UPRAVLJANJE I ORKESTRACIJA MREŽNIH USLUGA POMOĆU NSO PLATFORME

Puljić, Leon

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Specijalistički diplomski stručni studij Elektrotehnike

LEON PULJIĆ

ZAVRŠNI RAD

UPRAVLJANJE I ORKESTRACIJA MREŽNIH USLUGA POMOĆU NSO PLATFORME

Split, rujan 2023.

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Specijalistički diplomski stručni studij Elektrotehnike

Predmet: Održavanje komunikacijskih sustava

ZAVRŠNI RAD

Kandidat: Leon Puljić

Naslov rada: Upravljanje i orkestracija mrežnih usluga pomoću NSO platforme

Mentor: Ivan Visković

Split, rujan 2023.

SADRŽAJ

Sažetak

Upravljanje i orkestracija mrežnih usluga pomoću NSO platforme

Završni rad istražuje upravljanje i orkestraciju mrežnih usluga pomoću NSO (*Network Service Orchestrator*) platforme. Fokus je na analizi koncepta upravljanja i održavanja mrežnih sustava, primjeni automatizacije za olakšanje tih procesa, te specifično na uporabi Cisco NSO platforme. Cilj rada je razumjeti na koji način NSO platforma omogućuje efikasno upravljanje i orkestraciju mrežnih usluga te demonstrirati uporabu NSO platforme kroz praktične primjere.

Istraživanje je pokazalo da NSO platforma igra značajnu ulogu u unaprjeđenju tih procesa. Primjenom automatizacije i orkestracije, mrežne usluge se mogu brže i preciznije implementirati i izmijeniti. Zaključno, ovaj rad naglašava važnost NSO platforme u kontekstu upravljanja kompleksnim komunikacijskim mrežama i poziva na daljnje istraživanje kako bi se iskoristili njezini potencijali u budućnosti.

Summary

Management and orchestration of network services using NSO platform

Thesis explores the management and orchestration of network services using the NSO (Network Service Orchestrator) platform. Focus is on the analysis of the concept of management and maintenance of network systems, the application of automation to facilitate these processes, and specifically on the use of the Cisco NSO platform. The aim of the thesis is to understand how the NSO platform enables efficient management and orchestration of network services and to demonstrate the application of NSO platform through practical examples.

The research showed that the NSO platform plays a significant role in improving these processes. By applying automation and orchestration, network services can be deployed and modified more quickly and accurately. In conclusion, this paper highlights the importance of the NSO platform in the context of management of complex communication networks and calls for further research to exploit its potential in the future.

1. UVOD

Komunikacijske mreže postaju sve složenije te tradicionalno upravljanje mrežama postaje sve manje održivo. Stoga postoji sve veća potreba za uvođenjem automatizacije, posebice u dijelu konfiguracije mreža. Automatizacijski alati omogućavaju učinkovitije upravljanje izvođenjem ponavljajućih zadataka bez ljudskih pogrešaka, uz visoku razinu dosljednosti odnosno mrežne pouzdanosti.

U prvom poglavlju rada, opisano je upravljanje i održavanje mrežnih sustava, uključujući osnove upravljanja mrežama, SNMP protokol i FCAPS model. Opisane su različite funkcije upravljanja - upravljanje pogreškama, konfiguracijom, obračunom, performansama i sigurnošću.

U poglavlju o automatizaciji procesa upravljanja i održavanja mreža analizirana je važnost automatizacije u kontekstu upravljanja mrežama. Opisan je proces upravljanja konfiguracijom mreža, s fokusom na NETCONF protokol i YANG model.

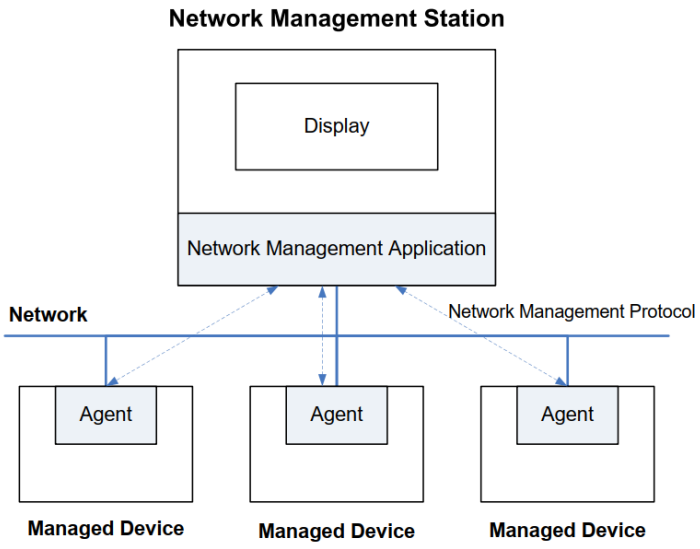
U trećem poglavlju, opisan je proces upravljanja kompleksnim mrežama koristeći Cisco NSO platformu, uključujući pregled osnovnih karakteristika te proširenih mogućnosti platforme.

U praktičnom dijelu rada prikazano je kreiranje L3 VPN usluge, modifikacija i brisanje koristeći Cisco NSO platformu. Također, obrađene su i napredne mogućnosti poput vraćanja konfiguracijskih promjena, te sinkronizacije konfiguracije između NSO platforme i uređaja u mreži.

2. UPRAVLJANJE I ODRŽAVANJE MREŽNIH SUSTAVA

Upravljanje mrežom predstavlja skup aplikacija, alata i procesa koji se koriste za pružanje, rad, održavanje, administriranje i sigurnost mrežne infrastrukture i mrežnih usluga. Uloga upravljanja mrežom je osigurati dostupnost elemenata mreže i mrežnih usluga u skladu sa zahtjevima korisnika mreže.

Kada su razvijeni IP protokoli, na kojima se temelje današnje mreže, malo se razmišljalo o samom upravljanju mrežom. Brzi razvoj prema većim i složenijim mrežama uzrokovao je značajno širenje tehnologija upravljanja mrežom. Početkom 1988. godine IAB (*Internet Architecture Board*) je odobrio SNMP (*Simple Network Management Protocol*) kao kratkoročno rješenje za upravljanje mrežom. Protokoli poput SNMP-a utrli su put standardiziranom upravljanju mrežom i razvoju inovativnih alata i aplikacija za upravljanje mrežom, te definirali NMS (*Network Management System*) kao skup aplikacija koje omogućuju nadzor i kontrolu mrežnih komponenti. Općenito, NMS sustavi imaju osnovnu arhitekturu kao što je prikazano na slici 2.1.



Slika 2.1. Tipična arhitektura sustava mrežnog upravljanja[1]

[1] https://www.egr.msu.edu/~renjian/pubs/network-management.pdf

Arhitektura se sastoji od dva ključna elementa: upravljačkog uređaja, koji se naziva upravljačka stanica ili upravitelj (eng. *manager*), i upravljanih uređaja, koji se nazivaju agenti za upravljanje ili jednostavno agenti. Upravljačka stanica služi kao sučelje između mrežnog operatera i sustava upravljanja mrežom. To je također platforma za upravljačke aplikacije za obavljanje upravljačkih funkcija kroz interakciju s upravljačkim agentima. Agent za upravljanje odgovara na zahtjeve upravljačke stanice. S obzirom na raznolikost upravljanih elemenata, kao što su usmjerivači, preklopnici, komunikacijska čvorišta, mrežni serveri, itd., te širok izbor operativnih sustava i programskih sučelja, protokol upravljanja je ključan za učinkovitu komunikaciju upravljačke stanice s agentima upravljanja.

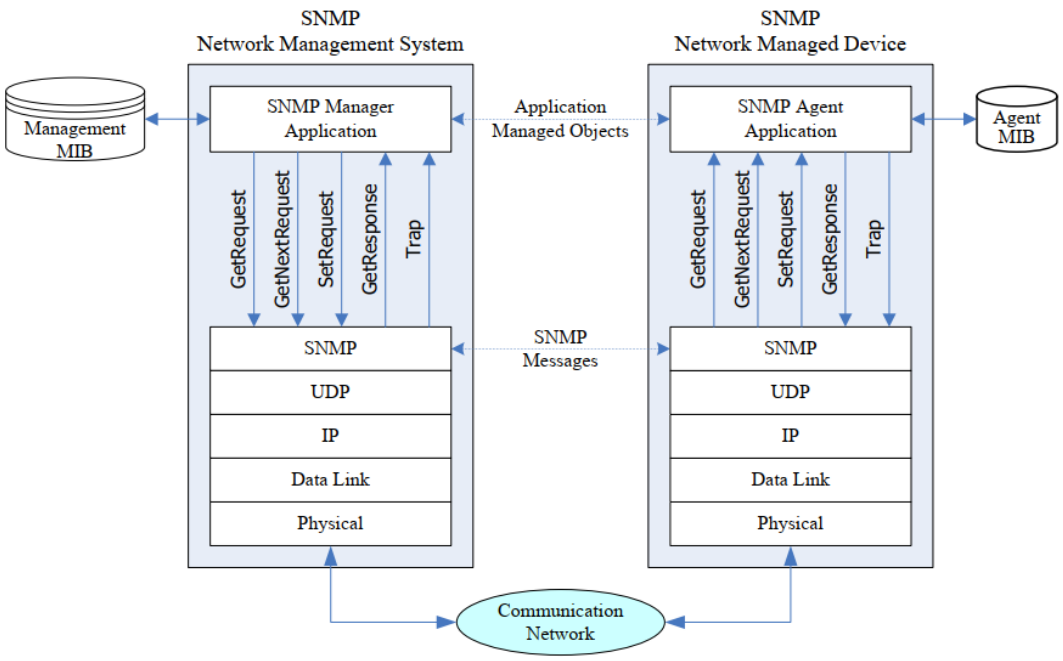
2.1. SNMP protokol

SNMP[2] protokol je najčešće korišteni protokol za upravljanje podatkovnom mrežom. Većina mrežnih komponenti koje se koriste u mrežnim sustavima imaju ugrađene mrežne agente koji mogu odgovoriti na SNMP sustav upravljanja mrežom. To omogućuje automatsko praćenje novih komponenti. Model upravljanja mrežom koji se koristi za upravljanje IP mrežom uključuje sljedeće ključne elemente:

- Stanica za upravljanje (skraćeno *manager*): sadrži aplikacije za upravljanje mrežom.
- Upravljački agent (skraćeno *agent*): pruža informacije sadržane u bazi podataka o upravljanju (MIB - *Management Information Base*) aplikacijama za upravljanje i prihvaća kontrolne informacije od upravljačke stanice.
- Baza podataka o upravljanju (MIB): definira podatke koje aplikacija za upravljanje može prikupljati i kontrolirati.
- Protokol upravljanja mrežom: definira protokol koji se koristi za komunikaciju između managera i agenata.

Arhitektura SNMP protokola prikazana na slici 2.1.1 sadrži ključne elemente okruženja za upravljanje mrežom. SNMP protokol je dizajniran kao jednostavan protokol aplikacijskog sloja temeljen na porukama. Upravljački proces ostvaruje upravljanje mrežom korištenjem SNMP protokola, koji je implementiran preko UDP (*User Datagram Protocol*) protokola. SNMP je protokol bez povezivanja, što znači da je svaka razmjena između upravljačke stanice i agenta zasebna transakcija.

[2] https://datatracker.ietf.org/doc/html/rfc3416



Slika 2.1.1. Arhitektura SNMP upravljačkog sustava[3]

Na slici 2.1.1 prikazano je i pet tipova paketa (PDU - *Packet Data Unit*) koje SNMP protokol podržava. Manager može poslati tri vrste PDU-ova: *GetRequest*, *GetNextRequest* i *SetRequest*. Sve tri poruke agent potvrđuje u obliku *GetResponse* poruke. Dodatna poruka koju agent generira je *Trap* poruka. Radi se o netraženoj poruci koja se generira kada se dogodi događaj koji utječe na normalne operacije MIB-a i temeljnih upravljanih resursa (npr. alarm).

