

# JEDNA METODA OSTVARIVANJA POZIVA TEMELJENA NA ODREĐENOM SUSTAVU OTVORENOG KODA TELEFONSKE CENTRALE

## *A Dial Call Method Based on Specific Open Source Switchboard System*

Tihi Bilas, mag. ing. comp.

Student diplomskog studija  
Odjel za elektrotehniku i računarstvo  
Sveučilište u Dubrovniku, Hrvatska  
E-mail: tihi.bilas@unidu.hr

prof. dr. sc. Vedran Batoš

Odjel za elektrotehniku i računarstvo  
Sveučilište u Dubrovniku, Hrvatska  
E-mail: vedran.batos@unidu.hr

doc. dr. sc. Boris Nemšić

OJC Vimpelcom, Moscow, Russia  
E-mail: bnemsic@beeline.ru

UDK 621.395

### **Sažetak**

*U radu su prikazani trendovi razvoja i implementacije VoIP nasuprot tradicionalnoj tehnologiji. Poseban naglasak dan je na statistike nekih europskih država, povećanje broja korisnika VoIP usluge i predviđanja razvoja spomenutoga u bliskoj budućnosti. Također je prikazana primjena sustava otvorenoga koda pri ostvarivanju telefonskih poziva. Na temelju analize postojećeg sustava, standardnih procedura i raspoložive tehnologije, predložena je jedna metoda ostvarivanja poziva temeljena na određenom sustavu otvorenoga koda telefonske centrale.*

*Ključne riječi: otvoreni sustav, telefonska centrala, biranje poziva, programiranje paketski govor.*

### **Summary**

*In this work the trends of development and implementation of the VoIP system are presented in comparison to the traditional technology. There is an emphasize for the statistics of some European countries, the increased number of VoIP users and a forecast of development in the near future. In addition, this paper has shown the application of open source systems in dial call procedures. Analysing the existing system, comparing the standard procedures as well as, available technologies, one dial call method has been proposed based on a specific open source switchboard system.*

*Key words: Open Source, Switchboard, Dial Plan, Programming, VoIP*

### **UVOD / Introduction**

Suvremena računalna mreža omogućila je razvoj tehnologije prijenosa govora internetskim protokolom VoIP (*Voice Over Internet Protocol*). Implementacija VoIP-a znatno je utjecala na razvoj poslovnih komunikacijskih

sustava. VoIP ne samo da nudi mnoštvo prednosti u tehničkom obliku, koje je teško ili gotovo nemoguće zamijeniti klasičnom telefonijom, već omogućuje integraciju dislociranih lokacija u jedinstvenu privatnu mrežu. Sustavi otvorenoga koda upotrijebljeni u

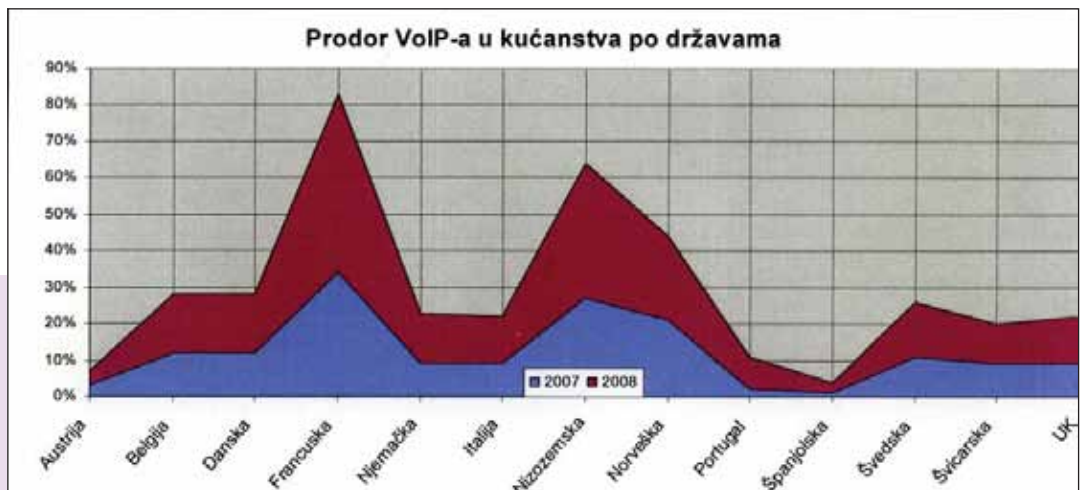
telefonskim centralama omogućuju programabilno ostvarivanje telefonskih poziva. Jedan od takvih sustava je Asterisk-sustav, kojemu je osnovna značajka *Dial plan* sastavljen od popisa uputa što ih sustav slijedi. Na temelju dostupnih sustava i raspoloživih podataka [3,4,9], analizirane su mogućnosti i predložena je jedna metoda ostvarivanja poziva utemeljena na navedenom sustavu otvorenoga koda. Za razliku od tradicionalnih sustava ograničenih svojim osnovnim mogućnostima, *Dial plan* sustavu daje sposobnost prilagodbe bilo kojim uvjetima i korisnikovim zahtjevima.

