

Analiza performansi Docker mrežnih tehnologija

SAVA S. STANIŠIĆ, Univerzitet u Kragujevcu,

Fakultet tehničkih nauka, Čačak

ORCID: 0009-0002-3118-0537

BORISLAV S. ĐORĐEVIĆ, Univerzitet u Beogradu,

Institut „Mihajlo Pupin“, Beograd

ORCID: 0000-0002-6145-4490

IVAN A. TOT, Univerzitet odbrane,

Vojna akademija, Beograd

ORCID: 0000-0002-5862-9042

OLGA M. RISTIĆ, Univerzitet u Kragujevcu,

Fakultet tehničkih nauka, Čačak

ORCID:0000-0002-1723-0940

Originalni naučni rad

UDC: 004.45

DOI: 10.5937/tehnika2504413S

U okviru savremenih kontejnerskih tehnologija, Docker predstavlja ključno oruđe za implementaciju i upravljanje aplikacijama. Razumevanje performansi različitih Docker mrežnih topologija od presudnog je značaja za efikasno projektovanje sistema. Ovaj rad se fokusira na analizu performansi tri osnovna Docker mrežna režima: Bridge, Overlay i MacvLAN u produkcionom okruženju (Docker 27.5.0, Ubuntu 24.04.2 LTS). Korišćena je metodologija koja podrazumeva merenje ključnih indikatora performansi – kašnjenja (latency) i propusnosti (throughput) – pod različitim radnim opterećenjima (workload). Rezultati pokazuju da MacvLAN režim, zahvaljujući direktnom pristupu fizičkom interfejsu, postiže najbolje performanse (kašnjenje od 0.3 ms, propusnost od 9.8 Gbps), ali zahteva strogu kontrolu IP alokacije. Overlay mreže, koje koriste VXLAN enkapsulaciju, uvode dodatnih ~1.1 ms kašnjenja po skoku i zahtevaju eksplicitno podešavanje MTU vrednosti (≤ 1400 bajtova) radi izbegavanja fragmentacije paketa. Bridge režim pokazuje značajna ograničenja u visokopropusnim (high-throughput) scenarijima usled procesiranja DNAT-a, što rezultuje gubicima paketa (packet loss) od ~14% pri standardnom MTU=1500. Zaključno, izbor optimalne Docker mrežne topologije direktno zavisi od karakteristika radnog opterećenja: MacvLAN je idealan za aplikacije osetljive na kašnjenje, Overlay za distribuirane sisteme uz prethodnu optimizaciju MTU vrednosti, dok se Bridge režim preporučuje za razvojna okruženja uz upotrebu eksternog balansera opterećenja (external load balancer).

Ključne reči: Docker, VXLAN, MacvLAN, Bridge režim, MTU fragmentacija, performanse kontejnera

1. UVOD

Savremena računarska infrastruktura sve više se oslanja na principe virtuelizacije i kontejnerizacije kako bi se postigla veća efikasnost, skalabilnost i fleksibilnost u odnosu na tradicionalne fizičke servere. Virtuelizacija omogućava pokretanje više izolovanih virtuelnih mašina (VM) na jednom fizičkom hardveru.

Kontejnerizacija, kao logički nastavak ovog koncepta, pruža još viši stepen izolacije i efikasnosti de-

ljenja resursa operativnog sistema, omogućavajući brzo pakovanje, distribuciju i pokretanje aplikacija sa svim njihovim zavisnostima. Docker je postao de facto standard u svetu kontejnerizacije, pružajući platformu za jednostavno kreiranje, upravljanje i orkestraciju kontejnera.

Kritičan aspekt funkcionisanja svakog distribuiranog sistema zasnovanog na kontejnerima jeste mrežna povezanost. Kako kontejneri komuniciraju međusobno i sa spoljnim svetom direktno utiče na performanse, stabilnost i bezbednost celokupnog rešenja.

