

UNIVERZA V LJUBLJANI
Fakulteta za elektrotehniko

Mojmir Tuljak

DOLOČANJE HITROSTI WiMAX
REGISTRACIJSKEGA STREŽNIKA
DIPLOMSKO DELO UNIVERZITETNEGA ŠTUDIJA

Mentor: doc. dr. Boštjan Murovec
Ljubljana, junij 2009

Zahvala

Zahvaljujem se mentorju doc. dr. Boštjanu Murovcu za pomoč, podporo in potrpežljivost med pisanjem diplomskega dela.

Zahvaljujem se tudi staršem, ki sta me med študijem podpirala.

Nenazadnje se zahvaljujem še moji Anici, ki me je kljub nekoliko podaljšanem študiju spodbujala in podpirala.

Povzetek

V diplomski nalogi sta opisana izdelava programa za simulacijo registracije naročniških postaj v omrežje in določanje hitrosti registracije Telsiminega registracijskega strežnika.

Delo je razdeljeno v tri dele. Prvi del naloge obsega opis Telsiminih baznih postaj in naročniških postaj StarMAX ter strežnikov, ki sestavljajo Telsimin sistem za nadzor omrežja StarMAX NMS. Poseben poudarek je na opisu postopka registracije, delovanju registracijskega strežnika in protokolu SNMP (angl.: Simple Network Management Protocol) na katerem temelji registracija.

Drugi del diplomske naloge je namenjen realizaciji simulacijskega programa ter določanju hitrosti registracijskega strežnika pri izvajanju posamezne registracijske faze (avtorizacija in nastavljanje profila) in celotnega postopka registracije.

V tretjem delu smo na podlagi izmerjene hitrosti registracije podali oceno za velikost omrežja, ki ga lahko testirani sistem upravlja. Podani so tudi predlogi za pohitritev registracije naročniških postaj in našteje dodatne možnosti za uporabo simulacijskega programa.

Ključne besede: registracijski strežnik, sistem za nadzor omrežja, protokol SNMP, Java RMI, JDBC, WiMAX, podatkovne baze.

Abstract

This thesis describes the development of the simulation server and how it was applied to measure the performance of the Telsima's registration server.

The work is divided into three parts. First part is a description of Telsima's StarMAX base stations, subscriber station and servers that constitute the network management system. Special emphasis is placed on the description of the registration process, registration server and SNMP protocol used by a registration process.

In the second part we have described development of a simulation program and how it was used to determine the performance of the registration server. Performance was observed for server's atomic operations and for a whole registration process.

In the third part we have provided estimates based on the simulation results for the size of the network that can be maintained by a single registration server. We have also suggested possible changes to the registration process, which would increase the overall performance of the system. Additionally we have explored possibilities where usage of the developed simulation program can be beneficial.

Key words: registration server, network management system, SNMP protocol, Java RMI, JDBC, WiMAX, databases.

Kazalo

1	Uvod	1
2	StarMax NMS	5
2.1	Podatkovna baza	5
2.2	Odjemalnik.....	7
2.3	Aplikacijski strežnik.....	7
2.4	Podatkovni strežnik.....	8
2.5	Registracijski strežnik	8
2.6	Klic oddaljenih metod.....	8
2.7	Povezava s podatkovno bazo	10
2.8	Protokol SNMP za upravljanje omrežij.....	11
3	Bazna postaja in naročniška postaja StarMAX	18
3.1	Svetovna medsebojna obratovalnost mikrovalovnega dostopa	21
3.2	Razvoj standarda IEEE 802.16.....	21
4	Registracija naročniških postaj	24
4.1	Avtorizacija	24
4.2	Nastavitev profila	25
4.3	Življenjski cikel registracije.....	26
4.4	Transakcije	29
5	Opis simulacijskega programa.....	31
5.1	Poenostavitve	31
5.1.1	Poenostavitve pri modelu naročniške postaje	31
5.1.2	Poenostavitve pri modelu bazne postaje.....	33
5.2	SNMP agent	34
5.3	Časovnik.....	37
6	Določanje hitrosti registracijskega strežnika	38
6.1	Opis opreme	39
6.2	Hitrost avtorizacije	39
6.2.1	Analiza obdelave zahtev	45
6.3	Hitrost nastavitve profila	47
6.4	Hitrost registracije	52
7	Dimenzioniranje omrežja	55
8	Zaključek	57
8.1	Kako izboljšati hitrost registracije	57
8.2	Ostale možnosti uporabe	58

Kazalo slik

Slika 1.1: Arhitektura omrežja StarMAX.....	3
Slika 2.1: Primerjava ISO/OSI referenčnega modela z RMI modelom.....	10
Slika 2.2: Primerjava ISO/OSI referenčnega modela z JDBC za Oracle podatkovno bazo.....	11
Slika 2.3: Primerjava ISO/OSI referenčnega modela s SNMP.....	12
Slika 3.1: StarMAX 6100 ohišje	18
Slika 3.2: StarMAX 6400 ohišje	19
Slika 3.3: Postavitev vmesnikov na rezini StarMAX 6000 WiMAX.....	19
Slika 3.4: Naročniške postaje serije StarMAX 2100.....	21
Slika 3.5: Primerjava modela ISO/OSI s protokolom IEEE 802.16	23
Slika 4.1: Časovni potek avtorizacije naročniške postaje.....	25
Slika 4.2: Časovni potek nastavitve profila naročniške postaje.....	26
Slika 4.3: Življenski cikel registracije naročniške postaje	28
Slika 4.4: Transakcije.....	29
Slika 5.1: Primerjava realnega in simulacijskega okolja.....	33
Slika 6.1: Bazen niti	38
Slika 6.2 Število aktivnih niti in čakalna vrsta.....	41
Slika 6.3 Hitrost avtorizacij v odvisnosti od hitrosti pošiljanja zahtev.	42
Slika 6.4 Procesorska obremenitev simulacijskega programa	43
Slika 6.5 Procesorska obremenitev registracijskega strežnika pri avtorizaciji	44
Slika 6.6: Hitrost avtorizacij v odvisnosti od hitrosti pošiljanja zahtev – 5 niti.....	45
Slika 6.7 Obdelovalni, čakalni in celotni čas	46
Slika 6.8 Kumulativno število poslanih in obdelanih avtorizacij.....	47
Slika 6.9 Število aktivnih niti in čakalna vrsta.....	49
Slika 6.10 Hitrost nastavitve profila v odvisnosti od hitrosti pošiljanja zahtev.	49
Slika 6.11 Procesorska obremenitev registracijskega strežnika pri nastavljanju profila	50
Slika 6.12 Obdelovalni, čakalni in celoten čas	51
Slika 6.13 Kumulativno število obdelanih zahtev za nastavitev profila.....	52
Slika 6.14 Število aktivnih niti in čakalna vrsta.....	53
Slika 6.15 Hitrost posameznih faz v odvisnosti od hitrosti pošiljanja zahtev	54
Slika 6.16: Poraba virov registracijskega strežnika.....	54

Kazalo tabel

Tabela 1: Registracijska stanja s časovnimi omejitvami	28
Tabela 2: Pričakovane zahteve med prometno konico	55

1 Uvod

Podjetje Telsima d.o.o. je bilo ustanovljeno leta 2004 v Trzinu in je del Harris Stratex Networks s sedežem v Morrisville, Severna Karolina, ZDA. Telsima je eno izmed vodilnih razvijalcev in dobaviteljev opreme za širokopasovni brezžični dostop, ki temelji na tehnologiji WiMAX [1]. Kratica WiMAX pomeni svetovna medsebojna obratovalnost mikrovalovnega dostopa (angl.: worldwide interoperability for microwave access) in je skupek brezžičnih protokolov. Temelji na standardu IEEE 802.16, ki ga je razvil inštitut inženirjev elektrotehnike in elektronike (angl.: Institute of Electrical and Electronics Engineers - IEEE), oziroma točneje delovna skupina Wireless MAN 802.16.

Ime WiMAX je bilo določeno s strani neprofitne organizacije WiMAX Forum. Poglavitni cilj WiMAX Forum je pomoč industriji pri koordinaciji, interpretaciji in spremembah originalne, precej ohlapno napisane specifikacije standarda IEEE 802.16. WiMAX Forum vodi tudi program certificiranja, ki zagotavlja skladnost s standardom in interoperabilnost opreme različnih proizvajalcev. Proizvajalcem, ki opravijo teste, podelijo naziv »WiMAX Forum Certified« [2]. Nekateri proizvajalci opremo označujejo kot "WiMAX-ready", "WiMAX-compliant" ali "pre-WiMAX", kar ni enako in ne zagotavlja interoperabilnosti med napravami, ki so certificirane s strani WiMAX Forum. V primeru bolj znane tehnologije Wi-Fi (IEEE 802.11) je za skladnost skrbela organizacija Wi-Fi Alliance. Prav ustanovitev te organizacije je bil ključni korak za množično uporabo tehnologije Wi-Fi.

Na območju Republike Slovenije ima pooblastila za dodeljevanje radijskih frekvenc agencija za pošto in elektronske komunikacije (APEK). Le-ta je oktobra 2006 za obdobje 10 let podelila dve WiMAX licenci oziroma odločbi o dodelitvi radijskih frekvenc na frekvenčnem območju od 3410 MHz do 3600 MHz za opravljanje storitev fiksne brezžičnega širokopasovnega dostopa. Licenci sta bili dodeljeni družbi Telekom Slovenije d. d. in Tok telekomunikacije d.o.o.

Telekom Slovenije je marca 2008 začel tudi uradno s trženjem WiMAX storitev. Ob koncu triletnega obdobja si nadejajo imeti 98,4-odstotno pokritost celotnega

prebivalstva, od tega 55,6-odstotno pokritost prebivalstva na podežlju, kar bo za 38,4 oziroma 22,3 odstotka presegalo razpisne zahteve agencije za pošto in elektronske komunikacije, ki je podelila koncesijo [3]. Za podeželje se upoštevajo občine, v katerih živi manj kot 50 prebivalcev/km². Šteje se, da je ponudnik ponudil storitve širokopasovnega brezžičnega omrežja na območju občine, kadar storitve ponuja na več kot 75 odstotkih površine občine, pri čemer se površina, na kateri živi manj kot 5 prebivalcev/km² ne upošteva [4].

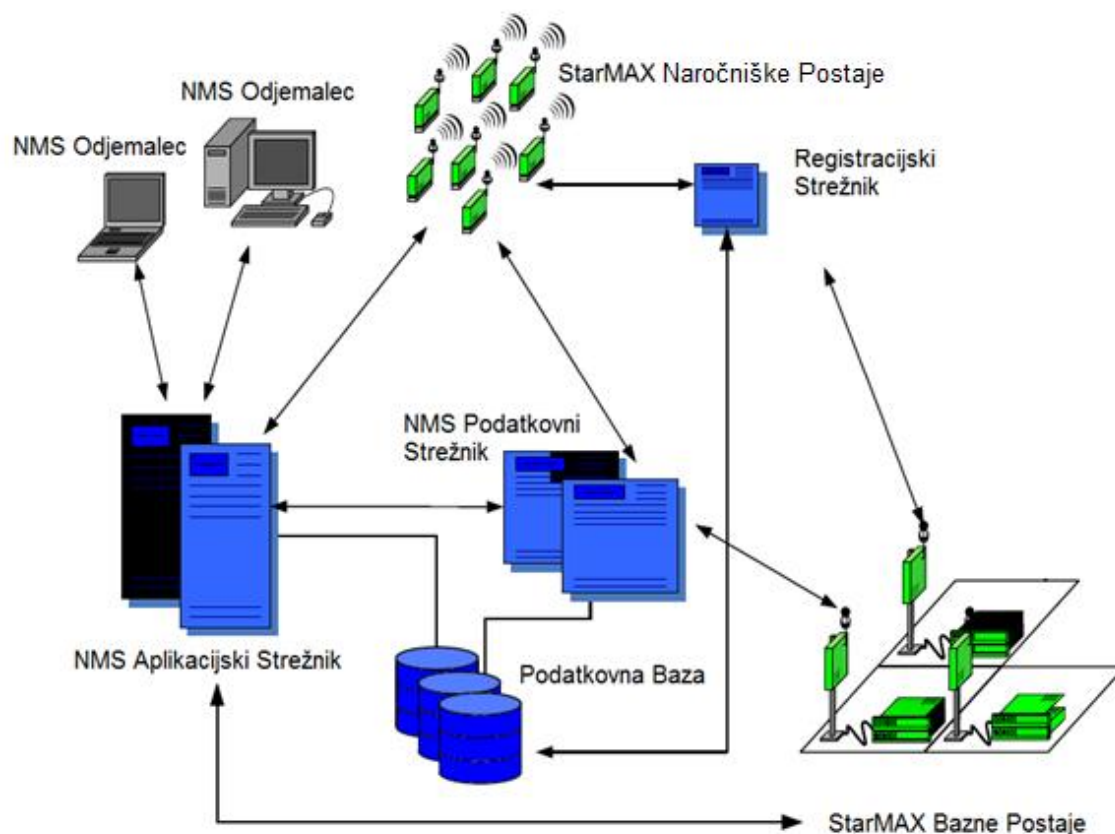
WiMAX tehnologija ima zaradi svojih konkurenčnih lastnosti možnost uveljavitve tako na trgih razvitih držav kot držav v razvoju. V razvitih državah ima WiMAX kot storitev za zagotavljanje širokopasovnega dostopa veliko konkurenco pri optičnem, xDSL in kablenskem dostopu do interneta, zato je pričakovati, da se bo WiMAX najprej uveljavil v področjih, kjer ostali načini širokopasovnega dostopa niso mogoči, to je na podežlju in v bolj oddaljenih krajih, šele kasneje kot mobilni širokopasovni dostop. Za razvijajoče se države je tehnologija zanimiva, ker pri njih zemeljska telekomunikacijska infrastruktura še ni razvita. Zanje WiMAX predstavlja hitro in poceni rešitev za dostop do telekomunikacijskih storitev.

V Telsimi razvijamo celostno rešitev za omrežja, ki temeljijo na tehnologiji WiMAX. To pomeni, da razvoj obsega bazne postaje (angl.: base stations) in naročniške postaje (angl.: subscriber stations), ter aplikacije za upravljanje omrežja. Sistem za upravljanje omrežja (angl.: network management sistem - NMS) se imenuje StarMAX NMS in je sestavljen iz več aplikacij:

- odjemalnik,
- aplikacijski strežnik,
- podatkovni strežnik,
- registracijski strežnik in
- podatkovna baza.

Arhitektura omrežja StarMAX je predstavljena na sliki 1.1, kjer je prikazana povezava med omrežnimi elementi in NMS. Centralna točka sistema je podatkovna baza. V njej so shranjene vse pomembne nastavitve za delovanje omrežja. Neposredno lahko z omrežnimi elementi in podatkovno bazo komunicirajo le aplikacijski, podatkovni in registracijski strežnik. Če operater želi upravljati omrežje, se mora najprej z

odjemalnikom povezati na aplikacijski strežnik, ki v njegovem imenu izvršuje ukaze. V osnovi so odjemalec, aplikacijski strežnik in podatkovni strežnik namenjeni upravljanju in nadzoru omrežja, medtem ko ima registracijski strežnik ključno vlogo pri delovanju omrežja in sodeluje pri registraciji naročniških postaj v omrežje.



Slika 1.1: Arhitektura omrežja StarMAX

Registracijski strežnik vsako naročniško postajo pred vstopom v omrežje avtorizira in ji nastavi uporabniški profil. Uporabniški profil vsebuje nastavitve za storitve (internet dostop, telefonija...), ki jih je naročnik upravičen. Posledično postane eno od vprašanj pri postavitvi omrežij, koliko naročniških postaj lahko registracijski strežnik v nekem časovnem obdobju registrira v omrežje. Če poznamo odgovor, lahko ocenimo, kako veliko omrežje lahko zgradimo, oziroma ali trenutno postavljeni sistem omogoča dodatno razširitev omrežja. Slednji vprašanji sta povod za nastanek te diplomske naloge.

Hitrost registracije je mogoče določiti na dva načina:

- z uporabo realnega sistema ali
- z uporabo simulacije.

Odločili smo se, da bomo hitrost registracije določili s simulacijo. Za realni sistem se nismo odločili zaradi omejenega števila naročniških postaj in baznih postaj, ki jih imamo na razpolago pri izvedbi študije. Predvsem nas zanima delovanje strežnika pod večjimi obremenitvami, za kar potrebujemo nerazpoložljivo veliko naročniških postaj. S simulacijo je bistveno lažje simulirati večje število postaj in tako dobiti bolj uporabne rezultate, zavedamo pa se, da simulacija nikoli ne more v popolnosti zajeti realne situacije.

Realizirani simulacijski program lahko brez večjih posegov vključimo v že obstoječo postavitev sistema NMS. Delovanja registracijskega strežnika ni bilo potrebno prilagajati. V sistem smo vnesli le nastavitve za naročniške postaje in bazne postaje, s katerimi smo izvajali simulacije.

Diplomska naloga je ločena v tri tematske sklope. Prvi del je namenjen arhitekturi omrežja. Podrobneje so opisane aplikacije, ki sestavljajo NMS, omrežne naprave (bazne in naročniške postaje) ter povezave med njimi (komunikacijski protokoli). Posebno poglavje je namenjeno protokolu SNMP (angl.: Simple Network Management Protokol), saj se postopek registracije izvaja preko njega.

Drugi del se osredotoča na realizacijo simulacijskega programa ter določanje hitrosti posameznih registracijskih faz in celotnega postopka registracije. Predstavljeni so potek registracije in ključni deli simulacijskega programa.

V zadnjem delu podajamo predloge za izboljšanje hitrosti registracijskega strežnika in dodatne možnosti uporabe simulacijskega programa.

2 StarMax NMS

StarMax NMS je sistem za nadzor in upravljanje omrežja WiMAX. Vse NMS aplikacije razen podatkovne baze so napisane v programskem jeziku Java, ki ga je razvil Sun Microsystems [5, 6]. Java je bila prvič javnosti predstavljena 23. maja 1995. V tem času je bila zgolj programski jezik za popestritev spletnih strani, na njej pa še niso temeljili resni projekti, poleg tega je bila počasna.

Marca 1997 je Sun predstavil novo različico javanskega razvojnega kompleta z oznako 1.1. Java je do tega trenutka doživela precejšnje spremembe, podvojilo se je število razredov, dodali so standardno knjižnico za delo z zbirkami podatkov, prišlo pa je tudi do sprememb v samem jeziku (nov model obravnave dogodkov, nove rezervirane besede). Sedaj je Java moderen programski jezik, opremljen z bogato standardno knjižnico programskih struktur in funkcij (za delo z datotekami, dostop do podatkovnih baz, mrežnimi povezavami, hkratno izvajanje več programov, grafične aplikacije, aplikacije, vgrajene v brskalnike, itn.). NMS za delovanje uporablja Java verzijo 1.5 ali višjo različico.

2.1 Podatkovna baza

Podatkovna baza je zbirka med seboj pomensko povezanih podatkov, ki so shranjeni v računalniškem sistemu. Dostop do njih je centraliziran in omogočen s pomočjo sistema za upravljanje podatkovne baze (angl.: database management system - DBMS) [7]. DBMS nudi vse osnovne storitve, ki so potrebne za urejanje in vzdrževanje podatkovne baze.

Podatkovna baza je ločena na metapodatkovno bazo in fizično podatkovno bazo. Metapodatkovna baza shranjuje opise fizičnih podatkov, in sicer, kje se le-ti nahajajo v pomnilniku, kaj pomenijo, ter katerim uporabnikom so dostopni. V fizični podatkovni bazi so shranjene dejanske vrednosti podatkovnih elementov.

Na kakšen način so podatki shranjeni in organizirani v podatkovni bazi je odvisno od uporabljenega tipa podatkovne baze. Poznamo več tipov podatkovnih zbirk [8], kot so:

- hierarhične,
- mrežne,
- relacijske in
- objektne.

Danes so v uporabi predvsem podatkovne baze, ki temeljijo na relacijskem modelu. Mrežni in hierarhični tip praktično nista več v uporabi. Objektni tip pa se uporablja samo za posebne namene.

Relacijska podatkovna baza temelji na preprostih dvodimenzionalnih tabelah. Vrstice tabel predstavljajo posamezne zapise, stolpci pa označujejo polja (lastnosti) teh zapisov. Vsak zapis v tabeli vsebuje informacijo o določenem predmetu ali pojmu, o katerem zbiramo podatke.

Posamezne tabele so med seboj povezane z relacijami, ki vzpostavljajo odnose med zapisi v tabelah. Relacija deluje tako, da povezuje zapise v dveh ali več tabelah, ki imajo ujemajoče se podatke v poljih s ključem. Običajno so ti ujemajoči se podatki v poljih z enakim imenom stolpca v obeh tabelah. V večini primerov sta ti ujemajoči se polji primarni ključ v prvi tabeli in tuj ključ v drugi tabeli. Tuji ključ vsebuje eno ali več polj v tabeli (stolpcev), ki se sklicujejo na polje ali polja primarnega ključa v drugi tabeli. Tuji ključ kaže na relacije med tabelami. Poznamo več vrst relacij, med drugim »ena proti mnogo«, »mного proti mnogo« in »ena proti ena«.

