Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.41
DOI: https://doi.org/10.24867/06BE22Klasnic

PROGRAMSKO REŠENJE ZA PRAĆENJE MREŽNOG SAOBRAĆAJA SCADA PROTOKOLA

SOFTWARE SOLUTION FOR MONITORING SCADA NETWORK TRAFFIC

Zvezdana Klašnić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U radu je prezentovano programsko rešenje za praćenje mrežnog saobraćaja SCADA protokola. Implementirano programsko rešenje može se koristiti za nadgledanje mrežnog saobraćaja, analizu poruka SCADA protokola, za formiranje i prikazivanje konfiguracije mreže u vidu stabla na osnovu obrađenih poruka, kao i za otkrivanje problema u mreži.*

Ključne reči: *SCADA, Real-time praćenje, Obrada SCADA poruka, Konfiguracija mreže.*

Abstract – *The document presents a software solution for monitoring SCADA network traffic. The implemented software solution can be used to monitor network traffic, analyze SCADA protocol messages, to form and display network configuration based on processed messages and to troubleshoot network problems.*

Keywords: *SCADA, Real-time monitoring, SCADA message processing, Network configuration.*

1. UVOD

Računarske mreže susreću se sa brojnim problemima, kao što su različiti napadi na mrežu i greške u konfiguraciji. Zadatak mrežnih administratora je da otkriju i otklone takve probleme kako bi održali mrežu bezbednom, pri čemu im može pomoći analiza paketa. Ona predstavlja proces snimanja i interpretacije podataka koji se razmenjuju kroz mrežu, kako bi se ustanovilo ima li neželjenih dešavanja u mreži. Postoji mnoštvo alata namenjenih za analizu mrežnog saobraćaja. Oni presreću pakete koji se prenose kroz mrežu, uzimaju sirove podatke iz paketa, pokušavaju da detektuju korišćeni protokol i analiziraju informacije koje se prenose. Dobar izbor alata može da unapredi mrežne performanse i da obezbedi sigurnost sistema.

Neki od često korišćenih alata su Wireshark [1], TCPdump [2], NetDecoder [3] i Capsa [4]. Zajedničko za sve ove alate je to što koriste dobro poznate i predefinisane portove za detektovanje korišćenog protokola, što predstavlja njihovu manu. Ne znaju sami da prepoznaju korišćeni aplikativni protokol, što je potrebno za parsiranje poruke, već je za svaki komunikacioni link potrebno ručno uneti kom protokolu pripada. Ova mana bi naročito došla do izražaja u slučaju da je potrebno pratiti hiljade adresa za više različitih protokola, jer bi tada bilo potrebno ručno uneti isto toliko komunikacionih linkova.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Srđan Vukmirović, vanr. prof.

Navedeni alati nisu eksplicitno namenjeni za praćenje mrežnog saobraćaja SCADA (*Supervisory Control And Data Acquisition*) protokola. Većina pokriva samo najpoznatije SCADA protokole (IEC 104, DNP3 i Modbus), dok bi sve ostale protokole bilo potrebno samostalno implementirati. Ukoliko za određenu mrežu nije poznato koje SCADA protokole koristi *remote unit*, bilo bi potrebno isprobavati protokole i detaljno ih analizirati, da bi se ustanovilo da li je odabran dobar protokol. Navedeni alati ne pokrivaju mogućnost uvida u konfiguraciju telemetrije u mreži.

Sa ciljem otklanjanja prethodno navedenih mana implementirano je programsko rešenje koje je tema ovog rada. Ovo rešenje može se koristiti za *real-time* praćenje mrežnog saobraćaja, analizu poruka SCADA protokola, otklanjanje problema u mreži, kao i za formiranje i prikazivanje konfiguracije mreže u vidu stabla na osnovu obrađenih poruka. Razlikuje se od prethodno navedenih alata po načinu prepoznavanja aplikativnog protokola i parsiranja uhvaćenog paketa. Zbog nepouzdanosti korišćenja predefinisanih portova za detekciju korišćenog protokola, implementirano rešenje u te svrhe koristi proveru preambule, parsiranje aplikativnog zaglavlja, validiranje pojedinih polja protokola, kao i *packet payload* informacija. Još jedna prednost je to što je na osnovu obrađenog mrežnog saobraćaja moguće dobiti hijerarhijski prikaz računarske mreže, gde su prikazani RTU-ovi sa svojim tačkama.