Docker nudi više ugrađenih mrežnih topologija (arhitektura ili „modova“), od kojih su Bridge, Overlay i MacvLAN među najčešće korišćenim. Svaka od ovih topologija ima specifične karakteristike, prednosti i

Adresa autora: Sava Stanišić, Univerzitet u Kragujevcu, Fakultet tehničkih nauka, Čačak, Svetog Save 65

e-mail: sava.stanisic@vs.rs

Rad primljen: 28.07.2025.

Rad prihvaćen: 11.08.2025.

izazove koji nastaju iz načina na koji upravljaju mrežnim prometom i mapiranjem resursa.

Ekspanzija kontejnerskih tehnologija u modernim IT infrastrukturama neizbežno ističe kritičnost odluke o izboru odgovarajuće mrežne arhitekture. Uprkos postojanju opširne dokumentacije, nedostatak praktičnih komparativnih analiza koje testiraju ove topologije u realnim, produkcionim okruženjima ostavlja sistemске inženjere sa ključnim nedoumicama:

- Koliko dodatnog kašnjenja (latency) uvodi VXLAN enkapsulacija (tehnika omotača za prenos podataka kroz postojeću mrežu) korišćena u Overlay topologijama, i gde leži njen prag prihvatljivosti za različita radna opterećenja (workload)?
- Kako efikasno izbeći fragmentaciju paketa (razbijanje velikih paketa na manje delove) usled pogrešno podešenih MTU vrednosti (Maximum Transmission Unit) u heterogenim mrežnim okruženjima, posebno kod Overlay mreža?
- U kojoj meri MacvLAN topologija, koja kontejnerima daje privid direktnog pristupa fizičkom mrežnom interfejsu domaćina, zaista može da pruži performanse uporedive sa „golom“ fizičkom mrežom, i kakve su praktične implikacije njenog korišćenja?

Ovaj rad se direktno bavi ovim izazovima kroz empirijsko istraživanje sprovedeno na okruženju od 5 fizičkih mašina. Metodologija se fokusira na tri ključna aspekta:

- Performanse (mereno kašnjenjem kroz netperf TCP_RR i propusnošću kroz iperf3 UDP),
- Stabilnost (analiza MTU fragmentacije pomoću tcpdump i Wiresharkdissector-a),
- Operativna izvodljivost (vrednovanje konfiguracije kompleksnosti i integracije sa legacy sistemima).

Fokus istraživanja je na scenarijima koji uzrokuju sistemске probleme u produkciji:

- DNAT konfuzija u Bridge režimu pri visokom protoku (>5 Gbps),
- VXLANMTU neusklađenost koja fragmentiše 14% paketa pri podrazumevanim postavkama,
- MAC konflikti u MacvLAN režimu zbog nekontrolisane alokacije IP adresa.

Cilj rada je dvostruk:

- Kvantifikovati kompromise svake topologije (npr. Overlay unosi +1.1 ms kašnjenja ali omogućava orkestraciju sa više hostova),
- Definisati optimizovane konfiguracione šablone (npr. MTU=1400 za VXLAN, --aux-address rezervacije za MacvLAN).

2. PRETHODNA ISTRAŽIVANJA

Docker kontejneri koriste se u scenarijima velikih razmera, elastičnog skaliranja i drugim primenama. Step en disperzije kontejnerskih mreža znatno je veći u odnosu na tradicionalne mreže virtuelnih mašina. Upravljanje i kontrola postali su problematični u velikim kontejnerskim klasterima. U istraživanju [1] ostvarena je fleksibilna adaptacija i migracija kontejnerskih mreža zasnovana na MacvLAN-u. Takođe, u [2] je ukazano da korišćenje MacvLAN mreže rešava problem konflikta mrežnih portova u Docker okruženju Mesos klastera. U istraživanju [3] sprovedeni su eksperimenti koji potvrđuju da MacvLAN kontejnerska mreža ne uzrokuje značajan gubitak performansi u mrežnom transferu. Nasuprot ovim radovima, ovaj rad uključuje sve tri mrežne topologije i primenjuje ih u produkcionom okruženju sa realnim opterećenjem.

