



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ  
ΚΑΤΕΥΘΥΝΣΗ ΨΗΦΙΑΚΩΝ ΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ  
ΔΙΚΤΥΩΝ**

## **Εφαρμογές VoIP στην πλατφόρμα Asterisk**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

του  
**ΙΩΑΝΝΗ Γ. ΚΡΑΜΠΗ**

Πειραιάς  
Μάρτιος 2010





**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ  
ΚΑΤΕΥΘΥΝΣΗ ΨΗΦΙΑΚΩΝ ΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ  
ΔΙΚΤΥΩΝ**

## **Εφαρμογές VoIP στην πλατφόρμα Asterisk**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

του  
**ΙΩΑΝΝΗ Γ. ΚΡΑΜΠΗ**

Πειραιάς  
Μάρτιος 2010



## ΠΕΡΙΛΗΨΗ

Η παρούσα μεταπτυχιακή εργασία ασχολείται με την τεχνολογία Voice over IP και πιο συγκεκριμένα με την ανάπτυξη εφαρμογών για το τηλεφωνικό κέντρο ανοιχτού κώδικα Asterisk. Για την υλοποίηση της εργασίας αυτής εγκαταστήσαμε δυο κέντρα Asterisk με την χρήση του image Asterisknow. Η εργασία αποτελείται από πέντε κεφάλαια στα οποία αναλύεται η τεχνολογία VoIP, τα πρωτόκολλα που χρησιμοποιεί καθώς επίσης η δομή του Asterisk αλλά και οι εφαρμογές που αναπτύξαμε πάνω σε αυτό. Το πρώτο κεφάλαιο περιγράφει την τεχνολογία Voice over IP ξεκινώντας με μια Ιστορική αναδρομή αυτής της τεχνολογίας και συνεχίζοντας με τα βασικά χαρακτηριστικά της αλλά και τα μεγέθη που χαρακτηρίζουν την ποιότητα των παρεχόμενων υπηρεσιών. Στο δεύτερο κεφάλαιο αναλύονται τα πρωτόκολλα και οι αλγόριθμοι κωδικοποίησης που χρησιμοποιεί αυτή η τεχνολογία. Το τρίτο κεφάλαιο αναφέρεται στο softswitch Asterisk περιγράφοντας τι είναι το Asterisk, πως ξεκίνησε, τι απαιτήσεις έχει σε hardware αλλά και τι είναι συμβατό με αυτό, τα πλεονεκτήματα και τα μειονεκτήματα του, την αρχιτεκτονική του, ορισμένα σενάρια χρήσης του Asterisk καθώς και τα βασικά χαρακτηριστικά του ως τηλεφωνικό κέντρο VoIP. Στο Τέταρτο Κεφάλαιο περιγράφονται όλες οι απαραίτητες διαδικασίες για την εγκατάσταση και διαμόρφωση του Asterisk ώστε αυτό να καταστεί ένα πλήρως λειτουργικό τηλεφωνικό κέντρο. Τέλος στο πέμπτο κεφάλαιο περιγράφουμε και δοκιμάζουμε όλες τις εφαρμογές που αναπτύχθηκαν πάνω στο Asterisk στα πλαίσια αυτής της εργασίας. Η πρώτη εφαρμογή που αναπτύξαμε υλοποιεί την καταγραφή των κλήσεων που γίνονται μέσω του Asterisk. Η δεύτερη εφαρμογή υλοποιεί την σύνδεση του Asterisk με έναν από τους παρόχους υπηρεσιών VoIP, ώστε να μπορούμε να βγάζουμε κλήσεις προς το pstn δίκτυο χωρίς χρήση επιπλέον hardware αλλά και με πολύ χαμηλές χρεώσεις. Η τρίτη εφαρμογή υλοποιεί την διασύνδεση δύο κέντρων Asterisk μεταξύ τους έτσι ώστε να μπορούν οι χρήστες του ενός να επικοινωνούν με αυτούς του άλλου. Η τέταρτη εφαρμογή αναφέρεται στην δημιουργία δωματίων τηλεδιάσκεψης ώστε να μπορούν πολλοί χρήστες να επικοινωνούν ταυτόχρονα χωρίς να είναι απαραίτητο να βρίσκονται στον ίδιο χώρο. Η τελευταία εφαρμογή υλοποιεί την υπηρεσία call park, όπου οι εισερχόμενες κλήσεις αφού απαντηθούν μπορούν να τοποθετηθούν σε κάποια θέση με κάποιο χαρακτηριστικό νούμερο, ώστε να μπορούν στην συνέχεια να εξυπηρετηθούν από κάποιον άλλο χρήστη του Asterisk.



## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ .....	9
ΚΕΦΑΛΑΙΟ 1 .....	13
Τεχνολογία Voice Over IP (VoIP).....	13
1.1 Σύντομη περιγραφή της τεχνολογίας VoIP .....	13
1.2 Ιστορική αναδρομή .....	13
1.3 Βασικές αρχές λειτουργίας.....	18
1.4 Μεγέθη χαρακτηρισμού της ποιότητας .....	19
1.4.1 Καθυστέρηση.....	19
1.4.2 Jitter.....	19
1.4.3 Ηχώ (echo).....	20
1.4.4 Λειτουργία αναγνώρισης φωνής (Voice Activity Detection -VAD).....	21
ΚΕΦΑΛΑΙΟ 2 .....	23
Βασικά Πρωτόκολλα και αλγόριθμοι κωδικοποίησης (codec) του VoIP .....	23
2.1 Πρωτόκολλα σηματοδοσίας.....	23
2.1.1 SIP.....	24
2.1.2 H.323.....	29
2.1.3 IAX .....	32
2.2 Πρωτόκολλα μεταφοράς δεδομένων .....	33
2.2.1 Real-time Transport Protocol (RTP).....	34
2.2.2 Secure Real-time Transport Protocol (SRTP).....	35
2.3 Αλγόριθμοι κωδικοποίησης δεδομένων (codecs).....	35
ΚΕΦΑΛΑΙΟ 3 .....	39
Εισαγωγή στο Asterisk .....	39
3.1 Περιγραφή του Asterisk.....	40
3.2 Η Digium και το Asterisk .....	41
3.3 Hardware που χρησιμοποιείται στο Asterisk.....	42
3.4 Πλεονεκτήματα και μειονεκτήματα του Asterisk.....	43

3.4.1 Πλεονεκτήματα .....	44
3.4.2 Μειονεκτήματα .....	46
3.5 Η Αρχιτεκτονική του Asterisk .....	47
3.5.1 Ο πυρήνας του Asterisk .....	48
3.5.2 APIs για την φόρτωση των Modules .....	48
3.6 Περιπτώσεις χρήσης του Asterisk .....	53
3.6.1 Εταιρία με γραφεία σε διαφορετικές πόλεις και χώρες .....	54
3.6.2 Χρήση ως IP PBX (softswitch).....	55
3.6.3 Χρήση ως Softswitch παράλληλα με ένα συμβατικό PBX .....	55
3.6.4 Χρήση ως τηλεφωνικό κέντρο εξυπηρέτησης πελατών .....	56
3.6.5 Χρήση ως εξωτερικός Server εφαρμογών .....	57
3.6.6 Χρήση του Asterisk ως Media Gateway.....	58
3.7 Χαρακτηριστικά, δυνατότητες και λειτουργίες του Asterisk .....	59
ΚΕΦΑΛΑΙΟ 4 .....	63
Εγκατάσταση και διαμόρφωση του Asterisk.....	63
4.1 Εγκατάσταση του Asterisk .....	63
4.2 Διαμόρφωση του Asterisk.....	65
4.2.1 Δημιουργία χρηστών.....	66
4.2.2 Το Dialplan του Asterisk .....	69
4.2.3 Τα αρχεία sip.conf και iax.conf .....	79
4.2.4 Διαμόρφωση των softphone.....	80
4.3 Δοκιμή του Asterisk.....	82
ΚΕΦΑΛΑΙΟ 5 .....	87
Εφαρμογές πάνω στην πλατφόρμα του Asterisk .....	87
5.1 Εφαρμογή για την καταγραφή των συνομιλιών .....	87
5.1.1 Καταγραφή κλήσεων με ειδοποίηση του καλούντος.....	88
5.1.2 Καταγραφή κλήσεων χωρίς την ειδοποίηση του καλούντος.....	91
5.2 Σύνδεση του Asterisk με πάροχο Υπηρεσιών VoIP .....	93
5.2.1 Διαμόρφωση της εφαρμογής μέσω του γραφικού περιβάλλοντος Asterisk GUI .....	95



5.2.2 Διαμόρφωση της εφαρμογής μέσω της απευθείας διαμόρφωσης των αρχείων του Asterisk.....	97
5.2.3 Δοκιμή της εφαρμογής.....	99
5.3 Διασύνδεση δυο Asterisk Server με χρήση του πρωτοκόλλου IAX (Inter-Asterisk eXchange) .....	104
5.4 Εφαρμογή για την δημιουργία «δωματίων» τηλεδιάσκεψεων .....	112
5.4.1 Διαμόρφωση της εφαρμογής τηλεδιάσκεψης με απευθείας διαμόρφωση των αρχείων .....	113
5.4.2 Διαμόρφωση της εφαρμογής τηλεδιάσκεψης με χρήση του γραφικού περιβάλλοντος.....	117
5.5 Call Park.....	119
5.5.1 Διαμόρφωση της εφαρμογής call park με απευθείας διαμόρφωση των αρχείων .....	119
5.5.2 Διαμόρφωση της εφαρμογής call park με χρήση του AsteriskGui.....	120
5.5.3 Δοκιμή της εφαρμογής Call Park.....	120
Βιβλιογραφικές Αναφορές.....	123



## ΠΡΟΛΟΓΟΣ

Στην εποχή την οποία ζούμε η επικοινωνία μεταξύ ανθρώπων που βρίσκονται σε διαφορετικές γεωγραφικές θέσεις είναι πολύ σημαντική τόσο σε προσωπικό όσο και σε επαγγελματικό επίπεδο. Εταιρίες με γραφεία διασκορπισμένα σε όλον τον πλανήτη επιθυμούν την δυνατότητα επικοινωνίας όλων των εργαζομένων τους με το χαμηλότερο δυνατό κόστος. Ο όρος επικοινωνία στην εποχή μας σημαίνει κάτι παραπάνω από απλή συνομιλία 2 ανθρώπων και επεκτείνεται στην διάφορες εφαρμογές όπως τηλεδιασκέψεις, κλήσεις video, συνδυασμός εφαρμογών άμεσων μηνυμάτων και ομιλίας κτλ.

Έτσι αυτή η διαρκής ανάγκη για επικοινωνία τις τελευταίες δεκαετίες έχει οδηγήσει στην δημιουργία διαφόρων τεχνολογιών και πρωτοκόλλων. Έχουν αναπτυχθεί πολλές τεχνολογίες οι οποίες υποστηρίζουν τις διάφορες εφαρμογές με νεότερη και αναπτυσσόμενη στις μέρες μας την τεχνολογία voice over IP (VoIP). Η τεχνολογία αυτή υποβοηθούμενη από την ωρίμανση του διαδικτύου και γενικότερα των IP δικτύων έχει γνωρίσει μεγάλη ανάπτυξη στις μέρες μας όπου πάρα πολλές εφαρμογές στηρίζονται σε αυτή. Η τεχνολογία αυτή απαρτίζεται από μια σειρά πρωτοκόλλων της Internet Engineering Task Force (IETF) όπως τα SIP, RTP, TLS, UDP, αλλά και της International Telecommunication Union (ITU) όπως τα H.323 και MGCP. Η βασική της ιδέα είναι το να περνάει δεδομένα φωνής και video πάνω από τα υπάρχοντα δίκτυα IP δίνοντάς μας την δυνατότητα να δημιουργήσουμε πολλές εφαρμογές μειώνοντας παράλληλα το κόστος προς τον τελικό χρήστη.

Μία διαδεδομένη στις μέρες μας εφαρμογή της τεχνολογίας αυτής είναι τα Voice over IP softswitch. Με τον όρο softswitch εννοούμε ένα τηλεφωνικό κέντρο που δρομολογεί κλήσεις μεταξύ τερματικών τηλεφώνων καθώς και μεταξύ δικτύων μεταγωγής πακέτου και μεταγωγής κυκλώματος, το οποίο αποτελείται από μια υπολογιστική μονάδα (server) πάνω στην οποία τρέχει το κατάλληλο λογισμικό για την δρομολόγηση των κλήσεων. Ουσιαστικά ένα Voice over IP softswitch είναι ένα softswitch στο οποίο η

δρομολόγηση και η διαχείριση των κλήσεων γίνεται χρησιμοποιώντας την τεχνολογία VoIP. Στις μέρες μας υπάρχει μια πληθώρα εμπορικών softswitch από πολλές καταξιωμένες εταιρίες του χώρου αλλά επίσης και ορισμένα συστήματα Open Source αρχιτεκτονικής.

Ένα από τα πιο διαδεδομένα Open Source συστήματα VoIP softswitch είναι το Asterisk. Το Asterisk αποτελείται από κώδικα ο οποίος τρέχει πάνω στο λειτουργικό Linux. Μπορούμε να χρησιμοποιήσουμε τον κώδικα αυτό πάνω σε όποια διανομή του Linux θέλουμε είτε να κατεβάσουμε και να εγκαταστήσουμε μια από τις έτοιμες διανομές του Asterisk (AsteriskNow, Trixbox) που περιλαμβάνουν τόσο το λειτουργικό όσο και το softswitch. Το Asterisk υποστηρίζει πολλές εφαρμογές και επιπλέον ως λογισμικό ανοιχτού κώδικα μπορεί να επεκταθεί και με εφαρμογές που θα αναπτυχθούν από τον ίδιο τον Χρήστη.

Στα πλαίσια αυτής της εργασίας ασχοληθήκαμε με το Asterisk και τις εφαρμογές που μπορούν να αναπτυχθούν πάνω σε αυτό. Εγκαταστήσαμε κάποια κέντρα Asterisk και τα διαμορφώσαμε κατάλληλα ώστε να είναι πλήρως λειτουργικά. Επιπλέον αναπτύξαμε ορισμένες εφαρμογές πάνω σε αυτά, οι οποίες είναι αντίστοιχες των εμπορικών softswitch και μπορούν να χρησιμοποιηθούν πάνω σε τηλεφωνικά κέντρα Asterisk ώστε να παρέχονται επιπλέον υπηρεσίες προς τους χρήστες.

Η μεταπτυχιακή αυτή εργασία πραγματοποιήθηκε στα πλαίσια του Μεταπτυχιακού προγράμματος σπουδών με κατεύθυνση τις Ψηφιακές Επικοινωνίες και Δίκτυα του τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιά υπό την επίβλεψη του Επίκουρου καθηγητή Γεώργιου Ευθύμογλου. Θα ήθελα να ευχαριστήσω τον κύριο Ευθύμογλου για την δυνατότητα που μου έδωσε να ασχοληθώ με ένα πολύ ενδιαφέρον ερευνητικό κομμάτι όπως η τεχνολογία VoIP αλλά και το Asterisk.

ΓΑΛΕΡΙΣΤΗΜΟ ΠΕΡΑΙΑ



# **ΚΕΦΑΛΑΙΟ 1**

## **Τεχνολογία Voice Over IP (VoIP)**

### **1.1 Σύντομη περιγραφή της τεχνολογίας VoIP**

Το VoIP (Voice Over IP) αναφέρεται στην τεχνολογία της μετάδοσης της φωνής πάνω από το διαδίκτυο, αφού προηγουμένως μετατραπεί σε πακέτα. Περιλαμβάνει όλες τις τεχνολογίες τόσο σε υλικό όσο και σε λογισμικό κυρίως, που καθιστούν εφικτή την τηλεφωνική κλήση. Η τεχνολογία VoIP χρησιμοποιεί τα υπάρχοντα δίκτυα μεταγωγής πακέτου (διαδίκτυο, εταιρικά δίκτυα κλπ) πάνω από τα οποία εδραιώνει τηλεφωνικές κλήσεις.

### **1.2 Ιστορική αναδρομή**

*1995-96*

Το 1995 ήταν η χρονιά όπου πολλοί άρχισαν να καταλαβαίνουν την αξία της δυνατότητας να στέλνουν πακέτα φωνής πάνω από τα δίκτυα μεταγωγής πακέτου σε σχέση με την κλασική τηλεφωνία PSTN. Την χρονιά αυτή η Vocaltec, μια Ισραηλινή εταιρία δημιούργησε το πρώτο λογισμικό για την τηλεφωνία πάνω από το Internet, το "Internet Phone" [1]. Πρόκειται για ένα τηλέφωνο σχεδιασμένο να λειτουργεί σε έναν προσωπικό υπολογιστή. Η βασική αρχή λειτουργίας του παραμένει μέχρι και σήμερα η

βασική αρχή των τηλεφωνικών συστημάτων: Η μετατροπή του αναλογικού σήματος της ανθρώπινης φωνής σε ψηφιακά δεδομένα και η ενθυλάκωση των δεδομένων αυτών σε IP πακέτα. Η εφαρμογή απαιτούσε και από τον καλών και από τον καλούμενο να είναι συνδεδεμένοι την ίδια χρονική στιγμή. Η εφαρμογή αυτή προωθήθηκε εμπορικά από τις ίδιες τις εταιρείες που πουλούσαν υπηρεσίες πρόσβασης στο διαδίκτυο (ISP) καθώς προσέφεραν κλήσεις σε μεγάλη απόσταση με τιμές αστικής χρέωσης (η σύνδεση στο Internet γινόταν με dial up).

### 1996

Με την πάροδο ενός χρόνου οι κατασκευαστές κατάλαβαν ότι για να ξεπεραστούν οι περιορισμοί του IP δικτύου και να γίνει πιο εμπορική η νέα αυτή μορφή τηλεφωνίας, θα έπρεπε να βρεθεί κάποιος τρόπος να διασυνδεθεί το IP δίκτυο με το κλασικό τηλεφωνικό δίκτυο (PSTN). Τον Αύγουστο του 1996 αναπτύχθηκε το πρώτο λογισμικό για κλήσεις από H/Y προς το PSTN από την εταιρεία Net2Phone. Η τάση αυτής της εποχής ήταν η κατασκευή τηλεφωνικών πυλών που ένωναν τηλεφωνικές συσκευές και τηλεφωνικά δίκτυα και παράλληλα η χρησιμοποίηση της τεχνολογίας VoIP ώστε να χαμηλώσουν τα λειτουργικά κόστη των τηλεφωνικών πάροχων.

Εμπορικά, αυτή την εποχή προκύπτουν κάποιες συγχωνεύσεις μεγάλων εταιριών :

- Vocaltec με Cisco
- Net2phone με NMS Communications
- iBasis με τη Cisco

### 1996-1999

Το διάστημα αυτό αναπτύσσονται οι πύλες πολυμέσων (media gateways), το πρώτο εμπορικό προϊόν το οποίο απέφερε κέρδη στους κατασκευαστές. Η βασική λειτουργία τους ήταν να ενώσουν το VoIP δίκτυο που προσέφεραν οι ISP με τα σημεία διασύνδεσης που διατηρούσαν με τους παρόχους PSTN (POP : Point of presence). Οι μεγαλύτεροι κατασκευαστές αυτή την εποχή είναι:

- Vocaltec
- Nuera (εξαγοράστηκε από την Audiocodes)
- Vienna Systems (εξαγοράστηκε από την Nokia)



- Cisco
- Clarent (εξαγοράστηκε από την Verso)
- Ascend (εξαγοράστηκε από την Lucent).

Εκτός από τις κλασικές διεργασίες συμπίεσης/αποσυμπίεσης και μετατροπής σε πακέτα δεδομένων, οι πύλες πολυμέσων αρχίζουν και υλοποιούν λειτουργίες όπως αυτές ορίζονται από τις στοίβες πρωτοκόλλων όπως το H.323 και το SS7. Αυτή την εποχή διαχωρίζεται και ο θυροφύλακας (VoIP Gatekeeper) ως ξεχωριστή συσκευή διαχείρισης και ελέγχου των πυλών πολυμέσων. Κατά την περίοδο αυτή η τεχνολογία VoIP δεν ερχόταν σε σύγκρουση με το κλασικό PSTN δίκτυο. Η αρχική τους εφαρμογή αυτής της νέας τεχνολογίας ήταν η μερική αντικατάσταση ηλικιωμένων συστημάτων μεταγωγής κυκλώματος για την πραγματοποίηση υπεραστικών και διεθνών κλήσεων. Το 1998 οι VoIP τηλεφωνικές κλήσεις αποτελούν το 1% της παγκόσμιας τηλεφωνικής κίνησης. Επίσης κατά την περίοδο αυτή υπάρχουν αρκετές εμπορικές προπληρωμένες υπηρεσίες (prepaid platforms) και ολοκληρωμένες εφαρμογές μακρινών κυρίως αποστάσεων.

#### 1997-2000

Μόλις οι νέες αυτές τεχνολογίες κέρδισαν την εμπιστοσύνη των κατασκευαστών, άρχισαν να αναζητούν τρόπους να τις εισαγάγουν και σε μεγάλης κλίμακας εθνικά δίκτυα. Βασικός τρόπος για την ανάπτυξη ήταν η δημιουργία κεντρικών συστημάτων διαχείρισης και ελέγχου. Τέτοια συστήματα είναι οι θυροφύλακες (VoIP Gatekeepers), καθώς και λογισμικά για τον έλεγχο περιφερειακών συστημάτων και διασύνδεσής τους με τρίτα δίκτυα (soft switch). Το 1998 τρεις μεγάλοι κατασκευαστές εξοπλισμού IP (switch, router) ενσωματώνουν λογισμικό μεταγωγής VoIP (VoIP switching) σε διάφορα μοντέλα δρομολογητών. Αυτά τα συστήματα προοριζόταν για φορείς που ήδη είχαν μεγάλα IP δίκτυα κορμού, τα οποία μπορούσαν να χρησιμοποιήσουν για να παρέχουν υπηρεσίες φωνής μέσω VoIP.

Παράλληλα όλοι οι κατασκευαστές προσπαθούσαν να κατασκευάσουν soft switches. Όμως οι εμπορικές συγκυρίες της εποχής δεν ευνοούσαν την αναπροσαρμογή των ήδη υπάρχοντων εξοπλισμών. Κανένας από τους μεγάλους παρόχους δεν ήταν διατεθειμένος

να επανασχεδιάσει και να αναβαθμίσει το υπάρχον δίκτυό του. Ήταν μία εποχή που οι μεγάλοι προμηθευτές τηλεπικοινωνιακού εξοπλισμού προσπαθούσαν να βρουν την θέση τους στη νέα αγορά.

#### 1999-2000

Το H.323 αποδεικνύεται ότι δεν είναι επεκτάσιμο σε μεγάλες εφαρμογές. Αυτή την περίοδο αναπτύσσεται το SIP και αρχίζουν να δημιουργούνται οι πρώτες αρχιτεκτονικές που το υλοποιούν. Η βασική αρχή του SIP είναι ότι το στρώμα δικτύου δεν υλοποιεί καμία έξυπνη λειτουργία και όλη η ευφυΐα είναι στην εφαρμογή που τρέχει στο τερματικό.

#### 2000-2002

Το VoIP τα έχει πάει εξαιρετικά καλά από το 1995 που είχε ξεκινήσει την επανάσταση στην αγορά τηλεπικοινωνιών μέχρι την ύφεση που παρουσιάστηκε στην αγορά το 2000. Η πρώτη εμπορική αποτυχία ήταν οι πρώτες εταιρείες που προσέφεραν δωρεάν PC-to-Phone υπηρεσίες, αφού δεν μπορούσαν να καλύψουν τις PSTN χρεώσεις λόγω των μικρών κερδών από διαφημίσεις που ήταν και τα μοναδικά έσοδά τους. Επομένως κατά την ύφεση τα κόστη διασύνδεσης ανέβηκαν και τα εμπορικά μοντέλα της εποχής απέτυχαν.

Οι πάροχοι είχαν να αντιμετωπίσουν τις χρεοκοπίες των συνεργατών τους, κάτι το οποίο τους μείωνε δραματικά τα έσοδα τους. Από την πλευρά του εξοπλισμού, ύφεση παρουσιάζεται και στην αγορά των soft switch όπου πριν την ύφεση υπήρχε μια τάση προς την εξέλιξη τους.

#### 2003

Στην εποχή πριν την ύφεση υπήρχε μεγάλη επένδυση στη χωρητικότητα των συνδέσεων διαδικτύου. Λόγω της ύφεσης επιβραδύνθηκε η ανάπτυξη εφαρμογών οι οποίες θα εκμεταλλεύονταν αυτήν την αυξημένη χωρητικότητα με αποτέλεσμα να μείνει ανεκμετάλλευτη. Παράλληλα, οι συνδέσεις συνδρομητών με το διαδίκτυο αυξανόταν συνεχώς. Υπό αυτές τις συνθήκες αρχίζουν να εμφανίζονται αρκετές εφαρμογές για PC-to-PC τηλεφωνία μέσω VoIP με ανεκτή ποιότητα.

2003-2008

Παρόλο που την περίοδο αυτή υπάρχει μια μείωση των κεφαλαίων στον χώρο των τηλεπικοινωνιών, η τηλεπικοινωνιακή αγορά συνεχίζει να κατευθύνεται προς τη μεγαλύτερη χρησιμοποίηση του VoIP. Κατά την περίοδο αυτή το VoIP ωριμάζει σημαντικά. Το 2003 οι VoIP τηλεφωνικές κλήσεις έχουν φτάσει 25% της παγκόσμιας τηλεφωνικής κίνησης [2]. Η διαρκής μετακίνηση προς το VoIP πραγματοποιείται εδώ και 10 χρόνια. Όλοι οι μεγάλοι τηλεφωνικοί πάροχοι σχεδιάζουν και προχωρούν στην μετακίνησή τους προς το VoIP και μπορούμε να συμπεράνουμε ότι η μετακίνηση προς το VoIP είναι μη αναστρέψιμη. Το 2007 σχεδόν το 40% της υπεραστικής και διεθνούς τηλεφωνικής κίνησης δρομολογείται από VoIP συστήματα. Υπάρχουν οι πρώτες πολύ μεγάλης κλίμακας υλοποιήσεις νέων τεχνολογιών. Εκτός από τα σχέδια για αναβαθμίσεις και μετακινήσεις σε VoIP, παρουσιάζονται και σχέδια αντικατάστασης των υπάρχοντων παραδοσιακών τεχνολογιών με VoIP, αυξάνοντας την υπάρχουσα χωρητικότητα. Παράλληλα, οι νέα αυτή τεχνολογία διεισδύει τα τελευταία χρόνια και στους οικιακούς χρήστες ευζωνικών συνδέσεων. Επίσης η τεχνολογία VoIP κατά την διάρκεια των τελευταίων ετών έχει διεισδύσει και στον χώρο των εταιρικών τηλεπικοινωνιών. Εκατομμύρια από IP PBX αναπτύσσονται καταργώντας τα υπάρχοντα παραδοσιακά TDM τηλεφωνικά κέντρα. Μεγάλες εταιρίες του χώρου (Cisco, Siemens Enterprise Communications, Avaya, Alcatel-lucent) προσφέρουν ολοκληρωμένες λύσεις για εταιρικά περιβάλλοντα. Τα τελευταία δεκατέσσερα χρόνια το VoIP έχει γνωρίσει τεράστια ανάπτυξη. Αναμφίβολα, όπως συμβαίνει συχνά σε ταχέως αναπτυσσόμενες τεχνολογίες, παρουσιάστηκαν και μερικές υπερβολές. Παρόλα αυτά, αν αγνοήσουμε τις υπερβολές, αυτό που έχει πετύχει η τεχνολογία VoIP μπορεί να θεωρηθεί μια πραγματική επανάσταση.

### 1.3 Βασικές αρχές λειτουργίας

Οι βασικές απαραίτητες διαδικασίες για μία VoIP κλήση είναι οι εξής:

- 1) Με την χρήση ενός από τα διαθέσιμα πρωτόκολλα σηματοδότησης (SIP, H.323, IAX) γίνεται η αρχικοποίηση (initiation) της κλήσης. Κατά την διαδικασία αυτή γνωστοποιείται στον καλούντα η διεύθυνση του καλών και τούμπλαλιν, καθώς επίσης αρχικοποιούνται και διάφορες παράμετροι όπως οι πόρτες που θα χρησιμοποιηθούν, οι codec και διάφορες άλλες παράμετροι σχετικοί με τα διάφορα features που υποστηρίζει το κέντρο.
- 2) Στη συνέχεια το σήμα της αναλογικής φωνής του καλούντα μετατρέπεται σε ψηφιακή μορφή.
- 3) Στην συνέχεια έχουμε συμπίεση και μετατροπή του ψηφιακού σήματος σε διακριτά IP πακέτα.
- 4) Μετάδοση των πακέτων αυτών πάνω από το διαδίκτυο ή από ένα άλλο IP δίκτυο.
- 5) Τέλος τα πακέτα αυτά αφού ληφθούν από τον καλούντα, αποκωδικοποιούνται, αποσυμπιέζονται και τελικός μετατρέπονται σε αναλογικό σήμα φωνής στην πλευρά αυτού που λαμβάνει την κλήση.

Η ψηφιοποίηση και η μετάδοση της αναλογικής φωνής σαν μία δέσμη πακέτων πραγματοποιείται πάνω από ένα δίκτυο δεδομένων το οποίο μεταφέρει πακέτα βασισμένο στο IP πρωτόκολλο. Αυτό το δίκτυο μπορεί να είναι το εσωτερικό LAN μίας εταιρείας, μια μισθωμένη γραμμή ή ακόμα και το διαδίκτυο. Η διαδικασία της συμπίεσης πραγματοποιείται με την χρήση ενός από τους διαθέσιμους codec, όπου ανάλογα με την συμπίεση που παρέχει μειώνεται και το απαιτούμενο εύρος ζώνης.

## 1.4 Μεγέθη χαρακτηρισμού της ποιότητας

Κατά το σχεδιασμό και την υλοποίηση ενός συστήματος παροχής υπηρεσιών VoIP είναι σημαντικό να προβλέπονται και να αντιμετωπίζονται μια σειρά από παράγοντες που μπορούν να υποβαθμίσουν την ποιότητα των παρεχόμενων υπηρεσιών φωνής. Σε αυτήν την παράγραφο θα παρουσιάσουμε τους σημαντικότερους από αυτούς τους παράγοντες.

### 1.4.1 Καθυστέρηση

Ως καθυστέρηση ορίζουμε το χρόνο που χρειάζεται η φωνή από τη στιγμή που εισέρχεται στο τηλεπικοινωνιακό σύστημα μέσω του μικροφώνου του πομπού μέχρι τη στιγμή που εξέρχεται του συστήματος από το ακουστικό του δέκτη [3].

Υπάρχουν τρεις διαφορετικοί παράγοντες καθυστέρησης:

- 1) Καθυστέρηση διάδοσης: Αναφέρεται στον συνολικό χρόνο που απαιτείται για τη μετάδοση του σήματος και εξαρτάται κυρίως από την χωρητικότητα του δικτύου.
- 2) Καθυστέρηση επεξεργασίας: Αναφέρεται στην καθυστέρηση που υπεισέρχεται κατά τα στάδια επεξεργασίας του σήματος (συμπίεση, κβαντοποίηση, ενθυλάκωση σε πακέτα) από τον εξοπλισμό του τηλεπικοινωνιακού δικτύου.
- 3) Καθυστέρηση δρομολόγησης: Αναφέρεται στην καθυστέρηση των πακέτων στις ουρές των δρομολογητών (Queuing Delay).

### 1.4.2 Jitter

Ως jitter ορίζεται η μεταβολή του χρονικού διαστήματος μεταξύ της άφιξης διαδοχικών πακέτων στο δέκτη. Το jitter παρουσιάζεται μόνο σε δίκτυα μεταγωγής πακέτου και επηρεάζει κυρίως τις εφαρμογές πραγματικού χρόνου. Η αιτία αυτού του φαινομένου είναι οι διαφορετικές καθυστερήσεις στις οποίες υπόκεινται τα πακέτα που

ξεκινούν από τον πομπό στον δέκτη (Σχήμα 1.1) [4]. Αυτή η διαφορετική καθυστέρηση οφείλετε σε διάφορους παράγοντες όπως ότι τα πακέτα μπορεί να ακολουθήσουν διαφορετική διαδρομή μέσα στο δίκτυο, καθώς επίσης μπορεί να παραμείνουν τυχαίο χρόνο στην ουρά αναμονής των δρομολογητών μέχρι να εξυπηρετηθούν.



Σχήμα 1.1: Ρυθμός άφιξης πακέτων.

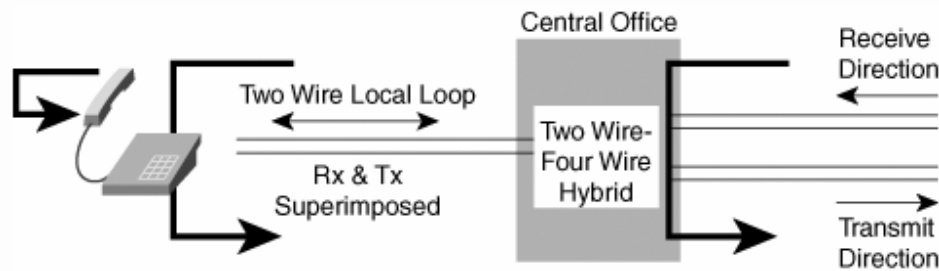
Το jitter αποτελεί ένα από τους σημαντικότερους παράγοντες υποβάθμισης της ποιότητας των VoIP τηλεφωνικών κλήσεων. Για την αντιμετώπισή του χρησιμοποιούνται τεχνικές Quality of service (QoS) στο δίκτυο πάνω από το οποίο περνάνε οι τηλεφωνικές κλήσεις. Βέβαια αυτό δεν είναι πάντα εύκολο να υλοποιηθεί, όταν για παράδειγμα το δίκτυο που χρησιμοποιούμε για τις τηλεφωνικές μας κλήσεις είναι το διαδίκτυο η εφαρμογή QoS δεν μπορεί να υλοποιηθεί σε όλο το μέρος του δικτύου. Από την άλλη σε εταιρικά δίκτυα καθώς και σε δίκτυα παρόχων αυτές οι τεχνικές διασφάλισης της ποιότητας μπορούν να υλοποιηθούν με πολύ καλά αποτελέσματα.

### 1.4.3 Ηχώ (echo)

Είναι το φαινόμενο επανάληψης του ηχητικού σήματος που εκπέμπεται στο ακουστικό του καλούντος μετά από μια μικρή καθυστέρηση [5].

Στα παραδοσιακά τηλεφωνικά δίκτυα το πρόβλημα δημιουργείται από τον κακό συνδυασμό των σύνθετων αντιστάσεων του τετρασυρματικού καλωδίου από την πλευρά του δικτύου, και του δισυρματικού καλωδίου από την πλευρά του συνδρομητή (Σχήμα 1.2).

Στα παραδοσιακά PSTN δίκτυα το πρόβλημα αντιμετωπίζεται με τις αναλογικές συσκευές καταπίεσης ηχούς που τοποθετούνται στα σημεία πολυπλεξίας των συνδρομητικών βρόχων. Αυτός ο μηχανισμός δεν είναι ιδιαίτερα αποτελεσματικός και δημιουργεί επιπλέον προβλήματα στην ποιότητα των παρεχομένων υπηρεσιών.



Σχήμα 1.2: Δημιουργία Echo σε PSTN δίκτυα [5].

Στα σύγχρονα VoIP δίκτυα ο μηχανισμός αυτός καταπίεσης της ηχούς υλοποιείται με δύο διαφορετικούς τρόπους:

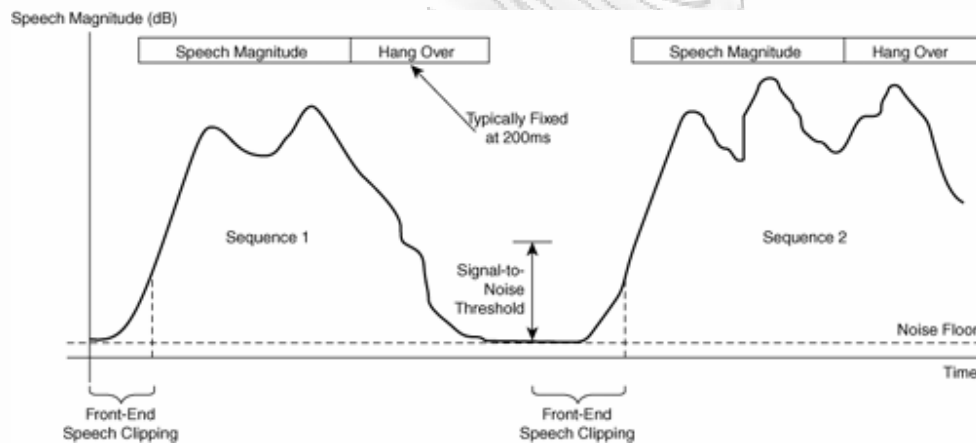
- 1) Με λογισμικό
- 2) Με συσκευές που περιέχονται στις DSP κάρτες που είναι υπεύθυνες για τη συμπίεση του ήχου και τη δημιουργία των πακέτων φωνής.

Ο μηχανισμός που λειτουργούν και οι δύο προηγούμενες επιλογές είναι παρόμοιος. Επιλέγεται ένα συγκεκριμένο χρονικό παράθυρο της τάξεως των ορισμένων millisecond, για το οποίο η πύλη που βρίσκεται στην πλευρά του καλούντος κρατά μία αντίστροφη εικόνα της ομιλίας του A-party κατά το προεπιλεγμένο χρονικό παράθυρο. Παράλληλα ο καταπιεστής ηχούς παρακολουθεί τα πακέτα που επιστρέφονται από την πλευρά του καλούμενου (λόγω του φαινομένου echo) και απομακρύνει όσα είναι όμοια με την εικόνα που έχει αποθηκευμένη στη μνήμη του.

