

Analiza zmogljivosti prehoda WebService/Konnex

Mark Umberger

*Entia d.o.o., Tržaška 2, 1000 Ljubljana, Slovenija
e-pošta: mark.umberger@entia.si*

Povzetek. V zadnjem času so se močno uveljavili sistemi in storitve za upravljanje procesov v zgradbah (UPZ), ki temeljijo na konceptu računalništva v oblaku, kar telekomunikacijskim operaterjem omogoča uvajanje novih M2M storitev s področja inteligentnega doma. Ker storitve UPZ delujejo na podlagi specifičnih komunikacijskih standardov (Konnex), ki se razlikujejo od standarda IP, na podlagi katerega deluje koncept računalništva v oblaku, je treba za zagotovitev povezljivosti sistemov UPZ zagotoviti ustrezen prehod IP/UPZ (IP/Konnex). Tako se storitve UPZ izvajajo porazdeljeno na strani operaterja, zato se zastavi vprašanje glede vpliva omrežnega prometa na kakovost izvajanja storitev UPZ. Ker omrežja UPZ delujejo na bistveno manjši pasovni širini (9600 bit/s) kot sodobna širokopasovna omrežja (več kot 100 Mbit/s), lahko sklepamo, da je »ozko grlo« prehod IP/Konnex, na katerega ima omrežni promet, ki ga ustvarijo uporabniki storitev UPZ, največji vpliv. Kakovosti izvajanja posameznih storitev UPZ so torej odvisne od časa izvajanja storitev UPZ na prehodu pri različnih stopnjah prometa. Iz tega izhaja motivacija za raziskavo, ki se nanaša na analizo zmogljivosti prehoda WebService/Konnex pri izvajanju različnih storitev UPZ (write group in read group service) v sistemih UPZ, ki temeljijo na konceptu računalništva v oblaku. Iz rezultatov eksperimenta ugotovimo, da prehod omogoča zadovoljivo delovanje pri vrednostih prometa $n = (0, 42]$ ukazov/s za storitve write group service ter $n = (0, 20]$ ukazov/s za storitve read group service, pri večjih vrednostih prometa pa je delovanje prehoda nezadovoljivo.

Ključne besede: UPZ, računalništvo v oblaku, analiza zmogljivosti prehoda, obhodni čas, število izgubljenih paketov

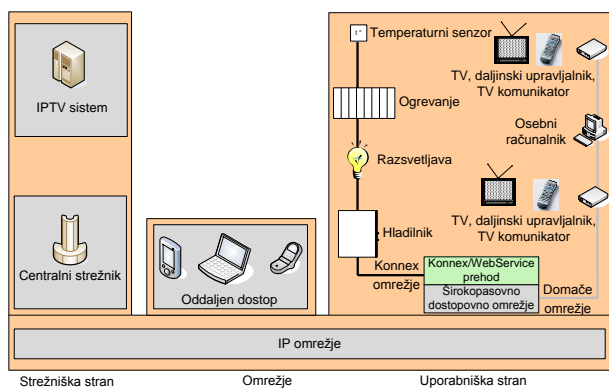
Performance analyses of WebService/Konnex gateway

Building automation and control systems (BACS) based on cloud computing is becoming very popular. Using these systems telcos is likely to provide additional M2M services for smart homes. The BACS services function on the basis of specific communication standards (i.e. Konnex) which are different from the IP standards. In order to ensure IP connectivity an IP/BACS (IP/Konnex) gateway must be provided. The cloud computing approach of the BACS allows for a distributed performance of all services by the operator, due to which the issue regarding the influence of network traffic on the quality of experience of the BACS services arises. Because BACS networks operate at much lower bandwidth (9600 bit/s) than modern broadband networks (more than 100Mbit/s) we assume that the bottleneck represents the BACS/IP gateway on which the network traffic created by the BACS service users has the greatest impact. The quality of individual BACS services therefore depends on the BACS services performance time on the gateway at different traffic levels. Our research refers to the analysis of the WebService/Konnex gateway performance in executing various BACS services (write and read group service) within BACS system based on cloud computing. Based on the experimental results we find that satisfactory performance of gateway is at traffic values $n = (0, 42]$ commands/s for write group service and $n = (0, 20]$ commands/s for the read group service, while the WebService/Konnex gateway performance at greater traffic values is inadequate.

1 UVOD

V zadnjih letih so se močno uveljavili sistemi in storitve za upravljanje procesov v zgradbah (UPZ), ki temeljijo na konceptu računalništva v oblaku [16]. Tako ni več treba imeti strežnika v vsakem domu posebej, ampak se lahko uporabi skupni strežnik za več domov hkrati, ki se nahaja v Internet omrežju oziroma pri operaterju, kar prikazuje slika 1. Ker sistemi in storitve UPZ delujejo na podlagi specifičnih komunikacijskih standardov in protokolov (npr. LonWorks, Konnex, CAN, BACnet, Z-wave itd.), ki se razlikujejo od standarda IP, na podlagi katerega deluje koncept računalništva v oblaku, je treba za zagotovitev povezljivosti IP zagotoviti ustrezen prehod UPZ/IP. Obstajajo številne standardizirane različice sistemov in storitev UPZ (npr. Konnex, LonWorks, BACnet itd.), ki omogočajo medsebojno povezovanje in komunikacijo s pomočjo različnih komunikacijskih protokolov. Glede na vsestransko uporabo, fleksibilnost in prevladujoč vpliv v Evropi smo se v raziskavi na področju UPZ strogo osredotočimo na standard Konnex.

Predlagani konvergenčni arhitekturni model omogoča porazdeljeno izvajanje vseh storitev na strani operaterja, zato se zastavi vprašanje glede vpliva



Slika 1: Konvergenčni arhitekturni model

omrežnega prometa na kakovost storitev UPZ. Ker omrežja UPZ delujejo na bistveno nižji pasovni širini (9600 bit/s) kot sodobna širokopasovna omrežja (več kot 10Mbit/s), lahko sklepamo, da je »ozko grlo« v konvergenčnem arhitekturnem modelu prehod WebService/Konnex, na katerega ima omrežni promet, ki ga ustvarijo uporabniki storitev UPZ, največji vpliv. Kakovosti posameznih storitev UPZ so torej odvisne od časa izvajanja storitev UPZ na prehodu pri različnih stopnjah prometa. Posamezen prehod je lahko nameščen v vsakem domu posebej, lahko pa se uporabi prehod za več domov hkrati. Cilj vseh ponudnikov internetnih storitev je uporabiti minimalno število prehodov, s pomočjo katerih je mogoče doseči zadovoljivo kakovost uporabniške izkušnje posameznih storitev UPZ. Vprašanje je, koliko domov je mogoče upravljati s pomočjo enega prehoda, kar pa je odvisno od odzivanja prehoda na različni omrežni promet, ki nastane pri uporabi storitev UPZ. Iz tega izhaja motivacija za raziskavo, ki se nanaša na analizo zmogljivosti prehoda WebService/Konnex pri izvajanju različnih storitev UPZ v predlaganem konvergenčnem arhitekturnem modelu, kar podrobno obravnavamo v tem članku. Konkretno bomo analizirali zmogljivost prehoda WebService/Konnex, ki smo ga uporabili v predlaganem konvergenčnem arhitekturnem modelu za povezavo med omrežji TCP/IP in UPZ. Ker je prehod sestavljen iz Konnex in WebService dela, bomo analizirali zmogljivost vsakega dela prehoda posebej.