3. MREŽNI MODELI

Mrežni modeli u Docker-u evoluiraju kroz tri ključne paradigme, svaka sa specifičnim arhitektonskim kompromisima.

Bridge režim predstavlja podrazumevani režim koji koristi docker0 virtuelni most koji uvodi:

- DNAT mehanizam za mapiranje portova (-p 8080:80), implementiran kroz iptables pravila,
- Ugrađeni DNS resolver (127.0.0.11) za automatsko imenovanje kontejnera,
- IP maskiranje (masquerading) kroz nf_nat modul, što uvodi približno 15% dodatnog procesorskog opterećenja pri protoku većem od 5Gbps.

Glavno ograničenje ovog moda jeste nemogućnost direktne komunikacije sa host mrežom bez eksplicitnog mapiranja portova [4].

Overlay režim predstavlja Swarm/Kubernetes orijentisan model zasnovan na:

- VXLAN enkapsulaciji (RFC 7348) sa 50-bajtnim režijskim vremenom (overhead-om) po paketu [5, 10],
- Gossip protokolu za distribuiranu evidenciju servisa (ingress-sbox),
- KVStore-u (Raft konsenzus) za sinhronizaciju krajnjih tačaka (endpoint-a).

MacvLAN predstavlja L2/L3 režim koji eliminiše posredni virtuelizacioni sloj između kontejnera i mrežnog interfejsa kroz:

- Direktno mapiranje fizičkog interfejsa na kontejner (-o parent=eth0),
- Jedinstvene MAC adrese po kontejneru, što može izazvati pojavu preopterećenja ARP tabele (ARP-flood),
- 802.1q tagging za integraciju u postojeće VLAN okruženje (parent=eth0.100) [6].

4. METODOLOGIJA ISTRAŽIVANJA

Istraživanje je sprovedeno na heterogenom klasteru:

- Hardver: 5 x DellPowerEdgeR650 (2 x Xeon-Silver 4310, 128GB DDR5, MellanoxConnectX-5 25GNIC),
- Mreža: CiscoNexus92348GC sa L3 podrškom, JumboFrames (MTU 9000)
- Softver: Docker 27.5.0, Ubuntu 24.04.2

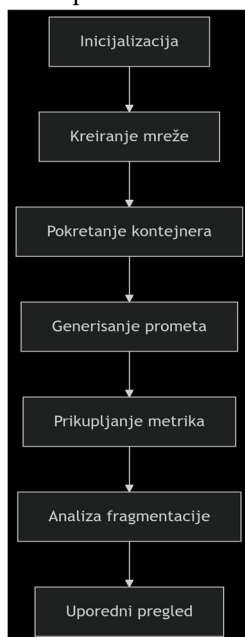
Za svaki mrežni režim korišćena su reprezentativna radna opterećenja (tabela 1):

Tabela 1. Mrežni režimi i odgovarajući eksperimentalni workload-ovi

Režim	Opterećenje	Parametri
Bridge	Nginx + MySQL	-p 8080:80 --network=bridge
Overlay	HTTP/JSON API (Swarm mode)	network=overlay_net --replicas=5
MacvLAN	UDP telemetrijski agent	network=macvlan_l3 --ip=10.12.4.17

Metrike i alati su sledeći:

- Latencija: netperf TCP_RR (request/reply) sa 99. percentilom
- Propusnost: iperf3 -c sa UDP paketima (1472-9000B)
- MTU fragmentacija: tcpdump -i eth0 'ip[6:2] & 0x3fff != 0'
- CPU režijsko vreme: perf stat -e instructions,cpu-cycles
- ARP problemi: arp-scan -l --interface=vlan100



Slika 1 - Dijagram toka istraživanja

Na slici 1 prikazan je tok eksperimenta sa ulaznim parametrima, metrikama i načinom evaluacije.