#### 1.4.4 Λειτουργία αναγνώρισης φωνής (Voice Activity Detection -VAD)

Σε μια τυπική ανθρώπινη επικοινωνία ο ένας από τους δύο συνομιλητές μιλάει και ο άλλος ακούει. Σε ένα παραδοσιακό τηλεφωνικό δίκτυο, για κάθε τηλεφωνική συνδιάσκεψη χρησιμοποιείται ένα αμφίπλευρο κανάλι φωνής (full-duplex) με τυπικό συνολικό εύρος ζώνης 64 Kbps (από 32 Kbps σε κάθε ένα από τα δύο αμφίδρομα κανάλια επικοινωνίας). Αυτό σημαίνει ότι τουλάχιστον το ένα από τα δύο κανάλια, δηλαδή 50% του συνολικού εύρους ζώνης, παραμένει αχρησιμοποίητο. Αυτό το ποσοστό αυξάνεται αν λάβουμε υπόψη μας τις παύσεις και τις διακοπές κατά τη διάρκεια της ομιλίας ενός ατόμου.

Σε VoIP τηλεφωνικά δίκτυα η παραπάνω σπατάλη του εύρους ζώνης που καταλαμβάνει κάθε τηλεφωνική κλήση μπορεί να αποφευχθεί με την χρήση ενός μηχανισμού ανίχνευσης φωνής (VAD) [6]. Ο μηχανισμός αυτός λειτουργεί ως εξής: Επιλέγετε ένα κατώφλι πλάτους του ηχητικού σήματος κάτω από το οποίο θεωρούμε ότι δεν υπάρχει συνομιλία. Στην συνέχεια κατά την διάρκεια κωδικοποίησης και συμπίεσης του ηχητικού σήματος παρακολουθείται το πλάτος του και ανάλογα με το κατώφλι που έχουμε επιλέξει, επιλέγονται ποια στιγμιότυπα του σήματος θα κωδικοποιηθούν και ποια θα απομακρυνθούν. Στα περισσότερα συστήματα VAD όταν ανιχνεύεται ένταση ηχητικού σήματος χαμηλότερη από το επιλεγμένο κατώφλι ο κωδικοποιητής περιμένει ένα διάστημα (Hang Over), μέχρι να διακόψει την κωδικοποίηση του σήματος σε πακέτα (Σχήμα 1.3).



Σχήμα 1.3: Λειτουργία μηχανισμού ανίχνευσης φωνής (VAD) [6].



## **ΚΕΦΑΛΑΙΟ 2**

### **Βασικά Πρωτόκολλα και αλγόριθμοι κωδικοποίησης (codec) του VoIP**

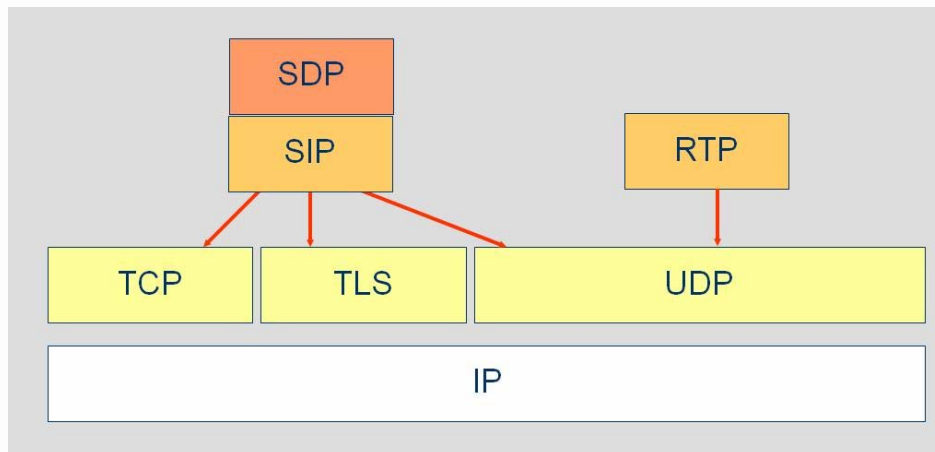
Τα πρωτόκολλα που χρησιμοποιούνται στην τεχνολογία VoIP μπορούμε να τα χωρίσουμε σε δύο βασικές κατηγορίες: Τα πρωτόκολλα για την σηματοδότηση και τα πρωτόκολλα για την μεταφορά δεδομένων. Τα βασικότερα πρωτόκολλα σηματοδότησης είναι το SIP, το H.323 και το IAX, ενώ για την μεταφορά δεδομένων υπάρχει ουσιαστικά μόνο ένα πρωτόκολλο καθώς και μια παραλλαγή του για ασφαλή μεταφορά (RTP, SRTP). Παρακάτω θα αναλύσουμε τα πρωτόκολλα αυτά καθώς και τους αλγόριθμους κωδικοποίησης δεδομένων (codecs) που χρησιμοποιούνται στο VoIP.

#### **2.1 Πρωτόκολλα σηματοδότησης**

Τα πρωτοκόλλα σηματοδότησης είναι αυτά που χρησιμοποιούνται στην εγκαθίδρυση μιας κλήσης, τον τερματισμό της καθώς και στις διάφορες εφαρμογές που ζητούνται από τους χρήστες κατά την διάρκεια μιας κλήσης. Στην συνέχεια θα παρουσιάσουμε τα πρωτόκολλα SIP, H323 και IAX. Η παρουσίαση αυτών των πρωτοκόλλων θα γίνει εν συντομία, καθώς κάτι τέτοιο δεν αποτελεί την κύρια επιδίωξη της παρούσας εργασίας.

### 2.1.1 SIP

Το πρωτόκολλο έναρξης συνόδου (Session Initiation Protocol) είναι ένα πρωτόκολλο σηματοδότησης κωδικοποιημένο σε κείμενο για την έναρξη, τη διαχείριση και τη λήξη των συνόδων φωνής καθώς και των συνόδων μεταφοράς εικόνας (video) στα δίκτυα πακέτων. Το πρωτόκολλο SIP δημιουργήθηκε από την Internet Engineering Task Force (IETF) και περιγράφεται στο RFC 3261 [7] (το οποίο αντικατέστησε το αρχικό RFC 2543). Η επικοινωνία μεταξύ των οντοτήτων που υποστηρίζονται από το πρωτόκολλο SIP είναι βασισμένη στις αιτήσεις και απαντήσεις και απαιτεί συναλλαγή μεταξύ των οντοτήτων που εμπλέκονται. Το πρωτόκολλο SIP χρησιμοποιεί το πρωτόκολλο περιγραφής συνόδου (Session Description Protocol-SDP) για τον ορισμό των δυνατοτήτων πολυμέσων. Ο τελικός στόχος μιας ανταλλαγής μηνυμάτων SIP είναι να κατασκευαστεί ένα ρεύμα RTP μεταξύ δύο συμβαλλόμενων μερών. Το πρωτόκολλο SIP προσφέρει ελευθερία κινητικότητας στον χρήστη με την δυνατότητα της χρήσης του προσδιοριστικού διευθύνσεων (SIP URI) σε διαφορετικά τερματικά. Το SIP στο μοντέλο OSI τοποθετείται πάνω από το επίπεδο μεταφοράς (transport layer). Θεωρητικά, είναι ανεξάρτητο της μεταφοράς δεδομένων, στην πράξη όμως χρησιμοποιεί UDP και TCP (Σχήμα 2.1). Επίσης στην περίπτωση που επιθυμούμε να έχουμε ασφάλεια απέναντι σε κινδύνους υποκλοπών και επιθέσεων κατά την διάρκεια εγκαθίδρυσης μιας κλήσης χρησιμοποιείται το πρωτόκολλο TLS (Transport Layer Security) για την μεταφορά των SIP μηνυμάτων. Το πρωτόκολλο TLS επιτρέπει στις εφαρμογές client/server να επικοινωνούν πάνω από ένα δίκτυο με τρόπο τέτοιο ώστε να αποτρέψουν την υποκλοπή και την παραποίηση των μηνυμάτων χρησιμοποιώντας κρυπτογραφία.



Σχήμα 2.1: Το Πρωτόκολλο SIP.

Το πρωτόκολλο SIP υποστηρίζει τόσο την μονόδρομη αποστολή δεδομένων (unicast), όσο και την περίπτωση της ταυτόχρονης αποστολής δεδομένων σε πολλά τερματικά (multicast). Η λογική της αρχιτεκτονικής του πρωτοκόλλου είναι παρόμοια με αυτή του HTTP. Έτσι το πρωτόκολλο SIP μπορεί να επεκταθεί προκειμένου να μπορεί να υπηρετεί διάφορες λειτουργίες και εφαρμογές όπως είναι ο έλεγχος και η διαχείριση τηλεφωνικών συνδέσεων, η αποστολή και λήψη μηνυμάτων, η υποστήριξη κινητικότητας καθώς και η δυνατότητα συνεργασίας με ήδη υπάρχοντα τηλεφωνικά κέντρα.

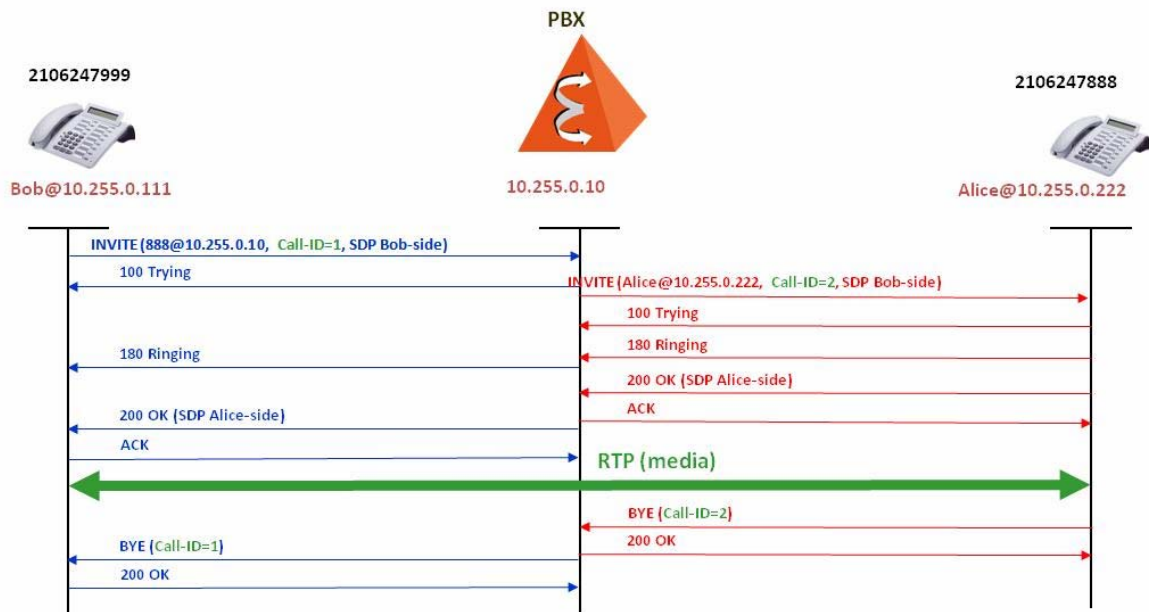
Οι τέσσερις λογικές οντότητες που υπάρχουν στην αρχιτεκτονική του πρωτοκόλλου SIP είναι οι εξής:

- 1) User Agent: Στο SIP ο User Agent (UA) ουσιαστικά αποτελεί τον τελικό χρήστη. Η βασική τους λειτουργία είναι να αποστέλλουν αιτήματα (requests) προς άλλες οντότητες και να δέχονται απαντήσεις (responses). Ο User Agent αποτελεί μία εφαρμογή στην οποία περιέχονται τόσο ο User Agent Client, όσο και ο User Agent Server. Κάποιες συσκευές που θα λειτουργούσαν σαν UA σε ένα SIP δίκτυο είναι ένα IP τηλέφωνο, ένα τηλεφωνικό gateway ή ακόμα κάποιοι τηλεφωνητές που δέχονται κλήσεις.
- 2) Registrar: Είναι ένας Server που δέχεται αιτήσεις καταγραφής από τους User Agents με σκοπό την ανανέωση της βάσης δεδομένων που περιέχει τις τοποθεσίες των χρηστών, με τα στοιχεία του User Agent που έκανε το συγκεκριμένο αίτημα.

- 3) SIP location server: Είναι ένας Server ο οποίος έχει μια βάση δεδομένων στην οποία είναι καταχωρημένη η χαρτογράφηση των SIP URI ανά πάσα στιγμή.
- 4) Proxy Server: Αποτελεί μία ενδιάμεση δικτυακή οντότητα η οποία λειτουργεί τόσο σαν Server όσο και σαν Client προκειμένου να δέχεται αιτήσεις από άλλους πελάτες (clients) και στη συνέχεια να δημιουργεί αιτήσεις εκ μέρους αυτών. Οι αιτήσεις αυτές μπορεί να εξυπηρετηθούν είτε εσωτερικά, είτε εναλλακτικά να διαβιβαστούν σε άλλους Server αφού μεταφραστούν αν αυτό είναι αναγκαίο. Ένας Proxy Server μπορεί να μεταφράζει ή ακόμα και να αναδιατυπώνει αν αυτό είναι απαραίτητο ένα αίτημα πριν το προωθήσει.
- 5) Redirect Server: Δέχεται SIP αιτήσεις και στην συνέχεια ο Redirect Server απαντά με την διεύθυνση του SIP UA για το οποίο ρωτήθηκε, παρέχοντας εναλλακτικές περιοχές όπου ο χρήστης μπορεί να είναι διαθέσιμος. Επίσης χρησιμοποιούνται για να υποστηρίξουν την ομαδική αποστολή δεδομένων σε πολλαπλούς χρήστες.

Η ακολουθία των μηνυμάτων που ανταλλάσσονται μεταξύ των User Agent για την δημιουργία και τον τερματισμό μιας βασικής κλήσης παρουσιάζονται στο παρακάτω σχήμα (Σχήμα 2.2a):

## Basic Call



Σχήμα 2.2a: Ακολουθία SIP μηνυμάτων για την έναρξη και τον τερματισμό μιας βασικής κλήσης.

Βλέπουμε ότι ο καλών στέλνει ένα μήνυμα INVITE στο κέντρο το οποίο περιέχει διάφορες παραμέτρους όπως ο αριθμός που θέλει να καλέσει, η διεύθυνση IP του καλούντα, το όνομα του κ.α. Το κέντρο στην συνέχεια στέλνει ένα μήνυμα INVITE στον καλούντα αφού έχει απαντήσει με το μήνυμα trying στον καλών. Στη συνέχεια εφόσον το τηλέφωνο του καλούντα απαντήσει με ένα μήνυμα Trying και εν συνεχεία με ένα Ringing στο κέντρο, αυτό στέλνει ένα μήνυμα Ringing στον καλών και από εκείνη την στιγμή ο καλών ακούει τόνο χτυπήματος (ringing tone). Στην συνέχεια μετά από δύο μηνύματα OK και δυο αντίστοιχα μηνύματα επιβεβαίωσης (Ack) δημιουργείται ένα κανάλι RTP όπου πλέον οι δύο συνδρομητές έχουν επικοινωνία μεταξύ τους. Ο τερματισμός της κλήσης γίνεται με την αποστολή ενός μηνύματος BYE από αυτόν που τερματίζει την κλήση. Να σημειώσουμε εδώ ότι στο πρωτόκολλο SIP το τερματικό ενσωματώνει αρκετή «ευφυΐα» έτσι ώστε να διαλέγει ποιες από τις πληροφορίες των

μηνυμάτων SIP που λαμβάνει χρειάζεται να αξιοποιήσει και απαντάει προσθέτοντας διάφορες πληροφορίες στα μηνύματα SIP που στέλνει.

Επιπλέον το πρωτόκολλο SIP παρέχει έναν μηχανισμό επικύρωσης βασισμένο στον μηχανισμό HTTP. Ο μηχανισμός αυτός ονομάζεται Digest authentication και η βασική λειτουργία του είναι κάθε φορά που ο εξυπηρετητής (server) δέχεται ένα αίτημα να ζητάει από τον πελάτη (client) να του παρέχει την πιστοποίηση της ταυτότητάς του. Ο μηχανισμός αυτός παρέχει μόνο την επικύρωση των μηνυμάτων και την αποφυγή της επανάληψής τους, ενώ δεν διασφαλίζει την ακεραιότητα και την εμπιστευτικότητα των μηνυμάτων (για τα οποία πρέπει να χρησιμοποιηθούν τα πρωτόκολλα TLS,SRTP). Αυτός ο μηχανισμός φαίνεται στο παρακάτω σχήμα (Σχήμα 2.2b) όπου έχουμε ένα SIP trace από μια βασική κλήση όπου χρησιμοποιείται digest authentication:

89	9.021509	192.168.1.64	192.168.1.66	SIP/SDP	Request: INVITE sip:2002@192.168.1.66;transport=UDP
90	9.022036	192.168.1.66	192.168.1.64	SIP	Status: 407 Proxy Authentication Required
91	9.023026	192.168.1.64	192.168.1.66	SIP	Request: ACK sip:2002@192.168.1.66;transport=UDP
92	9.023608	192.168.1.64	192.168.1.66	SIP/SDP	Request: INVITE sip:2002@192.168.1.66;transport=UDP

**Σχήμα 2.2b:** Βασική κλήση SIP όπου χρησιμοποιείται digest authentication

Βλέπουμε ότι ο καλών (192.168.1.64) στέλνει ένα μήνυμα invite στον server (192.168.1.66) και αυτός του απαντάει με ένα μήνυμα 407 proxy authentication Required, με το οποίο του ζητάει να πιστοποιήσει τον εαυτό του. Στην συνέχεια ο καλών ξαναστέλνει ένα μήνυμα invite το οποίο όμως περιέχει τα εξής επιπλέον στοιχεία με τα οποία πιστοποιείται προς τον εξυπηρετητή:

- Authentication Scheme: Digest
- Username: "2001"
- Realm: "asterisk"
- Nonce value: "103766fd"
- Authentication URI: sip:2001@192.168.1.66;transport=UDP
- Algorithm: MD5

### 2.1.2 H.323

Το πρωτόκολλο H.323 σε αντίθεση με το SIP δημιουργήθηκε από ένα καθαρά τηλεπικοινωνιακό φορέα την ITU (International Telecommunication Union). Η δημιουργία του οφείλεται στην απαίτηση για ένα τηλεπικοινωνιακό σύστημα πολυμέσων που θα λειτουργεί σε ένα δίκτυο βασισμένο στην λογική των πακέτων [8]. Τα δίκτυα αυτά καλύπτουν μία μεγάλη γκάμα περιπτώσεων, από τοπικά, εταιρικά δίκτυα μέχρι και δίκτυα δικτύων όπως είναι το διαδίκτυο. Οι βασικές οντότητες του πρωτοκόλλου H.323 όπως αυτές περιγράφονται στην προδιαγραφή της ITU είναι οι εξής [9]:

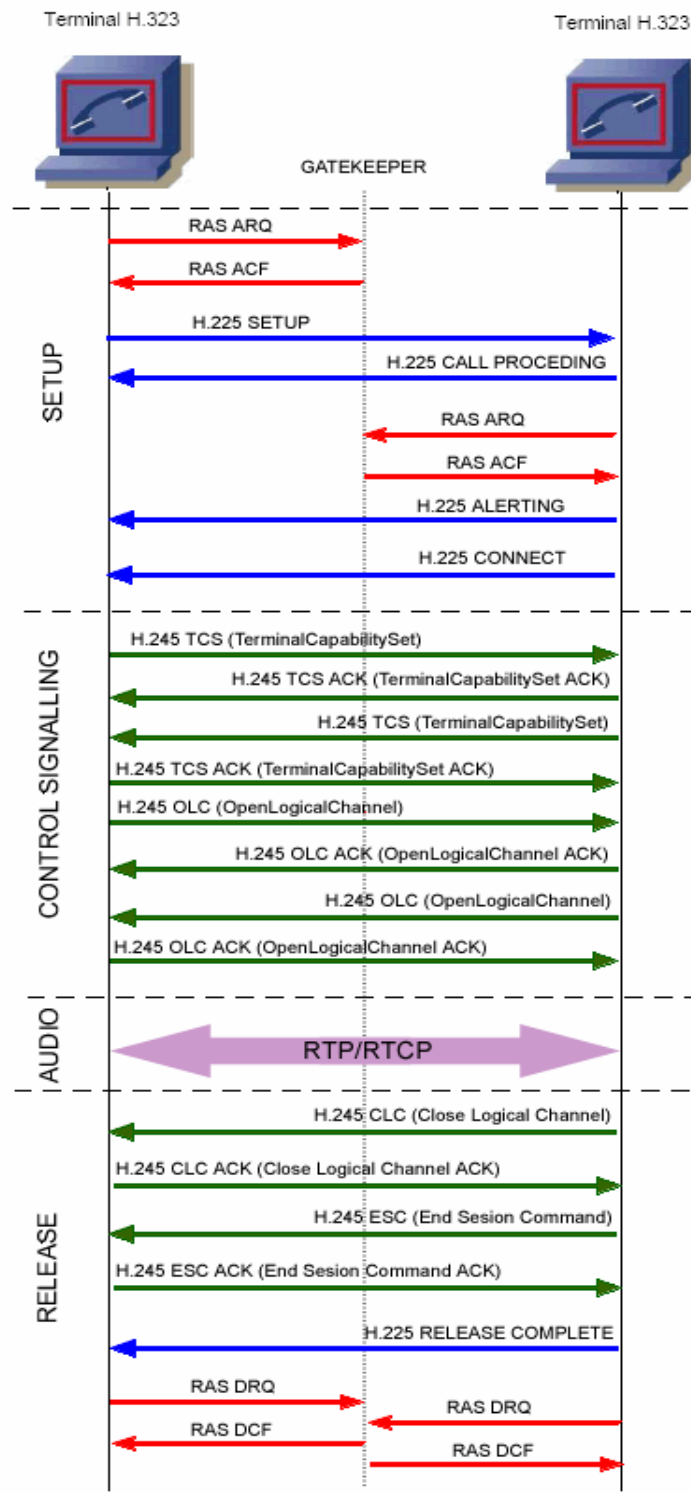
- Τερματικά που υποστηρίζουν H.323. Αποτελούν τα «άκρα» που παρέχουν επικοινωνίες φωνής και video πραγματικού χρόνου με άλλα H.323 τερματικά που βρίσκονται στο δίκτυο.
- MCU/MC/MPs. Τα Multipoint Controller Units (MCU) περιέχουν έναν Multipoint Controller (MC) και έναν ή περισσότερους Multipoint Processors (MP) που επιτρέπουν την διαχείριση συσκέψεων πολλών μερών. Το MC παρέχει έλεγχο των συσκέψεων και δίνει εντολές στα MP για τον χειρισμό των ροών δεδομένων ανάμεσα στους συμμετέχοντες.
- Gateways. Πρόκειται για συσκευές που επιτρέπουν την επικοινωνία ανάμεσα σε IP δίκτυα μεταγωγής πακέτου και σε υπάρχοντα δίκτυα μεταγωγής κυκλώματος όπως είναι το PSTN και το ISDN. Στην περίπτωση αυτή παρέχουν σηματοδότηση αλλά και μετατροπή ανάμεσα στα χρησιμοποιούμενα πρωτόκολλα.
- Gatekeepers. Επιτελούν τον ρόλο του διαχειριστή VoIP υπηρεσιών προς τα τερματικά. Οι βασικές τους λειτουργίες είναι η διευθυνσιοδότηση, η αυθεντικοποίηση του χρήστη και η εξουσιοδότηση του για χρήση των υπηρεσιών. Οι Gatekeepers χρησιμοποιούν συνήθως τεχνολογίες LDAP και DNS για την λειτουργία τους.

Το H.323 ουσιαστικά είναι ένα δομημένο σύνολο από άλλα πρωτόκολλα, όπως είναι τα εξής:

- H.225.0, το οποίο υλοποιεί την διαχείριση υπηρεσιών από τον Gatekeeper, την σύνδεση και την μεταφορά δεδομένων ανάμεσα στα τερματικά.
- H.245, το οποίο υλοποιεί την διαχείριση των κλήσεων.
- H.235, το οποίο παρέχει την απαιτούμενη ασφάλεια.
- H.450.x, το οποίο παρέχει τις συμπληρωματικές υπηρεσίες όπως είναι η μεταφορά κλήσεων και η προώθηση.

Το πρωτόκολλο H.323 στο μοντέλο OSI τοποθετείται πάνω από το επίπεδο μεταφοράς (transport layer). Θεωρητικά είναι ανεξάρτητο από την μεταφορά δεδομένων, αλλά στην πράξη τα πρωτόκολλα που αποτελούν το H.323 τρέχουν πάνω από το UDP, το ATM και το TCP. Στο παρακάτω σχήμα παρουσιάζεται η ακολουθία μηνυμάτων για την αρχικοποίηση και τον τερματισμό μιας βασικής κλήσης με την χρήση του πρωτοκόλλου H.323 (Σχήμα 2.3) [10].





Σχήμα 2.3: Ακολουθία μηνυμάτων για την έναρξη και τον τερματισμό μιας βασικής κλήσης με την χρήση του πρωτοκόλλου H.323 [10].

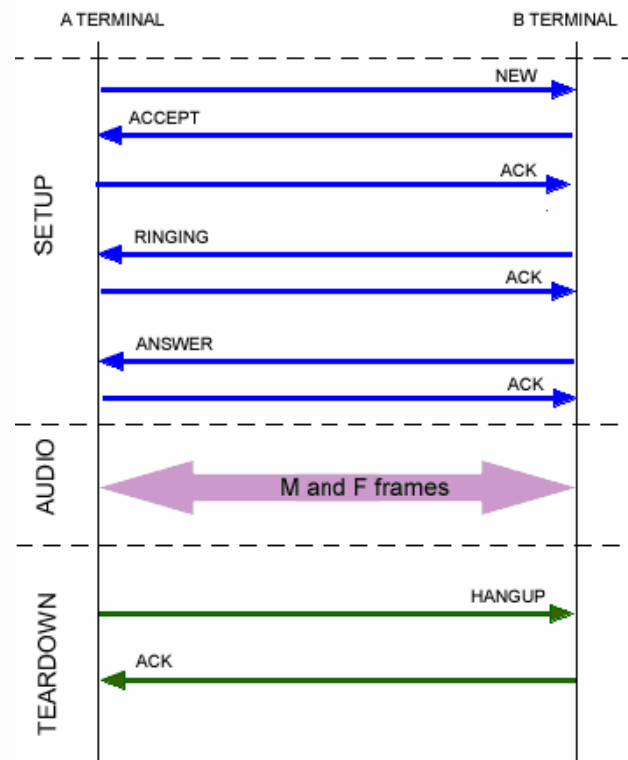
### 2.1.3 IAX

Το πρωτόκολλο IAX (Inter-Asterisk eXchange) [11] αναπτύχθηκε με στόχο να χρησιμοποιηθεί με το Asterisk PBX, αλλά πλέον υποστηρίζεται και από άλλα εμπορικά συστήματα softswitch και PBX. Χρησιμοποιείται για την δημιουργία VoIP συνδέσεων ανάμεσα σε εξυπηρετητές (servers) καθώς και για την επικοινωνία ανάμεσα σε πελάτες και εξυπηρετητές (client-server). Το πρωτόκολλο αυτό είναι ένα ουσιαστικά ένα αυτόνομο πρωτόκολλο που μπορεί να καταταχθεί τόσο στα πρωτόκολλα σηματοδοσίας όσο και στα πρωτόκολλα μεταφοράς δεδομένων, καθώς χρησιμοποιείται τόσο για την σηματοδοσία όσο και για την μεταφορά δεδομένων.

Τα βασικά χαρακτηριστικά του IAX είναι τα εξής:

- Παρέχει Βελτιστοποίηση για εφαρμογές VoIP.
- Θεωρείται πιο από το H.323 και το SIP για την μεταφορά VoIP κίνησης.
- Επιτυγχάνει μικρή «κατανάλωση» του διαθέσιμου εύρους ζώνης για την σηματοδοσία και την μεταφορά δεδομένων.
- Ενσωματώνει τεχνολογία NAT με την δυνατότητα μεταφοράς όλων των δεδομένων μέσω μίας γνωστής πόρτας UDP.
- Το πρωτόκολλο είναι πλήρες καθώς χρησιμοποιείται τόσο για την σηματοδοσία όσο και για την μεταφορά δεδομένων. Όλες οι πληροφορίες σηματοδοσίας και χρονισμού μίας κλήσης μεταφέρονται στα IAX πλαίσια (IAX frames).
- Απαιτεί μικρή υπολογιστική ισχύ.
- Είναι σχεδιασμένο για την εύκολη εγκατάσταση και χρήση.
- Μπορεί να χρησιμοποιηθεί για κάθε τύπο μεταφερόμενων δεδομένων, συμπεριλαμβανομένου και video.

Το IAX θεωρείται πως λειτουργεί πιο αποτελεσματικά σε σχέση με τα H.323 και SIP όταν λειτουργεί σε ένα μη προστατευμένο περιβάλλον. Αυτό το καθιστά ιδανικό για χρήση σε περιβάλλον με ύπαρξη NAT και κινητά τερματικά χρηστών. Στο παρακάτω σχήμα (Σχήμα 2.4) παρουσιάζεται η ακολουθία μηνυμάτων για την αρχικοποίηση και τον τερματισμό μιας βασικής κλήσης με την χρήση του πρωτοκόλλου IAX:



Σχήμα 2.4: Ακολουθία μηνυμάτων για την έναρξη και τον τερματισμό μιας βασικής κλήσης με την χρήση του πρωτοκόλλου IAX [10].

## 2.2 Πρωτόκολλα μεταφοράς δεδομένων

Τα πρωτόκολλα μεταφοράς δεδομένων χρησιμοποιούνται για την μεταφορά των δεδομένων από την εγκαθίδρυση μια συνόδου και μέχρι το τέλος αυτής. Το βασικό πρωτόκολλο που χρησιμοποιείται για την μεταφορά δεδομένων είναι το RTP (Real-time Transport Protocol) καθώς επίσης και μία παραλλαγή του για ασφαλή μεταφορά δεδομένων SRTP (Secure Real-time Transport Protocol). Ένα άλλο πρωτόκολλο που χρησιμοποιείται και για την μεταφορά δεδομένων όπως αναφέρθηκε παραπάνω είναι το IAX.

### **2.2.1 Real-time Transport Protocol (RTP)**

Το πρωτόκολλο μεταφοράς πραγματικού χρόνου (Real-time Transport Protocol - RTP) καθορίζει μια τυποποιημένο σχήμα για την μεταφορά πακέτων ήχου και εικόνας πάνω από ένα δίκτυο IP [12]. Αναπτύχθηκε από την ομάδα εργασίας μεταφοράς ήχου-βίντεο της IETF και δημοσιεύθηκε αρχικά το 1996 ως RFC 1889, ενώ αντικαταστάθηκε από το RFC 3550 το 2003.

Το RTP σχεδιάστηκε για την από άκρη σε άκρη μεταφορά στοιχείων πολυμέσων σε πραγματικό χρόνο. Το RTP ουσιαστικά απαρτίζεται από δύο πρωτόκολλα, το ίδιο το RTP καθώς και το πρωτόκολλο έλεγχου Real-time Transport Control Protocol (RTCP). Το RTP χρησιμοποιείται για τη μεταφορά των στοιχείων πολυμέσων, ενώ το RTCP χρησιμοποιείται για να στέλνονται περιοδικά οι πληροφορίες ελέγχου και οι παράμετροι του QOS (Quality of Service). Το πρωτόκολλο παρέχει την δυνατότητα καταπίεσης του Jitter καθώς της ανίχνευσης πακέτων που καταφθάνουν με λάθος σειρά.

Οι εφαρμογές πολυμέσων χρειάζονται για να λειτουργήσουν σωστά την έγκαιρη παράδοση των πακέτων και μπορούν να ανεχτούν ορισμένη απώλεια πακέτων. Για παράδειγμα η απώλεια ενός πακέτου δεδομένων φωνής έχει ως αποτέλεσμα την απώλεια ενός κλάσματος του δευτερολέπτου του ακουστικού σήματος, το οποίο όμως με κατάλληλους αλγόριθμους μπορεί να διορθωθεί. Στην περίπτωση όμως που υπάρξει συνεχείς απώλεια πακέτων τότε αυτό γίνεται αντιληπτό από τον τελικό χρήστη. Στις εφαρμογές πολυμέσων είναι σημαντικότερο να επιτυγχάνεται η έγκαιρη παράδοση των πακέτων σε σχέση με την αξιοπιστία παράδοσης αυτών. Για αυτό αν και το πρωτόκολλο TCP είναι τυποποιημένο για χρήση από το RTP συνήθως προτιμάται η χρήση του UDP για την μετάδοση των πακέτων RTP, λόγω των καθυστερήσεων στην άφιξη των πακέτων που εισάγονται από το TCP για την εγκαθίδρυση συνδέσεων και των ελέγχου των λαθών. Όπως έχει παρουσιαστεί και παραπάνω (Σχήμα 2.1) το πρωτόκολλο RTP στο μοντέλο OSI τοποθετείται πάνω από το επίπεδο μεταφοράς (transport layer).

### **2.2.2 Secure Real-time Transport Protocol (SRTP)**

Το ασφαλές πρωτόκολλο μεταφοράς πραγματικού χρόνου (Secure Real-time Transport Protocol - RTP) αποτελεί μία επέκταση του RTP η οποία προσφέρει την κρυπτογράφηση, την αυθεντικοποίηση και την ακεραιότητα μηνυμάτων κατά την μεταφορά τους πάνω από ένα δίκτυο IP [13]. Αναπτύχθηκε από μια μικρή ομάδα του πρωτοκόλλου IP καθώς και από τους ειδήμονες της κρυπτογραφίας της Cisco και της Ericsson συμπεριλαμβανομένων και των: David Oran, David McGrew, Mark Baugher, Mats Naslund, Elisabetta Carrara, Karl Norman, και Rolf Blom. Το πρωτόκολλο SRTP δημοσιεύτηκε από την IETF τον Μάρτιο του 2004 με ο RFC 3711.

Όπως και το RTP έτσι και το SRTP αποτελεί συνδυασμό δύο πρωτοκόλλων, το ίδιο το SRTP για την ασφαλή μεταφορά των δεδομένων πραγματικού χρόνου, καθώς και το Secure RTCP (SRTCP) για την αποστολή των πληροφοριών ελέγχου και των παραμέτρων του QOS. Η χρησιμοποίηση SRTP/SRTCP είναι προαιρετική αλλά ακόμα κι αν χρησιμοποιούνται τα πρωτόκολλα SRTP/SRTCP όλα τα παρεχόμενα χαρακτηριστικά γνώρισμα ασφαλείας (όπως η κρυπτογράφηση και η επικύρωση) είναι προαιρετικά και μπορούν να επιτραπούν ξεχωριστά σε συγκεκριμένους χρήστες. Η μόνη διαφορά από το RTP/RTCP είναι το χαρακτηριστικό γνώρισμα επικύρωσης μηνυμάτων που απαιτείται αναπόφευκτα κατά τη χρησιμοποίηση του SRTP/SRTCP.

