UNIVERZITET U BANJOJ LUCI

ELEKTROTEHNIČKI FAKULTET

Sistem za nadzor, analizu i upravljanje konfiguracijama mrežnih uređaja zasnovan na NETCONF protokolu

Master rad

Mentor:
Doc. dr Mihajlo Savić

Kandidat:
Bojana Martinović

Banja Luka, septembar 2025.

UNIVERSITY OF BANJA LUKA

FACULTY OF ELECTRICAL ENGINEERING

A System for Monitoring, Analyzing, and Managing Network Device Configurations Based on the NETCONF Protocol

Master thesis

Mentor:
Asst. Prof. Mihajlo Savić, PhD

Candidate:
Bojana Martinović

Banja Luka, September 2025.

Tema: **Sistem za nadzor, analizu i upravljanje konfiguracijama mrežnih uređaja zasnovan na NETCONF protokolu**

Mentor: **Dr Mihajlo Savić, docent,**
Elektrotehnički fakultet,
Univerzitet u Banjoj Luci

Naučna oblast: **Inženjerstvo i tehnologija**

Naučno polje: **Elektrotehnika, elektronika i informaciono inženjerstvo**

Ključne reči: *nadzor i upravljanje mrežom, konfiguracije mrežnih uređaja, bezbednost protokola za mrežno upravljanje, NETCONF, YANG, XML, modelovanje podataka*

Klasifikaciona oznaka (CERIF): **T120**

Sažetak: Ovaj rad se bavi integracijom nadzora mreže i NETCONF protokola u cilju proaktivnog upravljanja konfiguracijama mrežnih uređaja. Predloženi sistem ne samo da prikuplja podatke o konfiguracijama uređaja, već i detektuje nekonzistentnosti u njima i identifikuje potencijalne okidače koji su ih izazvali. Takvo rešenje značajno unapređuje stabilnost i bezbednost mreže. Na ovaj način se smanjuje rizik od pojave grešaka koje nastaju kao posledica ljudske nepažnje ili nedostatka stručnosti i obezbeđuje se očekivano stanje konfiguracije. Nadgledanje mreže je od velikog značaja u okruženjima u kojima često dolazi do promene u konfiguracijama. Kada je integrisan sa NETCONF protokolom, sistem može precizno odrediti koji deo konfiguracije je odstupio od očekivanog stanja, bilo da se radi o pogrešnoj vrednosti, nedostajućem elementu ili suvišnom elementu konfiguracije. Korišćenjem NETCONF protokola i XPath-a, moguće je detaljno izdvojiti i vizuelno predstaviti odstupanja. Ovo značajno olakšava analizu grešaka kao i mogućnost za njihovo otklanjanje.

Topic:	**A system for monitoring, analyzing, and managing network device configurations based on the NETCONF protocol**
Mentor:	**Dr Mihajlo Savić, Assistant Professor, Faculty of Electrical Engineering, University of Banja Luka**
Scientific area:	**Engineering and technology**
Scientific field:	**Electrical engineering, electronics and information engineering**
Keywords:	*network monitoring and management, network device configurations, network-level security and protection, NETCONF, YANG, XML, data modeling*
Classification label (CERIF):	**T120**
Creative Commons license:	**CC BY-SA**

Abstract: This thesis explores the integration of network monitoring and the NETCONF protocol for the purpose of proactive network device configuration management. The proposed system not only collects data on device configurations, but also detects inconsistencies and identifies potential triggers that may have caused them. Such a solution significantly improves network stability and security. It reduces the risk of errors caused by human oversight or lack of expertise and ensures that configurations remain in their expected state. Network monitoring is especially important in environments where configurations frequently change. When integrated with NETCONF, the system can precisely determine which part of the configuration deviates from the expected state — whether it involves incorrect values, missing elements, or redundant parameters. By leveraging the NETCONF protocol and XPath, deviations can be extracted in detail and visually represented. This greatly facilitates error analysis and provides a clear path to resolving configuration issues.

Mojoj porodici – onima koji me vode, podržavaju i vole, kao i onima kojih više nema, a nastavljaju da žive kroz mene.

LISTA KORIŠĆENIH SKRAĆENICA

SKRAĆENICA	ZNAČENJE
SDN	Software-Defined Networking
NETCONF	Network Configuration Protocol
TCP	Transmission Control Protocol
SNMP	Simple Network Management Protocol
IP	Internet Protocol
IETF	Internet Engineering Task Force
UDP	User Datagram Protocol
SSH	Secure Shell
TLS	Transport Layer Security
XML	Extensible Markup Language
XPath	XML Path Language
YANG	Yet Another Next Generation
RPC	Remote Procedure Call
LAN	Local Area Network
ML	Machine Learning
AI	Artificial Intelligence
RFC	Request for Comments
NMS	Network Management Station
MIB	Management Information Base
OID	Object Identifier
SMI	Structure of Management Information
ASN	Abstract Syntax Notation
CLI	Command Line Interface
UTF-8	Unicode Transformation Format – 8-bit
NETMOD	NETwork data MODel
YIN	YANG Independent Notation
URI	Uniform Resource Identifier
IoT	Internet of Things

SADRŽAJ

Uz rad je priložena elektronska verzija rada

1. UVOD

1.1. Predmet istraživanja

Nadzor mreže predstavlja ključni proces u obezbeđivanju efikasnog i pouzdanog rada mrežnih uređaja i same mrežne infrastrukture. Nadzor mreže podrazumeva kontinuirano praćenje performansi, dostupnosti, kao i konfiguracionog statusa mrežnih uređaja. Putem nadzora potrebno je omogućiti korisnicima da imaju uvid u istorijski status uređaja, što omogućava pravovremeno otkrivanje promena u konfiguraciji, kao i identifikaciju njihovih uzroka. S obzirom na to da se broj mrežnih uređaja u različitim oblastima primene stalno povećava, postaje sve izazovnije osigurati da aktivne konfiguracije odgovaraju očekivanom stanju.

Potreba za programabilnim, centralizovanim i automatizovanim upravljanjem mrežnim uređajima konstantno raste sa razvojem savremenih tehnologija, kao što je SDN (*Software-Defined Networking*). Među protokolima koji omogućavaju nadgledanje i upravljanje uređajima u mreži, izdvajaju se SNMP (*Simple Network Management Protocol*) i NETCONF (*Network Configuration Protocol*). NETCONF nudi standardizovan i strukturiran pristup upravljanju mrežnom infrastrukturom, što ga čini jednom od ključnih komponenti za konfiguraciju i nadzor mrežnih uređaja.

Za razliku od SNMP-a, NETCONF se smatra značajno bezbednijim protokolom, jer koristi TCP (*Transmission Control Protocol*) i enkripciju u komunikaciji. Ipak, ni ovakav pristup ne može u potpunosti sprečiti pojavu grešaka u konfiguraciji kao što su: pogrešan port, pogrešan broj portova, neispravne IP adrese i slično.

U prošlosti, kada je broj uređaja bio mali, a mrežne infrastrukture jednostavne i relativno statične, bilo je moguće ručno pratiti stanje uređaja. Međutim, u modernim mrežama, koje obuhvataju veliki broj uređaja i dinamički promenljivu topologiju mreže, čak i kratkotrajan prekid u radu može dovesti do ozbiljnih posledica. Nadzor mreže obuhvata konstantno praćenje i upravljanje mrežnim uređajima sa ciljem pravovremenog otkrivanja, identifikacije i otklanjanja grešaka, kako bi se obezbedilo da uređaji funkcionišu efikasno, neprekidno i ispravno. Rano otkrivanje grešaka predstavlja ključni faktor za sprečavanje ozbiljnijih problema i za očuvanje stalne dostupnosti uređaja u mreži. Nadgledanje u realnom vremenu omogućava detaljan uvid u rad mrežnih komponenti, brzu identifikaciju problema i odgovarajuću reakciju. Savremena rešenja za nadgledanje teže ka centralizovanom i automatizovanom pristupu monitoringu, gde sistem samostalno prikuplja informacije, obrađuje ih u realnom vremenu i dostavlja relevantne rezultate preko intuitivnog korisničkog interfejsa. Na ovaj način, korisnici mogu brzo da uoče probleme kao što su nekonzistentne konfiguracije, česte promene na istim ili sličnim uređajima, ili pojave grešaka koje ukazuju na određene rizike u bezbednosti.

U praksi, uređaji se konfigurišu u skladu sa specifičnim zahtevima, često korišćenjem različitih komandi koje se razlikuju u zavisnosti od proizvođača. Zbog toga, složenost procesa konfigurisanja predstavlja jedan od najčešćih izvora grešaka, a sistem koji omogućava detaljan nadzor i analizu konfiguracija postaje iznimno važna karika u lancu upravljanja mrežom. Zbog svega prethodno navedenog, razvijeni su različiti protokoli za upravljanje mrežom.

Krajem osamdesetih godina prošlog veka, IETF (*Internet Engineering Task Force*) grupa razvila je SNMP protokol, namenjen prikupljanju i organizaciji informacija o konfiguraciji mrežnih uređaja, kao i upravljanju njima. Prva verzija SNMP-a (SNMPv1) imala je brojna ograničenja, naročito u pogledu bezbednosti i prenosa većih količina podataka. Unapređene verzije, SNMPv2 i SNMPv3, donele su poboljšanja, pri čemu se SNMPv3 posebno ističe uvođenjem mehanizama za autentifikaciju, poverljivost i kontrolu pristupa. Za razliku od SNMP-a, koji se uglavnom oslanja na nepouzdani UDP protokol, NETCONF koristi TCP, što omogućava veću pouzdanost i kontrolu u komunikaciji. Ovaj protokol podržava očuvanje integriteta podataka, proveru identiteta, zaštitu od neovlašćenih izmena i pouzdanost, što se postiže primenom bezbednih transportnih protokola kao što su SSH (*Secure Shell*) i TLS (*Transport Layer Security*), koji omogućavaju siguran prenos podataka. Pomoću ovog protokola moguće je preuzeti konfiguracione podatke sa mrežnog uređaja, kao i postaviti novu konfiguraciju ili modifikovati postojeću. NETCONF omogućava automatizovan i praktičan pristup konfiguraciji i samom uređaju putem jasno definisanih programskih interfejsa.

Kombinacijom monitoringa i NETCONF-a dobija se moćan alat za upravljanje mrežom. Sistem osim što može da prikupi podatke o radu uređaja, on takođe može i da detektuje nekonzistentnost u konfiguracijama, kao i da automatski reaguje po potrebi. Ovakav pristup značajno unapređuje stabilnost i bezbednost mrežne infrastrukture, minimizuje mogućnost ljudske greške i obezbeđuje da konfiguracije ostanu u skladu sa očekivanim šablonima. Ovaj pristup nadgledanja mreže je od ključne važnosti u okruženjima u kojima često dolazi do promena u konfiguracijama, bilo zbog nadogradnji, ručnih intervencija ili automatizovanih procesa. Kombinacija monitoringa i NETCONF protokola omogućava ne samo prikupljanje podataka o stanju uređaja, već i analitičku proveru, kao i upoređivanje konfiguracija u realnom vremenu. Dok klasični monitoring omogućava uvid u performanse i dostupnost uređaja, integracija sa NETCONF-om pruža detaljan uvid u stvarnu i očekivanu konfiguraciju svakog uređaja u mreži. Ovakva veza omogućava da se uređaji nadgledaju i kontrolišu, tako što se konfiguracije automatski preuzimaju, proveravaju i upoređuju na nivou pojedinačnih XML čvorova.

Uvođenjem automatizovanog rešenja za poređenje konfiguracija mrežnih uređaja, kao i obaveštavanjem o razlikama u konfiguracijama u realnom vremenu, značajno se smanjuje vreme koje je potrebno mrežnom administratoru da reaguje i reši nekonzistentnosti koje su nastale u radu uređaja. Vreme koje je potrebno za reakciju mrežnog administratora se znatno smanjuje jer njegov posao više nije da detektuje problem, već on dobija obaveštenje od strane sistema za nadzor, analizu i upravljanje konfiguracijama mrežnih uređaja zasnovanog na NETCONF protokolu u realnom vremenu. Na ovaj način se efikasnost mrežnog administratora značajno povećava.

Cilj ovog rešenja, osim uočavanja da je došlo do promene i obaveštavanja korisnika o tome je i utvrđivanje konkretnog dela konfiguracije koji je odstupio, bilo da je reč o različitoj vrednosti parametra, nedostajućem polju u konfiguraciji uređaja ili postojanju suvišnog parametra. Zahvaljujući NETCONF protokolu i XPath (*XML Path Language*) alatima, moguće je detaljno izdvojiti i vizuelno predstaviti te razlike, što znatno olakšava otklanjanje grešaka. Istorijski podaci o statusu uređaja, koji bi se čuvali u bazi, omogućili bi da se prepoznaju ponavljajuće konfiguracione greške, što je posebno važno u velikim i dinamičnim mrežama. Sistem, osim što predstavlja moćan alat za ogromne mrežne infrastrukture, poželjno je da bude implementiran tako da omogući nezavisnost između komponenti, što u ovom slučaju i jeste. Takođe, jednostavna proširenja u budućnosti kao i jednostavnost prilikom instalacije i održavanja u različitim okruženima su neke od bitnih karakteristika ovog sistema.

Sistem ima prostora za proširenje i nadogradnju, jer se od ovog sistema koji omogućava nadgledanje mreže u realnom vremenu, kao i dostavljanje svih potrebnih informacija za pravovremeno reagovanje mrežnog administratora na pojavljivanje problema, može primenjivanjem određenih tehnika mašinskog učenja doći do sistema koji osim nadgledanja mreže, identifikovanja problema i dostavljanja svih informacija o problemu, može da pruži i funkcionalnost automatskog reagovanja na problem u smislu obučavanja neuronske mreže da za probleme koji se pojavljuju često i koji su specifični za određeni tip uređaja, pravovremeno reaguje izvršavanjem akcije ili niza akcija koje će sprečiti nastanak većih problema, ili rešiti problem odmah pri njegovom pojavljivanju. Takođe, ovaj pristup može biti iskorišćen i u nadograđivanju sistema u smislu bezbednosti, gde će mreža prepoznati ne samo tehničke greške već i potencijalno sumnjive izmene u konfiguracijama koje odudaraju od konfiguracija na koje je sistem navikao, odnosno na kojima je obučavan.

1.2. Cilj istraživanja

Ciljevi istraživanja su:

1. Istražiti mogućnosti NETCONF protokola u upravljanju konfiguracijama mrežnih uređaja.
2. Analizirati postojeće pristupe za praćenje promena u konfiguracijama mrežnih uređaja.
3. Razviti sistem za automatizovano prikupljanje, čuvanje i poređenje konfiguracionih podataka u cilju otkrivanja grešaka.
4. Ispitati ponašanje sistema u realnom okruženju sa različitim tipovima mrežnih uređaja.
5. Omogućiti vizuelizaciju razlika između konfiguracija tokom vremena radi lakšeg otkrivanja uzroka problema u mreži.

Očekivani rezultati rada su:

1. Razvijeno funkcionalno rešenje za nadzor i analizu konfiguracija mrežnih uređaja korišćenjem NETCONF protokola.
2. Definisan mehanizam za automatizovano otkrivanje promena i mogućih konfiguracionih grešaka.
3. Implementirana baza podataka koja čuva istorijat konfiguracija nadziranih uređaja.
4. Omogućen uvid u trenutno stanje konfiguracije svakog uređaja u realnom vremenu.
5. Realizovan interfejs za poređenje trenutne i očekivane konfiguracije, sa prikazom specifičnih razlika.

1.3. Metodologija

Izrada završnog rada obuhvata teorijski i praktični deo. U teorijskom delu biće sprovedena analiza postojećih rešenja za upravljanje konfiguracijama mrežnih uređaja, sa posebnim fokusom na NETCONF protokol kao savremeni standard za konfiguraciju i automatizaciju upravljanja mrežnim sistemima. Biće obrađeni ključni koncepti kao što su struktura YANG (*Yet Another Next Generation*) modela, XML reprezentacija konfiguracija, kao i mogućnosti koje NETCONF pruža za potrebe validacije i poređenja konfiguracionih podataka. Posebna pažnja posvetiće se izazovima koji nastaju prilikom nadzora većeg broja mrežnih uređaja, posebno u situacijama kada je neophodno pratiti promene kroz vreme kako bi se lakše otkrile greške i odstupanja. Razmotriće se i pristupi za efikasno poređenje XML dokumenata, uključujući upotrebu *XPath* tehnologije. U okviru praktičnog dela rada biće razvijeno rešenje za praćenje i analizu konfiguracija mrežnih uređaja korišćenjem NETCONF protokola. Sistem će omogućavati detekciju konfiguracionih grešaka u realnom vremenu, kao i čuvanje istorije rezultata radi lakše identifikacije ponavljajućih problema i praćenja razvoja konfiguracija kroz vreme. Ovi istorijski podaci će se koristiti kao vredan izvor podataka za prikazivanje određenih rezultata i statistika korisnicima sistema.

Rešenje će sadržati korisnički interfejs (UI) koji omogućava upravljanje i vizuelizaciju rezultata, a biće zasnovano na arhitekturi zasnovanoj na sistemu za razmenu poruka. Takav pristup obezbeđuje veću pouzdanost sistema, horizontalnu skalabilnost i mogućnost asinhronog rada, što je posebno značajno u okruženjima sa većim brojem uređaja i kompleksnijim mrežnim strukturama.

1.4. Struktura rada

Ovaj rad se bavi integracijom nadzora mreže i NETCONF protokola u cilju proaktivnog upravljanja konfiguracijama mrežnih uređaja. U uvodnom delu definiše se oblast rada, iznose se ključne činjenice koje su dovele do ideje za nastanak ovakvog rešenja, objašnjava se cilj rada i navode moguća proširenja. Zatim je priložena metodologija koja je korišćena u istraživanju, kao i struktura samog rada i objavljeni rezultati istraživanja.

Nakon uvodnog dela, u drugom poglavlju, obrađujemo pojam nadzora mreže, značaj nadzora mreže danas, kao i način na koji se razvijao pojam nadzora mreže kroz istoriju pa sve do danas. Prikazan je dijagram opšteg sistema za nadzor mreže uz detaljan opis zadataka koje bi svaka komponenta trebalo da obavlja i navedene su prednosti nadzora sistema uz korišćenje NETCONF protokola.

Treće poglavlje predstavlja osvrt na SNMP protokol, koji je prvenstveno korišćen za nadzor i upravljanje mrežnim infrastrukturama. Opisivanjem svih verzija SNMP protokola koji su nastali vremenom, istaknute su ključne funkcionalnosti kao i mane svake od verzija, naročito u domenu bezbednosti i pouzdanosti komunikacije. Ovim poglavljem dolazimo do zaključka da SNMP, iako je bio standard u nadzoru mreže i ima brojne prednosti, takođe nosi određena ograničenja, naročito u pogledu bezbednosti. Kao protokol koji prevazilazi ograničenja koja je imao SNMP protokol, NETCONF protokol je korišćen u ovom radu prilikom implementacije rešenja i o njemu se govori u narednom poglavlju.

Četvrto poglavlje obrađuje NETCONF protokol kroz objašnjenje kako je NETCONF nastao kao odgovor na ograničenja ranijih rešenja, pre svega SNMP protokola. Predstavljeni su slojevi NETCONF protokola, počevši od transportnog sloja, preko RPC (*Remote Procedure Call*) sloja, do sloja operacija i sadržaja. Istaknuta je bitna uloga YANG jezika kao jezika za modelovanje konfiguracionih podataka, koji omogućava laku manipulaciju konfiguracionim podacima, uz precizno izdvajanje svakog čvora zahvaljujući XML reprezentaciji konfiguracije i *XPath* biblioteci. U okviru ovog poglavlja su obrađene osnovne operacije koje NETCONF podržava, kao što su <get>, <get-config>, <edit-config> i druge. Ključne prednosti NETCONF protokola u poređenju sa SNMP protokolom predstavljaju zaključak ovog poglavlja i ujedno uvod za praktični deo rada.

Peto poglavlje nosi naziv „Sistem za nadzor, analizu i upravljanje konfiguracijama mrežnih uređaja zasnovan na NETCONF protokolu“ i predstavlja detaljan opis projektovanja i implementacije praktičnog dela rada. U uvodnom delu ovog poglavlja, opisani su zahtevi i ciljevi sistema, nakon čega je predstavljeno idejno rešenje sistema i arhitektura predloženog rešenja, kao i opis svih komponenti u arhitekturi. Nakon toga, izdvojene su posebne prednosti kombinacije NETCONF protokola i *Python* programskog jezika, zbog kojih su upravo oni izabrani za implementaciju ovog rešenja. Python poseduje bogate kolekcije biblioteka od kojih je jedna *ncclient*, koja je razvijena posebno za rad sa NETCONF protokolom, odnosno precizno i razumno omogućava preslikavanje XML kodovane NETCONF poruke u podatke razumljive programskom jeziku Python. Zatim se fokus stavlja na arhitekturu implementiranog rešenja uz detaljan opis i prednosti svake komponente. Nakon arhitekture implementiranog rešenja, prelazi se na sam sistem za nadzor, analizu i upravljanje konfiguracijama mrežnih uređaja. Opisuju se sve specifičnosti sistema uz dodatak slika koje prikazuju kako sistem funkcioniše. Na kraju su dati opis i prikaz mobilne aplikacije za ovaj sistem.

Sledeće poglavlje predstavlja diskusiju u kojoj se vrši analiza i poređenje predloženog sistema sa postojećim rešenjima. Predstavljene su sličnosti postojećih rešenja i našeg sistema, prednosti implementiranog sistema kao i ostale mogućnosti proširenja rešenja.

Završno poglavlje predstavlja sumirane rezultate istraživanja i stavlja naglasak na doprinose koje ovaj sistem sa sobom nosi kao i proširenja sistema koja je moguće implementirati u budućnosti kao rezultat budućih istraživanja.

Nakon zaključka, navedena je korišćena literatura.

1.5. Objavljeni rezultati istraživanja

Rezultati istraživanja predstavljeni su u sledećim radovima objavljenim na međunarodnim konferencijama:

[1] B. Martinovic, M. Vuleta, S. Popic and B. M. Todorovic, "Simple verification improvements of the NETCONF data model in programmable computer Networks," *2022 30th Telecommunications Forum (TELFOR)*, Belgrade, Serbia, 2022, pp. 1-4, doi: 10.1109/TELFOR56187.2022.9983688.

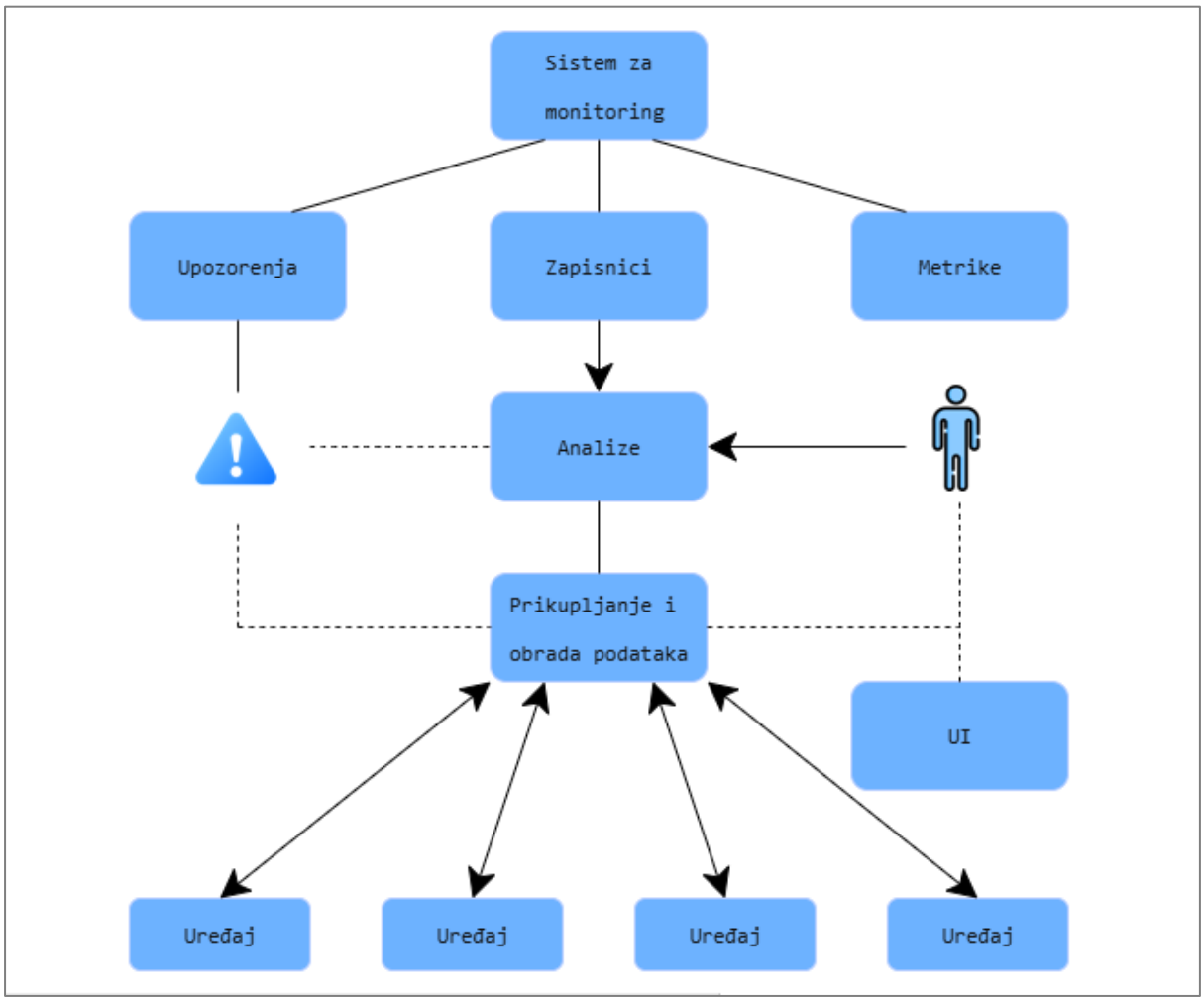
[2] B. Martinović, M. Ljubojević and M. Savić, "Monitoring and Analysis of Network Devices' Configurations Based on the NETCONF Protocol," *2025 24th International Symposium INFOTEH-JAHORINA (INFOTEH)*, East Sarajevo, Bosnia and Herzegovina, 2025, pp. 1-5, doi: 10.1109/INFOTEH64129.2025.10959286.

2. NADZOR MREŽNIH UREĐAJA

2.1. Značaj nadzora mrežnih uređaja

U savremenom digitalnom okruženju, kompanije i organizacije moraju da obezbede neometano funkcionisanje svojih mreža. Nadzor i analiza mreže su neophodni i predstavljaju jedan od ključnih faktora za obezbeđivanje pouzdanog i efikasnog rada mreže. Jedna od osnovnih metoda za postizanje tog cilja jeste kontinuirani nadzor mreže. [1]

Na slici 2.1. dat je uprošćen prikaz većine sistema za nadzor mreže.



