МЕРЕЖ В НАВЧАЛЬНИХ ПРОЄКТАХ ІНЖЕНЕРНО-ТЕХНІЧНОГО СПРЯМУВАННЯ / USE OF 3D MODELING AND NEURAL NETWORK ALGORITHMS IN EDUCATIONAL PROJECTS OF ENGINEERING AND TECHNICAL DIRECTION 82

Чуб М. В. ВИЯВЛЕННЯ ТА ПРОТИДІЯ АДАПТИВНИМ ВЕБ-АТАКАМ ЗА ДОПОМОГОЮ МОДЕЛЕЙ ГЛИБОКОГО НАВЧАННЯ 86

Шевченко Ю. О. ПОРІВНЯЛЬНЕ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ POSTGRESQL ТА INFLUXDB ДЛЯ ЗБЕРІГАННЯ ДАНИХ ФІЗИЧНОЇ АКТИВНОСТІ / COMPARATIVE STUDY OF POSTGRESQL AND INFLUXDB PERFORMANCE FOR STORING PHYSICAL ACTIVITY DATA 89

Al-Akwaa K. H. M. A. OPEN-SOURCE DATA SCIENCE FOR SUSTAINABLE DEVELOPMENT: A REPRODUCIBLE ANALYTICAL FRAMEWORK 93

Borysenko O. A. AN ARCHITECTURAL APPROACH TO LLM-BASED ROOT CAUSE ANALYSIS FOR CONTAINER ORCHESTRATION INCIDENTS 100

Burmaka I. LOCAL, OPEN, AND ENERGY-EFFICIENT: HARNESSING X86 NPUS FOR EVERYDAY AI TASKS 102

Byvoino T. P., Byvoino P. G. SIMULATION MODELING AND STATISTICAL ANALYSIS OF THE EFFICIENCY OF AN EMERGENCY MEDICAL SERVICE STATION 106

Didusenko O. A., Bula Y. I., Shymchuk G. V. REPRODUCIBLE FOSS FRAMEWORK FOR RESILIENCE EVALUATION OF CONTAINERIZED DISTRIBUTED SYSTEMS 109

Didusenko O. A., Bula Y. I., Shymchuk G. V. REPRODUCIBLE FOSS TESTBED FOR FAULT-TOLERANT MULTIPATH ROUTING IN IP NETWORKS 112

Ebirim V., Jadama A. F. R. RESILIENT DIGITAL EDUCATION IN DISRUPTED ENVIRONMENTS: THE CCDS APPROACH 115

Fraya DIGITAL TRANSFORMATION OF BUSINESS THROUGH FREE AND OPEN SOURCE SOFTWARE: OPPORTUNITIES, BARRIERS, AND STRATEGIC PATHWAYS 119

Kala P. RUN MONGODB YOUR WAY: A PRACTICAL OPEN-SOURCE STACK FOR SELF-MANAGED DEPLOYMENTS 121

Lukin O. A MODERN C++ KMIP CLIENT BASED ON A REUSABLE PROTOCOL LIBRARY 124

Mekhed D. B. DEVELOPING STUDENTS' CYBER HYGIENE SKILLS THROUGH THE USE OF FREE SPECIALIZED SOFTWARE 127

Nahornyi B. I., Lysenok D. V., Savenko A. O. AN OPEN BROWSER-BASED CO-SIMULATOR OF STM32 FIRMWARE AND ELECTRICAL CIRCUITS AS A FREE ALTERNATIVE TO PROTEUS 130

Romanenko M. V. FOSS APPROACH IN MOBILE DEVELOPMENT: CREATING AN OPEN-SOURCE ANDROID APPLICATION WITH THE INTEGRATION OF READY-MADE LIBRARIES FOR STYLISTIC RECOMMENDATIONS 134