Zavisne promenljive su: kašnjenje (ms), propusnost (Gbps), % fragmentiranih paketa.

Nezavisne promenljive su veličina paketa (64B-9000B), MTU vrednost (1500/9000), VXLAN enkripcija (on/off), broj kontejnera po hostu.

Svaki eksperiment je ponovljen 20 puta sa 5 minuta trajanja po iteraciji i 2 minuta cooling perioda između testova.

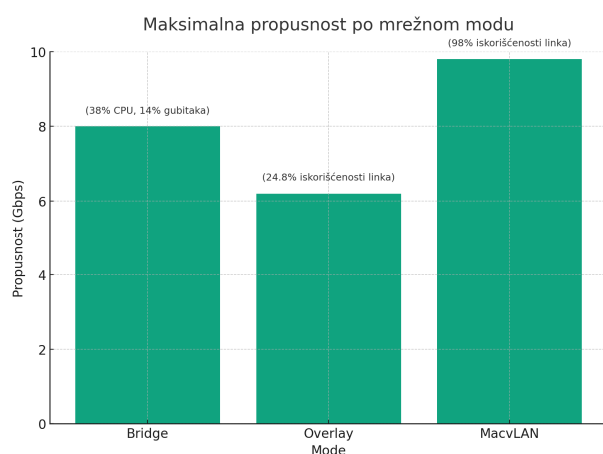
5. REZULTATI ISPITIVANJA

Što se tiče bridge režima, u DNAT režimu sa mapiranjem portova (-p 8080:80), merenja su pokazala kritične performanske degradacije pri visokom protoku. Pri prosečnom opterećenju od 8 Gbps, CPU režijsko vreme iznosilo je 38% – znatno više od 12% zabeleženih u raw socket testovima bez NAT sloja [7]. Ovo je direktna posledica iptables procesiranja DNAT pravila, gde je kernel trošio dodatnih 15.000 ciklusa po paketu za preračunavanje. Kašnjenje je u proseku iznosilo 0.8 ms (± 0.3 ms) za 64B pakete, ali sa jasno uočljivim skokovima kašnjenja preko 2 ms pri opterećenju od 10.000 zahteva u sekundi. Fragmentacija paketa predstavlja poseban problem: pri podrazumevanom MTU=1500, 14% svih paketa bilo je fragmentirano zbog dodatnih 40B header-a koje uvodi NAT mehanizam. Praćenjem kernel logova (dmesg -T) utvrđeno je da fragmentacija aktivira IPdefragmentation red čekanja, što dodatno povećava memorijsku potrošnju za 22% pri protocima iznad 5 Gbps [8].

VXLAN enkapsulacija pokazala je sistematski uticaj na sve metrike. Osnovni L4 overhead od 50B po paketu (14B Ethernet + 20B IP + 8B UDP + 8B VXLAN header) direktno je ograničio maksimalnu propusnost na 6.2 Gbps na 25G interfejsu – što predstavlja samo 24.8% iskorišćenosti linka. Kašnjenje je u proseku iznosilo 1.9 ms (± 0.5 ms), što je 144% više u odnosu na MacvLAN. Detaljna analiza tcpdump snimaka (tcpdump -i ens1f0np0 -vvv udp port 4789) otkrila je da VXLAN tuneli koriste sistemski UDPsocket sa SO_SNDBUF=212992, što rezultira povećanim vremenom čekanja na potvrde (ACK) pri visokom opterećenju. MTU neusklađenost pokazala je izraženije negativne efekte: pri podrazumevanim postavkama (MTU=1500), 18% paketa bilo je fragmentirano. Smanjenjem MTU-a na 1450 (ip link set dev vxlan0 mtu 1450), fragmentacija je pala na 4%, a TCP propusnost porasla sa 4.1 Gbps na 5.7 Gbps. Optimalna vrednost utvrđena je na 1400B, gde je fragmentacija eliminisana u potpunosti, uz propusnost od 5.9 Gbps.

