



УНИВЕРЗИТЕТ У НОВОМ САДУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У  
НОВОМ САДУ



Драган Ракита

**Проширење система за  
аутоматизацију куће са  
могућношћу даљинског надзора**

**ДИПЛОМСКИ РАД**  
**- Основне академске студије -**

Нови Сад, 2014



## КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:			
Идентификациони број, ИБР:			
Тип документације, ТД:	Монографска документација		
Тип записа, ТЗ:	Текстуални штампани материјал		
Врста рада, ВР:	Завршни (Bachelor) рад		
Аутор, АУ:	Драган Ракита		
Ментор, МН:	Др Иштван Пап, доцент		
Наслов рада, НР:	Проширење система за аутоматизацију куће са могућношћу даљинског надзора		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2014		
Издавач, ИЗ:	Ауторски репримт		
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО: (поглавља/страница/цитата/табела/слика/графика/прилога)	8 поглавља / 29 страница / 25 referenci / 11 слика		
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарска техника		
Предметна одредница/Кључне речи, ПО:	Паметна кућа, Обло, ТР-069, удаљена контрола		
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:	Постојећу платформу за аутоматизацију куће засновану на ZigBee комуникацији потребно је проширити са модулом за повезивање са постојећим системом за удаљену дијагностику и надзор. Додати модул потребно је повезати са остатком система у циљу обезбеђивања удаљене контроле, преко интернета.		
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:	Др Илија Башичевић, доцент	
	Члан:	Др Небојша Пјевалица, доцент	Потпис ментора
	Члан, ментор:	Др Иштван Пап, доцент	



## KEY WORDS DOCUMENTATION

Accession number, ANO:		
Identification number, INO:		
Document type, DT:	Monographic publication	
Type of record, TR:	Textual printed material	
Contents code, CC:	Bachelor Thesis	
Author, AU:	Dragan Rakita	
Mentor, MN:	Dr Ištván Pap, docent	
Title, TI:	Remote monitoring for home automation system	
Language of text, LT:	Serbian	
Language of abstract, LA:	Serbian	
Country of publication, CP:	Republic of Serbia	
Locality of publication, LP:	Vojvodina	
Publication year, PY:	2014	
Publisher, PB:	Author's reprint	
Publication place, PP:	Novi Sad, Dositeja Obradovica sq. 6	
Physical description, PD: (chapters/pages/ref./tables/pictures/graphs/appendices)	8 chapters / 29 pages / 25 ref. / 11 pictures	
Scientific field, SF:	Electrical Engineering	
Scientific discipline, SD:	Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, S/KW:	Smart home, OBLO, TR-069, Remote Control	
UC		
Holding data, HD:	The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, N:		
Abstract, AB:	Existing platforms for home automation based on Zigbee communication is needed to be expand with module connecting to the existing system for remote diagnosis and monitoring over internet.	
Accepted by the Scientific Board on, ASB:		
Defended on, DE:		
Defended Board, DB:	President:	Dr Ilija Bašičević, docent
	Member:	Dr Nebojša Pjevalica, docent
	Member, Mentor:	Dr Ištván Pap, docent
		Menthor's sign

## **Zahvalnica**

Ovim putem se posebno zahvaljujem svojoj porodici na pruženoj sveobuhvatnoj podršci tokom studiranja.

Zahvaljujem se i mentoru dr Ištvanu Papu i stručnom saradniku Romanu Pavloviću na savetima i pomoći tokom izrade ovog rada.

## SADRŽAJ

1.	Uvod .....	6
2.	Teorijske osnove.....	7
2.1	OBLO sistem .....	7
2.1.1	OBLO menadžer.....	8
2.2	Komunikacioni protokol TR-069.....	10
2.2.1.1	ACS HTTP komunikacioni protokol .....	11
2.3	Android .....	13
2.3.1	OBLO Android aplikacija .....	13
3.	Koncept rešenja .....	16
3.1	Proširenje OHM-a.....	17
3.2	Proširenje OBLO Android aplikacije.....	17
4.	Programsko rešenje.....	18
4.1	CPE Kontroler.....	18
4.1.1	Model podataka TR-230 i proširenje CPE modula .....	19
4.2	Modifikacija OBLO Android aplikacije .....	20
4.2.1	Implementacija komunikacije sa ACS serverom. ....	21
4.2.1.1	Klasa InsightACSConnection .....	21
4.2.1.2	Klasa InsightACSResponse .....	23
4.3	Primer komunikacije.....	24
5.	Rezultati.....	26
5.1	Vreme odziva .....	27
6.	Zaključak .....	29
7.	Literatura .....	30

## SPISAK SLIKA

Slika 2.1 Prikaz uređaja OBLO pametne kuće .....	7
Slika 2.2 Blokovska arhitektura OBLO menadžera.....	9
Slika 2.3 Tok prosleđivanja poruke između uređaja i kontrolera .....	9
Slika 2.4 Arhitektura CWMP protokola .....	10
Slika 2.5 Protokol stek CWMP protokola .....	11
Slika 2.6 Primer JSON poruke za promenu parametara .....	12
Slika 2.7 Blokovski izgled navigacije OBLO Android aplikacije.....	14
Slika 2.8 Kontrola uređaja u sobi.....	14
Slika 3.1 Arhitektura sistema.....	16
Slika 4.1 Dijagram sekvene komunikacije od OBLO Android aplikacije do pametne utičnice .....	24
Slika 4.2 Dijagram sekvene komunikacije od pametne utičnice do OBLO Android aplikacije .....	24
Slika 5.1 Vremenski odziv sistema.....	27

## **SPISAK TABELA**

Tabela 1 Ispitni slučajevi ispravnosti OBLO Android aplikacije.....	26
Tabela 2 Ispitni slučajevi ispravnosti CPE Kontrolera.....	27

## SKRAĆENICE

- TV** – Televizija
- TR** – Tehnički izveštaj (eng. *Technical Report*)
- CPE** – Krajnji potrošački uređaj (eng. *Customer-Premises Equipment*)
- ACS** – Poslužilac za automatsku konfiguraciju (eng. *Auto Configuration Server*)
- WAN** – Mreža širokog područja (eng. *Wide Area Network*)
- CWMP** – Protokol CPE WAN upravljanja (eng. *CPE WAN Management Protocol*)
- SDK** – Skup računarskih alata (eng. *Software Development Kit*)
- API** – Sprega za programiranje aplikacija (eng. *Application Programming Interface*)
- STB** – Lokalni televizijski pretvarač (eng. *Set-Top Box*)
- JSON** – JavaScript objektna notacija (eng. *JavaScript Object Notation*)
- XML** – Proširivi jezik za označavanje (eng. *Extensible Markup Language*)
- LAN** – Lokalna Računarska Mreža (eng. *Local Area Network*)
- IP** – Internet protokol (eng. *Internet Protocol*)
- TCP** – Protokol za upravljanje prenosom (eng. *Transmission Control Protocol*)
- REST** – Prenos reprezentacionog stanja (eng. *Representational State Transfer*)
- OHM** – eng. *OBLO Home Manager*
- OUI** – Identifikator administrirajuće organizacije (eng. *Organizationally Unique Identifier*)
- ID** – Jedinstveni identifikator (eng. *Identification*)

## 1. Uvod

Pametna kuća je sistem kontrole i nadgledanja uređaja u stambenom objektu sa ciljem povećanja komfora, sigurnosti i smanjenje troškova potrošnje energije. Pametni uređaji koji se nalaze u kući omogućavaju šire mogućnosti kontrole od klasičnih standardnih kontrola.

ZigBee tehnologija postaje standard za komunikaciju između uređaja u automatizovanoj kući [1]. U predloženom rešenju koristi se ZigBee komunikacija između uređaja, dato rešenje se zove OBLO [2]. OBLO sadrži sprege kojima je moguća kontrola pametnih uređaja. Implementacija rešenja se sastoji od dodavanja modula koji omogućava udaljenu kontrolu preko interneta.

Kao najbolje rešenje odabira protokola za udaljenu kontrolu uređaja u kući uzet je TR-069 protokol koji će kasnije biti malo bolje opisan. Rešenje koje implementira TR-069 protokol se zove Insight [14]. Postojanje OBLO Android aplikacije koja koristi sprege OHM-a u lokalnoj mreži daje nam mogućnost zadržavanja znanog korisničkog iskustva uz dodatak nove funkcionalnosti.