2. TEORIJSKE OSNOVE

SCADA paketi se kroz mrežu šalju preko TCP/IP (*Transmission Control Protocol/Internet Protocol*) infrastrukture. TCP/IP model u mrežnom sloju koristi *Internet Protocol* (IP). IP obezbeđuje fragmentaciju datagrama, ne garantuje prijem paketa, a provera greške je ograničena na proveru kontrolne sume zaglavlja.

Glavna funkcija transportnog sloja je da omogući komunikaciju između programskih aplikacija koje se nalaze na različitim računarima u mreži. TCP/IP model podržava dva transportna protokola:

- *Transmission Control Protocol* (TCP)
- *User Datagram Protocol* (UDP)

U radu je akcenat stavljen na TCP konekcije, jer je za prenos SCADA komunikacije bitno da podaci stignu kompletni i u tačnom redosledu.

Aplikacioni sloj direktno komunicira sa korisničkim aplikacijama i predstavlja njihov interfejs ka mreži. Svaka aplikacija na računaru identifikuje se pomoću porta. Za najčešće korišćene aplikacije uvedena je lista dobro

poznatih portova, za koje postoji dodela porta određenom aplikativnom protokolu. Korišćenjem dobro poznatih portova, detekcija protokola aplikacionog nivoa može biti neuspešna kada pojedini protokoli ne koriste te portove.

Na aplikacionom nivou postoji mnoštvo protokola, međutim biće pomenuti samo pojedini SCADA protokoli, koji su od značaja za ovaj rad. SCADA protokoli su industrijski aplikativni protokoli, pomoću kojih je omogućena komunikacija u SCADA sistemima. SCADA protokoli podržani implementiranim rešenjem su:

- Ferranti VAN-COMM [5]
- DNP3.0 [6]
- IEC 60870-5-101 [7]
- IEC 60870-5-104 [8]

3. IMPLEMENTIRANO PROGRAMSKO REŠENJE

Implementirano rešenje može se koristiti za *real-time* praćenje, analizu poruka SCADA protokola, za formiranje i prikazivanje mrežne konfiguracije u vidu stabla, kao i za otkrivanje problema u mreži. Mrežni saobraćaj se iz sirovog binarnog formata konvertuje u čoveku razumljiv format, čime se olakšava analiza. Obrađene poruke SCADA protokola alat prikazuje u tabeli, gde su prikazani osnovni podaci.

Detaljan prikaz aplikativnog dela poruke prikazuje se u *data grid*-u kada se izabere željena poruka u tabeli. Ukoliko se u toku parsiranja poruke ustanovi da se ne radi o poruci SCADA protokola, ta poruka se odbacuje i alat je nigde ne prikazuje. Bitno svojstvo implementiranog rešenja je dobijanje informacija o konfiguraciji mreže tj. o RTU-ovima i njihovim tačkama, samo na osnovu obrađenih poruka. Pružena je i mogućnost filtriranja poruka po određenim parametrima, kao i sortiranje po kolonama, što olakšava analizu. Još jedno od svojstava implementiranog rešenja jeste algoritam, koji je osmišljen da bi iz tabele uklanjao poruke koje su pogrešno protumačene. Implementirano rešenje pruža i prikaz performansi u posebnom prozoru.

3.1. Arhitektura realizovanog rešenja

Osnovne komponente implementiranog rešenja su:

- NetworkListener,
- PacketListener i
- PacketAnalyzer.

NetworkListener komponenta predstavlja glavnu komponentu implementiranog rešenja. Ona pokreće i zaustavlja alat, instancira komponentu PacketListener, koja služi za hvatanje paketa iz mrežnog saobraćaja izabranog mrežnog interfejsa, a takođe instancira i komponentu PacketAnalyzer, pomoću koje se obrađuju uhvaćeni paketi. PacketAnalyzer komponenta instancira komponentu za parsiranje IP i TCP zaglavlja uhvaćenog paketa, parsere za pojedinačne protokole, komponentu za formiranje i prikazivanje mrežne konfiguracije, kao i komponentu za merenje performansi rešenja.

3.2. Implementacija realizovanog rešenja

Opis implementacije programskog rešenja biće podeljen u nekoliko poglavlja, kako bi se za svako svojstvo objasnilo na koji način je implementirano. Programsko rešenje implementirano je u C# programskom jeziku.