O podatkih poizvedujemo s strukturiranim povpraševalnim jezikom (angl.: Structured Query Language - SQL).

Objektni podatkovni tip temelji na objektih in metodah. V objektni podatkovni bazi so združeni elementi objektno-orientiranih programskih jezikov s funkcionalnostjo podatkovnih baz. Te baze so razširitev objektno-orientiranih programskih jezikov in nudijo veliko več, kot samo shranjevanje podatkov (objektov), saj podpirajo enkapsulacijo, dedovanje in abstraktne podatkovne tipe.

Z orodji, kot je Hibernate [9], je možno relacijsko podatkovni bazo navidezno uporabljati, kot bi bila objektna podatkovna baza. Hibernate je vmesna plast med

podatkovno bazo in objektnim programom, ki preslikuje iz Java razredov v tabele podatkovne baze in obratno. Na ta način so združene prednosti objektnih in relacijskih podatkovnih modelov.

NMS uporablja relacijsko podatkovno bazo MySQL [10] ali Oracle [11]. MySQL je priljubljena odprtokodna relacijska podatkovna baza. Razvija in distribuira jo Sun Microsystems. Podatkovna baza je na razpolago pod dvema različnima licenčnima sporazumoma:

- brezplačno pod licenco GPL [12] ali
- plačljivo v obliki vpisnine v MySQL mrežo za poslovne aplikacije.

MySQL teče na vseh razširjenih operacijskih sistemih, vključno z Linux, Unix in MS Windows. Je večnitni sistem in ima aplikacijske vmesnike za številne programske jezike, vključno s C/C++, Eiffel, Java, Perl, PHP, Python in TCL.

Oracle Database (ali Oracle) je relacijske podatkovna baza, ki jo razvija in trži Oracle Corporation. Podobno kot MySQL tudi Oracle teče na večini razširjenih operacijskih sistemov.

Kljub hitremu razvoju in pridobivanju novih funkcionalnosti MySQL še vedno zaostaja za Oracleom v na področju zmogljivosti, razširljivosti, varnosti in zanesljivosti. Ključna prednost MySQL baze pred Oracle bazo je v ceni.

2.2 Odjemalnik

Odjemalnik je aplikacija z grafičnim vmesnikom, ki jo operater uporablja za upravljanje omrežja. Le—ta operaterju omogoča enostaven in uporabniško prijazen dostop do sistema in z njim povezana administrativna opravila.

2.3 Aplikacijski strežnik

Operater se z odjemalnikom poveže na aplikacijski strežnik, preko katerega izvaja ukaze na omrežnih napravah ali bazi. Tak sistem omogoča operaterju, da upravlja

omrežje tudi, ko nima neposrednega dostopa do omrežnih naprav, če le ima dostop do aplikacijskega strežnika.

Aplikacijski strežnik vsebuje:

- logiko za nastavljanje SNMP agentov na baznih in naročniških postajah,
- logiko za delo s podatkovno bazo.

2.4 Podatkovni strežnik

Poleg odjemalcev se na aplikacijski strežnik poveže tudi podatkovni strežnik, ki je zadolžen za zbiranje alarmov in različnih statističnih podatkov. Bazne postaje pošiljajo alarme na podatkovni strežnik, kateri jih obdela in shrani v podatkovno bazo. Statistični podatki, kot je npr. kakovost radijskega signala ali količina prenesenih podatkov, se časovno zbirajo iz omrežnih naprav in shranjujejo v bazo, kjer so na voljo za kasnejšo statistično obdelavo.

Vlogi podatkovnega strežnika:

- sprejemanje, obdelava in shranjevanje alarmov,
- periodično zbiranje statističnih podatkov.

2.5 Registracijski strežnik

Registracijski strežnik je zadolžen za registracijo naročniških postaj v omrežje. V sistemu lahko deluje več registracijskih strežnikov hkrati, kar izboljša storilnost in poveča zanesljivost storitve. Za porazdelitev obremenitve med strežniki uporabljamo razdeljevalnik obremenitve (angl.: load balacer) [13].

2.6 Klic oddaljenih metod

NMS je porazdeljen sistem, ki se izvaja na več računalnikih. Za osnovno komunikacijo med aplikacijami bi lahko uporabljali vtičnice (angl.: socket), ki so prilagodljive in zadostujejo za splošno komunikacijo.

Vendar je z uporabo vtičnic potrebno na odjemalniku in strežniku izdelati aplikacijsko plast, ki kodira in odkodira izmenjevana sporočila. Razvoj tega nivoja je težak in zamuden.

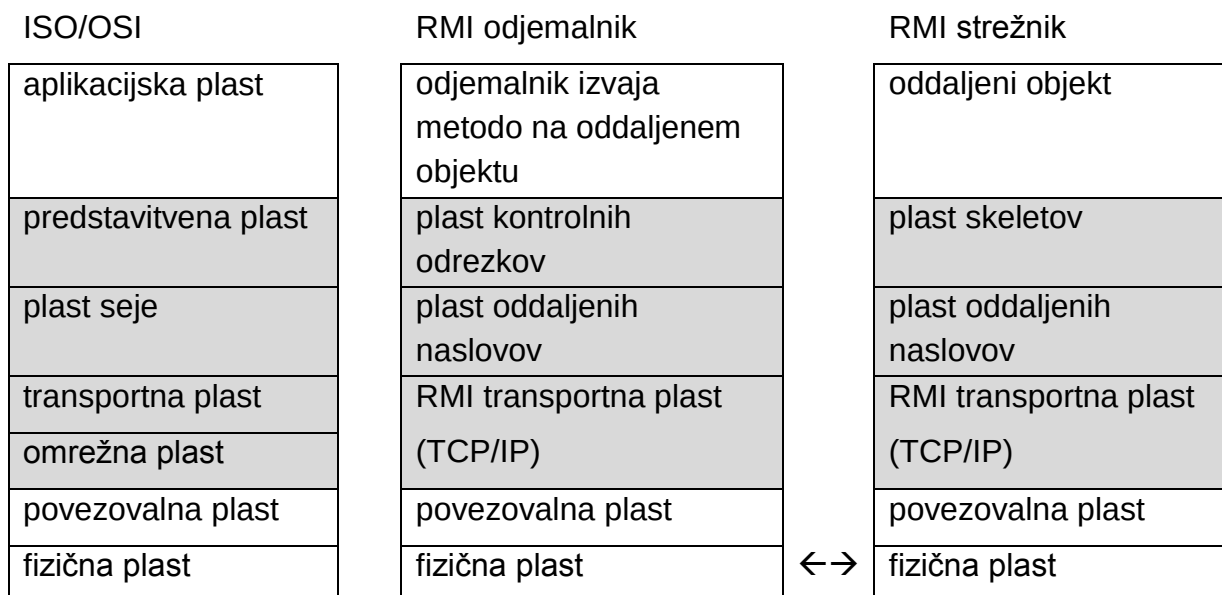
Alternativa vtičnicam je klic oddaljenih procedur (angl.: Remote Procedure Call - RPC), ki abstrahira komunikacijski vmesnik na nivo klica procedure. Žal se RPC ne vključuje dobro v objektni način programiranja. Posebej za potrebe semantike javanskega objektnega jezika je bil razvit model porazdeljenih objektov, imenovan klic oddaljenih metod (angl.: Remote Method Invocation - RMI [14]). RMI je zbirka protokolov, ki omogočajo, da javanski objekti komunicirajo z drugimi oddaljenimi javanskimi objekti.

RMI model je sestavljen iz treh plasti (slika 2.1):

- plast kontrolnih odrezkov in skeletov,
- plast oddaljenih naslovov in
- transportna plast.

Plast kontrolnih odrezkov skrbi za pretvarjanje podatkov ali objektov v podatkovni tok in obratno. Plast oddaljenih naslovov skrbi za ustvarjanje in upravljanje z naslovi oddaljenih objektov. Transportna plast ustvarja povezave med javanskimi virtualnimi napravami.

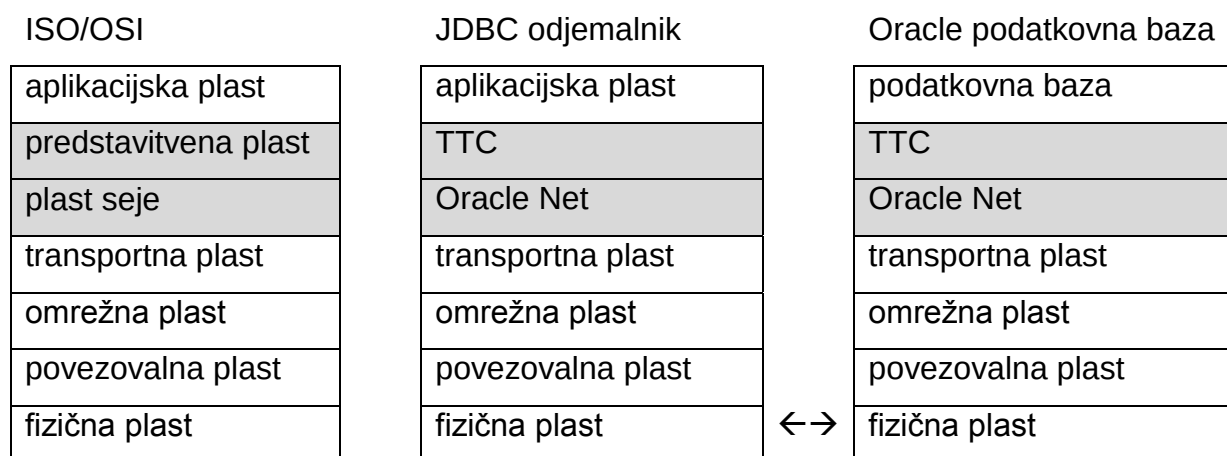
Slika 2.1 prikazuje primerjavo RMI modela in ISO/OSI referenčnega modela. Plast kontrolnih odrezkov spada v predstavitevno plast. Plast oddaljenih naslovov spada v sejno plast. Transportno plast utelešata protokola TCP ali UDP; pogosteje je uporabljen TCP. Omrežno plast uteleša protokol IP. Povezovalni in fizični nivo sta poljubna.



Slika 2.1: Primerjava ISO/OSI referenčnega modela z RMI modelom

2.7 Povezava s podatkovno bazo

Komunikacija med aplikacijami in podatkovno bazo je izvedena z aplikacijskim programskim vmesnikom JDBC [15] (angl.: Java database connectivity). JDBC definira zbirko javanskih vmesnikov in razredov, ki omogočajo javanskim programom, da se povežejo s poljubno podatkovno bazo, ki je podprta z JDBC, izvajajo poizvedbe v SQL jeziku in procesirajo odgovore. JDBC za Oracle podatkovno bazo v ISO/OSI modelu pokriva predstavitveni in sejni novo, kar je prikazano na sliki 2.2 [16].



Slika 2.2: Primerjava ISO/OSI referenčnega modela z JDBC za Oracle podatkovno bazo.

Predstavitveni nivo, ki ga uporabljata odjemalnik in strežnik, je imenovan Dve-skupni-nalogi (angl.: two-task common - TTC). TTC zagotavlja preslikavo med različnimi nabori znakov in podatkovnimi tipi, če se ti med odjemalnikom in strežnikom ne ujemajo.

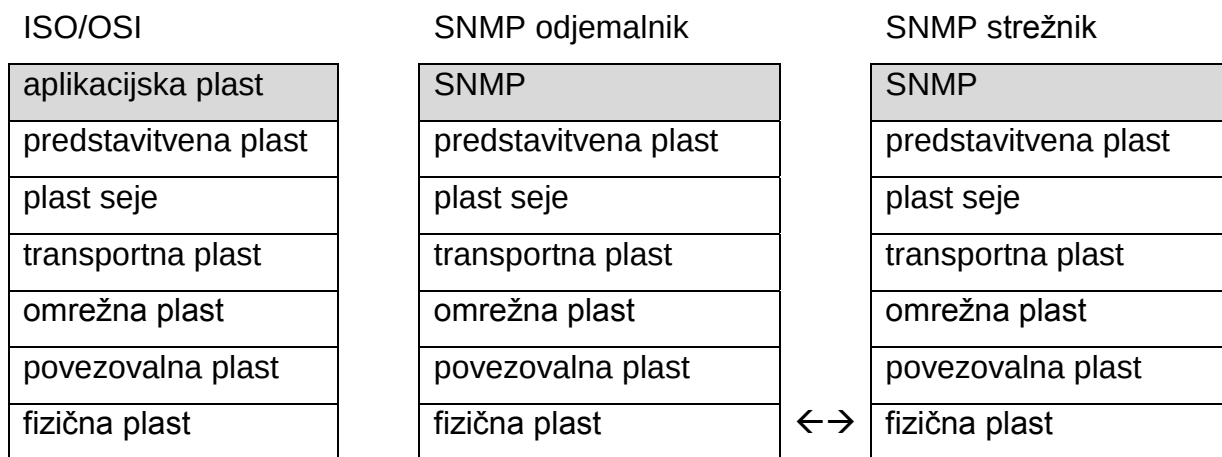
Sejni nivo predstavlja nivo »Oracle Net«, ki skrbi za vzpostavljanje in vzdrževanje povezave med odjemalnikom in podatkovno bazo, kot tudi za izmenjavo sporočil med njima.

2.8 Protokol SNMP za upravljanje omrežij

Za komunikacijo med omrežnimi elementi in NMS je uporabljen protokol SNMP (angl.: Simple Network Management Protocol) [17]. SNMP je predvsem uporabljen za nadzor in upravljanje sodobnih omrežij. Implementacija SNMP zahteva malo nastavitvev, kar ga naredi enostavnega. Proizvajalci strojne ali programske opreme enostavno vključijo SNMP agente v svoje izdelke in tako dodajo upravljalno funkcionalnost. SNMP agent je programska komponenta, ki teče na upravljanem sistemu in posreduje podatke preko SNMP do upravljalnega sistema. Enostavnost protokola je povzročila veliko popularnost podpore SNMP v omrežnih napravah.

SNMP je rešil še eno težavo, in sicer upravljanje naprav različnih proizvajalcev. Vsak proizvajalec je predhodno vgrajeval lastne rešitve za upravljanje, s prihodom SNMP

pa se je to poenotilo. SNMP sestavlja niz standardov za upravljanje omrežja, vključno s protokolom za aplikacijski nivo, sheme podatkovne baze in nabora podatkovnih objektov. Slika 2.3 prikazuje ISO/OSI model s SNMP protokolom na aplikacijskem nivoju.



Slika 2.3: Primerjava ISO/OSI referenčnega modela s SNMP

Nadzorni sistem iz nadzorovanih naprav pridobiva podatke z uporabo SNMP operacij GET, GETNEXT ali GETBULK, ali pa agent sam pošilja podatke, brez zahteve nadzornega sistema, z uporabo operacij TRAP ali INFORM.

Z operacijo GET dobimo vse spremenljivke, ki smo jih zahtevali. GETNEXT vrne prvo naslednjo spremenljivko, za tisto, ki smo jo zahtevali. Z GETBULK pa dobimo več zaporednih spremenljivk. TRAP in INFORM sta uporabljena za obveščanje upravljalnega sistema o določenih dogodkih. Edina razlika med njima je, da mora pri operaciji INFORM sprejemnik prejetje potrditi.

Nadzorni sistemi pošljejo tudi konfiguracijske popravke ali kontrolne zahteve z operacijo SET. Konfiguracijske in kontrolne operacije se uporabijo, kadar je potrebno izvesti spremembe omrežne infrastrukture. Za odgovor na vse operacije razen TRAP se uporablja operacija RESPONSE. Spremenljivke v okviru protokola SNMP so hierarhično organizirane. Organiziranost spremenljivk skupaj z meta podatki (npr. tip in opis) opisuje baza upravljalnih informacij (angl.: Management Information Base - MIB).

Baza upravljalskih informacij

Protokol SNMP ne definira, katere informacije (spremenljivke) naj nadzorovani sistem nudi navzven. Raje uporablja razširjeno zasnovo, kjer so razpoložljive informacije določene z bazami MIB. Te opisujejo strukturo nadzorovanih podatkov podsistema naprave, uporabljajo hierarhične imenske prostore (angl.: namespaces), ki vsebujejo označevalnike objekta (angl.: Object Identifier - OID). Vsak OID določa spremenljivko, ki jo lahko beremo ali nastavljamo preko protokola SNMP. Označevalnik objekta OID unikatno definira nadzorovan objekt v hierarhiji MIB. Baze MIB uporabljajo zapis, imenovan ASN.1 – abstraktna sintaksa zapisa ena (angl.: abstract syntax notation one).

Hierarhijo baz MIB si lahko predstavljamo kot drevesno strukturo, pri čemer koren nima imena, podnivoji pa so dodeljeni različnim organizacijam. Zgornji označevalniki OID baz MIB pripadajo različnim organizacijam, ki določajo standarde, na nižjem nivoju pa so označevalniki OID dodeljeni združenim (in manjšim) organizacijam.

Takšna struktura omogoča nadzor nad vsemi plastmi modela OSI, sam nadzor pa se lahko razširja tudi v aplikacije, kot so podatkovne baze ali elektronska pošta, saj se lahko baze MIB definirajo za najrazličnejša področja informacij in operacij. Vsaka nadzorovana naprava ima več nadzorovanih objektov (t.i. objekt MIB ali samo MIB). Nadzorovan objekt sestoji iz ene ali več instanc objekta, identificiranega z označevalnikom OID. Instanci objekta pravimo tudi spremenljivka. Obstajata dva tipa nadzorovanih objektov:

- skalarni objekti, ki definirajo eno spremenljivko,
- tabelarni objekti, ki definirajo več sorodnih spremenljivk, združenih v tabelo MIB.

Abstraktna sintaksa zapisa ena (ASN.1)

V telekomunikacijskih in računalniških omrežjih je zapis ASN.1 standarden in prilagodljiv zapis, ki opisuje strukturo podatkov za predstavitev, kodiranje, prenos, in dekodiranje podatkov. Nudi množico formalnih pravil za opis struktur objektov, ki so neodvisni od strojno-specifičnih kodirnih tehnik.

Zapis ASN.1 je skupen standard organizacij ISO in ITU-T, prvotno definiran leta 1984, kot del nekega drugega standarda. Kasneje, leta 1988, je zaradi velike uporabnosti prerasel svoj standard. Leta 1995 pa je bil še krepko posodobljen. Prilagojena podmnožica standarda ASN.1, tj. struktura upravljalskih informacij (angl.: Structure of Management Information - SMI) je navedena v protokolu SNMP za definiranje skupin povezanih objektov MIB, drugače imenovanih tudi moduli MIB.

Osnovne komponente protokola SNMP

Omrežje, nadzorovano s protokolom SNMP, sestavljajo tri ključne komponente:

1. nadzorovane naprave,
2. agenti in
3. sistemi za upravljanje omrežja.

Nadzorovana naprava je mrežno vozlišče, ki vsebuje SNMP agenta in se nahaja v nadzorovanem omrežju. Nadzorne naprave zbirajo in hranijo nadzorovane informacije ter jih dajejo na voljo sistemom NMS, ki uporabljajo protokol SNMP.

Agent je mrežno-upraviteljski programski modul, ki se nahaja v nadzorovani napravi in pozna lokalne nadzorovane informacije, ki jih je zmožen pretvoriti v obliko, združljivo s protokolom SNMP.

Sistem NMS izvaja programe, ki spremljajo in nadzirajo nadzorovane naprave. V vsakem nadzorovanem omrežju je lahko eden ali več sistemov NMS.

Podatkovni tipi SMI v verziji SNMPv1

Verzija SNMPv1 določa za strukturo SMI uporabo številnih podatkovnih tipov, ki jih delimo v dve skupini:

1. preprosti podatkovni tipi in
2. aplikacijski podatkovni tipi.

V strukturi SMI verzije SNMPv1 so definirani trije preprosti podatkovni tipi:

1. celoštevilčni tip podatkov v območju med -2^{31} do $2^{31}-1$,
2. oktetni niz znakov – to so urejena zaporedja od 0 do 65535 oktetov,

3. označevalniki OID – dobimo jih iz skupka vseh OID, glede na pravila, definirana v zapisu ASN.1.

V strukturi SMI verzije SNMPv1 obstaja sedem aplikacijskih podatkovnih tipov:

1. omrežni naslovi – predstavljajo naslov iz določene družine protokola. Verzija SNMPv1 podpira samo 32 bitne naslove IP,
2. števci – vsebujejo ne-negativne številčne vrednosti, ki se povečujejo dokler ne dosežejo največje dovoljene vrednosti, nato se vrnejo na nič. V verziji SNMPv1 poznamo samo 32 bitne števce,
3. merilniki (angl.: gauges) – vsebujejo ne-negativne številčne vrednosti, ki se lahko povečujejo ali zmanjšujejo med vnaprej predpisano najmanjšo in največjo dovoljeno vrednostjo,
4. časovna enota (angl.: time tick) – predstavlja čas nekega dogodka, izražen v stotinkah sekunde,
5. prosojni tip (angl.: opaque) – določa kodiranje, ki se uporablja za pridobivanje informacij iz niza znakov, ki ne podpirajo standarda, določenega z zapisom ASN.1,
6. predznačena celoštevilčna vrednost (angl.: integer),
7. nepredznačena celoštevilčna vrednost (angl.: unsigned integer).