## **2.3 Αλγόριθμοι κωδικοποίησης δεδομένων (codecs)**

Σε μία τηλεφωνική κλήση, πριν από την μετάδοση της φωνής πάνω από ένα IP δίκτυο πρέπει αυτή να μετατραπεί από αναλογικό σήμα σε ψηφιακή μορφή και να κωδικοποιηθεί. Επίσης γίνεται και συμπίεση των δεδομένων προκειμένου να μην υπάρχει σπατάλη στο διαθέσιμο εύρος ζώνης. Με την υποδοχή των δεδομένων φωνής στο άλλο άκρο της κλήσης ακολουθείται η αντίστροφη διαδικασία. Χρησιμοποιούνται διάφοροι αλγόριθμοι κωδικοποίησης τόσο για την φωνή όσο και για το video [14]. Τα πλέον συνηθισμένα codecs για την φωνή είναι το G.711 και το G.729, ενώ για το video

έχουμε τους H.261, H.263 και H.264. Τα codecs διαφέρουν στον αλγόριθμο που χρησιμοποιούν για την δειγματοληψία της αναλογικής φωνής και από τον τρόπο που κάνουν συμπίεση. Αυτό έχει αντίκτυπο στο απαιτούμενο εύρος ζώνης για την μετάδοση των κωδικοποιημένων δειγμάτων σε κάθε περίπτωση. Επίσης η διαφορά στο απαιτούμενο εύρος ζώνης εξαρτάται και από την χρήση η όχι κρυπτογράφησης. Στους παρακάτω πίνακες (Πίνακας 2.1 και 2.2) παρουσιάζονται αναλυτικά τα χαρακτηριστικά των βασικότερων αλγορίθμων κωδικοποίησης φωνής:

Codec	Codec Bit Rate (kbps)	Packetization Interval (ms)	Required Bandwidth (kbps)	Number of Calls Possible at a Given Link Speed		
				300 kbps	1 Mbps	2 Mbps
G.711 or G.722	64	10	99.84	3	10	20
		20	83.20	3	12	24
		30	77.65	3	12	25
G.723.1	6.4	30	17.75	18	56	112
G.726-16 or G.728	16	10	49.92	6	20	40
		20	33.28	9	30	60
		30	27.73	10	36	72
G.726-24	24	10	58.24	5	17	34
		20	41.60	7	24	48
		30	36.05	8	27	55
G.726-32	32	10	66.56	4	15	30
		20	49.92	6	20	40
		30	44.37	6	22	45
G.726-40	40	10	74.88	4	13	26
		20	58.24	5	17	34
		30	52.69	5	18	37
G.729	8	10	41.60	7	24	48
		20	24.96	12	40	80
		30	19.41	15	51	103
iLBC	15.2	20	32.45	9	30	61
		30	26.90	11	37	74
AMR	12.2	20	29.33	10	34	68
AMR-WB	23.85	20	41.44	7	24	48

**Πίνακας 2.1:** Χαρακτηριστικά των αλγορίθμων κωδικοποίησης χωρίς την χρήση κρυπτογράφησης

Codec	Codec Bit Rate (kbps)	Packetization Interval (ms)	Required Bandwidth (kbps)	Number of Calls Possible at a Given Link Speed		
				300 kbps	1 Mbps	2 Mbps
G.711 or G.722	64	10	108.16	2	9	18
		20	87.36	3	11	22
		30	80.43	3	12	24
G.723.1	6.4	30	20.52	14	48	97
G.728-16 or G.728	16	10	58.24	5	17	34
		20	37.44	8	26	53
		30	30.51	9	32	65
G.728-24	24	10	66.56	4	15	30
		20	45.76	6	21	43
		30	38.83	7	25	51
G.728-32	32	10	74.88	4	13	26
		20	54.08	5	18	36
		30	47.15	6	21	42
G.728-40	40	10	83.20	3	12	24
		20	62.40	4	16	32
		30	55.47	5	18	36
G.729	8	10	49.92	6	20	40
		20	29.12	10	34	68
		30	22.19	13	45	90
iLBC	15.2	20	36.61	8	27	54
		30	29.67	10	33	67
AMR	12.2	20	33.49	8	29	59
AMR-WB	23.85	20	45.60	6	21	43

**Πίνακας 2.2:** Χαρακτηριστικά των αλγορίθμων κωδικοποίησης με την χρήση κρυπτογράφησης

Από τους παραπάνω πίνακες μπορούμε να δούμε ότι η χρήση κρυπτογράφησης αυξάνει το απαιτούμενο εύρος ζώνης. Επίσης το απαιτούμενο εύρος ζώνης εξαρτάται από την χρονική απόσταση μεταξύ των πακέτων.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ



## **ΚΕΦΑΛΑΙΟ 3**

### **Εισαγωγή στο Asterisk**

Όπως και η πλειονότητα των προγραμμάτων ανοιχτού κώδικα, έτσι το Asterisk ξεκίνησε από την ανάγκη ενός ανθρώπου για την κάλυψη συγκεκριμένων αναγκών οικονομικά και ανεξάρτητα. Η ανάγκη για ένα σύστημα δρομολόγησης των τηλεφωνικών κλήσεων για την εταιρεία του οδήγησε τον Mark Spencer στην δημιουργία του Asterisk. Θεώρησε ότι η αγορά ενός παραδοσιακού PBX συστήματος παρουσίαζε ένα αδικαιολόγητο μεγάλο κόστος. Εξάλλου, θεωρητικά θα μπορούσε κάποιος απλά να συνδέσει τις τηλεφωνικές γραμμές σε έναν Η/Υ μέσω καρτών επέκτασης και να γίνει εκεί η επεξεργασία τους με το κατάλληλο λογισμικό. Έτσι ξεκίνησε την δημιουργία του κατάλληλου λογισμικού για τις κάρτες αυτές, το οποίο θα μπορούσε να παρέχει υπηρεσίες μεταγωγής φωνής προκειμένου να εξαλείψει την ανάγκη αγοράς ενός PBX συστήματος. Ο κώδικας που προέκυψε από αυτήν την προσπάθεια ήταν η πρώτη έκδοση του Asterisk, η οποία στη συνέχεια βελτιώθηκε και έγινε πιο ευέλικτη, καταλήγοντας στις σημερινές επίσημες εκδόσεις του Asterisk.

Στην συνέχεια ο Mark Spencer συνεργάστηκε με τον Jim Dixon της Zapata Telephony για την δημιουργία ενός οικονομικού συστήματος που θα ενσωμάτωνε κάρτες επέκτασης πάνω σε υπολογιστές της Intel προκειμένου να προκύψει μία πλατφόρμα του Asterisk με δυνατότητες διασύνδεσης στο PSTN δίκτυο. Η βασική ιδέα του εγχειρήματος, ήταν πως κάθε υπολογιστής με Linux ως λειτουργικό σύστημα με την ενσωμάτωση μιας κάρτας που θα παρείχε μερικές FXO/T1/E1 θα μπορούσε να μετατραπεί σε ένα PBX σύστημα με πλήρεις δυνατότητες με την εγκατάσταση του Asterisk.

### 3.1 Περιγραφή του Asterisk

Το Asterisk αποτελεί ένα λογισμικό που υλοποιεί ένα ανοιχτής αρχιτεκτονικής PBX (Private Branch eXchange – ιδιωτικό σύστημα τηλεφωνικού κέντρου) σύστημα. Ουσιαστικά είναι αυτό που όλες οι εταιρίες ονομάζουν Softswitch. Με την εγκατάσταση του συστήματος ο χρήστης έχει ένα πλήρως επεκτάσιμο PBX με αυξημένες δυνατότητες. Το Asterisk μπορεί να χρησιμοποιηθεί από οικιακούς χρήστες, σε εταιρικά περιβάλλοντα καθώς και από πάροχους υπηρεσιών VoIP. Σε αυτό βοηθάει ότι το Asterisk είναι ένα ανοιχτό σύστημα το οποίο μπορεί να τροποποιηθεί κατάλληλα από τον κάθε χρήστη.

Το Asterisk προσφέρει υπηρεσίες και δυνατότητες αντίστοιχες εκείνων των softswitches τα οποία πωλούνται από εταιρίες όπως Cisco, Siemens Enterprise Networks, Alcatel-Lucent, Nokia Siemens Networks κ.τ.λ. Επιπλέον συγκρινόμενο με τα κλασσικά PBX κέντρα το Asterisk προσφέρει αρκετές επιπλέον δυνατότητες με βασικότερη την διασύνδεση μεταξύ VoIP και PSTN δικτύων. Παρακάτω παρουσιάζονται κάποιες από τις δυνατότητες που μπορεί να έχουν οι χρήστες του Asterisk:

- Οι υπάλληλοι μίας εταιρείας μπορούν με την χρήση μιας σύνδεσης με το διαδίκτυο να συνδέονται στον Asterisk server (μέσω VPN στο δίκτυο της εταιρίας) και με αυτό τον τρόπο μπορεί να έχουν πρόσβαση στο εταιρικό τους τηλέφωνο καθώς και σε όλες τις άλλες υπηρεσίες που παρέχονται από τον Server.
- Μια εταιρία με την χρήση του διαδικτύου ή μισθωμένων γραμμών μπορεί να συνδέει τα γραφεία της σε κάθε μεριά του πλανήτη, μειώνοντας με τον τρόπο αυτό τα τηλεφωνικά κόστη της.
- Δίνεται η δυνατότητα στους χρήστες να έχουν υπηρεσίες φωνητικού ταχυδρομείου (Voicemail), βιντεοκλήσεων καθώς και τηλεφωνικών συσκέψεων (Teleconference).
- Είναι δυνατή η δημιουργία εφαρμογών IVR (Interactive Voice Response) ανάλογα με τις ανάγκες της εκάστοτε εταιρίας, όπου δίνεται η δυνατότητα στον

- καλούντα να έχει πρόσβαση σε υπηρεσίες και σε πληροφορίες χρησιμοποιώντας απλά τα πλήκτρα της τηλεφωνικής του συσκευής.
- Χρήστες που ταξιδεύουν, έχουν την δυνατότητα να έχουν πρόσβαση στο εταιρικό PBX από οπουδήποτε και αν βρίσκονται χρησιμοποιώντας απλά μία ευρυζωνική (Broadband) σύνδεση και έτσι μπορούν να έχουν συνεχώς την δυνατότητα να δεχτούν και να κάνουν κλήσεις από το εταιρικό τους τηλέφωνο, να έχουν πρόσβαση στο voicemail καθώς και σε όλες τις υπηρεσίες που παρέχονται από το κέντρο.
  - Μέσω των CDR (Call Detail Record) δίνεται η δυνατότητα στον διαχειριστή του συστήματος να βλέπει τα στατιστικά των κλήσεων και να χρεώνει ανάλογα τον κάθε χρήστη στην περίπτωση που το asterisk χρησιμοποιείται από πάροχο VoIP υπηρεσιών.
  - Υπάρχει η δυνατότητα αναπαραγωγής μουσικής στην αναμονή όταν έχει σχηματιστεί μία ουρά από καλούντες. Υποστηρίζεται τόσο η αναπαραγωγή μουσικών αρχείων μορφής mp3, όσο και η αναπαραγωγή που προκύπτει από άμεση ροή μουσικών δεδομένων (streaming).
  - Υποστηρίζεται η δημιουργία ουρών αναμονής από καλούντες, τις οποίες ουρές μπορούν να παρακολουθούν, να τις διαχειρίζονται και να παραλαμβάνουν κλήσεις από αυτές υπάλληλοι οι οποίοι μάλιστα θα μπορούν να έχουν και διαφορετικά επίπεδα «εξουσίας» πάνω στο σύστημα (Super Users, Administrators).
  - Ενσωμάτωση δυνατοτήτων απαγγελίας γραπτού κειμένου από το σύστημα και επιπλέον φωνητικής αναγνώρισης.

### 3.2 Η Digium και το Asterisk

Η εταιρία Digium είναι ένας από τους βασικούς υπεύθυνους της εξέλιξης του Asterisk. Η εταιρία αυτή έχει συμβάλει επίσης στην προώθηση του Asterisk αλλά είναι και ο βασικός κατασκευαστής των καρτών που χρησιμοποιούνται σε αυτό. Η Digium προσφέρει τον Asterisk σε τρεις διαφορετικές εκδόσεις με βάση την άδεια χρήσης:

- Την έκδοση Asterisk Business Edition [15]. Η έκδοση αυτή είναι μία έκδοση που απευθύνεται κυρίως σε εταιρίες και για την χρήση της απαιτείτε αγορά αφού δεν διατίθεται δωρεάν. Συνήθως χρησιμοποιείται από εταιρίες που είτε δεν θέλουν, είτε δεν μπορούν να χρησιμοποιήσουν την έκδοση GPL, λόγω του ότι δεν θέλουν να διαρρεύσει ο πηγαίος κώδικας που χρησιμοποιούν μαζί με το Asterisk. Αυτό συμβαίνει γιατί η άδεια χρήσης που συνοδεύει την έκδοση GPL προβλέπει πως κάθε προσθήκη που γίνεται στον κώδικα του Asterisk από κάθε χρήστη θα πρέπει να είναι διαθέσιμη και για όλους τους άλλους χρήστες της έκδοσης GPL.
- Την έκδοση Asterisk OEM. Η έκδοση αυτή χρησιμοποιείται κυρίως από κατασκευαστές PBX συστημάτων, οι οποίοι δεν θέλουν να αποκαλύψουν στο αγοραστικό κοινό πως το λογισμικό τους βασίζεται στον Asterisk.
- Την έκδοση ανοιχτού κώδικα (open source) η οποία διέπεται από την γενική δημόσια άδεια (General Public License Asterisk, GPL) [16]. Πρόκειται για την πιο διαδεδομένη έκδοση. Σαν έκδοση, περιλαμβάνει όλα τα χαρακτηριστικά. Η έκδοση αυτή προσφέρεται δωρεάν και οι χρήστες έχουν την δυνατότητα να τροποποιούν το λογισμικό ανάλογα με τις ανάγκες τους. Η έκδοση αυτή παρέχεται στους χρήστες είτε σε μορφή πηγαίου κώδικα για εγκατάσταση πάνω σε κάποια διανομή του Linux, είτε σε μορφή Iso image όπου όλα τα αναγκαία αρχεία (Linux, Asterisk, FreePBX GUI) περιλαμβάνονται μέσα σε αυτό. Στην δεύτερη περίπτωση ο χρήστης ακολουθώντας μια απλή διαδικασία μέσα σε μερικά λεπτά μπορεί να στήσει ένα κέντρο Asterisk το οποίο περιλαμβάνει και γραφικό περιβάλλον (FreePBX GUI) για την διαχείρισή του. Αυτή η έκδοση ονομάζεται AsteriskNOW και είναι αυτή που χρησιμοποιήθηκε στα πλαίσια αυτής της εργασίας.

### 3.3 Hardware που χρησιμοποιείται στο Asterisk

Όπως αναφέρθηκε και προηγουμένως η επεκτασιμότητα του Asterisk στηρίζεται σε διάφορες κάρτες που επεκτείνουν τις δυνατότητες του Server. Οι τηλεφωνικές κάρτες που χρησιμοποιούνται από το Asterisk κατασκευάζονται κατά κύριο λόγο από την

Digium, την Sangoma και την Varium. Στις κάρτες αυτές ακολουθείται, όπως και στο Asterisk η λογική του ανοιχτού κώδικα και έτσι μπορούν να προσαρμοστούν στις απαιτήσεις του χρήστη. Η όλη προσπάθεια της δημιουργίας καρτών επέκτασης του Asterisk ξεκίνησε από τον Jim Dixon με το project Zapata.

Το χαρακτηριστικό γνώρισμα των καρτών που χρησιμοποιούνται στο Asterisk είναι ότι δεν ενσωματώνουν DSP (Digital Signal Processing) αλλά χρησιμοποιούν το cpu του υπολογιστή στον οποίο είναι τοποθετημένες για την επεξεργασία των ροών δεδομένων, την ακύρωση ηχούς αλλά και για τις απαραίτητες μετατροπές ανάμεσα στα χρησιμοποιούμενα codecs. Το βασικό πλεονέκτημα αυτής της αρχιτεκτονικής είναι η μείωση του κόστους αυτών των καρτών που φτάνει έως το ένα πέμπτο της τιμής μιας αντίστοιχης κάρτας που ενσωματώνει DSP. Από την άλλη το μειονέκτημα που προκύπτει με την χρήση αυτής της αρχιτεκτονικής είναι η μεγάλη ανάγκη που έχουν για χρήση της CPU, ενώ αν δεν πετύχουν την βέλτιστη χρήση της CPU του υπολογιστή, προκύπτει σημαντική επίπτωση στην ποιότητα της ομιλίας.

### **3.4 Πλεονεκτήματα και μειονεκτήματα του Asterisk**

Το βασικό πλεονέκτημα του Asterisk σε σχέση με τα υπόλοιπα εμπορικά softswitches είναι ότι αποτελεί λογισμικό ανοιχτού κώδικα χωρίς να υπολείπεται όσον αφορά τις δυνατότητες που προσφέρει. Αυτό σημαίνει ότι το Asterisk κοστίζει αρκετά λιγότερο αλλά και ότι ο χρήστης μπορεί να παραγοντοποιήσει το σύστημα στις δικές του ανάγκες μιας και ο κώδικας είναι ανοιχτός. Από την άλλη για το στήσιμο και την παραμετροποίηση του Asterisk απαιτούνται κάποιες βασικές γνώσεις από τον χρήστη καθώς και χρόνος. Επίσης για την υποστήριξη όσον αφορά τα διάφορα προβλήματα που μπορούν να προκύψουν ο χρήστης μπορεί να αναζητεί βοήθεια στα φόρουμ που υπάρχουν αλλά και αυτό απαιτεί γνώσεις και χρόνο. Σε αυτόν τον τομέα τα εμπορικά συστήματα αν και αρκετά ακριβότερα προσφέρουν ένα ολοκληρωμένο πακέτο που περιλαμβάνει την εγκατάσταση ανάλογα με τις ανάγκες του χρήστη καθώς και

ολοκληρωμένη υποστήριξη κατά την χρήση του συστήματος. Μια ενδιάμεση λύση είναι η ανάθεση της εγκατάστασης του Asterisk αλλά και της υποστήριξης κατά την διάρκεια της χρήσης του σε κάποια από τις εταιρίες που ειδικεύονται σε αυτό. Επίσης εάν συγκρίνουμε το Asterisk με ένα παραδοσιακό PBX τότε θα δούμε ότι το Asterisk υπερτερεί παρέχοντας δυνατότητες όπως η δυνατότητα για IVR (Interactive Voice Response) ή η ηχογράφηση των κλήσεων.

### 3.4.1 Πλεονεκτήματα

Παρακάτω παρουσιάζονται τα πλεονεκτήματα της εγκατάστασης ενός συστήματος Asterisk.

- Μείωση κόστους. Όπως αναφέρθηκε και παραπάνω το Asterisk συγκρινόμενο με ένα από τα εμπορικά softswitch έχει πολύ μικρότερο κόστος. Από τη άλλη το κόστος ενός παραδοσιακού PBX σε σχέση με το κόστος ενός συστήματος Asterisk που περιλαμβάνει τις απαραίτητες ψηφιακές διεπαφές και τις τηλεφωνικές συσκευές είναι αντίστοιχα με το Asterisk να είναι ελάχιστα φτηνότερο. Όμως σε αυτήν την περίπτωση το Asterisk υπερέχει σε σχέση με τις εφαρμογές που παρέχει. Εκεί όμως που η χρήση του Asterisk υπερτερεί είναι το γεγονός ότι ενσωματώνει προηγμένα χαρακτηριστικά όπως είναι το voicemail, η λειτουργία ACD (Automatic Call Distribution), IVR (Interactive Voice Response) και CTI (Computer-Telephony Integration). Πέρα από τα οφέλη της επιλογής του Asterisk όμως, με την επιλογή ενός softswitch ο χρήστης κερδίζει αρκετά αφού τα τηλεφωνικά κόστη μειώνονται αρκετά ανάλογα και με την τοπολογία του συστήματος όπως θα δούμε παρακάτω.
- Αποτελεσματικό και ευέλικτο DialPlan. Το Dialplan του Asterisk είναι εύκολο στον χειρισμό ειδικότερα στις εκδόσεις με γραφικό περιβάλλον (πχ AsteriskNow) και παρέχει ευελιξία στον χρήστη. Για παράδειγμα με το Asterisk η επιλογή βέλτιστης δρομολόγησης (LCR-Least Cost Routing) είναι μία εύκολη διαδικασία σε σχέση με τα παραδοσιακά PBX όπου αυτό είναι είτε ανέφικτο είτε κοστίζει πολύ.

- Ευελιξία στην προσαρμογή του συστήματος. Λόγω του γεγονότος ότι ο κώδικας του Asterisk είναι ανοιχτός καθώς επίσης και με τις δυνατότητες επέκτασης που υπάρχουν (Κάρτες, Scripts μέσω AGI ή AMI), δίνεται στο χρήστη να φτιάξει ένα σύστημα ανάλογο των αναγκών του. Αυτός είναι και ο λόγος για τον οποίο κάποιοι πάροχοι υπηρεσιών που χρησιμοποιούν Asterisk δεν παρέχουν στον χρήστη τον κωδικό (Password) του συστήματος ή την βιβλιογραφία της διαμόρφωσής του γιατί πολύ απλά ο χρήστης μπορεί να προσθέσει εφαρμογές πάνω στις ήδη υπάρχουσες.
- Ενσωματωμένες εφαρμογές. Η πλατφόρμα του Asterisk περιλαμβάνει ενσωματωμένες στην βασική της δομή προηγμένες εφαρμογές όπως Voicemail, CTI (Computer Telephony Integration), ACD (Automatic Call Distribution), IVR (*Interactive Voice Response*), αναπαραγωγή μουσικής στην αναμονή και ηχογράφηση συνομιλιών.
- Δυνατότητα δημιουργίας νέων εφαρμογών. Η δημιουργία νέων εφαρμογών στο Asterisk είναι μια απλή διαδικασία όπου με την χρήση scripting γλωσσών προγραμματισμού όπως είναι η PHP και η Perl χρησιμοποιώντας τις AMI (Asterisk Manager Interface) και AGI (Asterisk Gateway Interface) διεπαφές. Επίσης όπως έχει ήδη αναφερθεί το Asterisk είναι μία πλατφόρμα ανοιχτού κώδικα στον οποίο μπορεί να παρέμβει ο χρήστης. Ο πηγαίος κώδικας του είναι γραμμένος σε ANSI C γλώσσα προγραμματισμού.
- Υποστήριξη του Asterisk. Γύρω από την πλατφόρμα του Asterisk έχει δημιουργηθεί μία μεγάλη και ενεργή κοινότητα από χρήστες. Η κοινότητα αυτή των χρηστών επικοινωνεί μέσω αφιερωμένων για τον λόγο αυτό sites και forums. Με τον τρόπο αυτό εντοπίζονται εξαιρετικά γρήγορα τυχόν προβλήματα (bugs) του λογισμικού και εν συνεχεία αναπτύσσονται οι λύσεις τους. Λόγω της μεγάλης διάδοσης του Asterisk καθώς και της ενασχόλησης των χρηστών του με το λογισμικό μπορούν να εντοπιστούν πάρα πολλά προβλήματά του. Χαρακτηριστικά κατά την μετάβαση από την έκδοση 1.0 στην 1.2 εντοπίστηκαν και διορθώθηκαν 3000 bugs στον πηγαίο κώδικα. Η διαδικασία αυτή σε πολύ μεγάλο βαθμό εγγυάται την αξιοπιστία και την σταθερότητα του λογισμικού. Τέλος η ύπαρξη αυτής της ενεργής κοινότητας χρηστών παρέχει στον χρήστη

την δυνατότητα να διατυπώσει ερωτήματα για προβλήματα που αντιμετωπίζει στο επίσημο forum και σε ελάχιστο χρόνο να λαμβάνει απαντήσεις με λύσεις από χρήστες από όλο τον κόσμο. Γιατί είναι πολύ πιθανό το ίδιο πρόβλημα να έχει παρουσιαστεί στο παρελθόν και σε άλλους χρήστες του Asterisk οι οποίοι μπορούν να μας βοηθήσουν με την εμπειρία του στο συγκεκριμένο θέμα.

### 3.4.2 Μειονεκτήματα

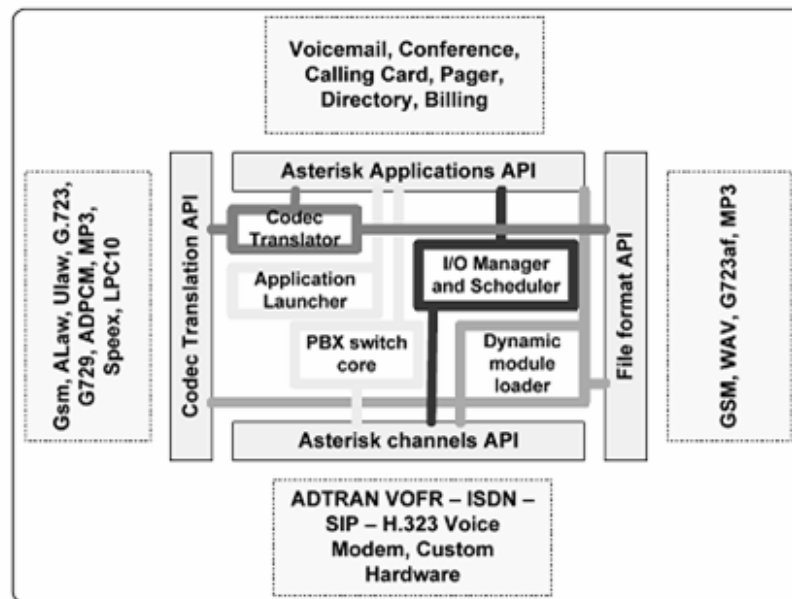
Παρακάτω παρουσιάζονται κάποια πλεονεκτήματα που παρουσιάζει η πλατφόρμα του Asterisk:

- Προβλήματα (bugs) του λογισμικού. Όπως αναφέρθηκε και προηγουμένως λόγω της εκτεταμένης χρήσης του Asterisk καθώς και των διαφόρων sites συζητήσεων γύρω από αυτό, εντοπίζονται πάρα πολλά προβλήματα κατά την χρήση του τα οποία στην συνέχεια λύνονται σε επόμενες εκδόσεις του. Αυτό όμως αποτελεί από μόνο του ένα βασικό μειονέκτημα. Δηλαδή το ότι τα προβλήματα εντοπίζονται κατά την χρήση του λογισμικού από τους χρήστες και όχι πριν την έκδοση του λογισμικού σε αυτούς είναι ένα μειονέκτημα σε σχέση με τα εμπορικά softswitches, όπου η κάθε έκδοση ενός λογισμικού περνάει από ποιοτικούς ελέγχους πριν από την παράδοσή του στους τελικούς χρήστες.
- Υποστήριξη του Asterisk. Αναφέραμε προηγουμένως την πλειάδα των ιστοσελίδων που υπάρχουν σχετικά με την υποστήριξη του Asterisk, όμως αυτός ο τρόπος υποστήριξης απαιτεί χρόνο και γνώσεις από τον χρήστη ώστε να μπορέσει να εντοπίσει το πρόβλημα και στην συνέχεια να εφαρμόσει την λύση που θα του προταθεί από τους άλλους χρήστες του Asterisk. Αντιθέτως οι εταιρίες που πωλούν τα εμπορικά softswitches παρέχουν παράλληλα έναντι αμοιβής ένα πακέτο τεχνικής υποστήριξης του πελάτη. Έτσι ο πελάτης το μόνο που έχει να κάνει είναι να περιγράψει το πρόβλημα στην εταιρία και στην συνέχεια αυτοί θα φροντίσουν για τον εντοπισμό και την λύση αυτού του προβλήματος.



### 3.5 Η Αρχιτεκτονική του Asterisk

Στο παρακάτω σχήμα (Σχήμα 3.1) παρουσιάζεται η βασική αρχιτεκτονική των λογικών οντοτήτων της πλατφόρμας του Asterisk:



Σχήμα 3.1: Η αρχιτεκτονική των λογικών Οντοτήτων της πλατφόρμας του Asterisk [17]

Η βασική αρχή σχεδίασης του Asterisk είναι η ευελιξία και η συνδεσιμότητα, όπου ορισμένα APIs καθορίζουν τον πυρήνα του PBX συστήματος. Η αρχιτεκτονική του Asterisk του επιτρέπει να χειρίζεται με τρόπο απλό και ξεκάθαρο τις εσωτερικές διασυνδέσεις ανεξάρτητα από πρωτόκολλα, κωδικοποιήσεις και τηλεφωνικό υλικό. Με τον τρόπο αυτό ο Asterisk μπορεί να χρησιμοποιήσει τα σήμερα διαθέσιμα υλικά και τεχνολογίες, αλλά και να μπορεί να ενσωματώσει αυτά που μελλοντικά θα αναπτυχθούν.

Παρακάτω παρουσιάζονται και αναλύονται οι βασικές λογικές οντότητες του Asterisk που φαίνονται στο παραπάνω σχήμα [17].

### 3.5.1 Ο πυρήνας του Asterisk

Ο πυρήνας του Asterisk αποτελείται από τις παρακάτω λογικές οντότητες:

- Μεταγωγέας PBX (PBX Switch Core). Η βασική λειτουργία του Asterisk είναι αυτή του συστήματος PBX, συνδέοντας κλήσεις μεταξύ χρηστών. Η δουλειά του πυρήνα μεταγωγής είναι να συνδέει τους χρήστες από διάφορες τις διάφορες διεπαφές του λογισμικού ή του υλικού.
- Εκτελεστής εφαρμογών (Application Launcher). Η βασική του λειτουργία είναι η εκτέλεση των εφαρμογών που παρέχουν λειτουργίες όπως είναι η αναπαραγωγή αρχείων και ο αυτόματος τηλεφωνητής.
- Μεταφραστής codec (Codec Translator). Χρησιμοποιεί modules για την κωδικοποίηση και την αποκωδικοποίηση διαφόρων τύπων συμπίεσης ήχου που εφαρμόζονται στην τηλεφωνία. Υποστηρίζονται διάφοροι codecs για να μπορέσει να επιτευχθεί μία ισορροπία μεταξύ ποιότητας ήχου και βέλτιστης χρήσης του εύρους ζώνης.
- Προγραμματιστής Χρόνου και Ελεγκτής Εισόδου/Εξόδου (I/O Manager and Scheduler). Αναλαμβάνει λειτουργίες χρονοπρογραμματισμού επιτρέποντας την επίτευξη της βέλτιστης επίδοσης σε κάθε περίπτωση φόρτου εργασίας.

### 3.5.2 APIs για την φόρτωση των Modules

Όπως φαίνεται και στο παραπάνω σχήμα (Σχήμα 3.1) υπάρχουν τέσσερα APIs τα οποία παρέχουν την διαλειτουργικότητα μεταξύ υλικού και πρωτοκόλλων. Η αρχιτεκτονική αυτή επιτρέπει στον πυρήνα του Asterisk να μην χρειάζεται να γνωρίζει τον τρόπο σύνδεσης του χρήστη, τα χρησιμοποιούμενα codecs κ.α.

Με τη χρήση αυτών των APIs, το Asterisk επιτυγχάνει πλήρη ανεξαρτησία μεταξύ της βασικής λειτουργίας του ως PBX και της πληθώρας των τεχνολογιών που υπάρχουν στο χώρο της τηλεφωνίας. Η δομή αυτή του επιτρέπει να συνεργάζεται απόλυτα τόσο με τα παραδοσιακά συστήματα (κάρτες PRI), όσο και με τις νέες τεχνολογίες μετάδοσης πακέτων φωνής (VoIP). Η δυνατότητα που έχει το Asterisk να φορτώνει ένα module για

κάθε codec, του επιτρέπει να πραγματοποιεί μετάδοση πακέτων φωνής τόσο σε δίκτυα με μικρό εύρος ζώνης (σύνδεση μέσω modem) με χρήση codec υψηλής συμπίεσης, όσο και σε ευρυζωνικά δίκτυα, παρέχοντας υψηλής ποιότητας ήχο.

Παρακάτω παρουσιάζονται τα τέσσερα APIs της πλατφόρμας του Asterisk:

- Asterisk Channels API.

Ένα κανάλι στον ψηφιακό κόσμο είναι το αντίστοιχο μιας αναλογικής τηλεφωνικής γραμμής. Στο Asterisk κάθε κανάλι αποτελεί ουσιαστικά μια σύνδεση που φέρνει μια κλήση ή δεδομένα γενικότερα προς το Asterisk. Κάθε εξερχόμενη ή εισερχόμενη κλήση χρησιμοποιεί και ένα ξεχωριστό κανάλι. Ένα κανάλι θα μπορούσε να είναι μια σύνδεση σε ένα συνηθισμένο pstn τηλέφωνο ή ένα λογικό κανάλι όπως στην περίπτωση μιας κλήσης πάνω από το διαδίκτυο με την χρήση της τεχνολογίας VoIP. Παλαιότερα όλες οι τηλεφωνικές συνδέσεις ήταν αναλογικές και κατά συνέπεια ευπαθείς στην ύπαρξη ηχούς και θορύβου. Τα τελευταία χρόνια τα περισσότερα συστήματα έχουν μετατραπεί σε ψηφιακά, με τον αναλογικό ήχο να μετατρέπεται σε ψηφιακό συνήθως με χρήση της PCM (Pulse Code Modulation) τεχνικής. Αυτή η μορφή επιτρέπει την μετάδοση φωνής με ρυθμό 64 kilobits/second χωρίς συμπίεση. Τα βασικότερα κανάλια που υποστηρίζει το Asterisk είναι τα παρακάτω:

- 1) `chan_console`. Το οποίο υποστηρίζει την ύπαρξη κάρτας ήχου στον υπολογιστή. Αποτελεί την διεπαφή με τον driver της κάρτας στην πλευρά του Asterisk.
- 2) `chan_sip`. Ένα από τα βασικά κανάλια το οποίο αποτελεί την διεπαφή για το πρωτόκολλο SIP υποστηρίζοντας την μετάδοση φωνής πάνω από το IP πρωτόκολλο (VoIP) με την χρήση του SIP .
- 3) `chan_iax`. Ένα από τα βασικά κανάλια το οποίο αποτελεί την διεπαφή για το πρωτόκολλο IAX2 υποστηρίζοντας την μετάδοση φωνής πάνω από το IP πρωτόκολλο (VoIP) με την χρήση του IAX2 .
- 4) `chan_h323`. Το H.323 είναι ένα από τα παλαιότερα και πιο συχνά χρησιμοποιημένα VoIP πρωτόκολλα. Το κανάλι αυτό είναι πολύ χρήσιμο για

την διασύνδεση του Asterisk με ένα υπάρχον H.323 δίκτυο. Στο Asterisk υπάρχουν διαφορετικές εκδόσεις του H.323, το κανάλι chan\_h323 καθώς και το chan\_oh323. Με το κανάλι chan-h323 το Asterisk μπορεί να χρησιμοποιηθεί σαν πύλη (gateway).

- 5) chan\_mgcp. Υποστηρίζει το VoIP πρωτόκολλο χρησιμοποιώντας MGCP (Media Gateway Control Protocol). Προς το παρόν το Asterisk υποστηρίζει τηλέφωνα που χρησιμοποιούν MGCP, αλλά δεν μπορεί να συνδεθεί σε έναν VoIP πάροχο που χρησιμοποιεί MGCP.
- 6) chan\_sccp. Το κανάλι αυτό αποτελεί την διεπαφή για το VoIP πρωτόκολλο της Cisco. Υπάρχουν δύο εκδόσεις, το κανάλι chan\_skinny καθώς και το κανάλι chan\_sccp2. Οι δύο εκδόσεις αυτές υποστηρίζουν την πλειοψηφία των Cisco τηλεφώνων.
- 7) chan\_unicall. Χρησιμοποιεί σαν πρωτόκολλο σηματοδότησης το MFC/R2 για E1 και χρησιμοποιείται κυρίως στην Κίνα και την Λατινική Αμερική καθώς και σε κάποιες άλλες περιοχές. Υποστηρίζεται από έναν οδηγό για το κανάλι που ονομάζεται Unicall.
- 8) chan\_agent. Το κανάλι αυτό χρησιμοποιείται για την λειτουργία ACD (Automatic Call Distribution). Δεν συνδέεται με κάποιο συγκεκριμένο hardware ή πρωτόκολλο. Επιπλέον μπορεί να χρησιμοποιηθεί για την αύξηση της κινητικότητας του συστήματος, επιτρέποντας σε οποιοδήποτε χρήστη να χρησιμοποιεί οποιοδήποτε τηλέφωνο απλά με την είσοδο του στο σύστημα.
- 9) chan\_local. Πρόκειται για ένα ψευδό-κανάλι το οποίο αντιστοιχεί σε μία διαφορετική λειτουργία του Dial plan

- **Codec Translation API.**

Συνήθως σε ένα δίκτυο δεδομένων, προσπαθούμε να χρησιμοποιούμε τον μέγιστο δυνατό αριθμό συνδέσεων φωνής επιτυγχάνοντας έτσι την καλύτερη χρησιμοποίηση του διαθέσιμου εύρους ζώνης. Τα codec αποτελούν ένα σημαντικό εργαλείο προς αυτήν την κατεύθυνση. Η συμπύεση είναι ένα από τα πλέον σημαντικά χαρακτηριστικά, αφού επιτυγχάνονται ρυθμοί συμπύεσης μεγαλύτεροι από 8 προς 1. Κάποια άλλα χαρακτηριστικά τους συμπεριλαμβάνουν την ανίχνευση έναρξης ομιλίας (Voice Activity

Detection), τον περιορισμό της απώλειας πακέτων (Packet Loss) και την προσθήκη θορύβου για καλύτερη ακρόαση (Comfort Noise Generation). Ο Asterisk υποστηρίζει μία πληθώρα από codec, τα οποία μπορεί εύκολα να μετατρέψει από ένα είδος σε ένα άλλο. Μερικά codec χρησιμοποιούνται μόνο στην λειτουργία διέλευσης (pass through mode) και αυτά τα codec δεν μπορούν να μεταφραστούν από ένα τύπο σε κάποιον άλλο.

Παρακάτω παρουσιάζονται τα βασικά codec φωνής τα οποία υποστηρίζονται από τον Asterisk:

- 1) G.711 ulaw (USA) - (64 kbps).
- 2) G.711 alaw (Europe) - (64 kbps).
- 3) G.722 – (48, 56 και 64 kbps). Μόνο στην λειτουργία διέλευσης (pass through mode). Στην έκδοση 1.4 του Asterisk υποστηρίζεται μόνο για αναπαραγωγή και καταγραφή, ενώ από την 1.6 υποστηρίζεται για όλες τις εφαρμογές.
- 4) G.723.1 – (5.3 και 6.3 kbps). Μόνο στην λειτουργία διέλευσης (pass through mode).
- 5) G.726 - Υποστηρίζεται μόνο η έκδοση με τα 32 kbps. Στην έκδοση Asterisk 1.4 διορθώνεται ένα ελάττωμα του codec αυτού καθώς επίσης εισάγονται ο G.726 AAL2 όπως και ο g726nonstandard για συμβατότητα με την έκδοση Asterisk 1.2.
- 6) G.729 - (8 kbps). Απαιτεί άδεια χρήσης ενώ είναι ελεύθερος στην λειτουργία διέλευσης (pass through mode).
- 7) GSM – (12 – 13 kbps).
- 8) iLBC – (15 kbps).
- 9) LPC10 – (2.5 kbps).
- 10) Speex – (2.15 – 44.2 kbps).

Πιο αναλυτικά στο παρακάτω σχήμα παρουσιάζονται όλοι οι κώδικες που υποστηρίζονται από την έκδοση του Asterisk που χρησιμοποιήθηκε στα πλαίσια αυτής της εργασίας (Asterisk 1.4.18.1). Η λίστα των διαθέσιμων codec δίνεται πληκτρολογώντας την εντολή “core show codecs” στην κονσόλα CLI του Asterisk.