3.2.1. Hvatanje paketa

PacketListener komponenta za hvatanje paketa koristi *raw socket,* koji omogućava da se sirovi paketi primljeni na mrežnom sloju direktno prosleđuju *raw* soketima, čime se omogućava korisniku da pristupi i manipuliše zaglavljima i korisničkim podacima protokola nižih slojeva. Hvatanje paketa se vrši u asinhronom režimu sa mrežne kartice koja je izabrana.

3.2.2. Parsiranje paketa

Nakon što su podaci uhvaćeni, potrebno je isparsirati ih. Pošto su za hvatanje paketa korišćeni *raw* soketi, uhvaćeni paketi pored aplikativnog dela poruke mogu da sadrže i mrežno i transportno zaglavlje. Ukoliko paket ne sadrži IP, TCP i aplikativna zaglavlja, taj paket se može odbaciti jer sigurno ne sadrži SCADA aplikativni deo. Sledeći korak je pokušaj parsiranja protokola aplikacionog sloja tj. podržanih SCADA protokola.

3.2.2.1. Ferranti VAN-COMM

Prepoznavanje VAN-COMM poruke započinje proverom preambule. Ukoliko preambula ima vrednost 0xFE ili 0xFD pretpostavlja se da se radi o VAN-COMM poruci. Nakon toga proverava se format poruke, poruka se parsira i pojedina polja se validiraju, kako bi se potvrdilo da je zaista u pitanju VAN-COMM protokol. Jedan VAN-COMM frejm može nositi između 4 i 14 bajta korisničkih informacija, pa se iz tog razloga vrši provera dužine uhvaćenog frejma. Kako VAN-COMM odgovor u sebi ne sadrži originalni kod funkcije, kao ni tip poslatih podataka, bilo je potrebno da se pamti kakav je zahtev poslat na RTU da bi se odgovor na taj zahtev mogao tumačiti.

Odogovor se može sastojati iz više frejmova koje nije moguće tumačiti pojedinačno, pa je bilo potrebno sačekati sve frejmove odgovora, kako bi bilo moguće protumačiti ga u celosti. Početni deo frejma svih zahteva je isti, dok formati glavnog dela frejma sa komandnim parametrima mogu biti različiti, u zavisnosti od toga koji kod funkcije se nalazi u zahtevu.

Nakon što se na osnovu koda funkcije odredi vrsta zahteva, kreira se parser za tu vrstu zahteva i vrši se parsiranje. Na sličan način implementirani su i odgovori na zahteve. Ukoliko se poruka sastoji iz više frejmova, samo prvi frejm sadrži preambulu, adresu RTU-a i statusne bite. Format glavnog dela frejma odgovora određuje se na osnovu koda funkcije iz zapamćenog zahteva. Kreira se parser za određeni format odgovora i vrši se parsiranje poruke. Prilikom parsiranja vrednosti statusnih tačaka nije moguće znati pojedinačne vrednosti tačaka zbog nedostatka znanja o tipu grupe kojoj statusna tačka pripada. Iz tog razloga implementirano je da se isparsira vrednost cele grupe, a ne pojedinačnih statusnih tačaka.

Odgovor na zahtev za skeniranjem svih statusa ne sadrži informacije o tome koja vrsta statusa se nalazi u kojoj grupi, pa rešenje ne zna kako da tretira koju grupu, da li kao *memory*, *latching* ili *momentary* statusne tačke.

Kod parsiranja analognih vrednosti bez znanja o konfiguraciji nije moguće odrediti da li se radi o neoznačenoj ili označenoj vrednosti, pa implementirano rešenje tretira sve analogne vrednosti kao neoznačene.

Kod parsiranja *Raise Lower Immediate Execute* zahteva, vrednosti vremenskih intervala čuvaju se u vidu bajtova, jer bez poznavanja konfiguracije nije moguće tumačiti te vrednosti na osnovu protokolom definisane tabele. Kod

kontrolnih operacija (*Control Select, Control Execute i Control Immediate Execute*) implementirano rešenje samo čita koji su releji selektovani u poruci, ali zbog nedostatka znanja o konfiguraciji ne može da rastumači koja će od operacija (TRIP, CLOSE ili PULSE) biti izvršena.