2 PRETEKLE ŠTUDIJE

V preteklosti so bile narejene številne sorodne študije, ki obravnavajo analizo zmogljivosti različnih prehodov, ki omogočajo povezavo med omrežji TCP/IP in UPZ. V študiji [1], [2] avtorji podajajo analizo zmogljivosti prehoda Profibus/ATM. Profibus [3], [4] je komunikacijski standard, ki se uporablja za komunikacijo pri upravljanju procesov v industriji, obstajajo pa tudi primeri uporabe komunikacijskega standarda v zgradbah. V osnovi standard temelji na komunikacijskem modelu, ki ima definiran le aplikacijski sloj, podatkovni sloj in fizični sloj. Profibus omogoča komunikacijo s pomočjo dveh različnih

fizičnih komunikacijskih standardov za prenos podatkov; serijske komunikacije in komunikacije Manchester Coding Bus Powered [14]. Problem pri uporabi standarda Profibus je, da ima zaradi navedenih komunikacijskih standardov izjemno omejeno dolžino fizičnega medija, s čimer je zelo omejeno delovanje aplikacij v realnem času, zato avtorji v študiji [2] predlagajo uporabo prehoda Profibus/ATM. Z uporabo omrežja ATM je tako omogočeno delovanje aplikacij v realnem času in odpravljen problem dolžine Profibus fizičnega medija.

Analiza zmogljivosti prehoda Profibus/ATM v študiji [2] se nanaša na eksperimentalno merjenje zmogljivosti prehoda s pomočjo merjenja povprečnega obhodnega časa sporočila pri različnih vrednostih prometa na njem. Za podrobnejšo analizo avtorji analizirajo posebej Profibus in posebej za ATM del prehoda. Prav tako avtorji podajajo odvisnost med številom izgubljenih paketov in prometom. Problem prehoda Profibus/ATM pa je, da je Profibus v osnovi komunikacijski standard, ki se večinoma uporablja za upravljanje procesov v industriji, in ni široko uporaben v zgradbah. Naslednji problem pa je uporaba omrežij ATM, katerih uporaba znatno pada. V študiji [5] avtorji obravnavajo zmogljivost prehoda CAN/CAN. CAN [6] je komunikacijski standard, ki omogoča visoko hitrost prenosa s pomočjo serijskega vodila in je bil v osnovi narejen za komunikacijo med aplikacijami v avtomobilski industriji, obstajajo pa tudi primeri uporabe standarda CAN za UPZ. Odlična lastnost protokola CAN je, da zagotavlja visoko raven varnosti [7]. Problem omrežij CAN pa je omejitev dolžine fizičnega medija. Zato avtorji v [5] predlagajo rešitev problema s pomočjo delitve omrežja CAN v več segmentov, ki medsebojno komunicira s pomočjo omrežja Ethernet. Posamezni segmenti omrežja CAN pa se povežejo z omrežjem Ethernet s pomočjo predlaganega prehoda CAN/CAN. Avtorji v [5] obravnavajo zmogljivost prehoda CAN/CAN s pomočjo izdelane simulacije, kjer podajajo analizo vmesnikov CAN na prehodu pri različnih stopnjah prometa na njem s pomočjo merjenja povprečnega obhodnega časa izvajanja storitev standarda CAN. Prav tako avtorji podajajo odvisnost med številom izgubljenih paketov in stopnjo prometa na prehodu CAN/CAN. Zgoraj navedena študija [5] in študije [9], [10], [11], [12] omogočajo odpravo problematike glede omejitve dolžine fizičnega medija omrežja CAN tako, da navedeni prehodi omogočajo izvajanje funkcije ojačevalnika (angl. repeater), kar pa odpravlja problem le na lokalni ravni, in tako ne omogočajo rešitve za odpravo problema v porazdeljenih sistemih za UPZ.

Problem pa je mogoče popolnoma odpraviti z uporabo prehoda CAN/Ethernet [13] in prehoda CAN/ATM [15]. V študiji [15] avtorji podajajo tudi analizo zmogljivosti prehoda CAN/ATM s pomočjo modela prehoda, ki ga izdelajo s pomočjo simulacijskega orodja Network 2.5. Avtorji v študiji podajajo tudi odvisnost med obhodnim časom izvajanja

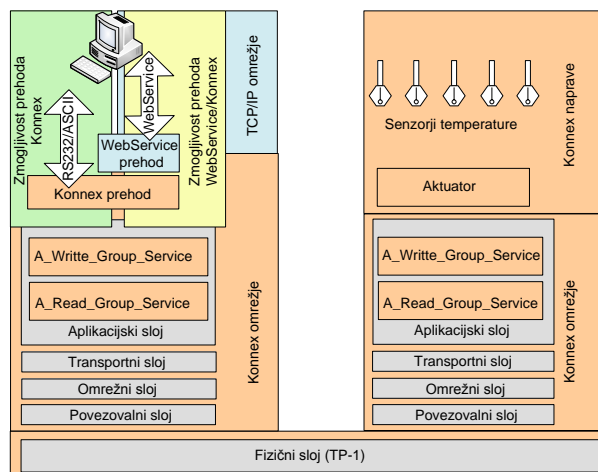
posameznih storitev standarda CAN pri različnih vrednostih prometa za vsak del prehoda posebej (CAN in ATM) in število izgubljenih paketov na prehodu pri različnih vrednostih prometa. Problem študij [2], [5], [13], [15] pa je v tem, da rezultati zmogljivosti prehodov temeljijo zgolj na simulaciji delovanja modelov posameznih prehodov, torej znatno primanjkuje praktična izvedba prehodov Profibus/ATM, CAN/Ethernet in CAN/ATM, na katerih bi bilo mogoče eksperimentalno izvesti analizo zmogljivosti posameznih prehodov. Modeli posameznih prehodov so sicer lahko dober približek realnemu delovanju prehodov, vendar je treba njihovo točnost ovrednotiti s pomočjo primerjave s podatki, pridobljenimi iz eksperimenta na realnem prehodu. Iz zgornjih raziskav je razvidno, da se nobena ne nanaša na analizo zmogljivosti prehoda Webservice/Konnex, zato bomo to podrobno obravnavali v doktorski disertaciji. V nasprotju s študijami [2], [5], [13], [15] bomo zmogljivost prehoda WebService/Konnex analizirali s pomočjo eksperimenta na realnem prehodu in ne zgolj na modelu prehoda oziroma simulaciji njegovega delovanja. Podobno kot v študijah [2], [15] bomo posebej obravnavali zmogljivost WebService in Konnex dela prehoda.