**TREND U RAZVOJU PRIJENOSA GLASA INTERNETSKIM PROTOKOLOM / Development trend of voice internet protocol**

Telefonska usluga prijenosa glasa internetom preoblikovala je tržište fiksne telefonske linije u Europi. Prema podacima Međunarodne telekomunikacijske zajednice [10], do sredine 2008. 30 milijuna korisnika služilo se VoIP telefonskim linijama u Zapadnoj Europi, a 20 milijuna samo godinu prije. Dok je brzina rasta korištenja VoIP-om u Europi veoma brza, takvi se trendovi dramatično razlikuju od zemlje do zemlje (slika 1.).

- Prodor na tržište sredinom 2008. godine kretalo se od nešto manje 50% kućanstava u Francuskoj do 3% u Španjolskoj.
- Stopa rasta kretala se od slabishnih 13% u Norveškoj pa sve do 544% u Portugalu.
- Trenutno na VoIP tržištu vode Francuska, Italija, Ujedinjeno Kraljevstvo, Nizozemska i Norveška, ali konkurentni operateri drže vodstvo na većini ostalih tržišta.
- Cijena *triple-play* paketa, VoIP i videousluga se kreću od 30 € pa sve do 80 € za mjesec uporabe.

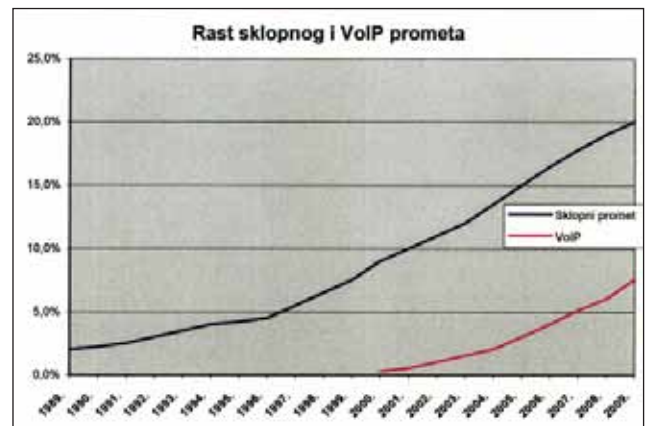
Slika 1. VoIP prodor u kućanstva EU  
Figure 1. VoIP Penetration of Households



Prema podacima godišnjega istraživanja međunarodnog tržišta prijenosa govora [11], objavljenima 2007. godine, prikazan je usporen rast međunarodnog prometa u 2006. godini, do 10%, što je najniža razina u razdoblju od 1986. do 2006. godine (slika 2.).

Dok cijene padaju, natjecanje i složenost nastavlja rasti, pa se za međunarodne prijenosnike očekuju teška vremena u sljedećim godinama.

Međutim, nije sve je tmurno; prekogranični VoIP promet rastao je više od 30% u 2006, i VoIP se pojavio kao ključni pokretač rasta ukupnog tržišta. Zapravo, 2006. je prekretnica na međunarodnom tržištu. VoIP promet povećan je jednako koliko i TDM promet, pa je VoIP prestigao TDM u 2007. godini, s tendencijom rasta do 2009. godine.



Slika 2. Rast sklopnog i VoIP prometa  
Figure 2. Switched and VoIP traffic growth

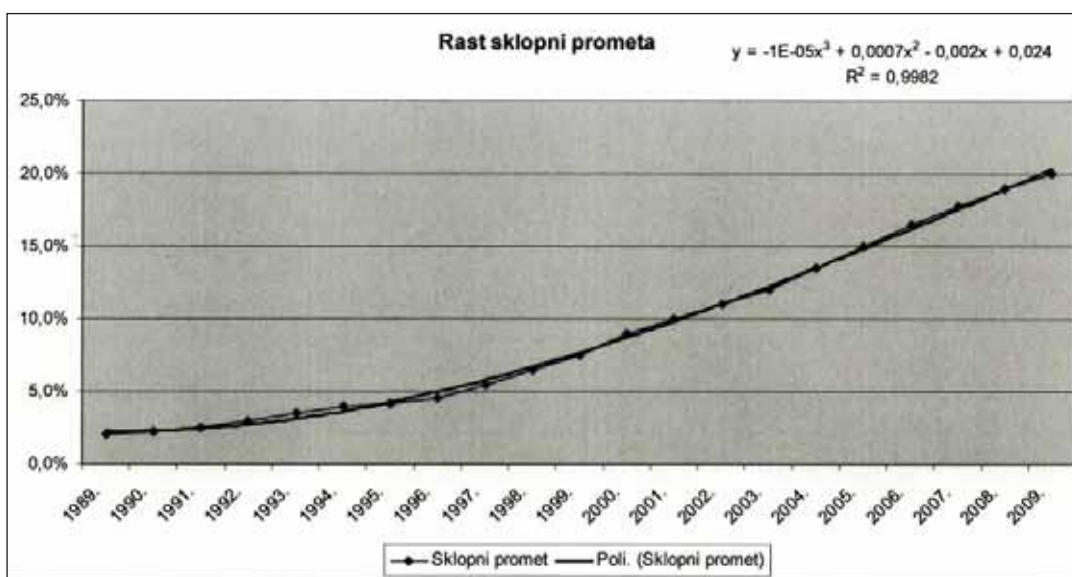
Modelom jednostavne krivolinijske regresije analitički se izražava statistički odnos među dvjema pojavama predočenima vrijednostima numeričkih varijabla - u

našem slučaju tradicionalni (sklopni) i VoIP. Model sadržava jednu zavisnu (ostvareni promet) i nezavisnu varijablu (vrijeme).