Βλέπουμε ότι εκτός από τους codec φωνής υπάρχουν επιπλέον δύο για την αναπαραγωγή εικόνων (jpg, png) καθώς και τέσσερις για την μεταφορά Video (h261, h263, h23p, h264).

```
-----
Usage: core show codecs [audio|video|image]
       Displays codec mapping
localhost*CLI> core show codecs
Disclaimer: this command is for informational purposes only.
           It does not indicate anything about your configuration.
-----
```

INT	BINARY	HEX	TYPE	NAME	DESC
1	(1 << 0)	(0x1)	audio	g723	(G.723.1)
2	(1 << 1)	(0x2)	audio	gsm	(GSM)
4	(1 << 2)	(0x4)	audio	ulaw	(G.711 u-law)
8	(1 << 3)	(0x8)	audio	alaw	(G.711 A-law)
16	(1 << 4)	(0x10)	audio	g726aal2	(G.726 AAL2)
32	(1 << 5)	(0x20)	audio	adpcm	(ADPCM)
64	(1 << 6)	(0x40)	audio	slin	(16 bit Signed Linear PCM)
128	(1 << 7)	(0x80)	audio	lpc10	(LPC10)
256	(1 << 8)	(0x100)	audio	g729	(G.729A)
512	(1 << 9)	(0x200)	audio	speex	(SpeeX)
1024	(1 << 10)	(0x400)	audio	ilbc	(iLBC)
2048	(1 << 11)	(0x800)	audio	g726	(G.726 RFC3551)
4096	(1 << 12)	(0x1000)	audio	g722	(G722)
65536	(1 << 16)	(0x10000)	image	jpeg	(JPEG image)
131072	(1 << 17)	(0x20000)	image	png	(PNG image)
262144	(1 << 18)	(0x40000)	video	h261	(H.261 Video)
524288	(1 << 19)	(0x80000)	video	h263	(H.263 Video)
1048576	(1 << 20)	(0x100000)	video	h263p	(H.263+ Video)
2097152	(1 << 21)	(0x200000)	video	h264	(H.264 Video)

```
localhost*CLI> █
```

Σχήμα 3.2: Τα codec που υποστηρίζονται από το Asterisk

- File Format API (API αρχείων).

Η βασική δουλειά αυτού του API είναι να χειρίζεται την ανάγνωση και το γράψιμο των διάφορων μορφών αρχείων για την αποθήκευση των στοιχείων στο σύστημα αρχείων (file system) του Asterisk.

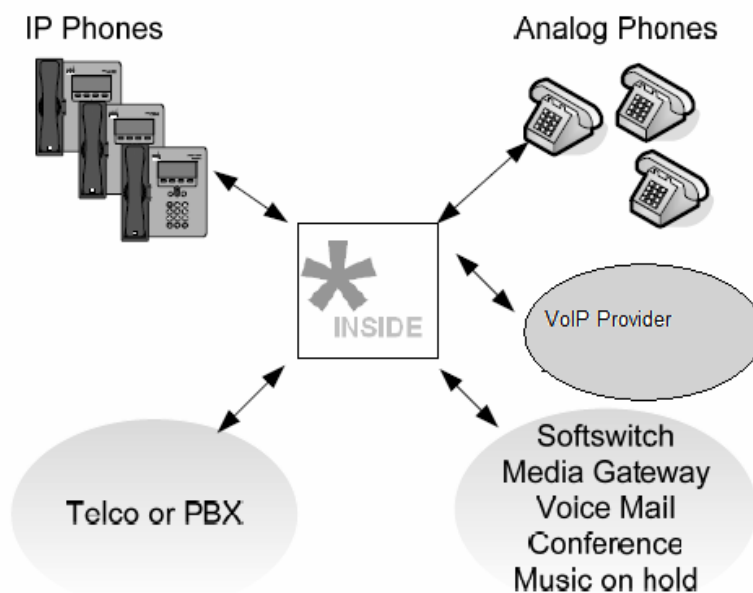
- Asterisk Applications API.

Τα περισσότερα χαρακτηριστικά του Asterisk, όπως το Voicemail και η τηλεφωνική συνδιάσκεψη αλλά και μια απλή κλήση εφαρμόζονται σαν εφαρμογές. Για παράδειγμα για να πραγματοποιηθεί μία κλήση ανάμεσα σε δύο τηλέφωνα χρησιμοποιείται μία εφαρμογή που ονομάζεται Dial(). Για να δει κάποιος τις διαθέσιμες εφαρμογές του Asterisk που διαθέτει πρέπει στην κονσόλα CLI να πληκτρολογήσει την εντολή « core

show applications». Τέλος καινούριες εφαρμογές μπορούν να εισαχθούν από το Asterisk-addons, από τρίτους προμηθευτές ή ακόμα και να τις αναπτύξει ο ίδιος ο χρήστης.

### 3.6 Περιπτώσεις χρήσης του Asterisk

Ο Asterisk είναι ένα ελεύθερο PBX το οποίο λειτουργεί σαν ένα υβριδικό PBX ενσωματώνοντας τεχνολογία TDM και IP τηλεφωνίας. Στο παρακάτω σχήμα (Σχήμα 3.3) βλέπουμε τις βασικές δυνατότητες του Asterisk [18]. Μπορούμε να δούμε ότι το Asterisk μπορεί να συνδεθεί με ήδη υπάρχοντα PBX με την χρήση αναλογικών και ψηφιακών διεπαφών όπως επίσης και το ότι υποστηρίζει αναλογικά αλλά και IP τηλέφωνα. Επίσης βλέπουμε τις λειτουργίες που ενσωματώνει (Voice Mail, Conference κλπ) καθώς επίσης ότι μπορεί να συνδεθεί με ένα VoIP Provider (για παράδειγμα VoIPDiscount) και με τον τρόπο αυτό να έχουμε χαμηλές χρεώσεις στις κλήσεις προς το pstn δίκτυο.

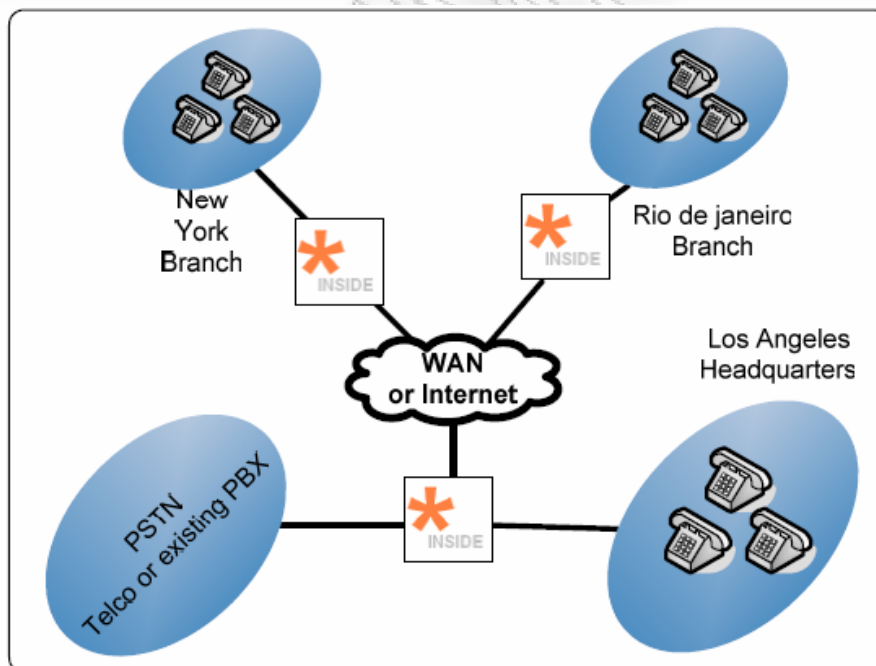


Σχήμα 3.3: Οι Δυνατότητες του Asterisk [18]

Υπάρχουν πολλές διαφορετικές υλοποιήσεις στις οποίες μπορεί να χρησιμοποιηθεί το Asterisk ανάλογα με τις ανάγκες του εκάστοτε χρήστη [18]. Παρακάτω παρουσιάζονται οι βασικότερες υλοποιήσεις στις οποίες χρησιμοποιείται το Asterisk.

### 3.6.1 Εταιρία με γραφεία σε διαφορετικές πόλεις και χώρες

Αναφερόμαστε στην υλοποίηση όπου μια εταιρία έχει γραφεία σε διάφορες πόλεις και χώρες και επιθυμεί την μείωση του κόστους της επικοινωνίας μεταξύ των υπαλλήλων [18]. Είναι μία πολύ χρήσιμη εφαρμογή της τεχνολογίας VoIP στην σύνδεση των γεωγραφικά απομακρυσμένων γραφείων μίας εταιρείας μέσω του διαδικτύου, ή μέσω ενός WAN δικτύου (μισθωμένη γραμμή). Έτσι με την χρήση του υπάρχοντος δικτύου δεδομένων της εταιρίας επιτυγχάνεται η τηλεφωνική σύνδεση όλων των γραφείων της εταιρίας. Η Υλοποίηση αυτή φαίνεται στο παρακάτω σχήμα (Σχήμα 3.4):

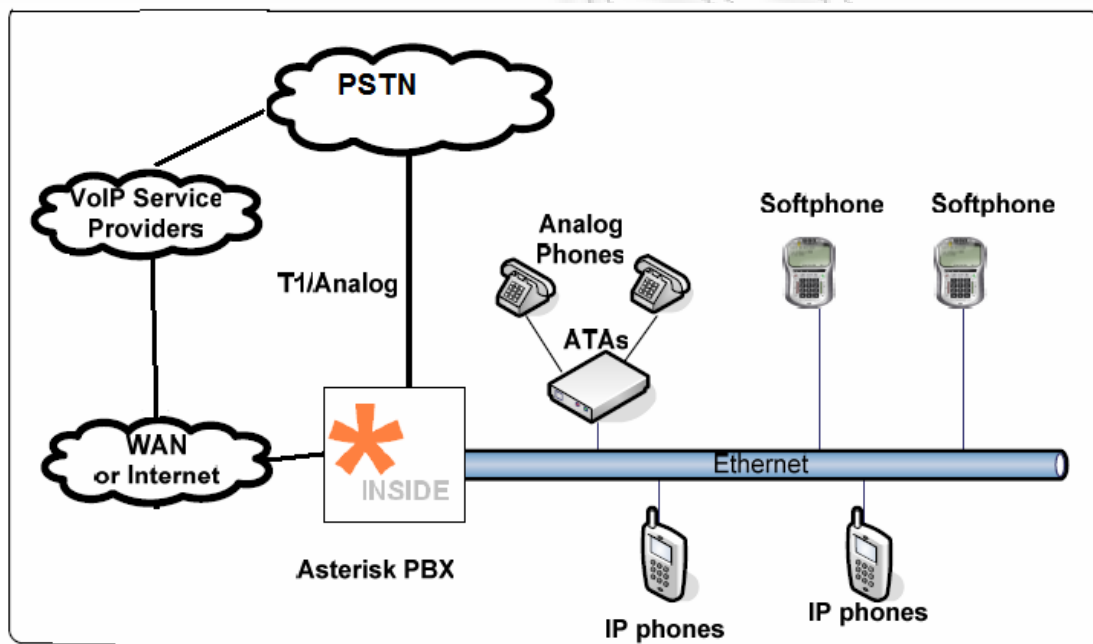


Σχήμα 3.4: Διασύνδεση Γεωγραφικά απομακρυσμένων περιοχών [18]



### 3.6.2 Χρήση ως IP PBX (softswitch)

Μία από τις πιο συνηθισμένες υλοποιήσεις του Asterisk είναι η χρήση του ως IP τηλεφωνικό κέντρο [18]. Αυτή η υλοποίηση προσφέρει μια πολύ οικονομική λύση η οποία όμως περιλαμβάνει αρκετά χαρακτηριστικά υψηλών προδιαγραφών. Σε αυτήν την υλοποίηση μπορούν να χρησιμοποιηθούν τηλέφωνα IP, softphones καθώς και μετατροπείς ATA (Analog Telephony Adapter) οι οποίοι επιτρέπουν την χρήση απλών αναλογικών συσκευών χωρίς την χρήση κάρτας PCI με θύρες FXO/FXS. Επίσης όπως βλέπουμε στο παρακάτω σχήμα (Σχήμα 3.5) υπάρχει και η δυνατότητα διασύνδεσης με το pstn δίκτυο μέσω μιας PCI κάρτας (T1/E1) ή μέσω κάποιου παρόχου VoIP (πχ VoIPDiscount) ο οποίος τελικά τερματίζει την εξερχόμενη κλήση μας στο pstn δίκτυο.

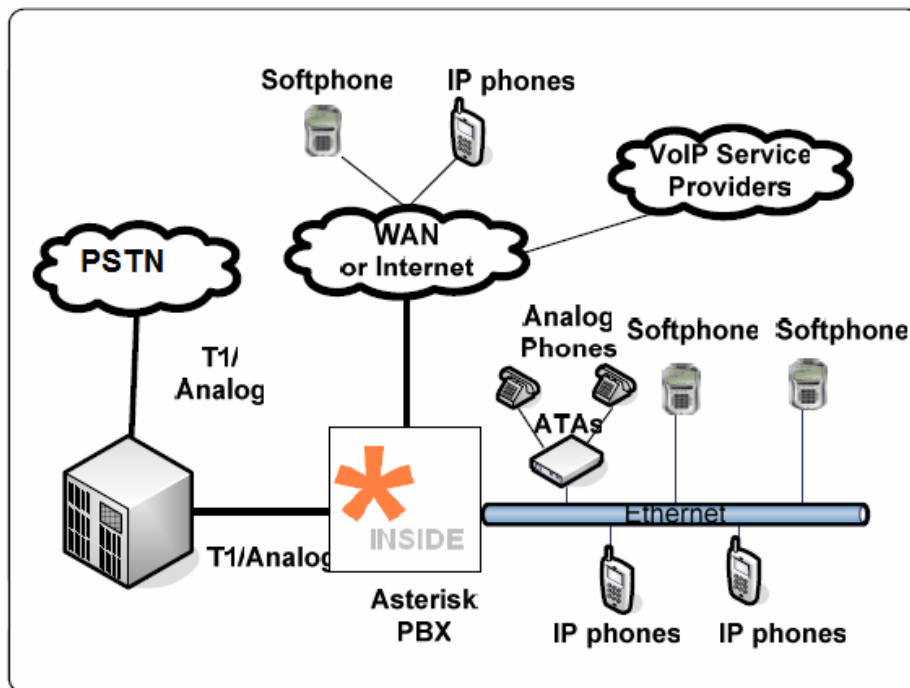


Σχήμα 3.5: Χρήση του Asterisk ως IP τηλεφωνικό κέντρο [18]

### 3.6.3 Χρήση ως Softswitch παράλληλα με ένα συμβατικό PBX

Μία επίσης πολύ διαδεδομένη υλοποίηση μιας και οι περισσότερες εταιρίες όταν επενδύουν σε νέες τεχνολογίες δεν επιθυμούν να παίρνουν μεγάλο ρίσκο καθώς επίσης και να δαπανούν μεγάλα κεφάλαια, είναι η χρήση του Asterisk παράλληλα με το υπάρχον συμβατικό PBX [18]. Η υλοποίηση αυτή αποτελεί μια πολύ αποδοτική λύση σε αυτές τις

περιπτώσεις. Σε αυτήν την υλοποίηση το Asterisk συνδέεται με το PBX χρησιμοποιώντας T1/E1 γραμμές (με την χρήση μιας κάρτας PCI). Και σε αυτήν την υλοποίηση ισχύουν όλα τα πλεονεκτήματα που αναφέρθηκαν παραπάνω για την χρήση του Asterisk ως softswitch. Μία επιπλέον δυνατότητα που παρουσιάζεται στο παρακάτω σχήμα (Σχήμα 3.6) και η οποία ισχύει και στην περίπτωση χρήσης του Asterisk ως απλού softswitch είναι η δυνατότητα χρήσης ενός εταιρικού IP τηλεφώνου ή softphone από διαφορετική γεωγραφική θέση, το οποίο μπορεί να κάνει register στο Asterisk μέσω ενός WAN δικτύου ή ακόμη και μέσω του Internet.

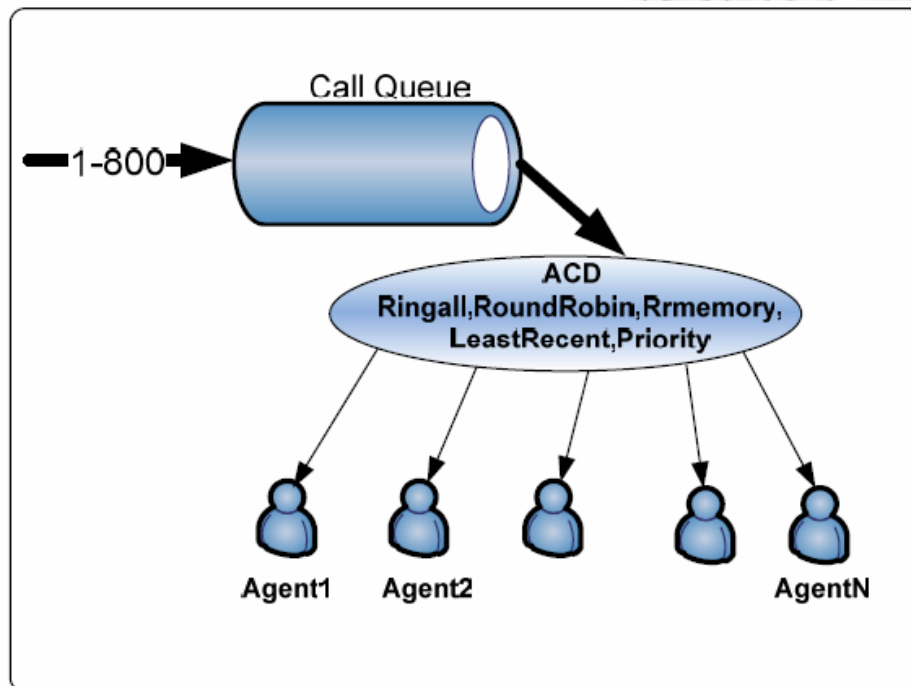


Σχήμα 3.6: Χρήση του Asterisk ως IP τηλεφωνικό κέντρο παράλληλα με κάποιο συμβατικό PBX [18]

#### 3.6.4 Χρήση ως τηλεφωνικό κέντρο εξυπηρέτησης πελατών

Ένα τηλεφωνικό κέντρο αποτελεί μία πολύπλοκη υλοποίηση η οποία συνδυάζει διάφορες τεχνολογίες όπως ACD (Automatic Call Distribution), IVR (Interactive Voice Response), εποπτεία κλήσεων (Call Supervision) και άλλες. Το Asterisk έχει ενσωματωμένη την λειτουργία ACD για κλήσεις που βρίσκονται σε ουρά. Επίσης η λειτουργία IVR μπορεί να πραγματοποιηθεί με την χρήση AGI (Asterisk Gateway

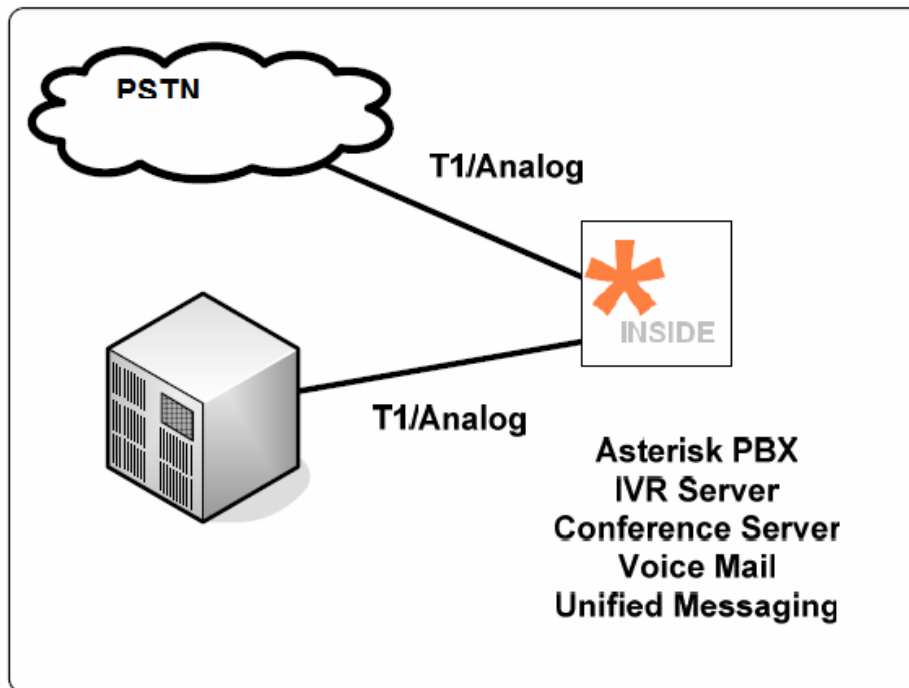
Interface) ή μέσω άλλων εσωτερικών μηχανισμών. Η λειτουργία CTI (Computer Telephony Integration) μπορεί να πραγματοποιηθεί με την χρήση AMI (Asterisk Manager Interface) ενώ τέλος η ηχογράφηση και η αρχειοθέτηση των κλήσεων είναι εφαρμογές ενσωματωμένες στο Asterisk. Στην εικόνα που ακολουθεί (Σχήμα 3.7) μπορούμε να δούμε την χρήση του Asterisk σαν τηλεφωνικό κέντρο [18].



Σχήμα 3.7: Χρήση του Asterisk ως τηλεφωνικό κέντρο εξυπηρέτησης πελατών [18]

### 3.6.5 Χρήση ως εξωτερικός Server εφαρμογών

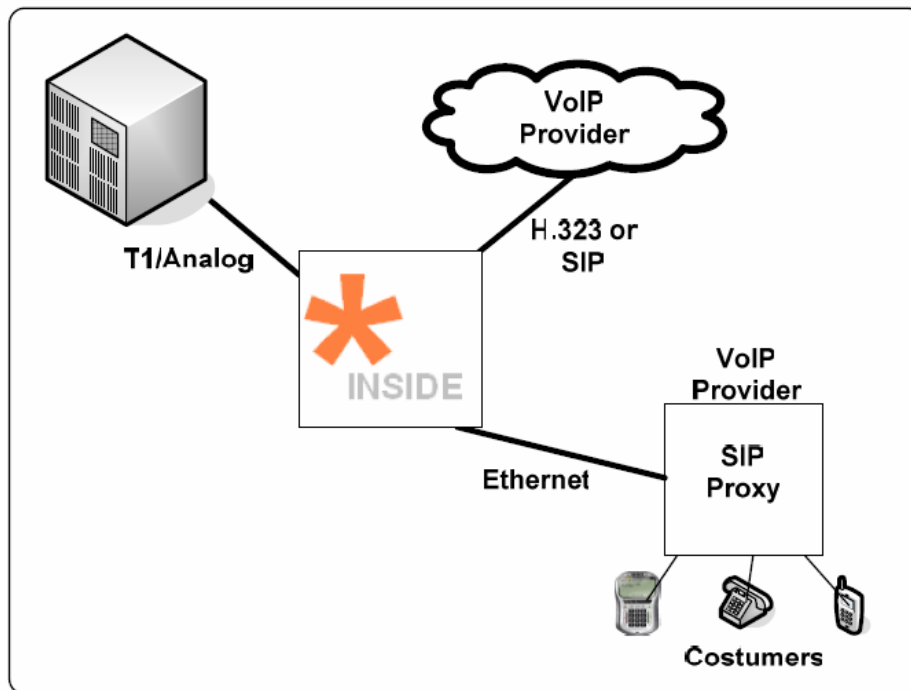
Το Asterisk μπορεί επίσης να χρησιμοποιηθεί σαν εξωτερικός Server εφαρμογών για ένα υπάρχον PBX ή ακόμα να συνδεθεί απευθείας στο PSTN δίκτυο. Σε αυτήν την υλοποίηση ο βασικός ρόλος του Asterisk είναι να προσφέρει υπηρεσίες όπως Voicemail, υποδοχή Fax, ηχογράφηση κλήσεων, IVR συνδεδεμένο σε μια βάση δεδομένων και σαν Server τηλεφωνικής συνδιάσκεψης [18]. Εάν μάλιστα ολοκληρωθεί το Voicemail και το Fax σε ένα υπάρχον e-mail, τότε προκύπτει ένα ενοποιημένο σύστημα μηνυμάτων (Unified Messaging System) κάτι που διαφορετικά αποτελεί μία πολύ ακριβή υλοποίηση.



Σχήμα 3.7: Χρήση του Asterisk ως εξωτερικός server εφαρμογών [18]

### 3.6.6 Χρήση του Asterisk ως Media Gateway

Οι περισσότεροι πάροχοι VoIP υπηρεσιών χρησιμοποιούν έναν SIP proxy για την καταγραφή της τοποθεσίας και την αυθεντικοποίηση όλων των SIP χρηστών. Πέρα από αυτό πρέπει όμως να δρομολογούν την κλήση απευθείας στο PSTN δίκτυο, ή να την δρομολογήσουν μέσω ενός παρόχου τερματισμού κλήσεων χρησιμοποιώντας SIP ή H323 πρωτόκολλο για την VoIP σύνδεση. Το Asterisk μπορεί να χρησιμοποιηθεί ως B2BUA (back to back user agent) ή ως Media Gateway υποκαθιστώντας τα πολύ ακριβότερα εμπορικά συστήματα [18]. Το Asterisk κοστίζει πολύ λιγότερα χρήματα σε σχέση με τα εμπορικά συστήματα και μπορεί να διαχειριστεί αρκετά πρωτόκολλα σηματοδοσίας καθώς και codecs όπως είδαμε στις προηγούμενες ενότητες.



Σχήμα 3.7: Χρήση του Asterisk ως Media Gateway [18]

### 3.7 Χαρακτηριστικά, δυνατότητες και λειτουργίες του Asterisk

Όπως έχει γίνει κατανοητό έως τώρα το Asterisk παρέχει μια πληθώρα δυνατοτήτων και χαρακτηριστικών άμεσα συγκρίσιμων με τα αντίστοιχα εμπορικά συστήματα. Επίσης όπως έχει επισημανθεί και προηγουμένως το Asterisk είναι απολύτως επεκτάσιμο και ο κάθε χρήστης μπορεί να αναπτύξει τις δικές του εφαρμογές. Επιπλέον αν συνυπολογίσουμε και τις δυνατότητες του Asterisk όπως: οι πολλαπλοί τύποι VoIP καναλιών, τα διάφορα υλικά διασύνδεσης, η γλώσσα δέσμης ενεργειών (AGI Scripting language), η Διασύνδεση Προγράμματος Εφαρμογής (API) κ.α, καταλαβαίνουμε ότι είναι δύσκολο να προσδιορίσουμε πλήρως τα χαρακτηριστικά και τις δυνατότητες του asterisk μιας και αυτές εξαρτώνται και από τις εκάστοτε επιθυμίες των χρηστών. Έτσι παρακάτω παρουσιάζονται μερικές από τις βασικές δυνατότητες του Asterisk [19].

- Alarm Receiver: Δυνατότητα ειδοποίησης ανάλογα με κάποια προσαρμοσμένα όρια που αφορούν την απόδοση του τηλεφωνικού μας κέντρου (π.χ. μεγάλη αναμονή).
- ADSI On-Screen Menu System: Εμφάνιση μενού σε screen phones μέσω του αναλογικού δικτύου για παροχή προσαρμοσμένων λειτουργιών.
- Automated Attendant: Επιτρέπει σε κάποιον να πληκτρολογήσει έναν κεντρικό αριθμό και στη συνέχεια να πληκτρολογήσει τον κωδικό κάποιας υπηρεσίας ή κάποιας extension. Μπορεί να χρησιμοποιηθεί σε συνδυασμό με το Dial by Name (βλ. παρακάτω) για να παρέχει π.χ. τη δυνατότητα κλήσης με χρήση ονόματος.
- Blacklists: Δημιουργία μαύρης λίστας εισερχομένων κλήσεων (συνήθως με χρήση caller id) και ξεχωριστή διαχείριση της ανάλογα με προσαρμοσμένους κανόνες
- Call Detail Records: Αρχείο καταγραφής κλήσεων με στοιχεία όπως η ώρα έναρξης της κλήσης, η διάρκεια της κλήσης, το νούμερο του καλούντα, την κατάσταση της κλήσης, κ.α.
- Call Forward: Προώθηση κλήσεων κατά βούληση ή ανάλογα με την κατάσταση (Κατειλημμένο, Δεν απαντά, κ.λ.π.)
- Call Monitoring: Παρακολούθηση κλήσεων σε πραγματικό χρόνο ή καταγραφή τους για διασφάλιση ποιότητας υπηρεσιών.
- Call Parking: Στάθμευση της κλήσης σε ένα εικονικό νούμερο το οποίο χρησιμοποιείται σαν χώρος στάθμευσης των κλήσεων και επανάκτηση της κλήσης κατά βούληση
- Call Queuing: Ουρές αναμονής κλήσεων με δυνατότητα αναπαραγωγής μουσικής ή ανακοινώσεων κατά τη διάρκεια αναμονής.
- Call Recording: Ηχογράφηση κλήσεων σε πραγματικό χρόνο
- Call Transfer: Μεταφορά κλήσεων από ένα νούμερο σε ένα άλλο
- Call Waiting: Αναμονή κλήσεων με δυνατότητα αναγνώρισης κλήσης της δεύτερης γραμμής, μουσική κατά τη διάρκεια της αναμονής και προώθηση κλήσης που βρίσκεται στην αναμονή.
- Caller ID: Αναγνώριση κλήσης

- Calling Cards: Δυνατότητα παροχής υπηρεσιών τηλεφωνίας με χρήση προπληρωμένων καρτών ή γενικότερα προπληρωμένων λογαριασμών
- Dial by Name: Δυνατότητα κλήσης με χρήση ονόματος αντί για νούμερο.
- Direct Inward System Access: Δυνατότητα απομακρυσμένης σύνδεσης σε λειτουργίες που είναι διαθέσιμες μόνο σε τοπικές extensions.
- Distinctive Ring: Δυνατότητα αλλαγής του ρυθμού κουδουνίσματος του τηλεφώνου ανάλογα με το ποιος καλεί
- Distributed Universal Number Discovery (DUNDI): Χρήση του DUNDI για εύρεση τηλεφώνου μέσω ερώτησης σε κάποιον γνωστό μας σύνδεσμο.
- ENUM: Χρήση του ENUM για ενοποίηση του τηλεφωνικού συστήματος αριθμοδότησης (E.164) με το σύστημα διευθυνσιοδότησης του διαδικτύου (DNS) και έμμεση αναζήτηση.
- Fax Transmit and Receive: Αποστολή/Λήψη Fax και προώθηση στο email
- Flexible Extension Logic: Ευέλικτη και παραμετροποιήσιμη αριθμοδότηση και διαχείριση των κλήσεων.
- Macros: Αυτόματη εκτέλεση πολύπλοκων πολλαπλών ενεργειών που εκτελούνται συχνά για εξοικονόμηση χρόνου και αποφυγή λαθών.
- Predictive Dialer: Αυτόματη κλήση σε τηλεφωνικά νούμερα. Χρησιμοποιείται σε τηλεφωνικά κέντρα (Τήλε-μάρκετινγκ) και πραγματοποιεί κλήσεις προς πιθανούς πελάτες με χρήση εξειδικευμένων αλγορίθμων πρόβλεψης.
- Open Settlement Protocol (OSP): Δυνατότητα τιμολόγησης VoIP υπηρεσιών
- Roaming Extensions: Δυνατότητα περιαγωγής ενός extension σε οποιοδήποτε σημείο του κόσμου με πρόσβαση σε τηλεφωνικό δίκτυο ή στο internet.
- Call Routing: Δρομολόγηση της κλήσης ανάλογα με τον αριθμό αυτού που καλεί, την ώρα κλήσης, το κόστος κλήσης, κ.α.
- SMS Messaging: Αποστολή γραπτών μηνυμάτων
- Streaming Media Access: Δυνατότητα βιντεοκλήσης,
- Talk Detection: Αναγνώριση ομιλίας με χρήση του sphinx
- Text-to-Speech: Εκφώνηση κειμένου μέσω του Festival, Cepstral κ.α.

- VoIP Gateways: Δυνατότητα διασύνδεσης VoIP τερματικών ανεξαρτήτως πρωτοκόλλου που χρησιμοποιεί το καθένα και μετατροπή των μη συμβατών πρωτοκόλλων.
- Voicemail: Αυτόματος τηλεφωνητής με δυνατότητες ειδοποίησης νέων μηνυμάτων μέσω email, αποστολής του μηνύματος ως προσάρτηση σε email, οπτικής απεικόνισης νέων μηνυμάτων στα τερματικά, οργάνωσης σε φακέλους, ομαδικής αποστολής και απομακρυσμένης διαχείρισης.
- Zapateller: Χρήση ειδικού ήχου (*Special Information Tone*) για αποφυγή τηλεφωνημάτων από αυτόματες κλήσεις π.χ. Τήλε-μάρκετινγκ.
- AGI (Asterisk Gateway Interface): Δυνατότητα αλληλεπίδρασης εξωτερικών προγραμμάτων με το Asterisk. Πλήρης διαχείριση του συστήματος σε επίπεδο λειτουργιών και κονσόλας
- Echo cancellation: Δυνατότητα εξάλειψης του φαινομένου της ηχώ με χρήση εξελιγμένων αλγορίθμων.
- Codecs: Υποστήριξη πληθώρας μεθόδων κωδικο-αποκωδικοποίησης και συμπίεσης όπως: ADPCM, G.711 (A-Law & μ-Law), G.722, G.723.1 (pass through), G.726, G.729(με αγορά άδειας χρήσης), GSM, iLBC, Linear, LPC-10, Speex
- Protocols: Υποστήριξη πληθώρας πρωτοκόλλων όπως: IAX (Inter-Asterisk Exchange), H.323, SIP (Session Initiation Protocol), MGCP (Media Gateway Control Protocol) και τέλος SCCP (Cisco Skinny).



## **ΚΕΦΑΛΑΙΟ 4**

### **Εγκατάσταση και διαμόρφωση του Asterisk**

Στο κεφάλαιο αυτό θα περιγράψουμε την διαδικασία της εγκατάστασης και της διαμόρφωσης του συστήματος Asterisk που στήθηκε στα πλαίσια αυτής της εργασίας. Χρησιμοποιήθηκε το Asterisknow 1.4 το οποίο είναι ένα iso για εγκατάσταση και περιέχει την έκδοση 1.4.18.1 του Asterisk. Για την εγκατάσταση του Asterisk χρησιμοποιήθηκε το πρόγραμμα VMware Workstation έτσι ώστε να μπορούμε να χρησιμοποιήσουμε 1 Η/Υ τόσο ως Asterisk server όσο και ως Softphone.

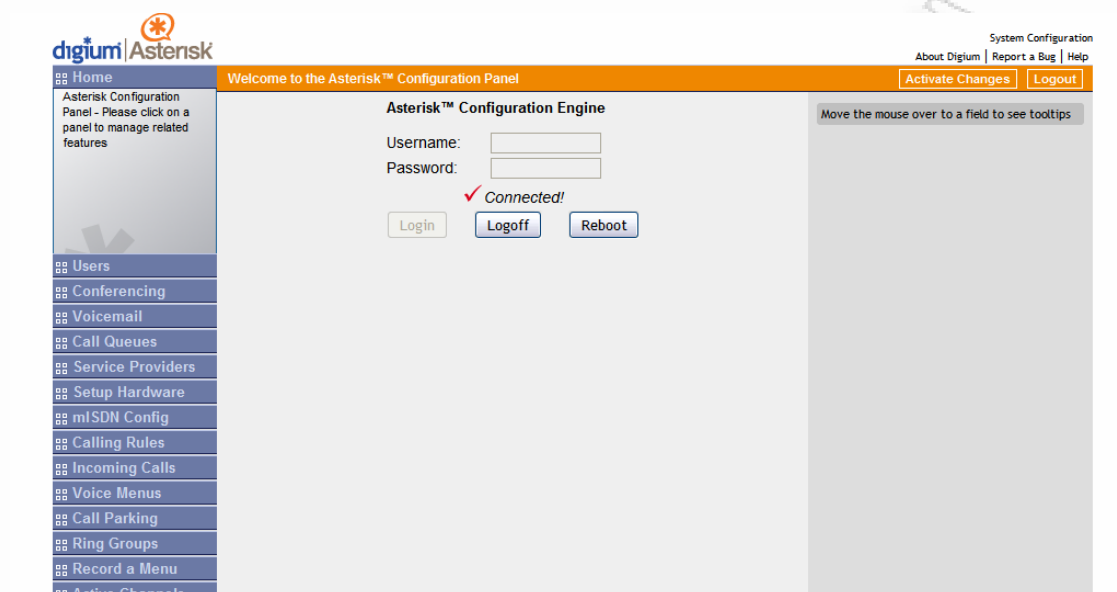
#### **4.1 Εγκατάσταση του Asterisk**

Για την εγκατάσταση του Asterisk υπάρχουν 2 διαφορετικοί τρόποι, είτε να εγκαταστήσουμε τις binaries του Asterisk σε ένα σύστημα που τρέχει Linux είτε να χρησιμοποιήσουμε μια από τις διανομές iso όπως το Asterisknow [20] ή το Trixbox [21]. Στην δεύτερη περίπτωση, όπου είναι και αυτή που υλοποιήσαμε σε αυτή την εργασία, η εγκατάσταση του Asterisk είναι μια σχετικά απλή διαδικασία όπου αφού επιλέξουμε στο bios του Η/Υ να κάνει boot από την μονάδα οπτικού δίσκου, τοποθετούμε το image του Asterisknow και επανεκινούμε το σύστημα. Στην συνέχεια ακολουθούμε κάποια βήματα μέσω γραφικού περιβάλλοντος (τα οποία είναι παρόμοια με την εγκατάσταση μια διανομής Linux) και σε 15 με 20 λεπτά ολοκληρώνεται η εγκατάσταση. Μετά την εγκατάσταση έχουμε ένα σύστημα στο οποίο υπάρχει και γραφικό περιβάλλον για την παραμετροποίηση και την διαχείριση του συστήματος.

