

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων

Ανάπτυξη υπηρεσιών VoIP στην πλατφόρμα Asterisk

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Παναγιωτόπουλος Ι. Ανδρέας

Μάρτιος 2009



Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	2
Περίληψη.....	5
1. Εισαγωγή στο VoIP και στον Asterisk.....	8
1.1 VoIP.....	8
1.1.1 Πλεονεκτήματα	8
1.1.2 Μειονεκτήματα	9
1.2.1 Αρχιτεκτονική του Asterisk.....	13
2.1 Εισαγωγή.....	20
2.1.1 Η ανάγκη για VoIP πρωτόκολλα.....	21
2.2 VoIP Πρωτόκολλα	22
2.2.1 IAX (Inter-Asterisk eXchange Protocol)	22
2.2.2 SIP	24
2.2.3 H.323.....	26
2.3 Codecs.....	28
2.4 Quality of Service.....	31
2.4.1 TCP , UDP και SCTP	32
2.4.2 Διαφορική Υπηρεσία (Differentiated Service).....	33
2.4.3 Εγγυημένη Υπηρεσία (Guaranteed Service).....	34
2.4.4 Μέγιστη Προσπάθεια (Best Effort).....	34
2.5 Echo (ηχώ).....	34
2.5.1 Γιατί Υπάρχει Η Ηχώ.....	35

2.5.2 Διαχείριση Echo Μέσω του ASTERISK	35
3.1 Εισαγωγή.....	37
3.2 Ρυθμίζοντας το SIP	37
3.3 Ρυθμίζοντας το IAX.....	41
3.4 Ρυθμίζοντας τα Extensions	42
3.5 Ρυθμίζοντας το Voice Mail.....	42
3.6 Ρυθμίζοντας Ειδικές Λειτουργίες	45
3.7 Ρυθμίζοντας την Απομακρυσμένη Διαχείριση.....	46
4.1 Εισαγωγή.....	49
4.2 Σύνταξη του Dialplan	49
4.2.1 Contexts	49
4.2.2 Extensions	53
4.2.3 Προτεραιότητες (Priorities)	54
4.2.4 Εφαρμογές (Applications)	55
4.3 Το Πρώτο Μας Dialplan	57
5.1 Εργαλεία Για Την Ανάπτυξη Εσωτερικής Λογικής.....	60
5.1.1 Μεταβλητές.....	60
5.1.2 Η εφαρμογή Goto() και Ετικέτες	62
5.1.3 Η εφαρμογή Gotoif().....	64
5.1.4 Συναρτήσεις (Functions)	66
5.2 Δημιουργία Εφαρμογών Με εσωτερική Λογική	67
5.2.1 Φωνητικό ταχυδρομείο	67
5.2.2 Πρόγνωση Καιρού	70
5.2.3 Date Time	72
5.3 Προγραμματισμός AGI.....	73
5.3.1 Τι είναι το AGI.....	73
5.3.2 Επιπλέον AGI Βιβλιοθήκες και Ανάπτυξη Αλγορίθμων	76

5.3.3 AGI Wake-Up Call Perl.....	77
6. Προσομοίωση Λειτουργιών Asterisk.....	79
6.1 Εισαγωγή.....	79
6.2 Dial To All Users.....	79
6.3 Voice Mail	82
6.4 Call Forward	88
6.5 Date and Time	89
6.6 Weather Forecast.....	91
6.7 Wakeup Call	92
7. Επίλογος και Μελλοντικές Προτάσεις.....	95
7.1 Επίλογος.....	95
7.2 Μελλοντικές Προτάσεις.....	95
Παράρτημα.....	97
Βιβλιογραφία.....	101

Περίληψη

Το *Voice over IP* ή *VoIP* ή τηλεφωνία μέσω διαδικτύου χαρακτηρίζει μια ομάδα πρωτοκόλλων-τεχνολογιών (H.323, SIP), η οποία προσφέρει φωνητική συνομιλία σε πραγματικό χρόνο με σχετικά καλή ποιότητα πλέον και στην ουσία χωρίς κόστος. Οι συνομιλίες αυτές παραδοσιακά γίνονταν αποκλειστικά μέσω PC που ήταν συνδεδεμένο με το Διαδίκτυο (Internet) και διέθετε μικρόφωνο, ακουστικά και το κατάλληλο λογισμικό. Η κλήση κατέληγε σε ένα άλλο, ανάλογα εξοπλισμένο, PC χωρίς να υπάρχει κάποια επιπλέον χρώση, εκτός από αυτή της πρόσβασης στο Διαδίκτυο, αφού στη συγκεκριμένη επικοινωνία δεν μεσολαβεί κάποιος παραδοσιακός φορέας τηλεπικοινωνιών (π.χ. ΟΤΕ) παρά μόνο το Διαδίκτυο.