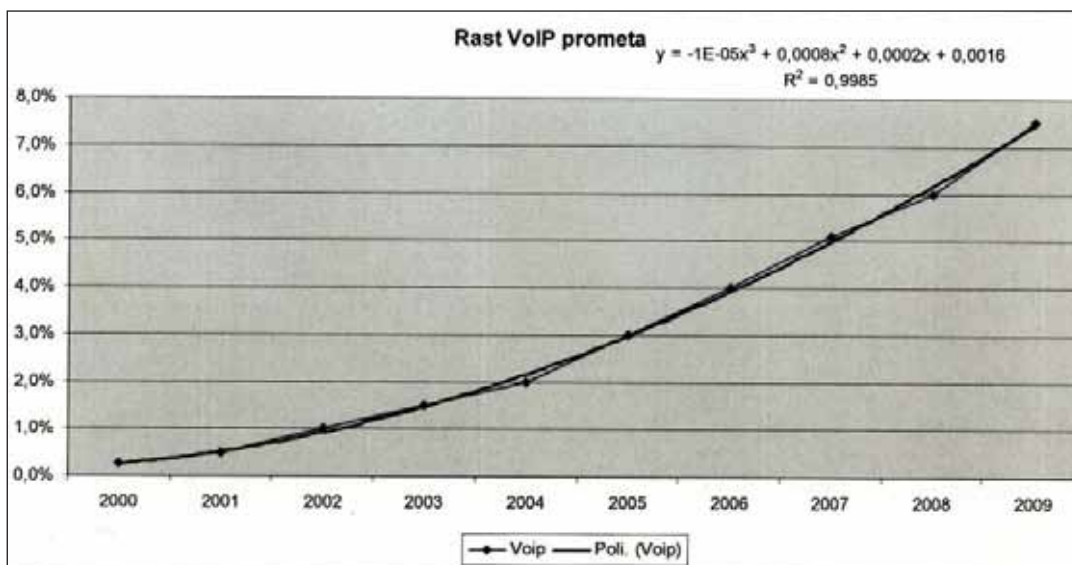
Dijagram rasipanja vrlo je prikladan pri određivanju modela. Za njegovu konstrukciju potreban je koordinatni sustav u koji se unose vrijednosti varijabla X i Y (vremena i ostvarenog prometa), nakon čega se jakost, oblik i smjer veze mogu zaključiti po rasporedu točaka.

Krivolinijski model i u jednom i drugom prikazu (slike 3. i 4.) je polinom trećeg stupnja.

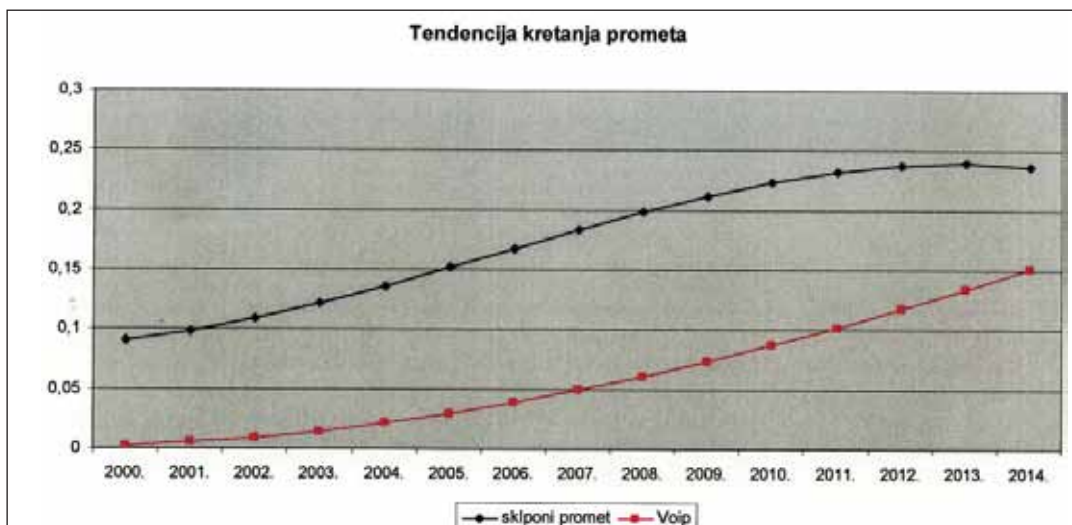
Koeficijent determinacije  $R^2$  pokazuje proporcija protumačenog zbroja kvadrata u ukupnom zbroju kvadrata odstupanja za sklopni promet  $R^2 = 0,9982$  i za VoIP  $R^2 = 0,9985$ . Prema tome je pokazatelj modela reprezentativan.