Tabele MIB v verziji SNMPv1

Struktura SMI v verziji SNMPv1 definira strogo strukturirane tabele, ki se uporabljajo za združevanje tabelaričnih objektov (tj. objekt, ki vsebuje več spremenljivk). Tabele imajo nič ali več vrstic, ki so oštevilčene tako, da lahko protokol SNMP pridobi ali spremeni celotno vrstico z eno operacijo, npr. GET, GETNEXT ali SET.

Strukture SMI v SNMPv2

Struktura SMI v verziji SNMPv2 je opisana v določilu RFC 2578 [18] (angl.: request for comment). Prinaša nekaj dopolnil in izboljšav glede na podatkovne tipe SMI v verziji SNMPv1, kot na primer bitne nize znakov, omrežne naslove in števce. Bitni nizi znakov so definirani samo v verziji SNMPv2 in so sestavljeni iz nič ali več bitov. Omrežni naslov predstavlja naslov iz določene družine protokolov. Števci so ne-negativna cela števila, ki se povečujejo do največje dovoljene vrednosti, potem pa se vrnejo na nič. V verziji SNMPv2 poznamo tako 32-bitne kot tudi 64-bitne števce.

Protokol SNMP deluje v aplikacijski plasti modela OSI. V verziji SNMPv1 je specificiranih pet podatkovnih enot protokola (angl.: Protocol data unit - PDU):

1. GET zahteva – se uporablja za pridobitev dela nadzorovanih informacij.
2. GETNEXT zahteva – se uporablja iterativno za zaporedno pridobitev nadzorovanih informacij.
3. RESPONSE odgovor – uporabi agent kot odgovor na GET ali SET zahtevo.
4. SET zahteva – se uporabi za inicializacijo in spreminjanje vrednosti.
5. TRAP – se uporabi za sporočanje alarmov ali drugih asinhronih dogodkov na nadzorovanem podsistemu. V modulih MIB v strukturi SMI verzije SNMPv1 je TRAP definirani z uporabo makroja TRAP-TYPE, v verziji SNMPv2 pa ga definiramo z uporabo makroja NOTIFICATION-TYPE.

V verziji SNMPv2 sta bili dodani naslednji enoti PDU:

1. GETBULK zahteva – hitrejša metoda za pridobitev več nadzorovanih podatkov.
2. INFORM – podobna operaciji TRAP, vendar zahteva odgovor.

Informacijski moduli SMI v verziji SNMPv2

Strukture SMI v verziji SNMPv2 prav tako navajajo informacijske module, ki določajo skupine sorodnih definicij. Obstajajo tri skupine informacijskih modulov SMI:

1. Moduli MIB – vsebujejo definicije sorodnih nadzorovanih objektov.
2. Izjave ustreznosti – zagotavljajo sistematičen način za opis skupine nadzorovanih objektov.
3. Izjave zmožnosti – uporabljene so za prikaz natančnega nivoja podpore, ki ga pričakuje agent od skupine MIB. Sistem NMS lahko prilagodi odnos do agenta glede na agentovo izjavo zmožnosti.

Verzija SNMPv3

Verzija SNMPv3 je definirana z določili od RFC 3411 [19] do RFC 3418 [20]. Verzija SNMPv3 zagotavlja večjo varnost komunikacije, hkrati pa izboljša možnost oddaljenega konfiguriranja. Ta verzija je od leta 2004 standardna verzija protokola SNMP. Organizacija IETF je označila verzijo SNMPv3 za popoln internetni standard, kar je najvišje priznanje, ki ga lahko prejme protokol, določen z določili RFC. Prejšnje

verzije protokola SNMP so zastarele (angl.: obsolete). Verzija SNMPv3 nudi pomembni varnostni funkciji:

1. Overitev (angl.: authentication) – zagotavlja, da je paket generiral veljavni vir.
2. Šifriranje paketov (angl.: encryption) – onemogoči nepooblaščenim virom prisluškovanje komunikaciji.

3 Bazna postaja in naročniška postaja StarMAX

Naročniške postaje in bazne postaje StarMAX so certificirane s strani WiMAX Forum in imajo naziv »WiMAX Forum Certified«, kar zagotavlja skladnost z opremo drugih proizvajalcev, ki imajo isti certifikat. Bazne postaje serije StarMAX 6000 so sestavljena iz več gradnikov, s katerimi je mogoče sestaviti različne konfiguracije baznih postaj. Gradnike je mogoče razdeliti v skupine izdelkov, glede na naloge, ki jih zajemajo:

- ohišja StarMAX 6000,
- rezine StarMAX 6000 WiMAX PMP (angl.: point to multi point),
- sinhronizacijska rezina StarMAX 6000,
- IP stikalo StarMAX 6000,
- zunanja enota (angl.: outdoor unit - ODU) StarMAX 8000.

Ohišja StarMAX 6000

Obstajata dve vrsti ohišij, in sicer StarMAX 6100 ter StarMAX 6400. Ohišja se med seboj razlikujejo po številu rež, v katere lahko vstavimo poljubne rezine. Ohišje serije 6100 sprejme le eno rezino, ohišje 6400 pa do 6 različnih rezin. Na sliki 3.1 je prikazano ohišje StarMAX 6100 z rezino WiMAX PMP.



Slika 3.1: StarMAX 6100 ohišje

Na sliki 3.2 je prikazano ohišje StarMAX 6400 s 5 rezinami; od spodaj navzgor si sledijo: IP stikalo, 4 rezine WiMAX PMP in sinhronizacijska rezina.



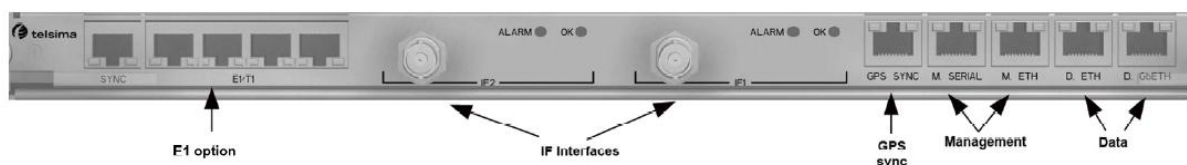
Slika 3.2: StarMAX 6400 ohišje

Rezine StarMAX 6000 WiMAX PMP

Rezine WiMAX PMP se razlikujejo glede na način delovanja ter po vrsti in številu vmesnikov PMP, ki jih vsebujejo:

- rezina 6012, vključuje dva vmesnika WiMAX PMP 802.16d,
- rezina 6011, vključuje en vmesnik WiMAX PMP 802.16d,
- rezina 6022, vključuje dva vmesnika WiMAX 802.16e PMP.

Na sliki 3.3 je prikazana postavitev vmesnikov na rezini. Na levi strani se nahaja osem vmesnikov E1 na štirih priključkih RJ-45, ki se lahko po potrebi uporabljajo za dodatni prenos podatkov po protokolu IP. Na sredini sta dva vmesnika PMP WiMAX, na desni strani pa priključek za zunanji modul GPS (sistem globalnega določanja položaja, angl: global positioning system), dva mrežna priključka za upravljanje in dva mrežna priključka za prenos IP podatkov.



Slika 3.3: Postavitev vmesnikov na rezini StarMAX 6000 WiMAX

Sinhronizacijska rezina StarMAX 6000

Za zmanjšanje interferenc v WiMAX omrežju je zelo pomembno, da so bazne postaje med seboj usklajene. To je še posebej pomembno v mešanih omrežjih, ki hkrati delujejo po standardu 802.16d in 802.16e. Če bazne postaje niso sinhronizirane med seboj, pride do interferenc, ki znatno vplivajo na hitrost prenosa podatkov, zato je GPS sinhronizacija vedno priporočena. Gleda na vrsto ohišja, ki ga uporabljamo, lahko izbiramo med dvema rezinama za sinhronizacijo:

- 6080 Urna sinhronizacijska rezina,
- 4940 GPS Urni delilnik.

Enota StarMAX 6080 je v obliki rezine. Namenjena je za vključitev v ohišja StarMAX 6400 medtem, ko je StarMAX 4940 samostojna rešitev za podporo sinhronizacije v primeru uporabe ohišja StarMAX 6100.

IP stikalo StarMAX 6000

IP stikalo se vgradi neposredno v ohišje StarMAX 6400. Stikalo poleg osnovne stikalne funkcionalnosti zagotavlja tudi vse potrebne mrežne povezave med rezinami v ohišju.

Naročniške postaja serije StarMAX 2100

Naročniška postaja podpira široko paleto mrežnih servisov, med najpomembnejšimi sta internetni dostop in telefonija preko internetnega protokola (angl.: Voice Over IP - VOIP). VOIP je tehnologija, ki prenaša glasovni signal v IP-paketih. To je osnova za prenos glasovnih podatkov prek IP-infrastrukture. Serija 2100 vsebuje več vrst naročniških postaj, namenjenih različnim načinom uporabe. Na sliki 3.4 so prikazane vse postaje serije StarMAX 2100. Postaje s serijsko številko 2120, 2130 in 2140 so namenjene uporabi v notranjih prostorih. Med seboj se razlikujejo po vrsti sprejemne antene. 2120 vsebuje notranjo dvosmerno anteno. 2130 vsebuje notranjo štiri sektorsko anteno, ki je sposobna dinamično preklapljati med sektorji, da doseže optimalni sprejem. 2140 pa ima zunanji antenski priključek, na katerega je možno priklopiti poljubno anteno. Postaje 2150 in 2160 so zunanje z integrirano anteno.



Slika 3.4: Naročniške postaje serije StarMAX 2100

3.1 Svetovna medsebojna obratovalnost mikrovalovnega dostopa

Kot smo zapisali v uvodu, pomeni kratica WiMAX svetovna medsebojna obratovalnost mikrovalovnega dostopa (angl.: worldwide interoperability for microwave access) in je skupek brezžičnih protokolov. Temelji na standardu IEEE 802.16, ki ga je razvilo združenje IEEE oziroma točneje delovna skupina Wireless MAN 802.16. Ime WiMAX je bilo določeno s strani neprofitne organizacije WiMAX Forum. Poglavitni cilj WiMAX Forum je pomoč industriji pri koordinaciji, interpretaciji in spremembah originalne, precej ohlapno napisane specifikacije standarda IEEE 802.16. WiMAX Forum vodi tudi program certificiranja, ki zagotavlja skladnost s standardom in interoperabilnost z opremo drugih proizvajalcev. Proizvajalcem, ki opravijo teste, podelijo naziv »WiMAX Forum Certified«.

3.2 Razvoj standarda IEEE 802.16

Prvi standard IEEE 802.16-2001 je bil sprejet decembra 2001. Definiral je širokopasovno brezžično povezavo točka do več točk (angl.: point to multipoint connection), v frekvenčnem pasu od 10 GHz do 66 GHz. Cilj tega standarda je bil

doseči veliko pasovno širino z uporabo visoke nosilne frekvence, kar je omejilo povezavo zgolj na vidni ravnini med oddajnikom in sprejemnikom (angl.: line of sight, LOS). Uporaba visokih nosilnih frekvenc in pogoj optične vidljivosti sta naredila standard neuporaben v urbanih okoljih, kar je privedlo do razvoja 802.16a standarda.

Dopolnitev standarda IEEE 802.16a je razširila frekvenčni pas od 2 do 10 GHz. Nižje nosilne frekvence omogočajo delovanje brez vidne ravnine (angl.: non line of sight, NLOS). Fizična plast je bila razširjena s podporo za ortogonalni frekvenčni multipleks (angl.: orthogonal frequency division multiplexing, OFDM) in ortogonalni frekvenčno porazdeljeni sodostop (angl.: orthogonal frequency division multiple access, OFDMA). Dopolnitev standarda IEEE 802.16a je bila ratificirana januarja 2003 in je predvsem namenjena zagotavljanju fiksne širokopasovnega dostopa za končne uporabnike.

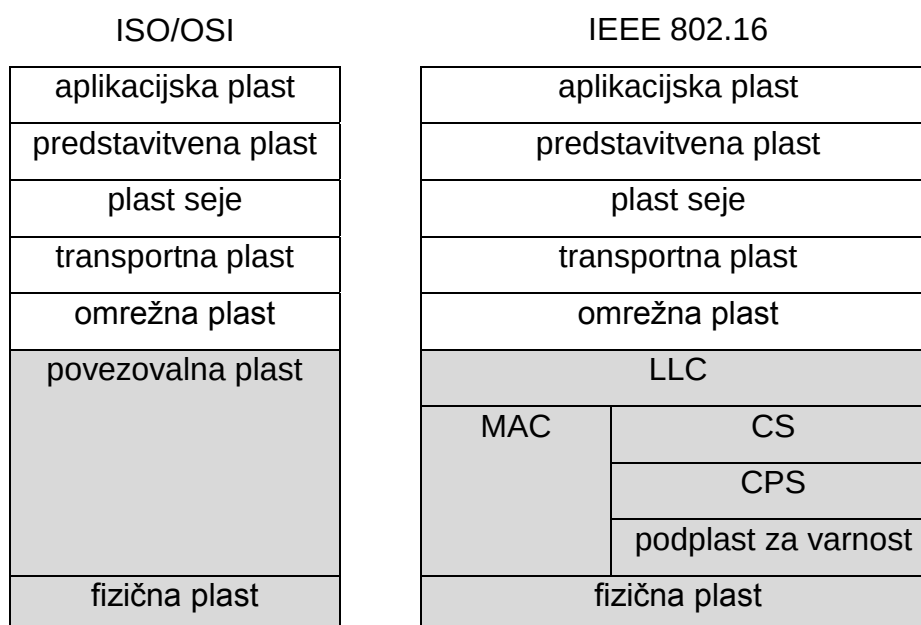
Dopolnitev standarda IEEE 802.16c je dodala podporo za uporabo licenčnega in nelicenčnega frekvenčnega spektra med 2 GHz in 10 GHz ter podporo za standard HiperMAN, ki ga je naredil evropski inštitut za telekomunikacijske standarde (angl.: European Telecommunications Standards Institute, ETSI) točneje skupina za širokopasovno radijsko dostopovno omrežje (angl.: Broadband Radio Access Networks, BRAN), da zagotovi brezžično povezavo v frekvenčnem pasu od 2 GHz do 11 GHz za Evropo in druge države ki sledijo standardom ETSI. HiperMAN je evropska alternativa standardu IEEE 802.16. Kratica HiperMAN pomeni zelo zmogljivo radijsko vlemestno mrežje (angl.: High Performance Radio Metropolitan Access Network). Dopolnitev je bila sprejeta leta 2002.

Odmevnejša standarda sta predvsem IEEE 802.16d-2004 ter IEEE 802.16e-2005. Standard IEEE 802.16d-2004 združuje vse predhodne standarde (802.16-2001, 802.16a in 802.16c) in je prvenstveno namenjen fiksnim odjemalcem kot cenovna alternativa kablskih ali xDSL storitev. Uporablja OFDMA modulacijo ter je specficiran za frekvenčno območje pod 11 GHz. Poznan je tudi pod imenom fiksni WiMAX.

IEEE 802.16e-2005 je sprememba standarda 802.16d-2004, katerega glavna naloga je pokrivanje mobilnosti. Sprejet je bil leta 2005. Poleg uvajanja mobilnosti

predvideva še številne izboljšave, glede predhodnega standarda, vključno z boljšo podporo za kakovost storitev (angl.: quality of service, QOS) in uporabo skalabilnega ortogonalno frekvenčno porazdeljenega sodostopa (angl.: Scalable Orthogonal Frequency Division Multiplexing Access, SOFDMA). Pogosto se zanj uporablja ime mobilni WiMAX.

Arhitektura protokola IEEE 802.16 definira povezovalno in fizično plast OSI modela (slika 3.5). Povezovalno plast sestavljata dve podplasti LLC (angl.: logical link control) in MAC (angl.: medium access layer). Sama MAC plast je še razdeljena na CS (angl.: convergence sublayer), CPS (angl.: common part sublayer) in varnostno podplast (angl.: security sublayer).



Slika 3.5: Primerjava modela ISO/OSI s protokolom IEEE 802.16

4 Registracija naročniških postaj

Registracijski strežnik je zadolžen za registracijo naročniških postaj. Strežnik je povezan s podatkovno bazo, iz katere črpa podatke o nastavitvah naročniških postaj in vanjo shranjuje informacije o njihovi aktivnosti. Nastavitve za vsako naročniško postajo so vnaprej pripravljene in jih operater lahko pregleduje in spreminja prek odjemalnika, ki je povezan na aplikacijski strežnik. Registracija poteka v dveh delih. Prvi del je avtorizacija, drugi pa je nastavitev profila.

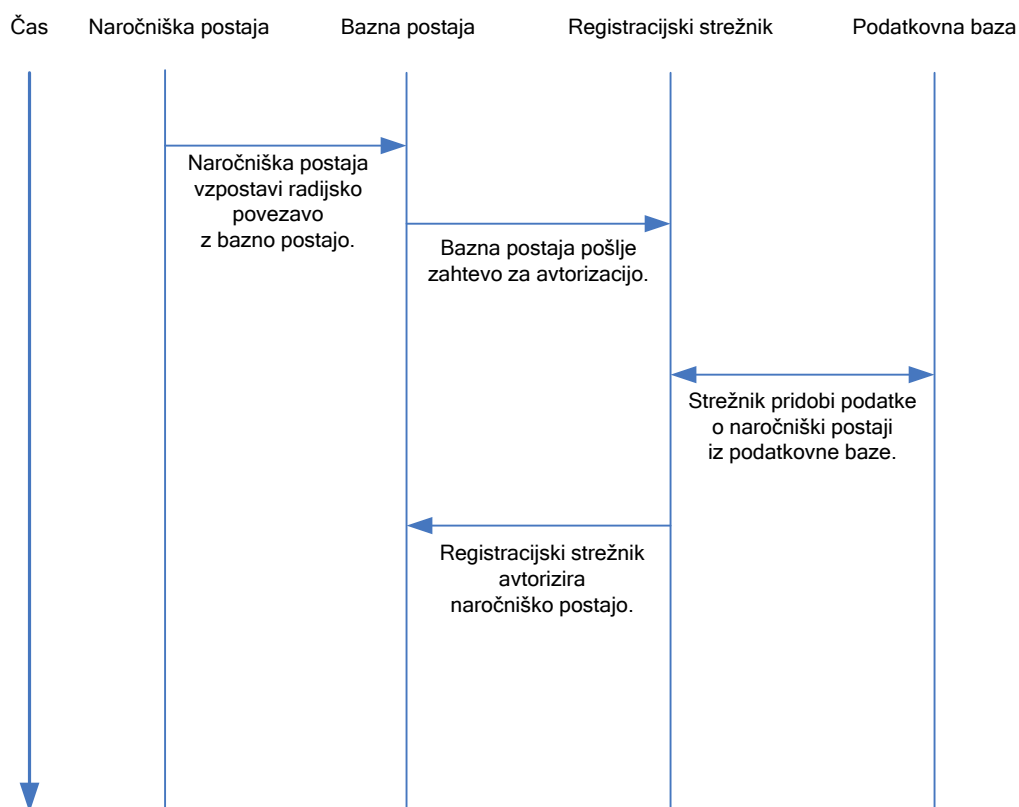
V naslednjih poglavjih opisujemo obe fazi registracije. Podrobnosti, kaj se pri registraciji nastavlja, so namenoma izpuščene, saj niso bistvene za določanje hitrosti strežnika. Poleg osnovnega opisa registracije opisujemo tudi, kako so faze registracije časovno omejene.

4.1 Avtorizacija

Avtorizacija je prva registracijska faza. Namen avtorizacije je ugotoviti, ali ima določena naročniška postaja dovoljenje za vstop v omrežje. Na bazni postaji se avtorizacijska faza začne, ko bazna postaja vzpostavi radijsko povezavo z naročniško postajo. Bazna postaja nato pošlje registracijskemu strežniku zahtevo za avtorizacijo naročniške postaje. Avtorizacijska zahteva je poslana v obliki operacije SNMP TRAP, ki vsebuje sledeče informacije:

- IP naslov bazne postaje,
- MAC naslov naročniške postaje in
- vrsto zahtevka, ki nam pove, da gre za avtorizacijo.