3.2.2.2. DNP3.0

Da bi se pretpostavilo da je korišćeni aplikativni protokol DNP3, vrednosti prva dva bajta frejma moraju biti 0x05 i 0x64. Ukoliko je taj uslov ispunjen, proverava se format poruke, vrši se parsiranje i validiraju se polja poruke. DNP3 poruke mogu se sastojati iz više frejmova koje nije moguće tumačiti pojedinačno, pa je bilo potrebno sačekati sve frejmove kako bi se poruka protumačila u celosti. Pomoću pristiglog frejma kreira se segment, odstranjujući zaglavlje sloja veze, i vrši se njegovo parsiranje. Sledeći bitan korak je validacija sekvence segmenta, gde se na osnovu broja sekvence, koji se nalazi u zaglavlju segmenta, proverava da li su segmenti primljeni u dobrom redosledu.

Kada se primi poslednji frejm, spajanjem svih prethodno primljenih frejmova, odnosno segmenata, kreira se fragment i vrši se njegovo parsiranje. Parsiranje fragmenta podrazumeva parsiranje zaglavlja aplikacionog sloja, parsiranje zaglavlja objekata, kao i samih DNP3 objekata. Na osnovu grupe i varijacije dobijenih parsiranjem zaglavlja objekta, određuje se tip DNP3 objekata. Pomoću mapiranja se na osnovu tipa objekta dobija funkcija, pomoću koje se kreira DNP3 objekat. Zatim se vrši parsiranje koje može biti različito, u zavisnosti od toga o kom se DNP3 objektu radi.

3.2.2.3. IEC 101 i IEC 104

Prvi korak prilikom pretpostavke da je aplikativni protokol uhvaćenog paketa IEC 101 podrazumeva proveru preambule, gde su moguće vrednosti 0x68, 0x10 i 0xE5. Na osnovu vrednosti preambule kreira se određeni format frejma i vrši se njegovo parsiranje uz validaciju polja frejma. Značenje bita u kontrolnom polju frejma razlikuje se u zavisnosti od toga da li je komunikacija balansirana ili nije. Zbog nedostatke znanja o konfiguraciji, implementirano rešenje u detaljnom prikazu poruke bite iz kontrolnog polja prikazuje sa nazivima iz oba formata, tako da njihovo značenje nije jasno određeno.

Kod IEC 101 protokola postoje dva formata ASDU poruka. Format je određen poljem SQ. Ukoliko je 0 vrednost polja SQ, ASDU nosi pojedinačne informacione objekte, gde svaki objekat ima svoju koordinatu. U suprotnom, ASDU ima jedan informacioni objekat, koji u sebi nosi sekvencu informacionih elemenata, gde samo prvi element ima koordinatu, a ostale koordinate se računaju na osnovu prve. Na osnovu polja koje određuje tip podataka i vrednosti polja SQ, određuje se koja je vrsta informacionog objekta u pitanju, tačnije kreira se odgovarajući parser za taj objekat i njegove elemente. Parseru se dodaje i odgovarajući deskriptor kvaliteta, kao i vremenska oznaka ukoliko je to potrebno. Zatim se vrši parsiranje i validacija polja.

Implementacija IEC 104 parsera je slična implementaciji IEC 101 parsera. Vrednost preambule za pretpostavku IEC 104 protokola mora biti 0x68. IEC 104 za prenos podataka koristi strukturu APDU, koja u sebi pored ASDU-a sadrži i APCI. Postoje tri tipa APCI-a, koja definišu tri APDU formata. I-format pored APCI-a sadrži i ASDU, koji može imati dva formata. Ovaj problem je rešen na isti način kao što je prethodno opisano za IEC 101. Kod rešenja smešten je u *CommonIEC* projekat, koji koriste parseri oba protokola.

3.2.3. Čuvanje i prikazivanje isparsiranih paketa

Poruke svih podržanih protokola imaju različite formate, međutim programsko rešenje koristi klasu *Protocol-DiagnosticsChannelData* za čuvanje informacija o raznorodnim protokolima. Parser nakon parsiranja popunjava *property*-e prethodno pomenute klase. Isparsirani paket se preko *event*-a dodaje u listu uhvaćenih paketa na *ViewModel*-u, čiji se elementi prikazuju u tabeli.