2.2. FCAPS model

Osnovni cilj upravljanja mrežom je osigurati dostupnost mrežnih resursa za prijenos korisničkih podataka. ISO (*International Organization for Standardization*) klasifikacija (model ISO/IEC 7498-4[4]) grupira funkcije upravljanja u pet područja:

- upravljanje pogreškama
- upravljanje konfiguracijom
- upravljanje obračunom

[3] https://www.egr.msu.edu/~renjian/pubs/network-management.pdf
[4] https://cdn.standards.iteh.ai/samples/14258/356879966ac041b7bddc5b090a8467d9/ISO-IEC-7498-4-1989.pdf

- upravljanje performansama
- upravljanje sigurnošću

Ova klasifikacija je široko prihvaćena, te se još naziva FCAPS model (***F**ault **C**onfiguration **A**ccounting **P**erformance **S**ecurity*)[5]. Cilj uvođenja ovog modela je odmaknuti se od reaktivnog pristupa upravljanja mrežom prema proaktivnom pristupu - omogućiti funkcije upravljanja mrežom na način da se kontinuirano prati rad mreže, te da se na vrijeme identificiraju i otklanjaju pogreške, prije nego što one uzrokuju velike probleme u mreži.

ITU-T (*International Telecommunications Union – Telecommunications Sector*) je 1990-ih uveo pojam TMN (*Telecommunications Management Network*) kojim je definirana zasebna upravljačka mreža koja ima sučelja prema komunikacijskoj produkcijskoj mreži. U sklopu rada na TMN-u, ITU-T je dodatno poboljšao FCAPS model kao dio preporuke M.3400[6] o funkcijama upravljanja. TMN pruža okvir za postizanje međusobne povezanosti i komunikacije preko heterogenih operativnih sustava i komunikacijskih mreža - definira točke međusobnog povezivanja između dviju mreža i specificira funkcije upravljanja. Prema ITU-T M.3010[7], TMN ima 4 arhitekture:

- Informacijska arhitektura
- Fizička arhitektura
- Funkcionalna arhitektura
- Logička slojevita arhitektura

Unutar logičke slojevite arhitekture, koja je najvažniji i opće prihvaćeni dio TMN koncepta, TMN identificira četiri sloja upravljanja mrežom:

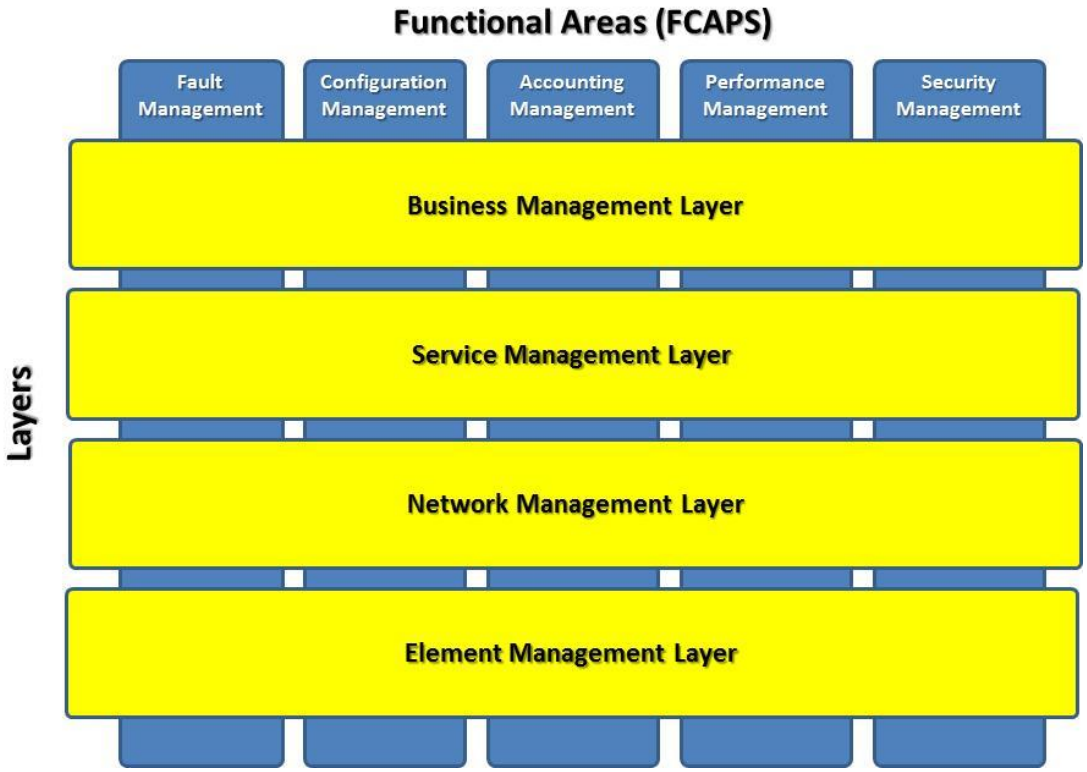
- Upravljanje poslovanjem (eng. *business management*)
    - Uključuje funkcije povezane s poslovnim aspektima, analiziranje trendova i pitanja kvalitete.
- Upravljanje uslugama (eng. *service management*)
    - Uključuje definiranje, administriranje i naplatu usluga.
- Upravljanje mrežom (eng. *network management*)

[5] https://www.techtarget.com/searchnetworking/definition/FCAPS
[6] https://www.itu.int/rec/T-REC-M.3400-200002-I
[7] https://www.itu.int/rec/T-REC-M.3100-200504-I

- Uključuje distribuiranje mrežnih resursa, konfiguraciju, kontrolu i nadzor mreže.

- Upravljanje elementima (eng. *element management*)
  - Podrazumijeva upravljanje pojedinačnim mrežnim elementima (NE - *Network Element*), što uključuje upravljanje alarmima, rukovanje informacijama, sigurnosno kopiranje, te održavanje hardvera i softvera.

FCAPS model se primjenjuje na sve slojeve logičke arhitekture TMN-a, kao što je prikazano na slici 2.2.1.



Slika 2.2.1. Primjena FCAPS funkcija u TMN slojevitoj logičkoj arhitekturi[8]

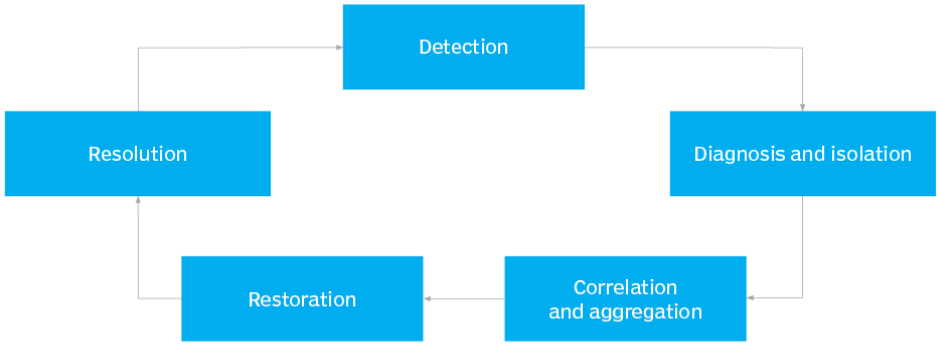
2.2.1. Upravljanje pogreškama

Upravljanje pogreškama uključuje otkrivanje, izolaciju, bilježenje i ispravljanje anomalija koje mogu uzrokovati kvar mreže (slika 2.2.1.1). Glavni cilj upravljanja pogreškama je osigurati da je mreža uvijek dostupna, a kada se kvar pojavi, da se može popraviti što je brže

[8] https://metanoia-inc.com/blog/2012/05/14/operator-metrics-that-matter-from-network-metrics-to-business-metrics/

moguće. Pogreške na mreži se stalno događaju zbog čega je ključno detektirati ih prije nego izazovu ozbiljne probleme.

Također, pregledom povijesnih podataka o pogreškama mrežni administratori mogu identificirati obrasce i trendove kako bi poboljšali proaktivne mjere koje značajno pomažu pri poboljšanju stabilnosti mreže. Na primjer, moguće je identificirati potencijalne buduće probleme i poduzeti korake da se spriječi njihovo pojavljivanje ili ponavljanje. Time se potencijalni zastoji u mreži smanjuju na minimum.



Slika 2.2.1.1. Tijek rada upravljanja pogreškama[9]

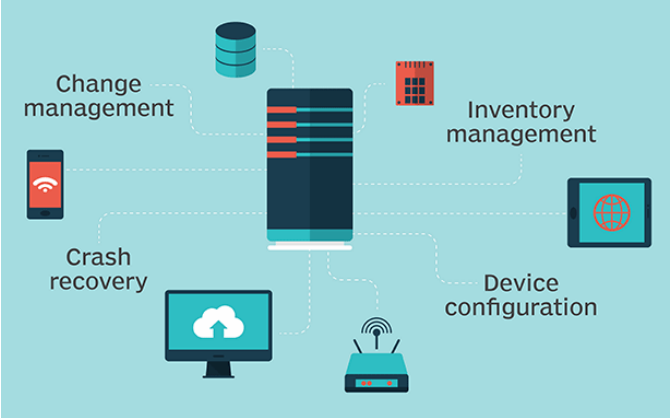
2.2.2. Upravljanje konfiguracijom

Upravljanje konfiguracijom omogućava inicijalizaciju mreže, osiguravanje mrežnih resursa i usluga, te nadzor i kontrolu mreže, što je prikazano na slici 2.2.2.1. Upravljanje konfiguracijom uključuju postavljanje i modifikaciju konfiguracije mrežne opreme (tzv. provizioniranje), funkcije planiranja mrežnih usluga i mreže, te ispitivanje postojećeg stanja konfiguracije mrežnih elemenata odnosno mreže. Na primjer, uključivanje određenog mrežnog elementa u mrežu zahtijeva definiranje odgovarajućih parametara, odnosno konfiguriranje samog uređaja. Također, tijekom rada mrežnog elementa potrebne su stalne modifikacije u konfiguraciji kao što su konfiguracije sučelja i konekcija, tablica usmjeravanja, QoS (*Quality of Service*) parametara i slično.

Upravljanje konfiguracijom podrazumijeva i upravljanje promjenama u mreži u pogledu hardvera i softvera mrežnih elemenata - dodavanje nove mrežne opreme i softvera,

[9] https://www.techtarget.com/searchnetworking/definition/FCAPS

modificiranje postojećih sustava i uklanjanje zastarjelih sustava i softvera. Također, vlasnici mreža mogu voditi inventar opreme i softvera, te ga redovito ažurirati.



Slika 2.2.2.1. Upravljanje konfiguracijom mreže[10]

2.2.3. Upravljanje obračunom

Upravljanje obračunom omogućuje mjerenje uporabe korištenih mrežnih resursa, što uključuje praćenje usluga koje se koriste, tko ih koristi, na koji način i kada. Usluge se prvenstveno obračunavaju s ciljem naplate. Upravljanje obračunom je potrebno da bi se usluge ispravno naplatile. U konačnici to rezultira izdavanjem računa korisnicima mreže za korištene usluge.

Ostvarene usluge mogu se obračunavati u ovisnosti o raznim parametrima kao što su duljina i vrsta veze, korištena usluga, zahtijevana kvaliteta usluge i slično. Ovisno o prirodi same usluge, primjenjuju se tri osnovna načina obračuna:

- obračun po vremenu (eng. *time based*) na temelju trajanja korištenja pojedine usluge
- obračun po količini (eng. *volume based*) na temelju količine iskorištenog prometa za pojedinu uslugu
- obračun po događaju (eng. *event based*) na temelju obavljenog događaja

Osim naplate, upravljanje obračunom se koristi i za analizu korištenja usluga u svrhu planiranja resursa i usluga – da li pojedinu uslugu dalje razvijati, ako da, u kojem smjeru

[10] https://www.techtarget.com/searchnetworking/definition/FCAPS

razvijati uslugu ili je ugasiti, i slično. Stoga se upravljanje obračunom provodi neovisno o tome da li se mreža koristi s ciljem naplate.

2.2.4. Upravljanje performansama

Upravljanje performansama omogućava procjenu i izvještavanje o ponašanju i učinkovitosti upravljanih mrežnih objekata. Sustav za nadzor mreže može mjeriti i prikazati status mreže, kao što je prikupljanje statističkih informacija o količini prometa, dostupnosti mreže, vremenu odziva i propusnosti, i slično.

Razina upravljanja performansama pomaže boljem upravljanju ukupnim performansama mreže. Cilj je omogućiti zahtijevanu propusnost mrežnih veza i sustava, izbjeći uska grla na mreži i identificirati potencijalne probleme. Glavni dio ovog procesa je odrediti koja poboljšanja daju ukupno najznačajnije poboljšanje performansi. Alati za upravljanje performansama omogućuju mrežnim administratorima praćenje performansi i rješavanje problema u stvarnom vremenu. Podaci o performansama mreže redovito se koriste za prepoznavanje obrazaca i trendova u svrhu predviđanja i planiranja mreže.