Shevchenko N. A., Kibitkin D. S., Shymchuk G. V. REPRODUCIBLE OPEN-SOURCE BENCHMARK OF CONCEPT-DRIFT DETECTORS FOR STREAMING REGRESSION 137

Shevchenko N. A., Kibitkin D. S., Shymchuk G. V. INTERPRETABLE OPEN-SOURCE MACHINE LEARNING FOR NATURAL GAS CONSUMPTION FORECASTING WITH WEATHER AND LAG FEATURES 140

Shivashankar C. ARCHITECTING DATABASE DEPLOYMENTS ON KUBERNETES: PATTERNS, TRADE-OFFS AND EXPERIENCES 143

Shymchuk G. V., Yatskiv P. O. DETECTION OF ANOMALOUS STATES IN IP NETWORKS USING MACHINE LEARNING AND OPEN-SOURCE SOFTWARE TOOLS 146

Shymchuk G. V., Yatskiv P. O. OPEN-SOURCE SOFTWARE ENVIRONMENT FOR MODELING FAULT-TOLERANT MULTIPATH ROUTING IN IP NETWORKS 149

Silva F. HOSTING SOMEONE ELSE'S APP: IFRAME CHOREOGRAPHY IN PMM 152

Šupić A. ZERO-DOWNTIME MIGRATION WITH PERCONA CLUSTERSYNC FOR MONGODB 154

Didusenko O. A., Bula Y. I., Shymchuk G. V.
Ternopil Ivan Puluj National Technical University, Department of Computer Science, Ternopil, Ukraine
Corresponding author: Grigorii Shymchuk, e-mail: gorych@gmail.com, ORCID: 0000-0003-2362-7386

REPRODUCIBLE FOSS TESTBED FOR FAULT-TOLERANT MULTIPATH ROUTING IN IP NETWORKS

Abstract. A reproducible open-source framework for studying fault-tolerant multipath routing in IP networks is proposed. The target testbed combines Mininet, Open vSwitch, FRRouting, iperf3, and Python so that topology, routing policy, failure injection, and metric collection can be reproduced without proprietary network equipment. A pilot model-based experiment with 50 failure episodes per strategy compared conventional single-path OSPF reconvergence, ECMP re-hashing, a precomputed backup path, and a health-aware multipath policy. The median modeled recovery time decreased from 2410.6 ms for the single-path baseline to 61.7 ms for health-aware multipath; mean packet loss decreased from 2.703% to 0.109%. The preliminary results do not substitute for packet-level emulation, but they define measurable hypotheses and a reproducible FOSS workflow for subsequent Mininet/FRRouting validation.

Keywords: open-source software; IP networks; multipath routing; fault tolerance; ECMP; FRRouting; Mininet; Open vSwitch.

Introduction

IP networks are expected to preserve connectivity despite link, interface, or router failures. Traditional shortest-path routing can restore reachability after control-plane reconvergence, but the interruption interval may still be relevant for interactive, industrial, and distributed services. The IP Fast Reroute framework therefore distinguishes mechanisms that precompute repair paths from ordinary post-failure convergence; equal-cost multipath (ECMP) and loop-free alternatives are among the basic repair options [1]. Resiliency mechanisms also emphasize path diversity and the use of disjoint paths so that a single failure does not affect all forwarding alternatives [2].

Recent research continues to treat multipath forwarding as both a resilience and traffic-engineering problem. Zhang and Geng showed that providing multiple routing choices can improve network availability [3], while programmable ECMP work demonstrated that conventional hash-based ECMP may be insufficient when a flow must be moved quickly away from a problematic path [4]. At the implementation level, FRRouting supports ECMP and weighted ECMP mechanisms, and Open vSwitch provides flow-based multipath actions [5, 6]. These components can be combined with Mininet, which runs real Linux networking and switch/application code in a virtual topology [7].