Rad je podeljen u pet delova:

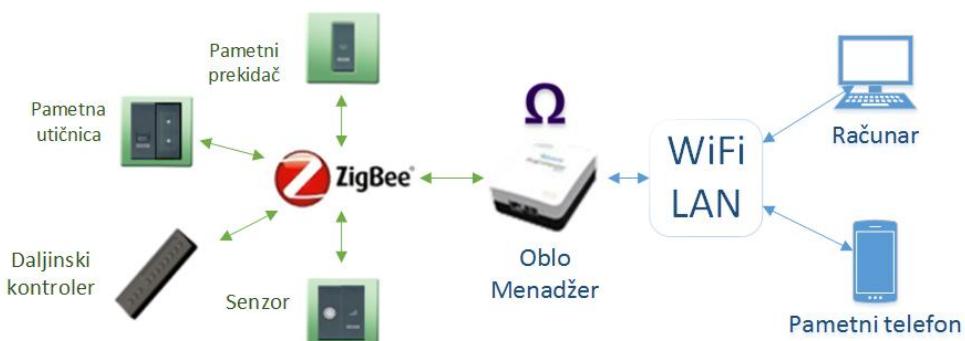
- Teorijska osnova – Opisuju se način funkcionisanja OHM i njegove postojeće OBLO Android aplikacije, OBLO sprežnog sistema i komunikacioni protokol TR-069.
- Koncept rešenja – Daje uvid u izmene i nadogradnje potrebne za dodavanje podrške postojećih rešenja.
- Programsко rešenje – Pruža detalje vezane za implementaciju predloženog rešenja. Daje uvid u izgled klasa i funkcija u realizovanog rešenja u sklopu Pametne kuće i OBLO Android aplikacije.
- Rezultati – Ispitivanje sistema.
- Zaključak – Iznešene su prednosti ovakvog rešenja.

## 2. Teorijske osnove

Teorijske osnove daju kratak pregled sistema, sprega i protokola na koje se rešenje oslanja. OBLO sistem je prvi opisan a akcenat je stavljen na OHM. OHM (eng. OBLO Home Manager) je centralna upravljačka jedinica OBLO sistema. Protokol TR-069 je opisan sledeći, dat je opis osnovnog načina funkcionisanja protokola i način proširenje osnovnog modela podataka. Definisan je protokol za komunikaciju koji uređaji treba da koriste prilikom pristupanja ACS servera. Poslednji deo poglavlja daje opis OBLO Android aplikacije.

### 2.1 OBLO sistem

OBLO sistem se sastoji od različitih pametnih uređaja koji se dele u tri grupe: senzori, aktuatori i kontroleri. Trenutno su podržana dva aktuatorska uređaja: pametni prekidač i pametna utičnica. Centralna kontrolna jedinica OBLO (OHM) ima za cilj uspostavljanje kontrole i upravljanja ostalih uređaja u OBLO sistemu. OBLO je primarno orijentisan ka osvetljenju i električnim uređajima u kući. Uređaji su umreženi u ZigBee mrežu sa mrežastom topologijom (eng. *mesh network*) povezivanja koja omogućava slanje od najudaljenijeg uređaja do OHM-a preko drugih uređaja time doprinoseći povećanu područja koje OHM može da kontroliše.



Slika 2.1 Prikaz uređaja OBLO pametne kuće

Na Slika 2.1 su prikazane veze između uređaja u OBLO sistemu. Pametni uređaji koji se nalaze u Oblu koji su povezani u ZigBee mreži su:

- Pametni prekidači  
omogućava uključivanje i isključivanje sijalice i kontrolu osvetljenja
- Pametne utičnice  
uz standardno prekidanje i uspostavljenje strujnog kola uz pomoć releja daju mogućnost merenja potrošnje uređaja koja su priključena na njih. Potrošnje se meri u pametnoj utičnici, a periodično se šalje izveštaj merenja koji sadrži srednju potrošnju električne energije nad tim periodom.
- Daljinski upravljač
- Senzori pokreta

Kontrola uređaja se može raditi ručno pritiskom na odgovarajući taster, pomoću daljinskog upravljača ili posredstvom OHM-a.

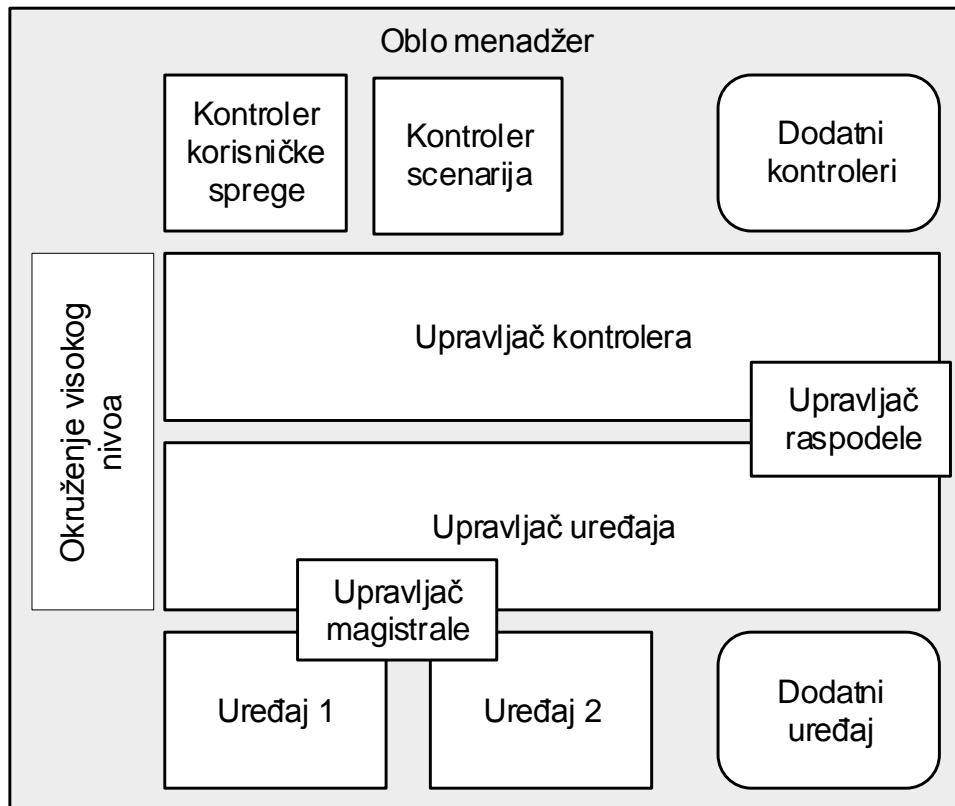
### **2.1.1 OBLO menadžer**

OHM je centralna kontrolna jedinica koja se može opisati kao most između ZigBee mreže u kojoj se nalaze uređaji pametne kuće i IP (LAN/Wi-Fi) mreže. Pomoću njega se konfigurišu, upravljaju i nadgledaju promena uređaja. On sadrži nekoliko sprega koje omogućavaju korisniku kontrolu nad OBLO sistemom, sprege su detaljnije opisane u ovom radu [3].

Arhitektura OHM programa se sastoji od četiri glavna upravljačka modula:

- Upravljač raspodele – Zadužen je za raspoređivanje uređaja po grupama. Grupe su logičke celine koje sadrže jedan ili više uređaja i definisane su od strane korisnika.
- Upravljač magistrale - Obezbeđuje apstrakciju pristupa elementima fizičke arhitekture za ZigBee komunikacije.
- Upravljač kontrolera – Odgovoran je za rukovanje drugim upravljačkim modulima. Implementira mehanizme za kontrolu upravljača i razmenu poruka između različitih upravljača.
- Upravljač uređaja – Zadužen je za rukovanje informacijama o uređajima.

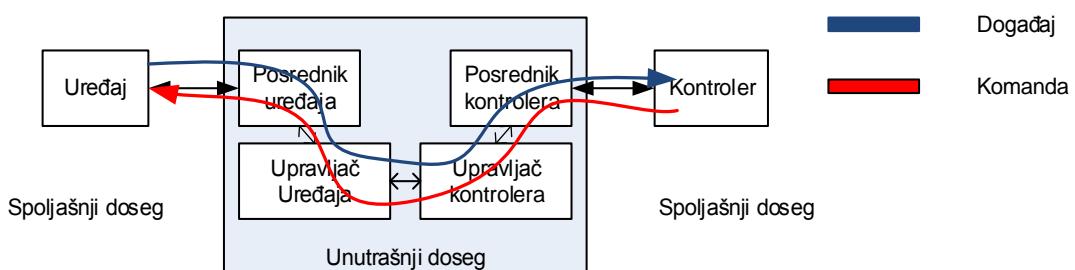
Blok dijagram osnovnih modula OHM-a se može videti na Slika 2.2. Slika sadrži i prikaz četiri upravljača koji su opisani i njihova relativna mesta u OHM-u..



