https://doi.org/10.37528/FTTE/9788673955056/POSTEL.2025.27

MOGUĆNOST PRIMENE TEHNIKE ZA OBLIKOVANJE SAOBRAĆAJA ZASNOVANE NA UPOTREBI SAVREMENIH SKRIPTING REŠENJA

Slobodan Mitrović[1], Andrijana Todosijević[2]
[1]Univerzitet u Beogradu - Saobraćajni fakultet, s.mitrovic@sf.bg.ac.rs
[2]Akademska Mreža Srbije (AMRES), andrijana.todosijevic@amres.ac.rs

Rezime: *Imajući u vidu sve veću složenost i dinamiku saobraćaja u lokalnim računarskim mrežama, u radu se predlaže evaluacija sistema za beleženje aktivnosti (syslog-ng) sa ciljem da se utvrdi da li se ovaj sistem može efikasno upotrebiti primenom DevOps skripting tehnika za detekciju anomalija i potrebe oblikovanja saobraćaja (traffic shaping). Fokus istraživanja je na ulozi grafovskih neuronskih mreža (GNN) kao i pratećih tehnika za upotrebu skriptova za navedene svrhe. U radu je dat pregled DevOps tehnika, pregled upotrebe GNN, kao i pregled savremenih alata za vizuelizaciju mrežnih aktivnosti. Analizirana je mogućnost upotrebe zapisa aktivnosti firewall uređaja, koji bi primenom odgovararajućih skriptova bili upotrebljeni za prenos telemetrije i obuku GNN modela kojim se nadalje autonomno pokreće promena konfiguracije upotrebom Netmiko alata u cilju primene tehnika za oblikovanje saobraćaja. Ovakva aktivnost se na kraju može grafički interpretirati primenom odgovarajućeg modela za indeksiranje i vizuelizaciju, baziranom na upotrebi alata poput ElasticSearch i Grafana.*

Ključne reči: *detekcija anomalija, nadgledanje, analiza mrežnog saobraćaja, GNN, oblikovanje saobraćaja*

1. Uvod

Ubrzan rast broja i strukture različitih *online* servisa u savremenim lokalnim računarskim mrežama, korespondira sa brojem različitih obrazaca saobraćaja, kako lokalnoj zoni samih mreža, tako i tokom interakcije sa spoljnim mrežama. Dinamika ovakvih promena u određenim slučajevima može dovesti u pitanje efikasnost primene tradicionalnih strategija statičkog upravljanja lokalnom mrežom. Imajući u vidu da segmentacija mreže primenom VLAN koncepta na *datalink* sloju, kao i primena subnet tehnike adresiranja, kao i rutiranja na mrežnom sloju OSI referentnog modela utiče na pojavu različitih obrazaca saobraćaja sa drugačijim nivoima intenziteta, potreba za automatizovanim, inteligentnim, adaptivnim oblikovanjem saobraćaja u situacijama kada se javlja zagušenje postala je neophodna. U slučajevima kada ulogu centralnog čvorišta za komunikaciju (posmatrano na mrežnom sloju) imaju *layer*-3 svičevi ili ruteri, a u nekim slučajevima i *firewall* uređaji (poput Cisco *Adaptive Security Appliance* - ASA),

ovi uređaji po pravilu služe kao podrazumevani prolazi (*default gateways*) za saobraćaj između korespondentnih podmreža, što ih tada čini i centralnim tačkama za sprovođenje kvaliteta usluge, kontrolu i primenu saobraćajnih politika ili oblikovanje saobraćaja.

Imajući u vidu da se konvencionalni sistem podešavanja parametara za filtraciju saobraćaja zasniva na ručnom podešavanja *firewall* pravila na paketskim filtrima postavlja se pitanje koliko je moguće efikasno pratiti fluktuacije u promeni karakteristika saobraćajnih obrazaca u realnom vremenu i blagovremeno izmeniti parametre filtera kao adekvatan odgovor na uočene situacije. Jedno od efikasnih rešenja za navedeni problem je primena adekvatnih pristupa baziranih na tzv. *DevOps* metodologiji [1]. Iako je *DevOps* pristup u upravljanju mrežnim komponentama i samom mrežom star gotovo 10 godina, novi oblici eksploatacije ovog koncepta imaju potpuno novi značaj u fazi kada se primena neuronskih mreža, mašinsko učenje, kao i primena odgovarajućih oblika veštačke ineligencije nalazi u punom zamahu. Ovo je posebno značajno u slučajevima onih institucija koje poseduju sopstvenu mrežnu infrastrukturu i čije računarske mreže poseduju značajne količine sopstvene opreme koja je statički konfigurisana, gde zamena sa opremom narednih generacija zahteva izrazito veliki nivo troškova.