Slika 2.1. Dijagram sistema za nadzor mreže

Implementacijom automatizovanih sistema koji mogu vršiti poređenje konfiguracija, kao i pravovremeno obaveštavanje o promenama koje su nastale, značajno se smanjuje vreme reakcije mrežnog administratora na novonastale probleme i povećava se njegova efikasnost. [2]

S obzirom na konstantno povećanje interesovanja i potreba za tehnološkim napretkom, nadzor mrežnih uređaja postao je izuzetno važna stavka u računarskim mrežama. Ranije, održavanje određene mreže bilo je moguće i bez kvalitetno implementiranog sistema za nadzor mreže, jer su to uglavnom bile jednostavne i statične mreže sačinjene od relativno malog broja uređaja, čije je stanje bilo lako pratiti čak i ručno. Danas, računarske mreže povezuju milione uređaja i korisnika širom sveta. Takve mreže su veoma kompleksne i dinamične. U takvom okruženju, prekid koji bi trajao jako kratko, mogao bi izazvati ozbiljne probleme u radu samog sistema. Zato je neophodno razviti adekvatan sistem za nadgledanje mrežnih infrastruktura i uređaja, koji će omogućiti otkrivanje problema u najranijim fazama, kao i dati potrebne informacije za brzo i efikasno rešavanje istih. U idealnom slučaju, potrebno je neprekidno pratiti stanje mrežnih uređaja i pravovremeno reagovati na svaku promenu u njihovom radu. Na taj način se obezbeđuju stabilnost i dostupnost mreže. [3]

Nadzor mreže je težak i zahtevan zadatak za mrežne administratore, jer je neophodno da obezbede i održe nesmetan rad mreže. U zavisnosti od delatnosti kojom se kompanija bavi, postoji veliki rizik da čak i kratki prekidi u velikoj meri ugroze rad, ili negativno utiču na produktivnost. Zato je neophodno da administratori mreže konstantno nadgledaju stanje mreže i mrežnih uređaja, kako bi kada dođe do određenih promena, mogli u najkraćem roku problem izolovati, pa zatim odmah i rešiti. [3]

Nadzor uređaja u mreži se može podeliti na dve vrste praćenja: aktivno i pasivno. Pasivno praćenje podrazumeva nadgledanje mreže bez izmene njenih konfiguracionih parametara, posmatra se samo stanje mreže s ciljem prikupljanja podataka o promenama koje se dešavaju i potencijalnim problemima. Aktivno praćenje mreže uz aktivnosti koje podrazumeva i pasivno praćenje mreže, dodatno omogućava izmenu konfiguracije kako bi stanje uređaja bilo stabilno. Kod jednostavnih konfiguracija koje sadrže malu količinu podataka, lako je koristiti metod aktivnog praćenja mreže ručno, jer je količina praćenih i izmenjenih podataka mala, ali ukoliko se praćenje odnosi na veliki broj mrežnih uređaja sa veoma komplikovanim konfiguracijama koje sadrže mnogo podataka, praćenje ovakve mreže donosi veliki broj prepreka. Nad ovakvim mrežama je nemoguće vršiti praćenje mreže ručno i neophodno je osmisliti dobar sistem koji će vršiti nadzor mreže u zavisnosti od određenih karakteristika mreže i ciljeva koje je potrebno postići ovim nadgledanjem. [3]

Većina arhitektura za nadzor mreže ima sličnu strukturu i odnose među komponentama. Uređaji koji se nadgledaju pokreću softver koji im omogućava da, ukoliko dođe do određenih odstupanja od očekivanog ponašanja, prijave potencijalan problem. Sa druge strane postoji entitet koji je zadužen za rešavanje ovih problema, čiji je zadatak da nakon prijave potencijalnog problema izvrši određene akcije koje će u najboljem slučaju sprečiti nastanak greške ili bar ublažiti njene negativne efekte. U zavisnosti od arhitekture koja se koristi za nadgledanje mreže, entiteti koji su zaduženi za upravljanje mogu periodično ispitivati mrežne uređaje kako bi proveravali stanje periodično. Ovo periodično ispitivanje mreže omogućava čuvanje svih rezultata u bazi podataka, na taj način se mogu dobiti istorijski rezultati uređaja i kvalitetnije se obrađivati potencijalni problemi. Čak i u jako stabilnim mrežama, dužnost mrežnih administratora ostaje kontinuirani nadzor, kako bi se pravovremeno detektovale određene pretnje, koje mogu dovesti do većih problema. [4]

Savremeni alati za nadzor mreže bi trebalo da sadrže sledeće funkcionalnosti[1]:

- Automatizovana upozorenja,
- Detaljni izveštaji,
- Grafički prikazi.

Analiza nadgledane mreže predstavlja proces u kojem se prikupljeni podaci prilikom nadzora mreže interpretiraju radi otkrivanja trendova, dijagnostike problema i predviđanja potencijalnih rizika. Ovaj proces rezultira sledećim postupcima[1]:

- Nadgledanje načina rada sistema,
- Detektovanje nepravilnosti,
- Predlog mera za unapređenje sistema.

Procesom analize mreže dolazimo do nekih dubljih informacija o samom funkcionisanju sistema, što može značajno doprineti donošenju određenih strateških odluka i optimizaciji. U savremenim sistemima, analiza mreže često koristi algoritme mašinskog učenja i veštačku inteligenciju, kako bi se unapredila tačnost predikcija, sprečili prekidi u radu i potencijalno automatizovala rešenja čestih problema.

Organizacije koje koriste tehnike za nadzor i analizu svojih mreža dobijaju bitne informacije o stanju mreže, što omogućava kvalitetnije upravljanje, kao i poboljšavanje performansi. [1]

2.2. Nadzor mrežnih uređaja kroz istoriju

Ranije, na samom početku razvoja računarskih mreža, alati za nadzor mreže su bili prilično jednostavni. Sa pojavom lokalnih mreža (LAN) 1980-ih godina, razvijeni su jednostavni alati za nadzor koji su pomagali u upravljanju mrežom i rešavanju problema u tim infrastrukturama. Ovi alati su bili veoma ograničenih mogućnosti, njihov osnovni cilj je bio da potvrde da se mrežne veze mogu uspostaviti i održavati. Kao rezultat, nudili su veoma malo informacija o stvarnoj dostupnosti mreže i uočenim greškama. 1990-ih godina, potrebe za nadzorom i analizom mreže su rasle proporcionalno složenosti mrežnih sistema, te su ovi alati postali nedovoljni za zadovoljenje tadašnjih potreba. Poboljšanje ovih alata se odnosilo na praćenje i analizu metrike performansi kao što su iskorišćenost propusnog opsega, vreme odziva i saobraćaj. Krajem 1990-ih i početkom 2000-ih, uveden je SNMP protokol, koji je omogućio pribavljanje i organizaciju informacija o upravljanim uređajima preko IP mreža. Do 2005. godine, više od 80% svih mreža u preduzećima koristilo je alate zasnovane na SNMP-u. [1]

Uvođenjem savremenih alata za nadzor mreže kao što su *SolarWinds* i *Nagios*, nadzor i analiza mreže dobijaju nove mogućnosti, kao i grafičke korisničke interfejse, automatska upozorenja i prilagodljive kontrolne table, što je omogućilo nadzor i analizu mreže na znatno naprednijem nivou.

Najnoviji alati za nadzor i analizu mreže podrazumevaju korišćenje mašinskog učenja (ML - *Machine Learning*) i veštačke inteligencije (AI - *Artificial Intelligence*). Zahvaljujući ovim tehnologijama, mrežnim administratorima je omogućeno da prikupljanjem informacija

putem konstantnog nadgledanja mreže mogu da predvide i isplaniraju odgovore pre nego što počnu da utiču na performanse.

Korišćenjem AI/ML algoritma, savremeni alati za nadzor mreže poboljšavaju svoju prediktivnu moć. Moguće je prognozirati rezultate, kao i automatizovati korake koji će predstavljati reakciju na novonastale situacije. Statistike pokazuju da skoro 90% kompanija sa *Fortune 500* liste koristi alate za nadzor mreže koji su unapređeni tehnologijama veštačke inteligencije, kao i da je usvajanje ovih alata dovelo do smanjenja zastoja mreže za oko 30% i povećanja operativne efikasnosti za oko 25%.[1]

Savremeni alati za nadzor mreže bi trebalo da obezbeđuju nadzor uređaja u realnom vremenu, izveštaje koji će sadržavati sve informacije koje mogu biti od koristi za dalju istorijsku analizu podataka i sposobnost da se brzo analizira ogromna količina podataka, kako bi se sa velikim stepenom sigurnosti mogao detektovati svaki problem pravovremeno. [1]

2.3. Nadzor mrežnih uređaja korišćenjem NETCONF protokola

SDN predstavlja inovativan pristup dizajnu mreže, koji donosi novu paradigmu u upravljanju mrežnim sistemima. Ova paradigma se odnosi na razdvajanje logike kontrole mreže od samih fizičkih uređaja. Na taj način se postiže centralizovano upravljanje mrežom i njena programabilnost. [5]

U okviru SDN paradigme, korišćenje NETCONF protokola u kombinaciji sa YANG modulom čije će specifičnosti biti predstavljene u nastavku rada, donosi brojne prednosti kada je u pitanju automatizovana programabilnost i nadgledanje mreže. Ovi standardi čine nadgledanje i programabilnost mreže jednostavnijim pružajući uniformni okvir za konfiguraciju mreže, smanjujući kompleksnost i povećavajući stepen automatizacije. Rizik od grešaka je takođe smanjen zahvaljujući standardizovanim modelima podataka koji omogućavaju konzistentnost i tačnost. Takođe kombinacijom NETCONF protokola i YANG modula poboljšava se pouzdanost i skalabilnost mreže, dok se rizik od ljudskih grešaka u velikoj meri smanjuje, a omogućava se daljinsko upravljanje velikim mrežnim infrastrukturama. [2]

Jedna od bitnih prednosti kada je u pitanju nadzor mrežnih infrastruktura koristeći NETCONF protokol, jeste da on omogućava visoku interoperabilnost, čime se izdvaja od ostalih protokola koji vrše nadzor određenih mrežnih infrastruktura. Naime, NETCONF protokol je u potpunosti nezavisan od konkretnog proizvođača opreme, što znači da se jedan centralizovani sistem za upravljanje, analizu i nadzor konfiguracijama mrežnih uređaja može koristiti za razne tipove uređaja, bez potrebe za razvojem pojedinih softverskih rešenja za različite tipove uređaja. [2]

Još jedna od ključnih funkcionalnosti NETCONF protokola kada je u pitanju nadzor mreže jeste mogućnost filtriranja konfiguracionih podataka korišćenjem *subtree* filtera, koji se zasniva na XML strukturi. XML subtree filtriranje je mehanizam koji omogućava odabir određenog XML podstabla koje će biti uključeno u <rpc-reply> odgovor za <get> ili <get-config> operaciju. *Subtree* filter se sastoji od XML elemenata i njihovih XML atributa.

Pet tipova komponenti koje mogu biti prisutne u subtree filteru[6]:

- **Namespace Selection** - selekcija po prostoru imena
- **Attribute Match Expressions** – omogućavaju filtriranje čvorova na osnovu vrednosti XML atributa
- **Containment Nodes** – omogućavaju hijerarhijsko pretraživanje (čvorovi koji sadrže druge čvorove)
- **Selection Nodes** – omogućavaju tačno selektovanje čvorova
- **Content Match Nodes** – omogućavaju precizno dohvatanje na osnovu sadržaja čvora

3. SNMP

Krajem 1980-ih godina, IETF grupa razvila je SNMP protokol. Ova grupa je odgovorna za standardizaciju i definisanje protokola i arhitektura koje omogućavaju postojanje i razmenu podataka putem Interneta. SNMP protokol definiše skup standarda za upravljanje mrežama, uključujući aplikacioni protokol, šemu baze podataka i definisane skupove podataka. SNMP vrši prikupljanje, organizaciju kao i modifikaciju informacija o konfiguraciji uređaja u mrežama.

Prva verzija SNMP protokola je **SNMPv1** [7], koje je definisana u RFC (*Request for Comments*) 1155, 1156 i 1157. Ova verzija SNMP protokola je imala velike nedostatke kao što je nemogućnost prenosa velike količine podataka, kao i značajne nedostatke u vezi bezbednosti same komunikacije. Čitav model bezbednosti se zasniva na tzv. *community stringovima* – nizovima karaktera koji funkcionišu kao deljene lozinke i zajednički pristupni ključevi. Ovakav princip ne omogućava ni šifrovanje, kao ni detaljnu kontrolu pristupa, što znači da je bezbednost na vrlo niskom nivou. Uprkos svim nedostacima koje sa sobom nosi SNMPv1, ovaj mrežni protokol se i dalje koristi, naročito od strane jednostavnijih ili jeftinijih uređaja, jer ga podržava veliki broj proizvođača mrežne opreme.

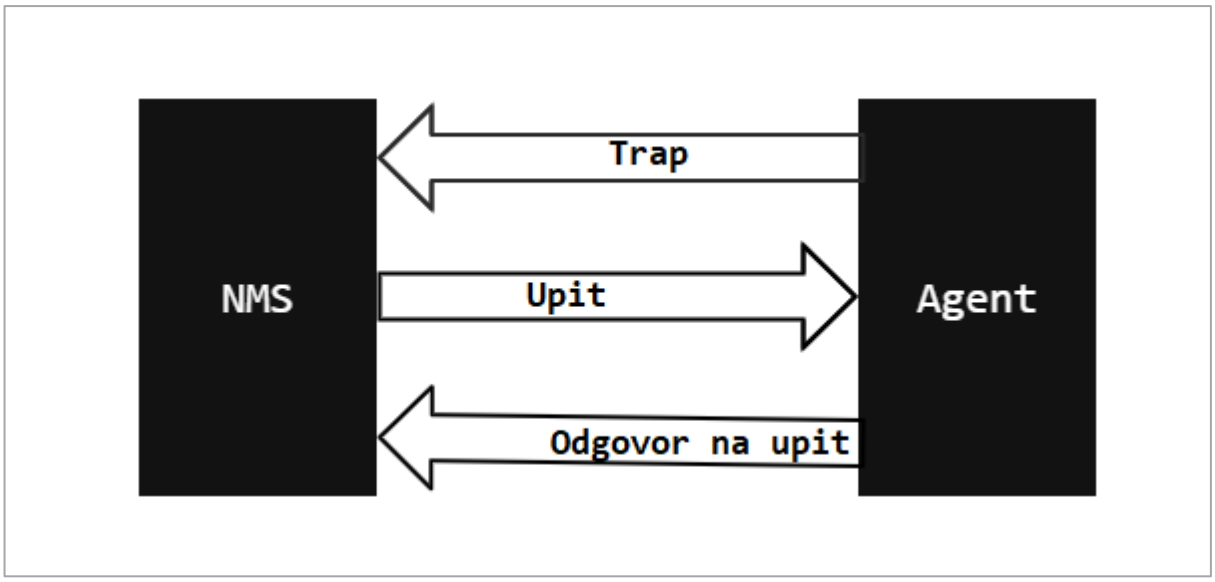
Vremenom, razvijene su nove verzije SNMP protokola – **SNMPv2** [8] i **SNMPv3** [9], koje su donele dodatna funkcionalna poboljšanja. SNMPv2 je definisana u RFC 1441-1452, a poboljšanja koje ova verzija sa sobom nosi se uglavnom tiču bezbednosnog modela, ali i performansi, pre svega uvođenjem GETBULK operacije. SNMPv3, definisan u RFC 3411-3418, predstavlja aktuelnu i trenutno najbolju verziju, naročito kada se u obzir uzmu unapređenja u oblasti bezbednosti. SNMPv3 verzija predstavlja jedinu zvaničnu verziju standarda u skupu IP protokola koju priznaje IETF i nosi naziv STD0062. Nastankom treće verzije SNMP protokola, obezbeđena je autentifikacija, poverljivost i kontrola pristupa. [10]

U komunikaciji SNMP protokolom postoje dve vrste entiteta:

- menadžer i
- agent.

Uređaji upravljani od strane SNMP protokola sadrže proces koji predstavlja SNMP agenta koji se kontinuirano izvršava u pozadini. Primarna uloga SNMP agenta jeste da prikuplja podatke sa uređaja nad kojim se vrši upravljanje i da omogući računaru koji vrši nadzor i upravljanje da im pristupa. Ova komponenta koja vrši nadzor i upravljanje je menadžer i uglavnom se naziva *Network Management Station* (NMS). NMS je odgovoran za prozivanje uređaja (engl. *polling*) i primanje obaveštenja (engl. *trap/inform*) od agenata u mreži. *Prozivanje* uređaja podrazumeva slanje upita agentu (koji se izvršava na ruteru, sviču itd.) za neku informaciju, dok je *obaveštenje* način da agent obavesti NMS da se nešto dogodilo. Slanje obaveštenja se izvršava asinhrono, a ne kao odgovor na upite NMS-a. Odgovornost NMS-a je da na osnovu informacija koje dobije od agenta izvrši radnje po potrebi. [11]

Na slici 3.1. predstavljena je veza između NMS-a i agenta.



Slika 3.1. Veza između NMS-a i agenta [11]

Bitno je napomenuti, da se prozivanja i obaveštenja mogu desiti istovremeno, odnosno da se ova komunikacija izvršava asinhrono i da ne postoje ograničenja kada NMS može poslati upit agentu, ili kada agent može da pošalje obaveštenje.

Naredbe koje NMS koristi kako bi vršio funkcije dobavljanja i modifikacije informacija na SNMP agentu su: *set* i *get*, dok upravljani uređaji mogu započeti komunikaciju ka glavnom sistemu putem operacija *trap* ili *inform*. Na ovaj način, upravljani uređaji šalju informacije o promenama i događajima koji su se pojavili. [10]

S obzirom da agenti često imaju ograničenu funkcionalnost, može se desiti da određene korisničke potrebe ne budu podržane od strane agenta. U tim situacijama neophodno je proširiti njegove funkcionalnosti. Ovo je moguće uraditi na tri načina [12]:

1. Izmena postojećeg agenta – Ovakav pristup podrazumeva direktno dodavanje potrebnih funkcionalnosti u kod agenta. Iako ovaj pristup omogućava veliku efikasnost, on sa sobom nosi određeni niz poteškoća: nedostupnost izvornog koda, pravna ograničenja vezana za autorska prava, kao i prilagođavanje nove funkcionalnosti postojećem agentu.

2. Upotreba posebnog, eksternog programa – razvijanje dodatnog softvera, koji se povremeno ili konstantno izvršava i dostavlja podatke originalnom agentu. Program implementiran na ovaj način uglavnom nije univerzalno upotrebljivo rešenje i zavisi od specifičnosti agenta.

3. Upotreba SNMP AgentX protokola – mehanizam koji omogućava *master-slave* odnos agenata, čime se omogućava proširenje funkcionalnosti uz poštovanje svih standarda, uz mogućnost kreiranja univerzalnog rešenja za sve SNMP agente.

Podaci kojima se pristupa u okviru SNMP-a, predstavljeni su strogom hijerarhijskom strukturom nalik stablu, koju čine virtuelne informacione baze za upravljanje, poznate kao MIB (*Management Information Base*). Skupovi povezanih podataka definišu se u MIB modulima.

Svaki podatak u ovoj strukturi ima svoj jedinstveni identifikator, koji se zove OID (engl. *Object Identifier*). Na vrhu ove strukture, nalazi se neimenovani koren, od kojeg polaze grane dodeljene glavnim međunarodnim organizacijama zaduženim za standardizaciju, kao što su ISO i ITU-T, nakon čega se grana na različite organizacione jedinice koje standardizuju i definišu MIB module.[12]

U okviru SMIv2 (*Structure of Management Information version 2*), definicije OID opisuju kako se hijerarhijski gradi celokupni prostor upravljanih objekata u SNMP arhitekturi. U tabeli 3.1 predstavljena je hijerarhijska organizacija OID stabla u SNMPv2-SMI. [13]

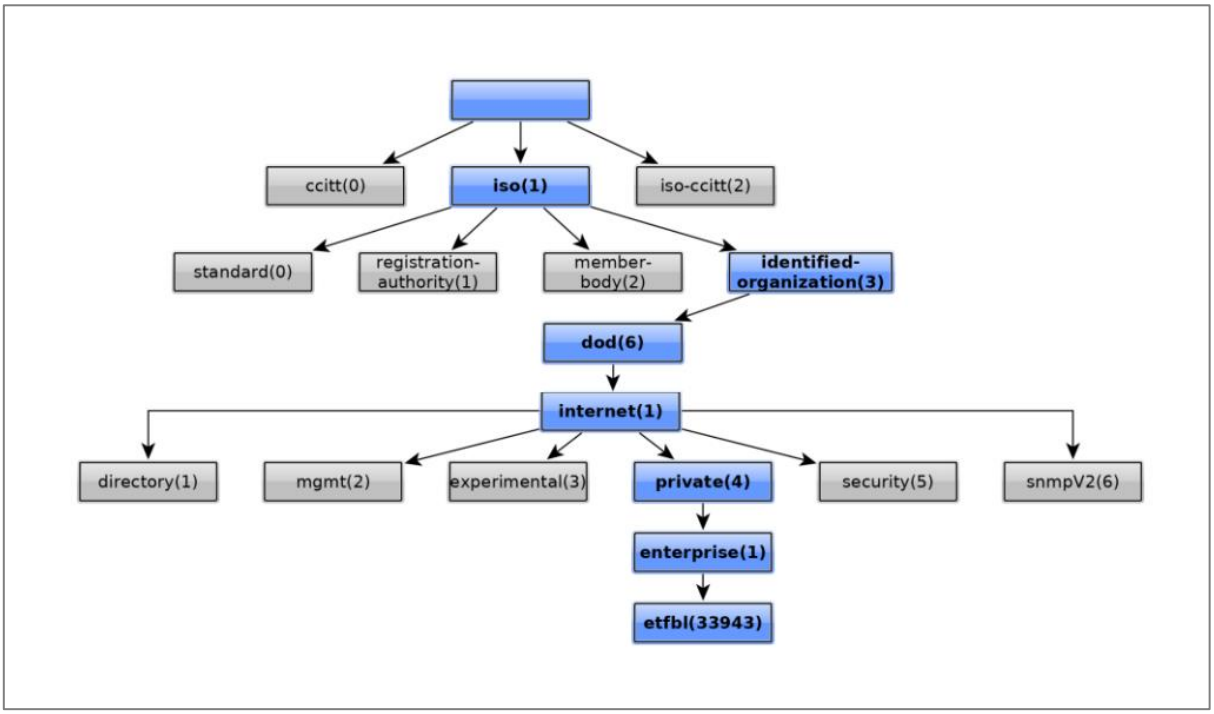
NAZIV ČVORA	OID DEFINICIJA	OPIS
org	{ iso 3 }	Organizacije pod ISO (iso = 1), OID: 1.3
dod	{ org 6 }	Department of Defense, OID: 1.3.6
internet	{ dod 1 }	Osnovna grana za internet-relativne objekte, OID: 1.3.6.1
directory	{ internet 1 }	Direktorijumske usluge
mgmt	{ internet 2 }	Standardne MIB definicije
mib-2	{ mgmt 1 }	Standardni MIB moduli (1.3.6.1.2.1)
transmission	{ mib-2 10 }	Prenosni protokoli
experimental	{ internet 3 }	Eksperimentalna upotreba
private	{ internet 4 }	Prostor za proizvođače i privatne entitete
enterprises	{ private 1 }	Identifikatori registrovanih pravnih lica, OID: 1.3.6.1.4.1
security	{ internet 5 }	Bezbednosni modeli i mehanizmi
snmpV2	{ internet 6 }	Proširenja vezana za SNMPv2
snmpDomains	{ snmpV2 1 }	Transportni domeni
snmpProxys	{ snmpV2 2 }	Transportni proksi
snmpModules	{ snmpV2 3 }	SNMP moduli

Tabela 3.1. Hijerarhijska organizacija OID stabla u SNMPv2-SMI. [13]

Identifikator registrovanih pravnih lica 1.3.6.1.4.1, rezervisan je za dodelu identifikatora pravnim subjektima kao što su proizvođači mrežne opreme, organizacije, univerziteti itd. Svaki entitet koji želi da dobije svoj jedinstveni identifikator OID, mora proći formalni proces registracije. Na sledećoj slici prikazan je primer OID za Elektrotehnički fakultet Univerziteta u Banjoj Luci, koji je nakon registracije dobio na upravljanje jedinstvenu granu sa identifikatorom: 1.3.6.1.4.1.33943, odnosno[12]:

.iso.org.dod.internet.private.enterprise.etfbl

OID hijerarhija koja prikazuje označenu putanju do korena grane dodeljene Elektrotehničkom fakultetu je prikazana na slici 3.2. [12]



Slika 3.2. Primer hijerarhije OID-a [12]

Podaci opisani u okviru MIB strukture pripadaju jednoj od sledećih kategorija[12]:

- Skalari – podaci koji predstavljaju jednu instancu objekta.
- Tabele – nula ili više međusobno povezanih instanci objekata organizovanih u tabelarnu strukturu.

U okviru SNMP-a, definisanje podataka vrši se kroz MIB module, koji koriste ASN.1 sintaksu. Ovakvo predstavljanje je semantički dosta zahtevno i ne omogućava veliku fleksibilnost, što ga čini nepovoljnijim rešenjem za upravljanje mrežom, već je pogodno za statične i jednostavne modele podataka.

U skladu sa RFC 2578, primer modelovanja jednog podatka u okviru MIB strukture prikazan je na sledećoj slici [13]:

```
HELLO-WORLD-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE, DisplayString
      FROM SNMPv2-SMI;

helloWorld MODULE-IDENTITY
   LAST-UPDATED "202407310000Z"
   ORGANIZATION "Example Organization"
   DESCRIPTION
      "MIB - 'Hello, World!'"
   ::= { enterprises 99999 }

system OBJECT IDENTIFIER ::= { helloWorld 1 }

message OBJECT-TYPE
   SYNTAX     DisplayString (SIZE (0..255))
   MAX-ACCESS  read-only
   STATUS     current
   DESCRIPTION
      "The 'Hello, World!' message."
   ::= { system 1 }

END
```

Slika 3.3. Primer modelovanja podataka kroz SNMP MIB.

Zbog mnogobrojnih nedostataka koje nosi SNMP protokol, kao što su kompleksnost modelovanja podataka i nedostatak bezbednosti jer ovaj protokol najčešće koristi UDP protokol, preko podrazumevanih portova 161 na strani agenta i 162 na strani glavnog uređaja, SNMP protokol se uglavnom koristi za nadzor mreža (engl. *network monitoring*), ali ne i za napredno i pouzdano upravljanje konfiguracijom uređaja u mreži. [10]

4. NETCONF PROTOKOL

Računarske mreže i Internet danas se zasnivaju na TCP/IP grupi protokola i mrežnim uređajima (ruteri, svičevi, itd.). Upravljanje ovakvim tipovima uređaja najčešće se obavlja putem komandne linije (CLI – engl. *Command Line Interface*), kao sopstvenog interfejsa operativnog sistema proizvođača. Ovakav pristup sa sobom, pored brojnih pogodnosti kao što su hijerarhijska organizacija naredbi, komandne linije i grupisanje srodnih komandi takođe nosi i određene probleme. Za ovakav pristup, neophodno je imati stručne mrežne administratore koji imaju iskustvo ili su obučeni za rad u komandnoj liniji ovog proizvođača operativnog sistema, jer se strukture komandi i podataka razlikuju u zavisnosti od proizvođača operativnog sistema. [14]

U RFC 3535 iz 2002. godine uočena su velika ograničenja tadašnjih rešenja, naročito SNMP protokola koji se tada koristio kao glavni protokol za nadzor i upravljanje mrežnim uređajima.

Ključni problemi koji su identifikovani u RFC 3535 doveli su do određenih zahteva [15]:

1. Jednostavnost korišćenja – ključni zahtev za bilo koju tehnologiju upravljanja mrežom iz ugla operatera.
2. Jasna razlika između konfiguracionih podataka, podataka koji opisuju operativno stanje i statističkih podataka.
3. Zaseban pristup konfiguracionim podacima, operativnom stanju i statistici sa uređaja, kao i mogućnost međusobnog poređenja ovih podataka između različitih uređaja.
4. Fokus na konfiguraciju mreže kao celine, a ne pojedinačnih uređaja.
5. Podrška za konfigurisanje više uređaja istovremeno – značajno bi se pojednostavilo upravljanje mrežnom konfiguracijom.
6. Mogućnost generisanja niza operacija koje prelaze sa jedne konfiguracije uređaja na drugu, uz minimalne promene stanja i što manji uticaj na mrežu i sisteme.
7. Standardizovan uvoz i izvoz konfiguracija.
8. Jednostavnost izvršavanja provere konzistentnosti konfiguracija tokom vremena.
9. Standardizacija centralizovanih baza podataka za mrežne konfiguracije.
10. Podrška za tekstualne alate (diff, SVN, git).

11.	Precizna kontrola pristupa – po ulozi i potrebi.
12.	Provera konzistentnosti ACL-ova na više uređaja.
13.	Razdvajanje distribucije i aktivacije konfiguracija.
14.	Podrška za oba modela kontrole pristupa: po zadatku i po podacima.