3 METODOLOGIJA

Glavni namen raziskave je analizirati zmogljivost prehoda WebService/Konnex pri različnih stopnjah prometa, ki nastaja pri uporabi storitev UPZ. V tem poglavju najprej predstavimo eksperimentalno okolje, s pomočjo katerega smo izmerili zmogljivost prehoda. V nadaljevanju poglavja nato podajamo podroben opis storitev, ki ustvarjajo promet na prehodu, ter v naslednjem koraku navedemo vse spremenljivke, ki jih v eksperimentu obravnavamo. V nadaljevanju nato podrobno predstavimo tudi postopek eksperimenta.

3.1 Eksperimentalno okolje

Prehod, katerega zmogljivost analiziramo, v osnovi omogoča izvajanje storitev UPZ, ki delujejo na podlagi protokolnega sklada Konnex, s strani spletnih naprav, ki delujejo na podlagi protokolnega sklada TCP/IP. Zmogljivost prehoda se v eksperimentu nanaša na merjenje obhodnega časa določene storitve UPZ od trenutka, ko se na prehodu pojavi zahteva po izvajanju določene storitve UPZ s strani omrežja TCP/IP, do trenutka, ko prehod prejme od naprave Konnex potrdilo o uspešnosti izvedbe storitve in ga nato posreduje na omrežje TCP/IP. Eksperimentalno okolje za merjenje zmogljivosti prehoda prikazuje slika 2. Ker je prehod sestavljen iz dveh delov (Konnex dela in WebService dela) [16], smo zmogljivost prehoda analizirali v dveh ločenih delih. Najprej smo analizirali Konnex del prehoda, kot prikazuje zeleni kvadrat na sliki 2, in nato še celoten prehod WebService/Konnex, kot prikazuje rumeni kvadrat na sliki 2.



Slika 2: Eksperimentalno okolje za merjenje zmogljivosti prehoda

Eksperimentalno okolje za merjenje zmogljivosti prehoda Konnex sestavljajo osebni računalnik, prehod Konnex in omrežje Konnex z napravami. Osebni računalnik je fizično povezan s prehodom Konnex s pomočjo serijskega vodila. Na osebni računalnik pa je nameščena programska aplikacija, ki omogoča pošiljanje in sprejemanje storitev Konnex na aplikacijskem sloju na serijsko vodilo v formatu ASCII. Torej glavna naloga prehoda Konnex je, da omogoča fizično povezavo na eni strani z osebnim računalnikom s pomočjo serijskega vodila in na drugi z omrežjem Konnex ter da omogoča pretvorbo storitev Konnex na aplikacijski ravni iz formata ASCII v format Konnex, kar je tudi razvidno iz zelenega kvadrata na sliki 2. Glavna naloga programske aplikacije pa je, da omogoča merjenje obhodnega časa, ki preteče med trenutkom, ko aplikacija pošlje sporočilo na prehod Konnex, ki vsebuje zahtevo po izvajanju določene storitve Konnex (angl. request service), in trenutkom, ko aplikacija prejme od prehoda Konnex sporočilo, ki vsebuje informacijo o potrditvi izvedbe določene storitve Konnex (angl. response service). Ta odgovor je seveda lahko pozitiven ali negativen. Ko prehod Konnex prejme sporočilo, ki vsebuje zahtevo po izvajanju določene storitve Konnex v formatu ASCII, jo pretvori v format Konnex sporočila, ki jo zahtevajo storitve aplikacijskega sloja protokolnega sklada Konnex, in nato pošlje na napravo Konnex, kjer se storitev Konnex tudi izvede. V eksperimentalnem okolju so naprave Konnex krmilnik za vklapljanje/izklapljanje šestih tokokrogov razsvetljave in petih merilnikov temperature.

Kot je razvidno iz slike 2, pa eksperimentalno okolje za merjenje zmogljivosti prehoda Webservice/Konnex sestavljajo osebni računalnik, prehod WebService/Konnex in omrežje Konnex z napravami. Podobno kot v prejšnjem primeru je na osebni računalnik nameščena programska aplikacija, ki pa ne pošilja ukazov na serijsko vodilo v formatu ASCII kot v prejšnjem primeru, ampak na spletni strežnik v obliki

SOAP/XML, ki je nameščen na strojni opremi Alix. Strojna oprema Alix torej pomeni del prehoda WebService, saj omogoča s pomočjo nameščene programske aplikacije sprejemanje/oddajanje storitev Konnex v SOAP/XML obliki s pomočjo Ethernet-vhoda in njihovo pretvorbo ter sprejemanje/pošiljanje v formatu ASCII na serijsko vodilo oziroma na prehod Konnex. Podobno kot v prejšnjem primeru pa je naloga programske aplikacije na osebнем računalniku, da omogoča merjenje obhodnega časa, ki preteče med trenutkom, ko aplikacija pošlje sporočilo na prehod WebService/Konnex v obliki SOAP/XML, ki vsebuje zahtevo po izvajanju določene storitve Konnex, in trenutkom, ko aplikacija prejme od prehoda sporočilo WebService/Konnex, ki vsebuje odgovor o potrditvi izvedbe določene storitve Konnex v obliki SOAP/XML. Ko prehod Konnex prejme sporočilo od prehoda WebService/Konnex, ki vsebuje zahtevo po izvajanju določene storitve Konnex v formatu ASCII, ga pretvori v format Konnex, ki ga zahtevajo storitve aplikacijskega sloja protokolnega sklada Konnex, in nato pošlje na napravo Konnex, kjer se storitev Konnex tudi izvede. V eksperimentalnem okolju so naprave Konnex krmilnik za vklop/izklop šestih tokokrogov razsvetljave in petih merilnikov temperature. V nasprotju s primerom, kjer obravnavamo le del prehoda Konnex, je torej pri analizi zmogljivosti prehoda WebService/Konnex obhodni čas daljši za čas, ki nastane pri pretvorbi podatkov na delu prehoda WebService.