3.2.4. Brisanje pogrešno protumačenih poruka

Da bi se unapredila analiza poruka osmišljen je pametni algoritam, koji uklanja poruke koje su pogrešno protumačene. Implementirano rešenje čuva informaciju o portovima na koje su stizale SCADA poruke i broj pristiglih poruka, kao i informaciju o ostalim portovima sa brojem neuspešnih pokušaja obrade na njima. Ukoliko se nakon pokušaja parsiranja zaključi da uhvaćena poruka ne pripada SCADA protokolima, poziva se metoda *CheckIsFailedTriesExceededLimit*.

Ova metoda proverava da li su na taj komunikacioni link prethodno stizale SCADA poruke. Ako jesu, proverava se da li je broj neuspešnih obrada poruka veći od 75% ukupnog broja pokušaja obrada na tom komunikacionom linku (uspešni + neuspešni). Ukoliko jeste, pomoću *event*-a se na *ViewModel*-u brišu iz tabele sve poruke koje sadrže tu IP adresu i port. Takođe se i iz konfiguracije briše komunikacioni link i RTU-ovi koji se nalaze pod njim.

3.2.5. Filtriranje uhvaćenih poruka

Programsko rešenje podržava filtriranje radi lakšeg praćenja mrežnog saobraćaja. Parametri po kojima se može filtrirati su imena kolona iz tabele, predstavljena skraćenim stringom, dok su operatori koji se mogu koristiti: ==, !=, !*contains*, *contains*, *and* i *or*. Ispravna forma filtera je: <ime kolone><operator><vrednost>.

3.2.6. Prikaz mrežne konfiguracije

Programsko rešenje pravi mrežnu konfiguraciju na osnovu isparsiranih SCADA poruka. U konfiguraciji postoji podela po protokolima, a ispod svakog protokola čuvaju se komunikacioni linkovi. Pod jednim komunikacionim linkom može se nalaziti više RTU-ova sa ulaznim i izlaznim tačkama. Kada se izvrši parsiranje poruke, unapređuje se mrežna konfiguracija ukoliko je to potrebno tj. proverava se da li se protokol, komunikacioni link, RTU i njegove tačke već nalaze u konfiguraciji ili je potrebno dodati ih.

3.2.7. Prikaz performansi procesiranja

Da bi se omogućilo praćenje performansi, koristi se štoperica, koja startuje kada kreće obrada paketa, a zaustavlja se kada se obrada završi.

Na početku obrade uvećava se ukupan broj uhvaćenih paketa. Nakon obrade, ukoliko je uspešno isparsiran SCADA protokol, uvećava se broj SCADA paketa i ukupno vreme SCADA obrade.

U suprotnom se uvećava broj paketa koji nisu SCADA i ukupno vreme neuspešnih obrada. Pamti se i broj neuspešnih pokušaja obrade za svaki komunikacioni link. Prozor za performanse prikazuje sledeće vrednosti:

- ukupan broj svih uhvaćenih paketa,
- broj uhvaćenih SCADA paketa,
- broj ostalih paketa,
- procentualni prikaz uhvaćenih SCADA paketa u odnosu na ukupan broj svih uhvaćenih paketa,
- prosečna vremena obrade SCADA i ostalih paketa, i
- port sa najviše neuspešnih pokušaja obrade.

4. VERIFIKACIJA PROGRAMSKOG REŠENJA

Verifikacija implementiranog programskog rešenja vršena je testiranjem četiri slučaja, koji se razlikuju po broju obrađenih poruka u određenom vremenskom periodu. Postepeno je povećavano opterećenje mreže, odnosno broj SCADA poruka u mreži. U sva četiri slučaja merenje je vršeno 10 minuta nakon pokretanja alata. Tabela 1 prikazuje rezultate testiranja.