Na osnovu ovih zahteva i preporuka, 2006. godine od strane IETF grupe razvijen je i standardizovan protokol za upravljanje mrežom pod nazivom NETCONF. Cilj nastanka ovog protokola je bio da se potrebe mrežnih operatera i proizvođača opreme usklade. Prva verzija NETCONF protokola je objavljena 2006. godine kao RFC 4741. Kasnije je revidiran u junu 2011. godine i objavljen kao RFC 6241. [15]

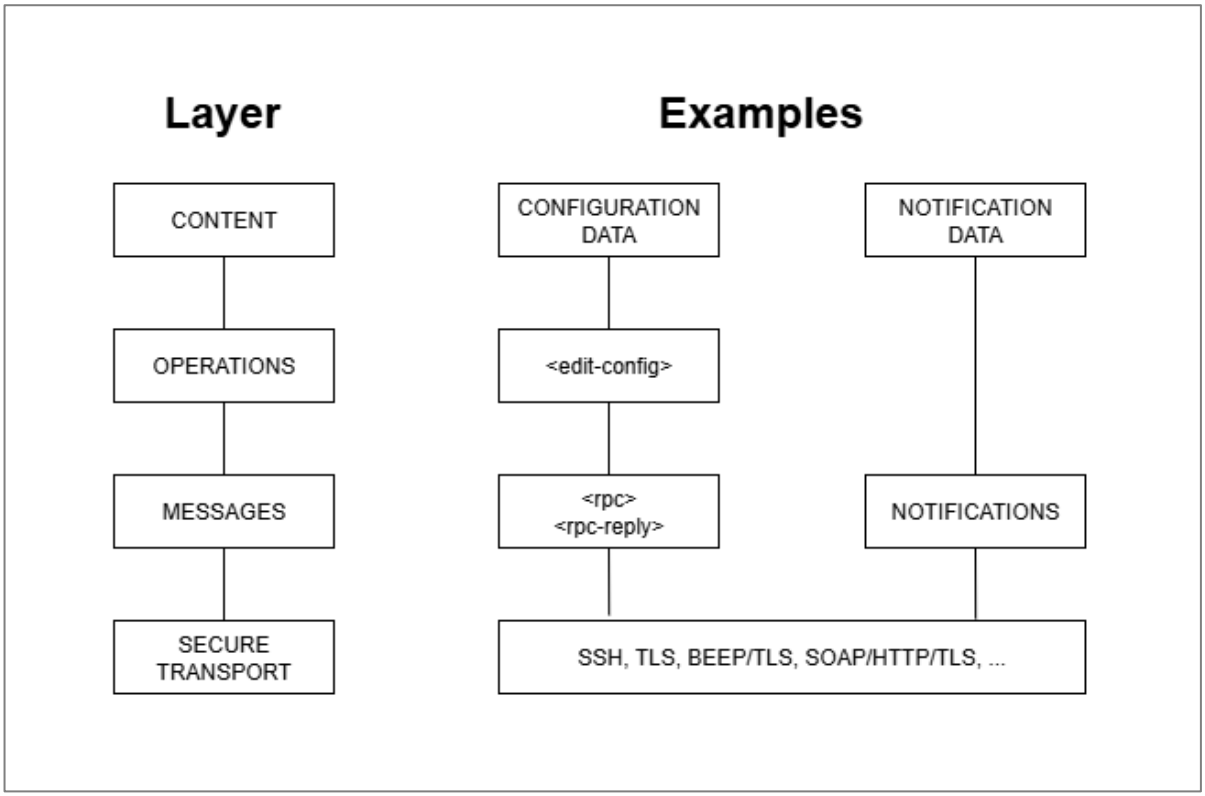
NETCONF je protokol namenjen za upravljanje konfiguracijama uređaja koje podrazumeva preuzimanje informacija o konfiguraciji, učitavanje i manipulaciju novim konfiguracionim podacima, kao i brisanje konfiguracije uređaja. Korišćenjem NETCONF protokola, aplikacije mogu koristiti potpuno formalizovan API za slanje i prijem konfiguracija, ili određenog dela konfiguracije. NETCONF koristi jednostavan sloj poziva udaljenih procedura (RPC), softverski komunikacioni protokol koji omogućava programerima da pišu isti programski kod nezavisno da li se potprogram izvršava lokalno ili na nekoj udaljenoj lokaciji. U ovoj komunikaciji najčešće je klijent skripta ili aplikacija, dok je server, uglavnom, mrežni uređaj. [6]

Svi NETCONF zahtevi i odgovori moraju biti napisani u ispravnom XML formatu kodiranom u UTF-8. Ako NETCONF server primi nevalidnu <rpc> poruku, trebalo bi da odgovori greškom *„malformed-message“*. Nevalidna <rpc> poruka je poruka koja nije validna XML struktura ili nije kodirana u UTF-8 formatu. Ako nije moguće poslati odgovor, server mora prekinuti NETCONF sesiju. [6]

4.1. Slojevi NETCONF protokola

NETCONF protokol se sastoji od četiri sloja (slika 4.1) [14]:

1. **Sloj sigurnog prenosa** (engl. *Secure transport*)
2. **Sloj poruka** (engl. *Messages*)
3. **Sloj operacija** (engl. *Operations*)
4. **Sloj sadržaja** (engl. *Content*)



Slika 4.1. Slojevi NETCONF protokola [14]

Sloj sigurnog prenosa omogućava sigurnu i pouzdanu komunikaciju između klijenta i servera. NETCONF protokol može koristiti bilo koji transportni protokol, dok god on zadovoljava sledeće uslove:

- **Autentifikacija** – proces verifikacije identiteta učesnika u komunikaciji.
- **Integritet** – zaštita od neovlašćenih izmena i kompromitovanja tokom prenosa.
- **Tajnost** – zaštita sadržaja poruke tako da ona ostane dostupna isključivo autorizovanim stranama.
- **Neporecivost** – učesnici u komunikaciji ne mogu da poreknu naknadno da su učestvovali u komunikaciji.

S obzirom da NETCONF protokol koristi transportni protokol, smatra se da se na taj način obezbeđuje odgovarajući nivo bezbednosti i poverljivosti prilikom komunikacije. Transportni protokol ima odgovornost da obezbedi autentifikaciju u oba smera: od servera prema klijentu i od klijenta prema serveru. Iz tog razloga, NETCONF protokol pretpostavlja da je autentifikacija proverena od strane transportnog sloja i da je identitet komunikacionog

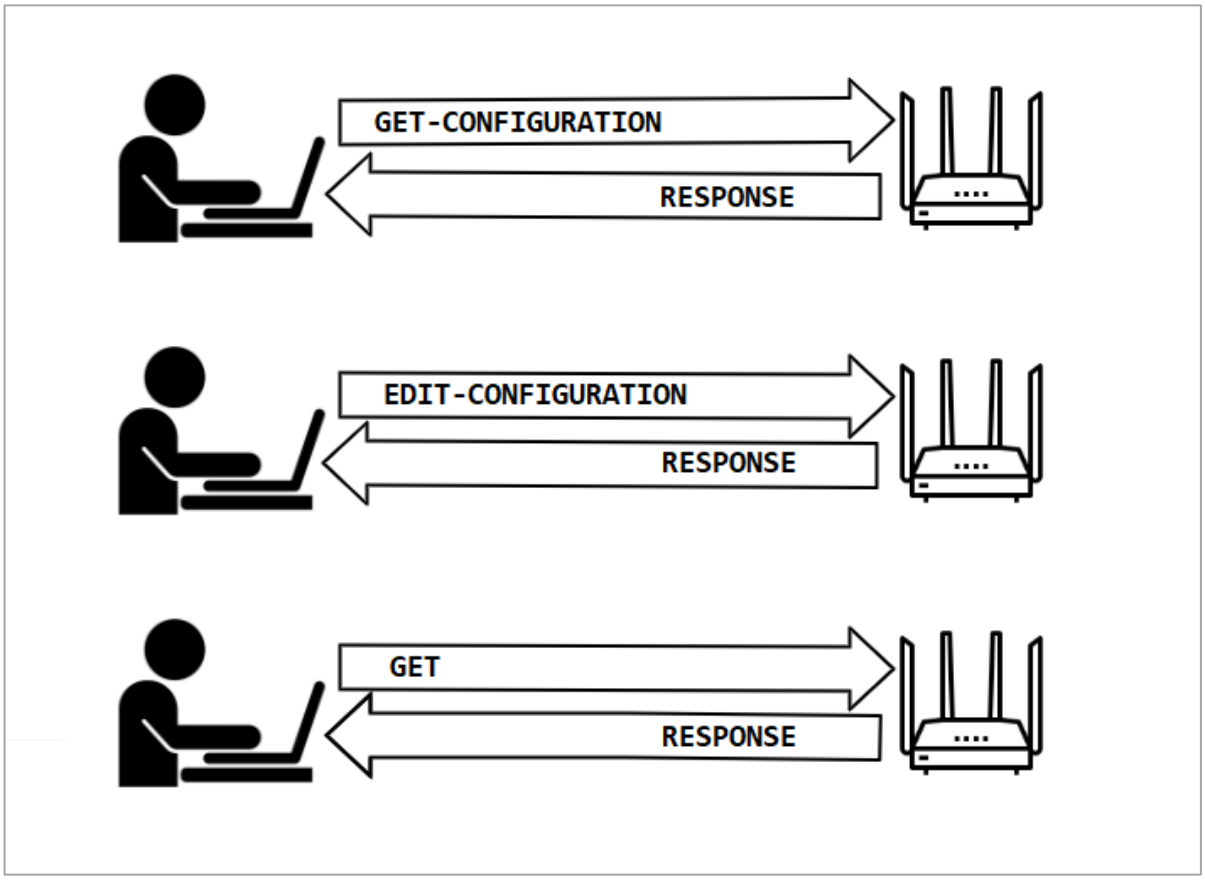
partnera pouzdano potvrđen. [6] Obavezna implementacija sigurnog prenosa podrazumeva „NETCONF preko SSH-a“, definisana RFC 6242.

Sloj poruka opisuje mehanizam kodovanja udaljenih poziva procedura i notifikacija. Ovaj sloj omogućava nezavisan i jednostavan mehanizam kodovanja poruka u odnosu na transportni sloj.

Osnovne NETCONF poruke su:

- RPC zahtev (<rpc> poruka),
- RPC odgovor (<rpc-reply> poruka),
- notifikacija (<notification> poruka).

Primer komunikacije NETCONF protokolom je ilustrovan na slici 4.2.



Slika 4.2. Primer komunikacije NETCONF protokolom

Sloj operacija definiše skup osnovnih operacija protokola koje omogućavaju dohvatanje i modifikovanje konfiguracionih podataka: *<get>*, *<get-config>*, *<edit-config>*, *<copy-config>*, *<delete-config>*, *<lock>*, *<unlock>*, *<close-session>*, *<kill-session>*.

Sloj sadržaja se sastoji od konfiguracionih podataka i podataka o notifikacijama. NETCONF protokol se ne bavi direktno sadržajem ovog sloja, već je za tu svrhu razvijen YANG jezik za modelovanje. [14]

Informacije koje je moguće dobiti za neki uređaj mogu se podeliti na dve grupe:

- podaci o konfiguraciji i
- podaci o stanju.

Podaci o konfiguraciji su promenljivi i definišu kako je uređaj podešen. To su podaci koji su potrebni da sistem pređe iz početnog, podrazumevanog stanja u trenutno stanje. Podaci o stanju su dodatni podaci o stanju uređaja. Ovi podaci se ne mogu konfigurisati i uglavnom služe za dublju analizu stanja, dijagnostiku i monitoring. Kada bi se prilikom izvršavanja nekih operacija koristili podaci koji podrazumevaju pored konfiguracionih podataka i podatke stanja, došlo bi do određenih problema. Skupovi podataka bi bili veliki i takve podatke je teško obraditi, dolazni podaci bi sadržavali nepotrebne informacije, poput različitih statistika. Nad ovakvim skupom podataka je vršenje poređenja znatno otežano, jer bi bilo idealno da se porede samo podaci koji igraju konfiguracionu ulogu za taj uređaj.

NETCONF protokol prevazilazi prethodno navedene probleme, jer jasno pravi razliku između konfiguracionih podataka i podataka o stanju. Za oba tipa podataka ima posebne operacije koje vrše dohvatanje željenih podataka. Operacije NETCONF protokola su detaljnije objašnjene u 4.3. [14]

4.2. YANG modul

YANG je jezik za modelovanje podataka za NETCONF protokol sa ciljem upravljanja konfiguracijama. Nastao je u oktobru 2010. godine od strane radne grupe NETMOD (*NETwork data MODel*) u okviru IETF-a organizacije. Koristi se za definisanje semantike operativnih podataka, konfiguracionih podataka, notifikacija i operacija. Ovaj jezik je definisan u RFC 6020, dok su tipovi podataka opisani u RFC 6021. YANG prikazuje strukturu podataka u formi XML stabla u kojem svaki čvor ima svoje ime i vrednost, ili grupu potčinjenih čvorova. Ovim je omogućen jasan i sažet opis svakog čvora u hijerarhiji, kao i njihovu međusobnu interakciju. [14]

YANG modeli su organizovani u module i podmodule, koji mogu uvoziti podatke iz drugih eksternih modula, ili uključivati podatke iz sopstvenih podmodula. Liste podataka mogu imati definisane ključeve, takođe se nad listama može vršiti određeni proces sortiranja od strane korisnika ili sistema, kao i upravljati redosledom elemenata u listi. Ovi modeli mogu sadržavati određena ograničenja nad podacima, uslovne strukture i pravila validacije. YANG definiše skup predefinisanih tipova, ali je moguće i definisanje novih tipova. Ukoliko se definišu novi tipovi, za njih je moguće dodatno ograničiti validne vrednosti osnovnog tipa kao i definisati konvencije korišćenja. [16]

YANG moduli se mogu prevesti u ekvivalentnu XML formu, što predstavlja veoma važnu karakteristiku YANG modula, jer će u ovom formatu imati mogućnost korišćenja bogatog skupa XML alata kao pomoć pri filtriranju, validaciji, generisanju koda, dokumentacije i slično. Ekvivalentna XML forma za YANG modul se zove *YANG Independent Notation* (YIN). Bitna karakteristika YIN forme u poređenju sa YANG-om jeste da sadrže ekvivalentne informacije samo predstavljene različitim notacijama. Konverzija iz YANG u YIN je bez gubitka informacija, što znači da se sadržaj iz YIN forme može bez ikakvog rizika vratiti u izvorni YANG. Međusobne veze između YANG-a i YIN-a su takve da je ime YIN elementa ekvivalentno YANG ključnoj reči, dok su argumenti YANG iskaza u YIN formatu predstavljeni kao XML atributi. Iz ovog možemo zaključiti da će svaki korenski element YIN dokumenta uvek biti element <module>. Svaki od ovih YIN elemenata pripada imenskom prostoru (engl. *namespaces*) čiji je URI jednak „*urn:ietf:params:xml:ns:yang:yin:1*“. Predstavljanjem podataka u YIN formi, dobijamo XML formu koja nam pruža mogućnost korišćenja bogatog skupa XML alata kao pomoć prilikom filtriranja, validacije, generisanja određenog koda i slično. Bitna stavka za XML format je i prilično dobra čitljivost za čoveka, a istovremeno vrlo pogodan za korišćenje u računarskom svetu. [16]

Kao i NETCONF protokol, YANG je dizajniran za laku integraciju sa sopstvenom infrastrukturom za upravljanje uređajem, što znači da implementacije mogu iskoristiti postojeće mehanizme za kontrolu pristupa kako bi zaštitile ili omogućile pristup određenim podacima. [16]

Na sledećoj slici je predstavljen uporedni prikaz YANG, YIN i XML formi koji sadrže jedan kontejner sa jednim list podatkom:

YANG:

```
container system {
     leaf message {
        type string;
        description "The 'Hello, World!' message.";
     }
   }
"
```

YIN:

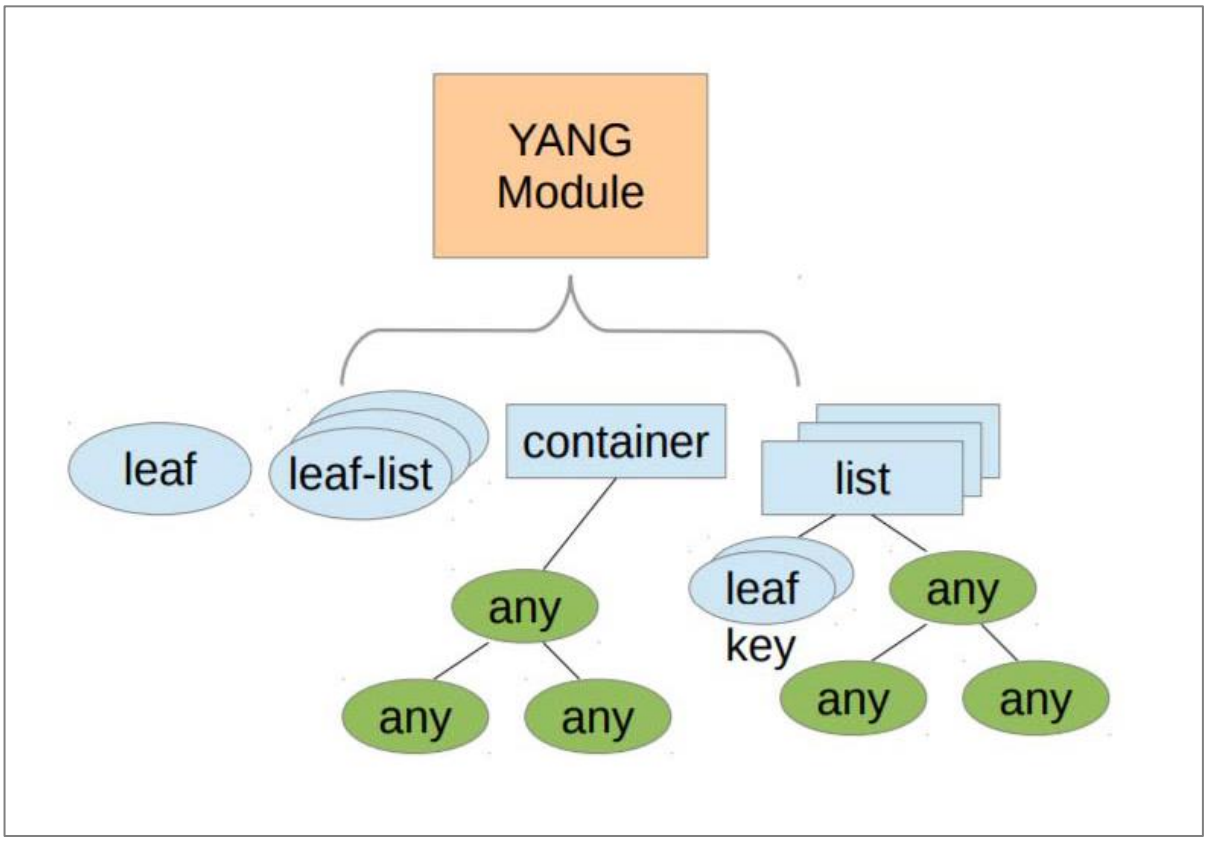
```
<container name = "system">
     <leaf name = "message">
        <type name = "string"/>
        <description>
           <text>The 'Hello, World!' message.</text>
        </description>
     </leaf>
  </container>
```

XML:

```
<system>
     <message>Hello, World!</message>
</system>
```

Slika 4.3. Uporedni prikaz YANG, YIN i XML formi

YANG sve svoje podatke smešta u *leaf* elemente XML stabla. Leaf predstavlja jednostavan čvor koji ima jednu vrednost. Lista svih elemenata koji imaju pojedinačnu vrednost definiše se pomoću *leaf-list* izraza. Srednji čvorovi u XML dokumentu se definišu pomoću *container* elementa. Za definisanje složenijih struktura, koje sadrže više podelemenata kao što su leaf, leaf-list, container i druge složenije strukture, koristi se izraz *list*. Primer jedne strukture YANG modula, prikazan je na slici 4.4.



Slika 4.4. Struktura YANG modula [17]

S obzirom da realni uslovi nisu isti kao idealni, u većini situacija uređaji ne implementiraju model tačno onako kako je definisan, uglavnom postoje određena odstupanja. Zato je potrebno da ukoliko uređaji nemaju mogućnost da implementiraju čitav model tačno onako kako je definisan, dokumentovati takva odstupanja. YANG ima mehanizam za odstupanje – *deviate*, koji omogućava formalno dokumentovanje takvih odstupanja i njihovo prenošenje aplikacijama za upravljanje. [18]

Zbog svih navedenih karakteristika koje nam pruža NETCONF protokol u kombinaciji sa YIN-om i YANG-om, zaključujemo da se suština predstavljanja konfiguracionih podataka koristeći ovaj protokol svodi na XML formu. Da bi XML forma bila validna, potrebno je da njeni podaci budu validni, odnosno da budu usklađeni sa XML šemom (engl. *XML Schema*), koja definiše strukturalna ograničenja i restrikcije atributa. Ukoliko XML dokument poštuje ograničenja definisana XML šemom, taj XML dokument se smatra validnim XML dokumentom. [19]

Validacija XML-a se može postići na dva načina:

- Inkrementalna validacija,
- Ponovna validacija.

Inkrementalna validacija podrazumeva proveru samo izmenjenih delova podataka. Ovaj pristup smanjuje vreme koje je potrebno za validaciju, jer se posmatra samo deo XML-a, a ne čitav dokument. [19]

Ponovna validacija podrazumeva proveru čitavog XML dokumenta i zahteva da XML dokument bude definisan u skladu sa XML šemom, kao i da podaci budu logički konzistentni, uključujući funkcionalne zavisnosti, pravila domena i druge specifične uslove. [20]

4.3. Osnovne operacije NETCONF protokola

NETCONF protokol definiše skup osnovnih operacija koje omogućavaju čitanje, modifikaciju, kopiranje i brisanje konfiguracionih podataka, kao i upravljanje sesijama. Dodatne operacije koje NETCONF takođe može oglasiti prilikom uspostavljanja sesije nazivaju se NETCONF sposobnosti (engl. *capabilities*). Nazivi ovih sposobnosti moraju biti izraženi u obliku URI (*Uniform Resource Identifier*) adresa.

U tabeli 4.1. predstavljene su sve operacije NETCONF protokola. [6]

OPERACIJA	OPIS
<get>	Preuzimanje *running* konfiguracije i informacija o stanju uređaja.
<get-config>	Preuzimanje čitave konfiguracije ili samo dela konfiguracije iz skladišta podataka.
<edit-config>	Izmena konfiguracije. Izmenu je moguće uraditi kreiranjem, brisanjem, spajanjem ili zamenom određenog sadržaja.
<copy-config>	Kopiranje čitavog skladišta konfiguracionih podataka u drugo skladište konfiguracionih podataka.
<delete-config>	Brisanje skladišta konfiguracionih podataka.
<lock>	Zaključavanje skladišta konfiguracionih podataka.
<unlock>	Otključavanje skladišta konfiguracionih podataka.
<close-session>	Zatvaranje aktivne NETCONF sesije.
<kill-session>	Prisilno prekidanje NETCONF sesije.

Tabela 4.1. Osnovne operacije NETCONF protokola [6]

Sve operacije ovog protokola se šalju kao RPC zahtevi u XML formatu, a odgovori mogu biti:

- Pozitivan odgovor: <rpc-reply> ,
- Negativan odgovor: <rpc-error>.

Kod izvršavanja osnovnih operacija NETCONF protokola treba se uzeti u obzir da uspešnost izvršenja operacije nije garantovana, što znači da je potrebno proveriti sadržaj odgovora nakon svake operacije. [6]

4.3.1. Dobavljanje konfiguracije

Operacija **<get-config>** služi za dobavljanje čitave konfiguracije ili specifičnog dela konfiguracije i prima dva parametra[6]:

- *Source* parametar predstavlja izvor, odnosno ime konfiguracije za koju se postavlja upit.

- *Filter* parametar predstavlja deo konfiguracije od interesa, ukoliko se on ne unese, podrazumeva se da se radi o čitavoj konfiguraciji. Ovaj parametar opciono može sadržati atribut *type*, koji predstavlja tip sintakse filtriranja koji se koristi. Podrazumevani tip filtriranja u NETCONF-u se naziva filtriranje podstabla (engl. *subtree filtering*). Na ovaj način se jednostavno može izabrati nula i više elemenata podstabla koji će činiti deo konfiguracije koji će biti dohvaćen.

Rezultat izvršavanja <get-config> operacije može biti pozitivan ili negativan. Ukoliko je uređaj u stanju da zadovolji zahtev, server šalje element <rpc-reply> kao odgovor koji sadrži <data> element sa rezultatima upita. Nasuprot tome, ako uređaj nije u stanju da zadovolji zahtev iz bilo kog razloga, server šalje element <rpc-reply> kao odgovor koji sadrži <rpc-error>, koji definiše grešku. [6]

Operacija **<get>** vrši preuzimanje running konfiguracije uređaja zajedno sa informacijama o stanju uređaja. Ova operacija prima jedan parametar:

- *filter* određuje konkretni deo konfiguracije uređaja i informacija o stanju uređaja koji se trebaju preuzeti. Ovaj parametar je opcioni, ukoliko nije naveden, njegovo podrazumevano stanje govori da treba dohvatiti čitavu konfiguraciju sa svim informacijama o stanju.

Rezultat izvršavanja <get> operacije može biti pozitivan ili negativan. Ukoliko je uređaj u stanju da zadovolji zahtev, server šalje element <rpc-reply> kao odgovor koji sadrži <ok> element. Nasuprot tome, ako uređaj nije u stanju da zadovolji zahtev iz bilo kog razloga, server šalje element <rpc-reply> kao odgovor koji sadrži <rpc-error>, koji definiše grešku. [6]

4.3.2. Izmena konfiguracije

Operacija **<edit-config>** kao i <get-config> operacija omogućava dohvatanje čitave konfiguracije ili njen deo, dodatno omogućava izmenu konfiguracije. Operacija za izmenu konfiguracije treba da sadrži atribut *operation* koji definiše operaciju koja se treba izvršiti nad konfiguracijom. Ovaj atribut može imati neku od sledećih vrednosti[6]:

- *merge* – Spajanje podataka o konfiguraciji sa već postojećim u konfiguracionom skladištu. Ovo spajanje podataka predstavlja podrazumevano ponašanje operacije.

- *replace* – Zamena postojećih konfiguracionih podataka u skladištu sa novim podacima o konfiguraciji.

- *create* – Dodaju se novi konfiguracioni podaci. Ukoliko neki konfiguracioni podaci u skladištu već postoje, server vraća <rpc-error> sa <error-tag> elementom koji ima vrednost „data-exists“, što govori da podaci već postoje.

- *delete* – Brišu se konfiguracioni podaci ako postoje u skladištu. Ukoliko konfiguracioni podaci ne postoje u skladištu, server vraća <rpc-error> sa <error-tag> elementom koji ima vrednost „data-missing“, što govori da nedostaju podaci.

- *remove* – Kao i kod prethodnog atributa *delete*, brišu se konfiguracioni podaci iz skladišta, ali u ovom slučaju ukoliko ti konfiguracioni podaci u skladištu ne postoje, zahtev se ignoriše bez ikakve greške.

Operacija <edit-config> prima sledeće parametre:

- *target* predstavlja ciljno skladište podataka nad kojim se vrši izmena.

- *default-operation* predstavlja podrazumevanu operaciju i on je opcioni parametar. Opcije koje ovaj parametar može imati su: *merge* (spajanje konfiguracionih podataka iz parametra *config* sa odgovarajućim konfiguracionim podacima u skladištu), *replace* (zamena konfiguracionih podataka u skladištu sa podacima iz *config* parametra) ili *none* (nema uticaja osim ako to nije eksplicitno zadato *operation* atributom).

  Ako default-operation parametar nije naveden, njegova podrazumevana vrednost je *merge*.

- *test-option* je opcioni element koji se može koristiti jedino ako uređaj podržava posebnu sposobnost :validate:1:1, odnosno uređaj mora prijaviti da podržava validaciju konfiguracije pre nego što dozvoli korišćenje <test-option> elementa. Moguće vrednosti za <test-option> su: *test-then-set* (prvo validacija, pa izvršavanje), *set* (izvršavanje bez validacije) i *test-only* (samo validacija, bez izvršavanja).

  Ukoliko <test-option> element nije naveden, podrazumevana vrednost je *test-then-set*.

- *error-option* je opcioni element koji definiše ponašanje u slučaju pojave greške i može imati jednu od sledećih vrednosti: *stop-on-error* (prekida se prilikom pojave prve greške), *continue-on-error* (nastavlja se bez obzira na pojavu greške) i *rollback-on-error* (vraća se na početno stanje ako dođe do greške).

  Ukoliko <error-option> element nije naveden, podrazumevana vrednost je *stop-on-error*.

- *config* predstavlja hijerarhiju konfiguracionih podataka definisanu modelom podataka uređaja.