2.2.5. Upravljanje sigurnošću

Upravljanje sigurnošću omogućava zaštitu mreže i sustava od neovlaštenog pristupa i sigurnosnih napada. Mehanizmi za upravljanje sigurnošću uključuju autentifikaciju, enkripciju i autorizaciju. Upravljanje sigurnošću također se bavi generiranjem, distribucijom i pohranjivanjem ključeva za šifriranje, kao i drugih informacija povezanih sa sigurnošću. Upravljanje sigurnošću može uključivati sigurnosne sustave poput vatrozida (eng. *firewall*) i sustava za otkrivanje (IDS – *Intrusion Detection System*) i prevenciju (IPS – *Intrusion Prevention System*) napada koji omogućuju praćenje događaja u stvarnom vremenu i zapise događaja.

Poželjno je koristiti više slojeva zaštite da bi se bolje zaštitilo mrežu, uključujući i rješenja fizičke zaštite mrežne opreme. Ovaj pristup pomaže u održavanju povjerljivosti korisničkih podataka gdje je to potrebno ili opravdano. Sigurnosni sustavi omogućuju mrežnim administratorima kontrolu ovlasti koje svaki pojedinačni ovlašteni korisnik ima unutar sustava.

3. AUTOMATIZACIJA PROCESA UPRAVLJANJA I ODRŽAVANJA MREŽA

Automatizacija mreže podrazumijeva uvođenje tehnologija i alata koji omogućuju automatizirano upravljanje, nadzor i održavanje mrežnih uređaja i usluga. Cilj je pojednostaviti i ubrzati administrativne zadatke, povećati efikasnost, smanjiti ljudske greške i omogućiti mrežnim inženjerima da se usmjere na složenije aspekte mrežne arhitekture, strateške inicijative i inovacije. Primjerice, umjesto ručnog napora za ažuriranje stotina ili tisuća konfiguracija mrežnih uređaja, automatizacijski alat omogućava jednostavnu i učinkovitu implementaciju konfiguracijskih promjena te kontinuirano izvještavanje o statusu konfiguracije. Drugi primjer je automatizacija testiranja. Automatizirani testovi se izvode brže i efikasnije od ručnih testova, dosljedni su i manje podložni ljudskim pogreškama, što povećava pouzdanost rezultata.

Automatizacija i orkestracija mrežnih usluga postaje ključna u ICT industriji kako bi odgovorili na zahtjeve korisnika za robusnim mrežama s velikim brzinama i propusnošću, te pouzdanim vezama. Ispunjenje takvih zahtjeva dolazi po cijenu povećane operativne složenosti i varijabilnosti mrežnog kapaciteta. Upravljanje mrežom na tradicionalan način, primjerice ručnim unošenjem mrežne konfiguracije na veliki broj uređaja, postaje presporo uz veliku mogućnost ljudske pogreške.

Nedostaci kod upravljanja mrežom bez automatizacije su sljedeći:

- Rad sa heterogenim mrežama s nizom mrežnih elemenata kojima upravljaju različiti sustavi
- Previše ručnih procesa i prilagođenih skripti
- Operativni procesi oko softverskih sustava razvijaju se spajanjem razlomljenih sustava, što dovodi do loše automatizacije i visokog srednjeg vremena za rješavanje grešaka
- Operativno osoblje koje koristi softverske sustave za obavljanje svojih dnevnih zadataka potencijalno nema potrebne vještine za poboljšanje ili promjenu softvera

Prednosti upravljanja mrežom kada se automatizacija koristi su:

- Smanjenje pogrešaka uzrokovanih ljudskim faktorom (tzv. „ručnih“ pogrešaka)

- Zamjena specifičnih procesa za uređaje različitih proizvođača jedinstvenim procesom za sve uređaje (neovisno o dobavljaču opreme)
- Brzo pokretanje novih usluga
- Brže provizioniranje postojećih usluga
- Omogućavanje klijentima veće kontrole nad njihovim uslugama

3.1. Automatizacija upravljanja konfiguracijom mreža

Automatizacija i orkestracija mrežnih usluga donosi veliku prednost u operiranju mreža upravo pri upravljanju mrežnom konfiguracijom. Ukoliko procesi nisu automatizirani, stručnjaci za određenu uslugu koji konfiguriraju mrežne uređaje na više lokacija sve moraju raditi ručno. Također, ručno se provjeravaju te ponovno validiraju E2E (eng. *End-To-End*) konfiguracije. Na ovaj način ulaže se ogroman trud, a ipak veliki postotak problema u mreži bude uzrokovan netočnom inicijalnom konfiguracijom ili greškama prilikom vraćanja na izvornu konfiguraciju.

Nasuprot tome, automatizacija i orkestracija oslanjaju se na programsku konfiguraciju vođenu podatkovnim modelom svih elemenata koji sudjeluju se nalaze u servisnom lancu za pojedinu uslugu. Ta se konfiguracija proteže od parametara na razini mrežne usluge preko automatizirane konfiguracije svakog mrežnog uređaja uključenog u uslugu. Automatizirana provjera valjanosti konfiguracije proteže se kroz lance usluga, od opreme u prostorijama korisnika sve do funkcija virtualne mreže u podatkovnom centru.

Prvi korak u upravljanju mrežom je prikupljanje informacija sa same mreže kojom se upravlja. Informacije dolaze s raznih mrežnih uređaja, kao što su usmjerivači, preklopnici, serveri, i slično. Postoji niz protokola koji se koriste za upravljanje mrežom:

- SNMP protokol se obično koristi na mrežnim uređajima kao desetljećima star pristup za dobivanje mrežnih informacija.
- NETCONF protokol omogućava konfiguraciju mreže i sadrži mehanizme za konfiguriranje povezanih mrežnih uređaja.
- RESTCONF protokol se nadovezuje na NETCONF koristeći pristup temeljen na RESTful API sučelju za ažuriranje i promjenu konfiguracije mreže.

Automatizacija mreže jedan je od ključnih zahtjeva za mreže u dobu *cloud computinga*, međutim, ne može se postići konvencionalnim metodama upravljanja mrežom: naredbenim

sučeljem (CLI – *Command Line Interface*) i SNMP protokolom za upravljanje mrežom. Tu na scenu stupa NETCONF protokol koji uzima sve više maha kada govorimo o automatizaciji mreža.

Konfiguracija temeljena na CLI-ju je složena i uvelike se razlikuje ovisno o dobavljaču, tako da korisnici moraju naučiti i razviti skripte prilagodbe za CLI svakog pojedinog dobavljača. Osim toga, česte promjene CLI strukture i sintakse otežavaju održavanje tih istih skripti. Izlaz naredbi ne ovisi o strukturi, nepredvidiv je i sklon promjenama, što uzrokuje velike poteškoće u automatskom analiziranju izlaza CLI skripti.

SNMP protokol ne podržava mehanizam transakcija, što rezultira niskom učinkovitošću konfiguracije, te se stoga obično koristi samo u svrhu praćenja mreže. SNMP koristi UDP protokol, koji ne može osigurati pouzdan i uređen prijenos podataka i nema učinkovit sigurnosni mehanizam. Dodatno, ne postoji mehanizam za podnošenje konfiguracijskih transakcija. Stoga se konfiguracija provodi zasebno za svaki pojedini objekt, a ne na razini mrežne usluge. Kada je potrebno konfigurirati više objekata, to može dovesti do neželjenih utjecaja na mreži za slučaj da su neki objekti uspješno konfigurirani, a neki nisu uspješno konfigurirani.

Kako bi se prevladali nedostaci CLI-ja i SNMP protokola, uveden je NETCONF protokol temeljen na XML-u, koji ima sljedeće prednosti:

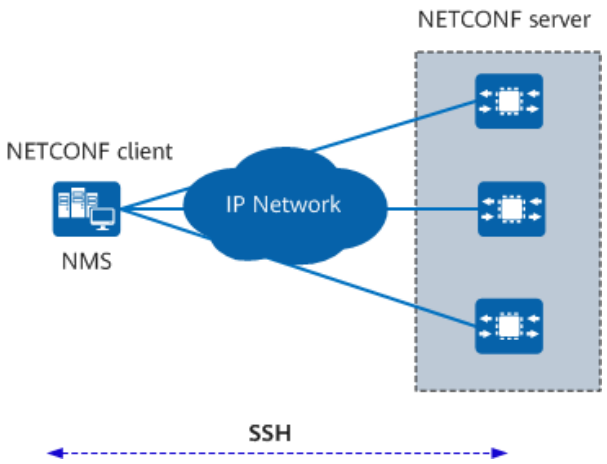
- Koristi hijerarhijski okvir protokola, što ga čini prikladnijim za zahtjeve mreža podignutih u cloud-u.
- Koristi XML kodiranje za formatiranje poruka te RPC mehanizam za izmjenu konfiguracijskih podataka. Ovo olakšava upravljanje konfiguracijskim podacima i interoperabilnost između mrežnih uređaja različitih dobavljača.
- Izvodi operacije na uređajima temeljene na YANG modelu, smanjujući mrežne greške uzrokovane pogreškama prilikom ručne konfiguracije.
- Pruža sigurnosne mehanizme poput autentifikacije i autorizacije radi osiguravanja sigurnosti prijenosa poruka.
- Osigurava transakcijski mehanizam za podršku klasifikaciji podataka, pohranjivanju i migraciji, podnošenju temeljenom na fazama, izolaciji konfiguracije, kao i općoj konfiguracijskoj isporuci, provjeri i povratu, minimizirajući utjecaj na mrežne usluge.

- Definira različita radna sučelja i podržava proširenje na temelju mogućnosti. To omogućuje dobavljačima definiranje vlastitih operacija protokola, kako bi implementirali jedinstvene funkcije upravljanja.

3.1.1. NETCONF protokol

NETCONF (*Network Configuration*)[11] protokol je protokol za upravljanje mrežom koji sustavu za upravljanje mrežom (NMS) omogućuje isporuku, izmjenu i brisanje konfiguracije mrežnih uređaja. Standardna API sučelja dostupna su na mrežnim uređajima kako bi NMS mogao upravljati uređajima pomoću NETCONF protokola. Protokol je razvijen i standardiziran od strane IETF (*Internet Engineering Task Force*) organizacije u prosincu 2006. kao RFC 4741[12], te je kasnije revidiran u lipnju 2011. i objavljen kao RFC 6241[13].

NETCONF koristi kodiranje podataka temeljeno na XML-u (*eXtensible Markup Language*) za podatke o konfiguraciji i protokolne poruke, te koristi RPC (*Remote Procedure Call*) mehanizam za implementaciju komunikacije između klijenta i poslužitelja. Klijent može biti skripta ili aplikacija koja se izvodi na NMS-u, dok je poslužitelj mrežni uređaj. Na slici 3.1.1.1 prikazana je osnovna mrežna arhitektura NETCONF protokola.



Slika 3.1.1.1. Osnovna NETCONF mrežna arhitektura[14]

[11] https://en.wikipedia.org/wiki/NETCONF
[12] https://datatracker.ietf.org/doc/html/rfc4741
[13] https://datatracker.ietf.org/doc/html/rfc6241
[14] https://info.support.huawei.com/info-finder/encyclopedia/en/NETCONF.html

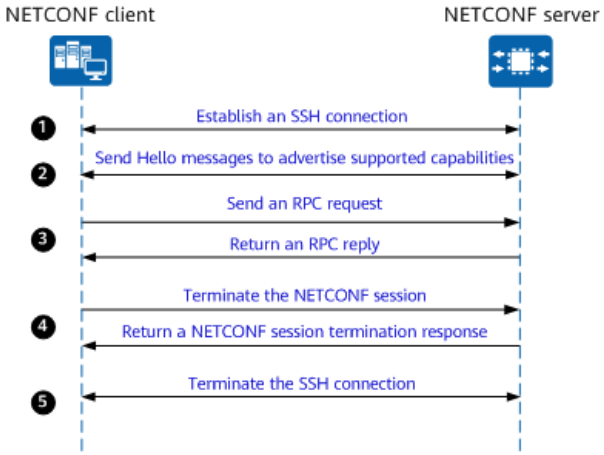
NETCONF arhitektura se sastoji od dvije komponente, klijenta i poslužitelja. **Klijent** podržava sljedeće funkcije:

- Upravlja mrežnim uređajima pomoću NETCONF protokola.
- Šalje RPC zahtjeve za upit ili izmjenu jedne ili više vrijednosti parametra prema NETCONF poslužitelju.
- Dohvaća status upravljanog uređaja na temelju alarma i događaja koje šalje NETCONF poslužitelj upravljanog uređaja.