Slika 2.2 Blokovska arhitektura OBLO menadžera

Pošto je OHM dosta kompleksan, u daljem ćemo se usredsrediti na upravljač kontrolera kao deo sistema čijem proširenjem se implementira željena funkcionalnost. Upravljač kontrolera omogućava kontrolerima da se sprežu međusobno i sa ostatom sistemom koristeći podsistem za obaveštenje i slanje komandi. Kontroleri se putem podsistema za obaveštenje registruju na odgovarajuće događaje. Komande se mogu slati drugim kontrolerima direktno unutar upravljača kontrolera ili uređajima u sistemu posredstvom upravljača uređaja. Upravljač kontrolera može da sadrži više različitih kontrolera.

Na Slika 2.3 se može videti tok prosleđivanja poruke događaja i komandne poruke od kontrolera do uređaja i obrnuto.

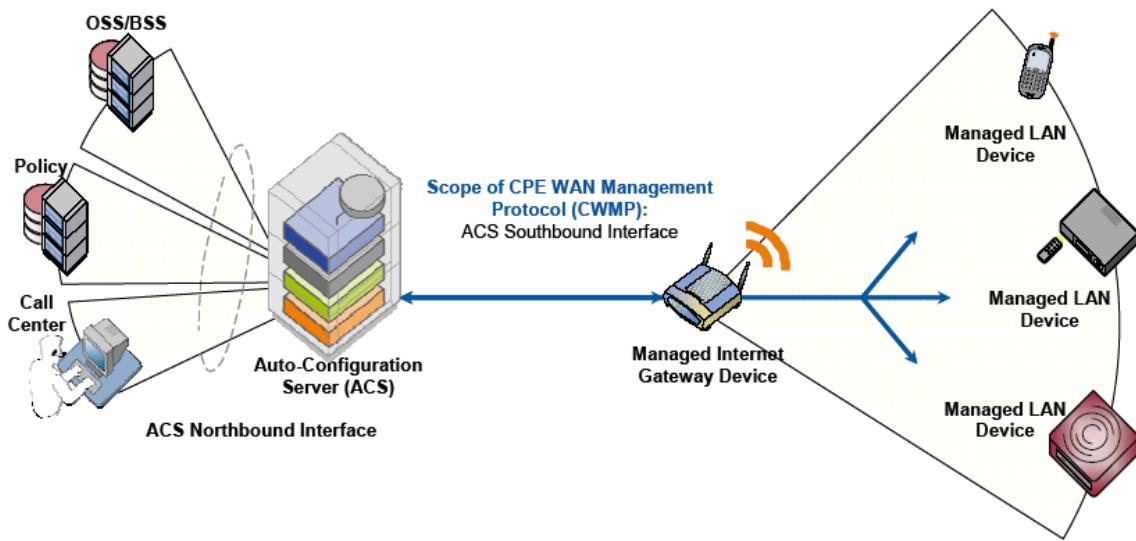


Slika 2.3 Tok prosleđivanja poruke između uređaja i kontrolera

Kontroleri se definišu u konfiguracionoj XML datoteci na samom OHM-u. Uređivač kontrolera pri pokretanju na osnovu konfiguracije očitava kontrolere sa njihovim podešavanjima. Podešavanje parametara kontrolera se može izvršiti iz ove datoteke ili pomoću određenih kontrolera OHM-a se pri zaustavljanju OHM-a čuvaju u ovoj datoteci.

## 2.2 Komunikacioni protokol TR-069

Je definisan od strane organizacije *Broadband forum*. Puni ime protokola je CPE WAN upravljački protokol (*eng. CWMP, Customer Premises Equipment Wide Area Network Management Protocol*) [8]. Protokol služi da spoji dva tipa uređaja, jedan tip je krajnji korisnički uređaj CPE (*eng. Customer Premises Equipment*) koji može biti lokalni televizijski pretvarač, usmerivač, pametni telefon, pametni televizor ili u ovom slučaju OHM uređaj, a drugi tip uređaja je autokonfiguracioni poslužitelj ACS (*eng. Auto-Configuration Server*). Na Slika 2.4 je predstavljena pojednostavljena arhitektura CWMP protokola.



Slika 2.4 Arhitektura CWMP protokola

Komunikacija između ACS i CPE uređaja je dvosmerna. CWMP sadrži mehanizam uz pomoć kojeg ACS može da uspostavi vezu sa CPE uređaj u mreži koja koristi NAT (*eng. Network Address Translation*). Mehanizam se bazira na UDP NAT-T (*eng. NAT Traversal*) i STUN (*eng. Session Traversal Utilities for NAT*) protokolima koji je definisan u dodatku G TR-069 specifikacije[11][21]. Ovo proširenje je ranije poznato kao TR-111 [25].

CWMP se zasniva na poznatim komunikacionim tehnologijama. Arhitektura steka protokola se prikazana na Slika 2.5.

RPC metode
SOAP
HTTPS
SSL/TLS
TCP/IP

Slika 2.5 Protokol stek CWMP protokola

Na dnu protokol steka je standardni TCP/IP protokol. TCP/IP sloj je proširen slojem za SSL/TSL enkripciju [16][17][18][19]. Ovaj sloj obezbeđuje sigurnosnu podršku slojevima višeg nivoa. Srednji sloj steka je HTTPS (*eng. HyperText Transfer Protocol Secure*) protokol [20]. *SOAP* (*eng. Simple Object Access Protocol*) protokol je zasnovan na XML-u (*eng. EXtensible Markup Language*) i koristi se za enkodovanje RPC (*eng. Remote Procedure Call*) poziva [21][22][23]. Naznačene RPC metode su definisane od strane CWMP protokola.

Tehnički izveštaj TR-106 određuje smernice izrade modela podataka koje trebaju slediti se svi TR - 069 uređaji [9]. Izveštaj precizira kako se definišu i strukturiraju modeli podataka. Podaci su organizovani u kolekcije objekata i parametara nad kojima generičke metode vrše konfigurisanje, dijagnozu i nadgledanje uređaja.

Smernice TR-106 izveštaja uključuju:

- strukturne zahteve koji se odnose na hijerarhiju podataka
- Zahteva se određivanje verzija modela
- zahteve za definisanje profila podataka.

Stvarni model podataka se definiše poštujući smernice iz TR-106 izveštaja.

### 2.2.1.1 ACS HTTP komunikacioni protokol

ACS daje mogućnost kontrole CPE uređaja putem REST-a (*eng. Representational State Transfer*) [26]. REST daje način da se stvaraju, čitaju, ažuriraju ili brišu informacije na poslužitelju, koristeći jednostavan HTTP poziv ka ACS poslužitelju. Sadržaj REST poruka se formira koristeći JSON.

JSON (*eng. JavaScript Object Notation*) je tekstualno baziran otvoreni standard dizajniran za ljudima razumljivu razmenu podataka. On je izведен iz JavaScript jezika za predstavljanje jednostavnih struktura podataka. Uprkos vezi sa JavaScript jezikom, on je jezički nezavistan. Primer JSON formatirane poruke se može videti na Slika 2.6

```
{
  "deviceIdStruct": {
    "manufacturer": "RT-RK",
    "oui": "FC-FE-88",
    "productClass": "RK-OBLO",
    "serialNumber": "5784324"
  },
  "paths": ["Device.EnergyManagementService.LightControl.Dimmer.1.Mode"],
  "values": ["On"],
  "username": "admin",
  "messageType": 64
}
```

Slika 2.6 Primer JSON poruke za promenu parametara

ACS podržava poruke koje omogućavaju ažuriranje podataka CPE uređaja. Obavezno polje koje sve poruke moraju da sadrže je ime korisnika. Određene komande zahtevaju da se specificira CPE uređaj. Spisak poruka koje ACS podržava se mogu videti u narednoj listi:

- Login  
Prijava uređaja na ACS Sa odgovarajućim akreditivima.
- searchCPEList  
Pretraživanje liste CPE uređaja po vrednostima parametara:
  - Manufacturer - Proizvođač
  - OUI (*eng. Organizationally Unique Identifier*) – identifikacioni broj od 24 bita.
  - Product Class – klasa CPE uređaja
  - Serial Number – serijski broj
- getCPEList  
Dobijanje cele liste CPE uređaja
- getHeatMap  
Dobijanje mape raspoređenosti CPE uređaja
- setCPECoordinate  
Dobijanje koordinate određenog CPE uređaja
- getCPEStatus  
Dobijanje statusa CPE uređaja, uključujući informacije o trenutnom stanju veze između ACS i CPE uređajima.