The objective of this work is to define a reproducible FOSS testbed and a quantitative evaluation procedure for comparing fault-recovery strategies in an IP network. The present contribution is deliberately positioned as a pilot stage: recovery metrics are first checked in a transparent stochastic model, after which the same scenarios are intended to be reproduced with real TCP/UDP flows in Mininet and FRRouting.

Materials and Methods

The planned software stack contains Mininet for topology orchestration, Open vSwitch for forwarding, FRRouting for OSPF/BGP control-plane functions, iperf3 for TCP/UDP traffic generation, and Python for experiment control and analysis. The current iperf3 project released version 3.22 on 29 September 2026, immediately before preparation of this work [8]. Figure 1 shows the compact topology used for the pilot design. Three internally separated paths connect R1 and R5, allowing link failures to be introduced without removing all end-to-end connectivity.

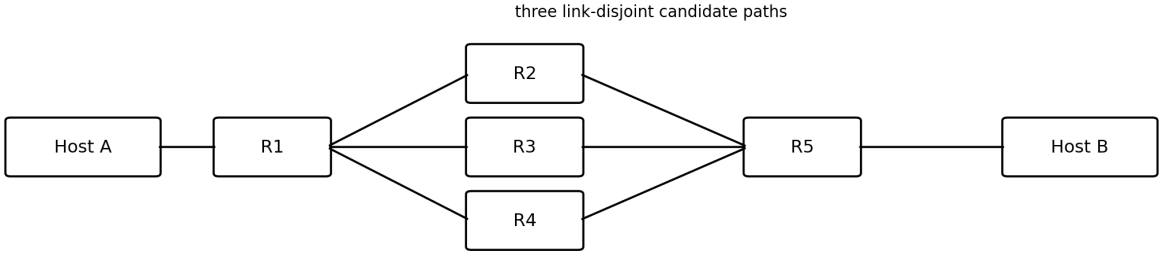

Figure 1 – Multipath topology for the reproducible FOSS testbed

Four strategies were compared: S1 – ordinary single-path forwarding followed by OSPF reconvergence; S2 – ECMP with flow re-hashing after a failed member is removed; S3 – a precomputed primary/backup path; and S4 – a health-aware multipath policy that ranks available next hops by recent reachability and delay. For the pilot experiment, 50 independent failure episodes were generated for each strategy with a fixed random seed. The model included failure-detection delay, route activation delay, transient packet loss, and post-recovery throughput retention. The numerical values are therefore model-based and are used to verify the comparison protocol rather than to claim production-network timing.

Table 1 – Failure scenarios and measured indicators

Scenario	Injected event	Primary metric	Secondary metrics
F1	Failure of one path link	Recovery time	Loss, throughput
F2	Failure near ingress router	Recovery time	Availability, RTT
F3	Failure under active load	Packet loss	Recovery, throughput
F4	Repeated path degradation	Stability	Switches, loss

Recovery time was defined as the interval between the injected failure and restoration of successful end-to-end forwarding:

$$T_{rec} = t_{restore} - t_{failure}. \quad (1)$$

Packet loss ratio was calculated as

$$PLR = \frac{N_{lost}}{N_{sent}} * 100\%. \quad (2)$$

For a 60-s observation window, service availability was estimated as $A = \frac{T_{total} - T_{down}}{T_{total}} * 100\%$, where T_{down} is the modeled outage interval. Throughput retention was the ratio of mean post-failure throughput to the no-failure baseline. Median and 95th-percentile recovery time were reported because recovery distributions are asymmetric.

Results

The pilot results are summarized in Table 2. All multipath strategies reduced the modeled interruption relative to ordinary single-path convergence. ECMP decreased median recovery time by approximately one order of magnitude, while precomputed and health-aware alternatives reduced the interruption further. The health-aware policy achieved a median recovery time of 61.7 ms and a 95th percentile of 96.0 ms. In the same model, mean packet loss was 0.109% and throughput retention was 98.36%.