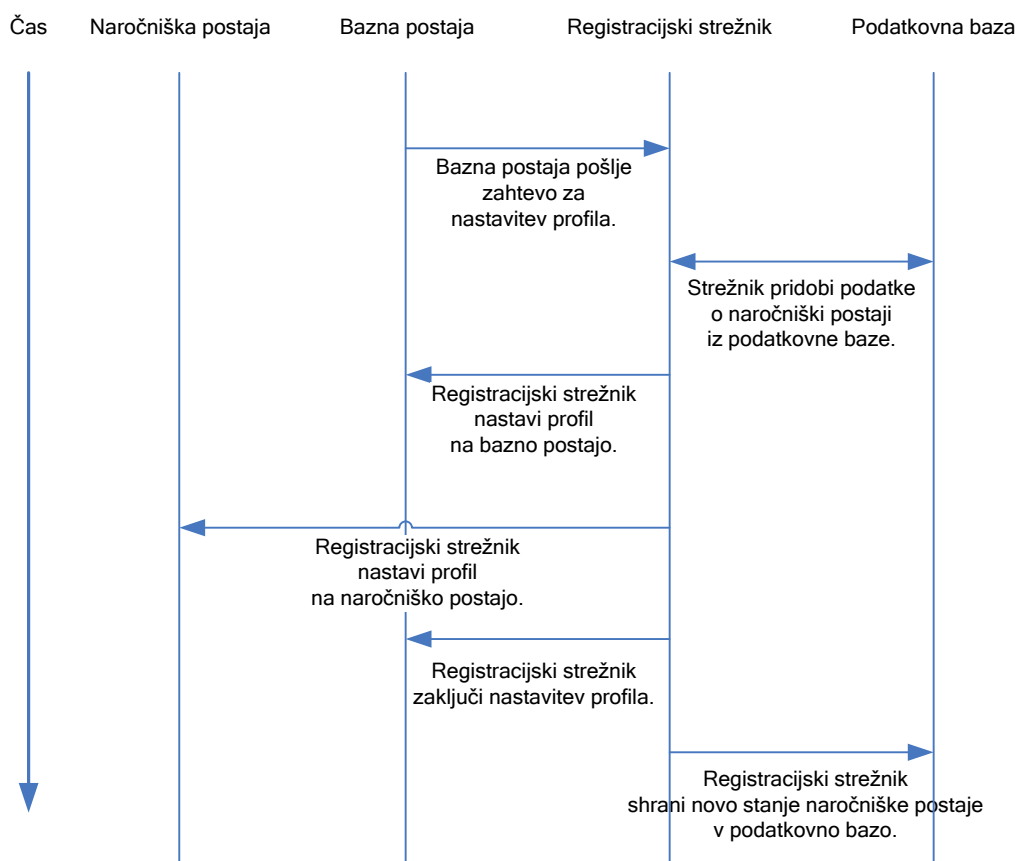
Registracijski strežnik pridobi nastavitve o naročniški postaji iz podatkovne baze in preveri, kakšne pravice ima. Odgovor je bazni postaji poslan nazaj z operacijo SNMP SET, v kateri so informacije, ali naj se naročniški postaji dovoli vstop v omrežje ali ne. Časovni potek avtorizacije je prikazan na sliki 4.1.



Slika 4.1: Časovni potek avtorizacije naročniške postaje.

4.2 Nastavitev profila

Po uspešni avtorizaciji bazna postaja pošlje zahtevo za nastavitev profila. Tudi ta zahteva je poslana v obliki operacije SNMP TRAP. Struktura zahteve je enaka kot za avtorizacijo, le vrsta zahteve je tokrat drugačna. Strežnik naloži profil iz podatkovne baze in ga z operacijami SNMP SET nastavi na bazno in naročniško postajo. Najprej se profil nastavi na bazni postaji, da se vzpostavijo ustrezni podatkovni toki do naročniške postaje, šele nato na naročniški postaji. Na koncu se bazni postaji sporoči, kako uspešno je bil nastavljen profil na naročniški postaji. V podatkovno bazo shranimo še novo stanje naročniške postaje, s čimer se zaključi postopek registracije. Časovni potek nastavljanja profila je prikazan na sliki 4.2.



Slika 4.2: Časovni potek nastavitve profila naročniške postaje.

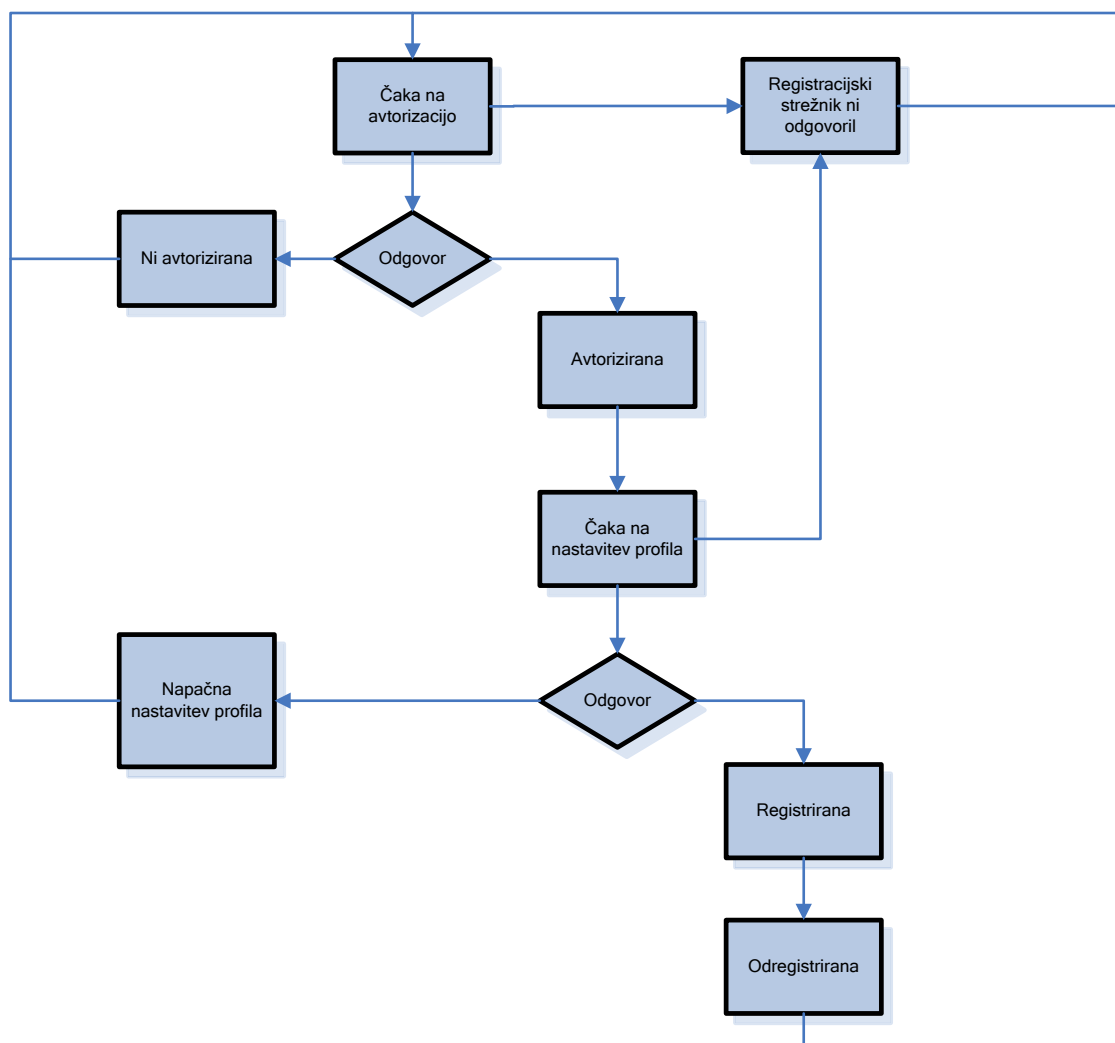
4.3 Življenjski cikel registracije

Na začetku vsake registracijske faze bazna postaja pošlje registracijskemu strežniku operacijo SNMP TRAP, ki vsebuje informacijo, da je določena naročniška postaja pripravljena za registracijo, točneje za avtorizacijo ali nastavitev profila. Faze so časovno omejene. Po preteku določenega časa naročniška postaja preide v novo stanje ali ponovi postopek registracije.

Registracija se začne z avtorizacijo. Ko se naročniška postaja poveže na bazno postajo, nastopi stanje »čaka na avtorizacijo« in bazna postaja pošlje SNMP TRAP z zahtevo po avtorizaciji. Registracijski strežnik ima zatem 30 sekund časa, da odgovori na zahtevo. Če bazna postaja po preteku tega časa ne dobi odgovora, se ponovi postopek registracije.

Postopek registracije se po 30 sekundah ponovi tudi v primeru, če je bazna postaja dobila odgovor, da naročniška postaja ni avtorizirana in ji je nastavila stanje »ni avtorizirana«. Naročniška postaja mora čakati vsaj 30 sekund pred ponovnim postopkom registracije. S tem preprečimo, da bi se neavtorizirane naročniške postaje poskušale prepogosto povezovati v omrežje in motile avtorizacijo drugih naročniških postaj.

V primeru, da je bazna postaja dobila pritrdilen odgovor v predvidenem času, začne s postopkom nastavitve profila. Stanje se nastavi na »čaka na profil« in pošlje ustrezen SNMP TRAP registracijskemu strežniku. Stanje »čaka na profil« je veljavno 5 minut. Če se profil pravočasno nastavi, se naročniški postaji nastavi stanje »registrirana«. Naročniška postaja je v tem stanju, dokler se ne odklopi. Če registracijski strežnik ni uspel v tem času nastaviti profila, se ponovi postopek registracije. V primeru napak pri nastavljanju profila se naročniški postaji nastavi stanje »napačna nastavitve profila«. V tem stanju naročniška postaja čaka 5 minut od začetka nastavitve profila, na kar ponovi registracijo. Življenjski cikel naročniške postaje je prikazan na sliki 4.3.



Slika 4.3: Življenski cikel registracije naročniške postaje

V tabeli 1 so zbrana vsa stanja s pripadajočimi časovnimi omejitvami.

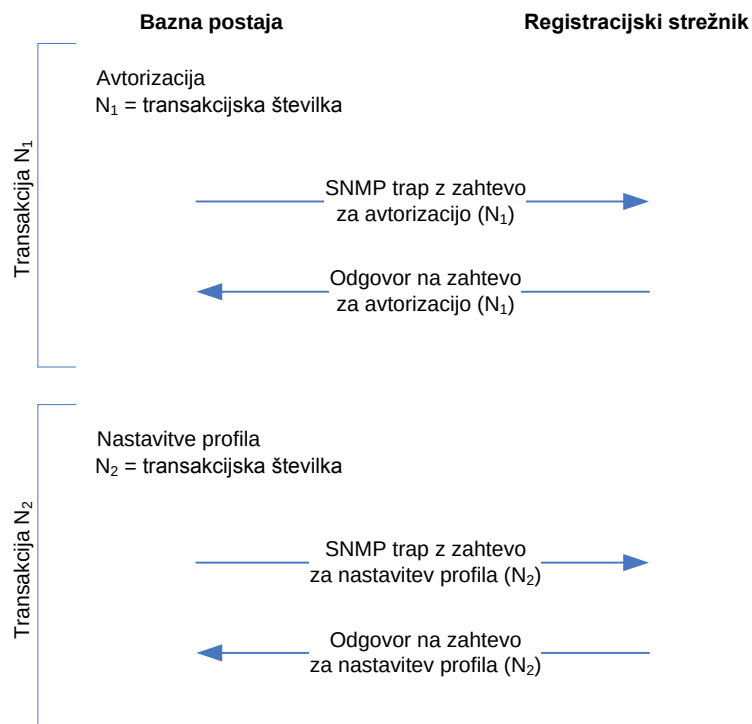
Stanje	Veljavnost stanja
Čaka na avtorizacijo	30 sekund
Ni avtorizirana	30 sekund od začetka avtorizacije
Čaka na profil	5 minut
Napačna nastavitev profila	5 minut od začetka nastavitve profila
Registrirana	Neomejeno

Tabela 1: Registracijska stanja s časovnimi omejitvami

4.4 Transakcije

Komunikacija med bazno postajo in registracijskim strežnikom je izvedena v obliki transakcij. Transakcija je način komunikacije, pri kateri si bazna postaja in registracijski strežnik poleg podatkov za registracijo izmenjujeta še transakcijsko številko. Transakcijska številka je edinstvena in se pri vsaki registracijski zahtevi (avtorizacija ali nastavitev profila) poveča. Transakcija se uspešno izvede samo v primeru, da sta številki na bazni postaji in registracijskemu strežniku enaki.

Slika 4.4 prikazuje primer registracije na podlagi transakcij. Transakcija se začne na bazni postaji in je veljavna dokler se registracijska faza ne zaključi. Vsaka zahteva se pošlje z edinstveno transakcijsko številko (N_1 za avtorizacijo ali N_2 za nastavitev profila). Transakcija se uspešno izvrši le, če registracijski strežnik na zahtevo odgovori z ustrežno transakcijsko številko (N_1 za avtorizacijo ali N_2 za nastavitev profila). V primeru odgovora na zastarelo zahtevo se transakcijski številki ne ujemata in posledično je poskus registracije neveljaven. Transakcijska številka se ujema le z zadnjo registracijsko zahtevo, zato so vse predhodne zahteve za registracijo neveljavne.



Slika 4.4: Transakcije

S transakcijami se izognemo težavam, ki bi nastopile, ko se naročniška postaja v kratkem času poskuša večkrat povezati na isto bazno postajo, ali ko je registracijski strežnik preobremenjen. Registracijski strežnik v teh primerih dobi več zahtev za registracijo iste naročniške postaje. Zahteve se shranijo v čakalno vrsto, kjer čakajo na obdelavo. Če veljavnost zahtev ne bi preverjali na podlagi transakcijske številke, bi obdelava zastarelih zahtev zmotila trenuten proces registracije.

5 Opis simulacijskega programa

Simulacijski program mora zadostiti sledečim zahtevam:

1. simulirati registracijo večjega števila naročniških postaj,
2. simulirati delovanje bazne postaje med registracijo,
3. vključiti se mora v obstoječo postavitev NMS sistema, kar pomeni da:
 - sistema ne smemo spreminjati ali celo posegati v kodo aplikacij in
 - v sistem je dovoljeno vnesti le nastavitve za naročniške postaje, s katerimi bomo izvajali simulacijo.

Da lahko simulacijski program opravlja našete funkcije, potrebuje SNMP agenta in časovnik. SNMP agent je zadolžen za komunikacijo z registracijskim strežnikom in vsebuje seznam vseh naročniških postaj z njihovimi trenutnimi stanji. Časovnik skrbi za pravilno delovanje registracije, kar vključuje proženje zahtev za registracijo ob točno določenem času in nadzor nad časovnimi omejitvami posameznih registracijskih stanj.

V nadaljevanju natančneje opisujemo ključne dele simulacijskega programa in poenostavitve, ki smo jih uvedli, za lažjo izvedbo simulacije.

5.1 Poenostavitve

Popolnega modela realnega okolja ni možno narediti, poleg tega nas pri simulaciji zanimajo samo določene lastnosti realnega sistema, zato popolnega modela ne potrebujemo. Da je simulacija enostavnejše izvedljiva, si pomagamo z določenimi poenostavitvami. Pri tem pazimo, da poenostavitve ne vplivajo na simulirano hitrost procesa registracije.

5.1.1 Poenostavitve pri modelu naročniške postaje

Osnovni pogoj za nemoteno delovanje omrežja je, da ima vsaka naprava v omrežju edinstven IP naslov. Simulacija teče na enem samem računalniku, zato je število naslovov, ki jih imamo na razpolago, omejeno s številom mrežnih kartic v računalniku. Če bi želeli, da imajo vse simulirane naročniške postaje edinstven IP

naslov, bi morali k obstoječi mrežni kartici dodati za vsako naročniško postajo nadomesti naslov (angl.: alias address).

Odločili smo se, da naročniške postaje predstavljamo registracijskemu strežniku z istim IP naslovom, saj s tem ne vplivamo na hitrost registracijskega procesa, kjer se še vedno nastavlja profil tudi na naročniški postaji. Kljub tej poenostavitvi noben del procesa registracije ni izpuščen. Z dodajanjem nadomestnih naslovov bi v simulacijo vnesli nepotrebno kompleksnost in povečali čas, potreben za pripravo simulacije.

Nadomestni IP naslovi

Ker smo že omenili dodajanje nadomestnih IP naslovov, opišemo tudi njihovo praktično izvedbo. Primer bomo podali za operacijska sistema MS Windows in Unix.

MS Windows:

Za prikazovanje in spreminjanje omrežnih nastavitev uporabljamo ukaz netsh [21]. Primer 1 prikazuje dodajanje nadomestnega IP naslova.

PRIMER 1:

```
netsh interface ip add address name = IME addr = IP mask = MASKA
```

IME ... ime mrežne naprave, ki ji dodajamo nadomestni naslov

IP ... nadomestni IP naslov

MASKA ... maska omrežja

Unix:

Tu za prikazovanje in spreminjanje nastavitev omrežja uporablja ukaz ifconfig [22]. Primer 2 prikazuje dodajanje nadomestnega IP naslova.

PRIMER 2:

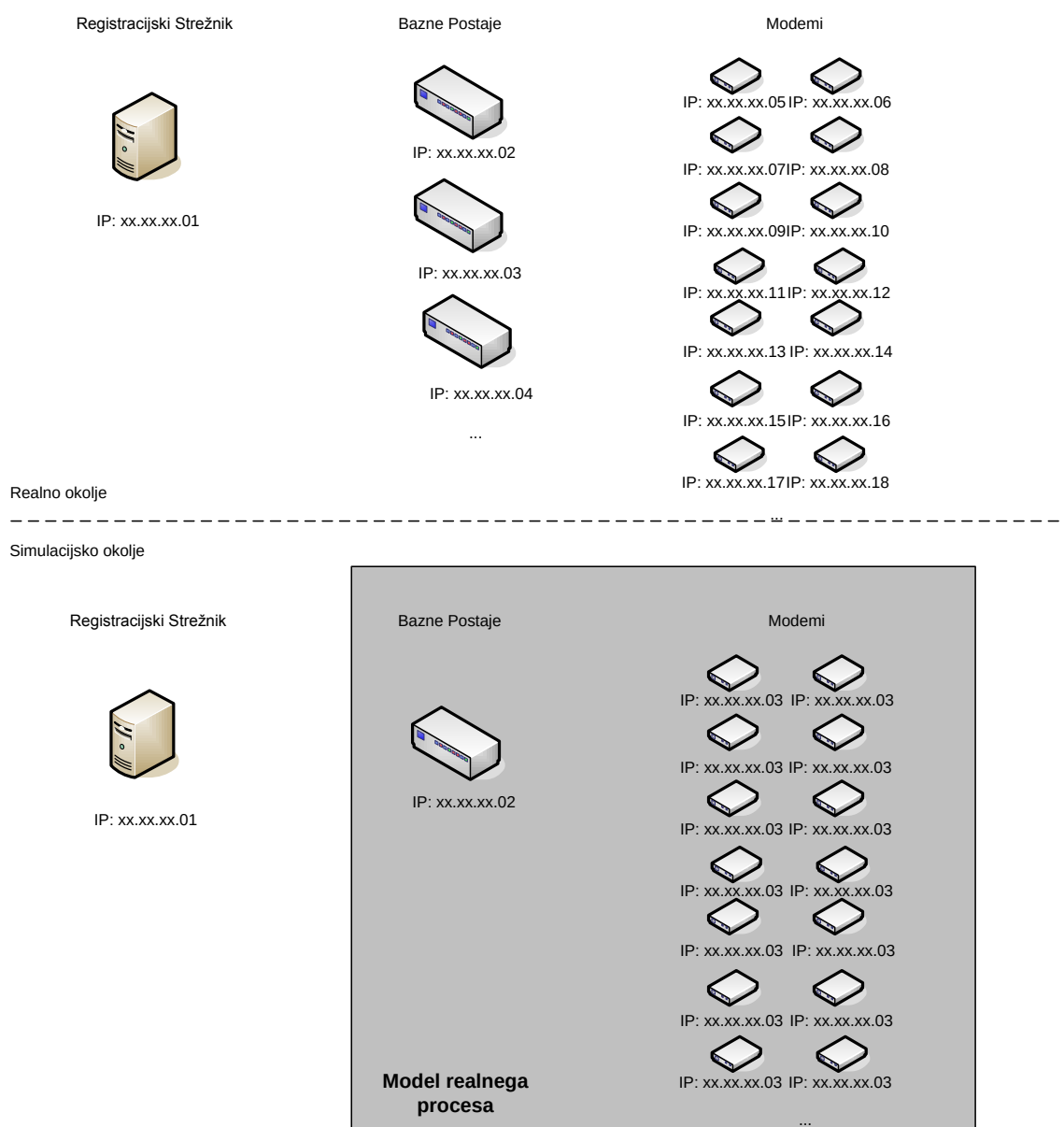
```
ifconfig IME IP up
```

IME ... ime mrežne naprave, ki ji dodajamo nadomestni naslov

IP ... nadomestni IP naslov

5.1.2 Poenostavitve pri modelu bazne postaje

Tudi pri modelu bazne postaje smo se poslužili poenostavitve. Bazna postaja lahko sprejme nekaj sto naročniških postaj. Pri tej predpostavki bi za nekaj tisoč naročniških postaj morali simulirati več deset baznih postaj. Poenostavili smo, da so vse naročniške postaje povezane na eno bazno postajo. S tem nismo izpustili nobenega dela registracijskega procesa, zato nismo vplivali na hitrost registracije. Primerjavo simulacijskega in realnega okolja prikazuje slika 5.1.



Slika 5.1: Primerjava realnega in simulacijskega okolja

Bazna postaja in naročniške postaje imajo različna IP naslova. Ta ločitev nam omogoča razločevanje namena ukaza na podlagi ciljnega IP naslova.

5.2 SNMP agent

SNMP agent je programska komponenta, ki teče na upravljanem sistemu in posreduje podatke preko SNMP do upravljalnega sistema. Ker komunikacija med registracijskim strežnikom in omrežnimi elementi (bazna in naročniška postaja) poteka po protokolu SNMP, ga je potrebno vgraditi v simulacijski program.