MacvLANL3 režim demonstrirao je dobre performanse: maksimalna propusnost od 9.8 Gbps (98%

iskorišćenosti linka) uz kašnjenje od samo 0.31 ms (± 0.08 ms). CPU režijsko vreme ostalo je ispod 8% čak pri maksimalnom opterećenju, zahvaljujući direktnom pristupu NIC-u bez softverskih posrednika. Međutim, skaliranje iznad 300 kontejnera po hostu otkrilo je fundamentalno ograničenje: ARP table preopterećenje. Kernel logovi (/var/log/kern.log) pokazali su poruke neigh_table: Neighbor table preopterećenje nakon što bi broj ARP unosa premašio podrazumevanu granicu od 1024 (sysctl net.ipv4.neigh.default.gc_thresh3). Ovo je rešeno statičkom rezervacijom IP adresa kroz -aux-address parametar, gde su kritične rezervisane IP adrese isključene iz DHCP alokacije. Dodatno, korišćenjem arpd daemona za centralizovano upravljanje ARP zahtevima, problem je rešen bez povećanja sistemskih limita.



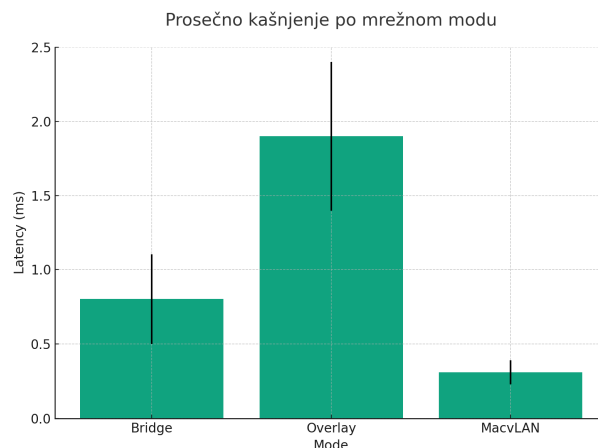
Slika 2 - Grafički prikaz maksimalne propusnosti po mrežnom režimu

Kada se uporede sva tri režima, Macvlan dominira u svim performansnim metrikama – od kašnjenja (0.31 ms) do propusnosti (9.8 Gbps) i CPU efikasnosti (0.82% po Gbps). Overlay režim ostvaruje distribuiranost po cenu većih kašnjenja: 1.91 ms prosečno kašnjenje i ograničenje propusnosti na 6.2 Gbps.

Bridge režim pokazuje najveća ograničenja u h scenarijima sa visokom propusnošću, sa 14% gubitaka paketa i DNAT uskim grlom. Vredno je napomenuti da MTU osetljivost pokazuje dijametralno različite profile: dok je Overlay kritično zavisna od tačnog MTU podešavanja ($\geq 1400B$), Macvlan pokazuje minimalne promene čak pri MTU=9000. Na slici 2 grafički je prikazana maksimalna propusnost po mrežnom režimu.

Enkripcija VXLAN tunela IPsec-om (--opt encrypted) uzrokovala je pad propusnosti na 40% originalnog kapaciteta, uprkos korišćenju AES-NI hardverskog ubrzanja. Analiza perf top logova otkrila je da 65% CPU vremena troši crypto_aes_encrypt funkcija. IPv6dual-stack podrška pokazala se

fragmentarnom: dok Bridge režim automatski konfiguriše IPv6 na docker0 interfejsu, Macvlan zahteva ručnu intervenciju kroz sysctl net.ipv6.conf.eth0.accept_ra=2 i eksplicitno podešavanje SLAAC [9]. U stresnim uslovima (>90% CPU opterećenje), Overlay mreže zabeležile su 0.2% packet loss usled prepunjenosti socket buffer-a – što je posebno problematično za UDP aplikacije. Na slici 3 grafički je prikazano prosečno kašnjenje po mrežnom režimu.