3.2 Analizirane storitve

Pri uporabi storitev UPZ v predlaganem konvergenčnem arhitekturnem modelu na sliki 1 s pomočjo enega od definiranih uporabniških vmesnikov se zahteva po izvajanju posamezne storitve UPZ najprej pošlje na centralni strežnik, ki je na strani operaterja, ter nato na prehod WebService/Konnex in nato na omrežje Konnex. Posamezno storitev UPZ pa je mogoče sprožiti oziroma izvesti s pomočjo različnih storitev aplikacijskega sloja protokolnega sklada Konnex, zato v tem poglavju izpostavimo vse storitve, ki jih v eksperimentu analiziramo. Naprave Konnex medsebojno izmenjujejo podatke po fizičnem vodilu s pomočjo okvirjev (angl. frames). Obstajata dve vrsti okvirjev, podatkovni okvir (angl. data frame) in potrdilni okvir (angl. acknowledgement frame). Pošiljatelj pošlje podatkovni okvir prejemniku takrat, ko nastane potreba po izvajanju določene storitve UPZ (npr. vklop/izklop luči – ob pritisku na določeno tipko na omrežju Konnex, ki v tem primeru pomeni pošiljatelja, le-ta najprej pošlje podatkovni okvir na krmilnik, ki nato omogoči vklop/izklop določene luči in je v tem primeru prejemnik). Potrdilni okvir pa vsebuje informacijo o uspešnosti izvedbe določene storitve UPZ (npr. vklop/izklop luči – krmilnik pošlje potrdilni okvir tipki, da je bila luč uspešno ali neuspešno vklopljena).

Podatkovni okvir sestavlja sedem bitnih polj različnih velikosti, in sicer *control field* (1 oktet), *source address* (2 oktet), *destination address* (2 okteta + 1 bit), *routing counter* (3 bit), *length* (4 bit), *user data* (1–16 oktetov) in *checksum field* (1 oktet). Funkcionalnost posameznih bitnih polj ni pomembna za potrebe našega eksperimentalnega dela, zato jih ne bomo opisovali, so pa podrobneje predstavljena v [18] in [17]. Iz strukture podatkovnega okvirja je razvidno, da je najkrajši podatkovni okvir sestavljen iz 9 oktetov (angl. bytes), najdaljši pa iz 23 oktetov. Dolžina podatkovnega okvirja pa je odvisna od posamezne storitve, ki jo definira protokolni sklad Konnex [17]. V eksperimentu analiziramo zmogljivost prehoda WebService/Konnex s pomočjo dveh storitev aplikacijskega sloja protokolnega sklada Konnex (*write group services* in *read group service*), ki omogočata izvajanje najpogosteje uporabljenih in najbolj reprezentativnih storitev UPZ in zato predstavljata analizirani storitvi v našem eksperimentu. Storitve *write group service* omogoča izvajanje najrazličnejših storitev UPZ (npr. diskretno upravljanje razsvetljave, zvezno upravljanje razsvetljave, upravljanje senčil, upravljanje ogrevanja in hlajenja, upravljanje močnostnih porabnikov itd.). Storitve *read group service* pa omogoča branje vrednosti različnih veličin iz naprav Konnex (npr. temperatura, osvetljenost, vlažnost itd.).

3.3 Spremenljivke

Spremenljivke v eksperimentu delimo na odvisne in neodvisne. Namen eksperimenta je izmeriti obhodni čas storitev *write group service* in *read group service* na prehodu WebService/Konnex pri različnih stopnjah prometa na njem. Obhodni čas se torej nanaša na čas, ki preteče medtem, ko programska aplikacija na sliki 2 pošlje zahtevo po izvajanju storitve *write group service* ali *read group service* na prehod WebService/Konnex, in medtem, ko programska aplikacija prejme povratno informacijo, ki vsebuje ali podatke o uspešnosti izvedbi storitve (v primeru storitve *write group service*) ali pa prejme želeno vrednost (v primeru storitve *read group service*). Zaradi boljše interpretacije rezultatov eksperimenta, ki se nanašajo na merjenje obhodnega časa pri različnih vrednostih prometa na prehodu, pa se v eksperimentu osredotočimo tudi na število izgubljenih paketov, ki se pojavi pri posamezni vrednosti prometa. V eksperimentu se meri obhodni čas pri različnih vrednostih prometa posebej za del prehoda Konnex in posebej za celoten prehod WebService/Konnex. Zato obhodni čas zato predstavlja odvisno spremenljivko in promet neodvisno spremenljivko v eksperimentu. Obhodni čas se meri s pomočjo časovne enote v sekundah, promet pa pomeni določeno število ukazov na sekundo in je izražen s pomočjo relacije $n = \text{št. ukazov/t}$. Prav tako se v eksperimentu meri število izgubljenih paketov pri različnih stopnjah prometa posebej za del prehoda Konnex in posebej za celoten prehod WebService/Konnex. Število izgubljenih

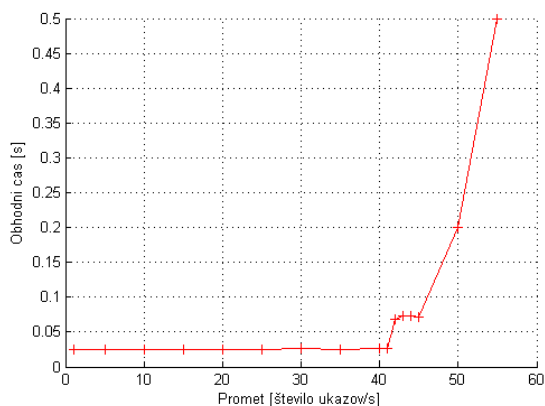
paketo zato upoštevamo kot odvisno spremenljivko in promet kot neodvisno spremenljivko. Število izgubljenih paketov se meri s pomočjo relativne vrednosti, ki vsebuje podatek o razmerju med izgubljenimi paketi v primerjavi z vsemi poslanimi paketi.