Rezultat izvršavanja <edit-config> operacije može biti pozitivan ili negativan. Ukoliko je uređaj u stanju da zadovolji zahtev, server šalje element <rpc-reply> kao odgovor koji sadrži <ok> element. Nasuprot tome, ako uređaj nije u stanju da zadovolji zahtev iz bilo kog razloga, server šalje element <rpc-reply> kao odgovor koji sadrži <rpc-error>, koji definiše grešku. [6]

4.3.3. Kopiranje konfiguracije

Operacija **<copy-config>** vrši kreiranje nove konfiguracije ukoliko skladište konfiguracionih podataka ne postoji, suprotno ako postoji vrši zamenu postojeće konfiguracije u skladištu konfiguracionih podataka sadržajem drugog kompletnog skladišta konfiguracionih podataka.

Operacija <copy-config> prima sledeće parametre:

- *target* predstavlja ciljno skladište podataka nad kojim se vrši kreiranje ili zamena konfiguracije.

- *source* predstavlja konfiguraciju čiji se sadržaj kopira u skladište konfiguracionih podataka.

Rezultat izvršavanja <copy-config> operacije može biti pozitivan i negativan. Ukoliko je uređaj u stanju da zadovolji zahtev, server šalje element <rpc-reply> kao odgovor koji sadrži <ok> element. Nasuprot tome, ako uređaj nije u stanju da zadovolji zahtev iz bilo kog razloga, server šalje element <rpc-reply> kao odgovor koji sadrži <rpc-error>, koji definiše grešku. [6]

4.3.4. Brisanje konfiguracije

Operacija **<delete-config>** vrši brisanje skladišta konfiguracionih podataka. Ova operacija prima sledeći parametar:

- *target* predstavlja ciljno skladište podataka nad kojim se vrši operacija brisanja.

Rezultat izvršavanja <delete-config> operacije može biti pozitivan ili negativan. Ukoliko je uređaj u stanju da zadovolji zahtev, server šalje element <rpc-reply> kao odgovor koji sadrži <ok> element. Nasuprot tome, ako uređaj nije u stanju da zadovolji zahtev iz bilo kog razloga, server šalje element <rpc-reply> kao odgovor koji sadrži <rpc-error>, koji definiše grešku. [6]

Operacije **<lock>** i **<unlock>** omogućavaju zaključavanje i otključavanje sistema za skladištenje podataka o konfiguraciji uređaja. Ove operacije postoje kako bi se izbegla interakcija sa drugim NETCONF klijentima prilikom izvršavanja nekih promena. Ako je skladište zaključeno, svaki drugi pokušaj zaključavanja biće neuspešan, dok se god skladište ne otključa ili se NETCONF sesija ne zatvori. Takođe, operacija zaključavanja će biti neuspešna i u slučajevima kada je ciljna konfiguracija <candidate>, već je izmenjena, ali te promene nisu sačuvane ili vraćene nazad, ili ako je ciljna konfiguracija <running>, a druga NETCONF sesija već ima potvrđeno čuvanje. Operacija otključavanja neće biti uspešno izvršena ako navedeno zaključavanje nije aktivno ili ako sesija koja izdaje operaciju <unlock> nije ista sesija koja je izdala operaciju <lock>.

Ove operacije primaju jedan parametar:

- *target* predstavlja ciljno skladište podataka za zaključavanje ili otključavanje.

Rezultati izvršavanja <lock> i <unlock> operacija mogu biti pozitivni ili negativni. Ukoliko je uređaj u stanju da zadovolji zahtev, server šalje element <rpc-reply> kao odgovor koji sadrži <ok> element. Nasuprot tome, ako uređaj nije u stanju da zadovolji zahtev iz bilo

kog razloga, server šalje element <rpc-reply> kao odgovor koji sadrži <rpc-error>, koji definiše grešku. [6]

4.3.5. Završetak sesije

Operacija **<close-session>** pruža mogućnost završetka NETCONF sesije na prirodan i elegantan način, što znači da će se prilikom izvršavanja ove operacije, sva zaključavanja otkazati i svi će resursi povezani sa tom sesijom biti otpušteni i veze prekinute. Nakon ove operacije, svi zahtevi za tu sesiju će biti ignorisani.

Operacija **<kill-session>** takođe vrši prekidanje NETCONF sesije ali na prisilan način. Prijemom zahteva <kill-session>, prekidaju se sve operacije koje su trenutno u procesu, otključavaju se sva zaključavanja, otpuštaju se resursi i zatvaraju sve veze. Ako ova operacija prekine neku operaciju u izvršavanju, NETCONF server je dužan da vrati konfiguraciju u stanje koje je bilo pre izvršavanja promena vezanih za tu operaciju. Ova operacija prima jedan parametar:

- *session-id* označava identifikator NETCONF sesije koja će biti prekinuta.

Rezultati izvršavanja <close-session> i <kill-session> operacija mogu biti pozitivni ili negativni. Ukoliko je uređaj u stanju da zadovolji zahtev, server šalje element <rpc-reply> kao odgovor koji sadrži <ok> element. Nasuprot tome, ako uređaj nije u stanju da zadovolji zahtev iz bilo kog razloga, server šalje element <rpc-reply> kao odgovor koji sadrži <rpc-error>, koji definiše grešku. Ako je session-id parametar <kill-session> operacije jednak ID-u trenutne sesije, vraća se greška „invalid-value“, koja govori da je uneta nevalidna vrednost. [6]

4.4. Mogućnosti NETCONF protokola

Modularnost je jedna osobina NETCONF protokola koja omogućava implementaciju protokola sa različitim mogućnostima (engl. *capabilities*) iz određenog skupa mogućnosti koje klijent ili server može implementirati. Na samom početku procesa razmene mogućnosti, potrebno je da i klijent i server oglase svoje mogućnosti.

Prilikom uspostavljanja sesije, klijent i server moraju poslati element <hello> koji sadrži listu mogućnosti tog entiteta. Svaki entitet mora da pošalje najmanje jednu (osnovnu) mogućnost [6]:

"urn:ietf:params:netconf:base:1.1"

Obe strane razumeju samo one mogućnosti koje su oglasile i sve ostale mogućnosti koje su za njega strane, mora da ignoriše.

Server koji šalje element <hello> mora da pošalje i element <session-id> koji sadrži ID sesije za ovu NETCONF sesiju. Suprotno tome, kada klijent šalje <hello> poruku, element <session-id> ne sme biti prisutan. Ukoliko server primi <hello> poruku od klijenta koja sadrži element <session-id>, mora prekinuti NETCONF sesiju. Slično, ako klijent primi <hello>

poruku bez <session-id> elementa, mora prekinuti NETCONF sesiju, bez prethodnog slanja <close-session>.[6]

NETCONF mogućnosti se identifikuju URI-jem i njihova definicija ima sledeći format:

urn:ietf:params:netconf:capability:{name}:1.x

U ovoj definiciji {name} označava naziv mogućnosti.

Na slici 4.5. dat je primer jednog oglašavanja od strane servera, prilikom kojeg oglašava osnovnu NETCONF mogućnost, zatim jednu NETCONF mogućnost definisanu u osnovnom NETCONF dokumentu i jednu mogućnost specifičnu za implementaciju:

```
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
    <capabilities>
       <capability>
            urn:ietf:params:netconf:base:1.1
       </capability>
       <capability>
            urn:ietf:params:netconf:capability:startup:1.0
       </capability>
       <capability>
            http://example.net/router/2.3/myfeature
       </capability>
     </capabilities>
     <session-id>4</session-id>
</hello>
```

Slika 4.5. Primer oglašavanja mogućnosti [6]

Osnovne mogućnosti NETCONF protokola su [6]:

- *Writable-Running Capability* – predstavlja mogućnost pisanja i identifikuje se sledećim URI formatom:

  urn:ietf:params:netconf:capability:writable-running:1.0

  Mogućnost pisanja ukazuje na to da uređaj podržava direktno pisanje u trenutno <running> skladište podataka konfiguracije. Odnosno operacije <edit-config> i <copy-config> je moguće vršiti nad <running> konfiguracijom, tj. u slučajevima ovih operacija, njihova <target> konfiguracija je <running> konfiguracija.

- *Candidate Configuration Capability* – predstavlja mogućnost konfiguracije kandidata i identifikuje se sledećim URI formatom:

  urn:ietf:params:netconf:capability:candidate:1.0

Uređaj koji oglasi mogućnost :capability zapravo podržava skladište podataka kandidata za konfiguraciju. Skladište pohranjuje konfiguracione podatke kojima se može manipulisati, bez rizika da će to uticati na trenutnu konfiguraciju uređaja. Izvršavanjem operacije <commit> u bilo kom trenutku, podešavamo vrednost <running> konfiguracije na vrednost konfiguracije kandidata.

Ukoliko se ova operacija izvrši uspešno, vraća se <rpc-reply> odgovor koji sadrži element <ok>, u suprotnom se vraća <rpc-reply> odgovor koji sadrži <rpc-error>. Ukoliko dođe do nekog problema, moguće je ukloniti sve izmene koje se nalaze na kandidatu izvršavanjem operacije <discard-changes>, koja će stanje kandidata postaviti na stanje trenutne <running> konfiguracije.

- *Confirmed Commit Capability* – oglašavanje ove mogućnosti znači da će server podržavati operaciju <cancel-commit>, kao i <commit> operaciju koja može za parametre imati: *confirmed, confirm-timeout, persist* i *persist-id*. Ova mogućnost se identifikuje sledećim URI formatom:

  urn:ietf:params:netconf:capability:confirmed-commit:1.1

  Da bi <commit> operacija bila potvrđena, neophodno je da sadrži parametar confirmed. Ako operacija ne sadrži ovaj parametar, ona mora biti poništena u vremenskom periodu od 600 sekundi, koje je podrazumevano vreme ako nije dat confirm-timeout parametar. Ukoliko je prisutan confirm-timeout parametar, on definiše vremensko ograničenje za potvrdu. Parametri persist i persist-id omogućavaju izvršavanje iste operacije nad bilo kojom sesijom navođenjem elementa persist-id sa vrednošću jednakoj vrednosti persist. Ukoliko parametar persist nije naveden, onda će izvršavanje ove operacije uvek biti nad istom sesijom.

  Operacija <cancel-commit> otkazuje trenutnu operaciju <commit>.

  Ukoliko se ova operacija izvrši uspešno, vraća se <rpc-reply> odgovor koji sadrži element <ok>, u suprotnom se vraća <rpc-reply> odgovor koji sadrži <rpc-error>.

- *Rollback-on-Error Capability* – predstavlja mogućnost vraćanja na prethodno stanje konfiguracije (engl. *rollback*), u slučaju greške prilikom izvršavanja <edit-config> operacije. Ova mogućnost se identifikuje sledećim URI formatom:

  urn:ietf:params:netconf:capability:rollback-on-error:1.0

- *Validate Capability* – omogućava validaciju, odnosno proveru sintakse i semantike u kompletnoj konfiguraciji. Ova mogućnost se identifikuje sledećim URI formatom:

  urn:ietf:params:netconf:capability:validate:1.1

  Ova mogućnost se može koristiti na dva načina: korišćenjem parametra *validate,* kao i parametra *test-option* u okviru operacije <edit-config>.

- *Distinct Startup Capability* – podržavanje odvojenih konfiguracija za running i startup konfiguraciona skladišta podataka. Ova mogućnost se identifikuje sledećim URI formatom:

  urn:ietf:params:netconf:capability:startup:1.0

- *URL Capability* – omogućava prihvatanje URL elementa kao *source* (srp. izvor) i kao *target* (srp. cilj) parametar. Ova mogućnost se identifikuje sledećim URI formatom:

  urn:ietf:params:netconf:capability:url:1.0?scheme={name, …}

  pri čemu *scheme* (srp. šema) predstavlja argument kom se dodeljuje lista naziva šema koje su podržane.

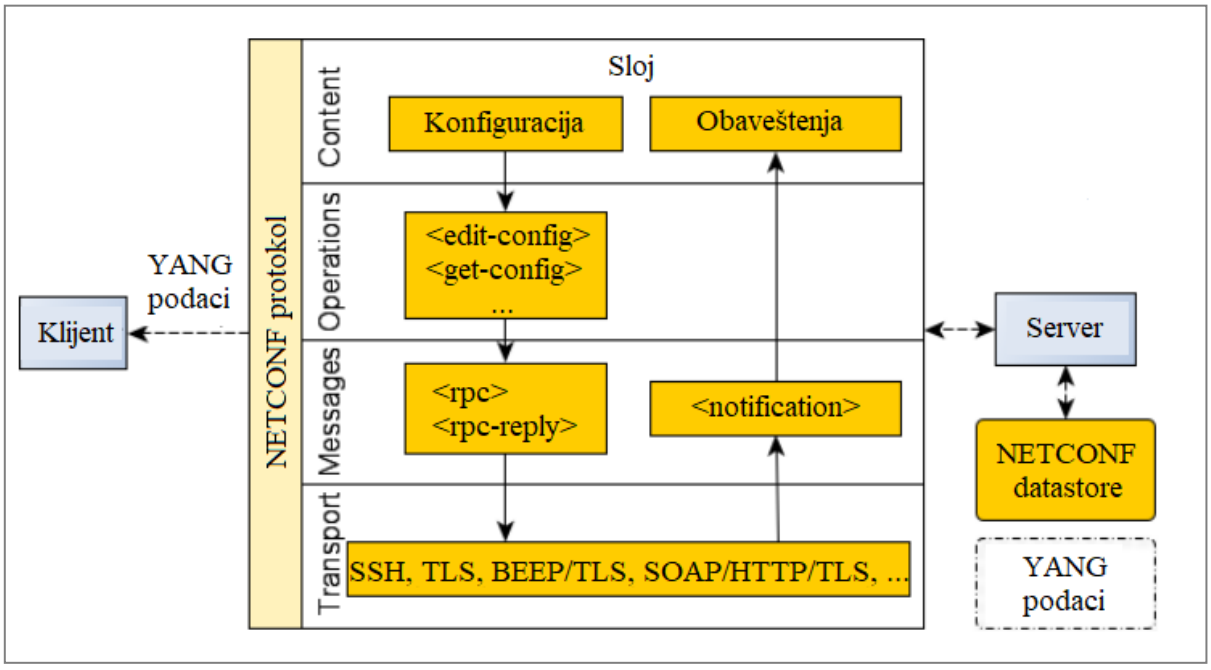
- *XPath Capability* – NETCONF klijent podržava upotrebu XPath izraza u okviru elementa filter. Ova mogućnost se identifikuje sledećim URI formatom:

  urn:ietf:params:netconf:capability:xpath:1.0

4.5. Razmena podataka putem NETCONF protokola

Prilikom komunikacije između klijenta i servera, neophodno je da klijent bude upoznat sa modelom podataka kako bi pravilno primio i protumačio pakete koje dobija. Komunikacija se ostvaruje putem RPC poruka koje su u XML formatu. Model podataka je definisan na strani servera u obliku YANG modula. [18]

Razmena podataka putem NETCONF protokola između klijenta i servera je prikazana na sledećoj slici:



Slika 4.6. Razmena podataka putem NETCONF protokola [21]

NETCONF definiše jednostavnu klijent-server arhitekturu za upravljanje mrežnim uređajima. Ovo uključuje preuzimanje, manipulaciju i slanje konfiguracionih podataka. U sadržajnom (engl. *Content*) sloju nalaze se konfiguracioni podaci koji se ugrađuju u NETCONF operacije (<edit-config>, <get-config>, itd.) u sloju operacija (engl. *Operations*). Operacija se zatim izvršava kao RPC poruka u sloju poruka (engl. *Messages*) i šalje se dalje preko transportnog (engl. *Transport*) sloja.

Na transportnom sloju dostupni su bezbednosni protokoli za prenos, poput TLS i SSH, koji obezbeđuju sigurnost protokola. [21]

Element <rpc> ima obavezni atribut *message-id*. Ovaj atribut je tipa string, bira ga pošiljalac RPC poruke i služi za povezivanje odgovora sa odgovarajućim zahtevom. Ovo je ključna stvar u komunikaciji sa <rpc> porukama, jer klijent može poslati niz od jedne ili više <rpc> poruka serveru, a klijent uzvraća niz od jedne ili više odgovarajućih <rpc-reply> poruka. NETCONF <rpc> zahtevi moraju da se obrađuju sekvencijalno na strani uređaja nad kojim se vrši upravljanje. Novi zahtevi mogu se poslati nakon prvog poslatog zahteva iako on još uvek nije završen, ali uređaj u tom slučaju šalje odgovore po redosledu kojim ih je dobio. Povezivanje ovih poruka vrši se na osnovu *message-id*, jer on mora biti identičan u <rpc> poruci i <rpc-reply> poruci. Ovo je ključno za preciznu sinhronizaciju u komunikaciji između klijenta i servera. Pošiljalac mora osigurati da je vrednost atributa *message-id* u skladu sa pravilima normalizacije XML atributa definisanim u okviru W3C preporuke XML 1.0 - W3C.REC-xml-20001006 [22], koja definiše pravila kodovanja i interpretacije XML sadržaja. Jedino osiguravanjem vrednosti *message-id* atributa možemo biti sigurni da će se string vratiti nepromenjen. [6]

Element <rpc-reply> osim obaveznog atributa *message-id* koji treba imati istu vrednost kao i atribut *message-id* u <rpc> elementu, mora vratiti sve dodatne atribute iz <rpc> elementa nepromenjene. Na slici 4.7. prikazan je primer zajedničkih atributa za <rpc> poruku i njen odgovarajući <rpc-reply> odgovor. [6]

```
<rpc message-id="101"
        xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
        xmlns:ex="http://example.net/content/1.0"
        ex:user-id="fred">
    <get/>
</rpc>

<rpc-reply message-id="101"
        xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
        xmlns:ex="http://example.net/content/1.0"
        ex:user-id="fred">
    <data>
        <!-- ... -->
    </data>
</rpc-reply>
```

Slika 4.7. Primer <rpc> poruke i odgovarajućeg <rpc-reply> odgovora [6]

4.6. Prednosti NETCONF protokola u poređenju sa SNMP protokolom

Jedan od glavnih ciljeva zbog kojih je nastao NETCONF protokol jeste da zameni nestandardni pristup konfigurisanja mrežnih uređaja putem komandne linije, kao i da pruži mogućnosti koje SNMP nema. NETCONF rešava ograničenja koja sa sobom nosi SNMP i omogućava da se mreže programiraju i automatizuju na pouzdaniji način od načina korišćenog kod SNMP-a. Ovo je omogućeno time što NETCONF protokol koristi TCP i time omogućava da se mreže programiraju i automatizuju na mnogo pouzdaniji način od SNMP koji je za ove svrhe uglavnom koristio UDP.

SNMP protokol nije imao mogućnost da razlikuje podatke o konfiguraciji i podatke o stanju, ova funkcionalnost je omogućena kod NETCONF protokola i ovim je administratorima omogućeno da prate i upoređuju stanja konfiguracija, dok su administratori kod SNMP-a morali ručno proveravati i identifikovati određena stanja. NETCONF protokol omogućava da se grupa semantički povezanih podataka o konfiguraciji modifikuje zajedno, odnosno da se izabrani ili svi parametri modifikuju u jednoj operaciji, dok se kod SNMP protokola mora modifikovati jedna po jedna vrednost podatka. Takođe, jedna od bitnih karakteristika NETCONF protokola jeste da ima sposobnost obavljanja transakcija. Ovo znači da može bez greške postaviti konfiguracije na uređajima, dok u slučaju pojave greške, konfiguracija neće ni biti izmenjena. [6]

U sledećoj tabeli prikazane su osnovne karakteristike bitne za nadzor mrežnih uređaja u dinamičnim mrežnim okruženjima uporedo za SNMP i NETCONF:

Karakteristika	**SNMP**	**NETCONF**
Fokus	Monitoring uređaja	Monitoring i konfiguracija uređaja
Format podataka	ASN.1 / MIB	XML / YANG
Bezbednost	Veoma niska bezbednost u SNMPv1 i SNMPv2. U verziji SNMPv3 je obezbeđena ograničena bezbednost	Visoka bezbednost preko SSH, TLS
Ekstenzibilnost	Ograničena (fiksna struktura preko MIB tabela)	Visoka (YANG model i XML)
Primena u IoT	Teško se prilagođava	Pogodniji model-based pristup

Tabela 4.2. Poređenje osnovnih karakteristika SNMP i NETCONF protokola bitnih za nadzor mrežnih uređaja

SNMP protokol je nastao sa ciljem da omogući nadzor i osnovno upravljanje mrežnim uređajima. Kada je reč o upravljanju, SNMP je u mogućnosti da izvrši osnovne „set“ operacije, iako je primarno optimizovan za prikupljanje statusnih operacija, ali je ipak nedovoljno nadograđen da bi mogao izvršiti transakcijsko upravljanje, validaciju i modelovanje podataka. Samim tim se čini nedovoljno nadograđenim kako bi vršio neke kompleksnije manipulacije podacima u savremenim mrežama, naročito u kontekstu IoT (*Internet of Things*) sistema. [23]

SNMP protokol koristi SMI (engl. *Structured Management Information*), a NETCONF koristi YANG kao jezik za modelovanje koji je čitljiviji, jednostavniji za korišćenje i koristi module i podmodule za skladištenje podataka. Na taj način je omogućena prirodna ekstenzibilnost. [14]

Server može implementirati više modula ili podmodula, ali se to i dalje posmatra kao jedan logički modul i na taj način se omogućava fleksibilno proširivanje modela prema potrebama korisnika ili proizvođača. Takođe, YANG jezik omogućava razdvajanje konfiguracionih i operativnih podataka, koristeći atribut *config*, što takođe predstavlja prednost NETCONF protokola u poređenju sa SNMP, jer SNMP ovo ne poseduje. [24]

Osnovni razlog nastanka SNMPv3 je upravo u nedostatku bezbednosti koji su prethodne verzije (SNMPv1 i SNMPv2) imale. Bezbednosni propusti su se odnosili na sledeće situacije [23]:

- Postojala je opasnost od izmena informacija, odnosno da neovlašćena strana može izmeniti poruke tokom prenosa, koje su generisane u ime ovlašćene strane.
- Na isti način, moguće je bilo i da se korisnik predstavi kao drugi korisnik i time dobije operacije koje njemu nisu primarno bile dozvoljene (engl. *Masquerade*).
- Izmena toka poruka koja nastaje kao posledica načina na koji SNMP funkcioniše preko transportnog servisa bez uspostavljanja veze, koji može raditi preko bilo koje podmrežne usluge. Ovo može dovesti do promene redosleda, kašnjenja ili ponavljanja poruka.
- Neovlašćena strana može prisluškivati razmenu poruku između SNMP entiteta.

Prethodno navedeni problemi koji su predstavljeni u RFC 3411, delimično su rešeni kroz SNMPv3 arhitekturu, kroz *User-based Security Model* (USM), definisan u RFC 3414 [25]. Ovaj dokument opisuje model bezbednosti zasnovan na korisniku (USM) u okviru SNMP arhitekture. Ideja za postizanje ovog modela bezbednosti se svodi na korišćenje koncepta korisnika (identifikovanog preko korisničkog imena - *username*) uz kojeg se vezuju bezbednosne informacije. U ovom dokumentu, rešenje određenih bezbednosnih problema je postignuto na sledeći način [25]:

- USM koristi HMAC-MD5-96 i HMAC-SHA-96 kako bi osigurao integritet poruka. Na ovaj način su sprečeni problemi koji su postojali a tiču se integriteta informacija i potvrde identiteta korisnika. Ovim pristupom onemogućeno je neovlašćenim stranama da vrše izmene nad porukama tokom prenosa, kao i drugim korisnicima da se predstavljaju kao neki drugi korisnici.
- SNMPv3 uvodi proveru vremenske ažurnosti poruka, čime se rešava problem ponavljanja poruka.

- Korišćenjem CBC-DES enkripcije za zaštitu sadržaja SNMP poruke, sprečava se problem prisluškivanja koji je postojao ranije.

Iako SNMPv3 unosi značajna poboljšanja kada je u pitanju bezbednost, još uvek postoje ograničenja kada je u pitanju formatiranje poruka, ekstenzibilnost i prilagođavanje IoT okruženjima. Zbog toga se u nastavku rada NETCONF protokol koristi za nadzor, analizu i upravljanje konfiguracijama mrežnih uređaja u velikim i dinamičnim mrežnim infrastrukturama. NETCONF protokol koristi XML za strukturu podataka, čime značajno utiče na formatiranje poruka, ekstenzibilnost i prilagođavanje IoT okruženjima.

Na osnovu svih tih činjenica, dolazimo do zaključka da se NETCONF protokol izdvaja svojim savremenim pristupom kada je u pitanju nadzor mreže, ali takođe i manipulacija konfiguracionim podacima na siguran i standardizovan način.

5. SISTEM ZA NADZOR, ANALIZU I UPRAVLJANJE KONFIGURACIJAMA MREŽNIH UREĐAJA ZASNOVAN NA NETCONF PROTOKOLU

5.1. Uvod

Praktični deo ovog rada predstavlja implementiran sistem za nadzor, analizu i upravljanje konfiguracijama mrežnih uređaja zasnovan na NETCONF protokolu. S obzirom na to da brojnost mrežnih uređaja koji čine jednu mrežu iz dana u dan postaje sve veća, postaje praktično nemoguće da se održi stabilnost i nesmetana funkcionalnost uređaja u mreži bez dobro implementiranog sistema, osmišljenog prema karakteristikama mreže, koji će omogućiti nadgledanje mreže, odnosno stanja svih uređaja u svakom momentu.

Kada je reč o konfiguracijama uređaja u mreži, ako uzmemo u obzir prethodno opisane protokole za manipulaciju konfiguracijama mrežnih uređaja, možemo zaključiti da je korišćenje NETCONF protokola koji značajno prevazilazi ograničenja koje ima SNMP protokol u pogledu bezbednosti i automatizacije mreže, odličan izbor. Iako NETCONF ima mnogo prednosti, još uvek postoji mogućnost pojave greške u konfiguraciji, na primer usled ljudske greške pri unosu.

Predloženo rešenje je sistem za nadzor, analizu i upravljanje konfiguracijama mrežnih uređaja zasnovan na NETCONF protokolu, koji omogućava pravovremeno otkrivanje grešaka u konfiguraciji, u bilo kom trenutku. Sa druge strane, ovaj sistem nudi praćenje velikog broja uređaja i čuvanje istorije rezultata nadzora uređaja, kao i uvid u učestalost pojave specifičnih grešaka, što olakšava identifikaciju problema u najranijoj fazi i omogućava sprečavanje potencijalnih trajnih uzroka problema.