Όπως αναφέρθηκε και παραπάνω χρησιμοποιήθηκε το πρόγραμμα VMware Workstation με το οποίο δημιουργούμε ένα εικονικό τερματικό πάνω στο φυσικό μας τερματικό. Έτσι μπορούμε να έχουμε ενεργό τον Asterisk Server μας και παράλληλα να τρέχουμε και άλλες εφαρμογές στο σύστημά μας (όπως ένα softphone). Αυτή η υλοποίηση όμως έχει ένα βασικό μειονέκτημα, το ότι ο server μας δεν έχει στην διάθεσή του αρκετή υπολογιστική ισχύ και έτσι καθυστερεί όταν πρέπει να εκτελέσει κάποιες «βαριές» εφαρμογές. Είναι φανερό ότι αυτή η υλοποίηση δεν μπορεί να χρησιμοποιηθεί για παράδειγμα για ένα εταιρικό τηλεφωνικό κέντρο Asterisk αλλά είναι αρκετή για τον σκοπό της παρούσης εργασίας. Στο VMware ορίσαμε το σύστημα μας να παίρνει ξεχωριστή IP από το subnet της κάρτας δικτύου και επίσης ορίσαμε στο Asterisk να έχει static IP (την 192.168.1.66) αλλάζοντας το περιεχόμενο του αρχείου ifcfg-eth0 (το οποίο βρίσκεται στο /etc/sysconfig/network-scripts/) με τα παρακάτω:

```
DEVICE=eth0
BOOTPROTO=none
IPADDR=192.168.1.66
NETMASK=255.255.255.0
GATEWAY=192.168.1.254
HWADDR= 00-0C-29-96-63-B3
ONBOOT=yes
TYPE=Ethernet
USERCTL=no
```

Για να εισέλθουμε στο γραφικό περιβάλλον διαχείρισης αρκεί να πληκτρολογήσουμε <http://192.168.1.66/> σε ένα Browser και δίνοντας το username και το password που έχουμε ορίσει κατά την εγκατάσταση μπαίνουμε στην βασική σελίδα (Σχήμα 4.1). Στην έκδοση του Asterisknow που χρησιμοποιήθηκε για αυτήν την εργασία χρησιμοποιείται το γραφικό περιβάλλον Asterisk GUI, ενώ από το Asterisknow 1.5 χρησιμοποιείται το freePBX.



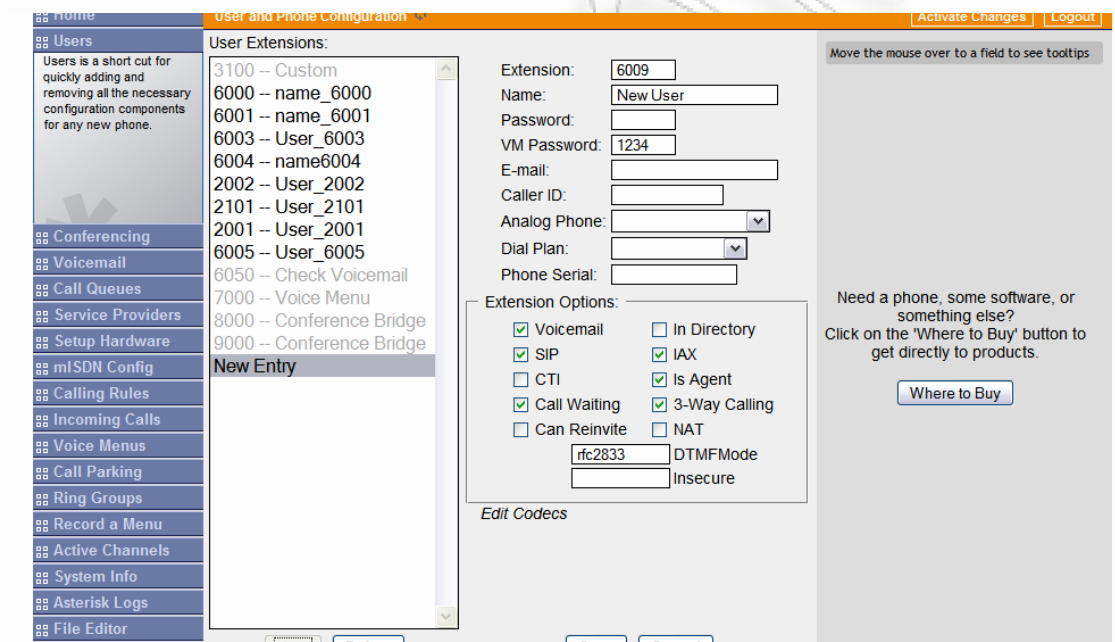
Σχήμα 4.1: Γραφικό περιβάλλον του Asterisknow

## 4.2 Διαμόρφωση του Asterisk

Για την διαμόρφωση του Asterisk μπορεί να χρησιμοποιηθεί είτε το γραφικό περιβάλλον απευθείας (το οποίο είναι και το πιο απλό), είτε να αλλάξουμε χειροκίνητα τα configuration files. Όπως και σε αρκετά προγράμματα, έτσι και το Asterisk διαμορφώνεται μέσω πολλών αρχείων (.conf) που συνεργάζονται εσωτερικά μεταξύ τους. Τα .conf αρχεία υπάρχουν στο φάκελο /etc/asterisk. Όλες οι λειτουργίες του Asterisk καθορίζονται μέσα σε αυτά τα αρχεία με μοναδική εξαίρεση να αποτελεί το αρχείο zaptel.conf που βρίσκεται στο φάκελο /etc και παρέχει ρυθμίσεις για το υλικό Zaptel. Παρακάτω θα περιγράψουμε και τις δύο διαδικασίες διότι για να κατανοήσουμε την λειτουργία του Asterisk πρέπει να κατανοήσουμε τον τρόπο συγγραφής και διασύνδεσης των configuration files. Επιπλέον για ορισμένες από τις εφαρμογές που περιγράφονται στο επόμενο κεφάλαιο είναι απαραίτητη η χειροκίνητη τροποποίηση των configuration files.

### 4.2.1 Δημιουργία χρηστών

Η πρώτη ενέργεια που πρέπει να κάνουμε κατά την διαμόρφωση ενός συστήματος Asterisk είναι να δημιουργήσουμε τους χρήστες που θέλουμε να έχουμε. Κατά την δημιουργία των χρηστών ορίζουμε το νούμερό τους, τον κωδικό που πρέπει να χρησιμοποιήσουν για να εγγραφούν στο σύστημα (register) καθώς και τις διάφορες υπηρεσίες που επιθυμούμε να είναι διαθέσιμες σε αυτούς. Για την δημιουργία ενός χρήστη πάμε στο πεδίο Users και επιλέγουμε την δημιουργία ενός νέου (Σχήμα 4.2). Ο κάθε χρήστης που δημιουργείται αποθηκεύεται στο αρχείο users.conf.



Σχήμα 4.1: Δημιουργία νέου χρήστη

Όπως φαίνεται στο παραπάνω σχήμα μπορούμε να δημιουργήσουμε έναν χρήστη με όποιον αριθμό και όνομα θέλουμε. Δίνουμε επίσης κάποια στοιχεία σε αυτόν τον χρήστη όπως το συνθηματικό του, τον κωδικό του για το Voicemail, το ID του, καθώς επίσης και σε πιο Dialplan θέλουμε να ανήκει. Επίσης του ορίζουμε κάποιες επιπλέον επιλογές όπως το πιο πρωτόκολλο θα χρησιμοποιεί (sip, iax), αν θα έχει voicemail, αν θα έχει διαθέσιμη την υπηρεσία αναμονής κλήσης κτλ. Από την δημιουργία του χρήστη μέσω του γραφικού περιβάλλοντος και μετά ο χρήστης μπορεί να κάνει κλήσεις και να χρησιμοποιήσει τις

υπηρεσίες που έχει διαθέσιμες, αφού το γραφικό περιβάλλον δημιουργεί αυτομάτως τις εγγραφές στα αντίστοιχα αρχεία (`extensions.conf`, `users.conf`, `voicemail.conf` κτλ). Στην συνέχεια θα αναλύσουμε εκτενέστερα αυτές τις επιλογές δημιουργώντας χειροκίνητα έναν χρήστη.

Αντίστοιχα μπορούμε να δημιουργήσουμε έναν νέο χρήστη προσθέτοντας ένα `context` στο αρχείο `users.conf`:

```
[2001]
cid_number=2001
context=numberplan-custom-2
fullname=User_2001
host=dynamic
nat=no
secret=123456
mailbox=2001
vmsecret=1234
dtmfmode=rfc2833
disallow=all
allow=alaw
canreinvite=yes
signalling=fxo_ks
threewaycalling=yes
callwaiting=yes
hasagent=yes
hasdirectory=no
hasiax=yes
hasmanager=no
hassip=yes
hasvoicemail=yes
deletevoicemail=no
registeriax=yes
registersip=yes
```

*autoprov=no*

Το περιεχόμενο (context) ενός χρήστη ορίζεται από το όνομα που του δίνουμε τοποθετημένο μέσα σε τετραγωνικά άγκιστρα []. Ένα context βρίσκεται σε ισχύ μέχρι να δηλωθεί το επόμενο μέσα στο αρχείο sip.conf. Με τον παραπάνω τρόπο δημιουργούμε τον χρήστη 2001 στο οποίο ορίσαμε και ορισμένα χαρακτηριστικά τα βασικότερα των οποίων περιγράφονται παρακάτω. Με τον ίδιο τρόπο μπορούμε να δημιουργήσουμε και τους υπόλοιπους χρήστες.

Με τις τρεις πρώτες εγγραφές ορίζουμε το νούμερο (*cid\_number*), το dial plan (*context*) καθώς και το όνομα του χρήστη (*fullname*). Με την εγγραφή *host=dynamic* λέμε στον Asterisk πως τα τερματικά που χρησιμοποιούμε δεν έχουν σταθερή IP τιμή και έτσι θα πρέπει κάθε φορά που εγγραφόμαστε σε αυτόν να την αναζητά. Εναλλακτικά θα μπορούσαμε να ορίσουμε για τον χρήστη μία στατική IP διεύθυνση, για παράδειγμα την 192.187.10.77 ως εξής *host=192.187.10.77*. Τέλος θα μπορούσαμε να δηλώσουμε και κάποιο domain name. Με την εγγραφή *nat* καθορίζουμε αν ο χρήστης είναι πίσω από κάποια NAT (Network Address Translation) συσκευή όπως firewall ή router. Η επιλογή αυτή εφόσον οριστεί *yes* λέει στον Asterisk να αγνοήσει τις πληροφορίες που δέχεται μέσα από το κανάλι της κλήσης και να χρησιμοποιήσει για επικοινωνία την διεύθυνση από την οποία δέχεται τις πληροφορίες. Η τιμή της εγγραφής *secret* είναι το συνθηματικό για την αυθεντικοποίηση του χρήστη. Δηλαδή για την εγγραφή του χρήστη στο asterisk θα του ζητηθεί από τον server αυτός ο κωδικός. Στην συνέχεια με τις εγγραφές *mailbox* και *vmsecret* ορίζουμε το username και το password του voicemail του εκάστοτε χρήστη. Με αυτά τα δεδομένα ο χρήστης αφού καλέσει τον αριθμό του voicemail και κάνει login με αυτά, μπορεί να ακούσει τα φωνητικά του μηνύματα. Στην συνέχεια με την εγγραφή *dtmfmode* ορίζουμε τον τύπο των dtmf τόνων που χρησιμοποιεί ο χρήστης. Η εγγραφή *canreinvite=yes* λέει στο Asterisk να επιτρέπει στους χρήστες να συνδέονται απευθείας μεταξύ τους έτσι ώστε από την εγκαθίδρυση της κλήσης και μετά τα RTP πακέτα φωνής να ανταλλάσσονται απευθείας μεταξύ των 2 τηλεφώνων. Οι εγγραφές *disallow*, *allow* αναφέρονται στους codecs τους οποίους υποστηρίζει ο χρήστης. Εδώ με αυτές τις 2

εγγραφές επιτρέπουμε την χρήση μόνο του G711(alaw). Τέλος με τις επόμενες εγγραφές καθορίζονται οι υπηρεσίες στις οποίες έχει δικαίωμα πρόσβασης ο χρήστης. Για παράδειγμα ο παραπάνω χρήστης μπορεί να χρησιμοποιήσει την υπηρεσία ταυτόχρονης κλήσεις με 3 συνδρομητές (three way call) και αναμονής κλήσης (call waiting).

#### 4.2.2 To Dialplan του Asterisk

Αφού έχουμε δημιουργήσει κάποιους χρήστες πρέπει στην συνέχεια να διαμορφώσουμε κάποια αρχεία ώστε να μπορούν αυτοί οι χρήστες να καλούν και να δέχονται κλήσεις. Την βασική ευθύνη της δρομολόγησης κλήσεων την έχει το αρχείο extensions.conf. Στην περίπτωση που η διαμόρφωση του Asterisk γίνει από το γραφικό περιβάλλον τότε μετά την δημιουργία των χρηστών είμαστε έτοιμοι να κάνουμε κλήσεις μεταξύ τους αφού δημιουργούνται αυτόματα κάποια macro στο αρχείο extensions.conf, τα οποία μας επιτρέπουν να κάνουμε κλήσεις μεταξύ των χρηστών που έχουμε δημιουργήσει. Παρακάτω θα παρουσιάσουμε την δομή του αρχείου extensions.conf και θα δείξουμε παραδείγματα χειροκίνητης διαμόρφωσής του, μιας και είναι ένα από τα σημαντικότερα αρχεία για την λειτουργία του Asterisk.

Η δομή του Dialplan αποτελείται από τέσσερα κύρια μέρη : τα contexts, τα extensions, τις προτεραιότητες και από τις εφαρμογές [22].

- Contexts

Τα Dialplans χωρίζονται εσωτερικά σε τομείς που ονομάζονται contexts. Ένα context ξεκινάει με το όνομά του μέσα σε τετραγωνικά άγκιστρα [ ] και τελειώνει όταν ξεκινήσει το επόμενο. Το context δεν είναι τίποτα άλλο παρά μία ομάδα από extensions. Δηλαδή, μέσω των context, μπορούμε να χωρίσουμε το πλήθος των extension που έχουμε στο σύστημά μας και να τις χειριστούμε εντελώς ανεξάρτητα. Τα extensions που βρίσκονται μέσα σε ένα context μπορούν να αλληλεπιδράσουν μόνο με αυτά που υπάγονται στο ίδιο context εκτός αν έχει δηλωθεί διαφορετικά με τη μέθοδο include. Άρα τα contexts μας επιτρέπουν να κρύψουμε ή να κάνουμε απροσπέλαστες συγκεκριμένες extensions από άλλα extensions.

Τα τρία βασικά context που πρέπει να ορίσουμε στην περίπτωσή μας είναι το [general], το [globals] καθώς και το [numberplan-custom-2] στο οποίο έχουμε δηλώσει ότι ανήκουν οι χρήστες μας.

```
[general]
static=yes
writeprotect=no
autofallthrough=yes
clearglobalvars=no
priorityjumping=yes
```

Στο context [general] ορίζονται κάποιες ιδιότητες του extensions.conf και γενικές συμπεριφορές του αρχείου. Για παράδειγμα μπορούμε να επιλέξουμε αν θέλουμε να σώζουμε το αρχείο από την κονσόλα του Asterisk με την εντολή save dialplan, θέτοντας τις επιλογές static=yes και writeprotect=no. Με την επιλογή autofallthrough μπορούμε να ορίσουμε το Asterisk να κλείνει το κανάλι της κλήσης αν του τελειώσουν οι διαθέσιμες επιλογές που του έχουμε ορίσει μέσω των extensions. Με την επιλογή clearglobalvars το Asterisk με κάθε επανεκκίνηση ή επαναφόρτωση των αρχείων με την εντολή reload, θα διαβάζει εκ νέου τις μεταβλητές που βρίσκονται στο [globals], σε κάθε άλλη περίπτωση ότι υπάρχει στο [globals] θα συνεχίζει να υπάρχει ακόμα και αν το διαγράψουμε από το extensions.conf. Τέλος με την επιλογή priorityjumping=yes, οι εφαρμογές που υποστηρίζουν άλματα προτεραιότητας ανάλογα με το αποτέλεσμα της λειτουργίας τους (Dial()) μπορούν να τα πράξουν.

Στην συνέχεια φτιάχνουμε ένα context με το όνομα του context στο οποίο έχουμε ορίσει ότι ανήκουν οι χρήστες μας (numberplan-custom-2).

```
[numberplan-custom-2]
include=default
exten=_2XXX,1,Dial(SIP/${EXTEN:0},20,tr)
```



Βλέπουμε ότι ξεκινάμε με το να συμπεριλάβουμε το context default μέσα στο numberplan-custom-2. Το context default όπως φαίνεται και παρακάτω περιλαμβάνει κάποια extensions για υπηρεσίες όπως το voicemail και το meetme conference. Παρακάτω θα εξηγήσουμε και την λειτουργία αυτών των extension.

```
[default]
```

```
exten=6050,1,VoiceMailMain
```

```
exten=8000,1,MeetMe(${EXTEN}|MI)
```

```
exten=9000,1,MeetMe(${EXTEN}|MIwA)
```

Στο context [globals] ορίζονται οι μεταβλητές οι οποίες ισχύουν σε όλο το dialplan για όλες τις extensions ανεξαρτήτου context που περιέχονται, και διευκολύνουν πολύ στην αναγνωσιμότητα του dialplan και στην αποφυγή τυπογραφικών λαθών.

Η σύνταξη μία global μεταβλητής είναι:

```
[globals]
```

```
ONOMA_METABΛΗΤΗΣ=ΤΙΜΗ
```

Για να χρησιμοποιήσουμε την τιμή της μεταβλητής μέσα στο dialplan θα πρέπει να την καλέσουμε ως `${ONOMA_METABΛΗΤΗΣ}`. Εναλλακτικά μπορούμε να δηλώσουμε μια global μεταβλητή, δυναμικά, σε κάποιο άλλο σημείο του dialplan με την εφαρμογή `SetGlobalVar(ONOMA_METABΛΗΤΗΣ=ΤΙΜΗ)`.

- Extensions

Όπως ήδη αναφέρθηκε, μέσα σε κάθε context περιέχεται ένα ή περισσότερα extensions. Ένα extension, είναι μία οδηγία που θα ακολουθήσει το Asterisk αν δεχθεί μία κλήση ή αν κάποιος χρήστης πληκτρολογήσει κάποια ψηφία. Τα extensions λοιπόν, ορίζουν το πώς η κλήση θα δρομολογηθεί μέσα από μία σειρά αποφάσεων και παραδοχών.

Η παραδοσιακή έννοια του όρου extension αναφέρεται στο νούμερο κάποιας συσκευής που συνδέεται στο PBX. Στο Asterisk όμως σημαίνει πολλά περισσότερα. Ένα extension μπορεί για παράδειγμα να είναι μόνο μία γραμμή που να συσχετίζει το extension με ένα συγκεκριμένο χρήστη:

```
exten=2001,1,Dial(SIP/2001)
```

Επίσης μπορεί να αφορά μια ομάδα χρηστών όπου για παράδειγμα το νούμερό τους ξεκινάει με τον αριθμό 2 και περιέχει τρία επιπλέον ψηφία. Για να το επιτύχουμε αυτό χρησιμοποιούμε κάποια wilcards. Τα διαθέσιμα wilcards του asterisk είναι τα εξής:

- [0126-9]: Όλοι οι αριθμοί που περιγράφονται μέσα σε αυτό το μοτίβο (στο συγκεκριμένο παράδειγμα 0,1,2,6,7,8,9).
- X: Οποιοσδήποτε αριθμός μεταξύ 0 και 9.
- Z: Οποιοσδήποτε αριθμός μεταξύ 1 και 9.
- N: Οποιοσδήποτε αριθμός μεταξύ 2 και 9.
- . : Οποιοσδήποτε αριθμός μία ή περισσότερες φορές

Επίσης στην περίπτωση αυτή επειδή δεν είναι γνωστό εκ των προτέρων το νούμερο στο οποίο το asterisk πρέπει να δρομολογήσει την κλήση, χρησιμοποιούμε την μεταβλητή  $\{EXTEN:0\}$  η οποία ισούται με το αριθμό χρήστη που έχουμε πληκτρολογήσει κατά την κλήση η οποία δρομολογείτε από το συγκεκριμένο extension. Το :0 σημαίνει ότι πρέπει να χρησιμοποιήσουμε ολόκληρο αυτόν τον αριθμό, ενώ αν για παράδειγμα θέλαμε να κόψουμε τα πρώτα 2 ψηφία θα χρησιμοποιούσαμε :2.

```
exten=_2XXX,3,Dial(SIP/${EXTEN:0})
```

Στο Asterisk όμως εκτός από το νούμερο ενός χρήστη, ένα extension μπορεί να συσχετιστεί με τον αυτόματο τηλεφωνητή, με μία ουρά αναμονής ή με

οποιαδήποτε υπηρεσία παρέχεται από το asterisk. Ένα extension γενικά ορίζεται με τη λέξη exten ακολουθούμενη από το σύμβολο =. Ο γενικός τύπος ορισμού μίας extension είναι:

exten = νούμερο (ή όνομα), προτεραιότητα, εφαρμογή.

Κάθε γραμμή ξεκινάει με την εντολή exten = και στη συνέχεια ακολουθεί το νούμερο (ή το όνομα) του extension. Το νούμερο αυτό δείχνει στο Asterisk ποια σειρά εντολών θα πρέπει να τρέξει. Αυτό το νούμερο μπορεί να ανιχνευτεί μέσω τριών βασικών τρόπων:

1. Να αποστέλλεται μαζί με την κλήση από την τηλεφωνική εταιρία. Η μέθοδος αυτή ονομάζεται Direct Inward Dialing (DID).
2. Οι χρήστες να τον πληκτρολογούν στην τερματική τους συσκευή.
3. Μέσω κάποιων ειδικών extensions που ορίζονται από τον Asterisk.

Τα πιο συνηθισμένα ειδικά extensions που μπορεί να συναντηθούν είναι αυτά που ακολουθούν:

- s (start): Χρησιμοποιείται όταν δεν είναι δυνατόν, να γνωρίζει το Asterisk με ποιο ακριβώς extension επιθυμεί να συνδεθεί ο καλών. Αυτή η περίπτωση συναντάτε όταν η κλήση προέρχεται απ' το δημόσιο τηλεφωνικό δίκτυο και δεν έχει συμφωνηθεί κάποιο DID νούμερο με τον πάροχο τηλεφωνίας.
- t (timeout): Αν απαιτείται από το χρήστη να εισάγει δεδομένα όπως συμβαίνει σ' ένα IVR σύστημα και ο χρήστης καθυστερήσει, τότε θα εκτελεστεί το ειδικό extension t.
- i (invalid): Αν απαιτείται από το χρήστη να εισάγει δεδομένα και τα δεδομένα που εισήγαγε είναι μη αποδεκτά, τότε θα εκτελεστεί το ειδικό extension i.
- fax: Αν το Asterisk αντιληφθεί ότι η εισερχόμενη κλήση είναι μήνυμα fax, τότε θα εκτελεστεί το ειδικό extension fax.

- Προτεραιότητες

Όπως είδαμε και παραπάνω μετά το όνομα, ακολουθεί η προτεραιότητα. Ο Asterisk ξεκινάει πάντα με την προτεραιότητα 1, εκτελεί την εφαρμογή και προχωράει στην επόμενη προτεραιότητα, 2 ή γενικότερα  $x+1$ . Κάποιες εφαρμογές όπως η Dial(), μπορούν να αναγκάσουν το Asterisk να κάνει άλμα προτεραιοτήτων και να πάει κατευθείαν στην προτεραιότητα  $x+101$  (όπου  $x$  ο αριθμός της προτεραιότητας της εφαρμογής που προκάλεσε το άλμα), δυνατότητα η οποία μας επιτρέπει να δρομολογήσουμε τις κλήσεις βασιζόμενοι σε αποφάσεις, όπως για παράδειγμα, αν ο προορισμός είναι κατειλημμένος. Εκτός από τον αριθμητικό συμβολισμό των προτεραιοτήτων, οι δημιουργοί του Asterisk, γνωρίζοντας το μέγεθος που μπορεί να αποκτήσει ένα σύνθετο dialplan και τις επιπτώσεις που θα είχε έστω και η παραμικρή αλλαγή σε αυτό, δημιούργησαν την ειδική προτεραιότητα  $n$ . Η προτεραιότητα αυτή, παίρνει αμέσως την επόμενη τιμή από την προηγούμενη της. Έτσι στην περίπτωση ύπαρξης μεγάλων αρχείων, αν διαπιστώσουμε πως απαιτείται να γίνουν αλλαγές στον τρόπο εκτέλεσης των εφαρμογών, αυτό τώρα μπορεί να γίνει εύκολα και ανώδυνα.

- Εφαρμογές

Τέλος έχουμε τις εφαρμογές. Μέσω των εφαρμογών υλοποιούνται όλες οι βασικές λειτουργίες του Asterisk. Οι εφαρμογές εκτελούν συγκεκριμένες λειτουργίες σε μία κλήση, όπως αναπαραγωγή ήχου, επεξεργασία τονικών ήχων εισόδου, ή ακόμα και τερματισμό της κλήσης.

Στην συνέχεια παραθέτουμε τις βασικότερες εφαρμογές που μπορεί να εκτελέσει ο Asterisk:

- Answer(): Απαντάει τη γραμμή. Πολλές εφαρμογές απαιτούν η γραμμή να έχει απαντηθεί πρώτα από το Asterisk προτού μπορέσουν να εκτελεστούν (Playback, BackGround, κ.λ.π.).
- Hangup(): Εκτελεί την ακριβώς αντίθετη λειτουργία από την Answer. Κλείνει δηλαδή το κανάλι επικοινωνίας που δημιουργήθηκε από κάποια άλλη εφαρμογή.

- Playback(αρχείο ήχου): Αυτή η εφαρμογή αναπαραγάγει ένα αρχείο ήχου .wav ή .gsm
- Background (αρχείο ήχου): Λειτουργεί παρόμοια με το Playback με τη διαφορά ότι ακούει για τονικούς ήχους.
- Queue(όνομα ουράς | επιλογές): Η εφαρμογή Queue τοποθετεί την κλήση στην ουρά αναμονής με το όνομα και τις επιλογές που δηλώθηκαν κατά την εκτέλεση της. Θα πρέπει ήδη να υπάρχει δηλωμένη αντίστοιχη ουρά στο αρχείο queues.conf.
- Wait (δευτερόλεπτα): Προκαλεί παύση για καθορισμένο χρονικό διάστημα προκειμένου να εκτελεστούν χρονοβόρες διαδικασίες όπως εγγραφή και ανάγνωση από αρχείο.
- NoOp (αλφαριθμητικό): Η εφαρμογή αυτή δεν εκτελεί κάποια συγκεκριμένη λειτουργία, αλλά τυπώνει στην κονσόλα του Asterisk το αλφαριθμητικό που έλαβε ως όρισμα. Χρησιμεύει όταν ψάχνουμε για σφάλματα στο dialplan.
- Voicemail (extension): Στέλνει την κλήση στον αυτόματο τηλεφωνητή του extension που δόθηκε σαν όρισμα κατά την εκτέλεσή της. Η υλοποίηση της εφαρμογής Voicemail αναπαραγάγει κάποια ηχογραφημένα μηνύματα ανάλογα με το γράμμα που έχουμε ορίσει πριν την δήλωση του extension, όπως φαίνεται παρακάτω. Επίσης για την λειτουργία του voicemail είναι απαραίτητη η δημιουργία του αντίστοιχου λογαριασμού στο αρχείο voicemail.conf
  - Με το γράμμα u πριν από το extension αναπαραγάγετε ο χαιρετισμός «Μη διαθέσιμος».
  - Με το γράμμα b πριν από το extension αναπαραγάγετε ο χαιρετισμός «Κατελιημμένος».
  - Με το γράμμα s πριν από το extension δεν αναπαραγάγετε κανένα χαιρετισμό.
- VoicemailMain(): Αυτή η εφαρμογή, επιτρέπει στους χρήστες να ακούσουν τα μηνύματα τους, να ηχογραφήσουν προσαρμοσμένους χαιρετισμούς και να αλλάξουν κάποιες ρυθμίσεις του λογαριασμού τους. Μπορεί να πάρει σαν όρισμα τη θυρίδα που επιθυμούμε να συνδεθεί κάποιος χρήστης απευθείας,

χωρίς την ανάγκη αυθεντικοποίησης του που είναι απαραίτητη με την χρήση της Voicemail (extension).

- Dial (τύπος/id, διάρκεια παύσης, επιλογές): Η εφαρμογή Dial() λέει στο Asterisk να καλέσει μέσω της τεχνολογίας που ορίζεται από την παράμετρο «τύπος» τον αριθμό «id» και όταν η γραμμή απαντηθεί να ενώσει τα δύο άκρα της επικοινωνίας. Οι επιλογές που μπορούμε να ορίσουμε στην Dial είναι:
  - t: επιτρέπει στον καλούμενο τη μεταφορά της κλήσης πατώντας το πλήκτρο # ή όποιον άλλον συνδυασμό πλήκτρων έχει οριστεί στο αρχείο features.conf.
  - r: μιμείται τον ήχο κλήσης στο ακουστικό του καλούντα.
  - m: παρέχει μουσική κατά τη διάρκεια της αναμονής.
  - g: Αν ο προορισμός κλείσει τη γραμμή τότε συνεχίζει στην επόμενη προτεραιότητα του extension.
  - w: Επιτρέπει την ηχογράφηση της κλήσης από τον καλούμενο με το πάτημα της αλληλουχίας των πλήκτρων που ενεργοποιούν το automon όπως αυτά έχουν οριστεί στο αρχείο features.conf.

Όπως αναφέραμε και προηγουμένως όταν χρησιμοποιούμε το Asterisknow και το αντίστοιχο γραφικό περιβάλλον Asterisk GUI τότε δεν χρειάζεται να κάνουμε κάποια επιπλέον διαμόρφωση στο Asterisk πέρα από την δημιουργία των χρηστών με τις επιλογές τις οποίες θέλουμε για τον κάθε ένα από αυτούς. Στην συνέχεια παρουσιάζουμε το dialplan που θα πρέπει να χρησιμοποιηθεί μετά την χειροκίνητη δημιουργία χρηστών όπως την περιγράψαμε παραπάνω. Το αρχείο extensions.conf θα περιέχει τα εξής:

```
[general]
static=yes
writeprotect=no
autofallthrough=yes
clearglobalvars=no
```

```
priorityjumping=yes
```

```
[default]
```

```
exten=6050,1,VoiceMailMain
```

```
exten=8000,1,MeetMe(${EXTEN}|MI)
```

```
exten=9000,1,MeetMe(${EXTEN}|MIwA)
```

```
[numberplan-custom-2]
```

```
include=default
```

```
exten=_2XXX,1,Noop()
```

```
exten=_2XXX,2,Dial(SIP/${EXTEN:0},10,tr)
```

```
exten=_2XXX,n,Goto(${EXTEN}-${DIALSTATUS},1)
```

```
exten=2001-NOANSWER,1,Voicemail(u2001)
```

```
exten=2001-BUSY,1,Voicemail(b2001)
```

```
exten=2001-.,1,Goto(2001-NOANSWER,1)
```

```
exten=2002-NOANSWER,1,Voicemail(u2002)
```

```
exten=2002-BUSY,1,Voicemail(b2002)
```

```
exten=2002-.,1,Goto(2002-NOANSWER,1)
```

```
exten=2003-NOANSWER,1,Voicemail(u2003)
```

```
exten=2003-BUSY,1,Voicemail(b2003)
```

```
exten=2003-.,1,Goto(2003-NOANSWER,1)
```

Βλέπουμε ότι στο context numberplan-custom-2 έχουμε όλη την δρομολόγηση των κλήσεων για τους χρήστες 2001, 2002, 2003. Αναλυτικότερα με την:

```
exten=_2XXX,1,Noop()
```

έχουμε ορίσει στο asterisk να μην κάνει καμία ενέργεια.

Στην συνέχεια με την επόμενη γραμμή λέμε στο asterisk ότι για όλα τα νούμερα που ξεκινάνε από 2 και ακολουθούνται από 3 οποιαδήποτε αριθμητικά ψηφία (2XXX) να καλέσει αυτόν τον χρήστη χρησιμοποιώντας το πρωτόκολλο sip (Dial(SIP/\${EXTEN:0},10,tr)) και να μην αφαιρέσει κανένα ψηφίο από αυτόν τον αριθμό (EXTEN:0). Η συσκευή του χρήστη του οποίου έχουμε πληκτρολογήσει το νούμερο (για παράδειγμα 2002) θα χτυπάει για 10 δευτερόλεπτα όταν δέχεται κλήση. Ο

καλών θα ακούει στο ακουστικό του έναν ήχο που μιμείται τον κλασσικό ήχο κλήσης που υπάρχει στις συμβατικές συσκευές. Τέλος η μεταβλητή t, επιτρέπει την μεταφορά της κλήσης.

```
exten=_2XXX,2,Dial(SIP/${EXTEN:0},10,tr)
```

Στην συνέχεια με την:

```
exten=_2XXX,n,Goto(${EXTEN}-${DIALSTATUS},1)
```

στέλνουμε την κλήση σε κάποιο extension ανάλογα με την κατάσταση του καλούντα (για παράδειγμα του 2002), ο οποίος μπορεί να μην απαντήσει ή να είναι απασχολημένος.

Ανάλογα με την ενέργεια του καλούντα (εδώ ο 2002) έχουμε ορίσει τα δύο παρακάτω extension για κάθε πιθανό σενάριο.

- exten=2002-NOANSWER,1,Voicemail(u2002)

Αν ο χρήστης 2002 δεν απαντήσει σε 20 δευτερόλεπτα, τότε ο Asterisk θα προχωρήσει στην αναπαραγωγή του μηνύματος The person at extension 2002 is unavailable please leave your message after the tone, ενημερώνοντας τον καλών πως ο χρήστης 2002 πιθανότατα απουσιάζει και στην συνέχεια θα τον οδηγήσει στο φωνητικό ταχυδρομείο του 2002.

- exten=2002-BUSY,1,Voicemail(b2002)

Αν ο χρήστης 2002 είναι κατειλημμένος, ο Asterisk θα προχωρήσει στην αναπαραγωγή του μηνύματος The person at extension 2002 is on the phone please leave your message after the tone, ενημερώνοντας τον καλών πως ο χρήστης 2002 είναι απασχολημένος και στην συνέχεια θα τον οδηγήσει στο φωνητικό ταχυδρομείο του 2002.

Τέλος με το παρακάτω extension λέμε στο asterisk ότι για οποιαδήποτε μη αναμενόμενη κατάσταση του καλούντος (Dialstatus) να θέτει το Dialstatus σαν



NOANSWER και να το στέλνει στο αντίστοιχο extension που είδαμε παραπάνω και το οποίο οδηγεί την κλήση στο voicemail:

```
exten=2002-.,1,Goto(2002-NOANSWER,1)
```

Για την λειτουργία του παραπάνω dialplan θα πρέπει να έχουμε προσθέσει και κάποια context στο αρχείο voicemail.conf, που να καθορίζουν την λειτουργία του φωνητικού ταχυδρομείου για τους χρήστες 2001, 2002 και 2003. Έτσι έχουμε τα 3 context που φαίνονται παρακάτω:

```
[general]
format=wav49|gsm|wav
serveremail=asterisk
attach=yes
skipms=3000
maxsilence=10
silencethreshold=128
maxlogins=3
emaildateformat=%A, %B %d, %Y at %r
sendvoicemail=yes

[zonemessages]
Greece=Europe/Athens|'vm-received' Q 'digits/at' R

[default]
2001=1234,Sip User,,,tz=Greece
2002=1234,SIP User,,,tz=Greece
2003=1234,SIP User,,,tz=Greece
```

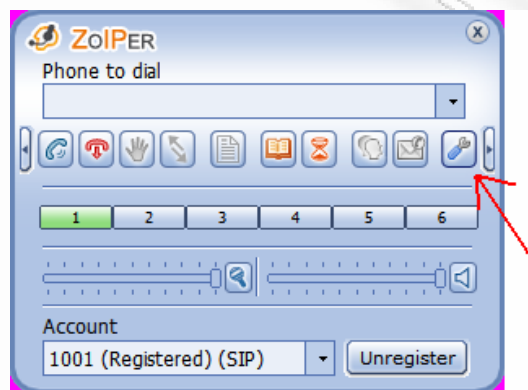
#### 4.2.3 Τα αρχεία sip.conf και iax.conf

Τα δύο αρχεία sip.conf και iax.conf χρησιμοποιήθηκαν για να δημιουργήσουμε ένα sip trunk και ένα iax trunk αντίστοιχα. Συγκεκριμένα όπως θα δούμε και στο επόμενο κεφάλαιο δημιουργήσαμε ένα sip trunk με τον πάροχο υπηρεσιών VoIP Voipdiscount,

μέσω του οποίου μπορούμε να δρομολογήσουμε κλήσεις προς το pstn δίκτυο. Επίσης δημιουργήσαμε ένα iax trunk για να συνδέσουμε δύο κέντρα asterisk μεταξύ τους. Τα δύο αυτά αρχεία θα τα δούμε αναλυτικότερα στο επόμενο κεφάλαιο.