Poslužitelj održava informacije o upravljanim uređajima i odgovara na zahtjeve koje šalje klijent.

- Po primitku zahtjeva od NETCONF klijenta, NETCONF poslužitelj analizira zahtjev te šalje odgovor klijentu.
- Ako se dogodi greška ili neka druga vrsta događaja na upravljanom uređaju, NETCONF poslužitelj prijavljuje alarm ili događaj klijentu putem mehanizma obavijesti. To omogućuje klijentu da sazna status upravljanog uređaja.

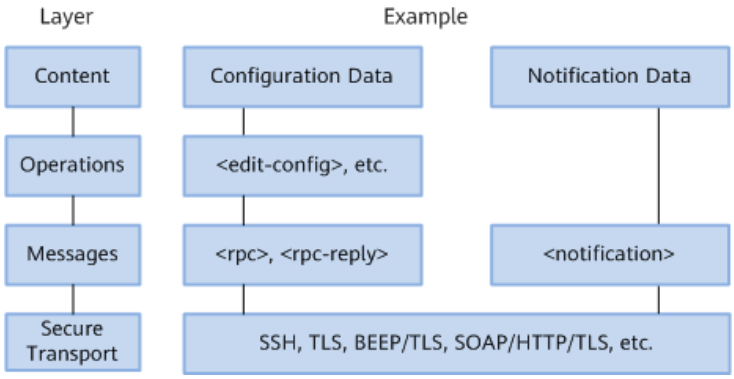
RPC mehanizam se koristi za komunikaciju između NETCONF klijenta i poslužitelja. Komunikacija je dopuštena tek nakon što se između klijenta i poslužitelja uspostavi sigurna sesija. Klijent šalje RPC zahtjev poslužitelju, a poslužitelj vraća odgovor klijentu nakon obrade zahtjeva. Proces uspostavljanja i prekida NETCONF sesije je prikazan na slici 3.1.1.2.



Slika 3.1.1.2. Proces uspostavljanja i prekida NETCONF sesije[15]

[15] https://info.support.huawei.com/info-finder/encyclopedia/en/NETCONF.html

NETCONF protokol koristi hijerarhijsku strukturu (slika 3.1.1.3). Svaki sloj sadrži određene funkcije i pruža usluge za gornji sloj. Ova hijerarhijska struktura omogućuje svakom sloju da se usredotoči samo na jedan aspekt NETCONF protokola i smanjuje ovisnosti između različitih slojeva. Na taj način, promjene unutarnje implementacije jednog sloja imaju minimalan utjecaj na druge slojeve.



Slika 3.1.1.3. Okvir NETCONF protokola[16]

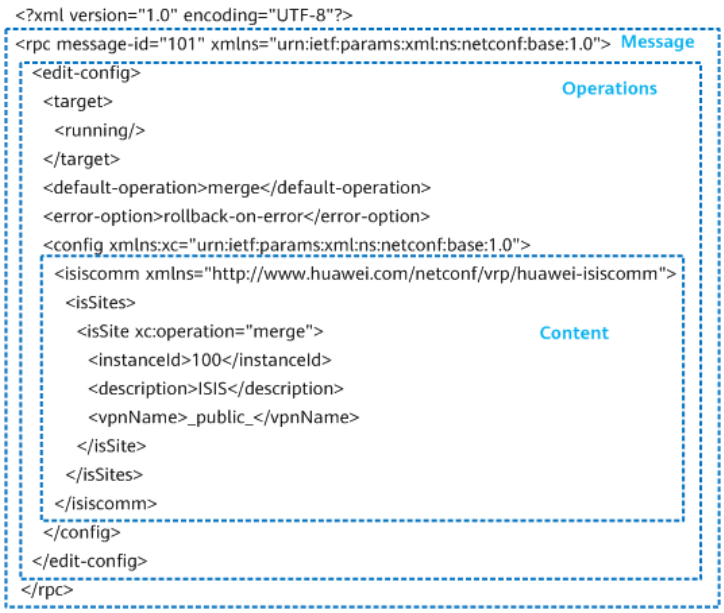
NETCONF protokol se konceptualno može podijeliti u četiri sloja:

- Sloj sigurnog transporta
  - Omogućava siguran komunikacijski put između klijenta i poslužitelja. Komunikacija NETCONF protokolom se može uspostaviti preko bilo kojeg transportnog protokola koji zadovoljava osnovne zahtjeve.
  - SSH (*Secure Shell*) je preferirani transportni protokol za prijenos XML informacija.
- Sloj poruka
  - Omogućava jednostavan mehanizam okvira za kodiranje RPC-ova i obavijesti neovisan o transportu.
  - Klijent enkapsulira RPC zahtjev u <*rpc*> element i šalje ga poslužitelju. Poslužitelj enkapsulira rezultat obrade ovog zahtjeva u element <*rpc-reply*> i šalje odgovor klijentu.

[16] https://info.support.huawei.com/info-finder/encyclopedia/en/NETCONF.html

- Sloj operacija
    - Definira skup operacija osnovnog protokola koje se pozivaju kao RPC metode s XML kodiranim parametrima.
- Sloj sadržaja
    - Podrazumijeva podatkovni model koji upravlja podacima. Glavni podatkovni modeli koji se koriste su Schema i YANG.
    - Schema je skup pravila definiranih za opisivanje XML datoteka. Uređaj koristi Schema datoteku (sličnu SNMP MIB datoteci) za pružanje konfiguracije uređaja i sučelja za upravljanje koje koristi NMS.
    - YANG je jezik za modeliranje podataka dizajniran specijalno za NETCONF protokol. Klijent može kompajlirati RPC operacije u XML poruke za implementaciju komunikacije između klijenta i poslužitelja u skladu s ograničenjima YANG modela.

Na slici 3.1.1.3 je prikazana struktura kompletne NETCONF YANG poruke na primjeru jednog zahtjeva. U NETCONF protokolu se koristi XML kodiranje što omogućuje izražavanje složenih hijerarhijskih podataka u tekstualnom formatu koji se može čitati, spremati i manipulirati s tradicionalnim alatima za obradu teksta i drugim XML specifičnim alatima.



Slika 3.1.1.4. Primjer strukture NETCONF YANG poruke[17]

[17] https://info.support.huawei.com/info-finder/encyclopedia/en/NETCONF.html

Mogućnosti NETCONF protokola uključuju standardne mogućnosti definirane od IETF-a za implementaciju osnovnih funkcija, kao i dodatne mogućnosti definirane od strane proizvođača za implementaciju proširenih funkcija. Uređaj može podržavati više operacija protokola dodavanjem proširenja, šireći time opseg operacija nad postojećim konfiguracijskim objektima.

Klijent i poslužitelj tijekom komunikacije mogu međusobno razmijeniti mogućnosti koje podržavaju. Kao rezultat toga, klijent šalje zahtjeve za operacijom samo unutar raspona mogućnosti koje podržava poslužitelj. Mogućnosti se razmjenjuju u porukama koje šalje svaki sudionik tijekom uspostavljanja sesije, takozvanim „Hello“ porukama koje sadrže <*hello*> element. Na taj način klijent i server mogu koristiti dogovorene mogućnosti za provedbu specifičnih funkcija upravljanja.

Rezultat pregovaranja standardnih mogućnosti (osim mogućnosti obavijesti) ovisi o mogućnostima koje podržava poslužitelj, dok rezultat pregovaranja o proširenim mogućnostima ovisi o tome koje mogućnosti podržavaju obje ravnopravne jedinice. NETCONF podržava sljedeći skup osnovnih operacija:

- <*get-config*>
    - Ispituje sve ili određene konfiguracijske podatke. Parametar <*source*> može se koristiti za određivanje konfiguracijske pohrane za koju će se postavljati upit.
- <*get*>
    - Traži konfiguracijske i statusne podatke samo iz <*running/*> pohrane konfiguracijskih podataka.
- <*edit-config*>
    - Učitava konfiguracijske podatke u specificiranu pohranu konfiguracijskih podataka *(*<*running/*> ili <*candidate/*>). Uređaj provodi autorizaciju za operaciju u <*edit-config*> i izvodi tražene izmjene samo ako je autorizacija uspješna.
- <*copy-config*>
    - Kopira konfiguracijske podatke iz jedne pohrane u drugu.
- <*delete-config*>

    - Briše pohranu konfiguracijskih podataka. <*running/*> pohrana konfiguracijskih podataka ne može se izbrisati.
- <*lock*>
    - Zaključava određenu pohranu konfiguracijskih podataka. Takva zaključavanja omogućuju klijentu isključivo dopuštenje za izmjene, čime se sprječavaju potencijalni sukobi (paralelne izmjene konfiguracije).
- <*unlock*>
    - Otključava pohranu konfiguracijsku podataka koja je prethodno zaključana operacijom <*lock*>. Klijentu nije dopušteno otključati pohranu konfiguracijskih podataka ukoliko je taj isti klijent nije zaključao.
- <*close-session*>
    - Zahtijeva precizno prekidanje NETCONF sesije.
- <*kill-session*>
    - Prisilno prekida NETCONF sesiju. Samo administrator može izvršiti ovu operaciju.

3.1.2. YANG model

Godine 2002. IAB je skrenuo pozornost na nedostatke SNMP protokola u upravljanju konfiguracijom, što je potaknulo razvoj NETCONF protokola. Iako je NETCONF protokol bio standardiziran, sadržaj podataka nije. Kao rezultat, razvijen je bolji jezik za modeliranje - YANG - koji model podataka čini jednostavnijim i lakšim za razumijevanje.

YANG (*Yet Another Next Generation*) je jezik za modeliranje podataka koji definira hijerarhijsku strukturu podataka koja se može koristiti za operacije temeljene na protokolima za upravljanje konfiguracijom mreže kao što je NETCONF/RESTCONF. Operacije uključuju konfiguraciju, statusne podatke, udaljene pozive procedura (RPC) i obavijesti.

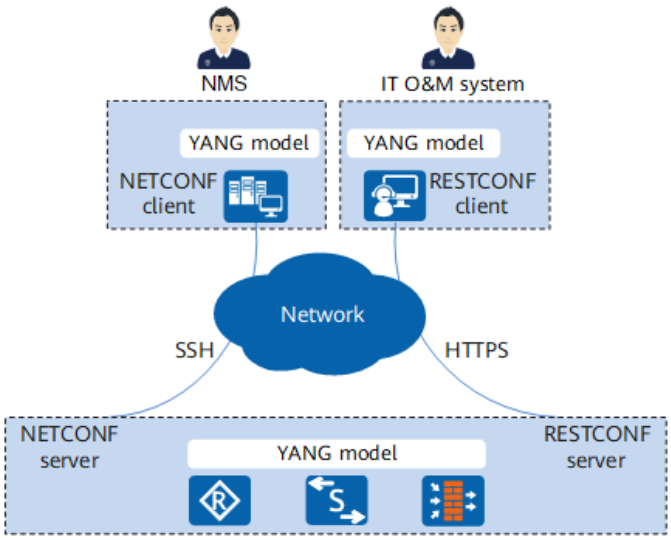
YANG jezik se koristi za generiranje YANG modela (koji se nazivaju i YANG datoteke) koji opisuju strukturu podataka, ograničenja integriteta podataka, kao i operacije, a definiran je u sljedećim RFC standardima:

- RFC 6020[18]: 2010. godine, IETF je prvi put definirao YANG kao jezik za modeliranje podataka za NETCONF protokol.

[18] https://datatracker.ietf.org/doc/html/rfc6020

- RFC 6021[19]: 2010. godine IETF je definirao različite tipove podataka koji se obično koriste u mrežnim komunikacijskim tehnologijama što omogućuje uvoz i korištenje unaprijed definiranih mrežnih tipova podataka bez njihovog redefiniranja prilikom izrade YANG modela.
- RFC 6991[20]: U 2013. godini IETF je dodao tipove podataka YANG modelu na temelju RFC 6021.
- RFC 7950[21]: IETF je 2016. izdao YANG1.1 verziju za ispravljanje dvosmislenosti i nedostataka u početnoj verziji (RFC 6020).

Kroz stalnu standardizaciju, YANG model postupno postaje dominantan model za opis podataka u ICT industriji. Kao što je prikazano na slici 3.1.2.1, YANG model je integriran u uređaje koji funkcioniraju kao poslužitelji. Mrežni administratori mogu koristiti NETCONF ili RESTCONF za središnje upravljanje, konfiguraciju i nadzor različitih mrežnih uređaja sposobnih za YANG, pojednostavljujući operaciju i održavanje mreža.