Agent smo realizirali z brezplačno različico AdventNet-ove knjižnice SNMP [23]. Agent opravlja naloge, povezane s komunikacijo, ter vodi seznam naročniških postaj in njihovih trenutnih stanj. Podatki o postaji so shranjeni v razredu `NarocniskaPostaja`, ki vsebuje sledeče parametre:

- MAC naslov naročniške postaje,
- trenutno stanje,
- transakcijsko številko in
- čas zadnje spremembe.

MAC naslov potrebujemo, da lahko postaje med seboj ločujemo. Trenutno stanje, transakcijsko številko in čas zadnje spremembe pa potrebujemo za nadzor nad pravilnim potekom registracije.

Del agenta, ki je zadolžen za sprejemanje ukazov, je sestavljen iz dveh razredov: `SnmpSprejemalnik` in `SnmpOdjemalnik`. `SnmpSprejemalnik` posluša na UDP vratih 161 in dobljene ukaze posreduje `SnmpOdjemalnik-u`. `SnmpOdjemalnik` dobljene ukaze obdeluje in vrača ustrezne odgovore.

`SnmpSprejemalnik`:

```
1 public class SnmpSprejemalnik {
2
3     public static final SnmpAPI api = new SnmpAPI();
4     public final SnmpSession session = new SnmpSession(api);
5
6     public SnmpSprejemalec () {
7         UDPProtocolOptions options = new UDPProtocolOptions();
```

```

8         options.setLocalPort(161);
9         session.setProtocolOptions(options);
10        try {
11            session.open();
12        } catch (SnmpException e) {
13            System.err.println("Napaka pri odpiranju seje: " + e);
14            System.exit(0);
15        }
16        session.addSnmpClient(new SnmpOdjemalnik());
17    }
18 }

```

V vrstici 1 je definiran razred `SnmpSprejemalnik`. V vrstici 3 je inicializiran objekt razreda `SnmpAPI`, ki nadzira delovanje vseh SNMP sej. V vrstici 4 se inicializira objekt »session« (razreda `SnmpSession`), preko katerega se komunicira z oddaljenimi SNMP agenti. Objektu »session« se v vrsticah od 7 do 9 nastavi, da SNMP pakete sprejema na UDP vratih 161. Nato se v vrstici 11 »session« odpre, kar ustvari vtičnico (angl: socket) za SNMP komunikacijo. V primeru napak se izvedeta vrstici 13 in 14, ki na konzolo izpišeta sporočilo in končata izvajanje programa. V vrstici 16 objektu »session« dodelimo objekt `SnmpOdjemalnik`, kjer se obdelujejo vsi prejeti SNMP pakete.

SnmpOdjemalnik:

```

1    public class SnmpOdjemalnik implements SnmpClient{
2
3        public boolean authenticate(SnmpPDU arg0, String arg1) {
4            return true;
5        }
6
7        public boolean callback(SnmpSession arg0, final SnmpPDU pdu, int arg2) {
8            // V tem delu je implementacija obdelave ukaza
9            return true;
10        }
11
12        public void debugPrint(String arg0) {
13            System.out.println(arg0);
14        }
15    }

```

V vrstici 1 je definiran razred `SnmpOdjemalnik`, ki implementira vmesnik `SnmpClient`. Vmesnik `SnmpClient` predvideva, da v razredu `SnmpOdjemalnik` definiramo metode za povratni klic (od vrstice 7 do 10), overitev (od vrstice 3 do 5) in razhroščevanje (od

vrstice 12 do 14). Vsakič, ko SnmpSprejemalnik sprejme SNMP paket, pokliče metodo za povratni klic, kjer se izvede obdelava SNMP paketa.

Na dobljene ukaze SNMP agent odgovarja z ukazom SNMP RESPONSE. Primer 1 prikazuje izvedbo SNMP odgovora s spremenljivko celoštevilčnega tipa.

PRIMER 1:

```
1      SnmpPDU pdu = new SnmpPDU();
2      UDPProtocolOptions option = new UDPProtocolOptions(remoteHost);
3      pdu.setProtocolOptions(option);
4      pdu.setCommand(SnmpAPI.GET_RSP_MSG);
5      SnmpOID oid = new SnmpOID(»1.2.3.4«);
6      SnmpVar var = SnmpVar.createVariable(1, SnmpAPI.INTEGER);
7      SnmpVarBind snmpVarBind = new SnmpVarBind(oid, var);
8      pdu.addVariableBinding(snmpVarBind);
9      session.send(pdu);
```

V vrstici 1 se inicializira SNMP PDU (podatkovna enota protokola). V vrsticah 2 in 3 se PDU-ju nastavi ciljni naslov, po potrebi se nastavijo tudi ciljna UDP vrata, če privzeta vrednosti 161 ni ustrezna. PDU-ju se v vrstici 4 nastavi ukaz SNMP RESPONSE. V vrsticah od 5 do 7 se pripravi novo spremenljivko z objektnim identifikatorjem »1.2.3.4« in celoštevilčno vrednostjo 1. V vrstici 8 se spremenljivka vključi v PDU. Nato se preko objekta »session« PDU pošlje oddaljenemu SNMP agentu.

Vsaka faza registracije se začne tako, da agent pošlje ukaz SNMP TRAP. Primer 2 prikazuje izvedbo SNMP TRAP-a.

PRIMER 2:

```
1      SnmpPDU pdu = new SnmpPDU();
2
3      UDPProtocolOptions option = new UDPProtocolOptions(remoteHost);
4      pdu.setProtocolOptions(option);
5
6      pdu.setCommand(SnmpAPI.TRP2_REQ_MSG);
7
8      SnmpOID oid = new SnmpOID(".1.3.6.1.2.1.1.3.0");
9      SnmpVar var = SnmpVar.createVariable(systemuptime, SnmpAPI.TIMETICKS);
10     ...
11     SnmpVarBind varbind = new SnmpVarBind(oid,var);
12     pdu.addVariableBinding(varbind);
```

```

13
14     oid = new SnmpOID(".1.3.6.1.6.3.1.1.4.1.0");
15     var = SnmpVar.createVariable(trapoidvalue, SnmpAPI.OBJID);
16     ...
17     SnmpVarBind varbind = new SnmpVarBind(oid, var);
18     pdu.addVariableBinding(varbind);
19
20     session.send(pdu);

```

V vrstici 1 se inicializira SNMP PDU. V vrsticah 3 in 4 se PDU-ju nastavi ciljni naslov in ukaz SNMP TRAP. PDU mora obvezno vsebovati spremenljivki za čas delovanja in objektni identifikator TRAP-a, ki sta definirani v vrsticah od 8 do 18. Po potrebi pa lahko vsebuje še druge spremenljivke. Na koncu je PDU preko objekta »session« poslan oddaljenemu SNMP agentu.

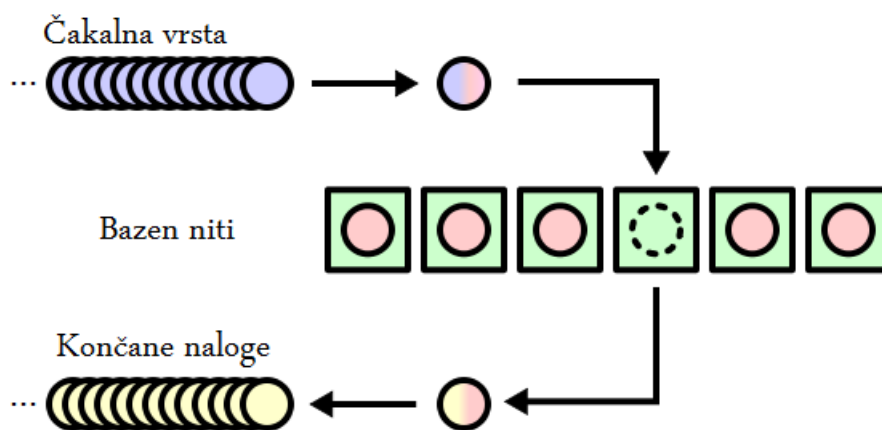
5.3 Časovnik

Naročniška postaja mora skozi več stanj, preden postane registrirana. Ker so stanja časovno omejena, mora simulacijski program vsebovati časovnik, ki preverja, kdaj določenemu stanju poteče veljavnost. Časovnik je razred oz. komponenta, ki omogoča izvajanje programa v večih nitih. Z njim lahko nastavimo zakasnitev začetka izvajanja niti programa. Časovnik je realiziran z razredom `ScheduledThreadPoolExecutor` [24], ki periodično izvaja preverjanje veljavnosti trenutnih stanj na podlagi shranjenih podatkov o določeni naročniški postaji. Za vsako postajo je znano, v katerem stanju je in kdaj je v to stanje prešla, zato časovnik lahko določi, kdaj mora postaji dodeliti novo stanje.

6 Določanje hitrosti registracijskega strežnika

Preden začnemo z določanjem zmogljivosti registracijskega strežnika je potrebno razumeti, kako le-ta deluje. Glavni strukturi za delovanje sta bazen niti in čakalna vrsta [25]. Bazeni niti vsebuje določeno število niti, ki obdelujejo zahteve. Čakalna vrsta shranjuje dobljene zahteve, dokler se ne sprosti nit v bazenu.

Delovanje bazena niti je prikazano na sliki 6.1. Modri krogi predstavljajo naloge, ki čakajo v vrsti, rumeni pa končane naloge. Zeleni kvadrati so niti. Ko nit konča nalogo, vzame naslednjo nalogo iz vrste, dokler vrsta ni prazna.



Slika 6.1: Bazen niti

Pri analizi delovanja strežnika upoštevamo diagnostične podatke, ki jih strežnik zapiše v dnevnik po vsaki registracijski fazi. Diagnostični izpisi vsebujejo sledeče informacije:

- vrsta registracijske faze (avtorizacija ali nastavljanje profila),
- MAC naslov naročniške postaje,
- čakalni čas,
- obdelovalni čas,
- velikost čakalne vrste in
- število aktivnih niti.

Z analizo teh podatkov določamo zmogljivost registracijskega strežnika. Cilj je določitev najvišje hitrosti za posamezno fazo in za celoten proces.

Vse meritve so bile opravljene s privzetimi nastavitvami strežnika, ki predvidevajo, da ima bazen niti, namenjen avtorizaciji, največ 50 niti, bazenu niti za nastavljanje profila pa največ 200 niti. Obe čakalni vrsti lahko hranita do 10 000 zahtev.

6.1 Opis opreme

Registracijski strežnik, podatkovna baza in simulacijski program so tekli vsak na svojem računalniku. S tem smo lažje določili, kaj omejuje hitrost registracije.

Registracijski strežnik:

- Procesor: UltraSPARC Ili – 650 MHz,
- RAM: 2 GB,
- OS: Sun Solaris 10.

Podatkovna baza:

- Vrsta baze: Oracle 10g,
- Procesor: Intel Core2 Duo E4500 - 2.20 GHz,
- RAM: 3 Gb,
- OS: Linux.

Simulacija:

- Processor: Intel Centrino 1.83 GHz,
- RAM: 2 GB,
- OS: Windows XP.

6.2 Hitrost avtorizacije

Simulacijski program je izveden tako, da se registracija zaključi po avtorizaciji. Nastavljanje profila se ne izvrši. Simulirali smo avtorizacijo 10 000 naročniških postaj. Postopoma smo višali frekvenco pošiljanja zahtev in opazovali, kako to vpliva na število obdelanih zahtev. Poleg hitrosti avtorizacije smo pozorno spremljali še

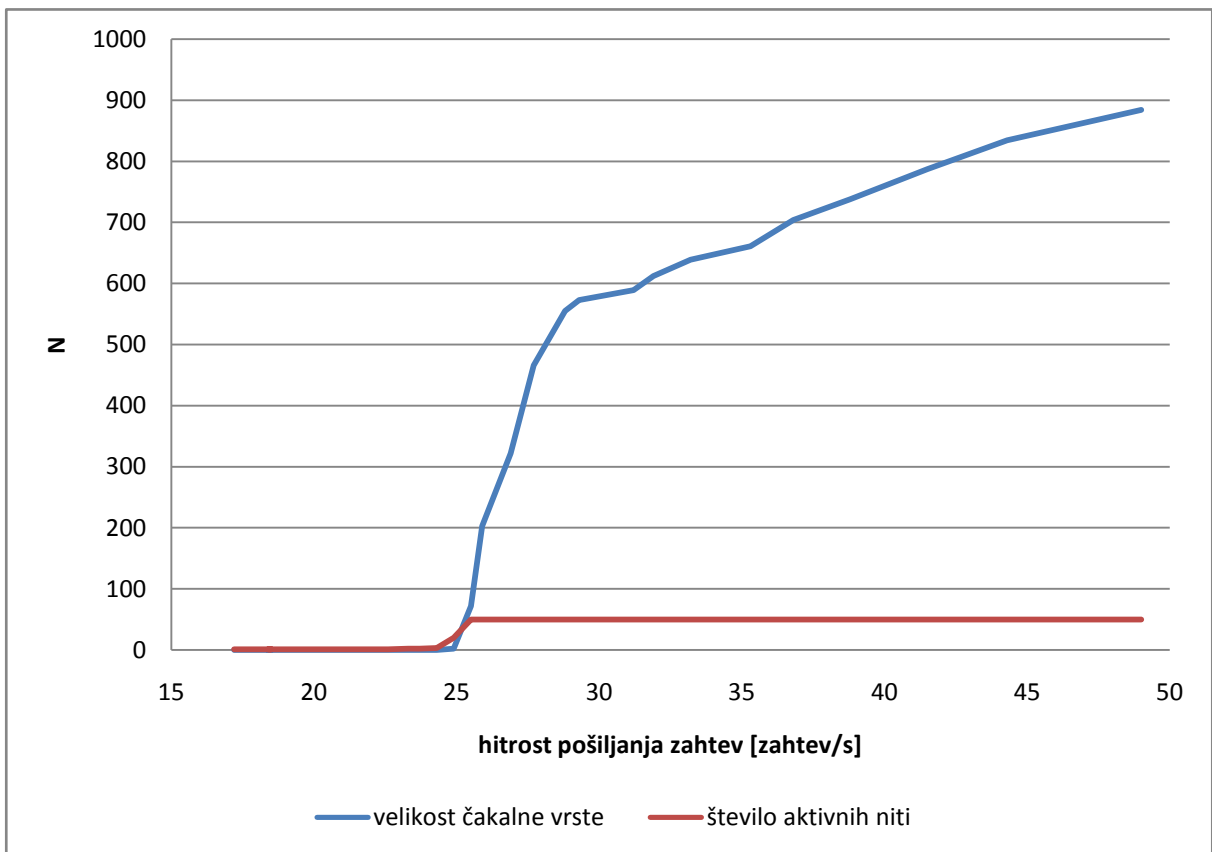
obremenjenost strežnikov, velikost čakalne vrste, število aktivnih niti in čas obdelave posameznih zahtev, ki so nam pomagali pri razumevanju njegovega delovanja.

Iz poskusnih meritev smo ocenili, da registracijski strežnik lahko obdela največ do 30 zahtev na sekundo. Da natančneje določimo delovanje strežnika okoli te točke, smo se odločili, da meritev opravimo za hitrost avtorizacije med 17 zahtev na sekundo in 50 zahtev na sekundo. Zahteve smo na začetku pošiljali v razmaku 58 ms (17 zahtev na sekundo). Razmik smo nato po vsakih 500 zahtevah zmanjšali za 2 ms do končnega razmika 20 ms (50 zahtev na sekundo).

Na začetku smo poskusili določiti najvišjo hitrost avtorizacije na podlagi opazovanja velikosti čakalne vrste in števila aktivnih niti. Obe vrednosti smo dobili iz diagnostičnih izpisov. V normalnih pogojih delovanja zahteve za avtorizacijo iz čakalne vrste preidejo takoj v obdelavo. Čakalna vrsta je v tem primeru prazna in aktiviranih je le nekaj niti. V trenutku, ko zaznamo strmo povečevanje čakalne vrste in števila aktivnih niti, pomeni, da smo presegli najvišjo hitrost obdelovanja avtorizacij, saj registracijski strežnik dobljenih zahtev ne zmore sproti obdelovati. Strežnik aktivira vse niti, ki jih ima na razpolago in zahteve začne kopičiti v čakalno vrsto.

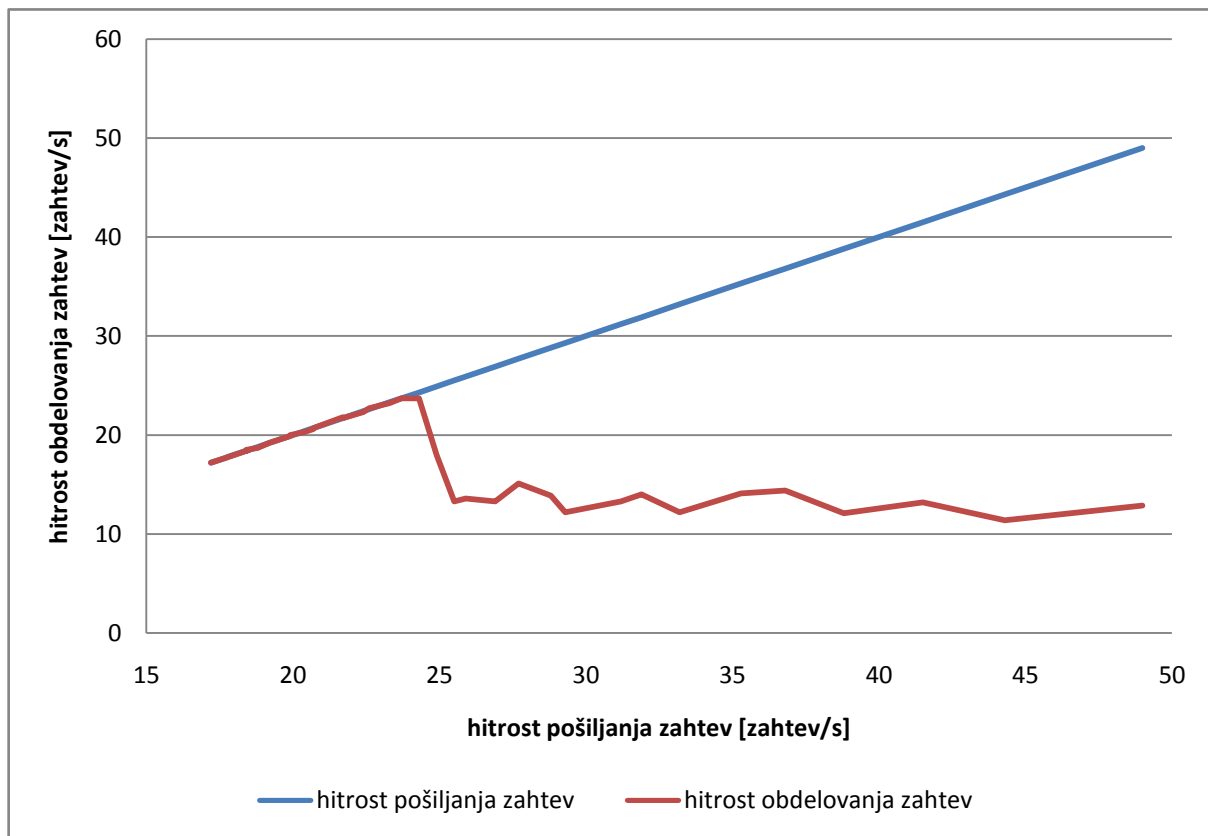
Oba parametra nam povesta samo, kdaj je presežena največja hitrost, ne izvemo pa kakšna je le-ta takrat bila. Da določimo hitrost, moramo primerjati čas, ko se čakalna vrsta in število aktivnih niti začneta povečevati, s podatki iz simulacije in ugotoviti s kakšno hitrostjo je v tistem trenutku simulacijski program pošiljal zahteve.

Slika 6.2 prikazuje velikost čakalne vrste in število aktivnih niti v odvisnosti od hitrosti pošiljanja zahtev za avtorizacijo. Iz velikosti vrste vidimo, da je najvišja hitrost obdelave zahtev za avtorizacijo 25 zahtev na sekundo. Število aktivnih niti prikazuje, da je najvišja hitrost avtorizacije 24 zahtev na sekundo. Razlika med hitrostma avtorizacije, je posledica načina delovanja bazena niti. Čakalna vrsta se začne polniti šele, ko so vse niti zasedene, zato čakalna vrsta zazna najvišjo hitrost z zamikom glede na število aktivnih niti. Zamik je sorazmeren s številom niti, ki jih strežnik lahko aktivira za obdelavo. Več kot je niti, več časa bo čakalna vrsta prazna.



Slika 6.2 Število aktivnih niti in čakalna vrsta

Najvišjo hitrost lahko določimo tudi z opazovanjem števila obdelanih zahtev. Pri normalnem delovanju se hitrost obdelave zahtev ujema s hitrostjo pošiljanja zahtev. V točki, ko hitrost obdelave zahtev ne sledi hitrosti pošiljanja zahtev, je dosežena najvišja hitrost. Odvisnost med hitrostjo obdelave zahtev in hitrostjo pošiljanja zahtev prikazuje slika 6.3, kjer razberemo, da hitrost obdelave zahtev odstopi od hitrosti pošiljanja zahtev pri hitrosti 24 zahtev na sekundo in predstavlja najvišjo hitrost avtorizacije.