U ovom radu analizirana je mogućnost primene skripting tehnika nad mrežnim uređajima sa ciljem da se mrežnim uređajima sa konvencionalnim načinom konfigurisanja proširi operativna sposobnost da pruže dinamički odgovor na određene pojave. U konkretnom slučaju, analizirana je mogućnost dinamičke rekonfiguracije rutera/*firewall* uređaja sa ciljem promene parametara za oblikovanje saobraćaja upotrebom grafovskih neuronskih mreža (*Graph Neural Networks - GNN*), kao jedne od tehnika za pružanje dinamičkog odgovora na određene pojave u lokalnom mrežnom saobraćaju. Takođe, biće pružen osvrt na primenu adekvatnih načina vizuelizacije podataka dobijenih procesima monitoringa aktivnosti koji se realizuju kroz *syslog-ng* servis, a koji se pripremaju za vizuelizaciju primenom *ElasticSearch* i *Grafana* alata.

Rad je organizovan na sledeći način: u drugom poglavlju prikazan je kratak pregled DevOps koncepta kao i uloge skripting tehnika. Koncept grafovskih mreža je prikazan u trećem poglavlju, dok je u četvrtom poglavlju analiziran scenario moguće konfiguracije za primenu skripting tehnika u slučaju lokalne mreža sa jednim čvorištem za rutiranje saobraćaja, nakon čega su dati odgovarajući zaključci.

2. *DevOps* i *skripting* tehnike

DevOps kao pojam, koji se u ovom slučaju mogao vezati za lokalne računarske mreže, opisuje se kao skup tehnika za softversko upravljanje informatičkom (mrežnom i serverskom) infrastrukturom, sa ciljem da se poveća stepen automatizacije i brzina konfiguracije/rekonfiguracije, pouzdanost i efikasnost rada posmatranih infrastrukturnih celina. U osnovi, ovakav skup tehnika ima za cilj da se smanji broj individualnih intervencija nad konfiguracijama uređaja, poboljša vidljivost rada i smanji vreme reakcije prilikom pojave različitih vrsta problema. U tom smislu, DevOps predstavlja integrativni faktor nad timskim poslovima, koji se odnose na izradu procedura, razvoj softverskih rešenja za posmatranu opremu, kao i osiguranje kvaliteta i operacija na posmatranoj mreži [2]. DevOps paradigma izlazi iz granica mrežnih i serverskih rešenja i nalazi takođe primenu i u mnogim drugim tehnološkim oblastima, koje su pogodne za primenu principa automatizacije procesa.

Jedan od ključnih načina na koji DevOps doprinosi automatizaciji mreže jeste upotreba *skripting* tehnika, koja uključuje upotrebu specijalizovanih alata, skripti i unapred definisanih radnih tokova. Ovi alati i skripte omogućavaju pouzdano upravljanje i konfigurisanje mrežnih uređaja, kao i automatizaciju složenijih zadataka kao što je inicijalizacija novih uređaja, centralizovano upravljanje konfiguracijom i kontinuirano praćenje performansi mreže. Na primer, softverski paket *Ansible* [3] može se koristiti za automatizaciju raspoređivanja mrežnih konfiguracija na više uređaja istovremeno, dok se softverski paket *Puppet* [4] često primenjuje za održavanje i ažuriranje konfiguracija koje moraju biti potpuno konzistentne u okviru velikih mrežnih infrastruktura. Praksa upotrebe ovih alata predstavlja sastavni deo savremenih zadataka vezanih za razvoj procesa automatizacije mreže i pruža administrativnom osoblju mogućnost primene napredne politike sistemske konfiguracije/rekonfiguracije uređaja u kompleksnim mrežnim okruženjima [5]. Sa druge strane, jednostavnije mrežne strukture pružaju mogućnost primene *Python* skriptova u procesu upravljanja mrežnim uređajima, kao jednostavnije alternative u poređenju sa primenom navedenih alata [6].