Slika 3.1.2.1. Arhitektura upravljanja mrežom temeljena na NETCONF/RESTCONF protokolu i YANG modelu[22]

[19] https://datatracker.ietf.org/doc/html/rfc6021
[20] https://datatracker.ietf.org/doc/html/rfc6991
[21] https://datatracker.ietf.org/doc/html/rfc7950
[22] https://info.support.huawei.com/info-finder/encyclopedia/en/YANG.html

U usporedbi s SNMP MIB modelom, YANG model je više hijerarhijski, može razlikovati konfiguracije i status, te pruža visoku proširivost. U nastavku je prikazan primjer dijela *if-mib* datoteke[23]. MIB je uređena tablica u kojoj su svi elementi *IfEntry*-a poredani jedan pored drugog, zbog čega je nemoguće razlikovati podatke o konfiguraciji i podatke o statusu.

```
  ifEntry OBJECT-TYPE
       SYNTAX      IfEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
               "An entry containing management information applicable
               to a particular interface."
       INDEX   { ifIndex }


       ::= { ifTable 1 }

   IfEntry ::=
       SEQUENCE {
           ifIndex                 InterfaceIndex,
           ifDescr                 DisplayString,
           ifType                  IANAifType,
           ifMtu                   Integer32,
           ifSpeed                 Gauge32,
           ifPhysAddress           PhysAddress,
           ifAdminStatus           INTEGER,
           ifOperStatus            INTEGER,
           ifLastChange            TimeTicks,
           ifInOctets              Counter32,
           ifInUcastPkts           Counter32,
           ifInNUcastPkts          Counter32,  -- deprecated
           ifInDiscards            Counter32,
           ifInErrors              Counter32,
           ifInUnknownProtos       Counter32,
           ifOutOctets             Counter32,
           ifOutUcastPkts          Counter32,
           ifOutNUcastPkts         Counter32,  -- deprecated
           ifOutDiscards           Counter32,
           ifOutErrors             Counter32,
           ifOutQLen               Gauge32,    -- deprecated
           ifSpecific              OBJECT IDENTIFIER -- deprecated
       }
```

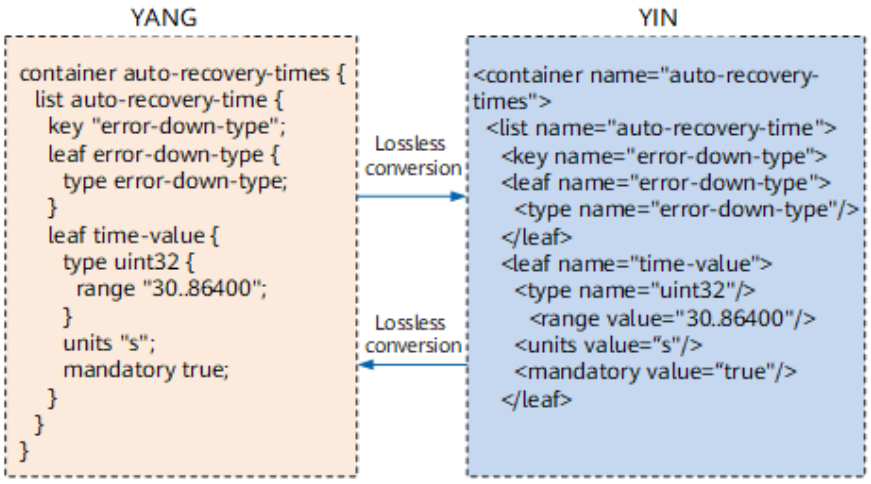
Primjer YANG modela[24] je prikazan ispod:

```
module huawei-ifm {
  namespace "urn:huawei:yang:huawei-ifm";
  prefix ifm;
  import huawei-pub-type {
    prefix pub-type;
```

[23] https://info.support.huawei.com/info-finder/encyclopedia/en/YANG.html
[24] https://info.support.huawei.com/info-finder/encyclopedia/en/YANG.html

```
}
organization
  "Huawei Technologies Co., Ltd.";
contact
  "Huawei Industrial Base
   Bantian, Longgang
   Shenzhen 518129
   People's Republic of China
   Website: http://www.huawei.com
   Email: support@huawei.com";
description
  "Common interface management, which includes the public
     configuration of interfaces.";
revision 2020-06-10 {
  description
    "Add units attribute.";
  reference
    "Huawei private.";
}
container auto-recovery-times {
  description
    "List of automatic recovery time configuration.";
  list auto-recovery-time {
    key "error-down-type";
    description
    "Configure automatic recovery time.";
    leaf error-down-type {
      type error-down-type;
      description
        "Cause of the error-down event.";
    leaf time-value {
      type uint32 {
        range "30..86400";
      }
      units "s";
      mandatory true;
      description
        "Delay for the status transition from down to up.";
    }
  }
}
```

Uređaji koriste YIN (*YANG Independent Notation*) datoteku za analizu YANG modela. YIN je YANG izražen u XML formatu. YIN i YANG koriste različite metode prikaza, ali sadrže ekvivalentne informacije (primjer konverzije je prikazan na slici 3.1.2.2). YIN se koristi za iskorištavanje postojećih alata kao što su XML parseri u raznim programskim jezicima. Ovi se alati mogu koristiti za filtriranje i provjeru podataka, automatsko generiranje koda i datoteka ili drugih zadataka, poboljšavajući učinkovitost parsiranja YANG modela.



Slika 3.1.2.2. Primjer konverzije između YANG-a i YIN-a[25]

[25] https://info.support.huawei.com/info-finder/encyclopedia/en/YANG.html

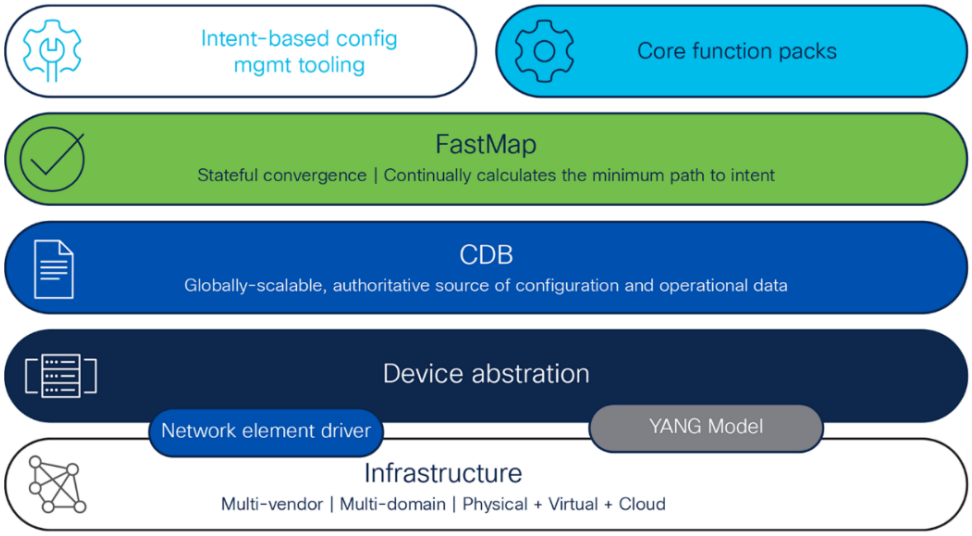
4. UPRAVLJANJE KOMPLEKSNIM MREŽAMA KORISTEĆI CISCO NSO

4.1. CISCO NSO platforma

Uporaba automatizacijskih alata općenito omogućava povećanje pouzdanosti mreže te učinkovitije upravljanje mrežnim i ljudskim resursima. Međutim, uvođenje automatizacije u mreže nije jednostavno. Zahtjevi za aplikacije i usluge su bezbrojni i stalno se mijenjaju, dok sama infrastruktura može obuhvaćati raznorazne tehnologije, dobavljače i generacije. Sposobnost premošćivanja izazova u ovakvom okruženju je ono u čemu se Cisco NSO ističe. Već nekoliko godina u produkciji u složenom mrežnom okruženju koje se suočava s ovim izazovima oblikovalo je NSO u vodeću platformu za automatizaciju u industriji.

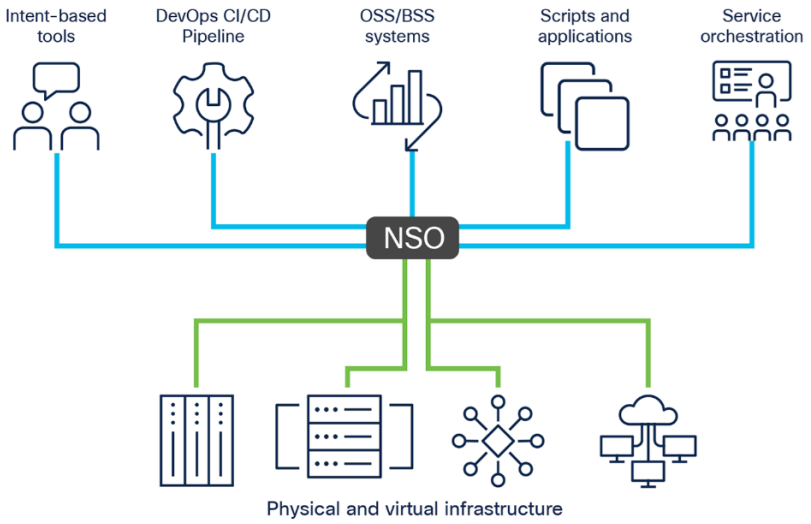
NSO platforma se sastoji od tri glavne komponente (slika 4.1.1):

- Programsko sučelje temeljeno na modelu koje omogućuje upravljanje mrežom, od jednostavnog uključivanja uređaja i upravljanja konfiguracijom, do sofisticiranog upravljanja životnim ciklusom mrežnih servisa.
- Brza, skalabilna, te visoko dostupna pohrana konfiguracijskih podataka (CDB - *Configuration Database*) koja je konačan izvor pouzdanih informacija o mreži.
- Sloj apstrakcije uređaja koji koristi upravljačke programe mrežnih elemenata (NED - *Network Element Driver*) za posredovanje u pristupu prema fizičkim i virtualnim uređajima od Cisco-a te više od 150 drugih proizvođača opreme.



Slika 4.1.1. Arhitektura Cisco NSO platforme[26]

Ove komponente omogućuju NSO platformi pružanje jedinstvenog mrežnog sučelja za sve mrežne uređaje i usluge – i fizičke i virtualne – koristeći model s jednim stanjem i konfiguracijsku bazu podataka (CDB). Kroz ovaj zajednički model, NSO djeluje kao most između aplikacija i vlasnika usluga, te same mrežne infrastrukture (slika 4.1.2).



Slika 4.1.2. NSO platforma kao most između mrežne infrastrukture i operatera[27]

NSO arhitektura pruža brojne funkcionalne prednosti pri razvoju okvira za automatizaciju:

- Bogat i raznolik skup *northbound* API-ja i softverskih sučelja, od programskih ili RPC-baziranih protokola kao što su NETCONF/RESTCONF, preko jezičnih veza kao što su Java i Python, do web sučelja (UI – *User Interface*) i CLI-ja.
- *NSO Developer Studio* pruža integrirano razvojno okruženje (IDE - *Integrated Development Environment*) koje omogućuje brz razvoj novih mrežnih usluga korištenjem svih mogućnosti koje NSO nudi. To omogućuje izravnu integraciju u postojeće poslovne sustave i lance operativnih alata kao što su *DevOps pipeline-i* za

[26] https://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/network-services-orchestrator/network-orchestrator-so.html
[27] https://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/network-services-orchestrator/network-orchestrator-so.html

kontinuiranu integraciju i implementaciju (CI/CD - *Continous Integration/Continous Delivery*).