Slika 6.3 Hitrost avtorizacij v odvisnosti od hitrosti pošiljanja zahtev.

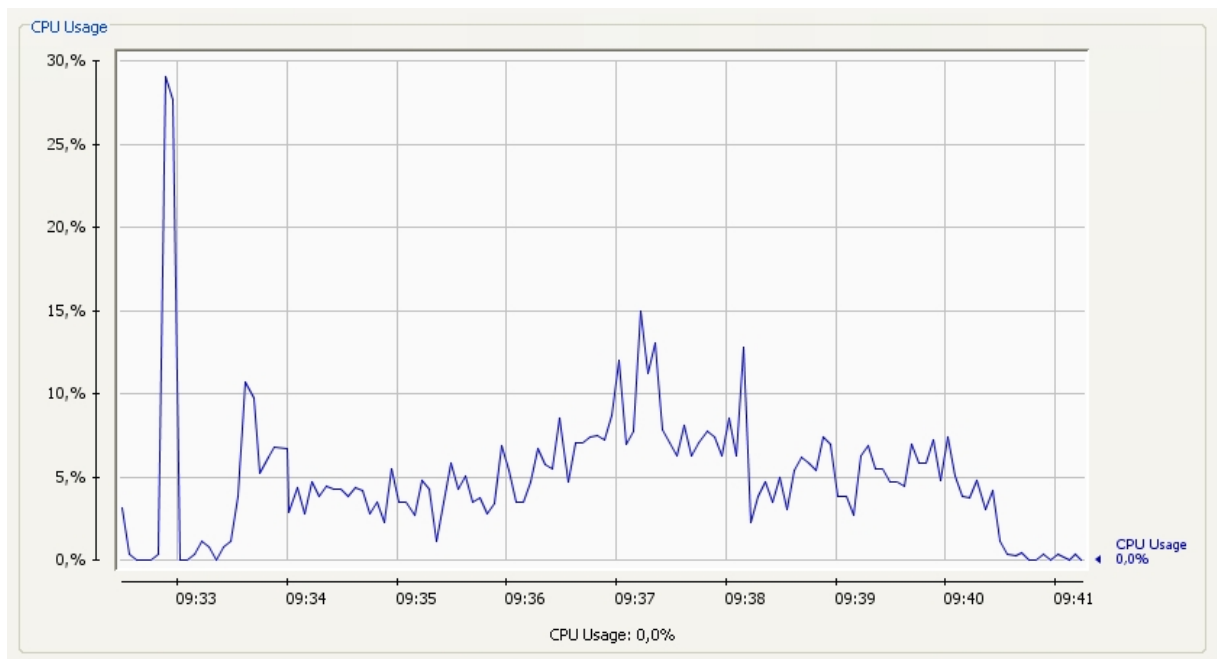
Graf pokaže tudi nepričakovano obnašanje. Hitrost obdelave zahtev se po kritični točki drastično zmanjša. Hitrost obdelanih zahtev pade na vrednost okoli 13 zahtev na sekundo. Pričakovali bi, da se maksimalna hitrost ohranja ali počasi zmanjšuje, vendar se je namesto tega pojavil stopničast padec. Vzrokov, zakaj število avtorizacij po maksimalni vrednosti pade, je lahko več:

1. simulacijski program je prepočasen in zavira delovanje registracijskega strežnika ali
2. baza omejuje delovanje registracijskega strežnika ali
3. registracijski strežnik je preobremenjen.

Prvo in drugo možnost smo ocenili z analizo procesorske obremenitve na računalnikih, kjer imamo podatkovno bazo in simulacijski program.

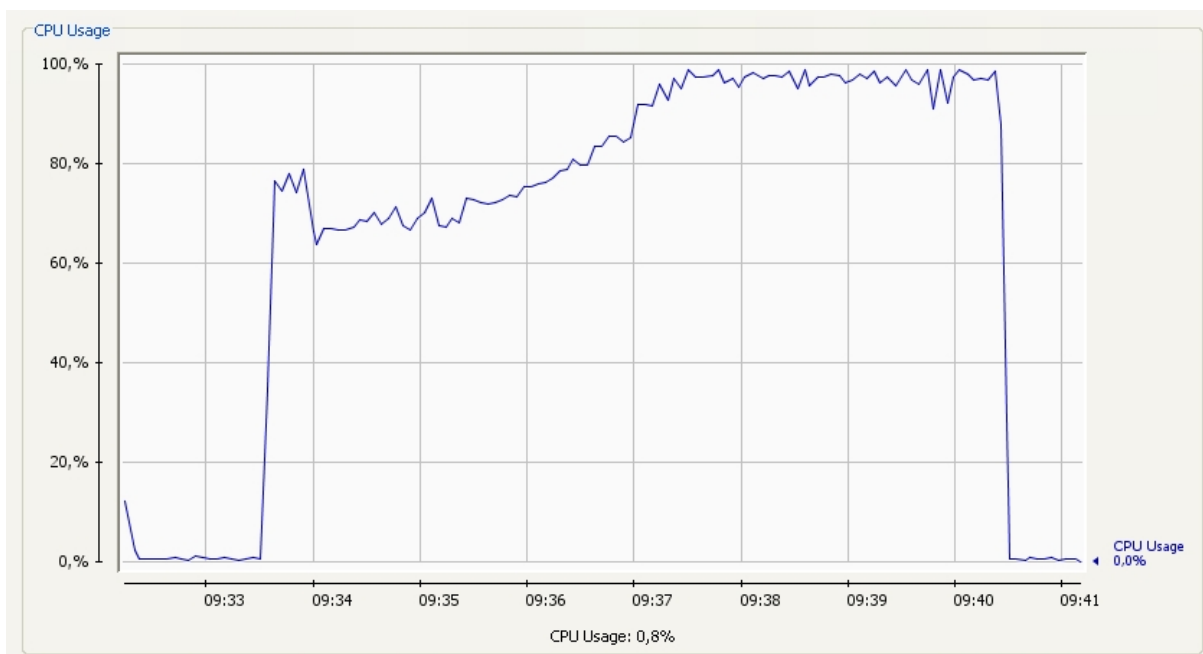
Obremenitev procesorja na podatkovni bazi smo opazovali z ukazom `top` [26], ki je pokazal, da je bila obremenitev procesorja le okoli 2 odstotka in tako ne more biti vzrok za omejitev hitrosti avtorizacije.

Simulacijski program smo nadzorovali s pomočjo JMX tehnologije (angl.: Java management extensions). JMX je javanska tehnologija, ki nudi orodja za upravljanje in nadzorovanje aplikacij. Za simulacijski program je izrisana poraba procesorske moči na sliki 6.4. Iz slike je razvidno, da je med delovanjem obremenjenost dosegla najvišjo vrednost 30 odstotkov polne obremenitve, po čemer sklepamo, da obremenitev računalnika, na katerem teče simulacijski program, ni razlog za upočasnitev delovanja.



Slika 6.4 Procesorska obremenitev simulacijskega programa

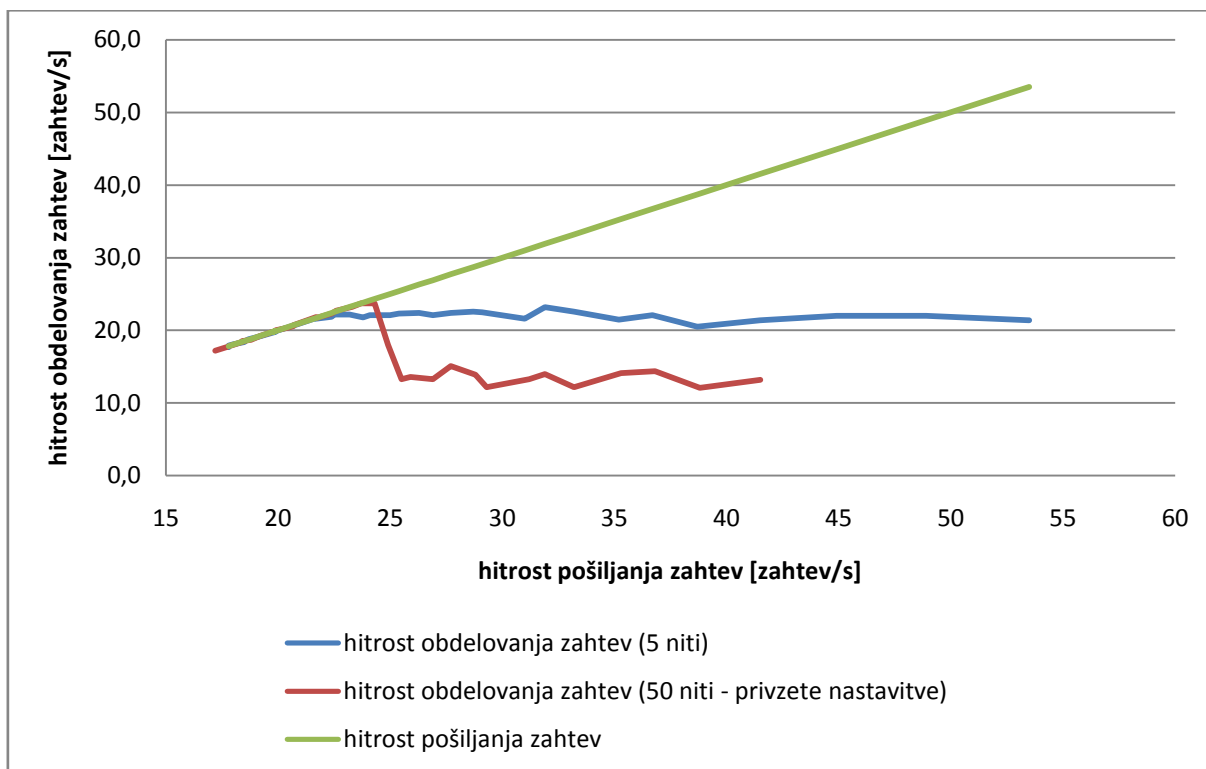
Izločili smo možnosti, da sta simulacija in baza omejevala hitrost sistema. Ostaja samo možnost, da je registracijski strežnik preobremenjen in ne zmore obdelati več kot 24 zahtev na sekundo. Porabo procesorske moči na registracijskem strežniku prikazuje slika 6.5. Iz slike se vidi, da je procesor od določenega trenutka naprej 100 odstotno obremenjen, po čemer sklepamo, da je preobremenjenost registracijskega strežnika razlog, da hitrost obdelave zahtev ne more biti večja.



Slika 6.5 Procesorska obremenitev registracijskega strežnika pri avtorizaciji

Razložili smo, kaj omejuje najvišjo hitrost tega sistema. Padec hitrosti po kritični točki, ki je prikazan, na sliki 6.3 je najverjetneje posledica aktivacije vseh niti, ki jih ima registracijski strežnik na razpolago za delovanje, pri čemer se dodatni procesorski čas porablja za preklope med njimi.

Da trditev dokažemo, smo ponovili isto meritev, vendar smo tokrat registracijskemu strežniku zmanjšali število niti v bazenu na 5. Na sliki 6.6 so prikazani rezultati meritev za strežnik s 50-imi nitmi (privzete nastavitve) in za strežnik s 5-imi nitmi. Pri strežniku s 50-imi nitmi smo že ugotovili, da pride do znatnega padca hitrosti, ko doseže najvišjo hitrost in se aktivirajo vse niti. Pri strežniku s samo 5-imi nitmi najvišja hitrost ostane konstantna in dokazuje, da je padec hitrosti posledica prevelikega števila aktivnih niti.

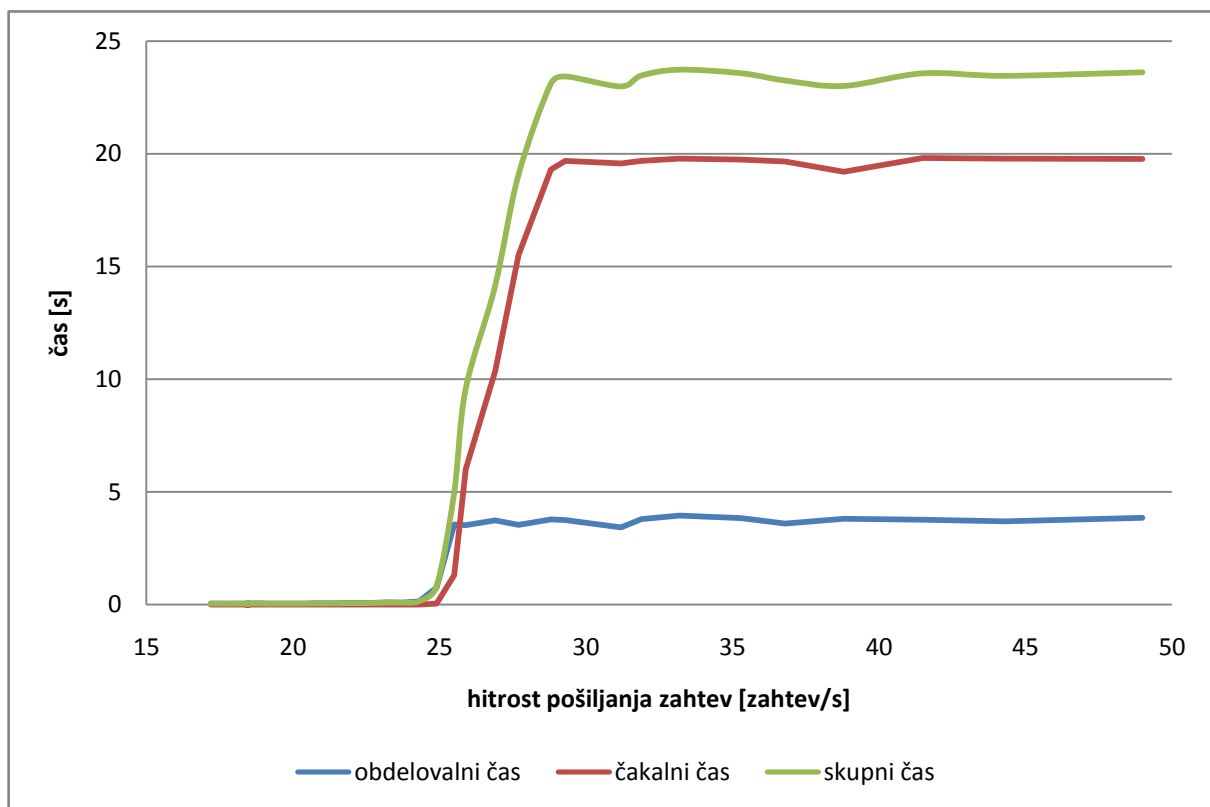


Slika 6.6: Hitrost avtorizacij v odvisnosti od hitrosti pošiljanja zahtev – 5 niti.

6.2.1 Analiza obdelave zahtev

Analizirajmo vpliv hitrosti pošiljanja zahtev na njihovo obdelavo. Predvsem nas zanima, koliko časa zahteve čakajo, preden pridejo v obdelavo in koliko časa se obdelujejo. Preverjamo tudi, če pride do izpuščanja zastarelih zahtev pred in po dosegu kritične hitrosti 24 zahtev na sekundo.

Skupen čas obdelave je sestavljen iz čakalnega in obdelovalnega časa. Čakalni čas zahteva čaka v vrsti preden gre v obdelavo. Obdelovalni čas potrebuje strežnik za obdelavo zahteve. Gibanje teh parametrov je prikazano na sliki 6.7.



Slika 6.7 Obdelovalni, čakalni in celotni čas

Preden smo dosegli kritično hitrost, zahteve niso čakale v vrsti, ampak so tokoj prešle v obdelavo. V povprečju so bile zahteve obdelane v 46 ms. Skupni čas je odvisen samo od časa obdelave. Po kritični točki so se vrednosti drastično spremenile. Zahteve so v čakalni vrsti čakale v povprečju 19 sekund. Za obdelavo je strežnik potreboval povprečno 4 sekunde. Povprečen skupni čas je bil 23 sekund.

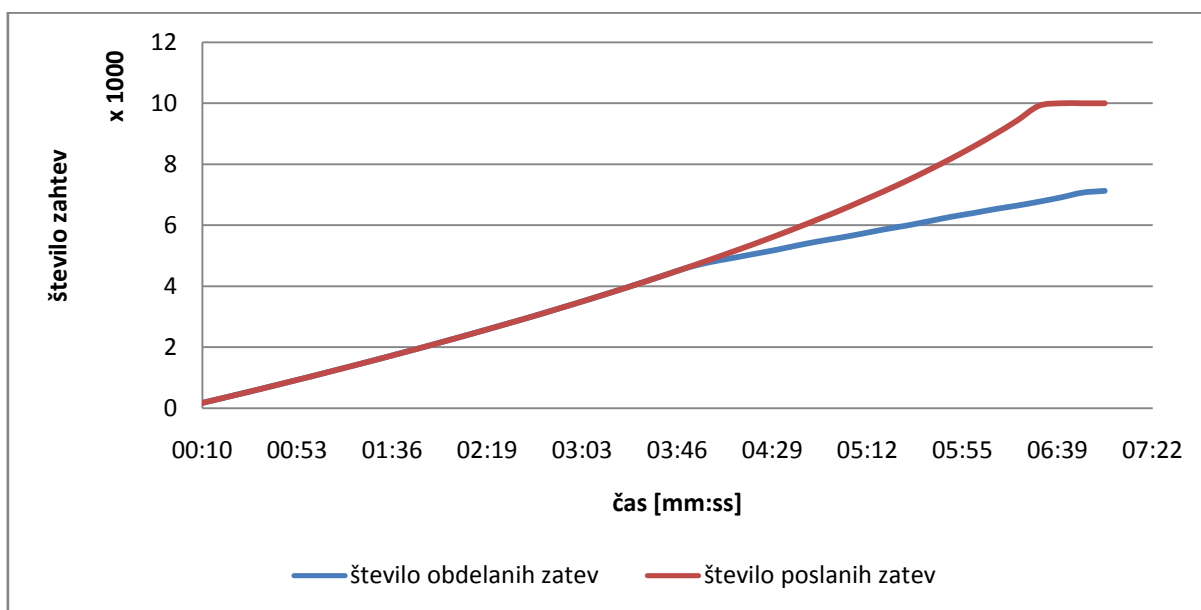
Avtorizacija na bazni postaji je omejena na 30 sekund. Po preteku tega časa se vse nastavitve, vezane na to naročniško postajo, zavrnejo in avtorizacija je neuspešna. Pomembno je, da skupni čas ne preseže 30 sekund. Na skupni čas vplivamo tako, da zmanjšamo čakalni čas ali čas obdelave.

Na čakalni čas vplivamo z nastavljanjem omejitve, ki določa največjo starost zahtev, ki jih strežnik še obdeluje. Pri prevzetih nastavitvah je ta omejitev nastavljena na 20 sekund, kar privede do tega, da je čakalni čas pri večji obremenitvi malenkost pod to mejo (19 sekund). Z zmanjšanjem te vrednosti omogočimo, da se lahko zahteva obdeluje dlje časa. Ima pa negativen učinek na število obdelanih zahtev, saj se zaradi prekoračitve omejitve več zahtev izpusti. V primerih, ko strežnik ni

preobremenjen lahko na zmanjšanje čakalnega časa vplivamo s povečanjem števila aktivnih niti. Strežnik bo obdeloval več zahtev sočasno in zahteve bodo zato manj čakale v čakalni vrsti.

Na obdelovalni čas lahko vplivamo le posredno. Če je vzrok za visok obdelovalni čas preobremenitev strežnika, ga lahko zmanjšamo z zmanjšanjem števila aktivnih niti. Strežnik obdeluje manj zahtev sočasno in posledično manj obremeni procesor. V primeru drugih vzrokov (nepravilno delovanje omrežja, podatkovne baze...) pa je potrebno ustrezno ukrepati pri izvoru problema.

Podatek, da je čakalni čas blizu nastavljene omejitve, nakazuje, da je prišlo do izpuščanja zahtev. Razmerje med številom poslanih in obdelanih zahtev je prikazano na sliki 6.8, kjer je podan kumulativni potek avtorizacij med simulacijo. Zaradi predolgega čakanja v vrsti se določen del naročniških postaj ne avtorizira.