Τον τελευταίο καιρό έχουν εμφανιστεί οι λεγόμενοι ιντερνετικοί τηλεπικοινωνιακοί φορείς, οι οποίοι προσφέρουν προώθηση των κλήσεων VoIP σε σταθερά δίκτυα τηλεπικοινωνιών σε εξαιρετικά χαμηλό κόστος, αλλά όχι το αντίστροφο. Μερικοί εξ αυτών έχουν παρουσιάσει και ειδικές τηλεφωνικές συσκευές USB VoIP, οι οποίες συνεργάζονται με το αντίστοιχο λογισμικό στον Η/Υ και καθιστούν τις κλήσεις μέσω Διαδικτύου σαφώς πιο λειτουργικές.

Αυτό όμως δεν είναι αρκετό, ειδικά σε επαγγελματικές λύσεις. Εδώ έρχεται η πλατφόρμα του Asterisk. Το Asterisk είναι ένα ολοκληρωμένο PBX software. Συνδυάζει τα πλεονεκτήματα του VoIP και ενός καλού ψηφιακού τηλεφωνικού κέντρου. Τρέχει σε Linux, BSD, Windows και OS X και παρέχει όλα τα χαρακτηριστικά που περιμένει ο χρήστης από ένα PBX. Το Asterisk παρέχει φωνή μέσα από το IP σε τέσσερα πρωτοκολλά, και μπορεί να επικοινωνήσει με σχεδόν όλα τα standards-based τηλέφωνα χρησιμοποιώντας ιδιαίτερα φθηνό hardware

Το Asterisk παρέχει Voicemail υπηρεσίες με Directory, Call Conferencing, Interactive Voice Response, Call Queuing. Έχει υποστήριξη για άλλους τρόπους

κλήσης, caller ID υπηρεσίες, ADSI, IAX, SIP, H.323 (client και gateway), MGCP (call manager μόνο) και SCCP/Skinny.

Το Asterisk δεν χρειάζεται extra hardware για μεταφορά φωνής μέσα από IP, παρά ταύτα εγκαθιστά έναν driver που δημιουργεί ένα dummy hardware. Ένας (ή πολλοί) VoIP πάροχοι μπορούν να χρησιμοποιηθούν για εξωτερικές οι εσωτερικές κλήσεις.

Για την διασύνδεση με ψηφιακό ή αναλογικό τηλεφωνικό εξοπλισμό, το Asterisk υποστηρίζει διάφορα hardware, άξια αναφοράς είναι τα υλικά που παράγονται από τον sponsor του Asterisk, την Digium.

Η παρούσα διπλωματική εργασία , έρχεται να συνδυάσει τα πλεονεκτήματα του Asterisk μέσω της τρομερής ανάπτυξης και εδραίωσης του VoIP. Τα πλεονεκτήματα αυτά θα παρουσιαστούν μέσω διαφόρων , που έχουν υλοποιηθεί στην εν λόγω πλατφόρμα. Οι εφαρμογές είναι αντιπροσωπευτικές και σκοπό έχουν να αναδείξουν τις δυνατότητες του Asterisk.

Η παρούσα εργασία αποτελείται από 7 κεφάλαια .

Το 1^ο κεφάλαιο είναι εισαγωγικό και αναφέρεται γενικά στο VoIP και στην πλατφόρμα του Asterisk.

Στο 2^ο κεφάλαιο θα αναλυθούν τα διάφορα πρωτόκολλα που απαρτίζουν το VoIP. Στην συνέχεια θα παρουσιαστούν οι codec που χρησιμοποιούνται για την κωδικοποίηση και την συμπίεση της φωνής , πριν αυτή μεταδοθεί μέσω του TCP/IP. Τέλος θα παρουσιαστούν τα διάφορα ζητήματα που προκύπτουν στην ποιότητα της υπηρεσίας (QoS).

Στο 3^ο κεφάλαιο θα αρχίζουμε να μπαίνουμε στις ρυθμίσεις που πρέπει να γίνουν στον Asterisk όσο αναφορά το πρωτόκολλο που θα χρησιμοποιηθεί , το οποίο στην προκείμενη περίπτωση είναι το SIP.

Στο 4^ο κεφάλαιο θα αναλύσουμε την καρδιά του Asterisk , το Dialplan. Η ανάλυση θα είναι συναρτήσεων των εφαρμογών-υπηρεσιών , που θα παρουσιάσουμε στο επόμενο κεφάλαιο.

Στο 5^ο κεφάλαιο αναλύονται και αναπτύσσονται διάφορες προηγμένες τεχνικές υπηρεσιών VoIP , οι οποίες είναι αντιπροσωπευτικές στο να αναδείξουν τις δυνατότητες της πλατφόρμας του Asterisk.

Στο 6^ο κεφάλαιο γίνεται προσομοίωση των υπηρεσιών που αναπτύχθηκαν για να διαπιστωθεί η καλή τους λειτουργία , καθώς επίσης αναλύεται και ο τρόπος που αυτές εκτελούνται.