Za efikasnu realizaciju procesa automatizacije mreže uz pomoć DevOps metodologije potrebna je i pristupa poput kontinuirane integracije (*Continuous Integration*, CI) i isporuke (*Continuous Delivery*, CD). Ovi pristupi podrazumevaju kontinuiranu integraciju promena u zajednički repozitorijum koda , detaljno testiranje tih promena, kao i njihovo automatizovano raspoređivanje u produkciono okruženje (*Continuous Deployment*). Na taj način omogućavaju se brže i efikasnije izmene mrežne infrastrukture, uz značajno smanjenu mogućnost grešaka i bolju transparentnost celog procesa. Istovremeno, CI/CD pristup unapređuje sposobnost tima da brzo identifikuje, prati i rešava potencijalne probleme, čime se poboljšava ukupna stabilnost i pouzdanost mreže [7] (Slika1).

Slika 1. Proces automatizacije uporebom DevOps metodologije [7]

Automatizacija mreže uz DevOps uključuje i korišćenje sistema za kontrolu verzija konfiguracionih fajlova, poput Git-a, koji omogućavaju detaljno praćenje, dokumentovanje i upravljanje konfiguracijama mreže tokom celog životnog ciklusa. Ovakav pristup olakšava povratak na prethodne verzije konfiguracija u slučaju nepredviđenih problema, omogućava jasan uvid u istoriju izmena i doprinosi dugoročnoj konzistentnosti i standardizaciji mrežnih okruženja. Na taj način organizacije dobijaju veću kontrolu nad infrastrukturom i mogućnost efikasnijeg planiranja budućih modifikacija [5].

Integracija DevOps tehnika sa tehnikama poput mašinskog učenja (ML), neuronskih mreža (NN) i veštačke inteligencije (AI), rezultovala je različitim oblicima primene i razvoja tehničkih rešenja [8], [9]. Detekcija anomalija primenom adekvatnih tehnika mašinskog učenja, sa ciljem da se određene anomalije što ranije uoče u analiziranim vremenskim serijama u okviru široke primene DevOps tehnika, prikazana je

u [10]. Uloga neuronskih mreža u okviru primene DevOps sa ciljem automatizacije detekcije degradacije performansi prikazana je spektru različitih primena, poput [11].

3. Grafovske neuronske mreže (GNN)

Grafovske neuronske mreže (GNN) predstavljaju specijalizovanu klasu neuronskih mreža dizajniranih za učenje nad podacima koji su strukturirani kao grafovi i čiji se način rada zasniva na mehanizmu agregacije informacija (tzv. poruka) [12]. Za razliku od tradicionalnih neuronskih mreža koje tretiraju ulazne podatke kao nezavisne instance, GNN modeli uzimaju u obzir strukturu grafa, gde čvorovi predstavljaju entitete, a grane odnose između njih. Ključna ideja GNN-a je da svaki čvor u grafu uči reprezentaciju $h_u^{(k)}$ zasnovanu na sopstvenim karakteristikama, ali i na informacijama prikupljenim od susednih čvorova *u* kroz proces poznat kao *message passing*. Formalno, za čvor ($u \in V$) u sloju (*k*), njegova nova reprezentacija može se definisati kao [13]:

$$h_u^{(k+1)} = UPDATE^{(k)}(h_u^{(k)}, AGGREGATE^{(k)}(\{h_u^{(k)}, \forall \upsilon \in N(u)\})), \text{ odnosno} \quad (1)$$

$$h_u^{(k+1)} = UPDATE^{(k)}(h_u^{(k)}, m_{N(u)}^{(k)}), \quad (2)$$

gde je ($N(\upsilon)$) skup susednih čvorova, *UPDATE* i *AGGREGATE* odgovarajuće diferencijabilne funkcije agregacije, dok $m_{N(u)}^{(k)}$ predstavlja agregiranu „poruku“ koja potiče iz skupa susednih čvorova čvora *u* ($N(u)$). Ovaj postupak omogućava da reprezentacije čvorova postupno integrišu širi kontekst grafa kroz više slojeva, što je ključno za modelovanje složenih međuzavisnosti. Na slici 2 prikazan je primer kako jedan čvor (A) agregira poruke iz skupa susednih čvorova.