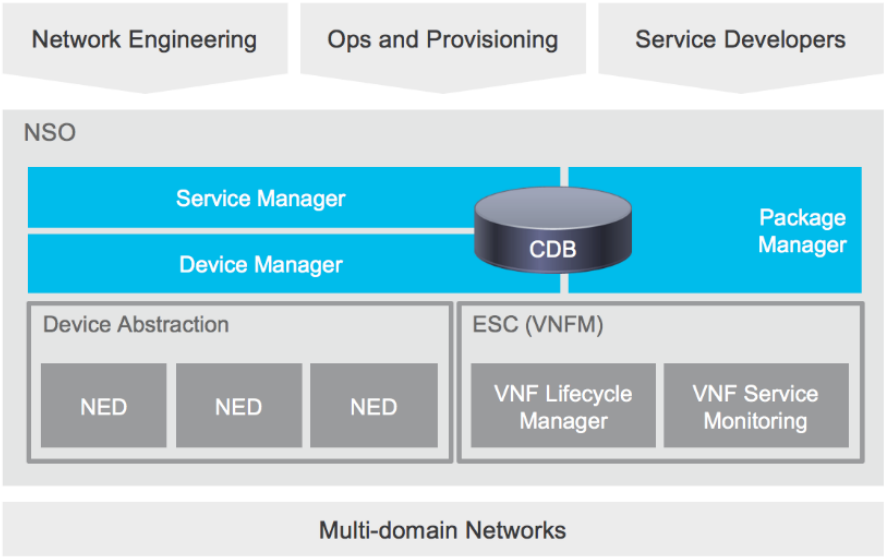
- Sloj apstrakcije uređaja raznih dobavljača za posredovanje u pristupu istima koji omogućuje automatizaciju tijeka rada više dobavljača i međudomena.
- Integrirane mogućnosti za održavanje cjelovitosti okruženja, sofisticirano rješavanje problema sa infrastrukturom i uslugama, te detaljnu reviziju i bilježenje.
- Proširivost NSO platforme s predugrađenim paketima funkcija, npr. za NFVI MANO i upravljanje kontejneriziranim mrežnim funkcijama u Kubernetes okruženju, ili kroz prilagođeni razvoj.

Korištenjem standardiziranog YANG jezika za modeliranje, Cisco NSO omogućava modeliranje i automatiziranje bilo koje vrste mrežnih uređaja, fizičkih ili virtualnih, adresiranih tradicionalno ili putem softverski definiranih mreža (SDN – *Software Defined Networking*). Osim uređaja, također je moguće modelirati bilo koju vrstu mrežne usluge ili pravila.

4.1.1. Upravljanje mrežnim servisima

NSO platforma podržava mreže više dobavljača kroz bogat izbor upravljačkih programa mrežnih elemenata (NED). Također, podržava proces provjere valjanosti, implementacije i apstrakcije konfiguracije mreže i mrežnih usluga.[28]

[28] https://developer.cisco.com/docs/nso/



Slika 4.1.1.1. Sistemske komponente NSO platforme[29]

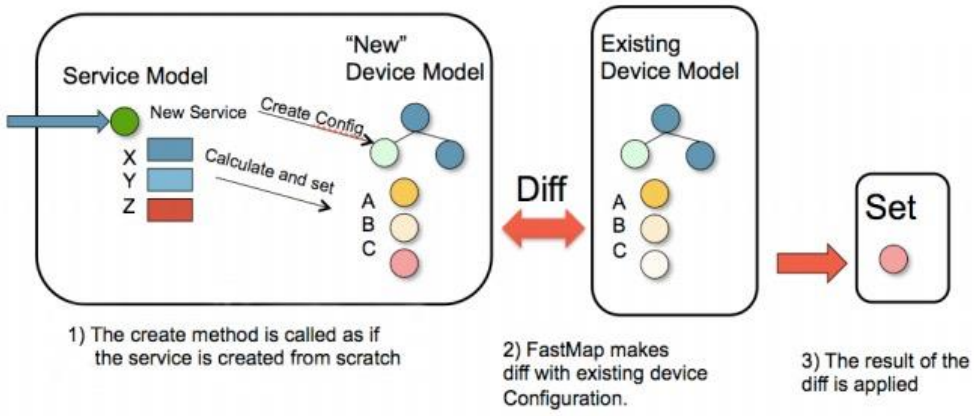
Kako je mrežna programabilnost počela dobivati na važnosti, shvatilo se da je za konfiguraciju mrežnih elemenata potrebno sučelje vođeno modernim modelom, što je dovelo do razvoja YANG jezika za modeliranje konfiguracije. NSO koristi YANG kao opći jezik za modeliranje za upravljanje uređajima i uslugama. YANG modeli opisuju sve NSO konfiguracije, uključujući konfiguraciju uređaja i konfiguraciju usluge. YANG je izvorno bio uparen s NETCONF protokolom, ali kako je REST postajao sve popularnije sučelje, standardiziran je i RESTCONF protokol. Oba protokola omogućuju definiranje API-ja pomoću YANG modela, kreirajući sučelje vođeno modelom; to znači da je osnovni rad protokola definiran standardima, a pojedinosti specifične za aplikaciju mogu se izvesti iz učitanih YANG modela.

U središtu NSO platforme nalazi se baza konfiguracijskih podataka (CDB) - stablasto strukturirana baza podataka kojom upravlja YANG shema. To znači da se sve informacije pohranjene unutar NSO platforme provjeravaju prema shemi. Svaka transakcija prema CDB-u ima ACID[30] svojstva, što između ostalog znači ili da se transakcija kao cjelina završi na svim uređajima koji sudjeluju i u glavnoj kopiji CDB-a, ili se alternativno cijela transakcija

[29] https://developer.cisco.com/docs/nso/#!nso-fundamentals/nso-fundamentals
[30] https://en.wikipedia.org/wiki/ACID

prekida i sve se promjene automatski vraćaju. CDB uvijek sadrži pogled NSO platforme na kompletnu konfiguraciju mreže.

Prilikom razvoja usluge potrebno je definirati mapiranje iz YANG modela usluge u odgovarajući YANG model uređaja. Radi se o deklarativnom preslikavanju, što znači da dok se mapira usluga nije potrebno brinuti o sintaksi CLI naredbi različitih uređaja ili redoslijedu kojim se te naredbe šalju uređajima. O svemu tome se brine NSO upravitelj uređaja. NSO smanjuje ovaj problem na jednu definiciju mapiranja podataka za scenarij "kreiranja". Nakon što je neka mrežna usluga pokrenuta, NSO će prikazati minimalnu razliku za svaku moguću promjenu usluge. Time upravlja FASTMAP algoritam koji pokriva cijeli životni ciklus usluge: kreiranje, mijenjanje i brisanje usluge. Rješenje zahtijeva minimalnu količinu koda za mapiranje s modela usluge na model uređaja. FASTMAP se temelji na generiranju promjena od početne izrade. Kada se stvori instanca usluge, rezultirajuća konfiguracija uređaja pohranjuje se zajedno s instancom usluge. Ako korisnik NSO platforme kasnije promijeni instancu usluge, NSO prvo primjenjuje (u transakciji) obrnuti *diff* usluge, učinkovito poništavajući prethodne rezultate koda za stvaranje usluge. Zatim pokreće logiku za ponovno stvaranje usluge i na kraju izvršava razliku prema trenutnoj konfiguraciji. Ova razlika se zatim šalje uređajima koji su dio usluge.



Slika 4.1.1.2. FASTMAP algoritam[31]

[31] https://developer.cisco.com/docs/nso/#!core-concepts/core-concepts

Upravitelj NSO uređaja je središte NSO platforme. Upravitelj uređaja održava listu svih upravljanih uređaja. NSO čuva glavnu kopiju konfiguracije za svaki upravljani uređaj u CDB-u. Kad god se izvrši promjena konfiguracije na popisu glavnih kopija konfiguracije uređaja, upravitelj uređaja će primijeniti ovu "promjenu konfiguracije mreže" na odgovarajuće promjene za stvarne upravljane uređaje. Upravitelj uređaja prosljeđuje potrebne promjene NED-ovima - upravljačkim programima mrežnih elemenata. NED treba biti instaliran za određenu vrstu operativnog sustava mrežnog uređaja - kao što su Cisco IOS, Cisco XR, Juniper JUNOS, itd. NED-ovi komuniciraju putem izvornog protokola uređaja i mogu pripadati sljedećim kategorijama:

- Uređaj koji podržava NETCONF protokol
    - Upravitelj uređaja će proizvesti NETCONF *edit-config* RPC operacije za svaki pojedini uređaj.
- SNMP uređaj
    - Upravitelj uređaja prevodi konfiguracijske promjene u odgovarajuće SNMP *SET* PDU-ove
- Uređaj s Cisco CLI-jem
    - Uređaj ima CLI s istom strukturom kao Cisco IOS ili XR usmjerivači. Upravitelj uređaja i CLI NED koriste se za izradu ispravnog slijeda CLI naredbi koje odražavaju konfiguracijske promjene.
- Drugi uređaji
    - Za uređaje koji se ne uklapaju ni u jednu od gore navedenih kategorija poziva se odgovarajući generički driver. Generički NED-ovi se koriste za vlasničke protokole kao što je REST, kao i za CLI-jeve koji ne podsjećaju na Cisco IOS ili XR. Upravitelj uređaja obavijestit će Generički NED o konfiguracijskim promjenama koji će ih prevesti u odgovarajuće operacije prema uređaju.

NSO može upravljati životnim ciklusom mrežnih usluga kao što su VPN-ovi, BGP peer-ovi, itd. Bitno je razumjeti što se podrazumijeva pod uslugom u ovom kontekstu:

- NSO izvlači detalje specifične za uređaj. Korisnik treba samo unijeti atribute relevantne za uslugu.
- Sama instanca usluge ima konfiguracijske podatke koji se mogu čitati i kojima se može manipulirati.
- Promjena konfiguracije instance usluge primjenjuje se na sve zahvaćene uređaje.

Ove značajke NSO koristi za podršku konfiguracije mrežnih usluga:

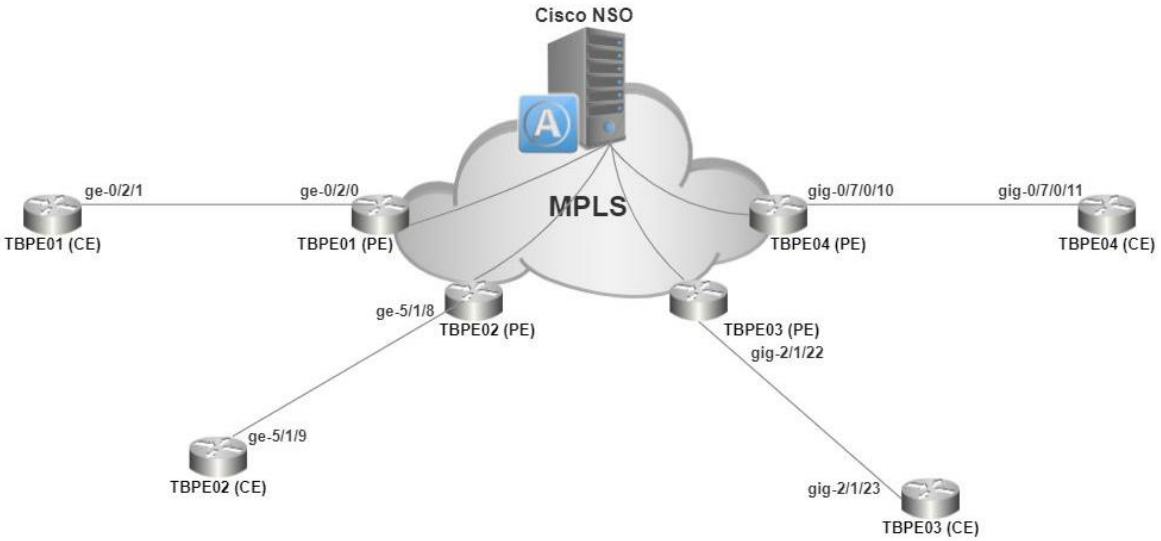
- Modeliranje usluge: mrežni inženjeri mogu modelirati atribute usluge i njihovo preslikavanje na konfiguracije uređaja. Na primjer, to znači da mrežni inženjer može specificirati podatkovni model za VPN-ove sa sučeljima usmjerivača, VLAN ID-om, VRF-om i diferencijatorom rute.
- Životni ciklus usluge: manje sofisticirani sustavi upravljanja konfiguracijom mogu stvoriti samo početnu instancu usluge u mreži, ali ne podržavaju promjenu ili brisanje instance usluge. NSO može u bilo kojem trenutku izmijeniti elemente usluge kao što je VLAN ID VPN-a, te generirati odgovarajuće promjene na mrežnim uređajima.
- Instanca NSO usluge ima konfiguracijske podatke koji se mogu prikazati i kojima se može manipulirati. *Run-time* servisni model ažurira sva NSO sučelja u stvarnom vremenu tako da mrežni inženjer može pregledavati i manipulirati instancom servisa preko CLI-ja, web UI-ja, REST-a itd.
- NSO održava reference između instanci usluge i konfiguracije uređaja. To znači da instanca usluge točno zna koje je konfiguracije uređaja stvorila/izmijenila. Svaka konfiguracija pohranjena u CDB mapira se na instancu usluge koja ju je kreirala.