3.4 Potek izvajanja eksperimenta

Eksperiment, ki ga prikazuje slika 2, je bil izveden tako, da smo na programski aplikaciji določili vrednost prometa in merili obhodni čas. Promet smo na začetku nastavili tako, da je bil časovni razmik med dvema ukazoma 200 ms, kar pomeni, da je bil v tem primeru promet $n = 5$ ukazov/s. V nadaljevanju smo promet povečevali za vrednost 5. Pri vsakem posameznem prometu smo izvajali merjenje na podlagi 2000 poslanih ukazov in merili posamezen obhodni čas vsake interakcije posebej. Tovrstni postopek izvajanja eksperimenta smo uporabili tako za analizo storitve *write group service* kot tudi storitve *read group service*. Razlika pa je bila v tem, da je programska aplikacija pri analizi storitve *write group service* posamezne zahteve za izvedbo storitve zaporedno pošiljala na pet različnih izhodov na krmilniku, ki je nato omogočal vklop/izklop petih svetilk. Pri analizi storitve *read group service* pa je programska aplikacija zaporedno pošiljala zahteve za branje vrednosti na pet različnih temperaturnih senzorjev, ki so nato nazaj pošiljali trenutno izmerjeno vrednost.

4 REZULTATI

Merjenje zmogljivosti prehoda Webservice/Konnex se nanaša na merjenje obhodnega časa storitev *write group service* in *read group service* na prehodu Webservice/Konnex pri različnih stopnjah prometa na njem. Obhodni čas se torej nanaša na čas, ki preteče medtem, ko programska aplikacija na sliki 16 pošlje zahtevo po izvajanju storitve *write group service* ali *read group service* na prehod Webservice/Konnex, in medtem, ko programska aplikacija prejme povratno informacijo, ki vsebuje ali informacijo o uspešni

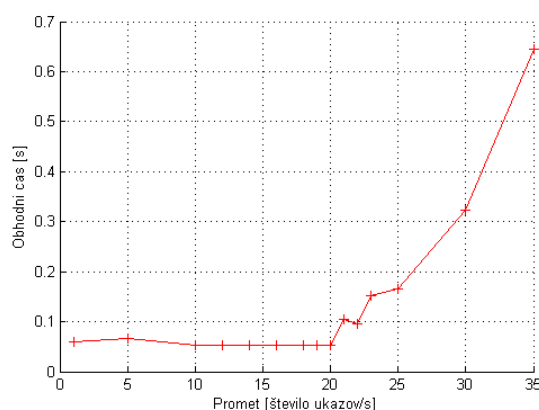


Slika 3: Obhodni čas pri izvajanju storitve *write group service* pri različnih stopnjah prometa na prehodu Konnex

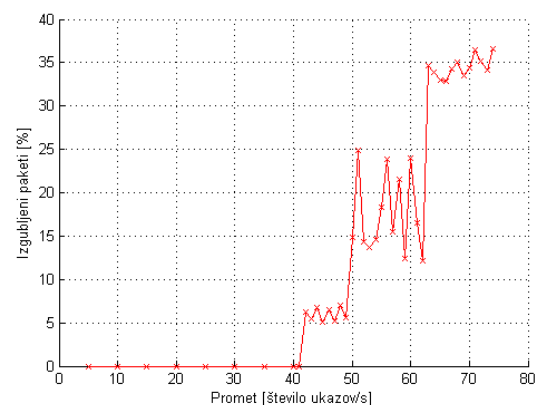
oziroma neuspešnosti izvedbi storitve (pri storitvi *write group service*), ali pa prejme želeno vrednost (pri storitvi *read group service*). Prav tako pa se rezultati nanašajo na merjenje števila izgubljenih paketov pri izvajanju posamezne storitve (*write group service* in *read group service*). Ker smo v eksperimentu analizirali vsak del prehoda posebej ločeno, podajamo rezultate za del prehoda Konnex in ločeno za celoten prehod Webservice/Konnex.

4.1 Zmogljivost prehoda Konnex

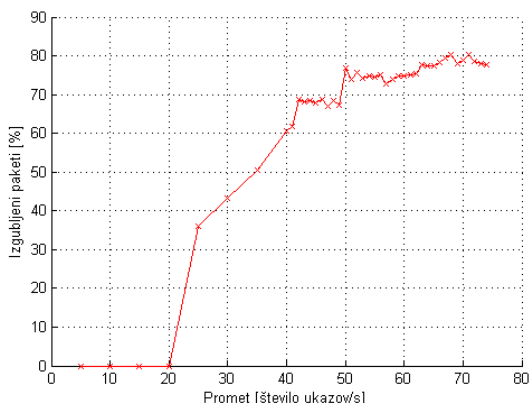
V tem poglavju predstavimo rezultate, ki se nanašajo na merjenje obhodnega časa pri različnih vrednostih prometa na prehodu Konnex pri izvajanju storitve *write group service* (slika 3) in *read group service* (slika 4). Prav tako v tem poglavju predstavimo rezultate, ki se nanašajo na merjenje števila izgubljenih paketov pri različnih vrednostih prometa na prehodu Konnex pri izvajanju storitve *write group service* (slika 5) in *read group service* (slika 6).



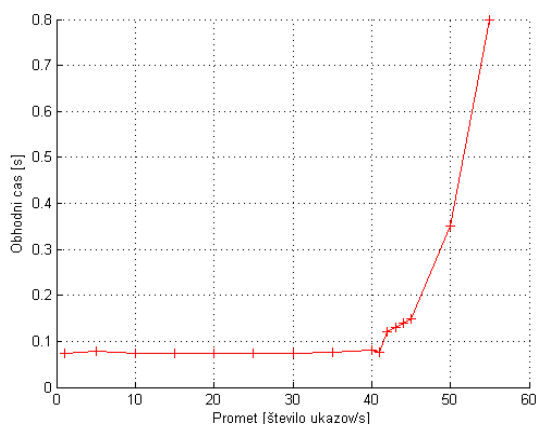
Slika 4: Obhodni čas pri izvajanju storitve *read group service* pri različnih stopnjah prometa na prehodu Konnex



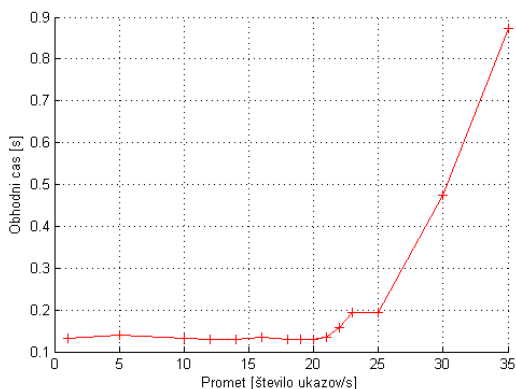
Slika 5: Izgubljeni paketi pri izvajanju storitve *write group service* pri različnih stopnjah prometa na prehodu Konnex