- `getCPEReferences`
  - Dobijanje referenci CPE uređaja
- `getCPETreeChildren`
  - Dobavljanje svih ugnježdenih parametara.
- `setCPEReferences`
  - Podešavanje vrednosti parametara.

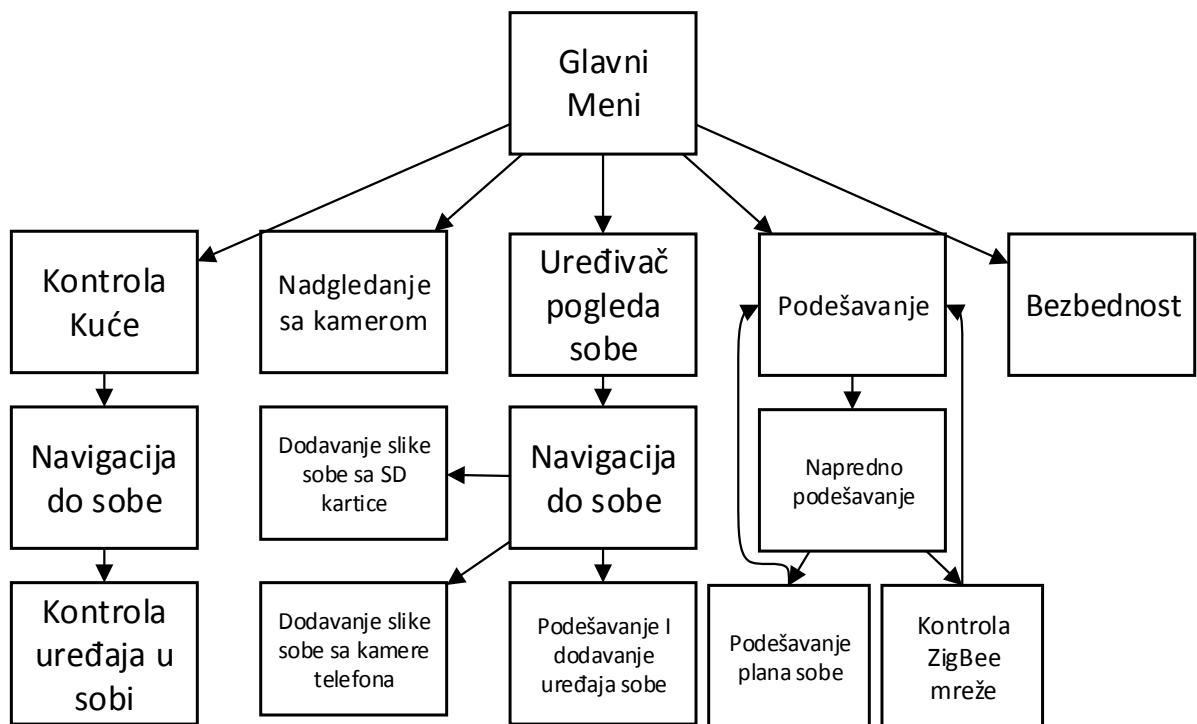
## 2.3 Android

Android je operativni sistem zasnovan na *Linux* jezgru sa korisničkim spregom koja se zasniva na direktnoj manipulaciji uređaja. Namenjen pre svega mobilnim uređajima sa ekranom osetljivim na dodir kao što su pametni telefoni i tablet računari. Operativni sistem koristi ekran osetljiv na dodir kao primarni mehanizam ka interakciji sa korisnikom. Kontrolnim pokretima kao što su pritiskivanje ekrana, pomicanje u stranu, primicanje i odmicanje prstiju se kontroliše uređaj. Uprkos tome što je pre svega namenjen za ekran osetljiv na dodir, takođe se koristi u televizorima, igračkim konzolama, digitalnim kamerama i potrošačkoj elektronici [15].

### 2.3.1 OBLO Android aplikacija

Postojeća Android klijentska aplikacija za kontrolu OBLO sistema se povezuje sa OHM-om u okviru lokalne mreže. OBLO Android aplikacija omogućuje korisniku kontrolu OBLO kuće sa njegovog tableta, pametnog telefona ili nekog drugog Android uređaja. Kontrolna sprega je intuitivna i organizovana u panelu. Uređaji su logički grupisani po sobama, a sobe su grupisane po spratovima.

Aplikacija se sastoji iz pet delova. Izgled navigacije OBLO Android aplikacije možete videti na Slika 2.7.



Slika 2.7 Blokovski izgled navigacije OBLO Android aplikacije

Opcija **Kontrola kuće** omogućava navigaciju kroz sobe. Svaka soba je obeležena sa njenim spratom i imenom. Svaka soba može da ima jedan ili više pogleda. Svaki pogled je definisan imenom i slikom pogleda. Izgled sobe i način kontrole možete videti na Slika 2.8.



Slika 2.8 Kontrola uređaja u sobi.

Na levom delu slike se vidi pogled na sobu. Crveni krugovi na slici su namenske kontrole za kontrolu uređaja. Izgled i pozicija svakog uređaja na ekranu se može uređivati iz uređivača pogleda sobe. Odabirom kontrole uređaja dobija se kontrolni prozor. Na desnom delu Slika 2.8 može se videti primer kontrolnog prozora pametnog prekidača. Prozor sadrži opcije za kontrolu uključivanja, isključivanja i podešavanje jačine osvetljenja sijalice. Za ostale uređaje kontrolni prozor se formira na osnovu osobina uređaja.

**Nadgledanje sa kamerom** daje mogućnost povezivanja IP kamere i gledanje snimka u sklopu OBLO Android aplikacije.

---

**Uređivač pogleda sobe** Omogućava dodavanje, brisanje slika od sobe i da pozicioniranje uređaja na pogledu, predstavljene kao crveni krugovi na Slika 2.8, na odgovarajuće mesto u sklopu tih slika. Postoje dva načina izabira slike:

- dodavanjem nove slike u sobu sa kamerom telefona ili
- izborom slike sa memorijske kartice telefona.

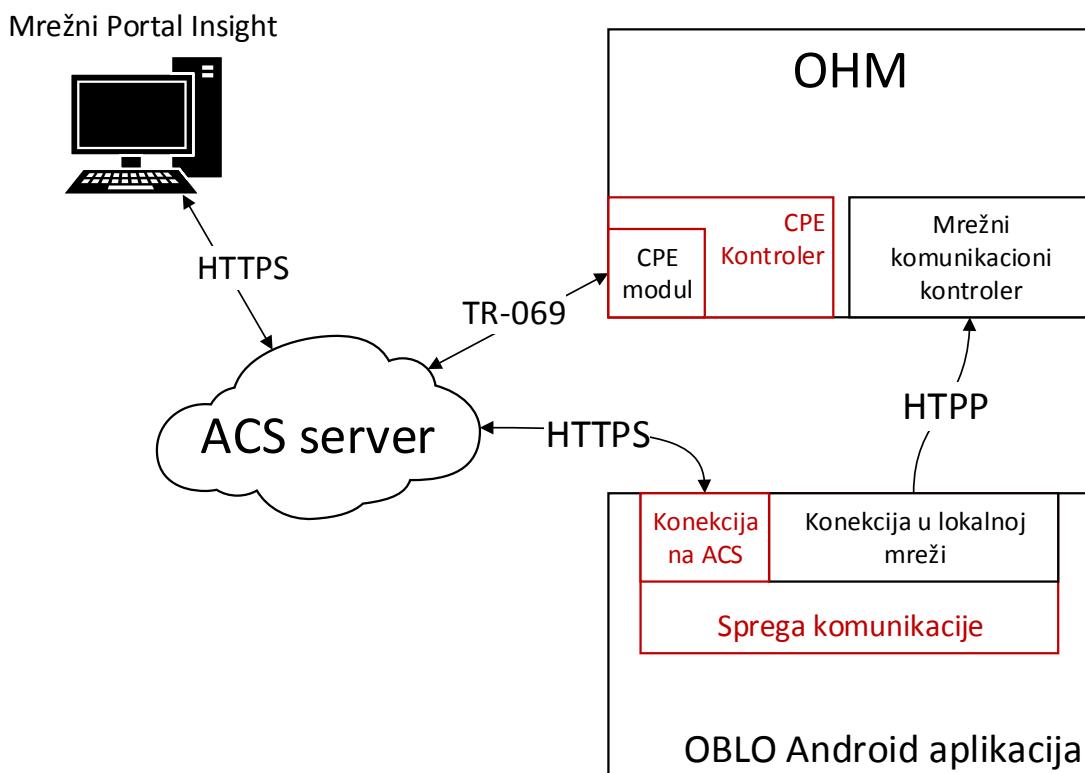
**Podešavanje** sadrži opcije za prijavljivanje na OBLO. Odabir OHM-a za kontrolu se može uraditi ručno unosom IP adrese ili koristeći SSDP mehanizam za automatsko otkrivanje uređaja na mreži. Prilikom prijavljivanja neophodno je dostaviti ime korisnika i njegovu šifru