Slika 3. Rast sklopnog prometa  
 Figure 3. Switched traffic growth

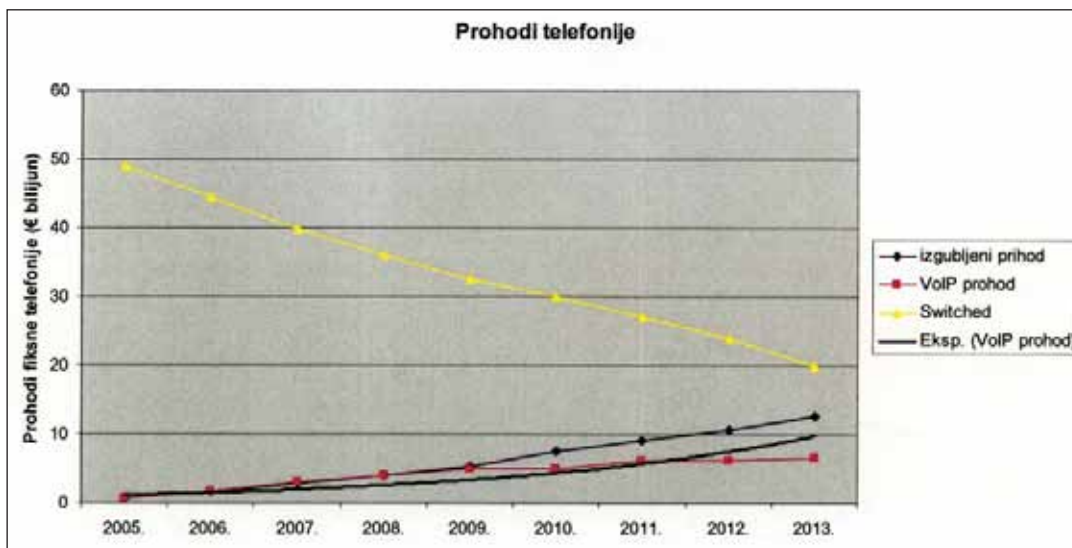


Slika 4. Rast VoIP prometa  
 Figure 4. VoIP traffic growth



Slika 5. Tendencija kretanja prometa  
 Figure 5. Tendency of traffic movement

Na prethodnom grafu prikazana je usporedba i tendencija kretanja sklopnog prometa naprema VoIP-u od 2000. do 2014.



Slika 5.a Tendencija kretanja prometa  
 Figure 5 a. Tendency of traffic movement

Prihodi od VoIP telefonije još su relativno skromni u usporedbu s tradicionalnom telefonijom. Ipak, velik je utjecaj VoIP-a (slika 5.a) pa se očekuju, i već bilježe, smanjeni prihodi tradicionalne telefonije u međunarodnim pozivima, dok u isto vrijeme rastu prihodi VoIP-a.

### SINTAKSA DIALPLANA / Syntax Of Dialplan

Dialplan je definiran u datoteci naziva *extensions.conf*, koja se uobičajeno nalazi u mapi na lokaciji sustava

*/etc/asterisk/*, ali može biti i negdje drugdje ovisno o tome kako instaliramo sustav (Asterisk).

Dialplan se sastoji od četiri glavna dijela [3,4]:

- **konteksta** (*contexts*),
- **ekstenzija** (*extensions*),
- **prioritet** (*priorities*),
- **aplikacija** (*applications*).

U nastavku je opisan svaki od njih i njegova uloga u stvaranju *dialplana*.

### Kontekst / Contexts

*Dialplan* je podijeljen u sekcije imena **kontekst** [3,4] gdje se nalaze ekstenzije, to jest kontekst jednostavno dijeli različite dijelove *dialplana* kako ne bi bilo međusobnog utjecaja. Ekstenzija koja je definirana u jednom kontekstu jednostavno je izolirana od ekstenzije iz bilo kojega drugog konteksta osim ako je međusobna interakcija dopuštena.

Jednostavan primjer može se dobiti ako se zamisle dvije kompanije koje dijele isti Asterisk-server. Ako za svaku kompaniju postavimo audioizbornik (koji se pokreće prilikom poziva), definitivno kompanije možemo odijeliti jednu od druge. To nam dopušta da neovisno definiramo što će se dogoditi kada je, recimo, ekstenzija „0“ birana: ako korisnik pritisne brojku 0, dobit će kompaniju „A“, a ako pritisne brojku 1, dobit će kompaniju „B“. Svaki kontekst može imati svoje ime koje postavljamo na vrhu u zagradama ([ ]). Imena se mogu sastojati od velikih ili malih slova (od A do Z engleske abecede), brojeva (od 0 do 9) i spojnice i podvlaka.

Primjerice, kontekst za dolazni poziv možemo nazvati: [incoming]. Instrukcije koje su upisane nakon definiranja konteksta dio su njega sve dok sljedeći *dialplan* nije definiran. Na početku *dialplana* postoje dva specijalna kontekst-imeni, [general] i [globals], koje ćemo poslije opisati, a zasad je potrebno samo znati da oni postoje.

### Ekstenzije / Extensions

Unutar svakog konteksta možemo definirati jednu ili više ekstenzija [3,4]. Ekstenzija je instrukcija koju će Asterisk slijediti a aktivira se prilikom dolaznog poziva ili broja koji biramo na određenom kanalu. Ekstenzije specificiraju što se događa s pozivom prilikom njegova puta kroz *dialplan*.

Sintaksa ekstenzije je riječ „*exten*“, nakon čeka slijedi znak „=“ i znak „>“.