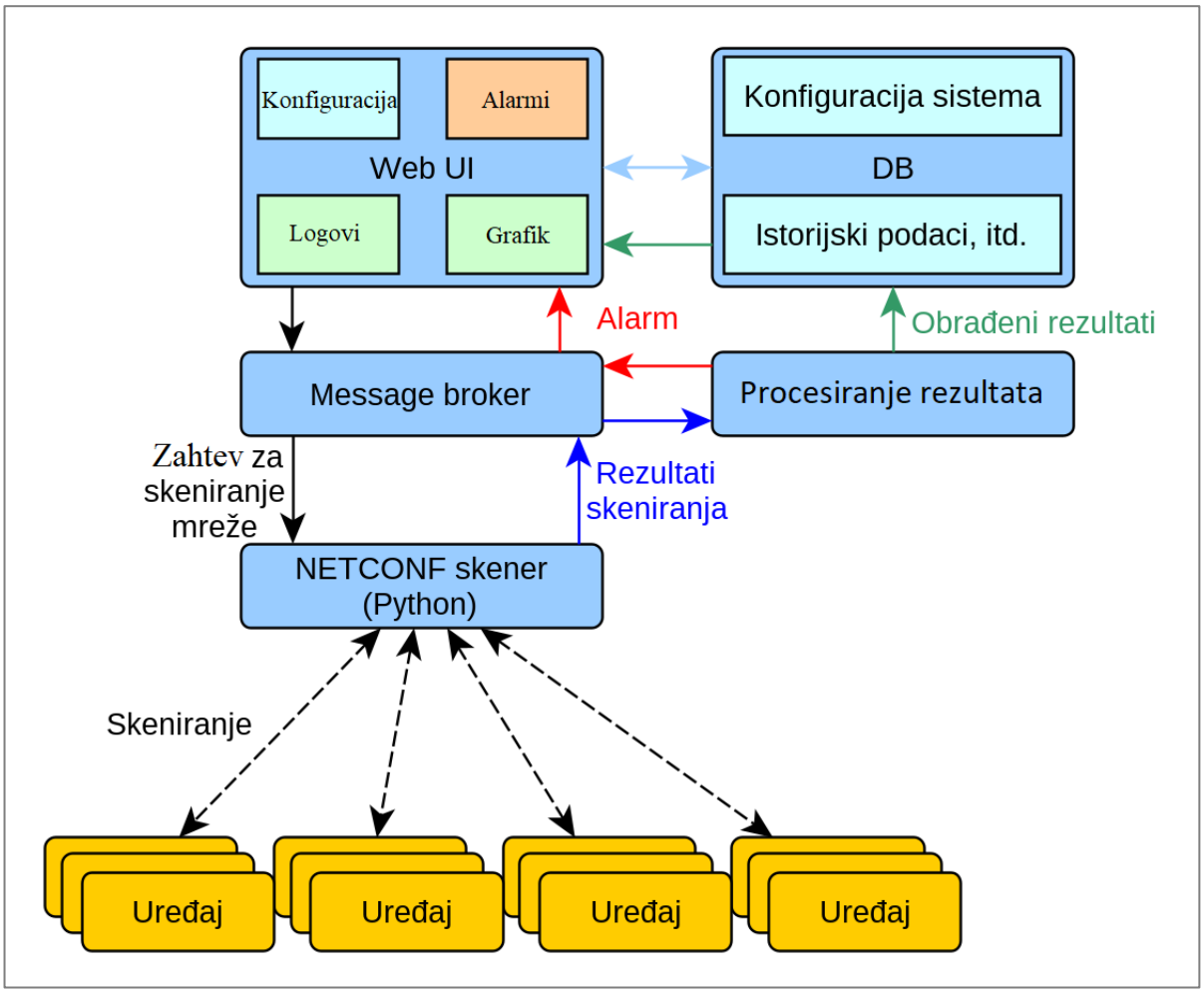
Dodatni zahtevi koje sistem treba da ispuni:

- WEB korisnički interfejs (WEB UI),
- Horizontalna skalabilnost sistema,
- Mogućnost budućeg proširenja funkcionalnosti.

U nastavku rada, detaljno su opisane sve funkcionalnosti ovog sistema, predstavljena je arhitektura sistema, opisane su prednosti korišćenja konfiguracije u XML formatu u kombinaciji sa NETCONF protokolom i YANG jezikom i modulom, horizontalna skalabilnost sistema, istorijski podaci, kao i implementirana mobilna aplikacija za ovaj sistem koja omogućava korisnicima sistema da u svakom momentu imaju uvid u trenutno stanje mreže.

5.2. Idejno rešenje sistema

Arhitektura prikazana na slici 5.1. predstavlja idejno rešenje sistema za nadzor, analizu i upravljanje konfiguracijama mrežnih uređaja zasnovanog na NETCONF protokolu, koji je razvijen kao praktični deo ovog rada.



Slika 5.1. Arhitektura idejnog rešenja sistema za nadzor, analizu i upravljanje konfiguracijama mrežnih uređaja zasnovanog na NETCONF protokolu

Sistem treba da omogući pravovremeno prikupljanje i obradu podataka o konfiguracijama uređaja koji se nalaze u mreži, zatim otkrivanje potencijalnih grešaka u konfiguraciji i vizuelizaciju ključnih informacija kroz intuitivni WEB UI. Zasniva se na upotrebi NETCONF protokola.

Ključne komponente sistema su:

- WEB UI,
- Baza podataka (DB),
- Message broker,
- NETCONF skener,
- Komponenta za procesiranje rezultata,

- Mrežni uređaji.

WEB UI je komponenta koja omogućava interakciju između korisnika i sistema. Ova komponenta treba da na intuitivan način pruži korisniku sve informacije koje su potrebne u procesu nadgledanja mreže, kao i da primi sve zahteve od korisnika i prosledi ih dalje message brokeru.

WEB UI treba korisniku da omogući prikaz sledećih informacija:

- Podaci o konfiguracijama uređaja za koje korisnik iskaže interesovanje, kao i stanje konfiguracija svih uređaja koji se trenutno nalaze u mreži,
- Obaveštenja u slučaju detekcije grešaka u konfiguracijama uređaja,
- Različite vrste logova – rezultati skeniranja, izmene u konfiguracijama, greške,
- Istorijski i trenutni podaci prikazani grafički, jer vizuelno predstavljanje olakšava dalju analizu.

Baza podataka sadrži informacije o konfiguracijama uređaja koji se nalaze u mreži, konfiguracijama uređaja koji su se u prošlosti nalazili u mreži, istorijskim podacima iz prethodnih skeniranja, prikupljenim rezultatima nadzora i analiza. Zahvaljujući ovoj komponenti, omogućena je pohrana svih podataka potrebnih za pravilan rad sistema, kao i za napredne analize i statistike.

Message broker omogućava asinhronu komunikaciju između WEB UI-a, NETCONF skenera i komponente za procesiranje rezultata. Na taj način omogućena je veća skalabilnost i modularnost sistema, veća otpornost na greške, kao i jednostavno proširenje sistema dodatnim funkcionalnostima.

NETCONF skener je implementiran u programskom jeziku *Python* jer ovaj programski jezik poseduje veliki ekosistem biblioteka, uključujući i *XPath*, koji nam omogućava da precizno izdvojimo potrebne čvorove iz XML stabla konfiguracije. Korišćenjem NETCONF protokola skener ostvaruje direktnu komunikaciju sa uređajima u mreži. Skener periodično šalje zahteve uređajima za skeniranje i prikuplja odgovore, koje zatim putem message brokera prosleđuje do komponente za procesiranje rezultata.

Procesiranje rezultata je komponenta koja vrši prijem podataka koji dolaze kao rezultat skeniranja od NETCONF skenera, zatim vrši obradu i transformaciju tih podataka i na kraju upisuje rezultate u bazu podataka.

Mrežni uređaji čine krajnji sloj sistema, s njima direktno komunicira NETCONF skener i preuzima podatke o njihovim konfiguracijama u svakom trenutku.

Idejno rešenje za ovaj sistem sa sobom nosi sledeće:

- *Skalabilnost* – paralelan nadzor velikog broja uređaja.
- *Bezbednost* – koja se postiže upotrebom NETCONF protokola koji komunikaciju može ostvariti preko SSH/TLS, i YANG modula za predstavljanje konfiguracije sa smanjenom mogućnošću za greške u konfiguracijama.

- *Precizna analiza konfiguracija* – Zahvaljujući kombinovanom korišćenju predstavljanja konfiguracija u XML formatu i XPath jezika koji omogućava jednostavno pronalaženje i izdvajanje podataka iz XML strukture, sistem može veoma lako analizirati podatke koje prikupi.
- *Vizuelizacija* – Fokus na grafičkim prikazima, omogućava korisniku lakše razumevanje problema i njihovu dublju analizu.
- *Modularnost* – Arhitektura omogućava lako proširenje sistema dodatnim komponentama, kao što su dodatni protokoli, AI moduli za predikciju problema ili mobilni interfejsi.

5.3. NETCONF i Python

Komponenta koja prima podatke od NETCONF skenera, vrši njihovu obradu i transformaciju, te upisuje rezultate u bazu podataka, implementirana je u Python programskom jeziku. Python programski jezik je izabran za implementaciju ove komponente zbog svoje jednostavnosti, fleksibilnosti i moćnih biblioteka. Dodatni razlozi za odabir Python programskog jezika za implementaciju ove komponente su sledeći:

- Python je jedan od najzastupljenijih programskih jezika u oblasti mrežne automatizacije i upravljanja mrežnim uređajima zbog svoje bogate kolekcije biblioteka i jednostavne sintakse. Biblioteka *ncclient* za Python je razvijena posebno za rad sa NETCONF protokolom. Cilj postojanja *ncclient* biblioteke je da pruži intuitivan API koji jasno preslikava XML kodovane NETCONF podatke u podatke razumne Python programskom jeziku.

  Ključne karakteristike *ncclient* biblioteke:

  - Podržava sve operacije i mogućnosti definisane u RFC 6241.
  - Podržava *pipelining*, što znači da pruža mogućnost lančanog povezivanja više zadataka i lančanog izvršavanja.
  - Omogućava asinhrone RPC zahteve.
  - Ne koristi rad sa XML-om osim ako je on zaista potreban.
  - Omogućava proširivost i lako prilagođavanje novim mogućnostima i operacijama.

- NETCONF protokol koristi XML format za razmenu poruka, a Python ima jaku podršku za obradu XML formata kroz module kao što su *xml.etree.ElementTree* i *lxml*, koji omogućavaju korišćenje XPath jezika za navigaciju kroz XML dokumente. Ovaj jezik omogućava jednostavno i precizno lociranje, selektovanje i dohvatanje podataka iz XML strukture. Takođe, sam princip poređenja konfiguracija se svodi na poređenje dve konfiguracije koje se nalaze u XML formatu, tako da su ovi moduli u velikoj meri doprineli jednostavnosti implementacije komponente sistema za poređenje očekivane i prave konfiguracije uređaja. Svrha XPath jezika je adresiranje čvorova u XML i JSON stablima.

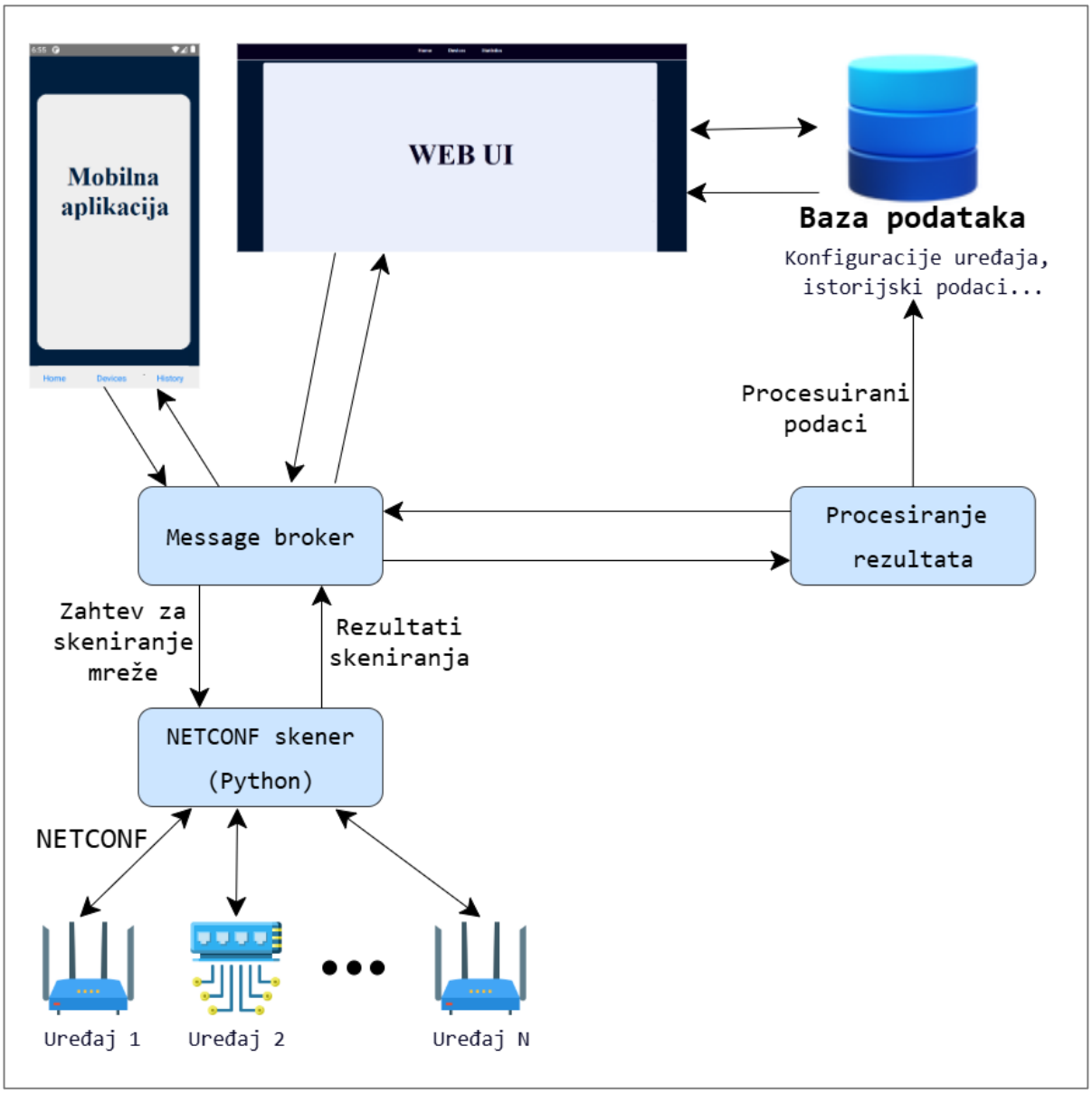
XPath je svoj naziv dobio jer koristi putanju za navigaciju kroz hijerarhijsku strukturu XML dokumenta. Bitna prednost XPath-a kao alata koji se koristi prilikom nadzora konfiguracija uređaja, odnosno poređenja realnih i očekivanih konfiguracija, jeste da XPath u potpunosti podržava imenske prostore. Ime čvora se kreira kao par koji se sastoji od lokalnog dela i URI-ja imenskog prostora (koji može biti prazan). Ovako kreirano ime čvora se naziva prošireno ime (engl. *expanded-name*). [26]

U implementiranom sistemu za nadzor, analizu i upravljanje konfiguracijama mrežnih uređaja zasnovanom na NETCONF protokolu, XPath se koristi prilikom dohvatanja konfiguracije uređaja čime se olakšava korišćenje i omogućava dalja modifikacija podataka za potrebe ovog sistema.

5.4. Arhitektura sistema

Sistem omogućava nadzor konfiguracija mrežnih uređaja korišćenjem NETCONF protokola, uz primenu savremenih principa distribuirane arhitekture, WEB UI-a i kontejnerizacije. Arhitektura sistema je prikazana na slici 5.2.

Sistem je implementiran kao WEB UI, čime je omogućen pristup funkcionalnostima sistema sa udaljenih lokacija putem REST API-ja, implementiranog u *Flask* okruženju. Ovaj WEB UI pruža pregledne i interaktivne prikaze rezultata analiza i konfiguracija korišćenjem *React* biblioteke. Korišćenjem *React* biblioteke implementirani su interaktivni prikazi uređaja u mreži, konfiguracija tih uređaja, grešaka u konfiguracijama, statističkih podataka, kao i jednostavno upravljanje sistemom od strane korisnika. Korišćenjem WEB UI-a, ovoj aplikaciji se može pristupiti u svakom trenutku nezavisno od lokacije korisnika, putem standardnog internet pretraživača, bez potrebe za dodatnom instalacijom softvera.



Slika 5.2. Arhitektura sistema za nadzor, analizu i upravljanje konfiguracijama mrežnih uređaja zasnovanog na NETCONF protokolu

Svaki uređaj u mreži prolazi kroz proces verifikacije konfiguracije na način da se trenutna konfiguracija uređaja poredi sa očekivanom konfiguracijom. Proces verifikacije je realizovan preko implementiranog skenera u Python programskom jeziku, koji korišćenjem NETCONF protokola prikuplja konfiguracije sa mrežnih uređaja koji se nalaze u mreži. Poređenje konfiguracija vrši se korišćenjem XML tehnika za poređenje u komponenti za procesiranje, tako što se čvorovi iz stvarne konfiguracije uređaja porede se odgovarajućim čvorovima iz očekivane konfiguracije. Sistem je implementiran tako da u realnom vremenu omogući detekciju razlika u vrednostima čvorova, kao i prisustvo određenog elementa (čvora) u jednoj konfiguraciji, a njegovo odsustvo u drugoj. Proces poređenja ovih konfiguracija implementiran je u Python programskom jeziku. Python programski jezik je odabran iz razloga što poseduje veliki ekosistem biblioteka, uključujući i XPath, koji omogućava precizno izdvajanje potrebnih čvorova iz XML stabla konfiguracije. Na ovaj način je upoređivanje konfiguracija znatno olakšano, a istovremeno i pouzdano.

Ovaj sistem takođe poseduje još jednu veliku prednost kada je reč o korišćenju NETCONF protokola, a to je da korisnici sistema ne moraju detaljno poznavati YANG modul, jer se ovo poređenje konfiguracija isključivo vrši na osnovu XML putanja, koje su prethodno konvertovane u UTF-8 format, čime je svaki čvor jednoznačno identifikovan na osnovu svog imenskog prostora. Ovo je od suštinskog značaja u procesu poređenja čvorova, jer uglavnom mrežne konfiguracije sadrže više čvorova sa istim nazivom, ali u različitim imenskim prostorima. Poželjno je ipak poznavati YANG jezik i module na osnovnom nivou, zato što su svi podaci koji se razmenjuju između klijenta i servera definisani putem YANG jezika.

Sistem je projektovan na način da obezbedi horizontalnu skalabilnost kroz upotrebu RabbitMQ sistema za razmenu poruka (message broker) i asinhronog izvršavanja operacija. Message broker u ovoj arhitekturi sistema omogućava sledeće:

- **Asinhrona komunikacija** – Korisnik šalje zahtev za skeniranje uređaja, broker preuzima poruku i prosleđuje je NETCONF skeneru, kada NETCONF skener završi, vraća rezultat kroz isti posrednički kanal. Za vreme dok NETCONF skener vrši skeniranje uređaja, korisnik nije blokiran čekanjem rezultata, on može slati druge zahteve dok čeka odgovor.
- **Decoupling** (Nezavisnost komponenti sistema) – Komponente mogu funkcionisati nezavisno, izmena jedne komponente ne utiče na drugu komponentu. Svaka komponenta je jedan nezavisan deo sistema.
- **Skalabilnost** – mogućnost pokretanja više instanci NETCONF skenera, koji će paralelno obrađivati poruke od brokera.
- **Otpornost i pouzdanost** – u slučaju otkazivanja neke od komponenti, poruke ostaju u svom stanju dok ne dođe do oporavka komponente, tada se normalno nastavlja sa radom.

Rezultate skeniranja preuzima komponenta koja je na slici 12 nazvana „Procesiranje rezultata“, ova komponenta nakon preuzimanja rezultata, vrši dodatnu obradu i te podatke upisuje u nerelacionu bazu podataka MongoDB. U bazi podataka se čuvaju svi istorijski podaci koji obuhvataju sve konfiguracije uređaja u mreži kroz istoriju, uključujući njihove promene kroz vreme, kao i određene statističke podatke. Na ovaj način se omogućava analiza konfiguracija kroz vreme, kao i identifikacija različitih konfiguracionih grešaka i problema kroz vreme. Ovakve analize mogu u budućnosti biti od velike koristi prilikom nadograđivanja sistema, odnosno obučavanja sistema za reakciju na određene probleme.

Sistem je kontejnerizovan korišćenjem Docker tehnologije, čime je omogućeno lako raspoređivanje, nadogradnja i održavanje svih komponenti sistema kao nezavisnih jedinica. Kontejnerizacija omogućava lako testiranje, reprodukovanje okruženja i distribuciju aplikacije bez obzira na ciljni operativni sistem. Za upravljanje, skaliranje i nadzor nad kontejnerima koristi se Kubernetes, koji omogućava automatsko restartovanje neuspelih servisa, raspoređivanje opterećenja, kao i horizontalno skaliranje u skladu sa brojem korisnika. Dodatno, korišćenjem React Native-a, razvijen je mobilni klijent, koji omogućava pristup ključnim funkcijama sistema za nadzor sa mobilnih uređaja u svakom trenutku, bez obzira na platformu. Ovim je obezbeđena mobilna fleksibilnost i pristupačnost sistemu korisnicima u svakom trenutku.

Ovakva arhitektura sistema pruža visoku modularnost, fleksibilnost u razvoju kao i u održavanju i jednostavno prilagođavanje budućim zahtevima korisnika, kao i dodavanju ili menjanju određenih funkcionalnosti.

5.5. Implementirani sistem

Sistem razvijen u okviru ovog rada obavlja nadzor, analizu i upravljanje konfiguracijama mrežnih uređaja koji koriste NETCONF protokol. Arhitektura sistema se sastoji od više funkcionalnih komponenti koje zajednički omogućavaju prikupljanje, analizu i vizualizaciju konfiguracija uređaja, uz mogućnost skaliranja i proširivanja sistema u budućnosti. Sistem je osmišljen tako da bude modularan i nezavisan od konkretne platforme uređaja i proizvođača opreme, čime se omogućava šira primenljivost u raznovrsnim mrežnim okruženjima.

Implementirani sistem je testiran u virtuelnom okruženju kako bi se omogućila simulacija širokog spektra realnih situacija koje je teško reprodukovati na zahtev u realnom okruženju. Ovo podrazumeva slučajeve u kojima funkcionisanje samih uređaja mora biti neispravno, kao i one u kojima se konfiguracija treba delimično promeniti ili se neki parametar treba pogrešno postaviti. Takvo ponašanje mreže ne sme biti dozvoljeno i iz tog razloga mora biti obuhvaćeno sistemom za nadzor mreže. Implementirani sistem periodično izvršava funkciju koja upoređuje trenutnu i očekivanu konfiguraciju uređaja za sve uređaje koji se trenutno nalaze u mreži. Ukoliko se interval podesi na manju vrednost, dolazimo do odličnog rešenja za kritične mreže. Na taj način je moguće identifikovati probleme u najranijim fazama. Takođe, uvidom u trenutne i istorijske podatke koji se nalaze u bazi podataka i omogućeni su korisniku kroz WEB UI, moguće je steći uvid u osnovne uzroke problema. Otkrivanje problema u najranijim fazama omogućava brzo rešavanje problema, čime se obezbeđuje stabilnost i sigurnost mreže.

Ovaj sistem je pogodan za konfiguraciju bilo kog tipa uređaja, jer se poređenje konfiguracija uređaja ne bazira na tipu uređaja, već se vrši precizno poređenje svih čvorova iz trenutne konfiguracije sa svim čvorovima iz očekivane konfiguracije. Ovakav pristup doprinosi većoj pouzdanosti sistema i eliminiše potrebu za specifičnim adapterima zbog tipa uređaja. Poređenje konfiguracionih podataka je implementirano u Python programskom jeziku koristeći XPath za efikasnu navigaciju kroz XML strukturu konfiguracija uređaja, dok je za manipulaciju konfiguracijom uređaja korišćen NETCONF protokol.

Poređenje konfiguracionih čvorova ima za cilj da se utvrdi da čvor u XML reprezentaciji konfiguracije uređaja koji nadgledamo postoji ili ne postoji u očekivanoj konfiguraciji, takođe ukoliko čvor postoji, treba da se utvrdi da li je vrednost tog čvora identična u obe konfiguracije. Ovo je omogućeno na način da se svaka konfiguracija prvo prevodi u UTF-8 format, zato što je ovaj format pogodniji za razlikovanje čvorova u zavisnosti od imenskog prostora u kom se čvor nalazi, kako ne bi došlo do greške da identičan čvor postoji, ali u drugom imenskom prostoru. Primer konfiguracije koja se poredi je prikazan na sledećoj slici:

```
<config>
    <data xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
        <netconf-server xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-server">
            <listen>
                <endpoint>
                    <name>default-ssh</name>
                    <ssh>
                        <tcp-server-parameters>
                            <local-address>0.0.0.0</local-address>
                            <keepalives>
                                <idle-time>1</idle-time>
                                <max-probes>15</max-probes>
                                <probe-interval>5</probe-interval>
                            </keepalives>
                        </tcp-server-parameters>
                        <ssh-server-parameters>
                            <server-identity>
                                <host-key>
                                    <name>default-key</name>
                                    <public-key>
                                        <keystore-reference>genkey</keystore-reference>
                                    </public-key>
                                </host-key>
                            </server-identity>
                            <client-authentication>
                                <supported-authentication-methods>
                                    <publickey/>
                                    <passsword/>
                                    <other>interactive</other>
                                </supported-authentication-methods>
                                <users/>
                            </client-authentication>
                        </ssh-server-parameters>
                    </ssh>
                </endpoint>
```

...

```
    </data>
</config>
```

Slika 5.3. Primer konfiguracije nad kojom je potrebno izvršiti operacije poređenja.

Korišćenjem XPatha implementiran je precizan i siguran pristup za poređenje ovih konfiguracija, na način da se jedinstveno identifikuju svi čvorovi, uključujući i one koji imaju isti naziv ali se nalaze u različitim imenskim prostorima. Korišćenjem regularnih izraza i odgovarajućih funkcija za manipulaciju nad stringovima, implementirana je funkcija return_all_paths (slika 5.4) koja vraća sve putanje do krajnjih čvorova, odnosno listova stabla konfiguracije koji nema podčvorove, već samo vrednost. Kada se dobiju sve putanje čvorova konfiguracije uređaja, kao i sve putanje čvorova očekivane konfiguracije, potrebno je uporediti te putanje. Ovo poređenje je implementirano na način da se svaka putanja iz prve grupe, poredi sa svim putanjama iz druge grupe, što znači da se svaki čvor iz prve konfiguracije, poredi sa svakim čvorom iz druge konfiguracije. Polazi se od čvorova koji su najviši u hijerarhiji, ukoliko se radi o istim čvorovima koji imaju istu putanju i nalaze se u istom imenskom prostoru, nastavlja se poređenje njihovih podčvorova, sve do trenutka dok ne budu jednaki čvor. Ako čvor koji nije ekvivalentan nije list ovog stabla, znači da ti čvorovi nemaju istu putanju (ne radi se o istim parametrima konfiguracije uređaja) i da to zasigurno nije čvor koji je tražen, tako da se poređenje nastavlja dalje sa sledećim čvorovima. Ukoliko je čvor koji nije ekvivalentan list

ovog stabla konfiguracije, identifikovan je čvor koji tražimo, samo njegova vrednost nije ista u obe konfiguracije i taj čvor nam postaje kandidat za čvorove koji predstavljaju problem u konfiguraciji uređaja.

```
def return_all_paths(string, is_real_config, namespace_hashmap):
    paths={}
    counter=5
    counter_no_filter = -1
    realConfiguration = False
    if is_real_config == True:
        counter = 0
        counter_no_filter = -1
    try:
        root = etree.parse(string)
        name_of_namespace = namespace_name(string)
        name_of_namespace = ''.join(('{',name_of_namespace,'}'))
        for e in root.iter():
            if e.text and ((counter >= 3 and namespace_hashmap==None) or
                   counter_no_filter >=3):
                path = root.getelementpath(e)
                pathSplit = path.split("/")
                lastElement = pathSplit[len(pathSplit)-1]
                if "[" in lastElement:
                    pattern = re.compile(r'\[.*\]')
                    newLastElement = re.sub(pattern, '', lastElement)
                    path = path.replace(lastElement, newLastElement
                allBrackets = re.findall("\{(.*?)\}", path)
                if(allBrackets):
                    for one in allBrackets:
                        if namespace_hashmap != None:
                            for nc, namespace in namespace_hashmap.items():
                                if namespace == one:
                                    oneWithBrackets = "{" + one + "}"
                                path = path.replace(oneWithBrackets, nc + ":")
                        else:
                            one = "{" + one + "}"
                            path = path.replace(one, "")


                if(realConfiguration):
                    regex = re.findall("^([^/]+\/[^/]+\/)", path)
                    for one in regex:
                        path = path.replace(one, "")
                path = re.sub(r'ns([0-9])+:', r'', path)
                if e.text.strip():
                    paths[path +'/'+e.text] = 0
                if path == ".":
                    paths.pop(path +'/'+e.text, None)
            else:
                if namespace_hashmap != None:
                    counter_no_filter += 1
                else:
                    counter+=1
                realConfiguration = True
        listPaths = get_all_paths(paths)
    except Exception as e:
        print(e)
        print("Not valid xml format")
        sys.exit(-1)
    return paths
```

Slika 5.4. Primer koda za dohvatanja svih putanja u formi pogodnoj za dalju analizu konfiguracija (Python)

Čvorove koje pronalazimo kao greške u konfiguracijama, možemo klasifikovati u tri grupe:

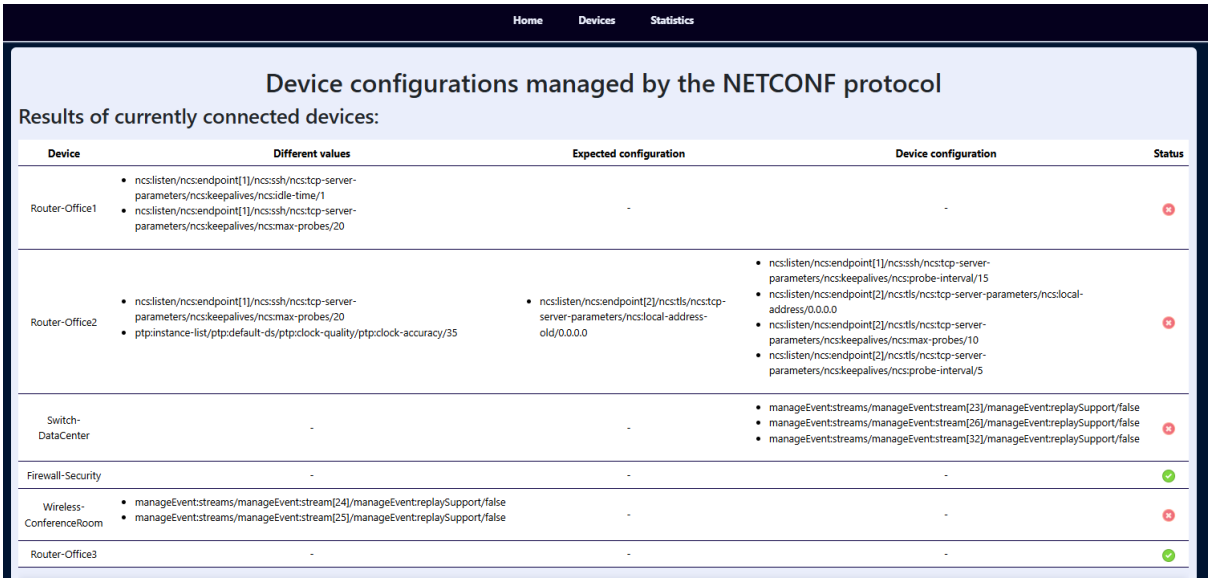
1. **Različite vrednosti istih konfiguracionih parametara** – konfiguracioni parametri koji postoje i u konfiguraciji uređaja i u očekivanoj konfiguraciji, ali se njihove vrednosti razlikuju,

2. **Konfiguracioni parametri koji postoje samo u konfiguraciji uređaja** – postojanje suvišnih elemenata u konfiguraciji,

3. **Konfiguracioni parametri koji ne postoje u konfiguraciji uređaja, ali bi trebalo da postoje** – konfiguracioni parametri postoje u očekivanoj konfiguraciji, što znači da su ti parametri nedostajući u konfiguraciji uređaja.