Table 2 – Pilot model-based fault-recovery results

Method	Median Trec, ms	P95 Trec, ms	Loss, %	Availability, %	Throughput retention, %
OSPF single path	2410.6	2744.9	2.703	96.030	83.94
ECMP re-hash	303.1	445.3	0.538	99.479	94.00
Precomputed backup	102.7	138.3	0.182	99.836	97.35
Health-aware multipath	61.7	96.0	0.109	99.897	98.36

Discussion

The results support the experimental hypothesis that path diversity is useful only when an alternative can be activated quickly. Single-path routing preserves eventual reachability but depends on failure detection and control-plane convergence. ECMP already maintains several equal-cost next hops and can therefore shorten interruption, yet flow hashing does not by itself guarantee precise movement away from a degraded path. This limitation is consistent with recent work on programmable ECMP [4]. Precomputed backup and health-aware selection reduce the modeled recovery interval because the forwarding alternative is available before full reconvergence.

The main contribution is reproducibility rather than a claim about absolute timing. Every element of the planned testbed is open-source: Mininet can instantiate the topology, Open vSwitch can implement multipath forwarding, FRRouting can provide dynamic routing, iperf3 can generate measured TCP/UDP load, and Python can inject faults and aggregate results. The same

configuration files and random seeds can therefore be shared and repeated in teaching laboratories or research environments without specialized routers.

The limitation is that Table 2 is obtained from a controlled model, not from packet-level execution of FRRouting daemons. Absolute recovery values must not be transferred to production networks. The next validation stage should execute the four strategies under identical Mininet topologies, vary link delay and offered load, repeat each failure at least 30 times, and compare measured confidence intervals. Additional experiments should consider simultaneous failures and unequal-cost paths.

Conclusions

A reproducible FOSS methodology for evaluating fault-tolerant multipath routing was developed. The proposed stack combines Mininet, Open vSwitch, FRRouting, iperf3, and Python and defines common failure scenarios and metrics. In the pilot model, moving from ordinary single-path convergence to health-aware multipath reduced median recovery time from 2410.6 ms to 61.7 ms and packet loss from 2.703% to 0.109%. These values are preliminary, but they provide explicit hypotheses for subsequent emulation.

Future work will replace the stochastic timing model with automated Mininet/FRRouting experiments, evaluate path-disjointness under several topologies, and study adaptive weighting that jointly considers reachability, delay, and link utilization.

References

1. Shand M., Bryant S. IP Fast Reroute Framework. RFC 5714, 2010. https://datatracker.ietf.org/doc/html/rfc5714.
2. Filsfils C. et al. Resiliency Use Cases in Source Packet Routing in Networking (SPRING) Networks. RFC 8355, 2018. https://datatracker.ietf.org/doc/html/rfc8355.
3. Zhang W., Geng H. Improving Network Availability through Optimized Multipath Routing and Incremental Deployment Strategies. Computers, Materials & Continua, 2024, 80(1), 427-448. https://doi.org/10.32604/cmc.2024.051871.
4. Liu Y., Xiao Y., Zhang X., Dang W., Liu H., Li X., He Z., Wang J., Kuzmanovic A., Chen A., Miao C. Unlocking ECMP Programmability for Precise Traffic Control. 22nd USENIX NSDI, 2025, pp. 87-106.
5. FRRouting Project. BGP Documentation: Weighted ECMP Using BGP Link Bandwidth. FRRouting 10.2 documentation. https://docs.frrouting.org/en/stable-10.2/bgp.html (accessed 01.09.2026).
6. Open vSwitch Project. Open vSwitch Documentation: Multipath Action. https://docs.openvswitch.org/ (accessed 01.09.2026).
7. Mininet Project. Mininet: An Instant Virtual Network on Your Laptop. https://mininet.org/ (accessed 01.09.2026).
8. ESnet. iperf3 Project News: iperf 3.22 Released, 29 September 2026. https://software.es.net/iperf/news.html (accessed 01.09.2026).