**exten =>**

Nakon ovoga slijedi ime ekstenzije. Kada se bavimo

telefonskim sustavima, skloni smo razmišljati o ekstenzijama kao o brojevima koje ćemo birati kako bi željeni telefon zvonio. U Asterisku, ime ekstenzije može se sastojati od kombinacije brojeva i slova.

Potpuna se ekstenzija sastoji od tri komponente:

- **ime** (*name*) ili broj ekstenzije,
- **prioriteti** (svaka ekstenzija može uključivati višestruke korake a njihov broj zovemo prioritetom (*priority*),
- **aplikacije** (ili naredbe) s kojom se obavlja neka akcija na poziv (*application*).

Te su tri komponente odijeljene zarezom kao u primjeru:

**exten => name,priority,application( )**

U nastavku ću prikazati jednostavan primjer kako bi prava ekstenzija mogla izgledati:

**exten => 123,1,Answer( )**

U ovom primjeru je ime ekstenzije **123**, prioritet je **1**, a aplikacija je **Answer( )**.

### Prioriteti / Priorities

Svaka ekstenzija može imati nekoliko koraka, koje zovemo prioritetom [3]. Svaki prioritet je numeriran (u nizu počevši od broja 1). Svaki prioritet izvodi točno jednu aplikaciju. Kao u primjeru (u nastavku), ekstenzija će odgovoriti na poziv (u prioritetu broja 1), a potom ga prekinuti (u prioritetu broja 2).

**exten => 123,1,Answer( )**

**exten => 123,2,Hangup( )**

### Aplikacije / Applications

Aplikacije su skladišta za *dialplanove*. Svaka aplikacija obavlja određenu akciju na određenom kanalu, kao što je puštanje zvuka, prihvatanje tonske signalizacije ili možda prekidanje poziva. U prethodnom smo primjeru predstavili dvije jednostavne aplikacije: **Answer( )** i **Hangup( )**. U nastavku bit će prikazano kako se to radi u realnom vremenu (Primjer 1.).

U novijim verzijama Asteriska, onim većima od verzije 1.2., uvodi se mogućnost postavljanja slova „n“ za ime prioriteta. To nam omogućava da se preuzme prethodni broj prioriteta i da se uveća za jedan. Na taj način se olakšava mijenjanje *dialplan*-a jer u tom slučaju ne moramo zadržavati numeraciju svakog koraka. U nastavku će se predstaviti primjer koji ima spomenute karakteristike:

**exten => 123,1,Answer( )**

**exten => 123,n, radi nešto**

**exten => 123,n, radi nešto drugo**

**exten => 123,n, radi zadnju naredbu**

**exten => 123,n, Hangup( )**

Verzija 1.2 također dopušta dopisivanje tekstualnih oznaka prioritetima. Za dodjelu teksta prioritetu, jednostavno dopišemo željeno unutar zagrada nakon oznake prioriteta, kao što je u nastavku:

**exten => 123,n(tekstualna oznaka), radi nešto**

Primjer 1. Rad aplikacije u realnom vremenu  
Example 1. Example of application in real time

Neke aplikacije, poput **Answer ()** i **Hangup ()**, ne trebaju neke druge instrukcije, dok ostale zahtijevaju dodatne instrukcije. Taj dio informacije, zvan argumenti, može biti dodan aplikaciji kako bi se dobile informacije o aktivnostima koje obavlja aplikacija.

### JEDNOSTAVNI DIALPLAN / *Simple dialplan*

Sada smo spremni kreirati naš prvi *dialplan* a početak ćemo s veoma jednostavnim primjerom. Dizajnirat ćemo *dialplan* koji će, nakon što se ostvari dolazni poziv, Asterisku omogućiti javljanje na poziv, puštanje neke glazbe te prekid poziva. Da bi primjeri u nastavku mogli raditi uredno, pretpostavit ćemo da je barem jedan **Zap kanal** kreiran i iskonfiguriran i da se svi dolazni pozivi šalju na *[incoming]* kontekst.

### “S” ekstenzija / “S” extension

Prije negoli krenemo s našim *dialplanom*, trebali bi kazati nešto više o specijalnoj ekstenziji naziva „s“. Kada poziv uđe u kontekst bez posebne odredišne ekstenzije, automatski će ga preuzeti „s“ ekstenzija. Prezentirat ću tri akcije koje se odvijaju pri dolasku poziva (javljanje na poziv *[answer it]*, puštanje zvuka *[play a sound file]*, prekid poziva *[hang it up]*) pa zato moram kreirati ekstenziju zvanu „s“ s tri prioriteta koje ću smjestiti unutar *[incoming]* konteksta (jer bi svi dolazni pozivi trebali započeti unutar tog konteksta).

**[incoming]**

**exten => s,1,application()**

**exten => s,2,application()**

**exten => s,3,application()**

Sada nam je potrebno napuniti aplikaciju (application), nakon čega ćemo imati kreiran ovaj prvi *dialplan*.

### Aplikacije Answer(), Playback() i Hangup() / *Answer(), Playback() and Hangup() Applications*

Da bi se omogućilo javljanje na poziv, puštanje glazbe i potom prekinulo taj isti poziv, potrebno je naučiti kako to i napraviti.