Slika 6: Izgubljeni paketi pri izvajanju storitve *read group service* pri različnih stopnjah prometa na prehodu Konnex



Slika 7: Obhodni čas pri izvajanju storitve *write group service* pri različnih stopnjah prometa na prehodu WebService/Konnex

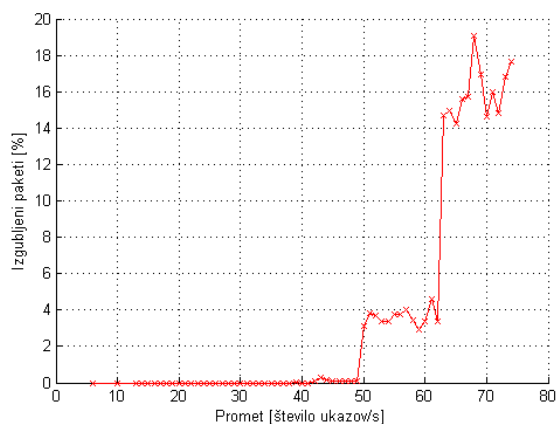


Slika 8: Obhodni čas pri izvajanju storitve *read group service* pri različnih stopnjah prometa na prehodu WebService/Konnex

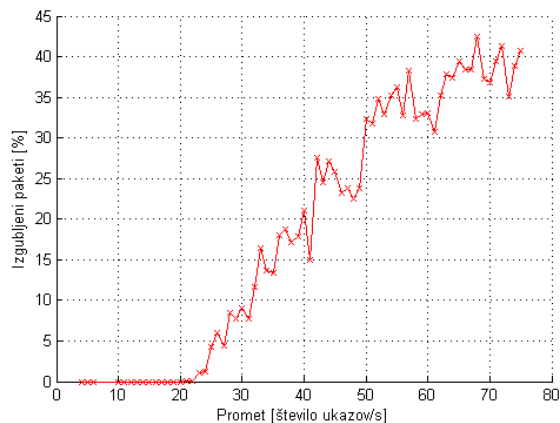
4.2 Zmogljivost prehoda WebService/Konnex

V tem poglavju predstavimo rezultate, ki se nanašajo na merjenje obhodnega časa pri različnih vrednostih prometa na prehodu WebService/Konnex pri izvajanju storitev *write group service* (slika 7) in *read group service* (slika 8). Prav tako v tem poglavju predstavimo rezultate, ki se nanašajo na merjenje števila izgubljenih

paketov pri različnih vrednostih prometa na prehodu WebService/Konnex pri izvajanju storitve *write group service* (slika 9) in *read group service* (slika 10).



Slika 9: Izgubljeni paketi pri izvajanju storitve *write group service* pri različnih stopnjah prometa na prehodu WebService/Konnex



Slika 10: Izgubljeni paketi pri izvajanju storitve *read group service* pri različnih stopnjah prometa na prehodu WebService/Konnex

5 RAZPRAVA

Rezultati eksperimenta, ki jih prikazujejo slike od 3 do 10, se navezujejo na merjenje obhodnega časa in števila izgubljenih paketov pri izvajanju dveh najpogosteje uporabljenih storitev aplikacijskega sloja standarda Konnex (*write group service* in *read group service*) pri različnih vrednostih prometa na prehodu. Z navedenimi storitvami standarda Konnex je mogoče izvajati vse storitve UPZ, navedene v tabeli 11. Podobno kot v študijah [2], [5], [15] smo za zagotovitev podrobne analize obhodni čas in število izgubljenih paketov izmerili posebej za del prehoda Konnex (slike od 3 do 6) in posebej za celoten prehod WebService/Konnex (slike od 7 do 10). V nasprotju s študijami [2], [5], [15] pa smo izvedli meritve na implementiranem prehodu WebService/Konnex in ne na njegovem modelu s pomočjo simulacije kot v študijah [2], [5], [15]. Tovrstni rezultati so bistveno bolj relevantni, saj so odraz realnega delovanja prehoda.

Slika 3 prikazuje odvisnost med obhodnim časom in vrednostjo prometa pri izvajanju storitve *write group service* na prehodu Konnex. Kot je razvidno s slike 3, je obhodni čas $\cong 25$ ms pri vrednosti prometa $n = (0, 42]$ ukazov/s. Obhodni čas izvajanja storitve *write group service* se v osnovi navezuje na čas, ki je potreben za prenos podatkovnega okvirja, ki vsebuje zahtevo za izvajanje storitve *write group service*, in za prenos potrdilnega okvirja, ki vsebuje informacijo, ali je bila zahteva uspešno izvedena ali ne. Obhodni čas je določen s pomočjo enačbe (1).

$$\text{obhodni čas} = \frac{\text{\textit{št. prenesenih bitov}}}{R} \quad (1)$$

Ker je teoretično definirana bitna hitrost v omrežju Konnex $R = 9600$ bit/s in ker je pri izvajanju storitve *write group service* število prenesenih bitov = 193 bit, je v omrežju Konnex obhodni čas = 20.11 ms [19]. Ker je na prehodu Konnex izmerjena vrednost obhodni čas $\cong 25$ ms (slika 3), lahko sklepamo, da je »ozko grlo« pri izvajanju storitve *write group service* omrežje Konnex, saj se skoraj ves čas porabi za prenos podatkov na omrežju Konnex. Prav tako lahko iz rezultatov na sliki 3 sklepamo, da razlika približno 5 ms pri merjenju nastane zaradi pretvarjanja podatkov na prehodu Konnex in zaradi izvajanja programske aplikacije, ki se uporablja za merjenje obhodnega časa v eksperimentalen okolju, zato lahko rezultate eksperimenta prikazane na sliki 3, ocenimo za relevantne. Pri vrednostih prometa $n = (42, \infty)$ ukazov/s pa je s slike 3 razvidno, da obhodni čas začne strmo naraščati. Razlog je v tem, da na prehodu začne nastajati čakalna vrsta, saj se novi ukazi za izvajanje storitve *write group service* na prehodu Konnex pojavijo, še preden se stari končajo. To pa je popolnoma razumljivo, saj se pri izmerjeni vrednosti obhodni čas $\cong 25$ ms lahko izvede brez čakalne vrste maksimalno 40 ukazov/s. Navedeno dejstvo prav tako kaže na to, da so pridobljeni rezultati iz eksperimenta, prikazani na sliki 3, relevantni tudi pri vrednosti prometa $n = (42, \infty)$. S slike 5, ki prikazuje odvisnost med številom izgubljenih paketov in vrednostjo prometa na prehodu Konnex pri izvajanju storitve *write group service*, pa je razvidno, da se paketi pri vrednostih prometa $n = (0, 42]$ ne izgubljajo, pri vrednostih prometa $n = (42, \infty)$ pa začne število izgubljenih paketov naraščati. Iz tega lahko povzamemo, da prehod Konnex oziroma omrežje Konnex zagotavlja zadovoljivo delovanje pri izvajanju storitve *write group service* pri vrednostih prometa $n = (0, 42]$, pri večjih vrednostih prometa pa je delovanje prehoda Konnex oziroma omrežja Konnex nezadovoljivo. Slika 4 pa prikazuje odvisnost med obhodnim časom in vrednostjo prometa pri izvajanju storitve *read group service* na prehodu Konnex. Kot je razvidno s slike 4, je obhodni čas $\cong 25$ ms pri vrednosti prometa $n = (0, 20]$ ukazov/s. Ker je teoretično definirana bitna hitrost v omrežju Konnex $R = 9600$ bit/s in ker je pri izvajanju storitve *read group service* število prenesenih bitov = 412 bit, lahko iz enačbe (a)