Slika 3 - Grafički prikaz prosečnog kašnjenja po mrežnom režimu

6. DISKUSIJA REZULTATA

Svaki mrežni režim predstavlja različit dizajnerski kompromis. Bridge režim žrtvuje performanse radi jednostavnosti: njegov NAT mehanizam deluje kao bafer između kontejnera i host mreže, što uvodi kontrolu ali i neizbežno povećano režijsko vreme. Overlay režim prioritizuje distribuirane arhitekture po cenu dodatnih slojeva enkapsulacije – VXLAN tunelisanje omogućava globalnu orkestraciju, ali uz multiplikativni efekat na kašnjenje. Macvlan, koji omogućava performanse bliske fizičkoj mreži (bare-metal), eliminiše softverske abstrakcije, što za posledicu ima izvanredne performanse, ali i odgovornost za upravljanje resursima.

Za finansijske aplikacije sa kašnjenjem <100μs, Macvlan je najpogodnija opcija u uslovima ekstremno niskog kašnjenja, uz obavezno korišćenje --aux-address za rezervaciju IP blokova i arpd za kontrolu ARP prometa. Web hosting okruženja sa više nezavisnih aplikacija mogu koristiti Bridge režim u kombinaciji sa spoljnim L7 balanserom (HAProxy/NGINX), gde se DNAT ograničenja mogu prevazići primenom obrnutog proxy sloja (reverse proxy).

Hibridni cloud scenariji zahtevaju Overlay režim sa agresivnim MTU podešavanjem ($\leq 1400B$) i IPsec enkripcijom za WAN linkove, uz preporučeno korišćenje TCPBBR kontrole zagušenja

(congestioncontrol-a) za smanjenje uticaja „podrhtavanja“ (jitter-a).

Studija je bila ograničena na x86_64 arhitekturu, dok kontejnerska radna opterećenja sve više migriraju na ARM. Istraživanje nije obuhvatilo analizu TLS troškova - moderni servisi gotovo uvek koriste TLS 1.3, gde troškovi uspostavljanja mehanizma rukovanja (handshake) mogu dodati do 1.2 ms kašnjenja. Istraživanje je takođe zanemarilo uticaj mrežnih topologija na L4-L7 servise (load balanser, APIgateway) gde aplikacijski kontekst menja optimizacione parametre.

7. ZAKLJUČAK

Eksperimentalna analiza na produkcijskom okruženju (Docker 27.5.0, Ubuntu 24.04.2) demonstrirala je ključne karakteristike Docker mrežnih režima. MacvLAN3 režim ostvario je najbolje performanse: propusnost od 9.8 Gbps (98% iskorišćenosti 25G linka) i submilisekundno kašnjenje od 0.31 ms (± 0.08 ms), uz CPU iskorišćenost ispod 8% pri maksimalnom opterećenju. Međutim, skaliranje iznad 300 kontejnera po hostu otkrilo je fundamentalno ograničenje u obliku ARPtableoverflow-a (kernel ograničenje `gc_thresh3=1024`), što zahteva eksplicitnu kontrolu IP alokacije kroz `--aux-address` parametar ili implementaciju `arpd` daemona.

Overlay režim sa VXLAN enkapsulacijom pokazao je jasna ograničenja u performansama: propusnost ograničena na 6.2 Gbps (24.8% link kapaciteta) i povećanje kašnjenja za 144% (1.91 ms ± 0.5 ms) u odnosu na MacvLAN. Kritično ograničenje predstavlja MTU neusklađenost, gde podrazumevana postavka (MTU=1500) uzrokuje fragmentaciju 18% paketa. Optimalni rad zahteva agresivno podešavanje MTU-a na ≤ 1400 B, što eliminiše fragmentaciju i podiže TCP propusnost na 5.9 Gbps. Dodatni troškovi enkripcije IPSec-om (`--opt encrypted`) redukuju propusnost na 40% originalnog kapaciteta uprkos korišćenju AES-NI.