Slika 6.8 Kumulativno število poslanih in obdelanih avtorizacij

6.3 Hitrost nastavitve profila

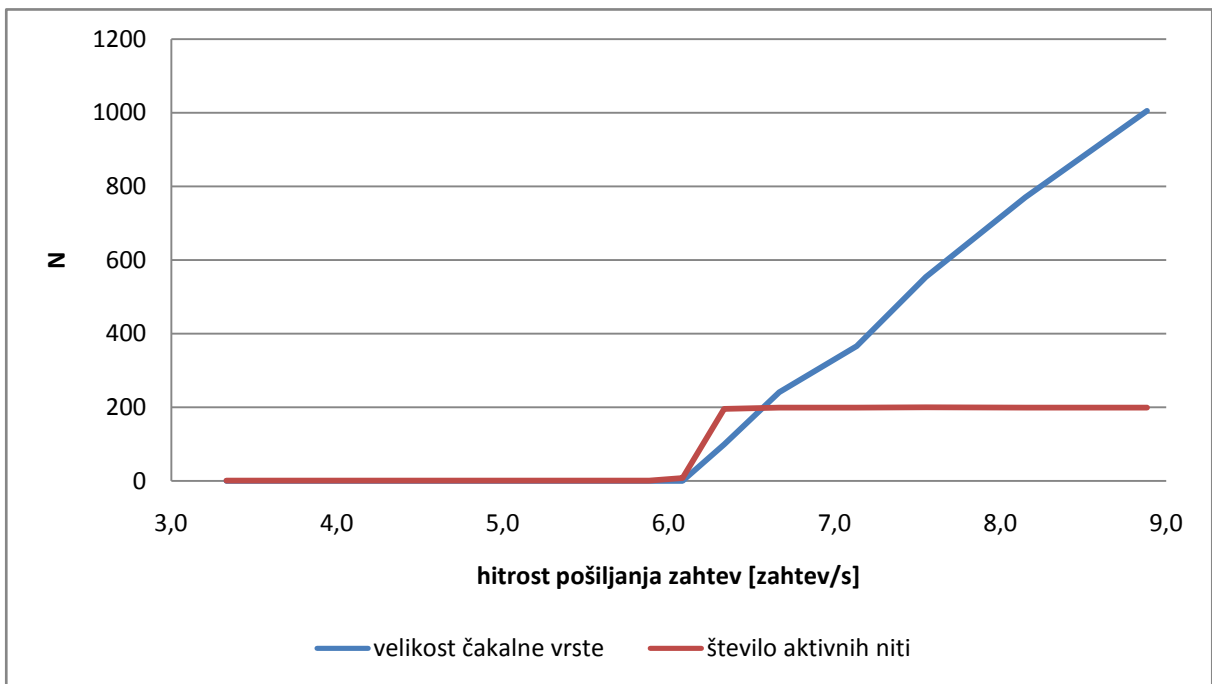
Po vsaki uspešni avtorizaciji je potrebno naročniški postaji nastaviti profil. Nastavljanje profila zahteva veliko več komunikacije s podatkovno bazo, bazno postajo in naročniško postajo, zato lahko vnaprej predpostavimo, da je ta faza nekoliko počasnejša. Dostop do podatkovne baze in baznih postaj omogočajo hitre in

zanesljive mrežne povezave. Dostop do naročniške postaje pa je vzpostavljen prek radijske povezave. V primeru, da je naročniška postaja oddaljena od bazne postaje oziroma je zaradi drugih razlogov radijska povezava slaba ali na meji delovanja, je nastavljanje profila na naročniški postaji zamuden proces.

SNMP temelji na UDP protokolu in posledično pri slabi radijski povezavi obstaja velika verjetnost, da se del paketov izgubi, zato je potrebno večkrat ponoviti nastavljanje profila. To je eden glavnih razlogov, da za procesiranje uporabljamo večje število niti. Kadar se nastavljanje profila na eni naročniški postaji upočasni in zahteva ponovitve, to ne vpliva na registracijo ostalih postaj, ki se procesirajo v svojih nitih. Pri simulaciji predpostavimo, da je radijska povezava zanesljiva, saj drugače ne moremo izmeriti, kje so meje zmogljivosti registracijskega strežnika.

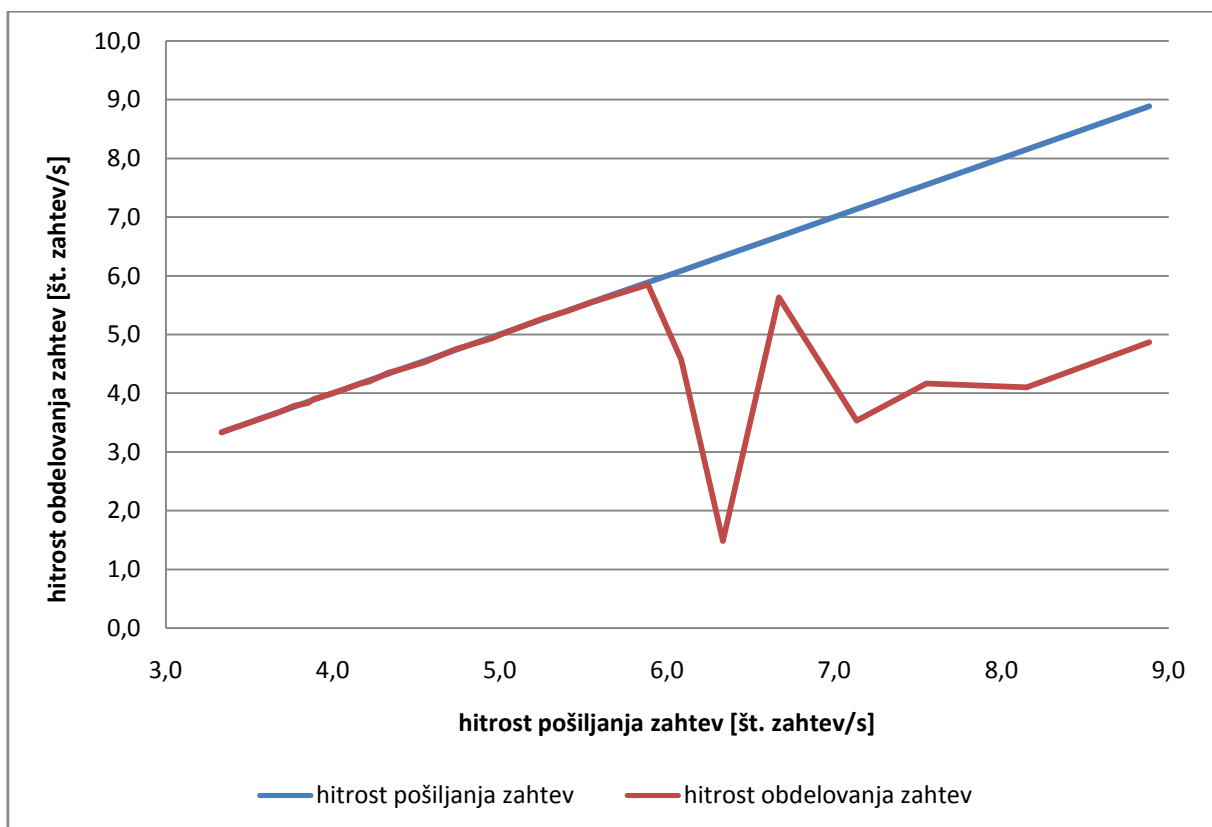
Podobno kot za meritev hitrosti avtorizacije, smo tudi za to meritev prilagodili simulacijski program. Tokrat se je izvajala samo nastavitev profila. Avtorizacijska faza je bila izpuščena. Ponovno smo simulirali 10 000 naročniških postaj. Na podlagi predhodnih testnih meritev smo ocenili, da je največja hitrost obdelave zahtev za nastavitve profila okoli 5 zahtev na sekundo. Z namenom da točneje določimo najvišjo hitrost, smo začetni razmik med zahtevki nastavili na 300 ms (3,3 zahteve na sekundo) in smo ga vsakih 500 naročniških postaj zmanjšali za 10 ms do končnega razmika 110 ms (8,9 zahtev na sekundo).

Velikost čakalne vrste in število aktivnih niti je tudi v tem primeru dober pokazatelj, koliko zahtev za nastavitve profila zmore obdelati registracijski strežnik. Slika 6.9 prikazuje velikost vrste v odvisnosti od hitrosti pošiljanja zahtev. Po 6 zahtevkih na sekundo začne velikost vrste in število aktivnih niti strmo naraščati, kar pomeni, da je registracijski strežnik presegel najvišjo hitrost.



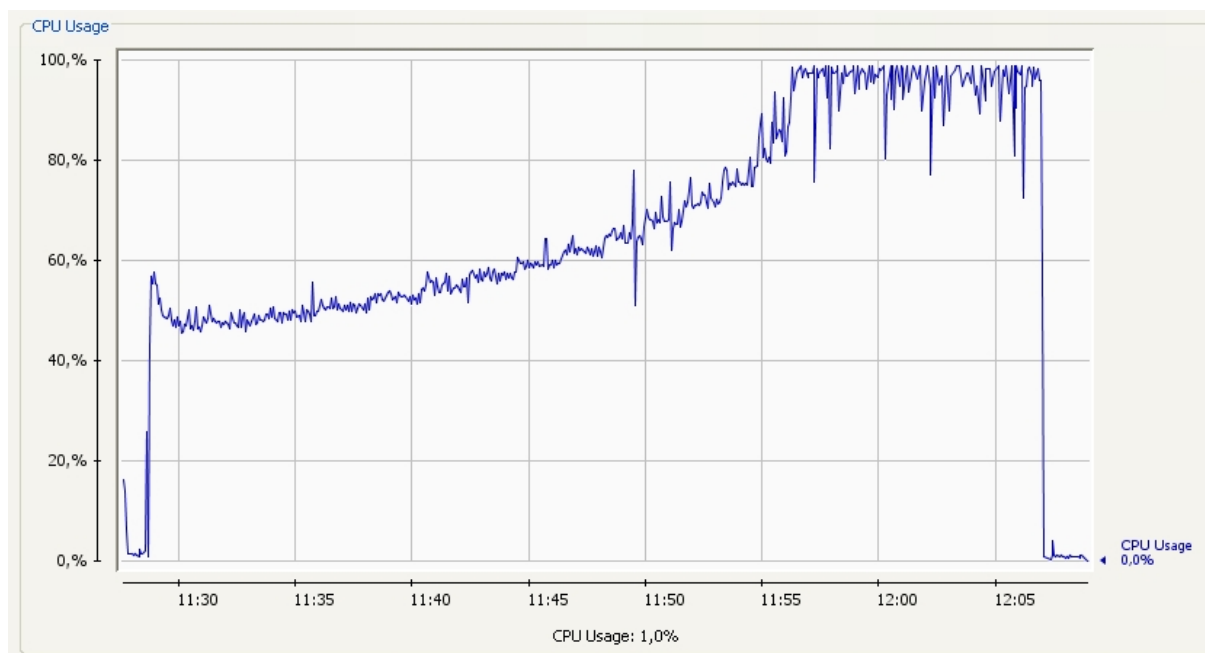
Slika 6.9 Število aktivnih niti in čakalna vrsta

Podobno kot smo to storili pri avtorizaciji, analiziramo še odvisnost med hitrostjo obdelanih zahtev in hitrostjo poslanih zahtev, ki jo prikazuje slika 6.10.



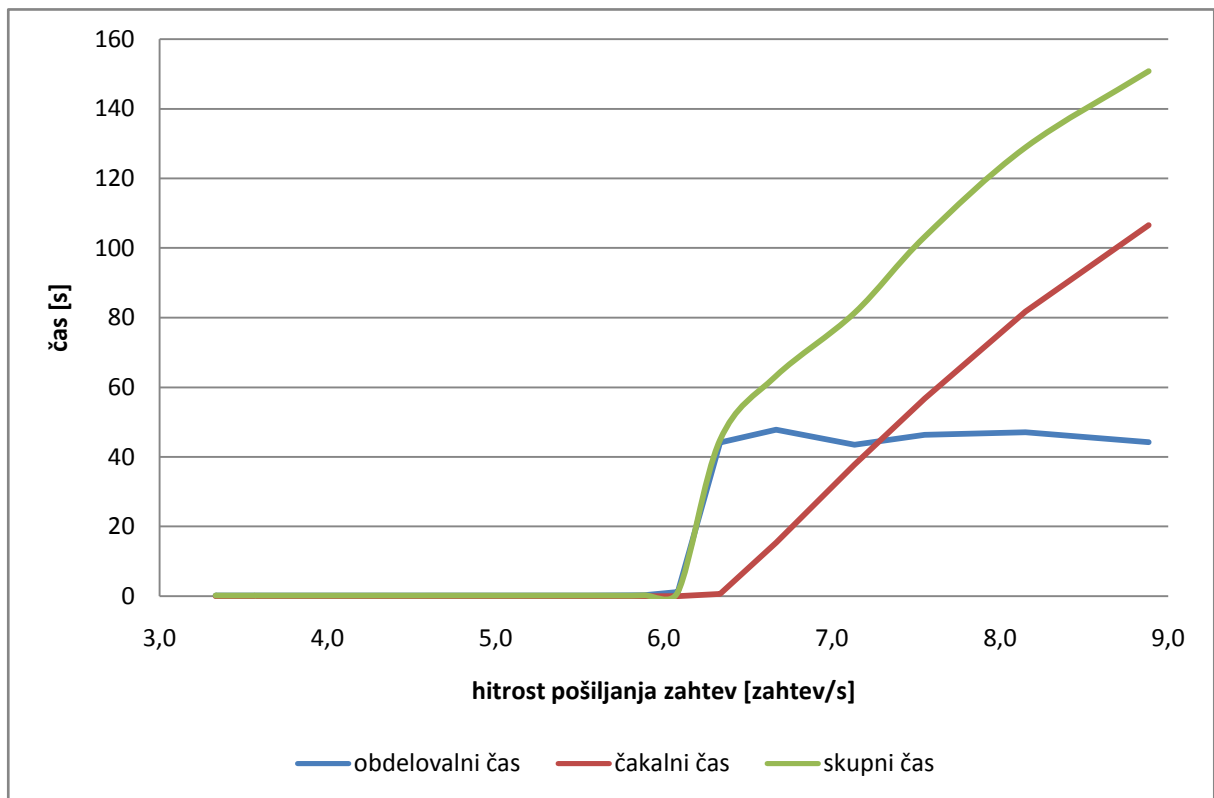
Slika 6.10 Hitrost nastavitve profila v odvisnosti od hitrosti pošiljanja zahtev.

Hitrost obdelanih zahtev po kritični točki strmo pade in se nato ustali pri hitrosti okoli 4 zahteve na sekundo. Razlogi za znaten padec hitrosti je enak kot pri avtorizaciji. Procesor na registracijskem strežniku je dosegel 100 odstotno obremenitev, kar je razvidno iz slike 6.11, ki prikazuje obremenitev procesorja na registracijskem strežniku. Registracijski strežnik z namenom, da hitreje obdelava vse zahteve, aktivira vse niti, ki jih ima na razpolago, in s tem še dodatno obremeni procesor.



Slika 6.11 Procesorska obremenitev registracijskega strežnika pri nastavljanju profila

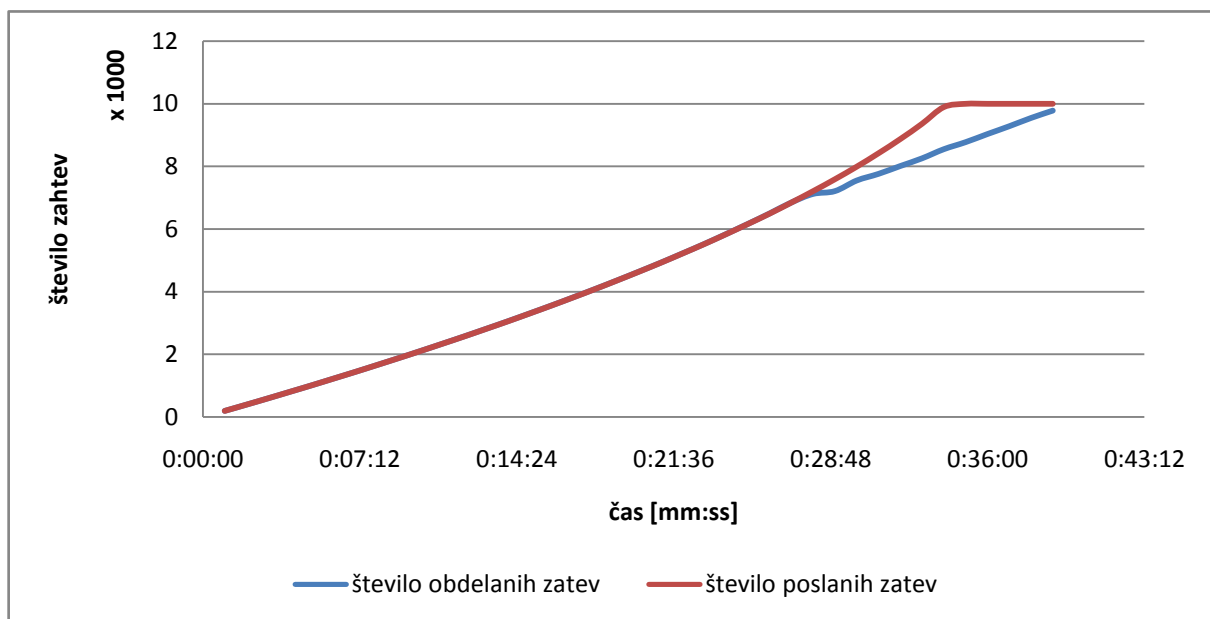
Uspešnost nastavitve profila je odvisna tudi od časa, ki ga registracijski strežnik porabi za nastavitev profila. Skupni čas je sestavljen iz čakalnega in obdelovalnega časa. Skupni čas ne sme presegati 5 minut, saj bi se v nasprotnem primeru nastavitev profila zavrnila s strani bazne postaje oz. simulacije. Slika 6.12 prikazuje čakalni čas, obdelovalni čas in skupni čas v odvisnosti od hitrosti pošiljanja zahtev.



Slika 6.12 Obdelovalni, čakalni in celoten čas

Pred kritično točko se zahteve obdelujejo v povprečju 160 ms in ne čakajo v čakalni vrsti. Celoten čas v tem primeru odpade samo na procesiranje. Po kritični točki čakalni čas strmo narašča. Proti koncu meritve doseže vrednost 106 sekund, vendar naraščajoči graf za čakalni čas nakazuje, da bi se le-ta še povečal, če bi meritev trajala več časa. Čakalni čas bi se ustalil pri vrednosti 240 sekund, saj ima strežnik privzeto nastavitev, da zahteve starejše od 240 sekund izpusti.

Podobno se je zgodilo tudi pri avtorizacijski fazi. Čas se je ustalil pri mejni vrednosti. Po kritični točki občutno poskoči tudi čas, porabljen za procesiranje. Ustali se na povprečni vrednosti 46 sekund. Glede na to, da se je obdelovalni čas ustalil pri vrednosti 46 sekund, bodo zahteve uspešno izvedene tudi, če čakalni čas doseže mejno vrednost 240 sekund, saj celoten čas ne bi presegal 300 sekund. V nasprotnem primeru je potrebno prilagoditi največjo vrednost čakalnega časa tako, da skupni čas ne preseže 300 sekund. Do izpuščanja zahtev ni prišlo, kar potrjuje slika 6.13. Vsi poslani zahtevki so bili obdelani, le po kritični točki so bili le-ti obdelani z zamikom.



Slika 6.13 Kumulativno število obdelanih zahtev za nastavitev profila

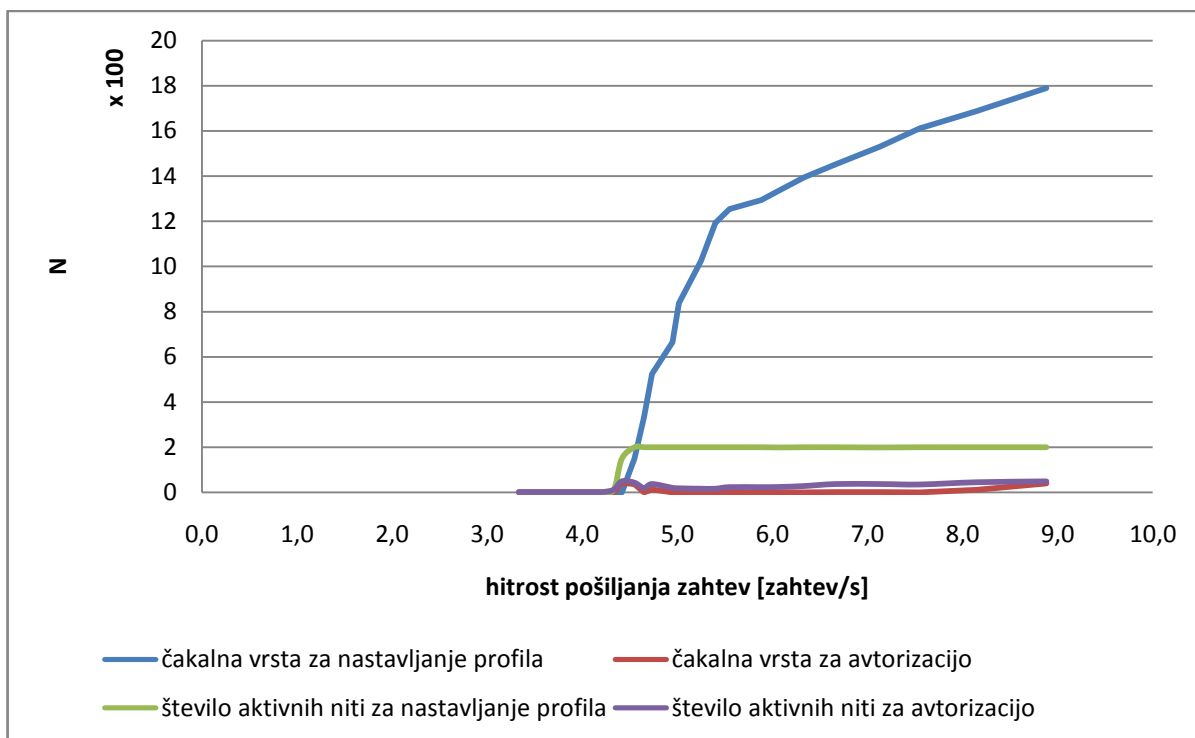
6.4 Hitrost registracije

Predhodne meritve so pokazale, kako hitro je registracijski strežnik zmožen obdelovati posamično fazo. Le-ta je zmožen avtorizirati 24 naročniških postaj na sekundo in nastaviti profil šestim naročniškim postajam na sekundo. Iz teh rezultatov lahko sklepamo, da bo najverjetneje na hitrost vplivalo predvsem nastavljanje profila. Pričakovana hitrost pa je manjša od 6 registriranih naročniških postaj na sekundo, saj se izvajata obe fazi sočasno.