izračunamo, da je v omrežju Konnex obhodni čas = 42.92 ms pri izvajanju storitve *read group service* [109]. Ker je na prehodu Konnex izmerjena vrednost obhodni čas $\cong 50$ ms (slika 4), lahko sklepamo, da je »ozko grlo« pri izvajanju storitve *read group service* prav tako omrežje Konnex, saj se skoraj ves čas porabi za prenos podatkov na omrežju Konnex. Prav tako lahko iz rezultatov na sliki 4 sklepamo, da razlika približno 8 ms pri merjenju nastane zaradi pretvarjanja podatkov na prehodu Konnex in zaradi izvajanja programske aplikacije, ki se uporablja za merjenje obhodnega časa v eksperimentalen okolju, zato lahko rezultate eksperimenta prikazane na sliki 4, ocenimo za relevantne. Pri vrednostih prometa $n = (20, \infty)$ ukazov/s pa je iz slike 4 razvidno, da obhodni čas začne strmo naraščati. Razlog je, da na prehodu začne nastajati čakalna vrsta, saj se novi ukazi za izvajanje storitve *read group service* na prehodu Konnex pojavijo, še preden se stari končajo. To pa je popolnoma razumljivo, saj se pri izmerjeni vrednosti obhodni čas $\cong 50$ ms lahko izvede brez čakalne vrste maksimalno 20 ukazov/s. Navedeno dejstvo prav tako kaže na to, da so pridobljeni rezultati iz eksperimenta prikazani na sliki 4, relevantni tudi pri vrednosti prometa $n = (20, \infty)$. S slike 6, ki prikazuje odvisnost med številom izgubljenih paketov in vrednostjo prometa na prehodu Konnex pri izvajanju storitve *read group service*, pa je razvidno, da se paketi pri vrednostih prometa $n = (0, 20]$ ne izgubljajo, pri vrednostih prometa $n = (20, \infty)$ pa začne število izgubljenih paketov strmo naraščati. Iz tega lahko povzamemo, da prehod Konnex oziroma omrežje Konnex zagotavlja zadovoljivo delovanje pri izvajanju storitve *read group service* pri vrednostih prometa $n = (0, 20]$, pri večjih vrednostih prometa pa je delovanje prehoda Konnex oziroma omrežja Konnex nezadovoljivo.

Slika 7 pa prikazuje odvisnost med obhodnim časom in vrednostjo prometa pri izvajanju storitve *write group service* na prehodu WebService/ Konnex. Kot je razvidno s slike 7, je obhodni čas $\cong 85$ ms pri vrednosti prometa $n = (0, 42)$ ukazov/s. V primerjavi s sliko 3, ki prikazuje isto odvisnost le za prehod Konnex, lahko sklepamo, da del prehoda WebService prispeva dodatnih približno 60 ms pri celotnem obhodnem času. Razlog za to je v sami specifikaciji protokola WebService, ki zahteva vzpostavitev dodatnih mehanizmov pri njegovi implementaciji, kar posledično pomeni daljši čas izvajanja storitev, delujočih na podlagi protokola WebService. Obstajajo sicer pristopi, ki omogočajo skrajšanje navedenega časa pri implementaciji protokola WebService, kar pa ni bilo predmet našega dela. S slike 7 je razvidno, da je ozko grlo še vedno omrežje Konnex, saj se v primerjavi s sliko 3 obhodni čas giblje relativno enako glede na vrednost prometa na prehodu WebService/Konnex, le da je velikost obhodnega časa za vrednost približno 60 ms večja. Prav tako kot na sliki 3 začne obhodni čas naraščati pri vrednosti prometa $n = (42, \infty)$, kar je posledica nastajanja čakalnih vrst na prehodu zaradi

fizične omejitve omrežja Konnex. S slike 9, ki prikazuje odvisnost med številom izgubljenih paketov in vrednostjo prometa na prehodu WebService/Konnex pri izvajanju storitve *write group service*, pa je razvidno, da se paketi pri vrednostih prometa $n = (0, 40]$ ne izgubljajo, pri vrednostih prometa $n = (40, \infty)$ pa začne število izgubljenih paketov naraščati. Število izgubljenih paketov začne sicer naraščati nekoliko počasneje kot na sliki 5, kar pa je posledica dodatnih mehanizmov, ki jih vsebuje protokol WebService oziroma protokol, ki ga WebService uporablja na transportnem sloju (TCP). Slika 8 pa prikazuje odvisnost med obhodnim časom in vrednostjo prometa pri izvajanju storitve *read group service* na prehodu WebService/Konnex. Kot je razvidno s slike 8, je obhodni čas $\cong 115$ ms pri vrednosti prometa $n = (0, 20]$ ukazov/s. V primerjavi s sliko 4, ki prikazuje isto odvisnost le za prehod Konnex, lahko sklepamo, da WebService del prehoda prispeva dodatnih približno 65 ms pri celotnem obhodnem času, kar je prav tako posledica mehanizmov, ki jih vsebuje protokol WebService. Prav tako pa je s slike 8 razvidno, da je ozko grlo še vedno omrežje Konnex, saj se v primerjavi s sliko 4 obhodni čas giblje relativno enako glede na vrednost prometa na prehodu WebService/Konnex, le da je velikost obhodnega časa za vrednost približno 65 ms večja. Prav tako kot na sliki 4 začne obhodni čas naraščati pri vrednosti prometa $n = (20, \infty)$, kar je posledica nastajanja čakalnih vrst na prehodu zaradi fizične omejitve omrežja Konnex.