Ovo poređenje konfiguracija se izvršava u saradnji sa komponentom sistema NETCONF skener čiji je zadatak da korišćenjem NETCONF protokola prikupi konfiguracije sa mrežnih uređaja koji se nalaze u mreži. Skeniranje se pokreće na osnovu zahteva iz message brokera i nakon poređenja konfiguracija u komponenti za procesiranje rezultata, rezultati se vraćaju message brokeru, na ovaj način se obezbeđuje horizontalna skalabilnost, kao i asinhrono izvršavanje.

Sa druge strane message brokera, kao deo sistema je implementiran WEB UI koji predstavlja ključnu komponentu arhitekture sistema jer omogućava korisnicima jednostavan i pregledan uvid o svim uređajima koji se nalaze u mreži, njihovom trenutnom statusu, kao i njihovim konfiguracijama, odnosno trenutnim rezultatima koji su rezultat nadzora i analize ove mreže. Tabelarni prikaz ovih informacija koje predstavljaju intuitivan i jednostavan prikaz ispravnosti svih aktivnih uređaja je prikazan na slici 5.5.



Slika 5.5. Tabelarni prikaz statusa uređaja u mreži

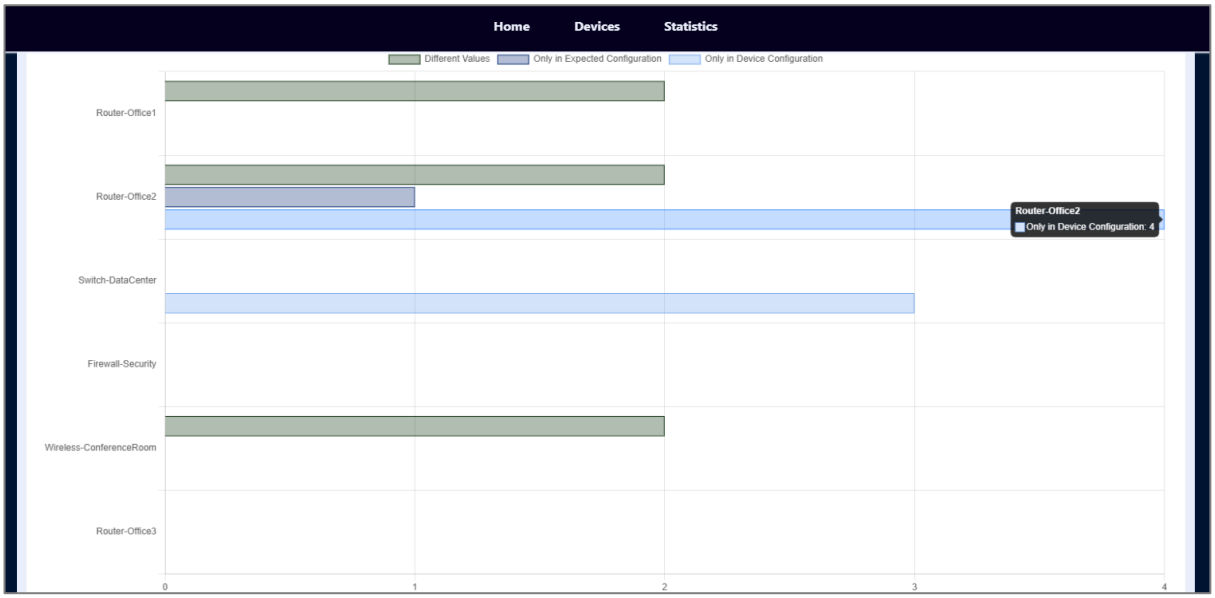
Ovakav pristup omogućava jednostavan uvid u to da li neki uređaj ima neki problem, kao i tačno koji konfiguracioni element odstupa od očekivane vrednosti, ili postoji u konfiguraciji

uređaja a ne postoji u očekivanoj konfiguraciji, ili obrnuto. Ovaj prikaz je omogućen korisnicima odmah na početnoj stranici, kao glavna funkcionalnost ove aplikacije. Trenutno stanje svakog uređaja je prikazano i vizuelnim indikatorom statusa – zeleni „ok“ simbol označava da je konfiguracija uređaja u skladu sa očekivanom konfiguracijom i ovakav uređaj trenutno ne zahteva dodatnu analizu, dok se u slučaju odstupanja od očekivane konfiguracije prikazuje crveni „not ok“ simbol, koji ukazuje na prisustvo greške u konfiguraciji uređaja. U ovom slučaju korisnik može odmah u kolonama *Different values*, *Expected configuration* i *Device configuration* da uoči o kakvim greškama je problem, odnosno koji konfiguracioni element zahteva pažnju i da zatim po potrebi pristupi ostalim funkcionalnostima sistema, koje omogućavaju detaljniju analizu problema, kako bi analizirao problem koji se pojavio na višem nivou. Ako je reč o konfiguracionom elementu čija vrednost odstupa od očekivane vrednosti, ova informacija će se naći u koloni *Different values*. Zatim, ako se radi o konfiguracionom elementu koji nedostaje u konfiguraciji uređaja, ali se očekuje da postoji, odnosno očekivana konfiguracija sadrži ovaj element, on će biti prikazan u koloni *Device configuration*, jer predstavlja nedostajući element u ovoj konfiguraciji. Suprotno, ako se radi o konfiguracionom elementu koji je neočekivano prisutan u konfiguraciji uređaja, odnosno očekivana konfiguracija ne sadrži ovaj konfiguracioni element, ta informacija će biti prikazana u koloni *Expected configuration*.

Ovakav prikaz omogućava korisnicima brz uvid u trenutno stanje mreže, bez potrebe za ulaskom u detalje pojedinačne konfiguracije, kao i identifikaciju kritičnih uređaja koji zahtevaju hitnu intervenciju. Ovim zaključkom se fokus mrežnih administratora može odmah usmeriti ka kritičnim uređajima koji predstavljaju najveću pretnju sistemu s obzirom da se uočava koji uređaj ima najviše grešaka i kakvog su tipa te greške.

Tabelarni prikaz je takođe prilagođen i korisnicima koji nemaju dovoljno tehničkog znanja o XML konfiguracijama ili protokolima, jer im na jednostavan način omogućava da lako prate stanje mreže.

Osim tabelarnog prikaza, sistem nudi i grafički prikaz ovih rezultata koji je prikazan na slici 5.6.

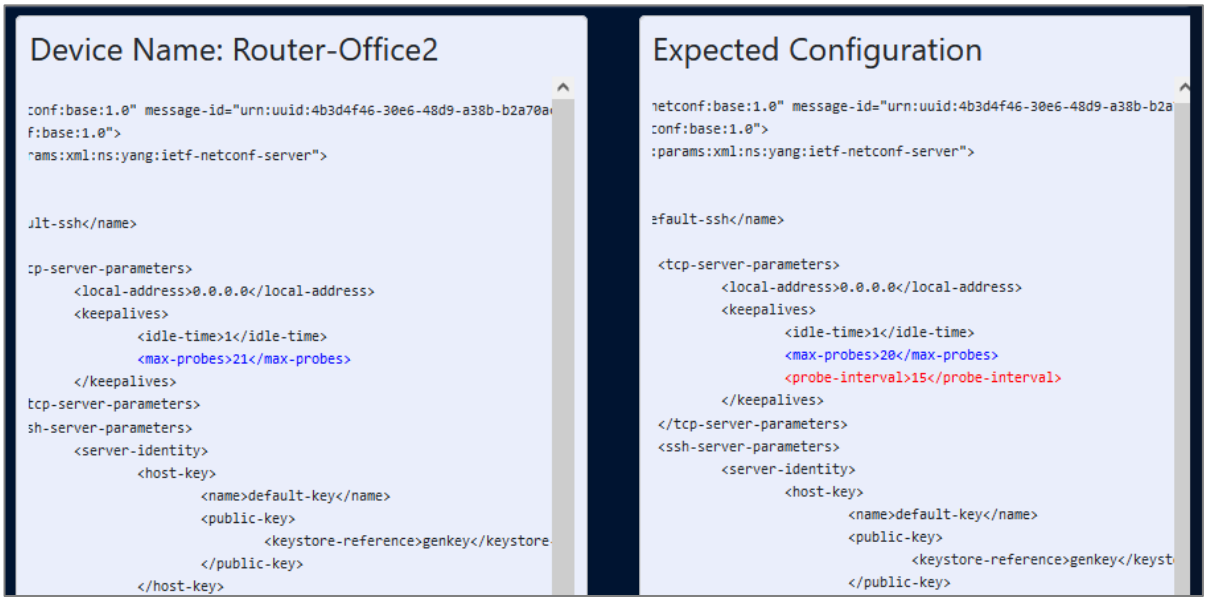


Slika 5.6. Grafički prikaz statusa uređaja u mreži

Ovaj prikaz koristi vizuelizaciju u obliku stubičastog dijagrama koji predstavlja odličnu opciju za prikazivanje količine i tipova grešaka po uređajima u mreži. Na ovom dijagramu, svaki stub predstavlja ukupan broj grešaka jedne vrste za jedan uređaj, kako bi se jasno istakli uređaji sa najvećim brojem grešaka, odnosno neslaganja u konfiguracijama. Kako bi se različiti tipovi grešaka predstavili na najintuitivniji način, različitim bojama su obeleženi različiti tipovi grešaka, što je i prikazano na legendi grafa. Zelenom bojom predstavljene su greške koje se odnose na različite vrednosti konfiguracionih parametara u očekivanoj konfiguraciji i konfiguraciji uređaja. Tamno plavom bojom prikazane su greške u konfiguraciji koje se odnose na postojanje konfiguracionog parametra samo u očekivanoj konfiguraciji, dok su svetlo plavom bojom označene greške koje predstavljaju postojanje suvišnih konfiguracionih parametara u konfiguraciji uređaja u mreži.

Grafički prikaz omogućava korisniku da na intuitivan način identifikuje kritične tačke u mreži, tj. uređaje koji najčešće prave probleme i zahtevaju dodatnu pažnju i analizu. Ova vrsta prikaza je posebno korisna u mrežama sa velikim brojem uređaja, gde bi pregledanje svakog uređaja ručno bilo izuzetno vremenski zahtevno, nepraktično, skupo i sklono greškama. Vizuelizacijom podataka u obliku stubičastog dijagrama se omogućava brzo donošenje odluka o prioritetima za analizu i održavanje, što će dovesti i do brzog rešavanja novonastalog problema.

Za svaki uređaj postoji i opcija da se uporedo prikažu trenutna i očekivana konfiguracija, pri čemu su razlike u konfiguracijama obeležene različitim bojama. Ovaj prikaz poređenja konfiguracija je prikazan na slici 5.7.



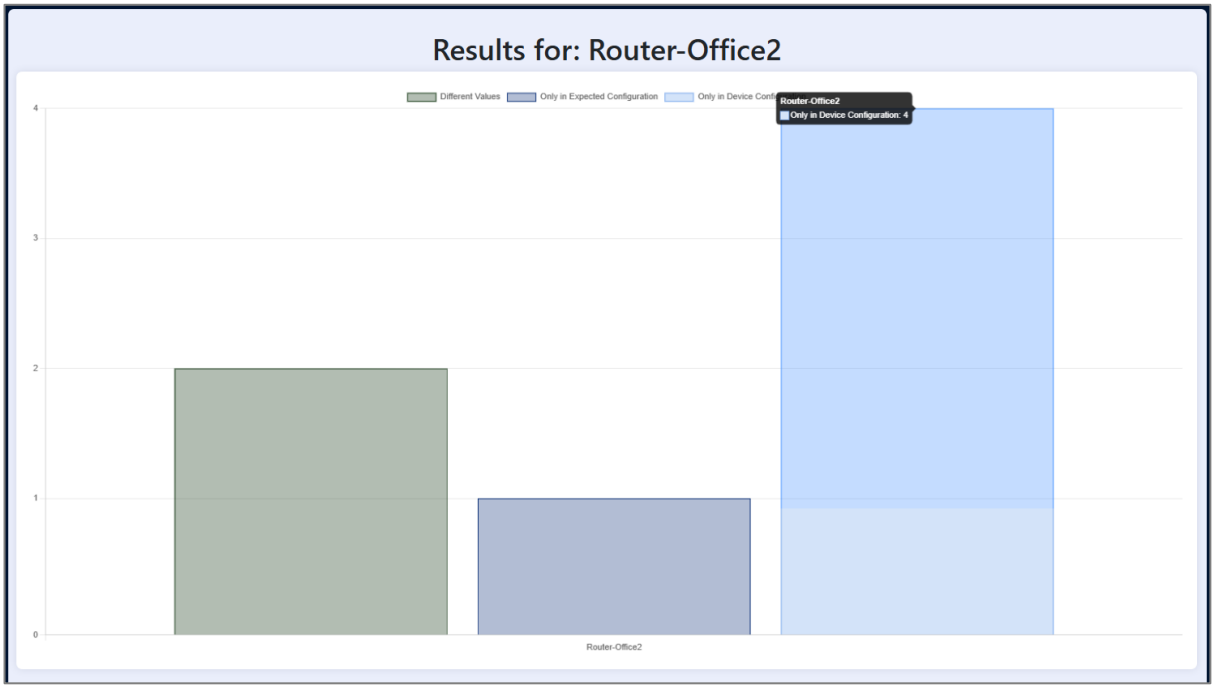
Slika 5.7. Uporedni prikaz trenutne konfiguracije uređaja i očekivane konfiguracije

Ovakav prikaz omogućava korisniku da precizno identifikuje koje se vrednosti ne poklapaju, bez potrebe za ručnim analiziranjem XML fajlova, koji predstavlja dugotrajan proces podložan greškama. Ovim se značajno smanjuje vreme potrebno za dijagnostiku problema i donošenja odluka u koracima koje treba preduzeti kako bi se greške rešile na najbolji mogući način. Delovi konfiguracije koji su u redu su prikazani kao običan tekst crne boje, dok su potencijalni problemi vizuelno označeni bojama, čime se dodatno povećava preglednost:

- Plavom bojom označeni su elementi konfiguracije koji se nalaze i u konfiguraciji uređaja i u očekivanoj konfiguraciji ali se njihova vrednost razlikuje. Ovim prikazom korisnik odmah može da uoči koja je razlika, odnosno koja je vrednost u konfiguraciji uređaja, a koja u očekivanoj konfiguraciji, što može pomoći prilikom rešavanja problema jer već posedujemo sve potrebne informacije.

- Crvenom bojom označeni su elementi koji se nalaze samo u jednoj od konfiguracija, ili u konfiguraciji uređaja (u tom slučaju je taj element prikazan samo u konfiguraciji uređaja), ili u očekivanoj konfiguraciji (u ovom slučaju je taj element prikazan samo u očekivanoj konfiguraciji).

Takođe, pored funkcionalnosti prikaza konfiguracije uređaja u tekstualnom obliku i vizuelnog izdvajanja konfiguracionih parametara koji nisu usaglašeni, moguće je videti i grafički prikaz ukupnog broja grešaka za izabrani uređaj, vizuelno sličan grafiku za sve uređaje koji je prikazan na početnoj stranici (slika 5.6). Ova vizuelizacija grešaka je implementirana kao stubičasti dijagram u kom svaki stub označava broj grešaka zavisno od tipa greške koja se pojavljuje. Iznad grafika, postoji legenda koja definiše vrstu greške na osnovu boje stuba. Grafik koji predstavlja ukupan broj grešaka za odabrani uređaj na osnovu tipa greške, prikazan je na slici 5.8.



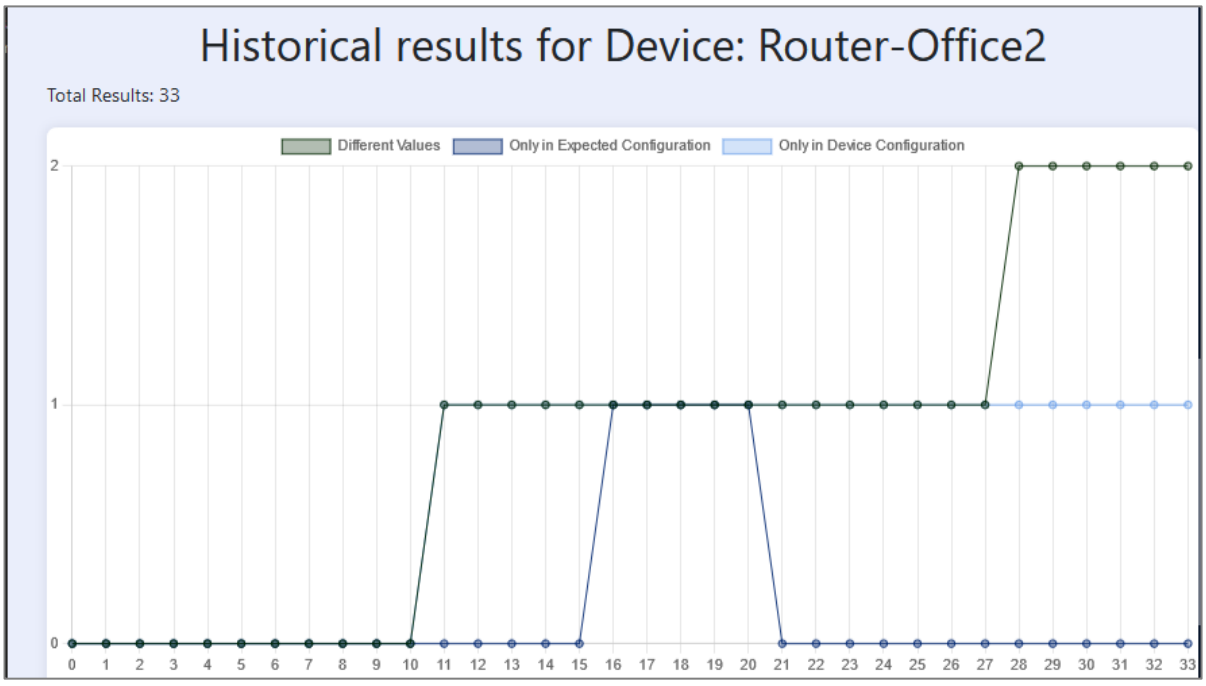
Slika 5.8. Grafički prikaz ukupnog broja grešaka za odabrani uređaj

Ispod grafičkog prikaza grešaka za uređaj koji smo odabrali, nalazi se i tabelarni prikaz, sličan tabeli za sve uređaje koja se nalazi na početnoj stranici (slika 5.5), za razliku od tabele na početnoj stranici, ovde su izdvojene samo greške koje se nalaze na odabranom uređaju. Na ovaj način, korisnik ima mogućnost da se fokusira samo na greške vezane za ovaj uređaj, što je bitno prilikom analize stanja uređaja u mreži. Tabelarni prikaz svih grešaka za odabrani uređaj, prikazana je na slici 5.9.

Different Values	Only in Expected Configuration	Only in Device Configuration
• ncs:listen/ncs:endpoint[1]/ncs:ssh/ncs:tcp-server-parameters/ncs:keepalives/ncs:max-probes/20 • ptp:instance-list/ptp:default-ds/ptp:clock-quality/ptp:clock-accuracy/35	• ncs:listen/ncs:endpoint[2]/ncs:tls/ncs:tcp-server-parameters/ncs:local-address-old/0.0.0.0	• ncs:listen/ncs:endpoint[1]/ncs:ssh/ncs:tcp-server-parameters/ncs:keepalives/ncs:probe-interval/15 • ncs:listen/ncs:endpoint[2]/ncs:tls/ncs:tcp-server-parameters/ncs:local-address/0.0.0.0 • ncs:listen/ncs:endpoint[2]/ncs:tls/ncs:tcp-server-parameters/ncs:keepalives/ncs:max-probes/10 • ncs:listen/ncs:endpoint[2]/ncs:tls/ncs:tcp-server-parameters/ncs:keepalives/ncs:probe-interval/5

Slika 5.9. Tabelarni prikaz svih grešaka raspoređenih prema tipu greške za odabrani uređaj

Još jedna važna funkcionalnost sistema je da je za svaki uređaj moguće prikazati i grafik koji predstavlja vizuelni istorijat njegovog statusa. Ovakvim načinom prikaza promena u konfiguraciji uređaja dobijamo mogućnost praćenja stanja konfiguracije kroz vreme, dobijajući mnogo jasniju sliku o dinamičnosti i stabilnosti uređaja na ovoj mreži. Na ovaj način se korisniku omogućava jasniji uvid u učestalost promena u konfiguraciji. Ovaj pregled konfiguracije uređaja kroz istoriju, omogućava korisniku da proceni pouzdanost određenog uređaja. Vizuelni istorijat predstavljen kroz vremensku osu i linijski grafik koji prikazuje istoriju konfiguracija za odabrani uređaj prikazan je na slici 5.10.



Slika 5.10. Grafički prikaz istorije grešaka za određeni uređaj

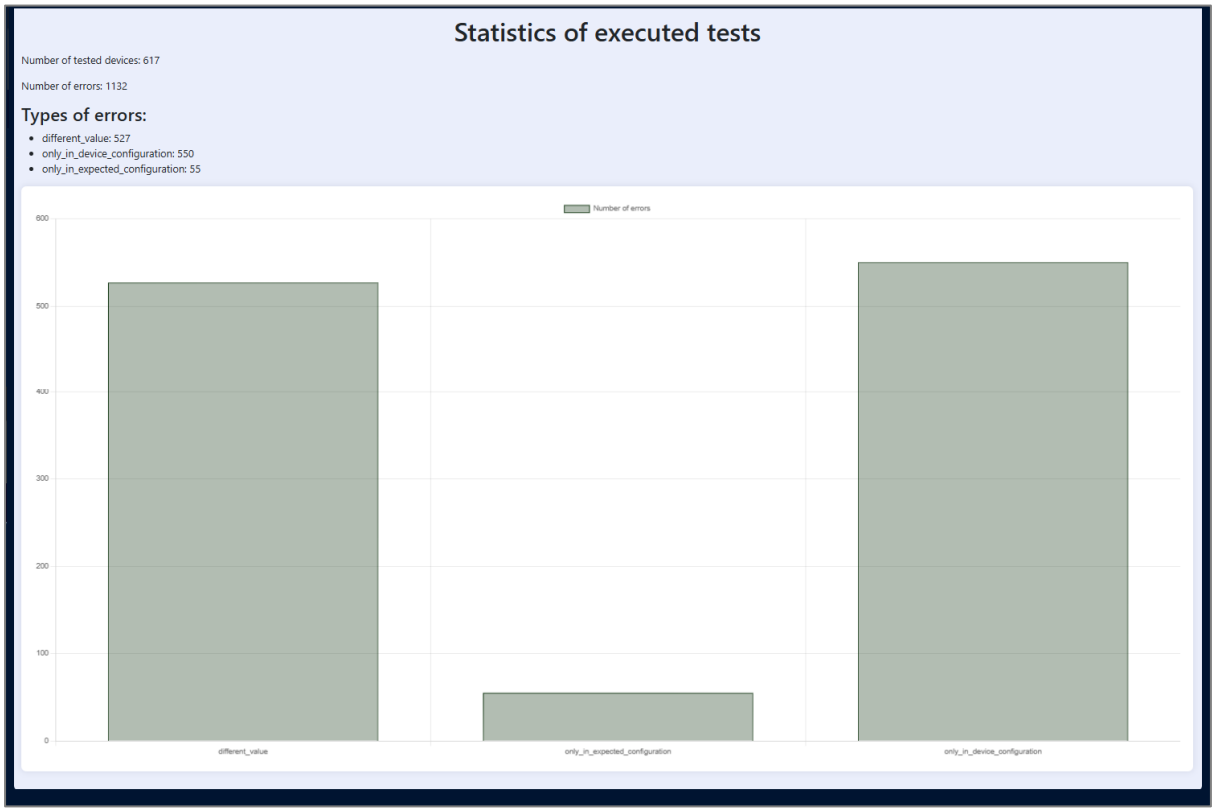
Ovakav pregled omogućava:

- Identifikaciju uređaja na kome se dešavaju učestale promene konfiguracije, što može ukazivati na potrebu za dodatnom stabilizacijom,
- Praćenje efekata nadogradnji, jer je ova neusaglašenost konfiguracija možda i posledica nadogradnji i promena koje su primenjene na uređaj,
- Analizu pouzdanosti i stabilnosti uređaja, što predstavlja veoma važnu karakteristiku za automatizovani nadzor mreže zbog dugoročnog planiranja i unapređivanja mrežne infrastrukture.

Na osnovu ovih podataka, moguće je detektovati potencijalne probleme koji dovode do učestalih grešaka, kao i kreirati strategiju preventivnog održavanja, čime će se umanjiti rizik od pojavljivanja novih i sličnih problema.

Sistem takođe omogućava i istorijski prikaz rezultata za sve uređaje koji se nalaze ili su se nalazili u mreži nadgledane kroz vreme, kao što je prikazano na slici 5.11.

Ovi podaci su zapravo grafički prikaz razlika u konfiguracijama, gde je prikazan ukupan broj grešaka u zavisnosti od tipa greške: određeni konfiguracioni parametar postoji u obe konfiguracije, ali su vrednosti u jednoj konfiguraciji pogrešne, određeni konfiguracioni parametar postoji u stvarnoj konfiguraciji uređaja ali ne postoji u očekivanoj konfiguraciji, ili obrnuto – konfiguracioni parametar postoji u očekivanoj konfiguraciji ali je izostao u realnoj konfiguraciji uređaja. Ovaj grafički prikaz je realizovan preko stubičastog dijagrama, gde svaki stub predstavlja različit tip greške.