4.2. Primjena NSO alata za upravljanje L3 VPN mrežnim servisom

Topologija ciljane L3 VPN (*Layer3 Virtual Private Network*) mreže prikazana je na slici 4.2.1. Mreža se sastoji od Juniper (**TBPE01** i **TBPE02**), Huawei (**TBPE03**) i Cisco usmjerivača (**TBPE04**). Konkretno u ovom primjeru, NSO komunicira sa usmjerivačima preko logički odvojene upravljačke mreže, koristeći specifični protokol i upravljački program (NED) koji se definiraju prilikom dodavanja svakog usmjerivača u NSO. Za komunikaciju sa usmjerivačima **TBPE01** i **TBPE02** NSO koristi NETCONF protokol i Juniper JUNOS upravljački program, za **TBPE03** koristi CLI preko SSH protokola i Huawei VRP upravljački program, a za **TBPE04** kao i za **TBPE03** koristi CLI preko SSH protokola, ali Cisco IOS upravljački program.

U korištenoj *lab* mreži ne postoje korisnički usmjerivači (CE - *Customer Edge*) već samo ulazni usmjerivači (PE - *Provider Edge*). Da bi simulirali pravu mrežu, jedno sučelje svakog od usmjerivača je iskonfigurirano kao CE usmjerivač, a drugo sučelje je iskonfigurirano kao PE usmjerivač (ulaz u provider mrežu, tzv. *endpoint*).

Potrebno je iskonfigurirati L3 VPN preko MPLS mreže koji sadrži navedene usmjerivače (PE) kao krajnje točke (eng. *endpoint*). Prethodno su upravljački programi za usmjerivače (NED-ovi) instalirani, usmjerivači iskonfigurirani i dodani na NSO, te su YANG modeli za L3 VPN unaprijed učitani u Cisco NSO.



Slika 4.2.1. Topologija mreže

Za komunikaciju prema Cisco NSO platformi koristi se NSO CLI – NBI (*NorthBound Interface*) sučelje NSO-a za reprezentaciju i upravljanje mrežnim uređajima i mrežnim uslugama. Postoje dva načina tj. moda rada kada se koristi NSO CLI: operativni i konfiguracijski. CLI se pokreće u operativnom modu rada. Prije kreiranja nove instance L3 VPN usluge - ili općenito prije bilo kakve izmjene konfiguracije u NSO-u - potrebno je biti u konfiguracijskom modu rada u koji se ulazi naredbom *config*. L3 VPN-a naziva **NSO_DEMO** sa **PE** usmjerivačima **TBPE01**, **TBPE02**, **TBPE03** i **TBPE04** kao krajnjim točkama je kreiran pomoću niza NSO CLI naredbi:

```
config
top

set vpn l3vpn NSO_DEMO endpoint TBPE01 interface ge-0/2/0 vlan 10 ip
address ip 10.10.10.1
set vpn l3vpn NSO_DEMO endpoint TBPE01 interface ge-0/2/0 vlan 10 ip
address mask /24
```

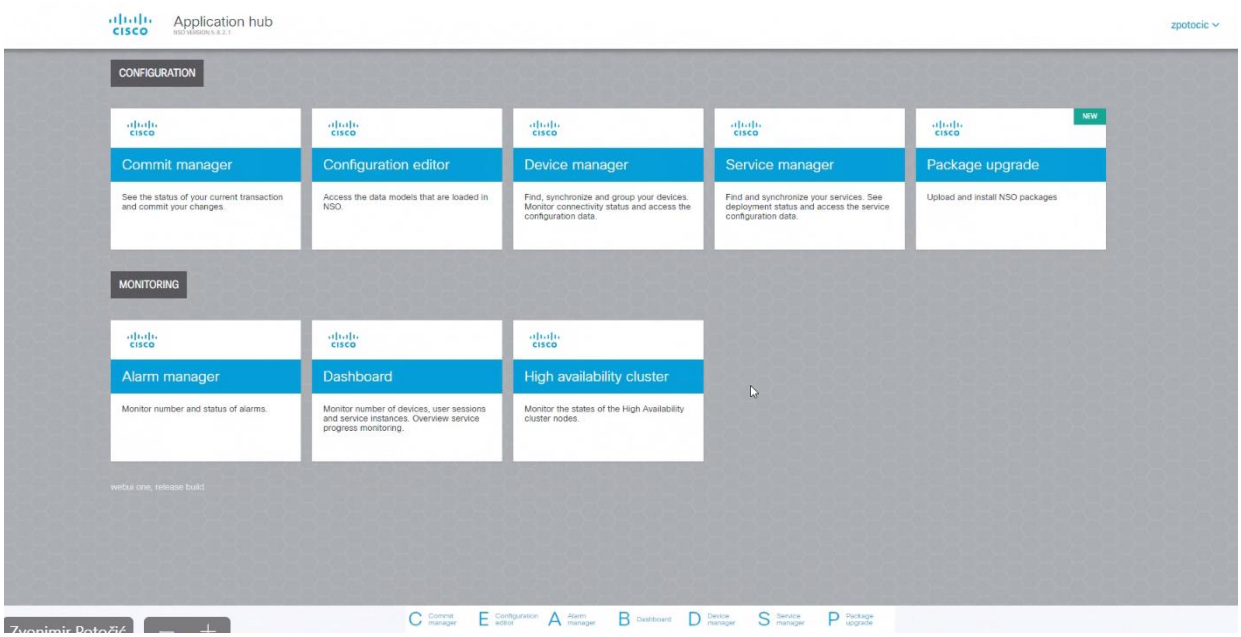
```
set vpn l3vpn NSO_DEMO endpoint TBPE02 interface ge-5/1/8 vlan 10 ip
address ip 10.10.20.1
set vpn l3vpn NSO_DEMO endpoint TBPE02 interface ge-5/1/8 vlan 10 ip
address mask /24

set vpn l3vpn NSO_DEMO endpoint TBPE03 interface
GigabitEthernet2/1/22 vlan 10 ip address ip 10.10.30.1
set vpn l3vpn NSO_DEMO endpoint TBPE03 interface
GigabitEthernet2/1/22 vlan 10 ip address mask /24

set vpn l3vpn NSO_DEMO endpoint TBPE04 interface
GigabitEthernet0/7/0/10 vlan 10 ip address ip 10.10.40.1
set vpn l3vpn NSO_DEMO endpoint TBPE04 interface
GigabitEthernet0/7/0/10 vlan 10 ip address mask /24
```

Na ovaj način je Cisco NSO korišten za manipuliranje apstrakcijama usluga povrh samih uređaja. Prije izvršavanja/spremanja promjena poželjno je pregledati rezultirajuću konfiguraciju izvršavanjem CLI naredbe *commit dry-run*.

```
zpotocic@ncs-lab# commit dry-run
cli {
    local-node {
        data  devices {
                  device TBPE01 {
                      config {
                          configuration {
                              interfaces {
                                  interface ge-0/2/0 {
            +                         unit 10 {
            +                             vlan-id 10;
...
                  device TBPE02 {
                      config {
                          configuration {
...
                  device TBPE03 {
                      config {
                          ip {
            +                 vpn-instance NSO_DEMO {
            +                     vpn-id 5391:19005;
...
                  device TBPE04 {
                      config {
                          vrf {
            +                 vrf-list NSO_DEMO {
...
             vpn {
            +    l3vpn NSO_DEMO {
            +        global-params {
            +            vpnid 19005;
            +            service-description NSO_DEMO;
            +        }
            +        endpoint TBPE01 {
...
```

```
               +              interface ge-0/2/0 {
               +                  vlan 10 {
               +                      ip {
               +                          address {
               +                              ip 10.10.10.1;
               +                              mask /24;
...
               +         endpoint TBPE02 {
...
               +         endpoint TBPE03 {
...
               +         endpoint TBPE04 {
...
zpotocic@ncs-lab# commit
```

Ove konfiguracijske operacije se isto tako mogu izvesti preko Cisco NSO Web UI-ja (slika 4.2.2) ili REST API-ja, međutim u praksi je uglavnom preferirana uporaba CLI sučelja.



Slika 4.2.2. Cisco NSO web UI

NSO primjenjuje usluge na mrežu tako da su pohranjene konfiguracije usluga zajedno s rezultirajućim promjenama konfiguracije uređaja. Ovo se koristi kao osnova za FASTMAP algoritam koji automatski izvodi promjene konfiguracije uređaja iz promjene usluge.

Na primjer, bilo koji dio kreirane L3 VPN **NSO_DEMO** usluge može se modificirati. Jednostavna promjena na usluzi poput dodavanja novog ili mijenjanja postojećeg *VLAN ID-ja* za neki usmjerivač rezultira mnogim promjenama u mreži. NSO te promjene radi automatski, kao što se može vidjeti u primjeru ispod.

```
zpotocic@ncs-lab# delete vpn l3vpn NSO_DEMO endpoint TBPE01 interface ge-0/2/0
vlan 10
zpotocic@ncs-lab# set vpn l3vpn NSO_DEMO endpoint TBPE01 interface ge-0/2/0 vlan
100 ip address ip 10.10.10.1 mask /24
zpotocic@ncs-lab# set vpn l3vpn NSO_DEMO endpoint TBPE01 interface ge-0/2/0 vlan
200 ip address ip 10.10.200.1 mask /24
zpotocic@ncs-lab# commit dry-run outformat cli
cli {
    local-node {
        data  devices {
                  device TBPE01 {
                      config {
                          configuration {
                              interfaces {
                                  interface ge-0/2/0 {
-                                     unit 10 {
-                                         vlan-id 10;
-                                         family {
-                                             inet {
-                                                 address 10.10.10.1/24;
-                                             }
-                                         }
-                                     }
+                                     unit 100 {
+                                         vlan-id 100;
+                                         family {
+                                             inet {
+                                                 address 10.10.10.1/24;
+                                             }
+                                         }
+                                     }
+                                     unit 200 {
+                                         vlan-id 200;
+                                         family {
+                                             inet {
+                                                 address 10.10.200.1/24;
+                                             }
+                                         }
+                                     }
                                  }
                              }
                              routing-instances {
                                  instance NSO_DEMO {
-                                     interface ge-0/2/0.10 {
-                                     }
+                                     interface ge-0/2/0.100 {
+                                     }
+                                     interface ge-0/2/0.200 {
+                                     }
                                  }
                              }
                          }
                      }
                  }
              }
              vpn {
                  l3vpn NSO_DEMO {
                      endpoint TBPE01 {
```

```
                        interface ge-0/2/0 {
-                           vlan 10 {
-                               ip {
-                                   address {
-                                       ip 10.10.10.1;
-                                       mask /24;
-                                   }
-                               }
-                           }
+                           vlan 100 {
+                               ip {
+                                   address {
+                                       ip 10.10.10.1;
+                                       mask /24;
+                                   }
+                               }
+                           }
+                           vlan 200 {
+                               ip {
+                                   address {
+                                       ip 10.10.200.1;
+                                       mask /24;
+                                   }
+                               }
+                           }
                        }
                    }
                }
            }
    }
}
zpotocic@ncs-lab# commit
Commit complete.
```