#### 4.2.4 Διαμόρφωση των softphone

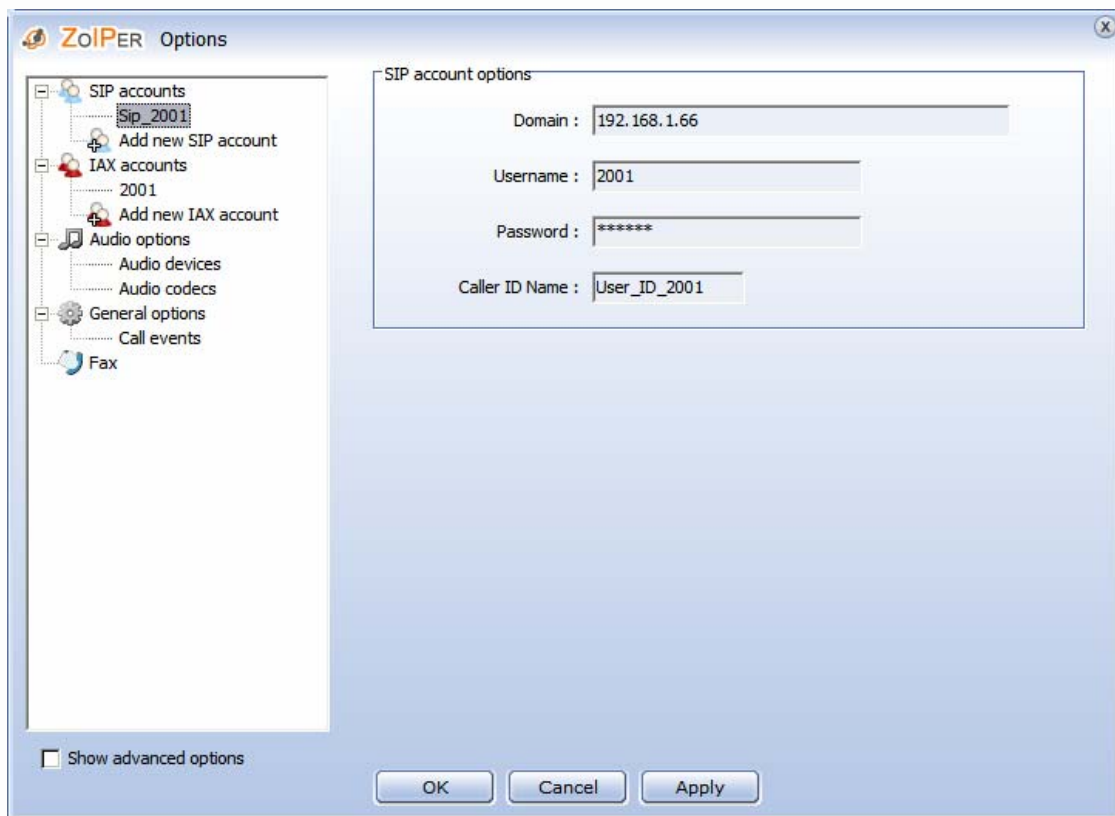
Στα πλαίσια αυτής της εργασίας χρησιμοποιήθηκε η δωρεάν έκδοση του zoiper softphone [23] το οποίο υποστηρίζει sip και iax χρήστες. Για την διαμόρφωση του zoiper επιλέγουμε το σχετικό εικονίδιο των ρυθμίσεων (Σχήμα 4.2)



Σχήμα 4.2: Το softphone Zoiper

Στην συνέχεια επιλέγουμε το πεδίο sip accounts και δημιουργούμε ένα καινούργιο account με τις εξής παραμέτρους (Σχήμα 4.3):

- Domain: IP ή host Name του Asterisk (εδώ 192.168.1.66)
- Username: Το username του χρήστη μας (εδώ 2001)
- Password: Τον κωδικό που έχουμε δώσει για την αυθεντικοποίηση του συγκεκριμένου χρήστη προς το Asterisk
- Caller ID Name : Δίνουμε ένα όνομα που θέλουμε στον χρήστη μας, το οποίο θα χαρακτηρίζει το συγκεκριμένο softphone.



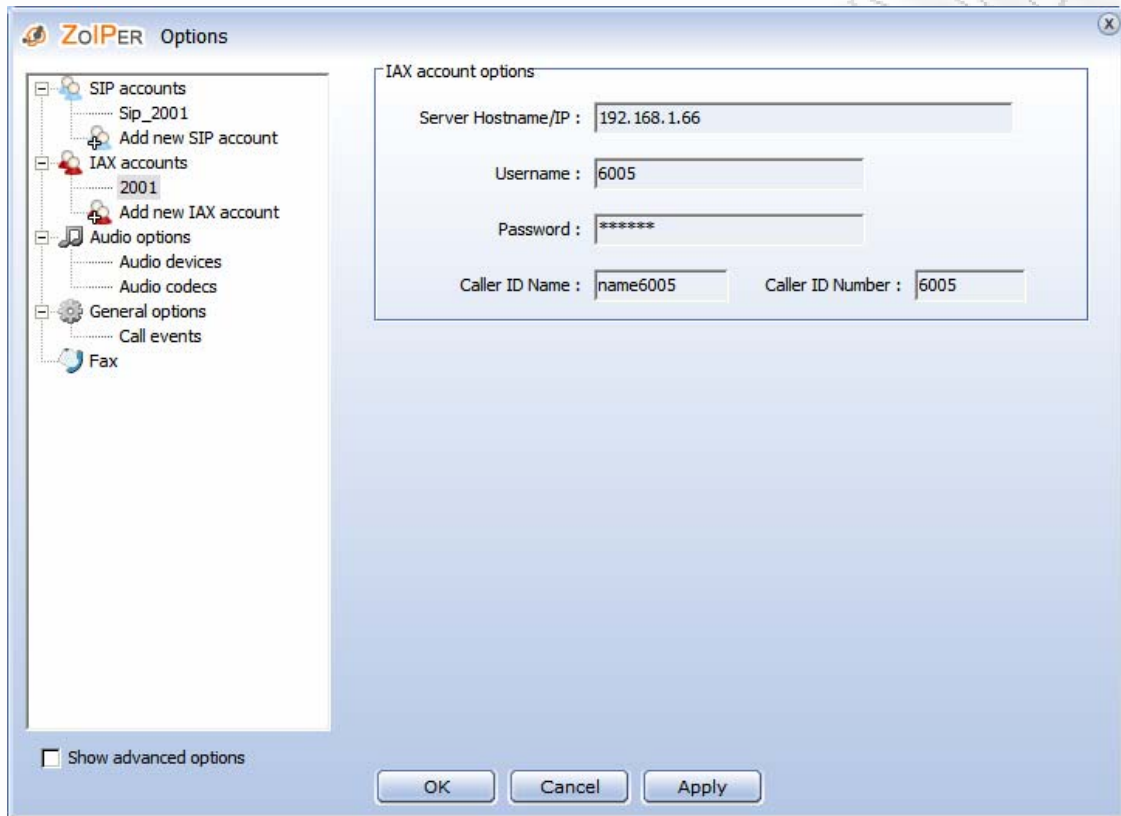
Σχήμα 4.3: Διαμόρφωση ενός sip χρήστη στο zoiper

Τέλος πατώντας το κουμπί register (Σχήμα 4.4) εγγράφουμε τον χρήστη στον server και πλέον είμαστε έτοιμοι να πραγματοποιήσουμε κλήσεις.



Σχήμα 4.4: Εγγραφή ενός sip χρήστη στον Server

Με παρόμοιο τρόπο μπορούμε να δημιουργήσουμε και έναν χρήστη iax στο zoiper όπως φαίνεται στο παρακάτω σχήμα (Σχήμα 4.5):

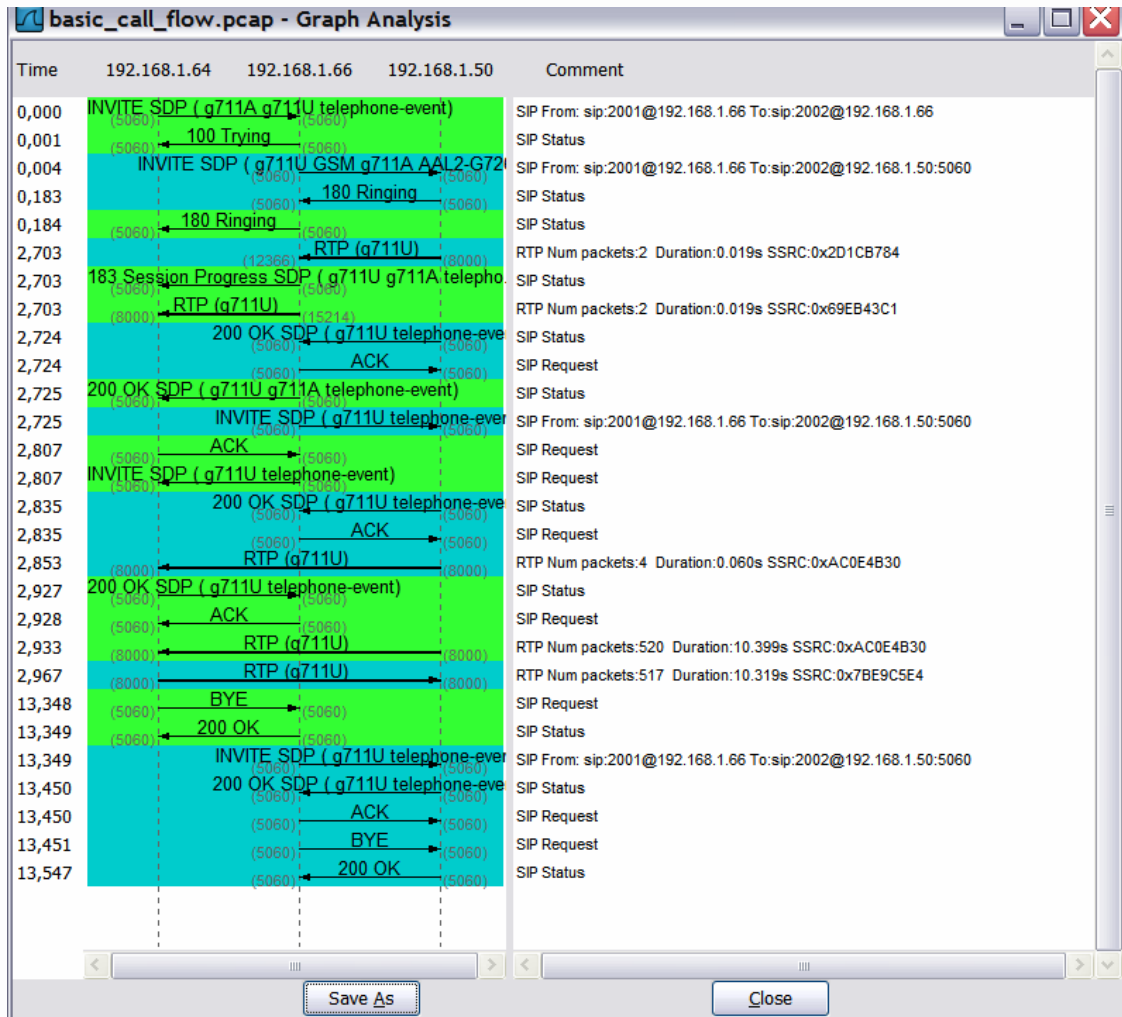


Σχήμα 4.5: Διαμόρφωση ενός iax χρήστη στο zoiper

### 4.3 Δοκιμή του Asterisk

Για να επιβεβαιώσουμε την σωστή λειτουργία των configuration files που διαμορφώσαμε παραπάνω θα κάνουμε κάποιες δοκιμαστικές κλήσεις μεταξύ των χρηστών 2001 και 2002. Στο παρακάτω σχήμα (Σχήμα 4.6) παρουσιάζουμε την κονσόλα CLI του Asterisk κατά την διάρκεια μιας κλήσης από τον 2001 στον 2002:





Σχήμα 4.7: Ethernet trace της κλήσης από τον χρήστη 2001 στον χρήστη 2002

Στο παρακάτω σχήμα (Σχήμα 4.8) βλέπουμε το σενάριο όπου ο 2001 καλεί τον 2002 ο οποίος δεν απαντά και έτσι μετά από 10 δευτερόλεπτα η κλήση πηγαίνει στο φωνητικό του ταχυδρομείο. Βλέπουμε ότι ακολουθούνται τα extension που έχουμε ορίσει για την περίπτωση που ο χρήστης δεν απαντάει και έτσι το Dialstatus παίρνει την τιμή NOANSWER (παράμετρος u σαν όρισμα στο Voicemail ώστε να ακουστεί το αντίστοιχο μήνυμα).

```
*CLI>
*CLI>      Executing [2002@numberplan-custom-2:1] NoOp("SIP/2001-0821f0d8", ""
) in new stack
      Executing [2002@numberplan-custom-2:2] Dial("SIP/2001-0821f0d8", "SIP/200
2:10!tr") in new stack
      Called 2002
      SIP/2002-08199b50 is ringing
      Nobody picked up in 10000 ms
      Executing [2002@numberplan-custom-2:3] Goto("SIP/2001-0821f0d8", "2002-NO
ANSWER!1") in new stack
      Goto (numberplan-custom-2,2002-NOANSWER,1)
      Executing [2002-NOANSWER@numberplan-custom-2:1] VoiceMail("SIP/2001-0821f
0d8", "u2002") in new stack
      <SIP/2001-0821f0d8> Playing 'vm-theperson' (language 'en')
      <SIP/2001-0821f0d8> Playing 'digits/2' (language 'en')
      <SIP/2001-0821f0d8> Playing 'digits/0' (language 'en')
      <SIP/2001-0821f0d8> Playing 'digits/0' (language 'en')
      <SIP/2001-0821f0d8> Playing 'digits/2' (language 'en')
      <SIP/2001-0821f0d8> Playing 'vm-isunavail' (language 'en')
      <SIP/2001-0821f0d8> Playing 'vm-intro' (language 'en')
```

Σχήμα 4.8: Σενάριο κλήσης από τον 2001 στον 2002 όπου ο 2002 δεν απαντά και η κλήση δρομολογείται στο φωνητικό του ταχυδρομείο.

Στο παρακάτω σχήμα (Σχήμα 4.9) βλέπουμε το σενάριο όπου ο 2001 καλεί τον 2002 ο οποίος είναι απασχολημένος (στο συγκεκριμένο παράδειγμα ο 2002 απορρίπτει την εισερχόμενη κλήση) και έτσι η κλήση πηγαίνει στο φωνητικό του ταχυδρομείο. Βλέπουμε ότι ακολουθούνται τα extension που έχουμε ορίσει για την περίπτωση που ο χρήστης είναι απασχολημένος και έτσι το Dialstatus παίρνει την τιμή BUSY (παράμετρος b σαν όρισμα στο Voicemail ώστε να ακουστεί το αντίστοιχο μήνυμα).

```
*CLI>      Executing [2002@numberplan-custom-2:1] NoOp("SIP/2001-0820b858", ""
) in new stack
      Executing [2002@numberplan-custom-2:2] Dial("SIP/2001-0820b858", "SIP/200
2:10!tr") in new stack
      Called 2002
      SIP/2002-08213688 is ringing
      Got SIP response 486 "Busy Here" back from 192.168.1.50
      SIP/2002-08213688 is busy
      Everyone is busy/congested at this time (1:1/0/0)
      Executing [2002@numberplan-custom-2:3] Goto("SIP/2001-0820b858", "2002-BU
SY!1") in new stack
      Goto (numberplan-custom-2,2002-BUSY,1)
      Executing [2002-BUSY@numberplan-custom-2:1] VoiceMail("SIP/2001-0820b858"
, "b2002") in new stack
      <SIP/2001-0820b858> Playing 'vm-theperson' (language 'en')
      <SIP/2001-0820b858> Playing 'digits/2' (language 'en')
      <SIP/2001-0820b858> Playing 'digits/0' (language 'en')
      <SIP/2001-0820b858> Playing 'digits/0' (language 'en')
      <SIP/2001-0820b858> Playing 'digits/2' (language 'en')
      <SIP/2001-0820b858> Playing 'vm-isonphone' (language 'en')
      <SIP/2001-0820b858> Playing 'vm-intro' (language 'en')
```

Σχήμα 4.9: Σενάριο κλήσης από τον 2001 στον 2002 όπου ο 2002 είναι απασχολημένος και η κλήση δρομολογείται στο φωνητικό του ταχυδρομείο.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΡΑΙΑ



## **ΚΕΦΑΛΑΙΟ 5**

### **Εφαρμογές πάνω στην πλατφόρμα του Asterisk**

Στο κεφάλαιο αυτό θα παρουσιάσουμε την εφαρμογές που αναπτύξαμε πάνω στην πλατφόρμα του Asterisk. Οι εφαρμογές που θα παρουσιάσουμε στην συνέχεια είναι:

- Call Recording: Εφαρμογή για την καταγραφή των συνομιλιών.
- Σύνδεση του Asterisk με πάροχο VoIP: Εφαρμογή με την οποία συνδέσαμε το Asterisk με έναν πάροχο VoIP ώστε να τερματίζουμε τις κλήσεις μας στο PSTN δίκτυο μέσω αυτού.
- Σύνδεση 2 Asterisk κέντρων μεταξύ τους: Εφαρμογή με την οποία διασυνδέσαμε δύο Asterisk server μεταξύ τους με την χρήση του πρωτοκόλλου IAX.
- Meet me conference: Εφαρμογή για την δημιουργία «δωματίων» τηλεδιασκέψεων.
- Call Park: Εφαρμογή για την διαχείριση εισερχομένων κλήσεων, ώστε αυτές να μπορούν να εξυπηρετηθούν από τον κατάλληλο τηλεφωνικό εκπρόσωπο.

#### **5.1 Εφαρμογή για την καταγραφή των συνομιλιών**

Η εφαρμογή αφορά την καταγραφή των κλήσεων που γίνονται από και προς το Asterisk. Υπάρχουν δύο διαφορετικοί τρόποι λειτουργίας αυτής της εφαρμογής, ο ένας είναι ο χρήστης που καλεί να ακούει ένα ενημερωτικό μήνυμα ότι η κλήση του για λόγους ασφαλείας καταγράφεται και στην συνέχεια να δρομολογείται στον χρήστη που έχει καλέσει. Ο άλλος τρόπος είναι να επιλέξουμε κάποιους χρήστες που επιθυμούμε να

καταγράφουμε τις κλήσεις τους και με την κατάλληλη διαμόρφωση στα configuration files, όλες οι κλήσεις του να καταγράφονται χωρίς αυτοί να έχουν την παραμικρή ένδειξη για αυτό.

Ο δεύτερος τρόπος καταγραφής αν δεν χρησιμοποιηθεί σωστά από τους διαχειριστές του Asterisk μπορεί να επισύρει νομικές κυρώσεις προς αυτούς. Βέβαια στις περισσότερες Χώρες υπάρχει το αντίστοιχο νομικό πλαίσιο που να επιτρέπει την καταγραφή υπό ορισμένες προϋποθέσεις και σε ορισμένους φορείς (Αστυνομία, μυστικές υπηρεσίες κτλ). Για παράδειγμα στις Ηνωμένες πολιτείες Αμερικής υπάρχει από το 1995 το νομικό πλαίσιο και οι τεχνικές προδιαγραφές για την παρακολούθηση τηλεφωνικών κλήσεων (CALEA-Communications Assistance for Law Enforcement Act) το οποίο στη συνέχεια επεκτάθηκε ώστε να προβλέπει και την παρακολούθηση VoIP κλήσεων καθώς και τις κινήσεις του διαδικτύου. Το αντίστοιχο πρότυπο στην Ευρωπαϊκή Ένωση είναι το LI (Lawful interception) .

### **5.1.1 Καταγραφή κλήσεων με ειδοποίηση του καλούντος**

Θεωρούμε ότι έχουμε μια εταιρία με 10 τηλεφωνικούς εκπροσώπους οι οποίοι εξυπηρετούν πελάτες της εταιρίας και για λόγους ασφαλείας θέλουμε οι κλήσεις αυτές να καταγράφονται. Πρέπει όμως πριν την εκκίνηση της καταγραφής οι πελάτες να ενημερώνονται ότι η κλήση τους επρόκειτο να καταγραφεί και να συνεχίζουν εφόσον το επιθυμούν. Παρακάτω παρουσιάζουμε τα extension που πρέπει να προσθέσουμε στο αρχείο extensions.conf:

```
exten= _200X,1,Answer
exten= _200X,2,wait(1)
exten= _200X,3,Playback(/var/lib/asterisk/sounds/record/record_notification)
exten= _200X,4,Set(CALLFILENAME=${CALLERID(num)}-${EXTEN}-
${STRFTIME(${EPOCH},,%d%m%Y-%H:%M:%S)})
exten= _200X,5,Monitor(wav,${CALLFILENAME})
exten= _200X,6,Dial(SIP/${EXTEN:0})
```

Βλέπουμε λοιπόν ότι έχουμε δημιουργήσει ένα σύνολο κανόνων για δέκα τηλεφωνικούς εκπροσώπους (2000-2009). Όταν λοιπόν κάποιος καλέσει έναν από τους δέκα αυτούς χρήστες η κλήση του θα απαντηθεί και μετά από μια παύση 1 δευτερολέπτου θα ακούσει ένα ηχητικό μήνυμα ότι η κλήση του ηχογραφείται για λόγους ασφαλείας. Το μήνυμα αυτό μπορούμε να το ηχογραφήσουμε μέσω της αντίστοιχης επιλογής που υπάρχει στο Asterisk GUI είτε να το δημιουργήσουμε με κάποιο άλλον τρόπο και να το εισάγουμε στο σύστημα αρχείων του Asterisk. Στην συγκεκριμένη περίπτωση το δημιουργήσαμε μέσω του Asterisk GUI και το αρχείο βρίσκεται αποθηκευμένο στο `/var/lib/asterisk/sounds/record/` με το όνομα `record_notification`. Έτσι με το παρακάτω extension αναπαραγάγουμε αυτό το μήνυμα στον καλών:

```
exten= _200X,3,Playback(/var/lib/asterisk/sounds/record/record_notification)
```

Με την επόμενη γραμμή δίνουμε ένα όνομα στο ηχητικό αρχείο που θα δημιουργηθεί από την καταγραφή αυτής της κλήσης. Έχουμε ορίσει αυτό το extension με τέτοιο τρόπο ώστε το όνομα του αρχείου να περιέχει το νούμερο του χρήστη που έχει καλέσει, το νούμερο του εκπρόσωπου τον οποίο κάλεσε καθώς την ημερομηνία και την ώρα που έγινε η κλήση.

```
exten= _200X,4,Set(CALLFILENAME=${CALLERID(num)}-${EXTEN}-  
${STRFTIME(${EPOCH},,%d%m%Y-%H:%M:%S)})
```

Στην συνέχεια λέμε στο Asterisk να αρχίσει την καταγραφή της κλήσης και να δημιουργήσει ένα αρχείο `wav` με το όνομα που θέσαμε παραπάνω. Η εντολή `Monitor` δημιουργεί το αρχείο αυτό στο `/var/spool/asterisk/monitor`.

```
exten= _200X,5,Monitor(wav,${CALLFILENAME})
```

Τέλος δρομολογούμε την κλήση στον τηλεφωνικό εκπρόσωπο το νούμερο του οποίου κάλεσε ο καλών:

```
exten=_200X,6,Dial(SIP/${EXTEN:0})
```

Στο παρακάτω σχήμα (Σχήμα 5.1) παρουσιάζουμε την λειτουργία της παραπάνω εφαρμογής όπως αυτή φαίνεται στην κονσόλα του Asterisk. Σε αυτό το παράδειγμα ο χρήστης 4001 καλεί τον τηλεφωνικό εκπρόσωπο 2001 και όπως βλέπουμε παρακάτω ακολουθούνται όλα τα extension που περιγράψαμε παραπάνω. Να αναφέρουμε επίσης ότι η λειτουργία αυτής της εφαρμογής μπορεί να επεκταθεί και για εξωτερικούς χρήστες (pstn) που καλούν κάποιον χρήστη του asterisk στην περίπτωση που το κέντρο μας είναι συνδεδεμένο με το pstn δίκτυο για παράδειγμα μέσω κάποιας κάρτας PRI.

```
*CLI> Executing [2001@numberplan-custom-2:1] Answer("SIP/4001-0822bed8",
"") in new stack
Executing [2001@numberplan-custom-2:2] Wait("SIP/4001-0822bed8", "1") in
new stack
Executing [2001@numberplan-custom-2:3] Playback("SIP/4001-0822bed8", "/va
r/lib/asterisk/sounds/record/record_notification") in new stack
<SIP/4001-0822bed8> Playing '/var/lib/asterisk/sounds/record/record_notif
ication' (language 'en')
Executing [2001@numberplan-custom-2:4] Set("SIP/4001-0822bed8", "CALLFILE
NAME=4001-2001-16012010-21:33:45") in new stack
Executing [2001@numberplan-custom-2:5] Monitor("SIP/4001-0822bed8", "wav:
4001-2001-16012010-21:33:45") in new stack
Executing [2001@numberplan-custom-2:6] Dial("SIP/4001-0822bed8", "SIP/200
1") in new stack
Called 2001
SIP/2001-08227208 is ringing
SIP/2001-08227208 answered SIP/4001-0822bed8
Spawn extension (numberplan-custom-2, 2001, 6) exited non-zero on 'SIP/4001
-0822bed8'
```

**Σχήμα 5.1:** Καταγραφή κλήσης από τον χρήστη 4001 στον 2001, όπου ο χρήστης 4001 ενημερώνεται με ένα ηχητικό μήνυμα ότι η κλήση του επρόκειτο να καταγραφεί.

Από το παραπάνω παράδειγμα δημιουργήθηκαν 2 αρχεία wav στον φάκελο /var/spool/asterisk/monitor:

- 4001-2001-16012010-21:33:45-in.wav
- 4001-2001-16012010-21:33:45-out.wav

Η δημιουργία δύο αρχείων εξηγείται από τον τρόπο λειτουργίας της εντολής Monitor όπου δημιουργεί ένα αρχείο για τα εισερχόμενα προς τον 2001 αρχεία φωνής καθώς και ένα για τα εξερχόμενα.

### 5.1.2 Καταγραφή κλήσεων χωρίς την ειδοποίηση του καλούντος

Ο σκοπός αυτής της εφαρμογής είναι η παρακολούθηση των κλήσεων ενός χρήστη του Asterisk χωρίς αυτός να μπορεί να αντιληφθεί ότι η κλήση του καταγράφεται. Για παράδειγμα με τα παρακάτω extension μπορούμε να παρακολουθήσουμε όλες τις εισερχόμενες κλήσεις προς τον 2002.

```
exten=2002,1,Answer
exten=2002,2,wait(1)
exten=2002,3,Set(CALLFILENAME=${CALLERID(num)}-${EXTEN}-
${STRFTIME(${EPOCH},,%d%m%Y-%H:%M:%S)})
exten=2002,4,Monitor(wav,${CALLFILENAME})
exten=2002,5,Dial(SIP/2002)
```

Βλέπουμε ότι η λειτουργία αυτών των extension είναι ίδια με αυτήν που περιγράψαμε στο προηγούμενο κεφάλαιο με την διαφορά ότι απουσιάζει το extension με το ηχητικό μήνυμα της ειδοποίησης ότι η κλήση καταγράφεται. Έτσι ο χρήστης δεν αντιλαμβάνεται ότι η κλήση του καταγράφεται.

Με την παραπάνω υλοποίηση όμως καταγράφουμε μόνο τις εισερχόμενες προς τον 2001 κλήσεις. Για να καταγράψουμε όλες τις κλήσεις ενός ή και περισσότερων χρηστών θα χρησιμοποιήσουμε τα παρακάτω extension. Έστω λοιπόν ότι το κέντρο μας έχει 100 συνδρομητές με νούμερα από 2000 έως 2099 και θέλουμε να παρακολουθήσουμε τις κλήσεις ενός η περισσότερων συνδρομητών.

```
a. exten= _20XX,1,Answer
   exten= _20XX,2,wait(1)
   exten= _20XX,3,Set(CALLFILENAME=${CALLERID(num)}-${EXTEN}-
   ${STRFTIME(${EPOCH},,%d%m%Y-%H:%M:%S)})
   exten= _20XX,4,Monitor(wav,${CALLFILENAME})
   exten= _20XX,5,Dial(SIP/${EXTEN}:0)

b. exten= _69XXXXXXXX,1,Answer
```

```

exten= _69XXXXXXXX,2,wait(1)
exten= _69XXXXXXXX,3,Set(CALLFILENAME=${CALLERID(num)}-${EXTEN}-
${STRFTIME(${EPOCH},,%d%m%Y-%H:%M:%S)})
exten= _69XXXXXXXX,4,Monitor(wav,${CALLFILENAME})
exten= _69XXXXXXXX,5,Dial(SIP/voipdiscount2/0030${EXTEN:0})

c. exten= _21XXXXXXXX,1,Answer
exten= _21XXXXXXXX,2,wait(1)
exten= _21XXXXXXXX,3,Set(CALLFILENAME=${CALLERID(num)}-${EXTEN}-
${STRFTIME(${EPOCH},,%d%m%Y-%H:%M:%S)})
exten= _21XXXXXXXX,4,Monitor(wav,${CALLFILENAME})
exten= _21XXXXXXXX,5,Dial(SIP/voipdiscount2/0030${EXTEN:0})

```

Με τα παραπάνω 3 σύνολα extension (a, b, c) έχουμε καλύψει όλες τις περιπτώσεις εισερχομένων και εξερχομένων κλήσεων στο κέντρο μας και έτσι στην συνέχεια με ένα φιλτράρισμα των αρχείων που έχουν δημιουργηθεί, με βάση το όνομά τους το οποίο όπως αναφέραμε περιέχει το νούμερο τόσο του καλών όσο και του καλούντα, μπορούμε να συλλέξουμε τα αρχεία των χρηστών που μας ενδιαφέρουν.

Πιο συγκεκριμένα τα πρώτα 5 extension (a) καταγράφουν οποιαδήποτε εσωτερική κλήση μεταξύ των συνδρομητών του κέντρου μας. Επίσης, στην περίπτωση που το κέντρο μας διαθέτει κάποια κάρτα που το συνδέει με το pstn δίκτυο, με τα extension αυτά καταγράφονται και όλες οι εισερχόμενες κλήσεις από το pstn δίκτυο προς τους συνδρομητές του κέντρου μας. Τα επόμενα 5 extension (b) καταγράφουν όλες τις εξερχόμενες κλήσεις προς τα κινητά τηλέφωνα (στην συγκεκριμένη περίπτωση αφορά τα δίκτυα των ελληνικών παρόχων κινητής τηλεφωνίας). Οι κλήσεις αυτές στο κέντρο μας γίνονται μέσω ενός παρόχου VoIP όπως θα εξηγήσουμε σε επόμενη παράγραφο. Αντίστοιχα τα τελευταία 5 extension (c) καταγράφουν όλες τις εξερχόμενες κλήσεις προς όλα τα σταθερά τηλέφωνα στην Αττική. Με ανάλογο τρόπο μπορούμε να προσθέσουμε extension για νούμερα άλλων περιοχών ή χωρών ή για κλήσης μέσω της κάρτας PRI.

Τα αρχεία wav των καταγεγραμμένων κλήσεων αποθηκεύονται όπως και στην προηγούμενη εφαρμογή στο /var/spool/asterisk/monitor. Στην συνέχεια (Σχήμα 5.2)

παρουσιάζουμε ένα παράδειγμα καταγραφής εξερχόμενης κλήσης από τον συνδρομητή 2001 σε κάποιο κινητό τηλέφωνο (έχουμε αποκρύψει τα τελευταία 4 νούμερα αυτού του τηλεφώνου).

```
*CLI> Executing [69774-XXXXXX@numberplan-custom-2:1] Answer("SIP/2001-0822e2f8", "") in new stack
Executing [69774-XXXXXX@numberplan-custom-2:2] Wait("SIP/2001-0822e2f8", "1") in new stack
Executing [69774-XXXXXX@numberplan-custom-2:3] Set("SIP/2001-0822e2f8", "CALLFILENAME=2001-69774-XXXXXX-23012010-23:58:24") in new stack
Executing [69774-XXXXXX@numberplan-custom-2:4] Monitor("SIP/2001-0822e2f8", "wav|2001-69774-XXXXXX-23012010-23:58:24") in new stack
Executing [69774-XXXXXX@numberplan-custom-2:5] Dial("SIP/2001-0822e2f8", "SIP/voipdiscount2/00306977-XXXXXX") in new stack
Called voipdiscount2/00306977-XXXXXX
SIP/voipdiscount2-08235290 is making progress passing it to SIP/2001-0822e2f8
SIP/voipdiscount2-08235290 answered SIP/2001-0822e2f8
Spawn extension (numberplan-custom-2, 69774-XXXXXX, 5) exited non-zero on 'SIP/2001-0822e2f8'
```

**Σχήμα 5.2:** Καταγραφή κλήσης από τον χρήστη 2001 σε ένα κινητό τηλέφωνο (6977XXXXXX) μέσω του πάροχου Voipdiscount, όπου και οι δύο χρήστες δεν γνωρίζουν ότι η κλήση τους καταγράφεται.

Από την παραπάνω κλήση δημιουργήθηκαν τα παρακάτω αρχεία wav:

- 2001-6977XXXXXX-23012010-23:58:24-in.wav
- 2001-6977XXXXXX-23012010-23:58:24-out.wav

Από τα παραπάνω αρχεία φαίνεται ξεκάθαρα ότι πρόκειται για μια εξερχόμενη κλήση από τον χρήστη 2001 προς το κινητό τηλέφωνο με αριθμό 6977XXXXXX (τα τελευταία ψηφία του αριθμού δεν δημοσιοποιούνται σε αυτήν την εργασία για λόγους εμπιστευτικότητας). Η κλήση αυτή έγινε στις 23 Ιανουαρίου 2010 στις 23:58.

## 5.2 Σύνδεση του Asterisk με πάροχο Υπηρεσιών VoIP

Με την εφαρμογή αυτή διασύνδεουμε το Asterisk με έναν πάροχο υπηρεσιών VoIP και συγκεκριμένα τον πάροχο Voipdiscount. Στόχος αυτής της διασύνδεσης είναι η δρομολόγηση κλήσεων από κάποιον χρήστη του Asterisk προς κάποιο τηλέφωνο του pstn δικτύου. Στην συγκεκριμένη εφαρμογή μπορούμε από κάποιον χρήστη του Asterisk

να καλέσουμε όλα τα ελληνικά κινητά τηλέφωνα και τα σταθερά τηλέφωνα του νομού Αττικής. Να αναφέρουμε εδώ ότι με ανάλογο τρόπο μπορεί αυτή η εφαρμογή να επεκταθεί για νούμερα άλλων περιοχών ή χωρών που μας ενδιαφέρουν καθώς και για την διασύνδεση με οποιονδήποτε πάροχο VoIP ο οποίος χρησιμοποιεί το πρωτόκολλο SIP.

Τα πλεονεκτήματα αυτής της εφαρμογής είναι ότι οι χρήστες του Asterisk μπορούν χωρίς καμία προσθήκη hardware να καλούν σταθερά και κινητά τηλέφωνα στο κλασικό pstn δίκτυο τηλεφωνίας. Επίσης οι περισσότεροι πάροχοι τηλεφωνικών υπηρεσιών μέσω VoIP (ITSP - Internet Telephony Service Provider) προσφέρουν πολύ χαμηλές χρεώσεις κλήσεων τόσο προς την Ελλάδα όσο και προς το εξωτερικό. Υπάρχουν αρκετοί πάροχοι τηλεφωνικών υπηρεσιών μέσω VoIP όπως για παράδειγμα: Voipdiscount, 12voip, VoipBuster, SIPDiscount, OMNIVOICE. Για να χρησιμοποιήσουμε κάποιον από του παρόχους η διαδικασία είναι απλή, αφού απαιτεί μόνο να ανοίξουμε ένα λογαριασμό στο site του παρόχου και να προπληρώσουμε ένα ποσό το οποίο επιθυμούμε. Στην συνέχεια μπορούμε με την χρήση του Username και του password μας καθώς και την IP ή το URL του παρόχου (συνήθως στις οδηγίες αναφέρετε ως Registrar) να διασυνδέσουμε το Asterisk με τον πάροχο.

Για να υλοποιήσουμε αυτήν την εφαρμογή πρέπει καταρχάς να δημιουργήσουμε ένα SIP-Trunk με τον πάροχο Voipdiscount και στην συνέχεια να δημιουργήσουμε τα αντίστοιχα extension για την δρομολόγηση των κλήσεων. Επίσης είναι προφανές ότι για την λειτουργία αυτής της εφαρμογής το σύστημά μας πρέπει να είναι συνδεδεμένο με το διαδίκτυο. Αυτές οι ενέργειες μπορούν να γίνουν τόσο από το γραφικό περιβάλλον όσο και με την απευθείας διαμόρφωση των αρχείων. Παρακάτω θα παρουσιάσουμε και τους δύο τρόπους δίνοντας βάση περισσότερο στην απευθείας διαμόρφωση μιας και έχουμε περισσότερες επιλογές διαμόρφωσης αλλά μας δίνει και μια καλύτερη εικόνα του τρόπου λειτουργίας αυτής της εφαρμογής. Τέλος θα παρουσιάσουμε κάποια παραδείγματα κλήσεων μέσω του παρόχου Voipdiscount προς εξωτερικά τηλέφωνα.



### 5.2.1 Διαμόρφωση της εφαρμογής μέσω του γραφικού περιβάλλοντος Asterisk GUI

Ξεκινάμε με την δημιουργία του SIP Trunk με τον πάροχο Voipdiscount πηγαίνοντας στο μενού Service Providers→Add Service Provider→ Custom VoIP και συμπληρώνουμε τα απαραίτητα πεδία όπως φαίνεται στο παρακάτω σχήμα (Σχήμα 5.3). Δίνουμε ένα όνομα στο SIP Trunk που θα δημιουργήσουμε (voipdiscount) καθώς και το πρωτόκολλο που θα χρησιμοποιεί (SIP). Επίσης με την ενεργοποίηση της επιλογής register λέμε στον Asterisk να στέλνει ένα μήνυμα SIP προς τον server του παρόχου για την εγγραφή του σε αυτόν. Τέλος ορίζουμε την IP ή το URL του server του παρόχου καθώς και τα username και password που έχουμε αποκτήσει από την εγγραφή στην υπηρεσία του.