Tabela 1: *Rezultati testiranja programskog rešenja*

Broj rtu-a	890	970	1070	1670
Broj tačaka	18.360	34.360	54.352	174.352
Ukupan broj paketa	391.788	506.554	620.417	1.203.788
SCADA paketi	153.958	204.134	252.270	499.345
Ostali paketi	237.830	302.429	368.147	704.442
Procenat SCADA paketa	39,3%	40,3%	40,7%	41,5%
Prosečno vreme obrade SCADA paketa [μs]	161	199	212	306
Prosečno vreme obrade ostalih paketa [μs]	9	10	11	12
Procenat vremena utrošenog na obradu ostalih paketa	7,9%	7,2%	7,1%	5,2%

U prvom slučaju obrađivano je oko 256 SCADA poruka u sekundi, u drugom slučaju 340, u trećem slučaju 420, a u četvrtom 832 SCADA poruke u sekundi. Sa porastom broja SCADA poruka raste i procenat SCADA poruka u odnosu na ukupan broj uhvaćenih poruka. Može se primetiti i da prosečno vreme obrade SCADA paketa postepeno raste, dok se prosečno vreme obrade ostalih paketa povećava za po jednu mikrosekundu. Procenat vremena utrošenog na obradu ostalih paketa smanjuje se sa porastom uspešnih SCADA obrada.

Ovim testiranjem verifikuje se da implementirano programsko rešenje uspešno uspeva da prepoznaje SCADA protokole, kao i da obradi i isparsira uhvaćene SCADA pakete, bez potrebe za prethodnim mapiranjem portova i protokola, što je i bio cilj ovog rada.

5. ZAKLJUČAK

Analiza mrežnog saobraćaja predstavlja bitnu stavku prilikom održavanja svake mreže. Ukoliko mrežni administratori primete nepravilnosti u radu mreže, kroz analizu snimljenog saobraćaja može da se otkrije uzrok problema. Cilj ovog rada bio je razvoj programskog rešenja, čiji je zadatak da prati mrežni saobraćaj SCADA protokola i da obrađuje uhvaćene pakete, kako bi se omogućila detaljna analiza mrežnog saobraćaja.

Većina postojećih alata za praćenje mrežnog saobraćaja za detekciju protokola koristi predefinisane portove mapirane na podržane protokole. To nije pouzdana tehnika, a i može oduzeti puno vremena, zbog potrebe za ručnim mapiranjem portova i protokola. Zato implementirano rešenje prepoznaje protokole na osnovu parsiranja poruke i validiranja pojedinih polja protokola. Time je omogućeno praćenje mrežnog saobraćaja bez potrebe za prethodnim poznavanjem mrežne konfiguracije i za mapiranjem portova i protokola.

Prednost u odnosu na većinu postojećih alata jeste i mogućnost dobijanja mrežne konfiguracije na osnovu obrađenih poruka SCADA mrežnog saobraćaja, što je značajno korisniku rešenja ukoliko se radi o njemu prethodno nepoznatom SCADA sistemu. Za razliku od postojećih alata, programsko rešenje eksplicitno je namenjeno za praćenje mrežnog saobraćaja SCADA protokola.

Jedan od mogućih pravaca daljeg razvoja jeste dodavanje parsera za brojne SCADA protokole koji nisu trenutno obuhvaćeni implementiranim rešenjem. Bilo bi korisno unaprediti logiku za detekciju protokola, pamćenjem koji je protokol detektovan na kom portu, da bi se sledeći put kad se uoči poznati port pokušalo prvo sa parsiranjem zapamćenog protokola, kao i pamćenjem neuspešnih detekcija protokola za svaki port, da se sledeći put ne bi gubilo vreme na neuspešne pokušaje. Takođe je poželjno i dodavanje veštačke inteligencije za detekciju protokola.

6. LITERATURA

[1] Wireshark. http://www.wireshark.org/ (Pristupljeno 23.07.2019.)
[2] All about TCPdump. http://www.TCPdump.org/ (Pristupljeno 23.07.2019.)
[3] NetDecoder. http://www.fte.com/docs/usermanuals/-NetDecoderAsyncUM.pdf (Pristupljeno 23.07.2019.)
[4] All about Colasoft Capsa. http://www.colasoft.com (Pristupljeno 23.07.2019.)
[5] Appendix B, Ferranti International Controls, *Message Standard: Van-Comm*
[6] DNP3 SPECIFICATION, Version 2.01, 2007
[7] G. Clarke, D.Reynders, E. Wright, *Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems*, Elsevier, 2004.
[8] P.Matoušek, *Description and analysis of IEC 104 Protocol*, Brno University of Technology, 2017.

Kratka biografija:

Zvezdana Klašnić rođena je u Novom Sadu 1994. godine. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva, smer Primenjeno softversko inženjerstvo, odbranila je 2019. godine.
Kontakt: zklasnic94@gmail.com