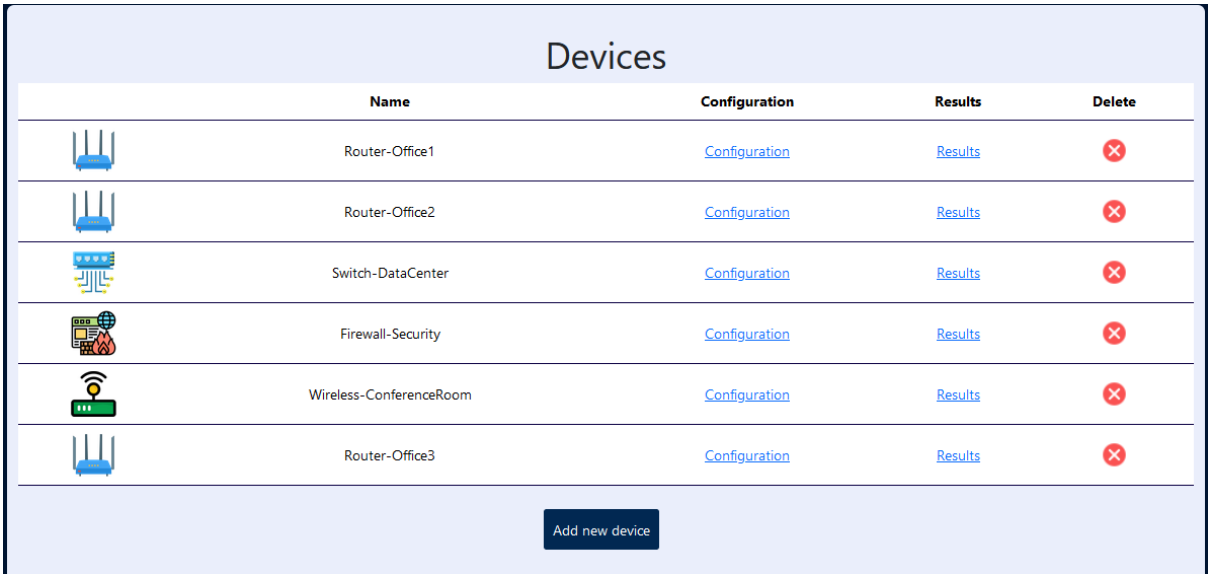
Slika 5.11. Grafički prikaz istorijskih podataka za sve uređaje

Grafik omogućava korisniku da brzo uoči obrasce grešaka i stekne uvid u dominantne tipove problema u mreži. Ova analiza može pomoći u planiranju unapređenja konfiguracionih politika i otkrivanju grešaka, jer npr. često pojavljivanje grešaka tipa „konfiguracioni parametar nedostaje u konfiguraciji uređaja“, može ukazivati na problem loše primenjene šeme konfigurisanja uređaja određene vrste.

Čuvanje istorijskih podataka za sve uređaje koji su se ikad nalazili u mreži krije puno veće značenje. Sistem trenutno nudi prikaz istorijskih podataka kao dijagram koji prikazuje ukupan broj grešaka na osnovu tipa greške za sve uređaje koji su se nalazili i koji se trenutno nalaze u mreži, ali ova mogućnost čuvanja istorijskih podataka, predstavlja dobru osobinu sistema za dublju analizu grešaka, sprečavanja pojavljivanja srodnih grešaka kao i moguća proširenja i automatizaciju.

Takođe u okviru WEB UI-a, kao što je prikazano na slici 5.12, sistem omogućava i upravljanje uređajima koji se nalaze u mreži, što značajno doprinosi njegovoj praktičnoj primeni i fleksibilnosti. Korisniku je omogućen jednostavan pregled svih aktivnih mrežnih uređaja kao i intuitivne kontrole za upravljanje sa njima. Pored osnovnog prikaza statusa konfiguracije svakog uređaja, sistem pruža opcije za dodavanje novih uređaja u sistem, čime se automatski inicira proces inicijalne provere konfiguracije i njeno upoređivanje sa unapred definisanom očekivanom konfiguracijom. Ova funkcionalnost je posebno korisna u dinamičnim mrežama u kojima se uređaji često dodaju ili uklanjaju.

Sa druge strane, ukoliko neki uređaj više nije aktivan ili je privremeno van funkcije, korisnik ima mogućnost da ga deaktivira iz sistema, odnosno obriše sa trenutne mreže. Brisanje uređaja sa mreže ne podrazumeva njegovo fizičko brisanje iz baze podataka, već se uređaj označava kao neaktivan, čime se sprečava njegovo dalje uključivanje u procese nadzora i analize. Na taj način se čuva istorija prethodnih konfiguracija i rezultata, što može biti od značaja za kasnije analize.



Slika 5.12. Operacije nad uređajima koji se nalaze u mreži

5.6. Mobilna aplikacija

U cilju unapređenja pristupačnosti i fleksibilnosti, osnovne funkcionalnosti ovog sistema za nadzor, analizu i upravljanje konfiguracijama mrežnih uređaja zasnovanom na NETCONF protokolu implementirane su i kroz mobilnu aplikaciju. Ova aplikacija omogućava administratorima mreže da pristupaju sistemu „u hodu“, nezavisno od lokacije na kojoj se nalaze. Ovakav pristup diže nivo stabilnosti mreže na još viši nivo, jer je nadzor moguće vršiti i van radnog okruženja što omogućava brzo reagovanje u situacijama kad je to neophodno.

Mobilna aplikacija omogućava:

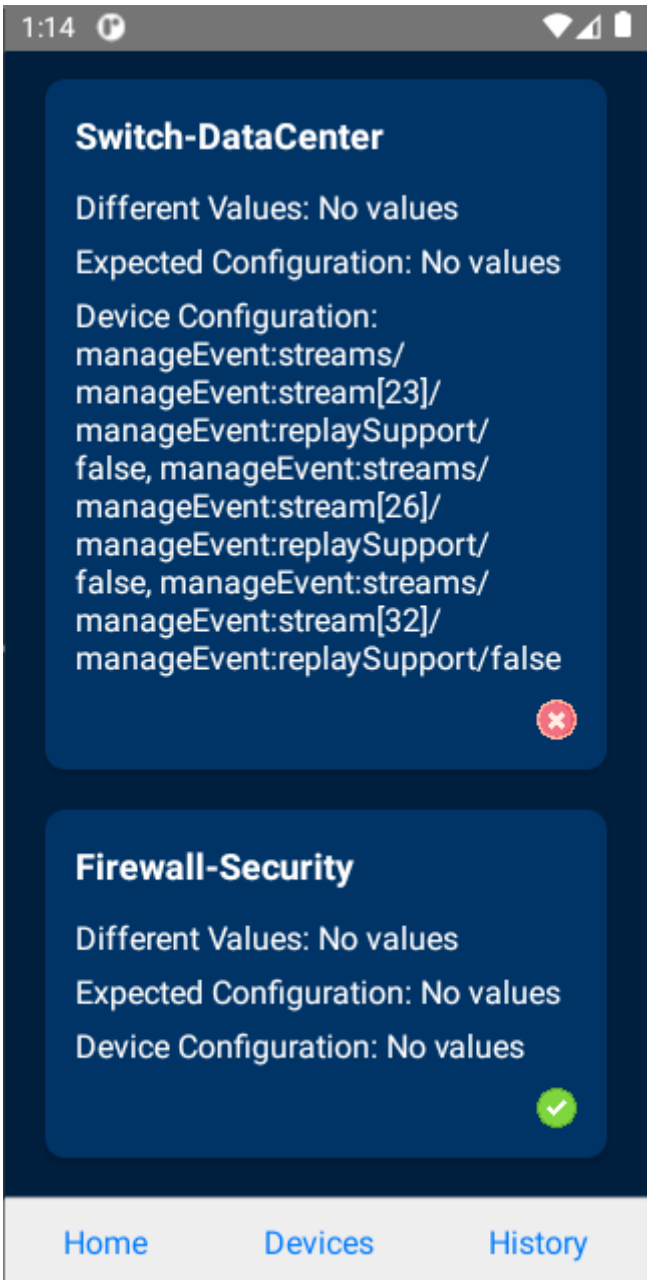
- Brz i pregledan uvid u trenutno stanje uređaja u svakom trenutku čime se obezbeđuje kontinuirano praćenje i reagovanje na potencijalne probleme u realnom vremenu,
- Prikaz rezultata poređenja konfiguracije uređaja u poređenju sa očekivanom konfiguracijom, kao i informacije o eventualnim greškama koje su nastale prilikom ovog poređenja konfiguracija,
- Prikaz istorijskih konfiguracionih promena za svaki uređaj koji se nalazi u mreži što omogućava analizu grešaka koja može dovesti do lakše detekcije uzroka grešaka ili ponavljajućih problema,
- Osnovne funkcionalnosti upravljanja uređajima.

Na početnoj stranici mobilne aplikacije za nadzor mrežnih uređaja putem NETCONF protokola prikazani su trenutni rezultati uređaja koji se nalaze u mreži, slično kao i na početnoj stranici veb-bazirane korisničke aplikacije, s tim da je ovaj prikaz prilagođen manjem ekranu i mobilnom korisničkom iskustvu. Cilj ovog prikaza je da korisnik može jednim pogledom da uoči eventualne greške i odstupanja sa bilo koje lokacije. Ovi rezultati podrazumevaju grafički prikaz grešaka u obliku stubičastog dijagrama na osnovu toga na kom se uređaju detektovala greška i kog je tipa. Iznad stubičastog dijagrama, nalazi se legenda, gde se može videti kojom bojom je označen koji stub u zavisnosti od tipa greške. Početna stranica prikazana je na slici 5.13.

Ovakav prikaz rezultata nadzora konfiguracija uređaja u mreži, kao i kod veb-bazirane korisničke aplikacije omogućava lako i jednostavno identifikovanje grešaka na uređaju, kao i koji je tip greške u zavisnosti od toga da li je konfiguracioni parametar sadržan u obe upoređene konfiguracije, ali se njihova vrednost razlikuje, da li je konfiguracioni parametar sadržan samo u konfiguraciji uređaja a ne i u očekivanoj konfiguraciji, ili se nalazi samo u očekivanoj konfiguraciji, odnosno nedostaje u konfiguraciji uređaja.

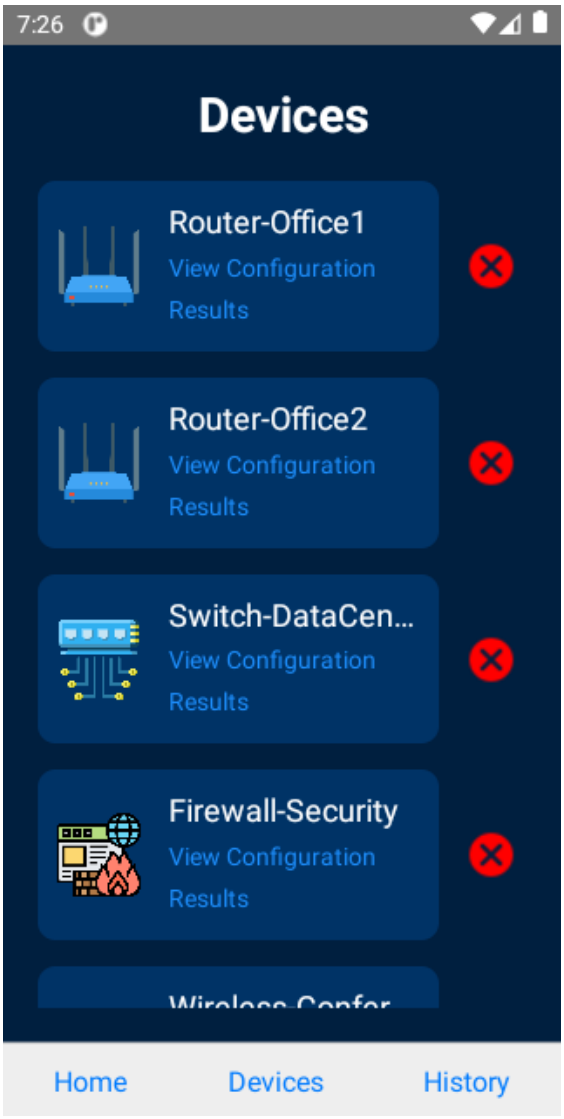
Slika 5.13. Grafički prikaz stanja uređaja u mreži – mobilna aplikacija

Ispod grafičkog prikaza, omogućen je detaljan prikaz svih grešaka, tj. čitava putanja greške je prikazana u zavisnosti od tipa kojoj greška pripada. Ispod prikaza svih grešaka za jedan uređaj, na osnovu vizualnog indikatora statusa predstavljeno da li je konfiguracija u redu ili postoje određeni problemi. Kao vizuelni indikator za status uređaja postavljen je zeleni „ok“ znak ako je sve u redu, ili crveni „not ok“ ukoliko postoji neka greška. Ako je status „not ok“, odmah na početnoj stranici je moguće videti koji konfiguracioni podaci predstavljaju potencijalnu pretnju za sistem, kao i kog tipa je greška vezana za taj parametar, što već omogućava mrežnom administratoru osnovne informacije za dalju analizu. Ovaj prikaz detaljnijih informacija za svaki uređaj na početnoj stranici prikazan je na slici 5.14.



Slika 5.14. Prikaz detaljnijih informacija o statusu svih uređaja koji se nalaze u mreži

Mobilna aplikacija kao i WEB UI, intuitivno kroz grafik pruža korisniku informacije o konfiguracijama uređaja koji se nalaze trenutno u mreži, što znači da korisnik može odmah prilikom pristupanja aplikaciji da ima uvid u to da li neki uređaj odstupa od očekivanog ponašanja, koji je to uređaj i koji tip greške uređaj ima. Klikom na tab *Devices* u glavnom meniju mobilne aplikacije, korisnik može preći na funkcionalnosti vezane za uređaje, što je prikazano na slici 5.15.

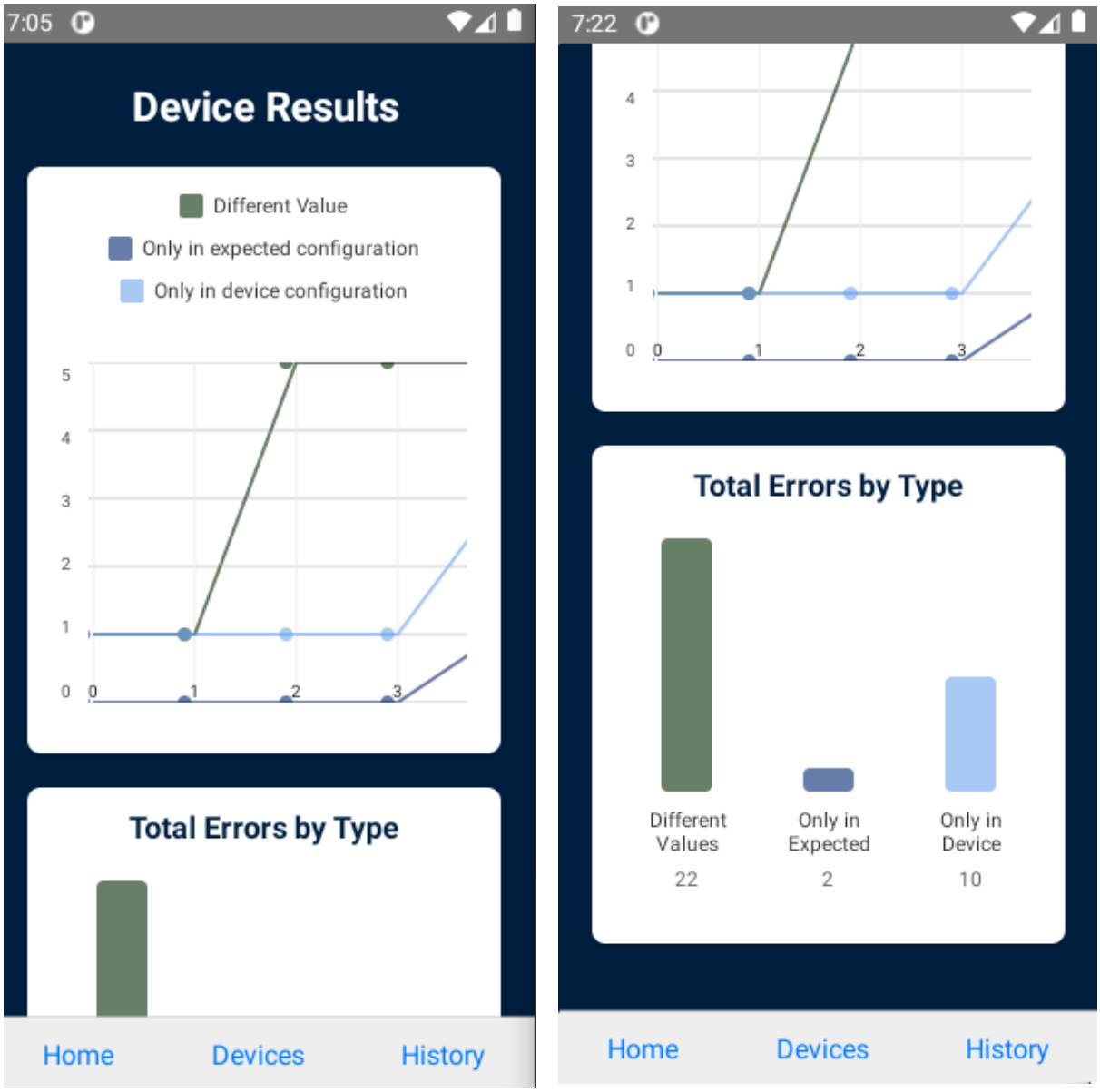


Slika 5.15. Stranica za upravljanje uređajima

Korisnik ima mogućnost da vidi konfiguraciju bilo kog uređaja koji se trenutno nalazi u mreži, kao i da pregleda rezultate za konkretni uređaj koji se mogu podeliti u dve grupe:

1. Grafički prikaz istorijskih neslaganja konfiguracija – ovo je prikazano kao vremenska osa sa jasno prikazanim tačkama u kojima su se pojavile određene greške za izabrani uređaj. Ovaj prikaz daje potrebne informacije koje se mogu iskoristiti kako bi se izračunala učestalost pojavljivanja ovog problema, a time se može doći do uzroka problema i isplanirati preventivno održavanje. Takođe, na ovaj način se procenjuje stabilnost uređaja.
2. Ukupan broj neslaganja konfiguracija na tom uređaju – stubičasti prikaz ukupno detektovanih grešaka na osnovu tipa greške, koji takođe pomaže pri analizi uređaja kada je u pitanju pouzdanost uređaja i mogućnost nadogradnje zarad poboljšanja stabilnosti uređaja.

Grafički prikaz neslaganja konfiguracija kroz vreme postojanja uređaja u mreži, kao i zbirno, prikazani su na slici 5.16. Ovi prikazi osim uvida u stanje uređaja, daju najbitniju osnovu za dublju analizu uzroka problema.



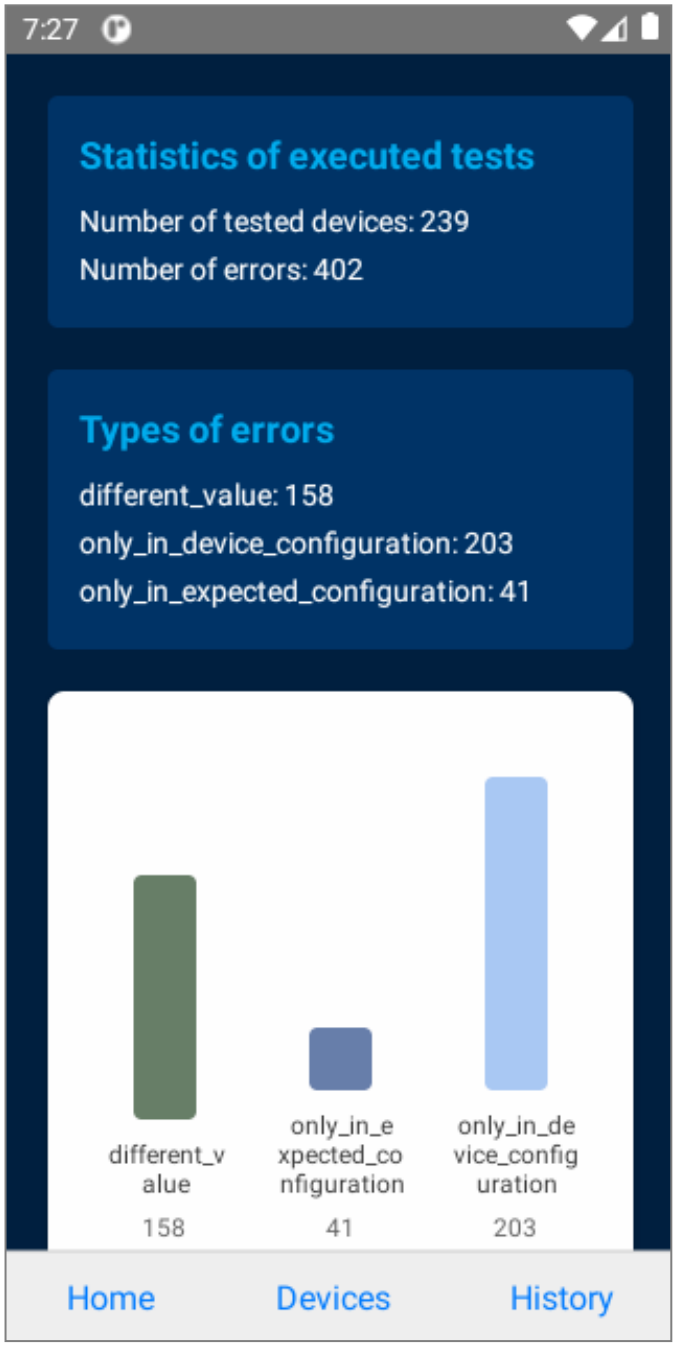
Slika 5.16. Grafički prikazi neslaganja konfiguracija kroz vreme i ukupan broj tih neslaganja

Na kraju, mobilna aplikacija, kao i WEB UI omogućava korisniku uvid u sveobuhvatnu statistiku nadgledanja ove mreže kroz grafički prikaz neslaganja konfiguracija po tipu kroz vreme. Ova statistika obuhvata podatke o uređajima koji se trenutno nalaze u mreži, kao i o svim uređajima koji su nekad bili nadgledani. Ova funkcionalnost aplikacije pruža širu sliku o celokupnom stanju infrastrukture, jer obuhvata i uređaje koji se trenutno nalaze u mreži, ali i one koji su ranije bili nadgledani.

Ovakva statistika je posebno bitna u velikim i dinamičnim mrežnim infrastrukturama gde se uređaji često dodaju, uklanjanju ili konfigurišu jer omogućava:

- Identifikaciju najčešćih tipova grešaka,
- Procenu stabilnosti mreže,
- Donošenje strateških odluka za bolje održavanje i unapređivanje stabilnosti mreže.

Grafički prikaz prikazan statistike nadgledanja mreže kroz vreme prikazan je na slici 5.17.



Slika 5.17. Statistika nadgledanja mreže kroz vreme

Razvoj ovakve mobilne aplikacije ima za cilj da postigne responzivan dizajn i jednostavnost pri korišćenju. Mobilna aplikacija implementirana na ovaj način predstavlja intuitivno, jednostavno, moderno i uvek dostupno rešenje za upravljanje konfiguracijama mrežnih uređaja, prateći savremene trendove i time omogućava veću efikasnost mrežnih administratora i veću stabilnost i sigurnost mreže.

6. DISKUSIJA

U ovom redu predložen je sistem za nadgledanje, analizu i upravljanje konfiguracijama mrežnih uređaja zasnovan na NETCONF protokolu. Kombinacija NETCONF protokola i YANG jezika za modelovanje čini precizan, jednostavan i praktičan pristup za analizu i nadzor konfiguracija mrežnih uređaja, zbog jednostavnosti dohvatanja, prikazivanja i manipulacije podacima. YANG predstavlja dobar izbor kao jezik za modelovanje podataka iz razloga što prikazuje strukturu podataka u formi XML stabla, što omogućava jasan i sažet opis svakog čvora u hijerarhiji, kao i međusobnu interakciju čvorova. XML kao format je dobar izbor u ovom pravcu, jer većina programskih jezika podržava rad sa XML formatom i poseduju određene biblioteke koje olakšavaju rad sa XML formatom i ujedno svaku operaciju nad XML formatom čine jednostavnijom i bržom.

Ideja za razvoj sistema za nadgledanje, analizu i upravljanje konfiguracijama mrežnih uređaja zasnovanog na NETCONF protokolu potiče iz činjenice da nadzor mreže predstavlja ključni proces za pravilno, pouzdano i efikasno funkcionisanje mrežnih uređaja i same infrastrukture. Pre mnogo godina, upravljanje mrežnim infrastrukturama je bilo mnogo jednostavnije i lakše, jer su i same mrežne infrastrukture bile manje, jednostavnije i statičnije nego danas. Danas je broj uređaja koji čini jednu mrežnu infrastrukturu uglavnom jako veliki, rad tih uređaja je jako dinamičan, a rizik koji prekid u radu nekog uređaja nosi sa sobom je znatno veći. Kako bi se obezbedilo da mrežni uređaji funkcionišu ispravno, efikasno i neprekidno, potrebno je izvršiti kvalitetan proces nadzora nad tim uređajima. Nadzor mrežnih uređaja podrazumeva konstantno praćenje mrežnih uređaja kako bi se pravovremeno otkrile, identifikovale i otklonile greške. Idealno bi bilo da se nadgledanje ovakve mreže vrši u realnom vremenu kako bi se omogućila najbrža moguća reakcija na problem i na taj način se očuvala stalna dostupnost uređaja u mreži.

6.1. Poređenje sa postojećim rešenjima iz literature

Analiza sličnih radova pokazuje da većina postojećih sistema za nadzor mreža i dalje koristi SNMP kao osnovni protokol. Iako je SNMP rasprostranjen i dobro podržan, njegova ograničenja u pogledu bezbednosti (oslanjanje na UDP, ograničena enkripcija) i detaljnosti podataka čine ga manje pogodnim za današnje kompleksne scenarije. NETCONF, za razliku od toga, omogućava transakcioni pristup konfiguraciji, bezbedan prenos podataka i jasno razdvajanje konfiguracionih i statusnih informacija. Time se ne samo povećava preciznost detekcije problema, već se i olakšava automatizacija procesa.

Rezultati dobijeni testiranjem pokazuju da predloženi sistem može da detektuje grešku u konfiguraciji u roku od nekoliko sekundi od njenog nastanka, dok klasična rešenja zasnovana na ručnom praćenju ili periodičnom SNMP prozivanju uređaja često imaju kašnjenje koje se meri minutima. Ova razlika je posebno značajna u okruženjima gde i kratkotrajni prekid usluge može dovesti do značajnih poslovnih gubitaka.

6.2. Funkcionalnost i prednosti sistema

Najmodernija rešenja za nadgledanje mrežne infrastrukture teže ka centralizovanom i automatizovanom pristupu nadzora, sa ciljem da implementirani sistem za nadzor samostalno prikuplja informacije, obrađuje ih u realnom vremenu i dostavlja obrađene rezultate preko intuitivnog korisničkog interfejsa. Upravo je to i ideja za nastanak ovog rešenja, da se koristeći NETCONF protokol koji znatno olakšava rad sa samim uređajem, koji podrazumeva dohvatanje konfiguracije, modifikovanje konfiguracije, kao i brisanje konfiguracije, implementira sistem koji će u realnom vremenu vršiti nadzor svih konfiguracija uređaja koji se nalaze u mreži, pravovremeno identifikovati probleme na određenim uređajima, na intuitivan način dostaviti rezultate nadzora korisniku sa fokusom na greškama koje su pronađene i omogućiti korisniku prikaz statističkih podataka, kako za sam uređaj, tako i za sve uređaje ikada u mreži. Ovi statistički podaci služe kao odličan izvor informacija za analizu grešaka, koja u velikoj meri može olakšati pronalaženje problema i njegovog rešenja, kao i buduću nadogradnju koja može obezbediti automatizovano rešavanje učestalih problema.

Kombinacijom svega prethodno navedenog dobija se moćan alat za upravljanje velikim i dinamičnim mrežnim infrastrukturama. Implementirani sistem prikuplja podatke o radu uređaja koristeći NETCONF protokol i oslanjajući se na sve specifičnosti koje sa sobom nosi YANG modul i Python programski jezik. Korišćenjem XPath alata omogućeno je precizno poređenje stvarne konfiguracije uređaja u mreži i očekivane konfiguracije za dati uređaj. Ujedno zbog specifičnosti NETCONF protokola i njegove komunikacije pomoću TCP protokola, ostvaren je visok nivo bezbednosti koji npr. ne bi bio postignut da se umesto NETCONF protokola koristio SNMP protokol. Osim što prikuplja podatke, sistem detektuje nekonzistentnosti u konfiguracijama i po potrebi, automatski reaguje, odnosno informiše korisnika o određenim problemima. Na ovaj način je znatno smanjena mogućnost nastanka greške od strane čoveka i obezbeđeno je konstantno funkcionisanje uređaja.