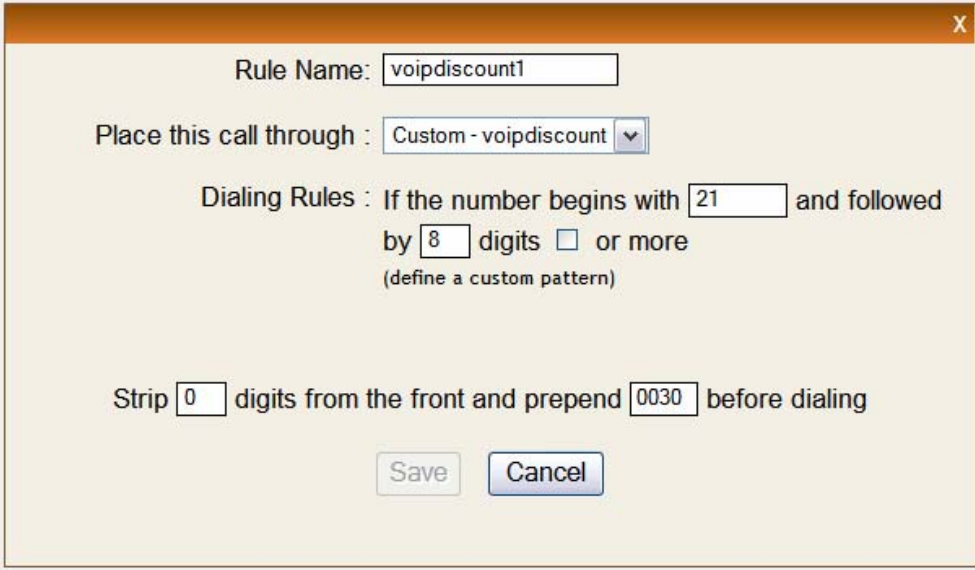


The screenshot shows the 'Edit Service Provider' window in the Asterisk GUI. The 'Provider Type' section has three radio buttons: 'Analog', 'VoIP', and 'Custom VoIP', with 'Custom VoIP' selected. Below this, there are several input fields: 'Comment' with the value 'voipdiscount', 'Protocol' with a dropdown menu set to 'SIP', 'Register' with a checked checkbox, 'Host' with the value 'sip.voipdiscount.com', 'Username' with the value 'johnadmingr', and 'Password' with a masked field of ten dots. At the bottom of the window are 'Save' and 'Cancel' buttons.

Σχήμα 5.3: Δημιουργία ενός SIP Trunk μέσω γραφικού περιβάλλοντος.

Στην συνέχεια αφού έχουμε δημιουργήσει το SIP Trunk με τον πάροχο πρέπει να δημιουργήσουμε κάποιους κανόνες για την δρομολόγηση των κλήσεών μας προς τον πάροχο αυτόν. Έτσι από το μενού Calling Rules→ Add a Calling Rule δημιουργούμε τον παρακάτω κανόνα (Σχήμα 5.4) που αφορά την δρομολόγηση κλήσεων προς όλα τα σταθερά τηλέφωνα στο Νομό Αττικής. Συγκεκριμένα έχουμε ορίσει ότι όλα τα νούμερα που ξεκινούν με τον αριθμό 21 και περιέχουν άλλα 8 ψηφία να δρομολογούνται μέσω

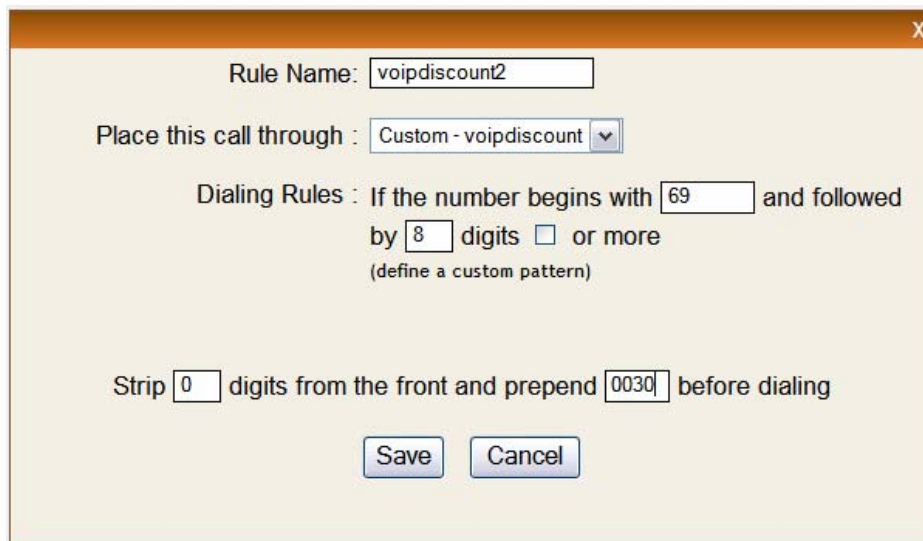
του SIP Trunk με το όνομα Voipdiscount προσθέτοντας το 0030 στην αρχή του αριθμού. Έτσι ο κάθε χρήστης καλεί τον αριθμό που θέλει με την μορφή 21XXXXXXXX και το Asterisk δρομολογεί την κλήση προς τον πάροχο Voipdiscount λέγοντάς του να καλέσει τον αριθμό 003021XXXXXXXX. Η παραπάνω προσθήκη των ψηφίων 0030 (όπου 0030 είναι ο κωδικός της Ελλάδος) γίνεται γιατί ο πάροχος Voipdiscount χρειάζεται και τον κωδικό της χώρας ώστε να καλέσει έναν αριθμό.



The screenshot shows a configuration window for a dialing rule. The 'Rule Name' is 'voipdiscount1'. The 'Place this call through' dropdown is set to 'Custom - voipdiscount'. The 'Dialing Rules' section is configured with 'If the number begins with 21 and followed by 8 digits or more (define a custom pattern)'. The 'Strip 0 digits from the front and prepend 0030 before dialing' option is checked. 'Save' and 'Cancel' buttons are at the bottom.

**Σχήμα 5.4:** Δρομολόγηση κλήσεων προς όλα τα σταθερά τηλέφωνα στην Αττική μέσω Voipdiscount

Με τον ίδιο τρόπο δημιουργούμε και ένα κανόνα για την δρομολόγηση των κλήσεων προς όλα τα ελληνικά κινητά τηλέφωνα (Σχήμα 5.5). Εδώ δρομολογούμε όλα τα νούμερα που ξεκινούν από 69 και περιέχουν άλλα 8 ψηφία μέσω του Voipdiscount προσθέτοντας όπως και πριν τον κωδικό της χώρας.



The screenshot shows a configuration window for a dialing rule. The 'Rule Name' is 'voipdiscount2'. The 'Place this call through' dropdown is set to 'Custom - voipdiscount'. The 'Dialing Rules' section is configured with 'If the number begins with 69 and followed by 8 digits or more (define a custom pattern)'. The 'Strip 0 digits from the front and prepend 0030 before dialing' option is selected. There are 'Save' and 'Cancel' buttons at the bottom.

Σχήμα 5.5: Δρομολόγηση κλήσεων προς όλα τα ελληνικά κινητά τηλέφωνα μέσω Voipdiscount

Αφού δημιουργήσουμε το SIP Trunk και τους κανόνες όπως περιγράψαμε παραπάνω μπορούμε να κάνουμε κλήσεις προς το pstn δίκτυο.

### 5.2.2 Διαμόρφωση της εφαρμογής μέσω της απευθείας διαμόρφωσης των αρχείων του Asterisk

Παρακάτω θα παρουσιάσουμε τον τρόπο με τον οποίο μπορούμε να διασυνδέσουμε το Asterisk με τον πάροχο Voipdiscount χωρίς την χρήση του γραφικού περιβάλλοντος αλλά με την κατάλληλη διαμόρφωση των αρχείων του. Αρχικά δημιουργούμε ένα Sip Trunk με τον πάροχο με την δημιουργία του παρακάτω context μέσα στο αρχείο sip.conf:

```
[general]
```

```
register=johnadmingr:hello@1454@sip.voipdiscount.com
```

```
[voipdiscount2]
```

```
type=friend
```

```
protocol=sip
```

```
username=johnadmingr
secret=hello@1454
fromuser=johnadmingr
host=sip.voipdiscount.com
dtmfmode=rfc2833
fromdomain=sip.voipdiscount.com
context=default
insecure=very
qualify=yes
nat=yes
allow=ulaw
progressinband=yes
canreinvite=no
context=outgoing
externip=83.212.106.218
localnet=192.168.1.0/255.255.255.0
```

Βλέπουμε ότι έχουμε προσθέσει μέσα στο context general μία εντολή με την οποία στέλνουμε μια αίτηση προς τον πάροχο για εγγραφή στην υπηρεσία του (*register=johnadmingr:hello@1454@sip.voipdiscount.com*), η οποία περιέχει το username και το password μας. Επίσης έχουμε δημιουργήσει ένα νέο context με το όνομα *voipdiscount2* στο οποίο έχουμε ορίσει διάφορες παραμέτρους όπως, ότι χρησιμοποιούμε το πρωτόκολλο sip, τα username και password του λογαριασμού που έχουμε ανοίξει στον πάροχο, το url του παρόχου (*sip.voipdiscount.com*), τον τύπο των dtmf τόνων που χρησιμοποιούμε (*rfc2833*) καθώς και τον codec συμπίεσης φωνής τον οποίο επιτρέπουμε και ο οποίος είναι ο G.711 (*ulaw*). Επίσης έχουμε ορίσει και κάποιες δικτυακές παραμέτρους όπως η δημόσια IP του modem μας (*externip=83.212.106.218*) μέσω της οποίας το Asterisk είναι συνδεδεμένο με το διαδίκτυο, το εσωτερικό τοπικό του δίκτυο (*localnet=192.168.1.0/255.255.255.0*), το ότι βρίσκεται πίσω από Nat (*nat=yes*) καθώς και ότι δεν μπορεί ο χρήστης να ανταλλάξει απευθείας μηνύματα με το server του παρόχου αλλά πρέπει να παρεμβάινει το Asterisk ενδιάμεσα (*canreinvite=no*).

Στην συνέχεια παρουσιάζουμε τα extensions που προσθέσαμε στο αρχείο extensions.conf ώστε να είναι δυνατή η δρομολόγηση κλήσεων προς όλα τα ελληνικά κινητά τηλέφωνα καθώς και τα σταθερά στο Νομό Αττικής.

```
exten=_69XXXXXXXX,1,Dial(SIP/voipdiscount2/0030${EXTEN:0})  
exten=_21XXXXXXXX,1,Dial(SIP/voipdiscount2/0030${EXTEN:0})
```

Βλέπουμε λοιπόν ότι όλες οι κλήσεις προς νούμερα που ξεκινούν με τον αριθμό 21 και ακολουθούνται από 8 ψηφία (σταθερά τηλέφωνα στην Αττική) καθώς και νούμερα που ξεκινούν με τους αριθμούς 69 και ακολουθούνται από 8 ψηφία (κινητά τηλέφωνα) δρομολογούνται μέσω του sip trunk με τον πάροχο voipdiscount. Επίσης όπως είδαμε και στην προηγούμενη παράγραφο πρέπει να προσθέσουμε μπροστά από το νούμερο και τον κωδικό της Ελλάδος ώστε η κλήση να είναι εφικτή μέσω του παρόχου (0030\${EXTEN:0}). Με ανάλογο τρόπο μπορούμε να προσθέσουμε extension που να καλύπτουν και άλλες περιοχές ή χώρες που μας ενδιαφέρουν. Έτσι για παράδειγμα εάν θέλαμε να καλούμε όλα τα σταθερά στην Ελλάδα και στην Γερμανία θα έπρεπε να έχουμε τα εξής extension:

```
exten=_2XXXXXXXX,1,Dial(SIP/voipdiscount2/0030${EXTEN:0})  
exten=_004989XXXXXXXX,1,Dial(SIP/voipdiscount2/${EXTEN:0})
```

Όπου έχουμε ορίσει ότι για να καλέσει κάποιος στη Γερμανία θα πρέπει να προσθέσει πρώτα τον κωδικό της χώρας (0049).

### 5.2.3 Δοκιμή της εφαρμογής

Στην παράγραφο αυτή θα δοκιμάσουμε την εφαρμογή που δημιουργήσαμε παραπάνω κάνοντας κάποιες κλήσεις μέσω του παρόχου voipdiscount. Ξεκινάμε ελέγχοντας το κατά πόσο το sip trunk που δημιουργήσαμε προηγουμένως έχει ολοκληρώσει την εγγραφή στον πάροχο. Αυτό μπορεί να γίνει δίνοντας την εντολή «sip show registry» στην κονσόλα CLI του Asterisk:

Command> sip show registry

```
Host                Username      Refresh State  Reg.Time
sip.voipdiscount.com:5060 johnadmingr 105 Registered Sun, 20 Dec 2009 21:27:09
```

Βλέπουμε λοιπόν ότι έχουμε εγγραφεί στην υπηρεσία του παρόχου με url sip.voipdiscount.com στην πόρτα 5060. Η εγγραφή αυτή αφορά τον χρήστη με το username johnadmingr και έγινε στις 20 Δεκεμβρίου του 2009 στις 21:17. Παρακάτω βλέπουμε (Σχήμα 5.6) την ανταλλαγή μηνυμάτων μεταξύ του Asterisk και του server του παρόχου για την εγγραφή μας στην υπηρεσία του.

Time	192.168.1.66	194.120.0.198	77.72.169.129	Comment
2,943				SIP: Request: OPTIONS sip:sip.voipdiscount.com
3,055				SIP: Status: 200 Ok
3,127				SIP: Request: REGISTER sip:sip.voipdiscount.com
3,218				SIP: Status: 401 Unauthorized (1 bindings)
3,219				SIP: Request: REGISTER sip:sip.voipdiscount.com
3,312				SIP: Status: 100 Trying (1 bindings)
3,314				SIP: Status: 200 Ok (1 bindings)

Σχήμα 5.6: Εγγραφή στην υπηρεσία του παρόχου voipdiscount με την χρήση url

Βλέπουμε λοιπόν ότι έχουμε ανταλλαγή sip μηνυμάτων μεταξύ του Asterisk (192.168.1.66) και 2 IP, της 194.120.0.198 η οποία ανήκει στην TVICONNECT η οποία είναι ένα πάροχος τηλεφωνικών υπηρεσιών και τερματισμού κλήσεων με έδρα την Ολλανδία και της 77.72.169.129 η οποία ανήκει στην Betamax GmbH & CO.KG η οποία είναι η εταιρία πίσω από την επωνυμία voipdiscount και η τοποθεσία αυτής της IP είναι στην Ολλανδία επίσης. Έχουμε λοιπόν ένα μήνυμα sip από το Asterisk προς το url sip.voipdiscount.com το οποίο μετά την αναζήτηση σε κάποιον dns server αντιστοιχίζεται στην 194.120.0.198, η οποία προφανώς ανήκει σε κάποιον server που γνωρίζει τους server της Betamax οι οποίοι παρέχουν τις διάφορες υπηρεσίες. Ο Sever αυτός απαντάει στο Asterisk και του λέει να επικοινωνήσει με την 77.72.169.129. Στη συνέχεια το Asterisk στέλνει ένα αίτημα εγγραφής προς αυτήν την IP (register) στο οποίο



λαμβάνει ως απάντηση ένα sip μήνυμα σφάλματος (401 unauthorized). Το μήνυμα αυτό σημαίνει ότι δεν έχουμε δώσει το σωστό username και password, το οποίο είναι λογικό αφού αν δούμε το Request register μήνυμα από το Asterisk δεν περιέχει καθόλου αυτά τα στοιχεία. Έτσι το Asterisk ξαναστέλνει ένα αίτημα εγγραφής το οποίο περιέχει τα username και password που έχουμε ορίσει στο αρχείο sip.conf. Τέλος το Asterisk λαμβάνει ένα μήνυμα 200 OK το οποίο αποτελεί επιβεβαίωση της εγγραφής μας στην υπηρεσία.

Στο παρακάτω σχήμα (Σχήμα 5.7) βλέπουμε την διαδικασία εγγραφής στην υπηρεσία του παρόχου στην περίπτωση όπου στο αρχείο sip.conf δεν είχαμε το url sip.voipdiscount.com αλλά την IP 77.72.169.129. Βλέπουμε ότι η διαδικασία είναι η ίδια αλλά παρακάμπτουμε τον server 194.120.0.198. Να σημειώσουμε εδώ ότι επιβάλλεται να χρησιμοποιούμε url και όχι IP διότι οι πάροχοι συνήθως χρησιμοποιούν διαφορετικούς server για τις διάφορες υπηρεσίες η εγγραφή (registar server), ο τερματισμός των κλήσεων κτλ.

Time	192.168.1.66	77.72.169.129	Comment
3,766	Request: OPTIONS sip:77.72.169.129		SIP: Request: OPTIONS sip:77.72.169.129
3,857	Status: 200 Ok		SIP: Status: 200 Ok
4,159	Request: REGISTER sip:194.120.0.198		SIP: Request: REGISTER sip:sip.voipdiscount.com
4,248	Status: 401 Unauthorized (1 bindings)		SIP: Status: 401 Unauthorized (1 bindings)
4,249	Request: REGISTER sip:194.120.0.198		SIP: Request: REGISTER sip:sip.voipdiscount.com
4,341	Status: 100 Trying (1 bindings)		SIP: Status: 100 Trying (1 bindings)
4,342	Status: 200 Ok (1 bindings)		SIP: Status: 200 Ok (1 bindings)

Σχήμα 5.7: Εγγραφή στην υπηρεσία του παρόχου voipdiscount με την χρήση IP

Στην συνέχεια θα δούμε ένα παράδειγμα κλήσης από κάποιον χρήστη του Asterisk προς ένα σταθερό τηλέφωνο. Στο Σχήμα 5.8 βλέπουμε την κονσόλα του Asterisk για μία κλήση από τον χρήστη 2001 προς το 210XXXXXXX όπου έχουμε κρύψει τα τελευταία ψηφία για λόγους εμπιστευτικότητας.

```
*CLI> Executing [210-00000000@numberplan-custom-2:1] Dial("SIP/2001-0822e2a8", "SIP/voipdiscount2/0030210-00000000") in new stack
Called voipdiscount2/0030210-00000000
SIP/voipdiscount2-08232220 is making progress passing it to SIP/2001-0822e2a8
SIP/voipdiscount2-08232220 answered SIP/2001-0822e2a8
Packet2Packet bridging SIP/2001-0822e2a8 and SIP/voipdiscount2-08232220
Spawn extension (numberplan-custom-2, 210-00000000, 1) exited non-zero on 'SIP/2001-0822e2a8'
```

Σχήμα 5.8: Κλήση από τον 2001 προς ένα σταθερό τηλέφωνο μέσω voipdiscount

Στο παρακάτω σχήμα (Σχήμα 5.9) βλέπουμε ένα ethereal trace της κλήσης από τον 2001 σε ένα σταθερό τηλέφωνο μέσω του παρόχου voipdiscount.

Time	192.168.1.64	192.168.1.66	194.120.0.198	80.239.235.186	Comment
5,278					INVITE SDP (g711A g711U telephone-event)
5,278					SIP From: sip:2001@192.168.1.66 To:sip:00302109935057@192.168.1.66
5,278					407 Proxy Authentication Required
5,280					SIP Status
5,280					ACK
5,280					SIP Request
5,280					INVITE SDP (g711A g711U telephone-event)
5,280					SIP From: sip:2001@192.168.1.66 To:sip:00302109935057@192.168.1.66
5,281					100 Trying
5,281					SIP Status
5,284					INVITE SDP (g711U GSM g711A AAL2-G726-32 DV4...)
5,284					SIP From: sip:asterisk@83.212.105.251 To:sip:00302109935057@sip.voipdiscount.com
5,390					401 Unauthorized
5,391					SIP Status
5,391					ACK
5,391					SIP Request
5,391					INVITE SDP (g711U GSM g711A AAL2-G726-32 DV4...)
5,391					SIP From: sip:asterisk@83.212.105.251 To:sip:00302109935057@sip.voipdiscount.com
5,505					100 Trying
5,505					SIP Status
5,633					183 Session progress SDP (g711U telephone-eve...)
5,633					SIP Status
5,634					183 Session Progress SDP (g711U g711A telepho...)
5,634					SIP Status
5,670					RTP (g711U)
5,670					RTP Num packets:98 Duration:1.959s SSRC:0xA1D91EE
5,670					RTP (g711U)
5,670					RTP Num packets:98 Duration:1.959s SSRC:0x45B5D7ED
5,769					RTP (g711U)
5,769					RTP Num packets:94 Duration:1.860s SSRC:0xFF0000
5,770					RTP (g711U)
5,770					RTP Num packets:94 Duration:1.860s SSRC:0x574C7A00
7,646					200 Ok SDP (g711U telephone-event)
7,646					SIP Status
7,646					ACK
7,646					SIP Request
7,647					200 OK SDP (g711U g711A telephone-event)
7,647					SIP Status
7,649					RTP (g711U)
7,649					RTP Num packets:168 Duration:3.340s SSRC:0xFF0000
7,651					RTP (g711U)
7,651					RTP Num packets:162 Duration:3.219s SSRC:0xFF0000
7,738					ACK
7,738					SIP Request
7,797					RTP (g711U)
7,797					RTP Num packets:154 Duration:3.060s SSRC:0xE681D760
7,800					RTP (g711U)
7,800					RTP Num packets:154 Duration:3.057s SSRC:0xE681D760
10,888					BYE
10,888					SIP Request
10,888					200 OK
10,888					SIP Status
10,889					BYE
10,889					SIP Request
10,988					200 Ok
10,988					SIP Status

Σχήμα 5.8: Ethereal trace για κλήση από τον 2001 προς ένα σταθερό τηλέφωνο μέσω voipdiscount

Βλέπουμε λοιπόν ότι ο 2001 (192.168.1.64) στέλνει αρχικά ένα μήνυμα Invite προς το Asterisk (192.168.1.66) το οποίο του απαντάει με ένα μήνυμα 407 Proxy authentication και στην συνέχεια ο 2001 ξαναστέλνει ένα μήνυμα invite το οποίο περιέχει τα



απαραίτητα στοιχεία για την αυθεντικοποίηση του χρήστη προς το Asterisk. Στην συνέχεια το Asterisk, αφού εσωτερικά με βάσει τα extension που έχουμε ορίσει παραπάνω έχει δρομολογήσει την κλήση προς τον πάροχο voipdiscount, στέλνει ένα μήνυμα Invite προς τον πάροχο (194.120.0.198) ο οποίος του απαντάει με ένα μήνυμα 401 Unauthorized ώστε να του στείλει τα username και password, το οποίο και πραγματοποιεί το Asterisk με το επόμενο μήνυμα invite προς τον πάροχο. Έτσι στην συνέχεια ο πάροχος απαντάει στο Asterisk με δύο διαδοχικά μηνύματα ένα 100 Trying το οποίο σημαίνει ότι έχει ξεκινήσει τις διαδικασίες για την εδραίωση της κλήσης προς τον αριθμό που ζητήθηκε, καθώς και με το μήνυμα 183 Session progress το οποίο είναι αντίστοιχο του 180 Ringing και σηματοδοτεί την έναρξη αναπαραγωγής ringing tone. Έτσι το Asterisk προωθεί το μήνυμα 183 Session progress στον 2001 ώστε αυτός να αρχίσει να καλεί (ο χρήστης ακούει ringing tone). Η χρήση του μηνύματος 183 Session progress αντί του 180 Ringing είναι για να έχουμε ringing tone στον καλών χωρίς να έχει γίνει ολοκληρωτικό στήσιμο της συνεδρίας (session) όπως με το 180 Ringing. Αυτό χρησιμοποιείται κυρίως στις διασυνδέσεις με τα δίκτυα pstn έτσι ώστε ο καλών να μην χρεώνεται κατά την διάρκεια που ακούει ringing tone αλλά μόνο αφού απαντήσει ο συνδρομητής που έχει καλέσει [24]. Στην συνέχεια ο πάροχος απαντάει με ένα μήνυμα 200 OK το οποίο το Asterisk αφού στείλει μια επιβεβαίωση ότι το έλαβε το προωθεί στον χρήστη 2001. Από εδώ και πέρα η κλήση έχει εδραιωθεί και αρχίζει η επικοινωνία μεταξύ των δυο τερματικών. Βλέπουμε ότι η κίνηση των πακέτων φωνής γίνεται μεταξύ του τερματικού του 2001 (192.168.1.64) και μιας νέας IP (80.239.235.186) η οποία προφανώς αντιστοιχεί σε κάποιον Server που χρησιμοποιεί ο πάροχος για τον τερματισμό των κλήσεων στο pstn δίκτυο (Gateway). Επίσης βλέπουμε ότι η κίνηση των πακέτων φωνής (RTP) περνάει μέσα από το Asterisk και δεν πηγαίνει απευθείας από τον 2001 προς τον πάροχο, όπως έχουμε ορίσει στα configuration files που παρουσιάσαμε παραπάνω. Τέλος για τον τερματισμό της κλήσης ο 2001 στέλνει ένα μήνυμα BYE στο Asterisk το οποίο προωθείται προς τον πάροχο και μετά την λήψη του 200 OK η κλήση τερματίζεται.

### 5.3 Διασύνδεση δυο Asterisk Server με χρήση του πρωτοκόλλου IAX (Inter-Asterisk eXchange)

Η εφαρμογή αυτή αφορά την διασύνδεση δυο Asterisk τηλεφωνικών κέντρων μεταξύ τους έτσι ώστε να μπορούν οι χρήστες του ενός να καλούν τους χρήστες του άλλου. Η εφαρμογή αυτή είναι πολύ χρήσιμη σε περιπτώσεις όπου μία εταιρία έχει γραφεία σε δύο η περισσότερες γεωγραφικές περιοχές. Έτσι για παράδειγμα έστω ότι έχουμε μία εταιρία όπου έχει γραφεία σε Αθήνα και Μόναχο τα οποία συνδέονται με VPN μεταξύ τους. Αν και στα δύο γραφεία η εταιρία έχει εγκαταστήσει κέντρα Asterisk, τότε με την εφαρμογή αυτή οι χρήστες του ενός θα μπορούν να καλούν τους χρήστες του άλλου με μηδενικό κόστος χρησιμοποιώντας την υπάρχουσα VPN σύνδεση.

Στην συγκεκριμένη υλοποίηση χρησιμοποιήσαμε δύο Asterisk Servers οι οποίοι βρίσκονται σε διαφορετικά subnets. Ο ένας τον οποίο θα ονομάζουμε ServerAthens βρίσκεται στο subnet 10.0.0.254/24 με IP 10.0.0.16, ενώ ο άλλος τον οποίο θα ονομάζουμε ServerMunich βρίσκεται στο subnet 192.168.1.254/24 με IP 192.168.1.66. Η κίνηση μεταξύ αυτών των δύο δικτύων δρομολογείται μέσω ενός router, διαμόρφωση η οποία είναι ανάλογη με το παράδειγμα των δύο γραφείων που συνδέονται με VPN μεταξύ τους που αναφέραμε παραπάνω. Να αναφέρουμε ότι και οι δύο server μας έχουν στατικές IP και έτσι δεν απαιτούνται αιτήσεις εγγραφής (register) από τον έναν στον άλλον ώστε να γνωρίζει ο ένας την διεύθυνση του άλλου.

Για την δημιουργία της διασύνδεσης των δύο Asterisk πρέπει να δημιουργήσουμε εγγραφές στα αρχεία iax.conf και των δύο Server. Έχουμε δύο τύπους οντοτήτων τον IAX2 user τον IAX2 peer, όπου ο user χρησιμοποιείται για να πραγματοποιήσουμε εξερχόμενες κλήσεις προς το δεύτερο Asterisk ενώ ο peer για να δεχόμαστε κλήσεις σε αυτό. Επίσης είναι πολύ σημαντικό να έχουμε διαφορετικά νούμερα στα δύο κέντρα έτσι ώστε να μην γίνεται μπερδεμα της εσωτερικής με την εξωτερική δρομολόγηση κλήσεων. Στο συγκεκριμένο παράδειγμα ο ServerAthens έχει χρήστες που ξεκινούν με τον αριθμό 5 και περιέχουν άλλα 3 ψηφία (5XXX) ενώ ο ServerMunich έχει χρήστες που ξεκινούν με τον αριθμό 2 και περιέχουν άλλα 3 ψηφία (2XXX).

Ξεκινάμε με την διαμόρφωση που χρειάζεται στα δυο κέντρα Asterisk για κλήσεις από τον ServerAthens προς τον ServerMunich. Έχουμε λοιπόν τον ServerAthens σε ρόλο user ο οποίος όπως είπαμε δημιουργεί κλήσεις και τον ServerMunich σε ρόλο peer που δέχεται κλήσεις.

#### 1) ServerAthens

Στο `iax.conf` του ServerAthens πρέπει να δημιουργήσουμε την παρακάτω εγγραφή για την αυθεντικοποίηση του εαυτού του προς τον peer (ServerMunich) πριν αυτός δεχτεί την κλήση. Την εγγραφή αυτήν την ονομάσαμε `serverMunich`:

```
[serverMunich]
type=peer
auth=md5
user=admin
secret=hello1454
host=192.168.1.66
context=local
```

Στο παραπάνω `context` έχουμε ορίσει κάποιες βασικές παραμέτρους όπως είναι η IP του Asterisk στον οποίο δρομολογείται η κλήση, τον τρόπο αυθεντικοποίησης (`md5`) μας προς αυτόν, το `username` και το `password` καθώς και τον τύπο του `server` οποίος αφού δέχεται κλήσεις είναι `peer`.

Επίσης χρειαζόμαστε και την προσθήκη κάποιων `extension` στο αρχείο `extensions.conf` για στην δρομολόγηση των κλήσεων:

```
exten= _2XXX,1,Dial(IAX2/serverMunich/${EXTEN:0},30,r)
exten= _2XXX,2,Congestion
```

Βλέπουμε λοιπόν ότι όταν κάποιος από τους χρήστες του ServerAthens καλέσει κάποιον από τους χρήστες του ServerMunich (`2XXX`) η κλήση δρομολογείται μέσω του

trunk που έχουμε δημιουργήσει με την προσθήκη του context [serverMunich] στο αρχείο `iax.conf`.

## 2) ServerMunich

Στο `iax.conf` του ServerMunich πρέπει να δημιουργήσουμε την παρακάτω εγγραφή ώστε να επιτυγχάνεται η εξακρίβωση του user πριν δεχτεί την κλήση από αυτόν.

```
[ServerMunich]
type=user
auth=md5
secret=hello1454
context=local
```

Στο παραπάνω context έχουμε ορίσει κάποιες βασικές παραμέτρους όπως ο τρόπος αυθεντικοποίησης (md5) και το password για εξακρίβωση του user πριν δεχτούμε την κλήση από αυτόν. Τα στοιχεία αυτά πρέπει να είναι ίδια με αυτά που έχουμε ορίσει στο αντίστοιχο context στο `iax.conf` του ServerAthens.

Αντίστοιχα διαμορφώνουμε τα αρχεία για κλήσεις από τον ServerMunich προς τον ServerAthens. Έτσι τελικά για να έχουμε αμφίπλευρη δρομολόγηση κλήσεων τα δύο κέντρα πρέπει να έχουν τις παρακάτω εγγραφές στα αρχεία `iax.conf` και `extensions.conf`.

- ServerAthens (10.0.0.16)

iax.conf

```
[serverMunich]
type=peer
auth=md5
user=admin
secret=hello1454
```

```
host=192.168.1.66  
context=local
```

```
[ServerAthens]  
type=user  
auth=md5  
secret=hello1454  
context=local
```

### extensions.conf

```
[numberplan-custom-1]  
.....  
exten= _2XXX,1,Dial(IAX2/serverMunich/${EXTEN:0},30,r)  
exten= _2XXX,2,Congestion
```

- ServerMunich (192.168.1.66)

### iax.conf

```
[serverAthens]  
type=peer  
auth=md5  
user=admin  
secret=hello1454  
host=10.0.0.16  
context=local
```

```
[ServerMunich]  
type=user  
auth=md5
```

```
secret=hello1454  
context=local
```

### extensions.conf

```
[numberplan-custom-2]  
.....  
exten=_5XXX,1,Dial(IAX2/serverAthens/${EXTEN:0},30,r)  
exten=_5XXX,2,Congestion
```

Στην συνέχεια για την επιβεβαίωση της σωστής λειτουργίας της παραπάνω διαμόρφωσης των δύο Asterisk θα παρουσιάσουμε ένα παράδειγμα κλήσης από τον χρήστη 5001 του ServerAthens προς τον χρήστη 2001 του ServerMunich. Πήραμε traces πάνω στις κάρτες δικτύου και των δύο server μιας και δεν υπήρχε η δυνατότητα για traces πάνω στον router. Στα δύο παρακάτω σχήματα παρουσιάζουμε το trace από τον ServerAthens (Σχήμα 5.9) και από τον ServerMunich (Σχήμα 5.10) αντίστοιχα.

Time	10.0.0.16	10.0.0.50	192.168.1.66	Comment
5,024	(5060)	Request: NOTIFY sip	(5060)	SIP: Request: NOTIFY sip:5001@10.0.0.50:5060;rinstance=4c46c9a618580883
5,026	(5060)	Status: 200 OK	(5060)	SIP: Status: 200 OK
5,221	(5060)	Request: INVITE sip	(5060)	SIP/SDP: Request: INVITE sip:2001@10.0.0.16;transport=UDP, with session description
5,222	(5060)	Status: 407 Proxy A	(5060)	SIP: Status: 407 Proxy Authentication Required
5,225	(5060)	Request: ACK sip 20	(5060)	SIP: Request: ACK sip:2001@10.0.0.16;transport=UDP
5,227	(5060)	Request: INVITE sip	(5060)	SIP/SDP: Request: INVITE sip:2001@10.0.0.16;transport=UDP, with session description
5,228	(5060)	Status: 100 Trying	(5060)	SIP: Status: 100 Trying
5,233	(5060)	Status: 180 Ringing	(5060)	SIP: Status: 180 Ringing
5,236	(4569)	IAX_source call# 1	(4569)	IAX2: IAX, source call# 1, timestamp 4ms NEW
5,238	(4569)	IAX_source call# 1	(4569)	IAX2: IAX, source call# 1, timestamp 4ms ACCEPT
5,239	(4569)	IAX_source call# 1	(4569)	IAX2: IAX, source call# 1, timestamp 4ms ACK
5,301	(4569)	Control_source cal	(4569)	IAX2: Control, source call# 1, timestamp 7ms RINGING
5,302	(4569)	IAX_source call# 1	(4569)	IAX2: IAX, source call# 1, timestamp 7ms ACK
11,672	(4569)	Control_source cal	(4569)	IAX2: Control, source call# 1, timestamp 6439ms stop sounds
11,672	(4569)	Control_source cal	(4569)	IAX2: Control, source call# 1, timestamp 6442ms ANSWER
11,672	(4569)	IAX_source call# 1	(4569)	IAX2: IAX, source call# 1, timestamp 6439ms ACK
11,673	(4569)	IAX_source call# 1	(4569)	IAX2: IAX, source call# 1, timestamp 6442ms ACK
11,675	(5060)	Status: 200 OK, wit	(5060)	SIP/SDP: Status: 200 OK, with session description
11,692	(5060)	Request: ACK sip 20	(5060)	SIP: Request: ACK sip:2001@10.0.0.16
11,695	(4569)	Voice_source call#	(4569)	IAX2: Voice, source call# 1, timestamp 6463ms, GSM compression
11,696	(4569)	IAX_source call# 1	(4569)	IAX2: IAX, source call# 1, timestamp 6463ms ACK
11,697	(14100)	PT=ITU-T G.711 PCMA	(14100)	RTP: PT=ITU-T G.711 PCMA, SSRC=0x7B2F00C0, Seq=4955, Time=160
11,711	(14100)	PT=ITU-T G.711 PCMA	(14100)	RTP: PT=ITU-T G.711 PCMA, SSRC=0x1F3A30E0, Seq=18018, Time=3718449584
11,712	(4569)	Voice_source call#	(4569)	IAX2: Voice, source call# 1, timestamp 6483ms, GSM compression
11,713	(4569)	IAX_source call# 1	(4569)	IAX2: IAX, source call# 1, timestamp 6483ms ACK
11,714	(4569)	Mini packet_source	(4569)	IAX2: Mini packet, source call# 1, timestamp 6483ms, GSM compression
11,715	(14100)	PT=ITU-T G.711 PCMA	(14100)	RTP: PT=ITU-T G.711 PCMA, SSRC=0x7B2F00C0, Seq=4956, Time=320
11,731	(14100)	PT=ITU-T G.711 PCMA	(14100)	RTP: PT=ITU-T G.711 PCMA, SSRC=0x1F3A30E0, Seq=18019, Time=3718449744
11,734	(4569)	Mini packet_source	(4569)	IAX2: Mini packet, source call# 1, timestamp 6503ms, GSM compression
20,535	(4569)	IAX_source call# 1	(4569)	IAX2: IAX, source call# 1, timestamp 15303ms HANGUP
20,535	(4569)	IAX_source call# 1	(4569)	IAX2: IAX, source call# 1, timestamp 15303ms ACK
20,536	(5060)	Request: BYE sip 50	(5060)	SIP: Request: BYE sip:5001@10.0.0.50:5060
20,715	(5060)	Status: 200 OK	(5060)	SIP: Status: 200 OK

Σχήμα 5.9: Ethereal trace πάνω στην κάρτα του ServerAthens για κλήση από τον χρήστη 5001 του ServerAthens προς τον χρήστη 2001 του ServerMunich