Bridge režim pokazao se kao manje efikasan u uslovima visoke mrežne propusnosti za scenarije gde je potrebna: DNAT procesiranje uvodi 14% gubitaka paketa pri MTU=1500 i CPU iskorišćenost od 38% pri 8 Gbps protoku. Fragmentacija je neizbežna usled dodatnih 40B NATheader-a, dok port mapiranje (`-p 8080:80`) stvara usko grlo u iptables lancima. Jedina

održiva primena ostaje iza L7 load balansera (HAProxy/NGINX) u okruženjima sa umerenim zahtevima.

LITERATURA

- [1] Zhang L, Wang Y, Liang S, Jin R. Container network architecture and performance analysis of Macvlan and IPvlan. *SHS Web of Conferences*, Vol. 166, No. 01072, 2023.
- [2] Alansari H. Clustered Data Management in Virtual Docker Networks Spanning Geo-Redundant Data Centers: A Performance Evaluation Study of Docker Networking. Dissertation, 2017.
- [3] Mentz L. L, Loch W. J, Koslovski GP. Comparative experimental analysis of Docker container networking drivers. *IEEE 9th International Conference on Cloud Networking (CloudNet)*, Piscataway, NJ, USA, pp. 1–7, 2020.
- [4] Liu P, Guitart J. Performance characterization of containerization for HPC workloads on InfiniBand clusters: an empirical study. *Cluster Computing*, Vol. 25, pp. 847–868, 2022.
- [5] Stanišić S, Luković V, Belotić B. Automation of Monitoring and Optimization of Docker Containers Using Artificial Intelligence. *24th International Symposium INFOTEH-JAHORINA (INFOTEH)*, March 2025.
- [6] Mahalingam, M., et al. VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. RFC 7348, *IETF*, August 2014.
- [7] Boeira C, Neves M, Ferreto T, Haque I. Characterizing network performance of single-node large-scale container deployments. *IEEE 10th International Conference on Cloud Networking (CloudNet)*, Cookeville, TN, USA, pp. 97–103, 2021.
- [8] Stanišić S, Vesković M, Ristić O, Đorđević B. Security Aspects of Container Orchestration in Kubernetes Environments. *24th International Symposium INFOTEH-JAHORINA (INFOTEH)*, March 2025.
- [9] Claassen J, Koning R, Grosso P. Linux containers networking: Performance and scalability of kernel modules. *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, Istanbul, Turkey, pp. 713–717, 2016.
- [10] Stanišić S, Đorđević B, Ristić O, Ivan T. Performance Optimization of File Systems for Docker Containers. *Sinteza 2025 Conference*, January 2025.

SUMMARY

DOCKER NETWORK TOPOLOGIES: ANALYSIS OF BRIDGE, OVERLAY, AND MACVLAN MODES

In modern container technologies, Docker is a key tool for deploying and managing applications. Understanding the performance of different Docker networking modes is crucial for efficient system design. This paper analyzes the performance of three core Docker network modes—Bridge, Overlay, and MacvLAN—in a production environment (Docker 27.5.0, Ubuntu 24.04.2 LTS). The methodology involves measuring key performance indicators—latency and throughput—under varying workloads. Results show that MacvLAN, with direct access to the physical interface, delivers the best performance (0.3 ms latency, 9.8 Gbps throughput) but requires strict IP allocation control. Overlay networks, using VXLAN encapsulation, introduce ~1.1 ms additional latency per hop and require careful MTU tuning (≤ 1400 bytes) to prevent packet fragmentation. Bridge mode shows limitations under high-throughput scenarios due to DNAT processing, leading to ~14% packet loss with a standard MTU of 1500. In conclusion, the optimal Docker network mode depends on workload characteristics: MacvLAN is ideal for latency-sensitive applications, Overlay suits distributed systems with optimized MTU settings, and Bridge is best for development environments when paired with an external load balancer.

Key Words: Docker, VXLAN, MacvLAN, Bridge mode, MTU fragmentation, container performance