Slika 2. GNN agregacija poruka kroz slojeve [13]

Agregacijom se skupljaju poruke od lokalnih grafičkih suseda čvora A (tj. B, C i D), a zatim se poruke koje dolaze od ovih suseda zasnivaju na informacijama agregiranim iz njihovih odgovarajućih susedstava i tako dalje. Kao što se može videti na prikazanoj slici, prenos poruka se u prikazanom slučaju vrši u dva sloja [13]. Jedna od najpoznatijih arhitektura u ovoj vrsti neuronskih mreža su *grafovske konvolucione mreže* (*Graph Convolutional Networks* - GCN), kod kojih se generalizuju konvolucione operacije poznate iz obrade slika na nepravilne graf strukture. U ovoj vrsti mreža, agregacija informacija iz susednih čvorova se često izvodi pomoću normalizovane Laplasove matrice grafa, čime se osigurava stabilnost učenja i smanjenje efekta različitih stepena čvorova. Grafovske mreže sa mehanizmom pažnje (*Graph Attention Networks* - GAT)

uvode dodatni mehanizam pažnje, gde se težina svakog suseda određuje prema njegovoj važnosti za zadatak, omogućavajući selektivno fokusiranje na relevantne veze [13].

4. Moguća skripting rešenja u slučaju primene tehnike oblikovanja saobraćaja

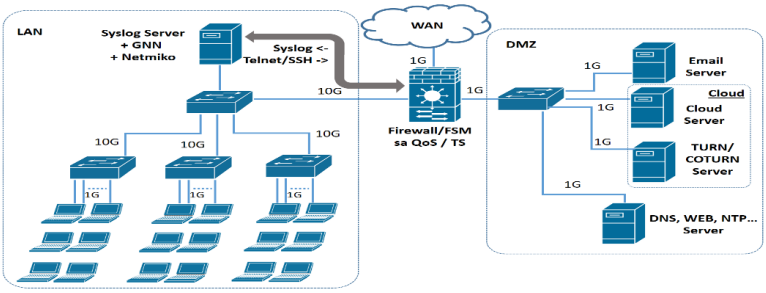
Savremene lokalne mreže koje poseduju svoje sopstvene *online* servise, po pravilu, poseduju i sopstvenu hosting infrastrukturu koja se nalazi u intenzivnoj eksploataciji od strane korisnika. Struktura savremenih mrežnih servisa ne uključuje samo tradicionalne tipove poput DNS, email, ili veb servisa, već često poseduje i napredne oblike, poput sopstvene *cloud* infrastrukture koja ne nudi samo ličnu i grupnu razmenu datoteka, već često poseduje i *real-time* servise kao što su video-konferencije, kolaborativne *cloud* aplikacije (grupno editovanje dokumenata) i/ili interaktivne *chat* servise. U ovakvim sistemima vrlo često je zastupljen TURN (*Traversal Using Relays around NAT*) protokol, koji pruža primene koncepta „premošćavanja" UDP tokova kada direktna veza nije moguća (na primer kroz NAT i/ili *firewall* zone), pri čemu pruža mogućnost minimalnog kašnjenja paketa. U navedenom slučaju TURN server funkcioniše kao posrednik, primajući podatke sa jednog uređaja i šaljući ih na drugi. TURN server i predstavlja poslednje sredstvo izbora kada STUN (*Session Traversal Utilities for NAT*) komponenta ne uspe da uspostavi direktnu vezu. Jedna od popularnih softverskih platformi, koja za interaktivnu komunikaciju koristi TURN server (kao dodatnu komponentu) je i NextCloud platforma za samostalni hosting oblaka i za potrebe sopstvenih korisnika u LAN i/ili WAN okruženju. Drugim rečima, udeo zahteva koji podrazumevaju upotrebu UDP protokola u ukupnoj strukturi saobraćaja raste kada se razmatra upotreba navedenih vrsta servisa (Tabela 1). Da bi se omogućila pouzdana i efikasna obrada takvih tokova, neophodno je implementirati strategije kvaliteta servisa (QoS) koje favorizuju vremenski osetljivu komunikaciju, kao i odgovarajući odnos zastupljenosti i drugih servisa važnih za korisnike, poput DNS, Web, IMAP i sl.