NSO prati sve promjene konfiguracije koje su korisnici napravili (CLI naredba *show commit list*) te omogućava jednostavno vraćanje određenih promjena (tzv. commit-ova) naredbom *rollback selective* <*commit_id*> - gdje <*commit_id*> predstavlja jedinstveni identifikator konfiguracijske promjene. Na primjer, slučajno je izbrisan usmjerivač **TBPE04** iz **NSO_DEMO** L3 VPN-a i cilj je vratiti se na stanje prije nego je brisanje napravljeno:

```
zpotocic@ncs-lab# delete vpn l3vpn NSO_DEMO endpoint TBPE04
zpotocic@ncs-lab# commit dry-run
cli {
    local-node {
        data  devices {
                  device TBPE04 {
                      config {
                          vrf {
-                             vrf-list NSO_DEMO {
-                                 description
NSO_DEMO::NSO_DEMO::VRF-L3VPN::VPN19005::NSO;
-                                 vpn {
-                                     id 5391:19005;
-                                 }
...
                vpn {
                    l3vpn NSO_DEMO {
```

```
-         endpoint TBPE04 {
-             vrf {
-                 rd 172.30.81.100:19005;
...
zpotocic@ncs-lab# commit
Commit complete.
zpotocic@ncs-lab# show vpn l3vpn NSO_DEMO endpoint ?
Description: Select endpoint device
Possible completions:
  TBPE01 - PE device name
  TBPE02 - PE device name
  TBPE03 - PE device name
zpotocic@ncs-lab# show commit list
2023-08-21 15:36:51
SNo.  ID       User       Client      Time Stamp          Label
Comment
~~~~  ~~       ~~~~       ~~~~~~      ~~~~~~~~~~          ~~~~~
~~~~~~~
15332 15332    zpotocic   cli         2023-08-21 15:33:05
15331 15331    zpotocic   cli         2023-08-21 15:31:34
...
15318 15318    admin      cli         2023-08-02 11:51:52
15317 15317    admin      cli         2023-08-02 11:50:16
...
zpotocic@ncs-lab# rollback selective 15332
zpotocic@ncs-lab# commit dry-run
cli {
    local-node {
        data  devices {
                  device TBPE04 {
                      config {
                          vrf {
+                             vrf-list NSO_DEMO {
...
              vpn {
                  l3vpn NSO_DEMO {
+                     endpoint TBPE04 {
+                         vrf {
+                             rd 172.30.81.100:19005;
...
zpotocic@ncs-lab# commit comment "Rollback vpn NSO_DEMO - add TBPE04"
zpotocic@ncs-lab# show vpn l3vpn NSO_DEMO endpoint ?
Description: Select endpoint device
Possible completions:
  TBPE01 - PE device name
  TBPE02 - PE device name
  TBPE03 - PE device name
  TBPE04 - PE device name
```

Na isti način na koji NSO može izračunati bilo koju promjenu konfiguracije usluge, također može automatski izbrisati konfiguracije uređaja koje su nastale stvaranjem usluge:

```
zpotocic@ncs-lab# no vpn l3vpn NSO_DEMO
zpotocic@ncs-lab# commit dry-run
cli {
```

```
    local-node {
        data  devices {
                  device TBPE01 {
                      config {
                          configuration {
                              interfaces {
                                  interface ge-0/2/0 {
            -                         unit 10 {
            -                             vlan-id 10;
...
             vpn {
            -    l3vpn NSO_DEMO {
            -        global-params {
            -            vpnid 19005;
            -            service-description NSO_DEMO;
            -        }
            -        endpoint TBPE01 {
...
zpotocic@ncs-lab# commit
```

4.3. Sinkronizacija konfiguracije između NSO platforme i uređaja u mreži

NSO platforma također omogućuje provjeru ažurnosti konfiguracija usluga i uređaja u mreži. Na primjer, ukoliko je netko ručno mijenjao konfiguraciju rubnog sučelja koje je prethodno konfigurirano koristeći NSO, konfiguracijska baza NSO-a i konfiguracija uređaja više neće biti usklađeni. Ukoliko se tada preko NSO-a pokuša ažurirati konfiguracija usluge koja uključuje uređaje čija konfiguracija nije usklađena sa konfiguracijom u NSO bazi, NSO će odbiti promjenu. Pomoću CLI naredbe *compare-config* provjeren je status konfiguracije mrežnih uređaja:

```
zpotocic@ncs-lab# show commit list
2023-08-21 15:36:51
SNo.  ID       User       Client      Time Stamp          Label
Comment
~~~~  ~~       ~~~~       ~~~~~~      ~~~~~~~~~~          ~~~~~
~~~~~~~
15332 15332    zpotocic   cli         2023-08-21 15:33:05
15331 15331    zpotocic   cli         2023-08-21 15:31:34
...
15321 15321    admin      cli         2023-08-02 11:56:37
15320 15320    admin      cli         2023-08-02 11:55:57
...
zpotocic@ncs-lab# rollback selective 15332
zpotocic@ncs-lab# commit dry-run
cli {
    local-node {
        data  devices {
                  device TBPE04 {
                      config {
                          vrf {
            +                 vrf-list NSO_DEMO {
...
```

```
              vpn {
                  l3vpn NSO_DEMO {
              +         endpoint TBPE04 {
              +             vrf {
              +                 rd 172.30.81.100:19005;
...
zpotocic@ncs-lab# commit comment "rollback VPN NSO_DEMO - add TBPE04"
Aborted: Network Element Driver: device TBPE04: out of sync
zpotocic@ncs-lab# *** ALARM out-of-sync: Device TBPE04 is out of sync
zpotocic@ncs-lab# request devices device TBPE04 compare-config
diff
 devices {
     device TBPE04 {
         config {
             interface {
                 GigabitEthernet-subinterface {
+                    GigabitEthernet 0/7/0/3.300 {
+                        encapsulation {
+                            dot1q {
+                                vlan-id 300;
+                            }
+                        }
+                    }
                 }
             }
         }
     }
 }
zpotocic@ncs-lab#
```

Iz izlaza naredbi iznad, jasno se može uočiti da usmjerivač **TBPE04** nije sinkroniziran sa konfiguracijom uređaja u NSO bazi.

Ukoliko je stvarna konfiguracija mrežnog uređaja smatrana mjerodavnom, CLI naredbom *sync-from* konfiguracija mrežnog uređaja se može primijeniti na NSO konfiguraciju. Nasuprot tome, ukoliko je NSO konfiguracija smatrana mjerodavnom, CLI naredbom *sync-to* NSO konfiguracija se može primijeniti na mrežni uređaj.

Sljedeći niz naredbi prikazuje kako primijeniti konfiguraciju uređaja kao mjerodavnu, ažurirati NSO bazu, i spremiti željene konfiguracijske promjene:

```
zpotocic@ncs-lab# request devices device TBPE04 sync-from
result true
zpotocic@ncs-lab# request devices device TBPE04 check-sync
result in-sync
zpotocic@ncs-lab# commit dry-run
cli {
    local-node {
        data  devices {
                  device TBPE04 {
                      config {
```

```
                            vrf {
              +                 vrf-list NSO_DEMO {
...
               vpn {
                   l3vpn NSO_DEMO {
              +        endpoint TBPE04 {
              +            vrf {
              +                rd 172.30.81.100:19005;
...
zpotocic@ncs-lab# commit comment "rollback VPN NSO_DEMO - add TBPE04"
Commit complete.
```

5. ZAKLJUČAK

Uvođenje automatizacijskih alata u procese upravljanja mrežama, posebice u dijelu upravljanja mrežnom konfiguracijom, omogućava povećanje pouzdanosti mreže te učinkovitije upravljanje mrežnim i ljudskim resursima. Međutim, uvođenje automatizacije u mreže nije jednostavno radi raznolikih zahtjeva na aplikacije te potrebe za stalnim promjenama.

NSO sustav za orkestraciju i upravljanje mrežnom konfiguracijom te mrežnim uslugama je jedinstvena platforma koja koristi standardizirana sučelja, protokole i modele u mrežama koje se, u principu, sastoje od opreme različitih proizvođača.

Analiza koja je provedena u radu, kao i demonstrirana praktična primjena NSO platforme na primjerima upravljanja L3 VPN mrežnim servisom, te sinkronizacije konfiguracije između same NSO platforme i uređaja u mreži, pokazuju mogućnosti brze, precizne i pouzdane implementacije promjena u mrežama.

Istraživanje otkriva ključnu važnost NSO platforme u kontekstu modernih komunikacijskih mreža. Njezina sposobnost automatizacije i orkestracije znatno unapređuje efikasnost, skalabilnost i pouzdanost upravljanja mrežama. NSO platforma pokazuje svoju snagu kao potencijalni katalizator za transformaciju upravljanja i orkestracije mrežnim uslugama, te ostavlja otvorena vrata za daljnje istraživanje i inovacije. Rad potvrđuje da je NSO platforma vrhunski alat koji omogućuje smanjenje složenosti upravljanja mrežnim uslugama i uređajima.

Buduća istraživanja bi se mogla usmjeriti na korištenje NSO platforme u složenijim mrežnim scenarijima. Primjerice, multidomensko okruženje gdje pored same komunikacijske mreže postoji mreža podatkovnih centara (eng. *Data Center Network*), mrežni elementi koji omogućuju mrežnu sigurnost (vatrozidi, IDS, IPS), te mrežna okolina u kojoj se nalaze zahtjevne mrežne funkcije kao što su DNS (*Domain Name System*), AAA (*Authentication, Authorization and Accounting*), DHCP (*Dynamic Host Configuration Protocol*), funkcije koje omogućuju specifične korisničke usluge i slično.

LITERATURA

[1] Ren. J., Li, T., *Network Management*, 2023, https://www.egr.msu.edu/~renjian/pubs/network-management.pdf [16.7.2023.]

[2] Presuhn, R. *RFC3416 - Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*, 2002, https://datatracker.ietf.org/doc/html/rfc3416 [11.8.2023.]

[3] Joint Technical Committee ISO/IEC JTC 1, Information technology, *Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 4: Management framework*, 1989, https://cdn.standards.iteh.ai/samples/14258/356879966ac041b7bddc5b090a8467d9/ISO-IEC-7498-4-1989.pdf [5.9.2023.]

[4] Zola, A., *What is FCAPS (Fault, Configuration, Accounting, Performance and Security)?*, 2021, https://www.techtarget.com/searchnetworking/definition/FCAPS [12.7.2023.]

[5] ITU, *M.3400 : TMN management functions*, 2000, https://www.itu.int/rec/T-REC-M.3400-200002-I [13.7.2023.]

[6] ITU, *M.3100 : Generic network information model*, 2005, https://www.itu.int/rec/T-REC-M.3100-200504-I [13.7.2023.]

[7] Smolka, J.R., *Operator Metrics that Matter: From Network Metrics to Business Metrics! | Metanoia, Inc. Blog*, 2023, https://metanoia-inc.com/blog/2012/05/14/operator-metrics-that-matter-from-network-metrics-to-business-metrics/ [26.8.2023.]

[8] *NETCONF – Wikipedia*, 2023, https://en.wikipedia.org/wiki/NETCONF [22.7.2023.]

[9] Enns, R., *RFC 4741 - NETCONF Configuration Protocol*, 2006, https://datatracker.ietf.org/doc/html/rfc4741 [22.7.2023.]

[10] Enns, R., Björklund, M., Bierman, A., Schönwälder, J., *RFC 6241 - Network Configuration Protocol (NETCONF)*, 2011, https://datatracker.ietf.org/doc/html/rfc6241 [22.7.2023.]

[11] Chunrong, Z., *What Is NETCONF? Why Do We Need It? – Huawei*, 2022, https://info.support.huawei.com/info-finder/encyclopedia/en/NETCONF.html [25.7.2023.]

[12] Björklund, M., *RFC 6020 - YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)*, 2010, https://datatracker.ietf.org/doc/html/rfc6020 [24.7.2023.]

[13] Schönwälder, J., *RFC 6021 - Common YANG Data Types*, 2010, https://datatracker.ietf.org/doc/html/rfc6021 [24.7.2023.]

[14] Schönwälder, J., *RFC 6991 - Common YANG Data Types*, 2013, https://datatracker.ietf.org/doc/html/rfc6991 [24.7.2023.]

[15] Björklund, M., *RFC 7950 - The YANG 1.1 Data Modeling Language*, 2016, https://datatracker.ietf.org/doc/html/rfc7950 [24.7.2023.]

[16] Huang, G., *What Is YANG? – Huawei*, 2021, https://info.support.huawei.com/info-finder/encyclopedia/en/YANG.html [26.7.2023.]

[17] Cisco Systems, Inc., *Cisco Crosswork Network Services Orchestrator Solution Overview – Cisco*, 2022, https://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/network-services-orchestrator/network-orchestrator-so.html [26.8.2023]

[18] Cisco Systems, Inc., *Cisco Network Services Orchestrator (NSO): The Bridge to Automation - Cisco*, 2023, https://www.cisco.com/c/en/us/solutions/service-provider/solutions-cloud-providers/network-services-orchestrator-solutions/bridge-to-automation.html [26.8.2023]

[19] Cisco Developer, *Network Services Orchestrator (NSO) Documentation and Downloads - Cisco DevNet*, 2023, https://developer.cisco.com/docs/nso/ [2.8.2023.]

[20] Cisco Developer, *Network Services Orchestrator (NSO) Documentation and Downloads - Cisco Developer*, 2023, https://developer.cisco.com/docs/nso/#!core-concepts/core-concepts [26.8.2023]

[21] *ACID – Wikipedia*, 2023, https://en.wikipedia.org/wiki/ACID [2.8.2023.]

POPIS SLIKA