Opciji **napredno podešavanje** se pristupa iz meniju podešavanja. Pristup opcijama naprednog podešavanja zahteva korišćenje odgovarajućih akreditiva. U toku režima rada naprednog podešavanja onemogućen je rad korisnika. Ovom opcijom klijentima su omogućene napredne funkcionalnosti dodavanja novih soba i dodavanje uređaja u odgovarajuću sobu kao i kontrola ZigBee mreže, njegovo otvaranje i zatvaranje.

Opcija glavnog menija **Bezbednost** pruža korisniku mogućnost da prati prisustvo ljudi u oblasti sistema. Ovo je omogućeno korišćenjem naprednih mehanizama za analizu radio polja ZigBee mreže.

### 3. Koncept rešenja

Arhitektura rešenja se sastoji od spajanja već postojećih rešenja OBLO sistema, ACS poslužitelja, CPE modula i OBLO Android aplikacije u jedinstven sistem. Cilj je da se postojeći oblo sistem proširi sa mogućnošću kontrole preko interneta bez narušavanja postojećih funkcionalnosti. Upotrebe protokola TR-069, koji spaja OHM sa ACS poslužiteljom, treba da omogući siguran prenos i mogućnost menjanja parametara. Blok dijagram arhitekture sistema proširenog sa podrškom za udaljenu kontrolu može se videti na Slika 3.1.



Slika 3.1 Arhitektura sistema

Crvenom bojom su označena proširenja. OHM je proširen novim CPE kontrolerom koji uz pomoć CPE modula komunicira sa ACS poslužiteljom. OBLO Android aplikacija je proširena sa mehanizmom za komunikaciju sa ACS-om spregom i mogućnošću biranja načina komunikacije kako bi se korisniku ostavila mogućnost da koristi postojeći mehanizam povezivanja sa OHM-om. OHM je proširen novim CPE kontrolerom koji vrši komunikaciju sa ACS poslužiteljem. OBLO

### 3.1 Proširenje OHM-a

Proširenje OHM-a se sastoji u pravljenju novog kontrolera. Kontroler je nazvan CPE Kontroler. Kontroler koristi postojeću CPE biblioteku za komunikaciju sa ACS poslužiteljem. Problem razlike između učestalosti periodične komunikacije sa ACS poslužiteljem i frekvencijom dobijanja merenja električne potrošnje je rešen usrednjavanjem vrednosti potrošnje na veći vremenski interval. Da bi se to postiglo vrednosti potrošnje od periodičnih izveštaja pametnih utičnica se moraju sačuvati u bazi. Baza koja je odabrana je SQLite baza kao pogodna za korišćenje u malim namenski uređajima kakav je OHM.

Definisan je novi model podataka koji CWMP protokol koristi. Uređaji koji su definisani u novom modelu su:

- Pametna utičnica
- Pametni prekidač

Model je definisan na način koji omogućuje lako proširenje isto novim uređajima u budućnosti. CPE podrška za uređaje je omogućena definisanjem novog modela podataka nazvanim TR-230 vodeći se TR-106 izveštajem.

### 3.2 Proširenje OBLO Android aplikacije

Komunikacija sa ACS serverom zahteva novu spregu na komunikacionom nivou kako se ne bi narušila postojeća funkcionalnost. Dodata sprega omogućava upotrebu oba načina komunikacije u istoj aplikaciji. Ova dva načina kontrole daje korisniku mogućnost komfornog upravljanja sa boljim odzivom u lokalnoj mreži i mogućnost udaljene kontrole sa konekcijom na ACS poslužitelj u jednoj aplikaciji.

## 4. Programsко rešenje

Analizom, kao što je opisano u predlogu rešenja, identifikovane su dve bitne celine na kojima su izvršena proširenja da bi se dobila tražena funkcija. Opisan je CPE kontroler i napravljeni novi model podataka koji omogućuje TR-106 izveštaj. U opisu proširenje OBLO Android aplikacije iznete su funkcije sprege i implementacija ka ACS serveru.

### 4.1 CPE Kontroler

Implementiran je u klasi *CCPEController* i u njemu se kontroliše CPE modul. U CPE modulu realizovana je kompletna komunikacija prema ACS serveru. CPE kontroler rukuje i podacima izveštaja potrošnje pametnih utičnica.

Klase koje se nalaze u CCPEController klasi:

- *CCPERunnableGarbageCollector*  
Ova klasa nasleđuje klasu Runnable koja služi pri pokretanju nove niti koja periodično briše zastarele periodične izveštaje. Zastareli izveštaji su oni koji su već poslati na ACS server.
- *CCPERunnableACSCCommandExecutions*  
Modul za rukovanje primljenim komandama. Modul se izvršava u zasebnoj niti.

Nasleđene funkcije koje svaki kontroler OHM-a mora da implementira:

- *bool Start()*

Funkcija se poziva prilikom pokretanja kontrolera. U njoj se pokreće CPE modul i postavljaju osnovni identifikacioni parametri OHM-a po kojima ACS raspozna pametnu kuću. Moduli *CCPERunnableGarbageCollector* i *CCPERunnableACSCCommandExecutions* se pokreću na samom kraju a pre toga se kontroler prijavljuje upravljaču kontrolera na sve buduće događaje OBLO sistema i postavljaju se funkcije koje će se pozivati iz CPE modula.

- *bool Stop()*  
radi se obrnute operacije od funkcije start. Stopira se CPE modul, kontroler se odjavljuje od svih događaja, zaustavlja se nit koja briše stare izveštaje potrošnje i nit obrade poruka od CPE modula.
- *bool ProcessEvent(OBJID \_ObjId, const string& \_szEvent, const Parameters\* \_pEventParams)*  
Funkcija obrađuje prispele poruke događaja od OBLO menadžera. Mogući tipovi događaja su:
  - Status OHM-a.  
Pokazuje da li je OHM u radnom režimu.
  - Periodični izveštaj o utrošenoj snazi. Vrednost se čuva u SQLite bazi
  - Promena grupe  
Promena se prosleđuje ka CPE modulu.
  - Promena Statusa uređaja  
Promena se prosleđuje ka CPE modulu.
  - Promena osvetljenja pametnog prekidača  
Promena se prosleđuje ka CPE modulu.
  - Dodavanje i brisanje novog uređaja u mrežu  
Pravi se ili briše Objekat, koji predstavlja uređaj, u modelu podataka CPE modula.

Ostale bitne funkcije *CCPEController* klase:

- *void wrapperPeriodicFunc(void\* pObject)*  
Ova funkcija se periodično poziva prilikom periodičnog slanja podataka na ACS poslužitelj. Pri pozivanju te funkcije stavljaju se nove vrednosti potrošnje uređaja.
- *tr230\_error\_t CCPEController::wrapperSetParameter(void\* \_pObject, const unsigned,int \_DeviceId, const char\* \_Value, const tr230\_deval\_data \_Type)*  
Funkcija se poziva pri promeni parametara sa ACS servera.