Tabela 1. Pregled UDP/TCP portova zastupljenih u slučaju upotrebe cloud servisa

Servis	**Protokol**	**Port**	**Opis**
TURN (standardni)	UDP	3478	TURN port za *relaying* audio/video tokova
TURN (TLS/DTLS)	TCP/UDP	5349	TURN portovi preko TLS/DTLS
TURN dinamički *relay*	UDP	49152-65535	Dinamički dodeljeni portovi za UDP *stream*
Kolaborativni	TCP	9980	Bazni servis za *onlin*e Office
WEB	TCP	80/443	Veb Servisi
IMAP StartTLS	TCP	143/587	Kriptovani email protokol

Imajući u vidu da u slučaju saobraćaja „vezanog za aktivnosti u oblaku" zastupljenost UDP saobraćaja po pravilu treba da funkcioniše zajedno sa servisima za prenos datoteka (dakle, sa konektivnim protokolima kao što je TCP) na vezama sa limitiranim kapacitetom, u tačkama rutiranja gde postoje paketski filtri često se uključuju saobraćajne politike ili mehanizmi za oblikovanje saobraćaja. Oblikovanje saobraćaja podrazumeva da se saobraćaj, čiji broj dolaznih paketa prelazi određeno ograničenje, stavlja u red (baferuje) i šalje tokom sledećeg vremenskog intervala. Treba napomenuti

da primena tehnika oblikovanja saobraćaja može pokazati željene rezultate u slučajevima kada se pojavljuje zagušenje veze. Konkretno, oblikovanje saobraćaja može ostvariti puni efekat ako postoji veza sa nižim protokom, poput slučaja koji je prikazan na slici 3, gde *uplink* veza između LAN segmenta mreže (gde su locirane korisnici) i rutera sa firewall-om ima veći kapacitet nego veza sa serverima u DMZ zoni (gde su locirani *cloud* servisi) ili veza sa WAN okruženjem. U ovakvom slučaju dolazi do dominacije TCP saobraćaja koji bez prisustva mehanizma kontrole (saobraćajnih politika ili primenjenih tehnika oblikovanja saobraćaja) može zauzeti i ceo raspoloživi kapacitet na posmatranom linku.



Slika 3. Mreža sa centralnom tačkom rutiranja

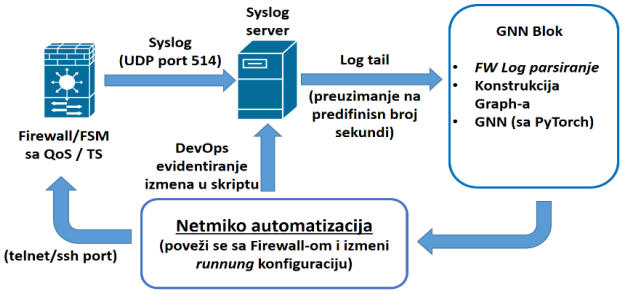
Savremene okolnosti često ukazuju na potrebu da se u slučajevima, kada se primenjuju tehnike oblikovanja saobraćaja, parametri prioriteta određenih tipova saobraćaja moraju češće adaptirati, pogotovo u slučajevima koegzistencije različitih generacija uređaja koji poseduju mrežne portove različitih kapaciteta. Da bi se ovaj problem rešio, neophodno je uključiti sisteme za praćenje aktivnosti mreže, koji mogu vršiti zapis relevantnih vrednosti parametara od kojih zavisi primena tehnika za oblikovanje saobraćaja. Očitavanjem zapisa nadalje se mogu donositi odluke o dodeljivanju odgovarajućih nivoa prioriteta određenim tipovima saobraćaja - ručno ili putem primene nekog od mehanizama za automatizovano donošenje odluka.

U ovom poglavlju se analizira mogućnost uvođenja mehanizma za automatizovano donošenje odluka primenom odgovarajućih skripting rešenja na nivou servera namenjenog za upravljenje mrežnim uređajima, pri čemu manipulacija skriptovima treba da funkcioniše u skladu sa DevOps konceptima. Mogućnost DevOps ručnog upravljanja mrežnim komponentama upotrebom odgovarajućih skriptova nije nov pristup i kao takav ima široku primenu. Nasuprot tome, uvođenje inteligentnih mehanizama u procese dinamičkog (a u nekim slučajevima i autonomnog) upravljanja zasnovanog na primeni skiptinga još uvek nema široku zastupljenost u produkcionim rešenjima, najčešće zbog širokog spektra mogućih pristupa za primenu nad određenim vrstama mrežne infrastrukture. U analiziranom slučaju (Slika 3), ispitana je mogućnost da se u proces dinamičkog upravljanja uvedu skripting rešenja, pretežno zastupljena u distribucijama Linux operativnih sistema. Mreža koja se razmatra sadrži LAN segment koji ima backbone kapacitet 10 Gb/s, centralni switch, kao i Cisco ruter/firewall kao centralnu tačku rutiranja u posmatranoj mreži. Pored navedenog, ova mreža sadrži i demilitarizovanu zonu (DMZ) sa serverskim instancama i linkovima kapaciteta 1 Gb/s. U posmatranom scenariju među navedenim instancama nalazi se i Nextcloud server sa TURN servisom. Scenario podrazumeva i da Firewall sa svojim karakteristikama ima