Aplikacija **Answer()** rabi se da bi se odgovorilo na kanal preko kojega se želi ostvariti poziv. Pri tom je potrebno namještanje na kanalu preko kojega imamo dolazni poziv. Kao što sam prije spomenuo, za **Answer()** nemamo argument.

Aplikacija **Playback()** služi za puštanje prethodno snimljenog zvuka preko kanala. Kada se koristimo spomenutom aplikacijom, bilo koja ulazna informacija od korisnika se ignorira. Da bi se rabio **Playback()**,

potrebno je specificirati ime snimljenog zvuka kao argument. Za primjer: **Playback** (ime\_snimljenog\_zvuka) će odsvirati zvuk imena *ime\_snimljenog\_zvuka.gsm* (uz pretpostavku da je spomenuta datoteka spremljena u predefiniranoj mapi za zvučne datoteke). Također možemo uključiti potpunu putanju do datoteke kao što je u nastavku:

**Playback(/home/tihi/sounds/ime\_snimljenog\_zvuka)**

Ovaj primjer će svirati **ime\_snimljenog\_zvuka.gsm** iz mape **/home/tihi/sounds/**. Za razliku od apsolutne putanje, također se može upotrebljavati relativna od Asterisk glazbene mape:

**Playback(muzika/ ime\_snimljenog\_zvuka)**

Ovaj primjer će svirati **ime\_snimljenog\_zvuka.gsm** iz mape **muzika/** koja se nalazi unutar unaprijed definirane mape za zvuk.

Aplikacija **Hangup()** služi za akciju koju opisuje samo ime aplikacije: prekida aktivni poziv na kanalu. Pozivatelj će primiti indikaciju da je poziv prekinut. Ovu aplikaciju rabimo na kraju konteksta kada želimo završiti trenutni poziv; to je potrebno kako bismo bili sigurni da se pozivatelj poziv nije nastavio u *dialplanu*. Ova aplikacija ne zahtijeva argumente.

### Izradba osnovnoga dialplana / *Creation of basic dialplan*

Nakon što smo kreirali ekstenziju i dodali tri različita prioriteta, te naučili o aplikacijama kojima ćemo se koristiti, možemo sve te dijelove sastaviti zajedno kako bismo kreirali naš prvi *dialplan*. Kao što je uobičajeno kod svih primjera u programiranju, naš prvi primjer će biti „Hello World!“.

U prvom prioritetu naše ekstenzije, javit ćemo se na poziv, potom u drugom prioritetu pustiti glazbu imena *hello-world.gsm* i u trećem bismo prekinuli poziv. Primjer toga bi izgledao kao u nastavku:

**[incoming]**

**exten => s,1,Answer()**

**exten => s,2,Playback(hello-world)**

**exten => s,3,Hangup()**

Iako je ovaj primjer kratak i jednostavan, on naglašava osnovne koncepte konteksta, ekstenzije, prioritete i aplikacije.

### Aplikacije Background() i Goto() / *Background() and Goto() Applications*

Jedna od bitnih činjenica za izgradnju interaktivnog **Asterisk**-sustava je **Background()** aplikacija, koja radi nešto slično poput aplikacije **Playback()**.

**Background()** aplikacija najčešće služi za kreiranje govornog izbornika koji upotrebljavaju mnoge tvrtke kako bi se velik broj poziva preusmjerio na govorne automate za osnovne informacije, te se na taj način olakšalo zaposlenicima na informacijskom pultu.

**Background()** ima istu sintaksu kao **Playback()** aplikacija:

```
exten => 123,1,Background(hello-world)
```

Druga zanimljiva aplikacija je **Goto()**. Kao što samo ime kaže, spomenuta aplikacija preusmjerava poziv u drugi kontekst, ekstenziju i prioritet. **Goto()** nam olakšava programsko prebacivanje poziva između različitih dijelova *dialplana*. Sintaksa **Goto()** aplikacije nas navodi kako proći kroz određeni kontekst, ekstenziju i prioritet kao argumente aplikacije, primjerice:

```
exten => 123,1,Goto(context,extension,priority)
```

U sljedećem primjeru odabrat ćemo **Background()** i **Goto()** aplikacije za kreiranje malo složenijeg *dialplana*. Omogućit ćemo pozivatelju komuniciranje sa sustavom koristeći se brojevima na telefonskoj tipkovnici.

```
[incoming]
exten => s,1,Answer( )
exten => s,2,Background(enter-ext-of-person)
exten => 1,1,Playback(digits/1)
exten => 2,1,Playback(digits/2)
```

U prethodnom primjeru sam pokazao da, kad korisnik pozivom pristupi *dialplanu*, čut će se pozdrav koji može biti oblika: „Molimo Vas, unesite broj koji želite pozvati.“ Ako pritisne broj 1, korisnik će čuti ton brojke 1, a pritisne li brojku 2, korisnik će čuti ton brojke 2.

U nastavku možemo uzeti aplikaciju **Goto()** kako bi se omogućilo ponavljanje, to jest preusmjerit ćemo poziv korisnika na početak *dialplana*.

```
[incoming]
exten => s,1,Answer( )
exten => s,2,Background(enter-ext-of-person)
exten => 1,1,Playback(digits/1)
exten => 1,2,Goto(incoming,s,1)
exten => 2,1,Playback(digits/2)
exten => 2,2,Goto(incoming,s,1)
```

Nakon što sam predstavio prvi kompletan govorni izbornik, mogu u njega dodati neke dodatne specijalne ekstenzije.