#### 4.1.1 Model podataka TR-230 i proširenje CPE modula

Model podataka TR-230 se sastoji od nekoliko parametara koji opisuju stanja uređaja u OBLO pametnoj kući. Parametri koji opisuju pametnu utičnicu i pametni prekidač su:

- *DeviceDescription* - Opis uređaja koji korisnim može da unese
- *DeviceModel* - Model uređaja, OBLO pametni prekidač ili OBLO pametna utičnica.

- *DeviceUID* - Identifikacija koju OBLO menadžer raspoznaće.
- *GroupID* - Index grupe
- *Mode* - Stanje uređaja, da li je upaljen ili ugašen.
- *DimmingLevel* - Ovaj parametar postoji samo za pametni prekidač, i predstavlja jačinu osvetljenja.
- *Consumption* - Ovaj objekat postoji samo za pametnu utičnicu, on sadrži tri parametra koji opisuju jedan vremenski interval i potrošnju električne energije nad tim vremenskim intervalom.
- *ReportingPeriod* - Parametar koji govori koja je veličina vremenskog intervala.

CPE modul je proširen sa funkcijama koje menjaju parametre novog TR-230 modela podataka. Svaki parametar je realizuje svoju funkciju. Modul sadrži funkcije za dodavanja i brisanja pametnih uređaja iz modela podataka, prijavljivanja i odjavljivanja parametara uređaja i brisanje ne aktivnih uređaja iz modela.

## 4.2 Modifikacija OBLO Android aplikacije

Modul za komunikaciju je proširen sa sprežnom klasom *IConnection* i vrednosti nabrojive liste mogućih povezivanja *ConnectionType* kao najbitnije parametre prepravljenog modula. Sve funkcije su preusmerene na IConnection spregu radi dalje obrade i slanje na zahtevani kanal komunikacije. U nastavku se mogu videti funkcije *IConnection* sprege koja svaka klasa komunikacije mora da implementira.

- *public void setHandler(Handler test);*  
Omogućava postavljanje objekta za rukovanje (eng. *Handler*) koji služi kao povratni kanal ka aktivnoj UI klasi.
- *public int connect(String ip, int port);*  
Uspostavljanje komunikacije ka serveru sa zadatom IP adresom.
- *public boolean closeConnection();*  
Zatvaranje komunikacije.
- *public boolean checkConnection();*  
Provera da li je komunikacija aktivna.
- *public void writeToSocket(String msg);*  
Slanje poruke.
- *public void uploadSetUp(Context context, String nameSetup);*  
Slanje cele postavke, slika soba i rasporeda soba, uređaja. Veza sa ACS poslužiteljom ne podržava ovu opciju
- *public void setDeviceRefresh(RoomView roomView);*

Funkcija koja se postavljaju uređaji u listu uređaja koje je potrebno osvežavati.

Koristi se samo u vezi sa ACS serverom.

Funkcije *IConnection* sprege su zadržane iz pređašnje aplikacije, implementacija funkcija za vezu u lokalnoj mreži je isto preuzeta iz pređašnje aplikacije uz manje izmene koda. Veza ka ACS poslužitelju ima komplikovaniju implementaciju jer zahteva prilagođavanje ACS protokolu komunikacije.

#### **4.2.1 Implementacija komunikacije sa ACS serverom.**

Komunikacija prema ACS poslužitelju se izvršava uz pomoć HTTPS protokola. Poruke se šalju u JSON formatu. Sam protokol je ukratko opisan u teorijskoj osnovi. Klase koje se nalaze u modulu za komunikaciju sa ACS poslužiteljem su *InsightACSConnection*, *InsightACSResponse* i *Defines* koji se nalaze u paketu *com.communicatio.insightacs*. Klasa *InsightACSConnection* implementira *IConnection* spregu i služi za slanje poruka. Obrada dobijene poruke se vrši u klasi *InsightACSResponse*.

##### **4.2.1.1 Klasa InsightACSConnection**

*InsightACSConnection* klasa sadrži u sebi četiri privatne klase:

- *SendLoopRunnable*

Klasa implementira slanje poruka HTTPS protokolom u posebnoj niti za slanje.

Poruke se dobavljaju iz reda čekanja

- *CommunicationQueueData*

Red čekanja koji se obrazuje pri slanju HTTPS poruke u *SendLoopRunnable* klasi koristi objekte ove klase.

- *CustomHttpClient*

Koristi se za HTTPS protokol radi ostvarenje zaštićene komunikacije.

- *RefreshDeviceRunnable*

Programska nit služi periodičnom osvežavanju stanja uređaja. Osvežavanje počinje pri ulasku u sobu. Uređaji se osvežavaju na pet sekundi i to samo uređaji kojima klijent ima pristup u toj sobi.

Važni parametri Klase *InsightACSConnection* su:

- *deviceIdPathIdMap* je mapa koja sadrži ID uređaja od ACS-a i ID uređaja od OBLO menadžera. Postoji i obrnuta mapa *reverseDeviceIdPathIdMap* koja uz pomoć identifikatora uređaja sa ACS servera daje ID uređaje sa OHM-a. Ova mapa se popunjava prilikom uspešne prijave korisnika i biranja CPE uređaja.
- *isDeviceStatusChangedMap*

Struktura podataka tipa mapa koja postoji radi optimizacije slanja promene parametara. Prilikom brzog slanja poruke za promenu parametara dolazi do zagušenja, ovaj problem je rešen sa dodavanjem oznake povećane za jedan svakoj uvezanoj poruci u red i proveravanjem pri uzimanju iz reda da li poruka promene parametara ima istu, najnoviju, oznaku. Poruka kojoj se ne poklapa oznake se odbacuje kako bi se ubrzalo slanje najnovijeg stanja uređaja.

- *private JSONObject cpeDesc*  
parametar potreban radi raspoznavanja odabranog CPE uređaja.

Važnije funkcije Klase *InsightACSConnection*:

- *public int connect(String url, int port)*  
Pokretanje niti koja vrši komunikaciju ka ACS serveru.
- *public boolean closeConnection()*  
Zaustavljanje niti za komunikaciju sa ACS poslužiteljem.
- *public void writeToSocket(String msg)*  
Funkcija implementira parsiranje poruke, koja je dobijena u JSON formatu. Parsiranjem dobija se informacija o tipu poruke koja se šalje kao i njegove parametre. Vrste poruke koje se raščlanjuju su:
  - Promena statusa uređaja. (Uključivanje/Isključivanje)
  - Promena prigušenja osvetljenja sijalice.
  - Prijavljanje korisnika na poslužitelj.
- *public void setACSPParam(int deviceId, String param, String value)*  
Omogućuje promenu parametara ACS poslužitelja. Parametar *DeviceId* je identifikacija iz OHM-a. Identifikator uređaja na ACS poslužitelju se razlikuje. Mapa *deviceIdPathIdMap* sadrži vezu identifikacija sa kojom ACS radi i identifikacije sa kojom OHM radi. Pravi se novi JSON objekat sa prosleđenim parametrima iz funkcije i uvezuje u red sa slanje poruka ka ACS serveru.
- *public void loginPost(String username, String password)*  
Uvezivanje u red poruke za prijavljivanje korisnika.
- *public void searchListPost(int offset, int limit, String criteria, String value)*  
Dobavljanje list CPE uređaja filtrirane za tip OBLO kuće.
- *public void getParamPost(JSONArray paramPaths, int msgType)*  
Dobavljanje vrednosti parametara uređaja.
- *public void getDevicesPost()*

Funkcija koja traži od ACS-a sve pametne utičnice u jednoj poruci i pametne prekidače u drugoj. U odgovoru ovih poruke dobijamo identifikacije uređaja u kući sa kojima ACS radi.

- `public void setDeviceRefresh(RoomView roomView)`

Preuzima sve aktivne uređaje iz sobe kojoj se pristupa da bi se mogli periodično osvežavati.

#### 4.2.1.2 Klasa InsightACSResponse

Važniji parametri klase InsightACSResponse:

- *boolean connected*

Privatna logička vrednost uspostavljene konekcije. Početno stanje ove promenjive je stanje tačno. Promenljiva prelazi u stanje netačno prilikom neuspelog pokušaja slanja poruke pomoću HTTPS protokola.

- *private int cpeStatus;*

Logička vrednost statusa CPE uređaja, označava da li je pametna kuća uspostavila komunikaciju sa ACS serverom.

Važnije funkcije InsightACSResponse klase:

- `public void httpCB(JSONObject strJsonRequest, String strJsonResponse, int responseType)`

Funkcija koja se poziva iz klase *InsightACSCollection*. U njoj se obrađuju odgovori poruka sa ACS servera na osnovu tipa poruke koja je poslata.