mogućnost primene tehnika oblikovanja, mogućnost primene politika i definisanje prioriteta paketskih redova, poput sledećeg primera u kome se za TURN saobraćaj dodeljuje 50% raspoloživog kapaciteta, što se vidi u prikazanom delu konfiguracije:

```
access-list TURN_ACL extended permit udp any any range 3478 3478
access-list TURN_ACL extended permit tcp any any range 5349 5349
class-map TURN_CLASS
match access-list TURN_ACL
policy-map PRIORITY_POLICY
class TURN_CLASS
priority percent 50
class class-default
bandwidth percent 50
service-policy PRIORITY_POLICY interface outside
```

Shodno navedenom, predlaže se sledeće *skripting* rešenje za moguću automatizaciju procesa oblikovanja saobraćaja (Slika 4).



Slika 4. Skripting rešenje za moguću automatizaciju procesa oblikovanja saobraćaja

ruter/*firewall* uređaj se konfiguriše za slanje syslog poruka ka centralizovanom syslog serveru korišćenjem UDP ili TCP, pri čemu se najčešće koristi UDP port 514. Ova konfiguracija obično uključuje definisanje adrese Syslog servera, tipove poruka koje se šalju serveru, kao i mrežni interfejs koji će biti zadužen za slanje. Tipovi poruka treba da uključe sve relevantne događaje, uključujući događaje vezane za QoS, kao i sve događaje u vezi sa mrežnim konekcijama koje u sebi sadrže i protokole od interesa. Za potrebe GNN analize bitni su određeni *syslog* identifikatori (ID) koje Cisco Firewall (ovom slučaju) može generisati tokom rada, jer predstavljaju direktne indikatore stanja mrežnih sesija, performansi i potencijalnih QoS problema (Tabela 2).

Tabela 2. Primeri identifikatora koji mogu biti od važnosti za treniranje GNN

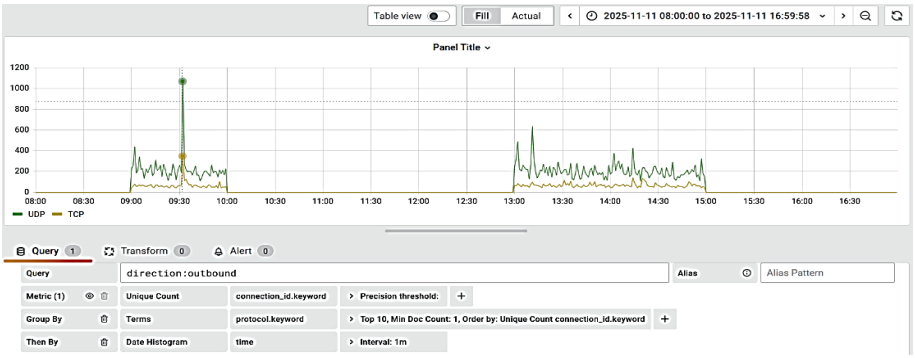
Identifikator (ID)	**Događaj / stanje**
302013–302016	connection built
302020	TCP state
302021–302023	connection teardown
411001, 415001, 419001	QoS, shaping drops
611101–611103	routing

Kombinacijom ovih identifikatora *syslog-ng* može da kreira detaljan zapis, koje GNN koristi kao atribute za formiranje karakteristika čvorova i grana u grafu radi detekcije preopterećenja i aktiviranja određene odluke (u vidu *flag*-a) za izmenu konfiguracije preko Netmiko paketa. Da bi se to moglo realizovati, Syslog podaci se preuzimaju iz log datoteke upotrebom *tail* komande u regularnim vremenskim intervalima (npr. svakih 5 sec), a zatim se moraju preformatirati u JSON format koji GNN modul koristi za učitavanje i formira graf koji se obrađuje kroz nekoliko slojeva konvolucije nad čvorovima kako bi se otkrile zavisnosti između mrežnih događaja. Obuka GNN-a vrši se u sledećim fazama:

1. Za svaki događaj biraju se atributi (portovi, protokoli, količina saobraćaja, trajanje, tipovi ID-eva i statističke metrike), koji se kodiraju u numerički vektor čvora. GNN zatim formira graf tako što povezuje uzastopne zapise istog toka, grupiše logove koji dele IP/protokol, vezuje one sa istim TURN/DNS/NTP portovima i povezuje anomalije sa njihovim susednim događajima, obezbeđujući modelu da uči kontekst, a ne izolovane zapise;
2. Model se trenira definisanjem ciljne vrednosti: klasifikacije događaja (npr. regularan, degradiran, kritičan, anomalija), regresije očitanih vrednosti (opterećenje linka, verovatnoća zagušenja, QoS prioriteti) ili preporuke izmene trenutne konfiguracije. Tokom treniranja GNN prolazi kroz slojeve *message passing*-a, računa *loss* vrednost (npr. *cross-entropy*), a zatim se greška propagira unazad i ažuriraju težine (npr. upotrebom Adam modula). Validacija se vrši nad grafovima koje GNN nije „upoznao“, kako bi se utvrdilo da li model prepoznaje obrasce, npr. da li prepoznaje TURN zagušenja iako ih nije video tokom treninga;
3. Kada je obuka završena, GNN generiše klasifikaciju svakog log događaja, preporuke, poput narednog primera: Ukoliko izlazni vektor dobije vrednost 0, tada se syslog događaj ocenjuje kao regularan, u suprotnom se ocenjuje kao identifikovana anomalija (npr. abnormalan TURN nivo saobraćaja, DoS, spike). Shodno tome, rezultat može delimično pripremljeno značenje, poput "TURN sesije na portu 3478 generišu visoko kašnjenje“, pa se u kodu definiše izlazna vrednost koja označava da mreža preporučuje QoS prioritet: „visok“, odnosno generiše se *flag*, koji npr. dobije vrednost „3“. Ovaj *flag* se, nadalje, prosleđuje u Netmiko skript koji ima predefinisane parametre pristupa Firewall-u, kao i uslovne korake kojima vrši izmena predefinisanih vrednosti ili komandi unutar same konfiguracije i to u zavisnosti od same *flag* vrednosti.

Monitoring prikazanog procesa (Slika 5) može se vršiti kroz vizuelizaciju syslog-ng saobraćaja, koji se pre toga mora konvertovati u odgovarajući format. To se može učiniti upotrebom *Python* skripta, koji ima zadatak da aktivira *tail* komandu nad odgovarajućom log datotekom u regularnim intervalima i zatim izvrši parsiranje i normalizaciju izdvojenih poruka, koje potom prosleđuje u Elasticsearch gde se podaci indeksiraju i čuvaju u formi dokumenata optimizovanih za brzu pretragu i agregaciju. Nakon indeksiranja, Grafana preuzima ulogu interaktivne frontend aplikacije, koja omogućava pregled i analizu logova kroz dashboard-e, grafike, filtere i pretrage zasnovane na *Grafana Query Language* (KQL). Generalno, Grafana omogućava kreiranje *real-time* panela koji automatski osvežavaju prikaz, što je važno za korelaciju događaja i otkrivanje incidenata [14]. Kombinacija *syslog-ng* sa paketima Elasticsearch i Grafana značajno olakšava otkrivanje trendova i anomalija u mrežnim logovima i predstavlja tipičnu

osnovu za dalju analitiku ili formiranje AI/ML *pipeline* linkova kao spoljnim mrežnim lokacijama.



Slika 5. Primer generisanog grafikona u Grafana interfejsu sa brojem paketa usmerenim ka DMZ zoni