Prvo, potrebna nam je neka ekstenzija za pogrešan unos, to jest kad korisnik, pozivatelj pritisne pogrešnu tipku na tipkovnici telefona.

Drugo, potrebna nam je ekstenzija koja će se znati nositi sa situacijom kada korisnik ne pritisne neku od tipki, to jest ne ponudi željeni ulazni odziv s obzirom

na ponuđeno u izborniku. Poziv će se preusmjeriti na „t“ ekstenziju ako pozivatelj predugo odugovlači s pritiskom tipke na telefonu nakon završetka sviranja glazbe iz **Background()** aplikacije.

```
[incoming]
exten => s,1,Answer( )
exten => s,2,Background(enter-ext-of-person)
exten => 1,1,Playback(digits/1)
exten => 1,2,Goto(incoming,s,1)
exten => 2,1,Playback(digits/2)
exten => 2,2,Goto(incoming,s,1)
exten => i,1,Playback(pbx-invalid)
exten => i,2,Goto(incoming,s,1)
exten => t,1,Playback(vm-goodbye)
exten => t,2,Hangup( )
```

Koristeći se „t“ ekstenzijom naš *dialplan* je složeniji pa zato i pristupačniji za korisnika.

### Aplikacija **Dial()** / **Dial()** Application

Jedna od najvrjednijih značajka Asteriska je njegova sposobnost u povezivanju različitih pozivatelja međusobno. To je posebno bitno kada se pozivatelji koriste različitim metodama komunikacije. Primjerice, jedan korisnik može komunicirati s pomoću standardne analogne telefonije, dok drugi može sjediti u kafiću miljama daleko koristeći se IP telefonijom. Srećom, Asterisk nam to može omogućiti, to jest preuzima težak posao oko komunikacije i prijevoda između različitih standarda i mreža. Baš zbog toga je bitno nešto više kazati o **Dial()** aplikaciji.

Sintaksa **Dial()** aplikacije nešto je složenija nego što je to u drugim aplikacijama koje su prethodno predstavljene. U **Dial()** su do četiri argumenta.

**Prvi argument** je određite koje želimo pozvati. Sastoji se od tehnologije, znaka „/“ i imena ili broja kanala preko kojega se ostvaruje poziv.

Na primjer, pretpostavimo da želimo pozvati Zap kanal naziva Zap/1, koji je FXS kanal s priključenim analognim telefonom. Tehnologija je „Zap“ a izvor 1. Slično tome je u drugim tehnologijama. Poziv prema SIP uređaju može imati određite SIP/1234, a poziv prema IAX uređaju može imati IAX/prijatelj određite. Ako želimo da Asterisk zvuči na Zap/1 kanalu kada je ekstenzija 123 postignuta u *dialplanu*, treba dodati sljedeću ekstenziju:

```
exten => 123,1,Dial(Zap/1)
```

Kada je ekstenzija izvedena, Asterisk će omogućiti poziv na telefon spojen na kanalu Zap/1. Ako se na telefon javimo, Asterisk će premostiti unutrašnji poziv s Zap/1 kanalom. Također možemo birati višestruke

kanale u isto vrijeme koristeći se znakom „&“, kao u primjeru:

```
exten => 123,1,Dial(Zap/1&Zap/2&Zap/3)
```

Dial() aplikacija će preusmjeriti unutrašnji poziv s bilo kojim kanalom na kojemu se prvo odgovori.

**Drugi argument** Dial() aplikacije je vrijeme čekanja (*timeout*), koji je određen vremenskom jedinicom u sekundama. Ako je dodano vrijeme čekanja, Dial() će pokušavati pozvati odredišta u vremenu koje smo odredili, prije nego što odustane od poziva. Ako vrijeme čekanja nije naznačeno, Dial() će birati sve dok ne dođe odgovor ili pozivatelj prekine. U sljedećem primjeru definirat će se vrijeme čekanja od 10 sekunda:

```
exten => 123,1,Dial(Zap/1,10)
```

Ako je na poziv odgovoreno prije isteka vremena čekanja, kanali će biti premošteni pa je *dialplan* završen. Ako odredište ne odgovara na poziv, Dial() ide na sljedeći prioritet u ekstenziji. Ako je odredišni kanal zauzet, Dial() će ići na sljedeći prioritet n+101 ako on postoji („n“ je prioritet gdje je Dial() aplikacija pozivana). To omogućuje različito rukovanje neostvarenim pozivima i onima gdje je odredišni kanal zauzet.

Primjer prethodnoga je u nastavku:

```
exten => 123,1,Dial(Zap/1,10)
exten => 123,2,Playback(vm-nobodyavail)
exten => 123,3,Hangup()
exten => 123,102,Playback(tt-allbusy)
exten => 123,103,Hangup()
```

Kao što je razvidno u primjeru, svira se *vm-nobodyavail.gsm* datoteka ako na poziv nije odgovoreno, ili *tt-allbusy.gsm* datoteka ako je Zap/1 kanal trenutno zauzet.