- `public void refreshDeviceParamsCB(JSONArray paramArray)`

funkcija koja parsira pristigle vrednosti parametara. Parsiranje se izvršava u više koraka. Prvi korak je grupisanje parametara po uređaju, a kasnije slanje poruke dalje u sistem obrade OBLO Android aplikacije. Poruka koja se dalje šalje je tipa događaj (Event) koja u vezi u lokalnoj mreži obaveštava OBLO Android aplikaciju o promeni stanja uređaja.

- `private void getTreeChildCB(JSONObject resp, String deviceName)`

Funkcija se poziva kao odgovor na poruku *getTreeChild* koja dostavlja sve pametne uređaje ili pametne prekidače u zavisnosti koja je poruka ranije poslata. Kada se dobiju svi uređaji pravi se nova poruka u kojoj se traži parametar identifikacije koju OHM koristi. Drugačije rečeno sa ovim odgovorom dobijamo oblo uređaje a šaljemo novu poruku da bi dobili njihove identifikacione indekse koje OBLO Android aplikacija može da koristi.

- `private void createIdMap(JSONArray params)`

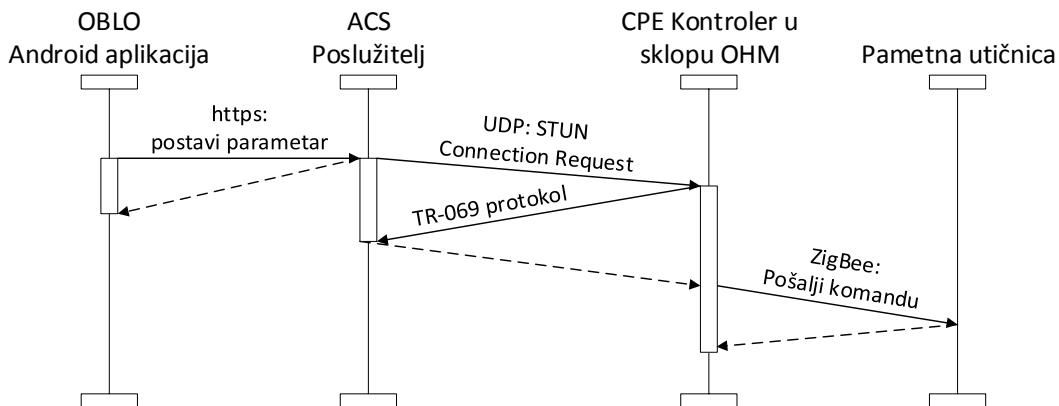
Primaju se parametri oblo uređaja koji predstavljaju jedinstveni identifikator uređaja. Jedinstveni identifikator se uparaje sa identifikatorom oblo uređaja koju ACS koristi i uvezuje u mapu deviceIdPathIdMap koja se nalazi u klasi InsightACSConnection.

- public void errorCB(String errMsg)

Obaveštavanje UI aktivnost da se desila greška u komunikaciji.

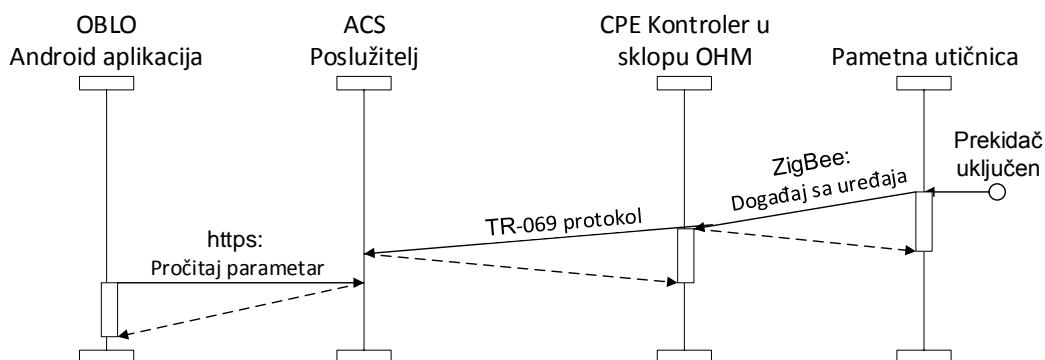
### 4.3 Primer komunikacije

Prikazani su primeri prosleđivanja poruka od Android aplikacije do pametnog prekidača i pametnog prekidača do Android aplikacije.



Slika 4.1 Dijagram sekvence komunikacije od OBLO Android aplikacije do pametne utičnice

Prvi primer, grafički prikazan kao dijagram sekvence na Slika 4.1, je promena parametra iz Android aplikacije. Android aplikacija putem HTTPS protokola šalje JSON poruku do ACS-a. ACS odgovara da li je uspešno primio parametar. JSON poruka za promenu parametara sadrži identifikaciju pametne kuće, vrednost i ime parametara, korisničko ime i tip poruke koje se šalje. pomoću STUN protokola inicira se komunikaciju sa CPE kontrolerom. Parametri se prosleđuju do centralnog menadžera preko TR069 protokola, u ovom primeru ta komunikacija je apstrahovana da bi se smanjila kompleksnost slike. Od centralnog menadžera se poruka ZigBee mrežom dobavlja do pametne utičnice koja izvršava zadatu akciju.



---

#### Slika 4.2 Dijagram sekvene komunikacije od pametne utičnice do OBLO Android aplikacije

Drugi primer, grafički prikazan kao dijagram sekvene na Slika 4.2, pokazuje način funkcionisanja sistema kada se sa prekidača pametne utičnice upali svetlo. Poruka se preko ZigBee mreže prosleđuje OHM-u. Kada OHM dobije poruku od pametnog prekidača CPE mrežna sprega bira šta treba da radi sa njim u zavisnosti od podešavanja za taj parametar. CPE mrežna sprega može da:

- Ne registruje promenu.
- Uspostava komunikacije se odmah inicira preko CPE modula.
- Registruje se promena ali se čeka periodična inicijacija CPE modula da bi se ta promena poslala ACS poslužitelju.

Podešavanja se mogu promeniti sa mrežnog portala. U ovom primeru odmah se inicira CWMP komunikacija i promena se šalje na ACS. Na kraju Insight aplikacija šalje JSON poruku ACS-u da bi dobio vrednost parametra, u odgovoru se dobija promenjen status parametar.

## 5. Rezultati

Realizacija rešenja dovodi do pametne kuće sa mogućnošću daljinskog upravljanja preko interneta.

OBLO Android aplikacija ne poseduje provere verifikacije uz kojeg bi ispitali i dodatni deo koji čini naše rešenje. Rešenje na Android aplikaciji je ispitano ručnim ispitivanjem. Ispitan je unos podataka prilikom povezivanja na ACS server, ispitano je ponašanju u slučaju zadavanja pogrešne adrese, korisničkog imena i korisničke šifre poremeti integritet aplikacije. Ispit je uspešno prošao.

Ispitivanje korektnosti komunikacionog modula OBLO Android aplikacije je obavljeno u pozadini posredstvom dnevnika Android aplikacije pomoću alata *Logcat*. *Logcat* omogućava grafičko prikazivanje dnevnika generisanog od strane Android aplikacije. Ispitivan je pravilan način slanja poruka prilikom redovne upotrebe aplikacije. Ispitni slučajevi se mogu videti u Tabeli 1.

Prijavljanje na ACS.	Ispit je uspešno prošao
Odabir CPE uređaja.	Ispit je uspešno prošao
Dobavljanje svih identifikacija OBLO pametnih uređaja nakon prijavljivanja.	Ispit je uspešno prošao
Kontrola pametnih uređaja promenom njihovih parametara.	Ispit je uspešno prošao
Osvežavanja parametara pametnih uređaja.	Ispit je uspešno prošao
Osvežavanje statusa CPE-a uređaja.	Ispit je uspešno prošao

Tabela 1 Ispitni slučajevi ispravnosti OBLO Android aplikacije

Ispitivanje ispravnosti CPE Kontrolera je isto obavljeno ručnim ispitivanjem, a rezultati su posmatrani sa strane ACS servera. Ispite slučajeve možete videti u Tabeli 1.