Time	10.0.0.16	192.168.1.66	192.168.1.64	Comment
12,780	(4569) IAX, source call# 1	(4569)	(4569)	IAX2: IAX, source call# 1, timestamp 4ms NEW
12,781	(4569) IAX, source call# 1	(4569)	(4569)	IAX2: IAX, source call# 1, timestamp 4ms ACCEPT
12,783	(4569) IAX, source call# 1	(4569)	(4569)	IAX2: IAX, source call# 1, timestamp 4ms ACK
12,784	(5060) Request: INVITE sip	(5060)	(5060)	SIP/SDP: Request: INVITE sip:2001@192.168.1.64:5060;instanc
12,840	(5060) Status: 180 Ringing	(5060)	(5060)	SIP: Status: 180 Ringing
12,844	(4569) Control, source cal	(4569)	(4569)	IAX2: Control, source call# 1, timestamp 7ms RINGING
12,845	(4569) IAX, source call# 1	(4569)	(4569)	IAX2: IAX, source call# 1, timestamp 7ms ACK
19,213	(5060) Status: 200 OK, wit	(5060)	(5060)	SIP/SDP: Status: 200 OK, with session description
19,213	(5060) Request: ACK sip:20	(5060)	(5060)	SIP: Request: ACK sip:2001@192.168.1.64:5060;instanc=4c6
19,214	(4569) Control, source cal	(4569)	(4569)	IAX2: Control, source call# 1, timestamp 6439ms stop sounds
19,214	(4569) Control, source cal	(4569)	(4569)	IAX2: Control, source call# 1, timestamp 6442ms ANSWER
19,216	(4569) IAX, source call# 1	(4569)	(4569)	IAX2: IAX, source call# 1, timestamp 6439ms ACK
19,216	(4569) IAX, source call# 1	(4569)	(4569)	IAX2: IAX, source call# 1, timestamp 6442ms ACK
19,238	(14530) PT=ITU-T G.711 PCMU	(14530)	(3000)	RTP: PT=ITU-T G.711 PCMU, SSRC=0xD294EB9A, Seq=48290, 1
19,238	(4569) Voice, source call#	(4569)	(4569)	IAX2: Voice, source call# 1, timestamp 6463ms, GSM compress
19,240	(4569) IAX, source call# 1	(4569)	(4569)	IAX2: IAX, source call# 1, timestamp 6463ms ACK
19,255	(4569) Voice, source call#	(4569)	(4569)	IAX2: Voice, source call# 1, timestamp 6483ms, GSM compress
19,256	(4569) IAX, source call# 1	(4569)	(4569)	IAX2: IAX, source call# 1, timestamp 6483ms ACK
19,257	(14530) PT=ITU-T G.711 PCMU	(14530)	(3000)	RTP: PT=ITU-T G.711 PCMU, SSRC=0x1848E2F3, Seq=62589, Ti
19,257	(14530) PT=ITU-T G.711 PCMU	(14530)	(3000)	RTP: PT=ITU-T G.711 PCMU, SSRC=0xD294EB9A, Seq=48291, 1
19,257	(4569) Mini packet, source	(4569)	(4569)	IAX2: Mini packet, source call# 1, timestamp 6483ms, GSM comp
19,277	(14530) PT=ITU-T G.711 PCMU	(14530)	(3000)	RTP: PT=ITU-T G.711 PCMU, SSRC=0xD294EB9A, Seq=48292, 1
19,277	(4569) Mini packet, source	(4569)	(4569)	IAX2: Mini packet, source call# 1, timestamp 6503ms, GSM comp
19,279	(4569) Mini packet, source	(4569)	(4569)	IAX2: Mini packet, source call# 1, timestamp 6503ms, GSM comp
19,279	(14530) PT=ITU-T G.711 PCMU	(14530)	(3000)	RTP: PT=ITU-T G.711 PCMU, SSRC=0x1848E2F3, Seq=62590, Ti
28,075	(5060) Request: BYE sip:as	(5060)	(5060)	SIP: Request: BYE sip:asterisk@192.168.1.66
28,075	(5060) Status: 200 OK	(5060)	(5060)	SIP: Status: 200 OK
28,077	(4569) IAX, source call# 1	(4569)	(4569)	IAX2: IAX, source call# 1, timestamp 15303ms HANGUP
28,078	(4569) IAX, source call# 1	(4569)	(4569)	IAX2: IAX, source call# 1, timestamp 15303ms ACK

Σχήμα 5.10: Ethereal trace πάνω στην κάρτα του ServerMunich για κλήση από τον χρήστη 5001 του ServerAthens προς τον χρήστη 2001 του ServerMunich

Βλέπουμε λοιπόν ότι η κλήση ξεκινάει από τον χρήστη 5001 (10.0.0.50) με το μήνυμα invite (χρησιμοποιώντας το πρωτόκολλο sip) που στέλνει προς το asterisk ServerAthens (10.0.0.16), ο οποίος αφού αυθεντικοποιήσει τον χρήστη 5001 στέλνει ένα μήνυμα New (χρησιμοποιώντας το πρωτόκολλο iax2) προς τον ServerMunich (192.168.1.66). Ο ServerMunich με την σειρά του απαντάει με ένα iax2 μήνυμα Accept και αφού δεχθεί την επιβεβαίωση από τον ServerAthens στέλνει ένα sip μήνυμα invite προς τον χρήστη 2001 (192.168.1.64). Ο 2001 απαντάει με ένα sip μήνυμα ringing προς το asterisk το οποίο προωθεί ένα iax2 μήνυμα ringing προς τον ServerAthens και αυτός με την σειρά του προωθεί ένα sip μήνυμα ringing στον 5001. Όταν ο 2001 απαντάει ακολουθείται η



ίδια διαδικασία με πριν όπου ο 2001 στέλνει ένα sip μήνυμα 200OK το οποίο προωθείται μέσω του iax2 μηνύματος Answer στον serverAthens και αυτός προωθεί ένα sip μήνυμα 200OK στον 5001. Μετά από αυτό η κλήση έχει εγκαθιδρυθεί και οι δύο χρήστες μιλάνε μεταξύ τους όπου οι χρήστες στέλνουν RTP μηνύματα φωνής στο asterisk το οποίο ανήκουν και αυτό τα μετατρέπει σε iax2 και τα προωθεί στο άλλο asterisk για να τα μετατρέψει με την σειρά του πάλι σε RTP και να τα στείλει στο χρήστη του. Η κλήση τερματίζεται όταν ο 2001 στέλνει ένα sip μήνυμα bye στον ServerMunich και αυτός το προωθεί με ένα iax2 μήνυμα hungup στον ServerAthens, για να προωθήσει με την σειρά του ένα sip μήνυμα bye στον 5001. Να σημειώσουμε ότι έχουμε διαγράψει τα περισσότερα μηνύματα φωνής και έχουμε αφήσει ορισμένα ενδεικτικά για λόγους περιορισμού του μεγέθους των εικόνων. Γενικά βλέπουμε ότι τα κέντρα Asterisk επικοινωνούν με τα πρωτόκολλα SIP και RTP με τους χρήστες τους (αφού έχουμε ορίσει αυτοί να χρησιμοποιούν το πρωτόκολλο SIP) και μετατρέπουν τα μηνύματα αυτά σε iax2 μηνύματα για να επικοινωνήσουν ο ένας με τον άλλον.

Στα παρακάτω σχήματα (Σχήμα 5.11 και 5.12) παρουσιάζεται η κονσόλα των δύο asterisk server κατά την διάρκεια της κλήσης από τον χρήστη 5001 του ServerAthens προς τον χρήστη 2001 του ServerMunich.

```
*CLI>          Accepting UNAUTHENTICATED call from 10.0.0.16:
> requested format = gsm,
> requested prefs = (),
> actual format = gsm,
> host prefs = (),
> priority = mine
Executing [2001@default:1] Macro("IAX2/SecServerAthens-1", "stdexten!2001
!SIP/2001&IAX2/2001") in new stack
Executing [s@macro-stdexten:1] Dial("IAX2/SecServerAthens-1", "SIP/2001&I
AX2/2001!20") in new stack
Called 2001
SIP/2001-00215ec0 is ringing
SIP/2001-00215ec0 answered IAX2/SecServerAthens-1
Spawn extension (macro-stdexten, s, 1) exited non-zero on 'IAX2/SecServerAt
hens-1' in macro 'stdexten'
Spawn extension (macro-stdexten, s, 1) exited non-zero on 'IAX2/SecServerAt
hens-1'
Hungup 'IAX2/SecServerAthens-1'
```

Σχήμα 5.11: Κονσόλα του asterisk ServerMunich για κλήση από τον χρήστη 5001 του ServerAthens προς τον χρήστη 2001 του ServerMunich

```
*CLI> Executing [2001@numberplan-custom-1:1] Dial("SIP/5001-00205d30", "IAX2/serverMunich/2001i30i") in new stack
Called serverMunich/2001
Call accepted by 192.168.1.66 (format gsm)
Format for call is gsm
IAX2/serverMunich-1 is ringing
IAX2/serverMunich-1 answered SIP/5001-00205d30
Hangup 'IAX2/serverMunich-1'
Spawn extension (numberplan-custom-1, 2001, 1) exited non-zero on 'SIP/5001-00205d30'
```

Σχήμα 5.12: Κονσόλα του asterisk ServerAthens για κλήση από τον χρήστη 5001 του ServerAthens προς τον χρήστη 2001 του ServerMunich

## 5.4 Εφαρμογή για την δημιουργία «δωματίων» τηλεδιάσκεψων

Το Asterisk μέσω της εφαρμογής τηλεδιάσκεψης MeetMe, επιτρέπει σε πολλούς χρήστες να μπορούν να συνομιλήσουν μεταξύ τους ταυτόχρονα. Οι δυνατότητες τηλεδιάσκεψης υλοποιούνται μέσω των τριών εφαρμογών MeetMe(), MeetMeAdmin() και MeetMeCount() στο dialplan. Οι βασικές λειτουργίες των εφαρμογών τηλεδιάσκεψης περιλαμβάνουν:

- Τη δυνατότητα τηλεδιάσκεψης ανεξαρτήτως του χρησιμοποιούμενου πρωτοκόλλου επικοινωνίας.
- Τη δυνατότητα δημιουργίας προστατευμένων με κωδικό πρόσβασης δωματίων τηλεδιάσκεψης.
- Τη διαχείριση μίας τηλεδιάσκεψης (σίγαση, κλείδωμα, εκδίωξη συμμετεχόντων) από έναν ή περισσότερους χρήστες-διαχειριστές.
- Την επιλογή σίγασης όλων των συμμετεχόντων εκτός από έναν (χρήσιμο για ανακοινώσεις, κλπ)
- Τη στατική ή δυναμική δημιουργία δωματίων τηλεδιάσκεψης.

Η διαμόρφωση της εφαρμογής τηλεδιάσκεψης μπορεί να γίνει είτε από το γραφικό περιβάλλον είτε από την απευθείας διαμόρφωση των αρχείων. Και στις δύο περιπτώσεις τελικά για την σωστή λειτουργία της εφαρμογής πρέπει να υπάρχει μια εγγραφή στο αρχείο `meetme.conf` η οποία να καθορίζει ένα δωμάτιο τηλεδιάσκεψης. Επίσης πρέπει να υπάρχει και ένα `extension` στο αρχείο `extensions.conf` που να δρομολογεί την κλήση στο δωμάτιο αυτό. Για να δημιουργήσουμε ένα νέο δωμάτιο τηλεδιάσκεψης πρέπει στο `context [rooms]` του αρχείου `meetme.conf` προσθέσουμε μία γραμμή που θα έχει τη μορφή:

```
conf=νούμερο δωματίου, κωδικός χρήστη, κωδικός διαχειριστή
```

Για παράδειγμα, παρακάτω δημιουργείται το δωμάτιο 3000, με κωδικό χρήστη 1234 και κωδικό διαχειριστή 123456.

```
[rooms]
conf=3000,1234,123456
```

#### **5.4.1 Διαμόρφωση της εφαρμογής τηλεδιάσκεψης με απευθείας διαμόρφωση των αρχείων**

Όπως αναφέραμε προηγουμένως για την σωστή λειτουργία της εφαρμογής χρειάζεται η δημιουργία ενός δωματίου τηλεδιάσκεψης και του αντίστοιχου `extension`. Ξεκινάμε με την δημιουργία του `extension` με την χρήση της εφαρμογής `MeetMe()` στο `dialplan`. Προκειμένου λοιπόν οι κλήσεις να μπορούν να δρομολογηθούν στο δωμάτιο 3000 πρέπει να δημιουργήσουμε το παρακάτω `extension` στο `context [numberplan-custom-2]`.

```
exten =3000,1,MeetMe(3000)
exten =3000,n,Hangup()
```

Όταν κάποιος χρήστης με πρόσβαση στο `context [numberplan-custom-2]` πληκτρολογήσει 3000 τότε θα εκτελεστεί η εφαρμογή `MeetMe()`, η οποία θα ζητήσει από

τον χρήστη τον κωδικό πρόσβασης του δωματίου (όπως ορίστηκε στο αρχείο `meetme.conf`) και θα επιχειρήσει να τον εισάγει στο δωμάτιο τηλεδιάσκεψης 3000.

Η σύνταξη της εφαρμογής `MeetMe()` είναι η εξής:

`MeetMe` (νούμερο δωματίου, επιλογές, κωδικός)

Όπου νούμερο δωματίου, είναι το νούμερο που έχουμε ορίσει στο αρχείο `meetme.conf`. Αν δεν καθοριστεί κάποιο νούμερο δωματίου τότε η εφαρμογή θα ζητήσει από το χρήστη να το εισάγει. Αν δοθεί σαν όρισμα στην `MeetMe()` κωδικός ο οποίος ανταποκρίνεται στον κωδικό χρήστη/διαχειριστή του `meetme.conf` τότε ο καλών θα εισέλθει απευθείας στο δωμάτιο με τα αντίστοιχα προνόμια χωρίς να του ζητηθεί να εισάγει τον κωδικό. Η παράμετρος επιλογές καθορίζει τις υπηρεσίες και επιλογές που θα είναι διαθέσιμες σε αυτό το δωμάτιο τηλεδιάσκεψης. Οι βασικές διαθέσιμες επιλογές είναι οι εξής:

- a: Ορισμός εισόδου διαχειριστή
- A: Χαρακτηρισμός του εισαχθέντος χρήστη ως ειδικού συμμετέχοντα (βλ. *w* και *x*).
- c: Ανακοίνωση αριθμού συμμετεχόντων στον εισαχθέντα χρήστη.
- d: Δυναμική δημιουργία νέου δωματίου τηλεδιάσκεψης.
- D: Δυναμική δημιουργία νέου δωματίου τηλεδιάσκεψης και απαίτηση από το χρήστη να ορίσει κωδικό εισόδου (Αν δεν επιθυμείται χρήση κωδικού ο χρήστης πρέπει να εισάγει “#” κατά την προτροπή).
- e: Επιλογή κενού δωματίου.
- E: Επιλογή κενού δωματίου που δεν απαιτεί κωδικό.
- i: Ανακοίνωση εισόδου/εξόδου συμμετεχόντων με πληροφορίες.
- I: Ανακοίνωση εισόδου/εξόδου συμμετεχόντων χωρίς πληροφορίες.
- m: Σύνδεση χωρίς δυνατότητα ομιλίας.
- M: Αναπαραγωγή μουσικής στην αναμονή αν υπάρχει μόνο ένας συμμετέχοντας στο δωμάτιο.

- ο: Βελτιστοποιεί την ποιότητα των συνομιλιών και μειώνει τις απαιτήσεις επεξεργαστικής ισχύς, σιωπώντας τους χρήστες που δε μιλάνε.
- p: Δυνατότητα αποσύνδεσης του συμμετέχοντα πατώντας το πλήκτρο “#”.
- P: Απαιτήση κωδικού, ακόμα και αν παρέχεται σαν όρισμα στην εφαρμογή.
- q: Δεν αναπαράγει τόνους ειδοποίησης κατά τη σύνδεση και αποσύνδεση των χρηστών.
- s: Αναπαραγωγή του μενού επιλογών με πάτημα του πλήκτρου “\*”.
- t: Σύνδεση χωρίς δυνατότητα ακρόασης.
- T: Εντοπισμός ομιλούντων μελών και εμφάνιση στην κονσόλα.
- w: Αναμονή έως την είσοδο του “χαρακτηρισμένου” χρήστη και αναπαραγωγή μουσικής στους άλλους συμμετέχοντες κατά την αναμονή.
- x: Κλείσιμο του δωματίου με την έξοδο του “χαρακτηρισμένου” χρήστη.
- X: Αποσύνδεση των συμμετεχόντων με την πληκτρολόγηση μίας μονοψήφιας extension του context που ορίζεται από την ειδική μεταβλητή  $\{MEETME\_EXIT\_CONTEXT\}$  ή του τρέχοντος context αν η μεταβλητή δεν έχει ορισθεί. Η επιλογή αυτή δεν μπορεί να συνδυαστεί με τις p και s.
- l: Δεν αναπαράγεται το μήνυμα “*You are the only person in this conference*” όταν συνδέεται ο πρώτος χρήστης.

Στην συνέχεια (Σχήμα 5.13) παρουσιάζουμε ένα παράδειγμα για την επιβεβαίωση της σωστής λειτουργίας του δωματίου 3000 που δημιουργήσαμε παραπάνω. Σε αυτό το παράδειγμα ο χρήστης 2001 καλεί τον αριθμό 3000 και αφού ακούσει το μήνυμα για την εισαγωγή του κωδικού τον πληκτρολογεί. Μετά την επιτυχή εισαγωγή του κωδικού ο χρήστης 2001 ακούει το μήνυμα ότι είναι ο μοναδικός χρήστης σε αυτήν την τηλεδιάσκεψη και μπαίνει σε αναμονή μέχρι να εισέλθει κάποιος άλλος χρήστης.

```
*CLI> Executing [3000@numberplan-custom-2:1] MeetMe("SIP/2001-08227db8",
"3000") in new stack
Parsing '/etc/asterisk/meetme.conf': Found
Created MeetMe conference 1023 for conference '3000'
<SIP/2001-08227db8> Playing 'conf-getpin' (language 'en')
<SIP/2001-08227db8> Playing 'conf-onlyperson' (language 'en')
Hangup 'Zap/pseudo-579674311'
Spawn extension (numberplan-custom-2, 3000, 1) exited non-zero on 'SIP/2001-08227db8'
```

Σχήμα 5.13: Κονσόλα του asterisk για κλήση του χρήστη 2001 για το δωμάτιο τηλεδιάσκεψης 3000

Στην συνέχεια θα δημιουργήσουμε το extension 3010 το οποίο θα καλεί την εφαρμογή MeetMe() χωρίς αριθμό δωματίου, κάνοντας χρήση της επιλογής D. Έτσι θα έχουμε δυναμική δημιουργία δωματίου τηλεδιάσκεψης μέσω του dialplan.

```
exten = 3010,1,MeetMe(DcsM)
```

```
exten = 3010,n,Hangup()
```

Βλέπουμε ότι με τα παραπάνω extension δεν οδηγούμε την κλήση σε κάποιο συγκεκριμένο δωμάτιο αλλά με την επιλογή D που έχουμε προσθέσει δημιουργούμε δυναμικά ένα δωμάτιο τηλεδιάσκεψης. Επίσης έχουμε προσθέσει τις επιλογές csM, όπου με την επιλογή c ανακοινώνεται ο αριθμός των συμμετεχόντων στον εισαχθέντα χρήστη, με την επιλογή s μπορούμε να επιλέξουμε την αναπαραγωγή του μενού επιλογών με πάτημα του πλήκτρου "\*", ενώ με την επιλογή M έχουμε αναπαραγωγή μουσικής στην αναμονή αν υπάρχει μόνο ένας συμμετέχοντας στο δωμάτιο.

Επειδή δεν έχει οριστεί αριθμός δωματίου θα ζητηθεί από το χρήστη να τον εισάγει προκειμένου να επιχειρηθεί η σύνδεση σε κάποιο δημιουργημένο δωμάτιο. Αν βρεθεί το δωμάτιο και δεν έχει κωδικό, τότε ο χρήστης εισάγεται σε αυτό. Αν βρεθεί και έχει κωδικό, τότε ζητείται από το χρήστη να τον εισάγει. Αν το δωμάτιο δε βρεθεί, τότε θα ζητηθεί και πάλι από το χρήστη να εισάγει κωδικό (ή "#" για κενό), αλλά αυτήν τη φορά τα στοιχεία θα χρησιμοποιηθούν για να δημιουργηθεί το δωμάτιο τηλεδιάσκεψης. Τα δωμάτια που δημιουργούνται δυναμικά ισχύουν για όσο διάστημα υπάρχουν χρήστες μέσα τους.

Στην συνέχεια (Σχήμα 5.14) παρουσιάζουμε ένα παράδειγμα δυναμικής δημιουργίας ενός δωματίου τηλεδιάσκεψης. Στο συγκεκριμένο παράδειγμα ο χρήστης 2001 καλεί το extension 3010 και αφού ακούσει το μήνυμα για την εισαγωγή ενός αριθμού εισάγει τον αριθμό 400. Μετά από αυτό του ζητείτε να εισάγει και ένα pin για το δωμάτιο τηλεδιάσκεψης και εισάγει το 123456. Τέλος ακούει το μήνυμα ότι είναι ο μόνος χρήστης σε αυτό το δωμάτιο τηλεδιάσκεψης και ακούει music on hold. Από αυτό σημείο και μετά όποιος άλλος χρήστης θέλει να μπει σε αυτό το δωμάτιο τηλεδιάσκεψης καλεί το extension 3010 και εισάγει τον αριθμό του δωματίου τηλεδιάσκεψης καθώς και τον αριθμό pin.

```
Executing [3010@numberplan-custom-2:1] MeetMe("SIP/2001-08229640", "iDcsM") in new stack
<SIP/2001-08229640> Playing 'conf-getconfno' (language 'en')
<SIP/2001-08229640> Playing 'conf-getpin' (language 'en')
Created MeetMe conference 1023 for conference '400'
<SIP/2001-08229640> Playing 'conf-onlyperson' (language 'en')
Started music on hold, class 'default', on SIP/2001-08229640
Stopped music on hold on SIP/2001-08229640
Hungup 'Zap/pseudo-1261374379'
Spawn extension (numberplan-custom-2, 3010, 1) exited non-zero on 'SIP/2001-08229640'
```

Σχήμα 5.14: Κονσόλα του Asterisk κατά την διάρκεια της δυναμικής δημιουργίας ενός δωματίου τηλεδιάσκεψης

#### 5.4.2 Διαμόρφωση της εφαρμογής τηλεδιάσκεψης με χρήση του γραφικού περιβάλλοντος

Σε αυτήν την παράγραφο θα δούμε πως μπορούμε να δημιουργήσουμε δωμάτια τηλεδιάσκεψης με την χρήση του γραφικού περιβάλλοντος AsteriskGui. Να αναφέρουμε εδώ ότι με αυτόν τον τρόπο δεν είναι διαθέσιμες όλες οι επιλογές που αναφέραμε παραπάνω καθώς επίσης δεν υπάρχει δυνατότητα για δυναμική δημιουργία δωματίων τηλεδιάσκεψης.

Για να δημιουργήσουμε ένα δωμάτιο τηλεδιάσκεψης επιλέγουμε από το μενού την επιλογή conferencing και στην συνέχεια new. Στην συνέχεια πρέπει να συμπληρώσουμε τα στοιχεία που φαίνονται στο παρακάτω σχήμα (Σχήμα 5.15).



**Conference Bridge Extensions Configuration**

Conference Bridges:

- 6001 -- User\_6001
- 6050 -- Check Voicemail
- 7000 -- Voice Menu
- New Entry**

Extension:

**Password Settings:**

PIN Code:

Admin PIN Code:

**Conference Room Options:**

- Play hold music for first caller
- Enable caller menu
- Announce callers
- Quiet Mode
- Wait for marked user
- Set marked user

**Σχήμα 5.15:** Δημιουργία δωματίου τηλεδιάσκεψης μέσω του γραφικού περιβάλλοντος AsteriskGui

Βλέπουμε ότι πρέπει να εισάγουμε έναν αριθμό δωματίου, το pin αυτού του δωματίου καθώς και το pin του διαχειριστή αυτού του δωματίου. Επίσης υπάρχουν διαθέσιμες και μερικές από τις επιλογές που είδαμε στην προηγούμενη παράγραφο, όπως η αναπαραγωγή music on hold στον πρώτο χρήστη, η δημιουργία χαρακτηρισμένου χρήστη κτλ. Με τον τρόπο αυτό δημιουργούνται αυτομάτως εγγραφές στα αρχεία `meetme.conf` και `extensions.conf` του Asterisk. Έτσι εάν για παράδειγμα δημιουργήσουμε το δωμάτιο 3300 με pin: 123 και pin διαχειριστή: 123456 και ενεργοποιημένες τις επιλογές `play Music on hold for the first caller` και `Announce callers`, στα δύο αυτά αρχεία θα δημιουργηθούν οι παρακάτω εγγραφές.

`Meetme.conf: conf=3300,123,123456`

`Extensions.conf: exten=3300,1,MeetMe(${EXTEN}|MI)`



## 5.5 Call Park

Πρόκειται για μια εφαρμογή η οποία αφορά την διαχείριση των εισερχομένων κλήσεων, ώστε αυτές να μπορούν να εξυπηρετηθούν από τον κατάλληλο τηλεφωνικό εκπρόσωπο. Συγκεκριμένα μπορούμε μία κλήση, αφού έχει απαντηθεί από κάποιον τηλεφωνικό εκπρόσωπο, να την «παρκάρουμε» σε κάποια «θέση» η οποία αντιστοιχεί σε κάποιον αριθμό. Έτσι στην συνέχεια μπορεί κάποιος άλλος τηλεφωνικός εκπρόσωπος από το κατάλληλο τμήμα να πάρει αυτήν την κλήση καλώντας τον αριθμό της θέσης στην οποία την έχουμε παρκάρει. Παρακάτω θα δούμε πως μπορούμε να διαμορφώσουμε αυτήν την εφαρμογή στο Asterisk τόσο χειροκίνητα όσο και από το γραφικό περιβάλλον.

### 5.5.1 Διαμόρφωση της εφαρμογής call park με απευθείας διαμόρφωση των αρχείων

Για την διαμόρφωση αυτής της εφαρμογής πρέπει αρχικά στο αρχείο features.conf να ορίσουμε το extension που θα καλούμε για να παρκάρουμε μια κλήση καθώς επίσης και ποιες θέσεις θα είναι διαθέσιμες. Έτσι προσθέτουμε τις παρακάτω εγγραφές οι οποίες ορίζουν το extension 700 και τις διαθέσιμες θέσεις από την 701 έως την 720.

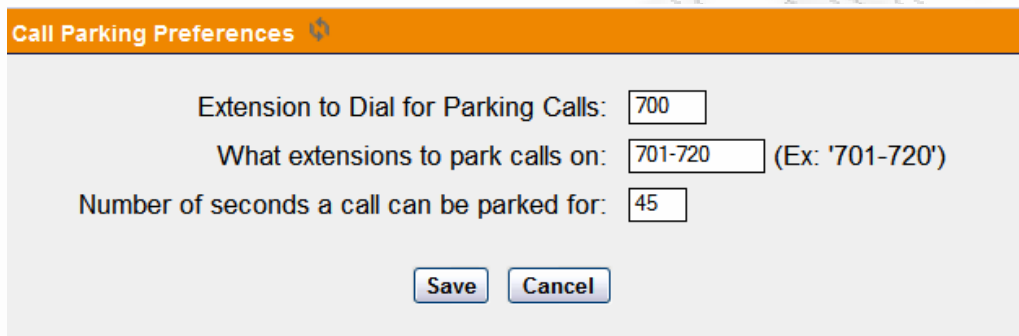
```
[general]
parkext = 700
parkpos = 701-720
```

Επίσης στο αρχείο extensions.conf βάζουμε την παρακάτω εγγραφή στο context του οποίου οι χρήστες θέλουμε να χρησιμοποιήσουν αυτήν την εφαρμογή:

```
[numberplan-custom-2]
include = parkedcalls
```

### 5.5.2 Διαμόρφωση της εφαρμογής call park με χρήση του AsteriskGui

Για την διαμόρφωση της εφαρμογής από το γραφικό περιβάλλον πηγαίνουμε στο μενού call parking (Σχήμα 5.16), συμπληρώνουμε το extension που επιθυμούμε για το call park καθώς και τις διαθέσιμες θέσεις. Επίσης μπορούμε να ορίσουμε και τον χρόνο για τον οποίο μια κλήση μπορεί να είναι παρκαρισμένη.



Call Parking Preferences

Extension to Dial for Parking Calls:

What extensions to park calls on:  (Ex: '701-720')

Number of seconds a call can be parked for:

Σχήμα 5.16: Διαμόρφωση της εφαρμογής Call Park μέσω του γραφικού περιβάλλοντος AsteriskGui

### 5.5.3 Δοκιμή της εφαρμογής Call Park

Στην συνέχεια θα παρουσιάσουμε ένα παράδειγμα της λειτουργίας της εφαρμογής call park για να επιβεβαιώσουμε την σωστή λειτουργία της εφαρμογής. Συγκεκριμένα στο παρακάτω παράδειγμα (Σχήμα 5.17) ο χρήστης 2001 του Asterisk είναι συνδεδεμένος με κάποιον pstn χρήστη (210XXXXXXX) και «παρκάρει» τον χρήστη αυτόν στην θέση 701. Για να παρκάρει την κλήση ο χρήστης 2001 την κάνει transfer στον αριθμό 700. Στην συνέχεια ο χρήστης 2002 καλεί το 701 και παίρνει την παρκαρισμένη κλήση και έτσι συνδέεται με τον pstn χρήστη. Βλέπουμε ότι κατά την διάρκεια που ο pstn χρήστης είναι παρκαρισμένος ακούει music on hold. Επίσης την ώρα που ο χρήστης 2001 παρκάρει την κλήση με τον pstn χρήστη, το Asterisk αναπαράγει τον αριθμό της θέσης που πρόκειται να παρκάρει αυτήν την κλήση ώστε ο 2001 ή κάποιος άλλος χρήστης να γνωρίζει το νούμερο που πρέπει να καλέσει για να πάρει την παρκαρισμένη κλήση.

```
Called voipdiscount2/0030210-
SIP/voipdiscount2-0821c608 is making progress passing it to SIP/2001-0822
25f8
SIP/voipdiscount2-0821c608 answered SIP/2001-082225f8
Started music on hold, class 'default', on SIP/voipdiscount2-0821c608
Spawn extension (numberplan-custom-2, 210, 5) exited non-zero on 'SI
PPeer/SIP/2001-082225f8<ZOMBIE>'
Started music on hold, class 'default', on SIP/voipdiscount2-0821c608
Parked SIP/voipdiscount2-0821c608 on 701@parkedcalls. Will timeout back to
extension [outgoing], 1 in 45 seconds
<SIP/2001-082225f8> Playing 'digits/7' (language 'en')
<SIP/2001-082225f8> Playing 'digits/0' (language 'en')
<SIP/2001-082225f8> Playing 'digits/1' (language 'en')
Added extension '701' priority 1 to parkedcalls
Executing [701@numberplan-custom-2:1] ParkedCall("SIP/2002-082225f8", "70
1") in new stack
Stopped music on hold on SIP/voipdiscount2-0821c608
Channel SIP/2002-082225f8 connected to parked call 701
Spawn extension (numberplan-custom-2, 701, 1) exited non-zero on 'SIP/2002-
082225f8'
```

Σχήμα 5.17: Παράδειγμα λειτουργίας της εφαρμογής call park

Στην συνέχεια (Σχήμα 5.18) έχουμε ένα παράδειγμα παρόμοιο με το παραπάνω όπου ο χρήστης 2001 παρκάρει τον pstn χρήστη αλλά μετά από 45 δευτερόλεπτα, αφού έχει ξεπεραστεί ο χρόνος που έχουμε ορίσει ότι μια κλήση μπορεί να μείνει παρκαρισμένη, η κλήση γυρνάει πίσω στον 2001 και αυτός την απαντάει ώστε να συνδεθεί πάλι με τον pstn χρήστη.

```
Called voipdiscount2/0030210:
SIP/voipdiscount2-08224fe0 is making progress passing it to SIP/2001-0821
f9e8
SIP/voipdiscount2-08224fe0 answered SIP/2001-0821f9e8
Started music on hold, class 'default', on SIP/voipdiscount2-08224fe0
Spawn extension (numberplan-custom-2, 210: 5) exited non-zero on 'SI
PPeer/SIP/2001-0821f9e8<ZOMBIE>'
Started music on hold, class 'default', on SIP/voipdiscount2-08224fe0
Parked SIP/voipdiscount2-08224fe0 on 701@parkedcalls. Will timeout back to
extension [outgoing], 1 in 45 seconds
<SIP/2001-0821f9e8> Playing 'digits/7' (language 'en')
<SIP/2001-0821f9e8> Playing 'digits/0' (language 'en')
<SIP/2001-0821f9e8> Playing 'digits/1' (language 'en')
Added extension '701' priority 1 to parkedcalls
Stopped music on hold on SIP/voipdiscount2-08224fe0
Registered extension context 'park-dial'
Added extension 'SIP/2001' priority 1 to park-dial
Timeout for SIP/voipdiscount2-08224fe0 parked on 701. Returning to park-dia
l,SIP/2001,1
Executing [SIP/2001@park-dial:1] Dial("SIP/voipdiscount2-08224fe0", "SIP/
2001:it") in new stack
Called 2001
SIP/2001-0821f9e8 is ringing
SIP/2001-0821f9e8 answered SIP/voipdiscount2-08224fe0
```

**Σχήμα 5.18:** Παράδειγμα λειτουργίας της εφαρμογής call park όπου μια παρκαρισμένη κλήση ξεπερνάει το χρονικό διάστημα που μπορεί να βρίσκεται σε κάποια θέση

## Βιβλιογραφικές Αναφορές

- 1) Bergman, Kevin, The History of Voice over Internet Protocol [online] [cited 08/09/2009]  
<<http://www.voip-articles.net/history-of-voip/the-history-of-voice-over-internet-protocol.html>>
- 2) Hallock, Joe, A Brief History of VoIP Document One – The Past [online] [cited 08/09/2009]  
<[http://www.joehallock.com/edu/pdfs/Hallock\\_J\\_VoIP\\_Past.pdf](http://www.joehallock.com/edu/pdfs/Hallock_J_VoIP_Past.pdf)>
- 3) Davidson, Jonathan, Peters, James 2000, Voice over IP Fundamentals, Chapter8: VoIP: An In-Depth Analysis, Cisco Press, pp.120- 121
- 4) Toncar, Vladimír, VoIP Basics: About Jitter [online] [cited 11/10/2009]  
<[http://toncar.cz/Tutorials/VoIP/VoIP\\_Basics\\_Jitter.html](http://toncar.cz/Tutorials/VoIP/VoIP_Basics_Jitter.html)>
- 5) Davidson, Jonathan, Peters, James 2000, Voice over IP Fundamentals, Chapter8: VoIP: An In-Depth Analysis, Cisco Press, pp.125- 126
- 6) Davidson, Jonathan, Peters, James 2000, Voice over IP Fundamentals, Chapter8: VoIP: An In-Depth Analysis, Cisco Press, pp.127
- 7) J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, “SIP: Session Initiation Protocol”, Internet Engineering Task Force RFC 3261, June 2002

- 8) ITU-T Recommendation H.323.0 (2006), Packet-based multimedia communications systems
- 9) Radvision Ltd, An Overview of H.323 - SIP Interworking [online] [cited 15/10/2009]  
<<http://www.radvision.com/NR/rdonlyres/1B7C291A-148C-4506-8312-D6DA2C58C7B7/0/OverviewofH323SIPInterworking.pdf>>
- 10) VoipForo, Images database [online], [cited 20/10/2009]  
<<http://www.voipforo.com/en/images/>>
- 11) Wikipedia, Inter-Asterisk eXchange, [online] [cited 26/10/2009]  
<<http://en.wikipedia.org/wiki/Iax>>
- 12) Wikipedia, Real-time Transport Protocol, [online] [cited 28/10/2009]  
< [http://en.wikipedia.org/wiki/Real-time\\_Transport\\_Protocol](http://en.wikipedia.org/wiki/Real-time_Transport_Protocol)>
- 13) Wikipedia, Secure Real-time Transport Protocol, [online] [cited 28/10/2009]  
< [http://en.wikipedia.org/wiki/Secure\\_Real-time\\_Transport\\_Protocol](http://en.wikipedia.org/wiki/Secure_Real-time_Transport_Protocol)>
- 14) Cisco, Voice Over IP - Per Call Bandwidth Consumption (Document ID: 7934), [online] [cited 15/11/2009]  
[http://www.cisco.com/en/US/tech/tk652/tk698/technologies\\_tech\\_note09186a0080094ae2.shtml](http://www.cisco.com/en/US/tech/tk652/tk698/technologies_tech_note09186a0080094ae2.shtml)
- 15) Digium, Asterisk Business Edition License [online] [cited 25/11/2009]  
<<http://www.digium.com/en/products/software/abe.php>>
- 16) Asterisk-The Open Source Telephony Project, General Public License Asterisk (GLP) [online] [cited 25/11/2009]  
<<http://www.asterisk.org/doxygen/trunk/Licensing.html>>

- 17) Gonçalves, Flavio, 2007, Configuration Guide for Asterisk PBX, 2nd Edition, Chapter1: Asterisk Introduction, V.Office Networks Ltda, pp.17- 20
- 18) Gonçalves, Flavio, 2007, Configuration Guide for Asterisk PBX, 2nd Edition, Chapter1: Asterisk Introduction, V.Office Networks Ltda, pp.24- 30
- 19) Digium blog, Asterisk Features List [online] [cited 10/12/2009]  
<<http://asteriskpbx-digium.blogspot.com/2009/12/asterisk-features-list.html>>
- 20) Asterisk-The Open Source Telephony Project, AsteriskNOW [online] [cited 20/12/2009]  
<<http://www.asterisk.org/asterisknow>>
- 21) Trixbox, Trixbox CE [online] [cited 20/12/2009]  
<[http://www.trixbox.org/downloads?utm\\_source=trixboxOrg&utm\\_medium=quick-link&utm\\_content=rightbartop-trixboxce&utm\\_campaign=fonalitysites](http://www.trixbox.org/downloads?utm_source=trixboxOrg&utm_medium=quick-link&utm_content=rightbartop-trixboxce&utm_campaign=fonalitysites)>
- 22) Benjamin Jackson, Champ Clark III, Long Johnny, Chaffin Larry, 2007, Asterisk Hacking, Chapter3: Configuring Asterisk, Syngress Publishing, Inc, Elsevier, Inc, pp.65- 115
- 23) Zoiper, Zoiper softphone [online] [cited 28/12/2009]  
<[http://www.zoiper.com/download\\_intermediate.php?os=Windows&ap=zc&location=zoiper\\_communicator\\_free\\_installer\\_exe\\_win](http://www.zoiper.com/download_intermediate.php?os=Windows&ap=zc&location=zoiper_communicator_free_installer_exe_win)>
- 24) A. Johnston, S. Donovan, R. Sparks, C. Cunningham, K. Summers, “Session Initiation Protocol (SIP) Public Switched Telephone Network (PSTN) Call Flows”, Internet Engineering Task Force RFC 3666, December 2003

РАНЕКІШНО ПЕРПА