S slike 10, ki pa prikazuje odvisnost med številom izgubljenih paketov in vrednostjo prometa na prehodu WebService/Konnex pri izvajanju storitve *read group service*, pa je prav tako razvidno, da se paketi pri vrednostih prometa $n = (0, 20]$ ne izgubljajo, pri vrednostih prometa $n = (20, \infty)$ pa začne število izgubljenih paketov naraščati. Število izgubljenih paketov začne sicer naraščati nekoliko počasneje kot na sliki 6, kar pa je posledica dodatnih mehanizmov, ki jih vsebuje protokol WebService oziroma protokol, ki ga WebService uporablja na transportnem sloju (TCP).

6 SKLEP

Sistemi in storitve UPZ, ki temeljijo na konceptu računalništva v oblaku, čedalje bolj pridobivajo na pomenu, saj telekomunikacijskim operaterjem omogočajo uvajanje novih storitev M2M s področja inteligentnega doma. Tako se storitve UPZ izvajajo porazdeljeno na strani operaterja, zato se zastavi vprašanje glede vpliva omrežnega prometa na kakovost izvajanja storitev UPZ. Ker omrežja UPZ delujejo na bistveno manjši pasovni širini (9600 bit/s) kot sodobna širokopasovna omrežja (več kot 100 Mbit/s), ugotovimo, da je »ozko grlo« prehod WebService/Konnex, na katerega ima omrežni promet, ki ga ustvarijo uporabniki storitev UPZ, največji vpliv. Zato v članku analiziramo zmogljivosti prehoda WebService/Konnex pri izvajanju storitev UPZ iz

aplikacijskega sloja protokolnega sklada Konnex, in sicer *write group service* in *read group services*. Iz rezultatov poskusa ugotovimo, da omogoča prehod zadovoljivo delovanje pri vrednostih prometa $n = (0, 42]$ ukazov/s za storitev *write group service* ter $n = (0, 20]$ ukazov/s za storitev *read group service*, pri večjih vrednostih prometa pa je delovanje prehoda nezadovoljivo. V nadaljnjem delu se bomo osredinili na optimizacijo delovanja prehoda s pomočjo prioritizirane upravljanja čakalnih vrst na njem.

LITERATURA

- [1] I. Ozcelik in H. Ekiz: Building an Interconnection Between PROFIBUS and ATM Networks. *Journal of Network and Computer Applications*, letn. 30, št. 2, april 2007; str. 800–817.
- [2] I. Ozcelik in H. Ekiz: Design, Implementation and Performance Analysis of the PROFIBUS/ATM Local Bridge. *Computer Standards and Interfaces*, letn. 26, št. 4, avgust 2004; str. 329–342.
- [3] PROFIBUS: International Standard. <http://www.profibus.com>, 2010.
- [4] ABB: PROFIBUS Technical Description (2005).
- [5] H. Ekiz, A. Kutlu, M. D. Baba in E. T. Powner: Performance Analysis of a CAN/CAN Bridge. Referat na konferenci Proceedings of the International Conference on Network Protocols, 1996; str. 181–188.
- [6] M. Farsi, K. Ratcliff in M. Barbosa: An Overview of Controller Area Network. *Computing and Control Engineering Journal*, letn. 10, št. 3, avgust 1999; str. 113–120.
- [7] Controller Area Network (CAN): LAN in Vehicle Communication Protocol. SEA Information Report, 1990; str. 226–248.
- [8] H. Ekiz, A. Kutlu, E. T. Powner in A. Odeh: Performance Analysis of CAN in Bridged Systems. Referat na konferenci Proceedings of 2nd Electrical Engineers Conference, 1996; str. 147–157.
- [9] H. Ekiz, A. Kutlu in E. T. Powner: Design and Implementation of a CAN/CAN Bridge. Referat na konferenci Proceedings of 2nd International Symposium on Parallel Architectures, Algorithms, and Networks, 1. junij 1996, Beijing, China; str. 507–513.
- [10] J. Eltze: Double-CAN Controller as Bridge for Different CAN Networks. Referat na konferenci Proceedings of 4th CAN Conference, 1997.
- [11] M. Tenruh, I. Erturk, E. Stipidis in A. Kutlu: Extending Controller Area Networks by Cut-through Bridging. Referat na konferenci The Proceedings of the 14th International Symposium on Computer and Information Sciences, 18.–20. oktober 1999, Turkey; str. 1053–1056.
- [12] M. Tenruh, I. Erturk, E. Stipidis in J. English: Performance Analysis of Cutthrough Bridged Controller Area Networks. Referat na konferenci International Network Conference, 3.–6. julij 2000, Plymouth, United Kindom; str. 75–82.
- [13] H. Ekiz, A. Kutlu in E. T. Powner: Design and Implementation of a CAN/ Ethernet Bridge. Članek v reviji *Computer Standards and Interfaces*, letn. 30, št. 3, marec 2008; str. 177–186.
- [14] M. Alves in E. Tovar: Engineering PROFIBUS Networks with Heterogeneous Transmission Media. Članek v reviji *Computer Communications*, letn. 30, št. 1, 2006; str. 17–32.
- [15] I. Ozcelik in H. Ekiz: Design, Implementation and Performance Analysis of the CAN/ATM Local Bridge. Članek v reviji *Computer Standards and Interfaces*, letn. 30, št. 3, 2008; str. 177–186.
- [16] M. Umberger, I. Humar, A. Kos, J. Guna, A. Žemva in J. Bešter: The Integration of Home-Automation and IPTV System and Services. Članek v reviji *Computer Standards and Interfaces*, letn. 31, št. 4, 2009; str. 675–684.
- [17] EIBA Handbook Series: System Specifications (2002).
- [18] H. Merz, T. Hansemann in C. Hubner: Building Automation, Communication Systems with Konnex, LON and BACnet. Springer Berlin Heidelberg, Germany, 2009.
- [19] M. Miskowicz: Analysis of LonTalk/EIA-709.1 Channel Performance under Soft Real-time Requirements. Referat na konferenci IEEE International Conference on Industrial Technology, 10.–12. december 2003, letn. 2, str. 705–708.