Glavna prednost ovog sistema jeste da osim preciznog i pravovremenog uočavanja neke promene u konfiguraciji, sistem pronalazi konkretan deo konfiguracije koji pravi problem, bilo da se radi o pogrešnoj vrednosti nekog parametra, nedostajućem parametru u konfiguraciji ili suprotno, postojanju suvišnog parametra. Koristeći NETCONF protokol, YANG modul, Python programski jezik sa bogatim skupom njegovih biblioteka, Flask, WEB UI implementiran u React-u, message broker RabbitMQ koji omogućava asinhronu komunikaciju i nesmetano funkcionisanje i MongoDB bazu podataka, moguće je detaljno izdvojiti razlike u konfiguracijama i njih vizuelno predstaviti, što znatno olakšava identifikaciju uzroka problema i njegovo otklanjanje.

6.3. Ograničenja trenutnog rešenja

Iako je implementirani sistem funkcionalan i pouzdan, postoje određena ograničenja koja treba naglasiti. Prvo, zavisnost od NETCONF agenta na mrežnim uređajima znači da primena nije univerzalna u okruženjima sa starijom opremom. Drugo, sistem u trenutnom obliku ne sprovodi automatsku korekciju uočenih grešaka — iako može da identifikuje problem i obavesti administratora, ljudska intervencija je i dalje potrebna. Treće, iako se prikupljaju istorijski podaci o konfiguracijama, trenutna baza podataka nije dovoljno velika da bi omogućila obuku naprednih modela mašinskog učenja za prediktivnu analitiku.

6.4. Potencijalne nadogradnje i primene

Čuvanje istorijskih podataka o statusu uređaja, pruža mogućnost za moćna proširenja ovog sistema, koja bi omogućila prepoznavanje ponavljajućih konfiguracionih grešaka, što bi obučavanjem neuronske mreže dovelo do mogućeg automatizovanog rešenja često ponavljanih problema. Ukoliko bi se sistem proširio u ovom smeru, omogućena bi bila i funkcionalnost prediktivnog upravljanja mrežom. Da bi bilo moguće ostvariti ovu funkcionalnost sistema, potrebno bi bilo izgraditi pouzdanu bazu podataka sa ogromnim brojem istorijskih primera, iz koje će se procesom učenja moći prepoznati obrasci ponašanja i uzročno-posledične veze između određene konfiguracije uređaja i greške koja se javlja. Implementirani sistem, već funkcioniše u tom smeru, u smislu da puni bazu podataka istorijskim podacima nadgledanja uređaja, kreirajući različite evidencije o dešavanju grešaka, kao i podeli prema tipu greške. Ukoliko bismo nastavili u ovom smeru proširivati sistem, trebalo bi se dublje fokusirati na analizu grešaka. U praksi, nadzor mreže podrazumeva nadzor velikog broja uređaja različitih tipova (ruteri, svičevi...). Svaki tip uređaja poseduje specifične konfiguracije koje su ključne za pravilno funkcionisanje mreže. Čak i najmanja greška u konfiguraciji može dovesti do određenog problema u radu ili do potpunog prekida.

Na osnovu implementiranog sistema, moguće je prikupiti ogroman broj različitih konfiguracija i njihovih grešaka. Koristeći ove podatke moguće je prepoznati scenarije koji se ponavljaju i da se u skladu sa tim reaguje. Nakon što sistem prepozna grešku, mogao bi automatski da koriguje određene parametre ili da uputi obaveštenje administratoru sa preciznim koracima koje je potrebno odraditi kako bi se uklonila greška. Kako bi se sistem proširio ovom funkcionalnošću, osim ogromnog broja konfiguracija i grešaka koje je potrebno skupiti i koje naš sistem trenutno ima u bazi podataka, neophodno je i dodatno obraditi te podatke u cilju njihovog pretvaranja u znanje koje sistem može da koristi. Dodatna nadogradnja sistema može uključivati tehnike mašinskog učenja. Moguće je obučiti neuronske mreže sa istorijskim podacima. Nakon obuke, ovakav sistem bi mogao u realnom vremenu da prepozna konfiguracije koje imaju određen problem i da definiše akcije koje se trebaju izvršiti za rešavanje ovog problema. Ovakav sistem bi u velikoj meri smanjio učinak čoveka prilikom nadgledanja mreže, ali ne u potpunosti, jer se uvek može očekivati neki novi i drugačiji scenario gde je potrebno imati stručnog mrežnog administratora. Glavna prednost ovakvog sistema bi bila u smanjenju broja grešaka koje zahtevaju reakciju mrežnog administratora, kao i smanjena mogućnost za izazivanje veće štete, jer se problemi rešavaju odmah u realnom vremenu. Sa ovom funkcionalnošću, sistem bi se mogao nazvati samo-oporavljajući mrežni sistem (engl. *self-healing networks*).

Osim ove funkcionalnosti, u istom smeru, sistem bi se mogao nadograditi i bezbednosnim slojem, odnosno da se osim tehničkih grešaka u konfiguraciji vrši i provera potencijalno sumnjivih aktivnosti koje odudaraju od uobičajenih konfiguracionih obrazaca.

Predloženi sistem za nadzor, upravljanje i analizu konfiguracija mrežnih uređaja zasnovan na NETCONF protokolu predstavlja alternativu postojećim rešenjima sa naglaskom na bezbednost, jednostavnost i pravovremenost otkrivanja grešaka u konfiguracijama mrežnih uređaja. Sistem iako ima prostor sa proširenje u smislu automatskog rešavanja problema koristeći neuronske mreže i mašinsko učenje, u ovom obliku predstavlja značajan alat za korišćenje jer pruža određene prednosti koje se mogu postići samo kombinacijom tehnologija koje su korišćene prilikom implementacije ovog sistema. Ove prednosti su centralizovano i uniformisano prikupljanje konfiguracionih podataka sa različitih mrežnih uređaja koji se nalaze na jednoj mreži korišćenjem NETCONF protokola, zatim automatsko skladištenje svih

dobijenih konfiguracija kao i određenih analiza i statistika nadgledanja, zatim verifikacija ovih konfiguracija, kao i intuitivni prikaz korisniku preko WEB UI-a i mobilne aplikacije.

7. ZAKLJUČAK

Nadzor mreže i uređaja u mreži danas predstavlja izuzetno važnu i gotovo neizostavnu stavku jer su mrežne infrastrukture sačinjene od velikog broja uređaja čije je ponašanje u mreži veoma dinamično. Kod ovakvih mreža se veoma često dešava nastanak određenih problema u vezi funkcionisanja nekog od uređaja, a koji mogu biti vrlo različite prirode. U ovom radu opisan je sistem koji vrši nadgledanje, analizu i upravljanje konfiguracijama mrežnih uređaja zasnovan na savremenom NETCONF protokolu.

Korišćenjem NETCONF protokola i pratećih tehnologija, implementirano je rešenje koje:

- Centralizovano prikuplja konfiguracione podatke iz heterogenih uređaja,
- Omogućava vizuelno poređenje trenutnog i očekivanog stanja konfiguracije,
- Smanjuje vreme reakcije administratora i rizik od ljudskih grešaka,
- Pruža osnovu za naprednu analizu kroz čuvanje istorijskih podataka.

Sistem je trenutno dostupan korisnicima putem WEB UI-a kao i preko mobilne aplikacije koja omogućava održavanje mreže „u hodu“. Korišćenjem NETCONF protokola omogućeno je dohvatanje konfiguracionih podataka sa različitih mrežnih uređaja, čime se smanjuje opterećenje mrežnih administratora u smislu povezivanja sa svakim uređajem. Korišćenjem NETCONF protokola obezbeđuje se doslednost i interoperabilnost sa uređajima različitih proizvođača što predstavlja veliki iskorak u poređenju sa protokolima koji su se ranije koristili u ove svrhe. Prikupljene konfiguracije sistem skladišti u bazu podataka i na ovaj način omogućava nadzor nad promenama i generisanje statističkih izveštaja.

Čuvanjem istorijskih konfiguracionih podataka u bazi podataka omogućava se uvid u ponašanje uređaja kroz vreme čime se omogućava dodatna kontrola nad promenama koje su se dešavale i događajima koji nisu bili očekivani. Suština sistema je u automatskoj verifikaciji konfiguracije uređaja u odnosu na očekivanu konfiguraciju. Ova funkcionalnost je implementirana u Python programskom jeziku, koristeći sve prednosti koje sa sobom nose NETCONF protokol, YANG modul, XML forma konfiguracije kao i XPath jezik. Korišćenjem sistema za razmenu poruka omogućena je nezavisnost komponenti u arhitekturi sistema i pouzdana asinhrona obrada podataka, koja je od ključnog značaja za sisteme koji vrše ovakav tip nadzora u realnom vremenu, naročito u kompleksnim mrežama sa velikim brojem uređaja.

Pored tehničkih prednosti koje sistem ima zbog izbora tehnologija koje su korišćene prilikom implementacije sistema, sistem je osmišljen na takav način da bude i skalabilan. Kontejnerizacijom i orkestracijom, koje su realizovane putem Docker-a i Kubernetes-a, omogućena je jednostavna instalacija, skaliranje i održavanje u različitim okruženjima.

U samom sistemu postoji prostor za proširenje i nadogradnju u smeru obučavanja neuronskih mreža za prepoznavanje grešaka koje su detektovane u zavisnosti od vrste greške i tipa uređaja, kao i automatsku reakciju sistema i otklanjanje greške ili obaveštavanja administratora o preporučenim koracima za popravku u realnom vremenu. I bez ovih proširenja

sistem predstavlja kvalitetan alat za nadzor, analizu i upravljanje mrežnim konfiguracijama jer omogućava:

- Brzu i pouzdanu detekciju grešaka,
- Efikasno upravljanje konfiguracijama,
- Intuitivan uvid u istoriju promena i statistiku grešaka,
- Pregled stanja svih uređaja koji se trenutno nalaze u mreži.

Doprinosi rada:

- Razvoj univerzalnog rešenja nezavisnog od proizvođača opreme, pogodnog za primenu u heterogenim mrežnim okruženjima,
- Integracija više tehnologija (NETCONF, YANG, XPath, Python, Flask, React, RabbitMQ, MongoDB) u jedinstven, funkcionalan sistem,
- Kreiranje arhitekture koja omogućava skalabilnost i proširivost ka naprednim funkcionalnostima poput prediktivne analitike i automatskog otklanjanja grešaka.

Pravci budućeg rada:

- Razvoj modula za automatsku korekciju grešaka korišćenjem tehnika mašinskog učenja,
- Uvođenje prediktivnih algoritama za detekciju potencijalnih problema pre nego što nastanu,
- Dodavanje naprednog bezbednosnog sloja za analizu konfiguracionih anomalija i prevenciju zlonamernih aktivnosti.

Razvijeni sistem predstavlja značajan doprinos modernizaciji nadzora mrežnih infrastruktura i otvara mogućnost daljeg razvoja ka potpuno autonomnim, samo-oporavljajućim mrežnim rešenjima, što je jedan od strateških ciljeva savremenog mrežnog inženjeringa.

LITERATURA

[1] Nwakeze, Osita. (2024). THE ROLE OF NETWORK MONITORING AND ANALYSIS IN ENSURING OPTIMAL NETWORK PERFORMANCE. 6. 3009-3025. 10.56726/IRJMETS59269.

[2] Amlou A., Abane A., Merzouki M., Ait Oucheggou L., Maasaoui Z., Battou A. Automated Network Programmability Using OpenConfig YANG Models and NETCONF Protocol, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA.

[3] Ghafir, J. Svoboda, and V. Prenosil, "Network monitoring approaches: an Overview," in Third International Conference on Advances in Computing, Communication and Information Technology (CCIT), pp. 118–123, May 2015. doi:10.15224/978-1-63248-061-3-72.

[4] L. D. Netak and A. W. Kiwelekar, “Efficient Network Management Using SNMP,” *Journal of Network and Systems Management*, vol. 14, no. 2, pp. 189–194, June 2006, doi:10.1007/s10922-006-9028-7.

[5] D. Kreutz, F. M. Ramos, P. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015. doi:10.1109/jproc.2014.2371999.

[6] IETF: RFC 6241 Network Configuration Protocol (NETCONF) (2011), RFC 6241 - Network Configuration Protocol (NETCONF) (ietf.org),

[7] M. Fedor, M. L. Schoffstall, J. R. Davin, and J. D. Case, "RFC 1157: Simple network management protocol (SNMP)," IETF Datatracker. https://datatracker.ietf.org/doc/html/rfc1157

[8] R. Presuhn, "RFC 3416: Version 2 of the protocol operations for the Simple Network Management Protocol (SNMP)," IETF Datatracker. https://datatracker.ietf.org/doc/html/rfc3416

[9] Concluded WG SNMP version 3 (snmpv3)," SNMP Version 3 (snmpv3). https://datatracker.ietf.org/wg/snmpv3/documents/

[10] M. Savic, "Bridging the SNMP GAP: Simple Network Monitoring the internet of things," Facta Universitatis - Series: Electronics and Energetics, vol. 29, no. 3, pp. 475–487, 2016. doi:10.2298/fuee1603475s.

[11] D. Mauro and K. Schmidt, Essential SNMP: Help for System and Network Administrators, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2005.

[12] M. Savić, *Poboljšanje performansi IP mreža upotrebom saobraćajno svesnog rutiranja*, doktorska disertacija, Elektrotehnički fakultet, Univerzitet u Banjoj Luci, Banja Luka, 2019.

[13] K. McCloghrie, D. Perkins, and J. Schoenwaelder, *Structure of Management Information Version 2 (SMIv2)*, RFC 2578, STD 58, Apr. 1999. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc2578

[14] Valenčić D., Mateljan V. 2019. Implementation of NETCONF Protocol, Proceedings of the 42nd International Convention, Croatian Society for Information and Communication Technology, Electronics and Microelectronics - MIPRO, Rijeka,

[15] IETF RFC 3535 Overview of the 2002 IAB Network Management Workshop (2002), RFC 3535 - Overview of the 2002 IAB Network Management Workshop (ietf.org),

[16] M. Björklund, "RFC 6020: Yang - A data modeling language for the Network Configuration Protocol (NETCONF)," IETF Datatracker. https://datatracker.ietf.org/doc/html/rfc6020 (accessed Dec. 8, 2024).

[17] S. Popic, B. Majstorovic, M. Vuleta, D. Saric, and B. M. Todorovic, "Efficient usage of resources in SDN by modifying Yang Modules in linux-based Embedded Systems," in 2019 27th Telecommunications Forum (TELFOR), pp. 1–4, Nov. 2019. doi:10.1109/telfor48224.2019.8971364.

[18] J. Schönwälder, M. Björklund, and P. Shafer, "Network Configuration Management using Netconf and Yang," IEEE Communications Magazine, vol. 48, no. 9, pp. 166–173, Sep. 2010. doi:10.1109/mcom.2010.5560601.

[19] B. Sun, X. Yuan, H. Kang, X. Huang, and Y. Guan, "Incremental validation of XML document based on simplified XML element sequence pattern," in *2010 Seventh Web Information Systems and Applications Conference*, pp. 110–114, Aug. 2010. doi:10.1109/wisa.2010.28.

[20] N. Wahid and E. Pardede, "XML semantic constraint validation for XML updates: A survey," in *2011 International Conference on Semantic Technology and Information Retrieval*, pp. 57–63, Jun. 2011. doi:10.1109/stair.2011.5995765.

[21] Popić, Srđan & Vuleta, Maksim & Cvjetkovic, Petar & Todorovic, Branislav. (2020). Secure Topology Detection in Software-Defined Networking with Network Configuration Protocol and Link Layer Discovery Protocol. 10.1109/INDEL50386.2020.9266137.

[22] World Wide Web Consortium (W3C). Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, October 2000. https://www.w3.org/TR/2000/REC-xml-20001006

[23] J. Case, R. Mundy, D. Partain, and B. Stewart, “An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks” RFC 3411, IETF, Dec. 2002. https://datatracker.ietf.org/doc/html/rfc3411

[24] M. Bjorklund, "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, IETF, Oktobar 2010. https://www.rfc-editor.org/rfc/rfc6020.html

[25] U. Blumenthal and B. Wijnen, “User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)”, RFC 3414, Internet Engineering Task Force, Dec. 2002. https://datatracker.ietf.org/doc/html/rfc3414

[26] J. Clark and S. DeRose, XML Path Language (XPath) Version 1.0, W3C Recommendation, 1999. https://www.renderx.com/~renderx/portal/Tests/xmlspec/xpath.pdf

Prilog 1.

Biografija autora

Lični podaci

Rođena 24.08.1997. u Požarevcu, Srbija.

Obrazovanje

Završila srednju Ekonomsku školu u Banjoj Luci.

Prvi ciklus studija upisala 2016. godine na Elektrotehničkom fakultetu Univerziteta u Banjoj Luci, studijski program Računarstvo i informatika, smer Softversko inženjerstvo. Diplomirala 2022. godine, sa prosečnom ocenom 7.83. Tema diplomskog rada: „Razvoj testnog okruženja za verifikaciju NETCONF operacija“.

Drugi ciklus studija upisala 2024. godine, na Elektrotehničkom fakultetu Univerziteta u Banjoj Luci, studijski program Računarstvo i informatika.

Radno iskustvo

Od 2022. godine zaposlena u preduzeću RT-RK u Banjoj Luci na radnom mestu “softverski inženjer“.

Прилог 2.

УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ
ПОДАЦИ О АУТОРУ ОДБРАЊЕНОГ МАСТЕР/МАГИСТАРСКОГ РАДА

Име и презиме аутора мастер/магистарског рада: **Бојана Мартиновић**

Датум, мјесто и држава рођења аутора: **24.08.1997, Пожаревац, Србија**

Назив завршеног факултета/Академије аутора и година дипломирања:
Електротехнички факултет Универзитета у Бањој Луци, 2022. године

Датум одбране завршног/дипломског рада аутора: **23.09.2022.**

Наслов завршног/дипломског рада аутора:
Развој тестног окружења за верификацију NETCONF операција

Академско звање које је аутор стекао одбраном завршног/дипломског рада:
дипломирани инжењер електротехнике (240 ECTS)

Академско звање које је аутор стекао одбраном мастер/магистарског рада:
мастер електротехнике - 300 ECTS – Рачунарство и информатика

Назив факултета/Академије на коме је мастер/магистарски рад одбрањен:
Електротехнички факултет Универзитета у Бањој Луци

Наслов мастер/магистарског рада и датум одбране:
Систем за надзор, анализу и управљање конфигурацијама мрежних уређаја заснован на NETCONF протоколу, 19. 9. 2025.

Научна област мастер/магистарског рада према CERIF шифрарнику**: T 120**

Имена ментора и чланова комисије за одбрану мастер/магистарског рада:
проф. др Милош Љубојевић, предсједник
доц. др Михајло Савић, ментор
доц. др Срђан Попић, члан

У Бањој Луци, дана 09.09.2025.

Декан

ИЗЈАВА О АУТОРСТВУ

Изјављујем да је
мастер/магистарски рад

Наслов рада: Систем за надзор, анализу и управљање конфигурацијама мрежних уређаја заснован на NETCONF протколу

Наслов рада на енглеском језику: A system for monitoring, analyzing, and managing network device configurations based on the NETCONF protocol

- ☑ резултат сопственог истраживачког рада,
- ☑ да мастер/магистарски рад, у цјелини или у дијеловима, није био предложен за добијање било које дипломе према студијским програмима других високошколских установа,
- ☑ да су резултати коректно наведени и
- ☑ да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

У Бањој Луци 09.09.2025.

Потпис кандидата

Мартиновић Бојана

Изјава 2

Изјава којом се овлашћује Електротехнички факултет/ Академија умјетности Универзитета у Бањој Луци да мастер/магистарски рад учини јавно доступним

Овлашћујем Електротехнички факултет/ Академију умјетности Универзитета у Бањој Луци да мој мастер/магистарски рад, под насловом

Систем за надзор, анализу и управљање конфигурацијама мрежних уређаја заснован на NETCONF протколу

који је моје ауторско дјело, учини јавно доступним.

Мастер/магистарски рад са свим прилозима предао/ла сам у електронском формату, погодном за трајно архивирање.

Мој мастер/магистарски рад, похрањен у дигитални репозиторијум Универзитета у Бањој Луци, могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице *(Creative Commons)*, за коју сам се одлучио/ла.

1. Ауторство
2. Ауторство - некомерцијално
3. Ауторство - некомерцијално - без прераде
4. Ауторство - некомерцијално - дијелити под истим условима
5. Ауторство - без прераде
6. Ауторство - дијелити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

У Бањој Луци 09.09.2025.

Потпис кандидата

Мартиновић Бојана

Изјава 3

Изјава о идентичности штампане и електронске верзије мастер/магистарског рада

Име и презиме аутора: Бојана Мартиновић

Наслов рада: Систем за надзор, анализу и управљање конфигурацијама мрежних уређаја заснован на NETCONF протоколу

Ментор: доц. др Михајло Савић

Изјављујем да је штампана верзија мог мастер/магистарског рада идентична електронској верзији коју сам предао/ла за дигитални репозиторијум Универзитета у Бањој Луци.

У Бањој Луци 09.09.2025.

Потпис кандидата

Мартиновић Бојана

УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
Патре 5
78000 Бања Лука

Проф. др Милош Љубојевић
Електротехнички факултет Универзитета у Бањој Луци

Доц. др Михајло Савић
Електротехнички факултет Универзитета у Бањој Луци

Доц. др Срђан Попић
Електротехнички факултет Универзитета у Бањој Луци

НАУЧНО-НАСТАВНОМ ВИЈЕЋУ

ЕЛЕКТРОТЕХНИЧКОГ ФАКУЛТЕТА УНИВЕРЗИТЕТА У БАЊОЈ ЛУЦИ

Одлуком Научно-наставног вијећа Електротехничког факултета Универзитета у Бањој Луци број 20/3.296-8/25 од 19.06.2025. године, именовани смо за чланове Комисије за завршни рад II циклуса, под називом "**Систем за надзор, анализу и управљање конфигурацијама мрежних уређаја заснован на NETCONF протоколу**", кандидата Бојане Мартиновић. Након прегледа приложеног рада подносимо сљедећи

ИЗВЈЕШТАЈ

1. БИОГРАФСКИ ПОДАЦИ КАНДИДАТА

Бојана Мартиновић, дипл. инж. електротехнике, дипломирала је на Електротехничком факултету Универзитета у Бањој Луци 2022. године (оцјеном 10) са завршним радом I циклуса под називом "Развој тестног окружења за верификацију NETCONF операција" и просјечном оцјеном у току студија од 7,83.

Студије II циклуса студија је уписала 2024. године на Електротехничком факултету Универзитета у Бањој Луци и положила све испите просјечном оцјеном 10. Од 2022. године је запослена у предузећу РТ-РК у Бањој Луци на радном мјесту "софтверски инжењер".

Кандидаткиња је бјавила два рада на научним конференцијама међународног значаја:

- B. Martinovic, M. Vuleta, S. Popic and B. M. Todorovic, "Simple verification improvements of the NETCONF data model in programmable computer Networks," 2022 30th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2022, pp. 1-4, doi: 10.1109/TELFOR56187.2022.9983688.
- B. Martinović, M. Ljubojević and M. Savić, "Monitoring and Analysis of Network Devices' Configurations Based on the NETCONF Protocol," 2025 24th International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 2025, pp. 1-5, doi: 10.1109/INFOTEH64129.2025.10959286.

2. ОСНОВНИ ПОДАЦИ О РАДУ

Завршни рад II циклуса студија кандидата Бојане Мартиновић, под називом "**Систем за надзор, анализу и управљање конфигурацијама мрежних уређаја заснован на NETCONF протоколу**", садржи 69 нумерисаних страница са укупно 28 слика и 3 табеле, а организован је у седам глава:

1. Увод
2. Надзор мрежних уређаја
3. SNMP
4. NETCONF протокол
5. Систем за надзор, анализу и управљање конфигурацијама мрежних уређаја заснован на NETCONF протоколу
6. Дискусија
7. Закључак

Списак коришћене литературе садржи 26 цитираних извора.

3. АНАЛИЗА РАДА

У уводном дијелу рада прво су описани мотивација за израду рада и предмет истраживања, а затим су наведени циљеви, методологија и остварени резултати истраживања. Потом је укратко приказан садржај рада по главама и наведени су научни радови у којима су објављени резултати истраживања. Основни мотив за истраживање и израду рада представља недоступност рјешења на тржишту које обједињује и имплементира све неопходне функционалности система за надзор, анализу и управљање конфигурацијама мрежних уређаја заснован на NETCONF протоколу, а како су дефинисане у овом раду.

Друго поглавље обрађује појам надзора мреже, те анализира значај надзора рачунарских мрежа, од настанка првих рјешења до савремених приступа надзору. Дат је приказ општег приступа пројектовању и реализацији система за надзор рачунарских мрежа, те кратки приказ употребе NETCONF протокола у надзору рачунарских мрежа.

Треће поглавље даје детаљан опис историје, развоја и свих основних компонената SNMP протокола, са посебном пажњом посвећеном агентима и мрежним надзорним системима, као и основним токовима података. Дат је и преглед начина организације података, те су описани OID идентификатори, MIB модули, као и скаларне и табеларне врсте података.

Четврто поглавље је посвећено NETCONF протоколу. Поглавље детаљно обрађује разлог за настанак NETCONF протокола, као и историјат развоја, нарочито у свјетлу идентификованих изазова у употреби SNMP протокола у надзору савремених рачунарских мрежа. Дат је опис свих релевантних слојева протокола, као и језика YANG намијењеног за моделовање конфигурационих података. Дата је и анализа употребе XML-а и специфичности парсирање овог вида серијализације података. Описане су основне операције NETCONF протокола, од добављања, ажурирања и брисања конфигурације, до завршетка сесије, као и преглед могућности употребе протокола у надзору рачунарских мрежа. Поглавље завршава и кратким поређењем NETCONF и SNMP протокола.

Пето поглавље садржи опис система за надзор, анализу и управљање конфигурацијама мрежних уређаја заснованом на NETCONF протоколу. Дат је детаљан опис дизајна и реализације практичног дијела рада. Поглавље почиње представљањем захтјева и циљева система, наставља се описом приједлогом рјешења, затим је дат приказ архитектуре система, као и опис сваке од компоненти система. Анализирана је употреба имплементираног система

у експлоатацији, као и употреба развијене мобилне апликације која омогућава употребу и са мобилних уређаја.

Шесто поглавље је посвећено дискусији реализованог система, као и поређењу овог система и постојећих система доступних на тржишту. Поглавље завршава разматрањем могућих будућих проширења и надградњи имплементираног система за надзор рачунарских мрежа.

Рад завршава седмим поглављем које садржи закључни приказ резултата истраживања, као и завршна разматрања остварених циљева и основних доприноса рада, те приједлог могућих будућих праваца истраживања и развоја система.

4. НАЈВАЖНИЈИ ДОПРИНОСИ

Комисија сматра да је кандидат, кроз спроведено истраживање, реализовао завршни рад II циклуса студија који садржи више значајних доприноса, од којих су најважнији сљедећи:

1. Развијено је функционално рјешење за надзор и анализу конфигурација умрежених уређаја употребом NETCONF протокола.
2. Дефинисан је механизам за аутоматизовано детектовање разлика између очекиваног и тренутног стања конфигурације надзираних уређаја.
3. Имплементирана је база података која омогућава чување историјских података о сваком надзираном уређају.
4. Развијено рјешење омогућава приступ тренутном стању конфигурације сваког надзираног уређаја у реалном времену.
5. Реализовани систем омогућава визуелизацију забиљежених разлика између очекиване и тренутне конфигурације надзираних уређаја.

5. ЗАКЉУЧАК И ПРИЈЕДЛОГ

Комисија сматра да завршни рад II циклуса под називом "Систем за надзор, анализу и управљање конфигурацијама мрежних уређаја заснован на NETCONF протоколу", кандидата Бојане Мартиновић, садржи све потребне елементе и резултате којима су остварени постављени циљеви истраживања, те са задовољством предлаже Научно-наставном вијећу Електротехничког факултета Универзитета у Бањој Луци да усвоји извјештај Комисије и одобри заказивање усмене јавне одбране.

Бања Лука, 26.08.2025. године

Проф. др Милош Љубојевић, предсједник

Доц. др Михајло Савић, ментор

Доц. др Срђан Попић, члан