5. Zaključak

Implementacija automatizovanih *skripting* rešenja u procesima primene tehnika za oblikovanje saobraćaja, zajedno sa alatima za vizuelizaciju predstavlja korak u evoluciji QoS metodologije u kojoj se primenjuju principi adaptivnog učenja. Ova sinteza podržava poboljšanu otpornost mreže kada je suočena sa zagušenjima u zonama „uskog grla“, obezbeđuje viši kvalitet servisa pod promenljivim opterećenjem i usklađuje administrativnu politike sa stvarnom dinamikom saobraćaja. Kako kompleksnost mreža nastavlja da raste, integracija analize bazirane na GNN-ovima sa tradicionalnim QoS alatima postaje sve neophodnija u slučajevima kada je u infrastrukturi prisutno nekoliko generacija mrežne i serverske opreme. Prednost tehnike upotrebe skriptova može se uvideti u činjenici da se nije ograničena na primenu tehnike oblikovanja saobraćaja, već može biti primenjeno i za rešavanje bilo kog drugog problema na mreži. Imajući u vidu da je u slučajevima kompleksnijih mreža komplikovano koristiti *tail* opciju u svakodnevnim poslovima administracije mreže, potrebno je razmotriti upotrebu alternativnih alata koji mogu da formiraju strimove podataka ka više različitih komponenti za podršku odlučivanju. To se ujedno može smatrati budućim pravcem daljeg istraživanja upotrebe *skripting* rešenja u lokalnim računarskim mrežama.

Literatura

[1] L. Bass, I. Weber i L. Zhu, DevOps: A software architect's perspective, Addison-Wesley Professional, 2015.

[2] C. Ebert, G. Gallardo, J. Hernantes i N. Serrano, „DevOps,“ *IEEE software*, t. 33, p. 94–100, 2016.

[3] M. Borgenstrand, *Network automation – the power of Ansible*, 2018, p. 116.

[4] S. Wagbrant i V. Dahlen Radic, *Automated Network Configuration : A Comparison Between Ansible, Puppet, and SaltStack for Network Configuration*, 2022, p. 41.

[5] PyNet Labs, *Network Automation Using DevOps,* 2025.

[6] N. A. Younes, *Network Automation: A Comparative Analysis of Ansible and a Custom Python-Based Tool,* 2024, p. 30.

[7] Civo, *The Role of the CI/CD Pipeline in Cloud Computing,* 2024.

[8] T. Mboweni, T. Masombuka i C. Dongmo, „A Systematic Review of Machine Learning DevOps," p. 1–6, July 2022.

[9] G. Ramesh, T. V. Pai, R. Birau, K. K. Poojary, A. R. Shingad, N. Sowjanya, V. Popescu, A. T. Mitroi, R.-M. Nioata, K. K. Raj i others, „A comprehensive review on scaling Machine Learning workflows using Cloud Technologies and DevOps," *IEEE Access,* 2025.

[10] A. Hrusto, E. Engström i P. Runeson, „Towards optimization of anomaly detection in DevOps," *Information and Software Technology,* t. 160, p. 107241, 2023.

[11] Z. Sun, Y. Li i L. Wen, „DevOps and Neural Network Based Full Lifecycle Management Model for Power Information Systems," *Procedia Computer Science,* t. 208, pp. 642-649, 2022.

[12] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner i G. Monfardini, „The Graph Neural Network Model," *IEEE Transactions on Neural Networks,* t. 20, p. 61–80, January 2009.

[13] W. L. Hamilton, „The Graph Neural Network Model," u *Graph Representation Learning,* Cham, Springer International Publishing, 2020, p. 51–70.

[14] A. Todosijević, K. Simonović i A. Arsović, *Upravljanje logovima i vizualizacija statistika korišćenja AMRES servisa upotrebom alata otvorenog koda,* University of Belgrade - School of Electrical Engineering and Academic Mind , 2022.

Abstract: *Considering the raising complexity and dynamics of traffic generated in local computer networks, the paper proposes an activity logging system (syslog-ng) evaluation with the aim of determining whether this system can be effectively used by DevOps scripting techniques in terms of the detection of traffic anomalies, as well as the decision support regarding the activation of traffic shaping policies. The focus of the research is given to the Graph Neural Networks (GNN) deployment along with corresponding scripting techniques. The paper provides an overview of DevOps techniques, an overview of the GNN, as well as an overview of modern tools for visualizing network activities. Therefore, the possibility of using firewall device activity records was analyzed, in the purpose of using telemetry in a process of GNN training. GNN output would be further used to autonomous change of traffic shaping policies using the Netmiko tool. All related activity could be graphically interpreted by applying a suitable model for indexing and visualization, based on tools like ElasticSearch and Grafana.*

Keywords: *anomaly detection, monitoring, network traffic analysis, Graph Neural Networks (GNN), traffic shaping*

DEPLOYABILITY OF TRAFFIC SHAPING TECHNIQUE BASED ON THE USE OF MODERN SCRIPTING SOLUTIONS

Slobodan Mitrović, Andrijana Todosijević