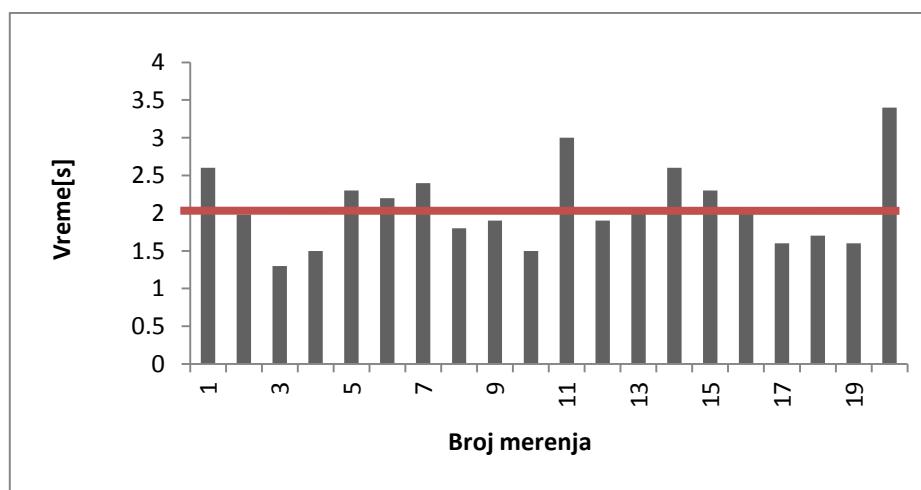
Dodavanje novih pametnih uređaja u OHM.	Ispit je uspešno prošao
Brisanje pametnih uređaja iz OHM.	Ispit je uspešno prošao
Promena parametara uređaja sa OBLO strane.	Ispit je uspešno prošao
Promena parametara uređaja sa ACS strane.	Ispit je uspešno prošao
Ispitivanje redosleda poslatih vrednosti električne potrošnje.	Ispit je uspešno prošao

Tabela 2 Ispitni slučajevi ispravnosti CPE Kontrolera

Dobijanjem rezultata ispitivanja možemo zaključiti da je dobijena funkcionalnost realizovanog rešenja sistema ispravna, ustvari da je uspešno izvršena daljinska kontrola preko interneta.

## 5.1 Vreme odziva

Ispitano je i vreme odziva sistema od mrežnog terminala ACS servera do krajnjeg uređaja OBLO-a. Na početku sistema na kojem je izvršeno testiranje nalazi se *internet* pretraživač. Prilikom klika na odgovarajuće dugme, uz pomoć mikrofona, snimamo zvuk testera miša. Pretraživač pristupa mrežnom portalu ACS-a koji se nalazi na javno dostupnom domenu. Sa javno dostupnom domenom ACS komunicira sa lokalnim računarom na kojem je pokrenut program pametne kuće. Lokalni računar pomoću hardverskog ključa šalje poruku ka pametnoj utičnici sa koje snimamo zvuk releja pametne utičnice koji se aktivira pri uspostavljanju strujnog kola.



Slika 5.1 Vremenski odziv sistema

Srednja vrednost odziva je oko dve sekunde. Odziv sistema zavisi od mnogo faktora jer se koristi dosta razlicitih protokola (HTTPS, TR-069, Zigbee) koje unose kašnjenje. Zavisi i od strukture mreže preko koje protokoli komuniciraju jer to može još usporiti komunikaciju. Ipak, odziv od dve sekunde je zadovoljavajući jer ovaj sistem nije namenjen za neposredni interaktivni rad nego za prikupljanje informacija i upravljanje na daljinu gde korisnik direktno ne vidi promene koje su posledica svojih akcija. Za upravljanje gde se korisnik nalazi u blizini pametne kuće koristi se direktna sprega sa Centralnim menadžerom preko lokalne mreže, što obezbeđuje značajno bolji odziv.

## 6. Zaključak

U ovom radu predstavljena je integracija OBLO pametne kuće u globalni sistem kontrole i nadzora preko interneta korišćenjem CWMP protokola. Predstavljen je OBLO menadžer kao centralni uređaj pametne kuće kroz koga je omogućena kontrola ostalih uređaja, CPE Kontroler u sklopu OBLO menadžera, OBLO Android aplikacija, protokol CWMP i njihova proširenja i veze koja su omogućila izradu ovo rešenja.

Prednost ovog rešenja se ogleda u korišćenju protokola CWMP koji je napravljen za kontrolu i nadzor krajnjih korisničkih uređaja u kući. Ovaj standardizovan protokol daje sigurnost i pouzdanost prenosa podataka i time obezbeđuje sigurnu kontrolu pametne kuće. Dodatak G TR-069 protokola nam daje mogućnost inicijacije komunikacije sa javne internet mreže time izbegavajući potrebu za stalnim uspostavljanjem veze od strane OBLO-a smanjujući protokol podataka [8]. TR-106 omogućava lako proširenje modela podataka sa novim tipovima uređaja.

Dalji pravci poboljšanja se sastoje od proširenja sistema sa mogućnosti kontrole ne samo pametnih uređaja vec i kontrolera OHM-a i ZigBee mreže. Time bi se dobila potpuna kontrola OBLO-a realizovana preko interneta. Drugi pravac bi bio analiza dobijenih podataka od OBLO kuće na ACS poslužitelju i pravljenje algoritama predikcije korišćenja pametnih uređaja ili međusobna komunikacija između dva ili više pametnih uređaja. Primer drugog načina bi bio samo paljenje svetala pri detekciji pokreta u sobi. Treći pravac bi bio pravljenje sačuvanih podešavanja uređaja koje bi korisnik mogao da podešava sa ACS poslužitelja. Primer trećeg načina bi bio podešavanje osvetljenja uređaja na predefinisanu sačuvanu vrednost.

## 7. Literatura

- [1] ZigBee <http://www.zigbee.org/>
- [2] OBLO pametna kuća <http://oblo.rt-rk.com/>
- [3]"Jedno rešenje sprežnog sistema za interaktivnu kontrolu pametnih kuća" Božić Milivoj, Bojan Mrazovac, Ištvan Pap, Mirko Vucelja, Miloš Janković; konferencija ETRAN, Zlatibor, jun 2012.
- [4] "A human detection method for residential smart energy systems based on Zigbee RSSI changes", Mrazovac, B.; Bjelica, M.Z.; Kukolj, D.; Todorovic, B.M.; Samardzija, D. Consumer Electronics, IEEE Transactions on , vol.58, no.3, pp.819,824, August 2012
- [5] "Multilayer approach to cost-efficient home automation", Bozic, Milivoj, Golan Giora, Mrazovac Bojan, Papp Istvan, and Bjelica Milan Z. , 2014 IEEE International Conference on Consumer Electronics – China, 04/2014, Shenzhen, China, (2014)
- [6] Insight <https://insight.rt-rk.com>
- [7] JSON Format podataka, <http://json.org/>
- [8] TR-069 Dopuna 4, Broadband Forum, jul 2011. - [http://www.broadband-forum.org/technical/download/TR-069\\_Amendment-4.pdf](http://www.broadband-forum.org/technical/download/TR-069_Amendment-4.pdf)
- [9] TR-106 Dopuna 6. Broadband Forum, jul 2011 [http://www.broadband-forum.org/technical/download/TR-106\\_Amendment-6.pdf](http://www.broadband-forum.org/technical/download/TR-106_Amendment-6.pdf)
- [10] SQLite - <https://sqlite.org/>
- [11] STUN protokol. <http://tools.ietf.org/pdf/rfc5389.pdf>
- [12] Android <http://www.android.com/>
- [13] Poco biblioteka <http://pocoprotocol.org/>
- [14] Insight <https://insight.rt-rk.com>
- [15] Android [http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))

- [16] TCP protokol <http://tools.ietf.org/html/rfc793>
- [17] IP protokol <http://www.ietf.org/rfc/rfc791.txt>
- [18] SSL protokol <http://tools.ietf.org/html/rfc6101>
- [19] TLS protokol <http://tools.ietf.org/html/rfc5246>
- [20] HTTP protokol <http://tools.ietf.org/html/rfc2616>
- [21] SOAP protokol <http://www.w3.org/TR/soap12-part1/>
- [22] XML <http://www.w3.org/TR/xml/>
- [23] RPC <http://tools.ietf.org/html/rfc5531>
- [24] NAT Traversal <http://tools.ietf.org/html/rfc5245>
- [25] TR-111 <http://www.broadband-forum.org/technical/download/TR-111.pdf>
- [26] REST [http://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)