V nadaljevanju smo simulirali registracijo 10 000 naročniških postaj z enakimi parametri kot pri nastavitvi profila. Začetni razmik med zahtevami je bil nastavljen na 300 ms (3,3 zahteve na sekundo) in smo ga vsakih 500 postaj zmanjšali za 10 ms do končnega razmika 110 ms (8,9 zahtev na sekundo).

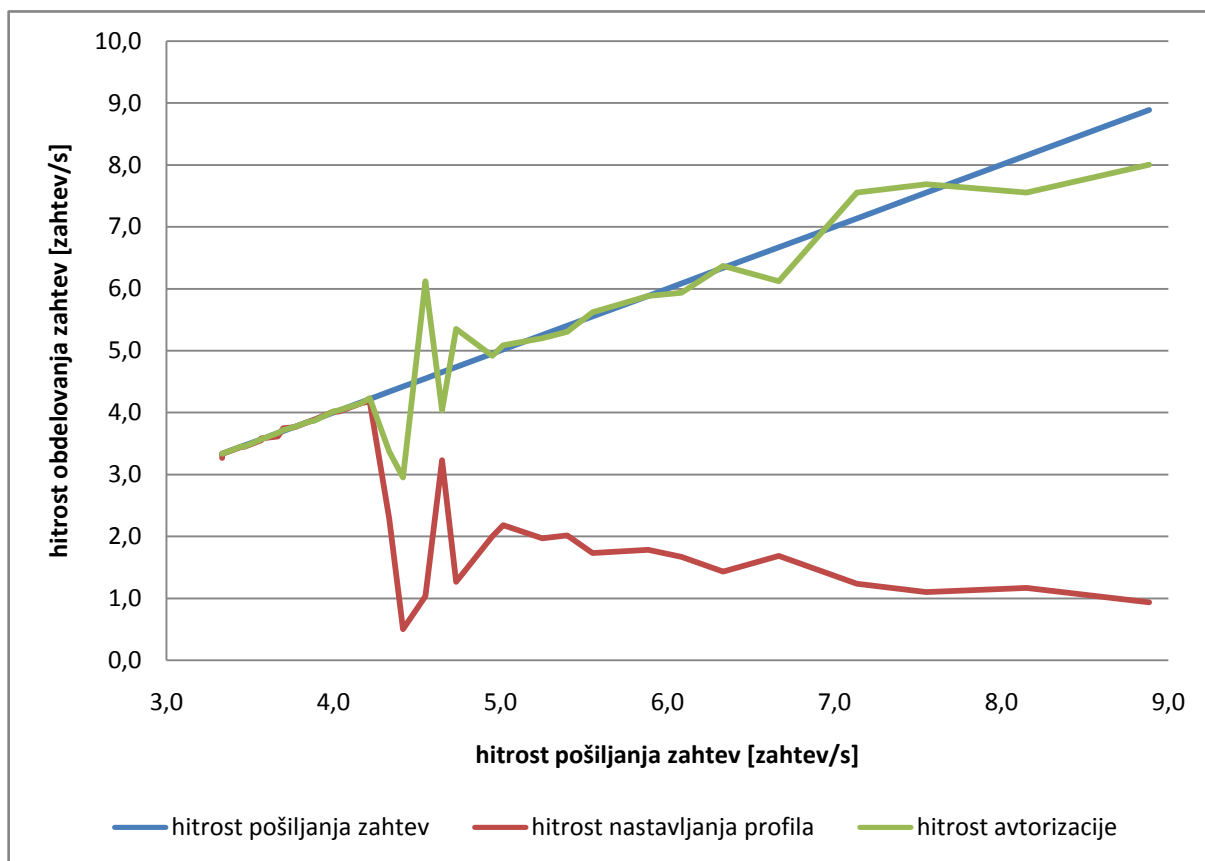
Prvi pokazatelj največje hitrosti je čakalna vrsta. Upoštevati moramo, da se zahteve za avtorizacijo in nastavljanje profila obdelujejo v ločenih bazenih niti in je zato potrebno preveriti oba. Potek obeh vrst je prikazan na sliki 6.14. Vrsta za nastavljanje profila se je začela naglo polniti, ko smo zahteve pošiljali s hitrostjo 4 zahteve na sekundo. Vrsta za avtorizacijo je bila večino časa skoraj prazna in kaže, da sistem pri

teh pogojih nima težav z izvajanjem avtorizacij. Hitrost registracije je, tako kot smo prej predvideli, odvisna od hitrosti nastavljanja profilov.



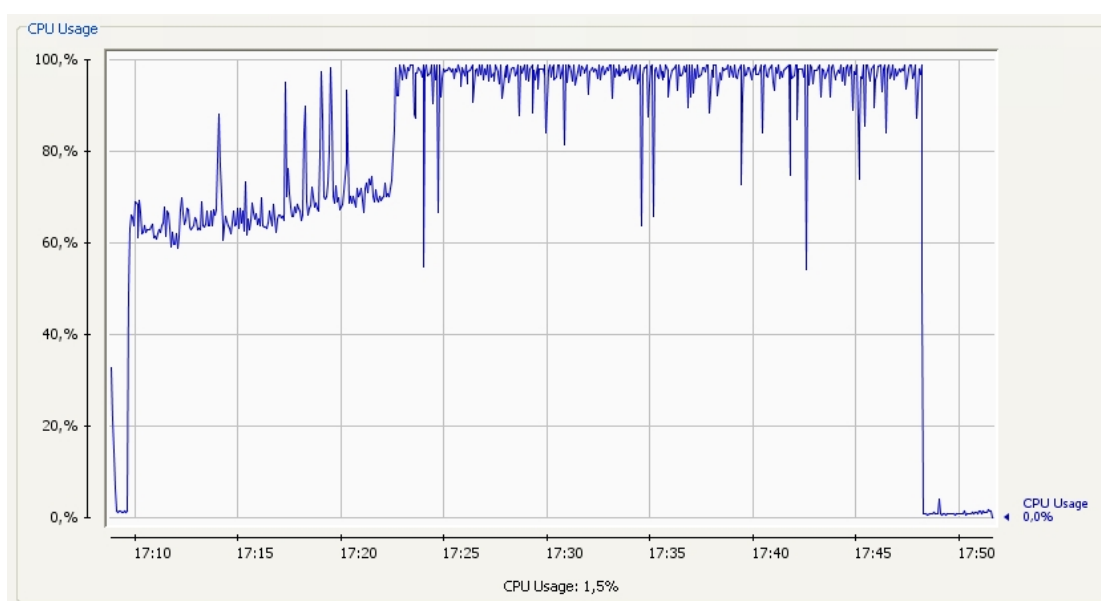
Slika 6.14 Število aktivnih niti in čakalna vrsta

Vpliv hitrosti pošiljanja zahtev na hitrost izvajanja registracije po kritični točki prikazuje slika 6.15. Najvišja hitrost je okoli 4 zahteve na sekundo. Povprečna hitrost nastavljanja profila po kritični točki pade na 2 zahtevi na sekundo in nakazuje, da se hitrost s povečanjem števila zahtev samo še znižuje. Avtorizacija ni kritična operacija, saj hitrost avtorizacije ne odstopa od hitrosti pošiljanja zahtev.



Slika 6.15 Hitrost posameznih faz v odvisnosti od hitrosti pošiljanja zahtev

Vzroke za tako delovanje smo že raziskali pri avtorizaciji in nastavljanju profila, kot dokaz vseeno prilagamo sliko 6.16 s porabo virov registracijskega strežnika, iz katerega se vidi, da je strežnik preobremenjen.



Slika 6.16: Poraba virov registracijskega strežnika.

7 Dimenzioniranje omrežja

Pri določanju velikosti omrežja, ki ga lahko testirani sistem nadzoruje, izhajamo iz pogoja, da mora sistem med prometno konico (angl.: Peak Busy Hour - PBH) obdelati vse registracijske zahteve.

Oceno, koliko registracijskih zahtev je poslanih med PBH, smo podali na podlagi števila aktivnih povezav med PBH in predpostavke, da se med PBH vse povezave vzpostavijo dvakrat. V tabeli 2 je prikazan pričakovan delež aktivnih povezav med PBH glede na tip uporabnikov [27].

Tip uporabnikov	Delež uporabnikov v omrežju	Delež aktivnih povezav med PBH
Poslovni uporabniki	50 %	20 %
Končni uporabniki	35 %	14 %
Občasni uporabniki	15 %	5 %
Povprečna vrednost		12,7 %

Tabela 2: Pričakovane zahteve med prometno konico

Uporabnike delimo na 3 tipe:

- Poslovni uporabniki so najbolj zahtevni uporabniki s stališča porabe širokopasovne povezave, saj le-to koristijo za izmenjavo datotek, video konference, elektronsko pošto... Predstavljajo polovico vseh uporabnikov. Med PBH je povezanih 20 odstotkov vseh poslovnih uporabnikov.
- Končni uporabniki uporabljajo širokopasovno povezavo predvsem za zasebno rabo (brskanje po spletu, izmenjavo datotek...). Predstavljajo 35 odstotkov uporabnikov. Med PBH je povezanih 14 odstotkov vseh končnih uporabnikov.
- Občasni uporabniki se v omrežje povežejo le za nekaj ur na dan predvsem za brskanje po spletu. Predstavljajo 15 odstotkov

uporabnikov. Med PBH je povezanih 5 odstotkov vseh občasnih uporabnikov.

Če upoštevamo delež različnih uporabnikov in število aktivnih povezav med PBH, dobimo podatek, da je med PBH povezanih 12,7 odstotkov vseh naročniških postaj. Ob predpostavki, da se vse aktivne naročniške postaje med PBH dvakrat povežejo v omrežje, mora registracijski strežnik v eni uri registrirati 25,4 odstotka vseh naročniških postaj. Končno število naročniških postaj, ki jih registracijski strežnik lahko nadzira, dobimo po naslednji enačbi.

$$\begin{aligned} \text{št. naročniških postaj} &= \frac{v}{dPBH} = \frac{14\,400 \frac{\text{naročniških postaj}}{h}}{0,254 \frac{1}{h}} \\ &= 56\,692 \text{ naročniških postaj} \end{aligned} \quad (7.1)$$

Oznaki pomenita:

V; hitrost registracije,

dPHB; delež vzpostavljenih povezav med PBH.

Testirani sistem pod enakimi pogoji, kot smo jih imeli med testiranjem, lahko upravlja približno 50 000 naročniških postaj.

8 Zaključek

S simulacijo smo določili zmogljivosti registracijskega strežnika. Hitrost smo določili za posamični registracijski fazi (avtorizacija in nastavitev profila) in za celoten postopek registracije. Testirani sistem zmore registrirati največ do 4 naročniške postaje na sekundo oziroma 14 400 naročniških postaj na uro. Meritev za posamične faze pa je pokazala, da sistem zmore avtorizirati do 24 naročniških postaj na sekundo in nastaviti profil do 6 naročniškim postajam na sekundo, kar nazorno kaže, da faza nastavljanja profila najbolj obremenjuje sistem.

Na podlagi najvišje hitrosti registracije smo podali tudi oceno velikosti omrežja, ki ga lahko ta sistem upravlja, da zadovolji potrebam med prometno konico. Pod enakimi pogoji, kot smo izvedli testiranje, bi lahko ta sistem upravljal do 50 000 naročniških postaj.

Zavedati se moramo, da so bili testi izvedeni pri skoraj idealnih pogojih. Vse omrežne naprave so delovale brezhibno in dostopni časi do njih so bili zanemarljivi, zato je v realnih sistemih pričakovati nižje hitrosti registracije in posledično mora biti tudi upravljanje omrežje manjše. Strežnik bo počasneje obdeloval posamične registracijske zahteve in aktiviral bo vse razpoložljive niti, kar smo ugotovili, da upočasni delovanje strežnika. Lahko bi zmanjšali število niti, ampak ne preveč, saj lahko že manjše število nepravilno delujočih naročniških postaj popolnoma onemogoči proces registracije s tem, da zasede vse razpoložljive niti. Najti je potrebno ustrezen kompromis med hitrostjo in zanesljivostjo sistema.

8.1 Kako izboljšati hitrost registracije

Obstaja več načinov večanja hitrosti registracije. Najenostavnejša rešitev bi bila uporaba hitrejšega računalnika za izvajanje registracije. Žal ta rešitev ni nujno cenovno najugodnejša.

Boljša rešitev je uporaba več računalnikov za registracijo, ki so povezani s skupno podatkovno bazo. Proces registracije je sedaj porazdeljen po več strežnikih, kar pozitivno vpliva na hitrost registracije. Tak sistem pa ima žal dve pomanjkljivosti. Prva

je zapletena konfiguracija sistema, saj je potrebno vsaki bazni postaji ročno nastaviti strežnik na katerega pošilja zahteve. Druga slabost je, da pri izpadu strežnika, registracija za bazne postaje, ki so vezane nanj, odpove.

Obe slabosti lahko premostimo z uporabo razdeljevalnika obremenitve. Vse bazne postaje nastavimo tako, da zahteve pošiljajo na razdeljevalnik obremenitve, s čimer poenostavimo in zmanjšamo možnost napak pri nastavitvah. Razdeljevalnik obremenitve po določenih algoritmih registracijske zahteve preusmerja na strežnike. Najenostavnejši algoritem za razporeditev dela je krožno razvrščanje, ki enakomerno porazdeljuje ukaze po strežnikih. S pomočjo drugih kompleksnejših algoritmov lahko dosežemo bolj optimalno obremenitev strežnikov. Razdeljevalnik obremenitve v določenih intervalih preverja delovanje strežnikov. V primeru, ko strežnik ne dela pravilno, preneha nanj pošiljati zahteve. Največja težava postavitve z razdeljevalnikom obremenitve je njegov izpad. Temu se lahko izognemo, da dodamo še en razdeljevalnik obremenitve, ki čaka v pripravljenosti in pri izpadu glavnega razdeljevalnika obremenitve prevzame njegovo funkcionalnost.

8.2 Ostale možnosti uporabe

Simulacija nam poleg določanja hitrosti registracije omogoča tudi druge možnosti uporabe. Simuliramo lahko izredne dogodke in opazujemo, kako se sistem obnaša ter ugotavljamo, ali je sistem sposoben vzpostaviti normalno stanje.

Pri optimizaciji postopka registracije lahko simulacijo uporabimo kot cenilko za določanje uspešnosti optimizacije.

Simulacija nam nudi možnost, da ocenimo vpliv novih funkcionalnosti na sistem s primerjavo delovanja pred in po uvedbi spremembe.

Z določenimi prilagoditvami je mogoče simulacijo uporabiti tudi za preizkušanje drugih nalog, ki jih opravlja NMS. Primera sta zbiranje statističnih podatkov in alarmov. Ocenimo lahko, za koliko naprav zbiramo podatke in kako večje število alarmov vpliva na delovanje sistema.

Posebna primera izrednih dogodkov, ki pa žal nista redkost v omrežjih držav v razvoju, je izpad električne energije ali omrežja v večjem delu mesta. Ob vzpostavitvi normalnega stanja se to pozna kot povečanje števila zahtev za registracijo. S simulacijskim programom tak primer ponazorimo v testnem okolju tako, da v zelo kratkem času pošljemo ogromno število zahtev. Pri tem preizkusu je pomembno ugotoviti, ali je sistem sposoben postaviti omrežje v normalno stanje.

Analiziramo lahko vpliv počasnega ali slabega delovanja mreže. Počasno delovanje simuliramo tako, da odgovore na zahteve ne pošiljamo takoj, ampak odgovor časovno zakasnimo za določen čas. Zamik je lahko naključno izbran iz določenega časovnega intervala. Slabo delovanje omrežja pomeni, da se paketi pri prenosu od vira do cilja izgubljajo. Izguba paketov se ponazori tako, da za določen delež naključnih paketov simulacija ne bi poslala odgovora.

Možnost imamo simuliranja nepravilnega delovanja določenih naročniških postaj, s tem da za zahteve, ki prihajajo od določene postaje, izvedemo posebno akcijo.

Zgoraj naštetih so le nekateri primeri uporabe. Simulacijo je možno porabiti še za mnogo drugih situacij.

Literatura

- [1] Nuaymi Loutfi: WiMAX: Technology for Broadband Wireless Access, John Wiley & Sons, 2007.
- [2] Frequently Asked Questions, WiMAX Forum:
<http://www.wimaxforum.org/resources/frequently-asked-questions>, dostopnost preverjena maja 2009.
- [3] Telekom Slovenije komercialno uvedel Wimax (18.03.2008), Siol:
http://www.siol.net/tehnologija/telekomunikacije/2008/03/telekom_slovenije_komercialno_uvedel_wimax.aspx, dostopnost preverjena maja 2009.
- [4] Razpisna dokumentacija, Javni razpis za dodelitev radijski frekvenc za fiksni brezžični širokopasovni dostopovni system, APEK, 2006.
(http://www.appek.si/sl/datoteke/File/javni%20razpisi/razpisni_pogoji-12.pdf, dostopnost preverjena maja 2006)
- [5] Eckel Bruce, Thinking in Java, Third edition, Prentice Hall, 2003.
- [6] Gosling James, Joy Bill, Steele Guy, Bracha Gilad, The Java Language Specification, Third Edition, Addison-Wesley, 2005.
(<http://java.sun.com/docs/books/jls/download/langspec-3.0.pdf>, dostopnost preverjena maja 2009)
- [7] Podatkovna baza, Wikipedija Prosta enciklopedija:
http://sl.wikipedia.org/wiki/Podatkovna_baza, dostopnost preverjena maja 2009.
- [8] Oppel Andrew J., Databases Demystified, McGraw-Hill Professional, 2004.
- [9] Bauer Christioan, King Gavin, Hibernate in Action, Manning, 2005.
- [10] Kofler Michael, The Definitive Guide to MySQL 5, Third edition, Apress, 2005.
- [11] Greenwald Rick, Stackowiak Robert, and Stern Jonathan: Oracle Essentials, Forth edition, O'Reilly, 2007.
- [12] GNU GENERAL PUBLIC LICENSE:
<http://www.fsf.org/licensing/licenses/gpl.html>, dostopnost preverjena junija 2009.
- [13] Bourke Tony: Server Load Blancing, O'Reilly, 2001.
- [14] Grosso William, Java RMI, O'Reilly, 2001.
- [15] JDBC: <http://java.sun.com/javase/6/docs/technotes/guides/jdbc/>, dostopnost preverjena junija 2009.

- [16] Oracle Database, Net Services Administrator's Guide, 10g Release 2, 2005.
(http://download.oracle.com/docs/cd/B19306_01/network.102/b14212.pdf,
dostopnost preverjena maja 2009)
- [17] Mauro Douglas, Schmidt Kevin: Essential SNMP, Second edition, 2005.
- [18] RFC 2578: <http://www.ietf.org/rfc/rfc2578.txt>, dostopnost preverjena maja 2009.
- [19] RFC 3411: <http://www.ietf.org/rfc/rfc3411.txt>, dostopnost preverjena maja 2009.
- [20] RFC 3418: <http://www.ietf.org/rfc/rfc3418.txt>, dostopnost preverjena maja 2009.
- [21] Using Netsh, Microsoft TechNet: <http://technet.microsoft.com/en-us/library/bb490939.aspx>, dostopnost preverjena maja 2009.
- [22] N. van Kempen Fred, Cox Alan, Blundell Phil, Kleen Andi, ifconfig(8) - Linux man page: <http://linux.die.net/man/8/ifconfig>, dostopnost preverjena maja 2009.
- [23] AdventNet: <http://www.adventnet.com/>, dostopnost preverjena maja 2009.
- [24] Java Platform, Standard Edition 6 API Specification:
<http://java.sun.com/javase/6/docs/api/index.html>, dostopnost preverjena maja 2009.
- [25] Thread pool pattern, Wikipedija The Free Encyclopedia:
http://en.wikipedia.org/wiki/Thread_pool_pattern, dostopnost preverjena maja 2009.
- [26] Unix Command top, About.com:
http://linux.about.com/od/commands//blcmdl1_top.htm, dostopnost preverjena maja 2009.
- [27] Mwesiga W. Barongo, Dimensioning Mobile WIMAX in the Access and Core Network: A case Study, Magistersko delo, Helsinki University of Technology, Faculty of Electronics, Communications and Automation, 2008.
(http://kambing.ui.edu/onnopurbo/library/library-ref-eng/ref-eng-3/physical/wimax/2007/mobile_wimax_deployment_alternatives.pdf, dostopnost preverjena maja 2009)

Izjava

Izjavljam, da sem diplomsko delo izdelal samostojno pod vodstvom mentorja doc. dr. Boštjana Murovca, univ. dipl. inž. el.

Mojmir Tuljak