**Treći argument** Dial() aplikacije je niz opcija. On može sadržavati jedan ili više znakova kako bi se promijenilo ponašanje Dial() aplikacije. Iako je popis mogućih opcija predug da bi sve prezentiralo, najpopularnija opcija je ona sa slovom „r“.

Ako postavimo slovo „r“ kao treći argument, pozivatelj će čuti ton zvonjenja u svojoj slušalici za vrijeme dok na strani pozivatelja zvonji telefon.

Da bi se dodala „r“ opcija našem zadnjem primjeru, jednostavno se promijeni prva linija koda:

```
exten => 123,1,Dial(Zap/1,10,r)
exten => 123,2,Playback(vm-nobodyavail)
exten => 123,3,Hangup()
exten => 123,102,Playback(tt-allbusy)
exten => 123,103,Hangup()
```

Ekstenzije koje imaju naziv 1 i 2 u našem *dialplanu* su beskorisne pa su služile samo da se prezentira

kako radi Dial() aplikacija. Stoga ih možemo zamijeniti ekstenzijama 101 i 102 koje će omogućiti uspostavu vanjskih poziva s lokalnim korisnicima (Mate i Ivan).

```
[incoming]
```

```
exten => s,1,Answer()
```

```
exten => s,2,Background(enter-ext-of-person)
```

```
exten => 101,1,Dial(Zap/1,10)
```

```
exten => 101,2,Playback(vm-nobodyavail)
```

```
exten => 101,3,Hangup()
```

```
exten => 101,102,Playback(tt-allbusy)
```

```
exten => 101,103,Hangup()
```

```
exten => 102,1,Dial(SIP/Ivan,10)
```

```
exten => 102,2,Playback(vm-nobodyavail)
```

```
exten => 102,3,Hangup()
```

```
exten => 102,102,Playback(tt-allbusy)
```

```
exten => 102,103,Hangup()
```

```
exten => i,1,Playback(pbx-invalid)
```

```
exten => i,2,Goto(incoming,s,1)
```

```
exten => t,1,Playback(vm-goodbye)
```

```
exten => t,2,Hangup()
```

**Četvrti, zadnji, argument** Dial() aplikacije je URL. Ako odredišni kanal omogućuje primanje URL-a u vrijeme poziva, specificirani URL će biti poslan.

Na primjer; ako imamo IP telefon koji omogućuje primanje URL-a, URL će se prikazati na zaslonu telefona, dok pri uporabi programskog telefona, URL bi nam mogao iskočiti u posebnom prozorčiću.



Slika 6. Primjer programskog telefona [8]  
Figure 6. A sample of phone program

Ako ostvarujemo poziv prema van preko FXO Zap kanala, može poslužiti sljedeća sintaksa za biranje broja na tom kanalu:

```
exten => 123,1,Dial(Zap/4/5551212)
```

Ovaj primjer će birati broj 555-1212 na Zap/4 kanalu. Za



druge vrste kanala, kao što su SIP i IAX, jednostavno zamijenimo kao u primjeru:

exten => 123,1,Dial(SIP/1234)

exten => 124,1,Dial(IAX2/Mate@unidu.hr)

Istaknuto je da argument može ostati prazan. Primjerice, želimo li specificirati opciju, ali ne i vrijeme čekanja, argument se vremena čekanja ostavi praznim kao u primjeru:

exten => 123,1,Dial(Zap/1,,r)

### ZAKLJUČAK / Conclusion

Paketski prijenos govora (VoIP) tehnologija je prijenosa s pomoću interneta. IP (*Internet Protocol*) telefonija otklanja većinu tehničkih ograničenja vezanih uz broj lokala ili drugih krajnjih točaka što se služe resursima VoIP-a. Time se omogućuje uporaba različitih naprednih funkcija koje je teško, ili gotovo nemoguće, primijeniti u klasičnoj telefoniji. Neke od takvih su govorna pošta, poziv na čekanju, a posebno, što je nemoguće implementirati u klasičnoj telefoniju, jest zaprimanje faksa na klasični uređaj pri čemu nam zaprimljeni sadržaj može biti usmjeren i na *e-mail* adresu. Uporaba ove tehnologije u sustavima otvorenoga koda omogućava ugradnju određenih metoda pri primanju telefonskih poziva, uz znatnu uštedu financijskih sredstava.

### LITERATURA / References

- [1] Hallock, J., *A Brief History of VoIP*, University of Washington, 2004
- [2] Kalaman, S., IP telefonija u poslovnom komunikacijskom sustavu MD110, Ericsson Nikola Tesla Revija 17(2004), 2004
- [3] Van Meggelen, J., Smith J., Madsen, L., Asterisk™; *The Future of Telephony*, O Reilly Media, 2005
- [4] Wallingford, T., *Switching to VoIP*, O Reilly Media, 2005
- [5] [www.carnet.hr](http://www.carnet.hr)
- [6] [www.digiumcards.com](http://www.digiumcards.com)
- [7] [www.liderpress.hr](http://www.liderpress.hr), članak: IP telefonija prepolovila telefonske račune
- [8] <http://www.zoiper.com>
- [9] <http://www.telegeography.com>
- [10] ITU - International Telecommunication Union, ICT Statistics, VoIP in Europe, ITU, 2009
- [11] TeleGeography's European VoIP & Triple Play Research Service, TeleGeography, 2008

Rukopis primljen: 30. 12. 2009.