Στο 7^ο κεφάλαιο τέλος, με βάση τις υπηρεσίες που αναπτύχθηκαν , γίνεται μια εξαγωγή συμπερασμάτων καθώς επίσης προτείνονται μελλοντικές προεκτάσεις με βάση την συγκεκριμένη διπλωματική εργασία .

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΑΣ

1. Εισαγωγή στο VoIP και στον Asterisk

1.1 VoIP

Το "VoIP" είναι τα αρχικά για την τεχνολογία που ονομάζεται Voice Over Internet Protocol, δηλαδή Φωνή Πάνω από Πρωτόκολλο του Διαδικτύου. Με απλά λόγια, η τεχνολογία VoIP μας επιτρέπει να αξιοποιήσουμε οποιαδήποτε δικτυακή υποδομή που χρησιμοποιεί το πρωτόκολλο IP για να μεταφέρουμε φωνή.

Όλες οι σύγχρονες δικτυακές υποδομές βασίζονται πλέον πάνω σε πρωτόκολλο IP, συμπεριλαμβάνοντας φυσικά το VoIP αλλά και ολόκληρο το Διαδίκτυο. Γι' αυτό, δεν είναι δύσκολο να αντιληφθεί κανείς πόσο εύκολο και χρήσιμο θα ήταν να αξιοποιηθεί μία τέτοια τεχνολογία.

1.1.1 Πλεονεκτήματα

Τα πλεονεκτήματα της VoIP τεχνολογίας θα πρέπει να συγκριθούν πάντα με την συμβατική τηλεφωνία, δηλαδή, την γνωστή υπηρεσία τηλεφωνίας με την οποία είμαστε όλοι εξοικειωμένοι. Μερικά από αυτά τα πλεονεκτήματα είναι:

- **Φθηνή υπηρεσία** - επειδή χρησιμοποιείται η υπηρεσία μέσω ενός δημοσίου δικτύου (Διαδίκτυο), η μεταφορά φωνής μεταξύ δύο σημείων μπορεί να γίνει χωρίς χρέωση.
- **Πρόσβαση** - ένας χρήστης μπορεί να έχει πρόσβαση στην υπηρεσία από οποιοδήποτε σημείο υπάρχει δίκτυο. Δεν περιορίζετε από ένα συγκεκριμένο καλώδιο το οποίο έρχεται στο σπίτι του.

- **Πλούσια υπηρεσία** - επειδή είναι δικτυακή υπηρεσία, το VoIP μπορεί να συνδυαστεί και με άλλες υπηρεσίες (όπως βίντεο για παράδειγμα) για να κάνει την εμπειρία του χρήστη πιο πλούσια.

1.1.2 Μειονεκτήματα

Όπως με κάθε τη καλό, υπάρχουν και κάποια μειονεκτήματα.

- **Ποιότητα φωνής** - Η τεχνολογία VoIP προσπαθεί να περάσει φωνή πάνω από μία υποδομή η οποία δεν σχεδιάστηκε αρχικά γι' αυτό το σκοπό. Επίσης, η φωνή περνάει πάνω από ένα δίκτυο το οποίο μεταφέρει κι άλλα δεδομένα, κάτι το οποίο μπορεί να δημιουργήσει διακοπές στην συνομιλία. Ωστόσο, μέτρα μπορούν και έχουν παρθεί ώστε να εξασφαλιστεί η ποιότητα της φωνής δίνοντας σχεδόν την ίδια ποιότητα με την συμβατική τηλεφωνία και σχεδόν την ίδια διαθεσιμότητα.

1.2 Asterisk^{[1],[5]}

Στην κοινότητα του Asterisk έχει επιχειρηθεί αρκετές φορές να δοθεί ένας περιεκτικός ορισμός του τι ακριβώς είναι το Asterisk και ποιες είναι οι δυνατότητες του. Οι δημιουργοί του θέλοντας να απλοποιήσουν την έννοια του Asterisk και να το κάνουν ποιο ελκυστικό στο ευρύ κοινό, συχνά χρησιμοποιούν τη φράση: “*Είναι απλά λογισμικό*”. Στην πραγματικότητα όμως, είναι πολλά περισσότερα. Κάποιος μπορεί να καταλάβει τι είναι το Asterisk κοιτάζοντας την ετοιμολογία του ονόματός του. Ο ειδικός χαρακτήρας του αστερίσκου (*) είναι συγχρόνως ένα πλήκτρο του τηλεφωνικού πληκτρολογίου καθώς επίσης και ένας ειδικός χαρακτήρας στα λειτουργικά συστήματα UNIX και DOS που μπορεί να συμβολίσει οποιοδήποτε αλφαριθμητικό χαρακτήρα (π.χ. rm -rf *). Έτσι και το Asterisk έχει σχεδιαστεί ώστε να μπορεί να διασυνδεθεί με οποιοδήποτε τηλεφωνικό υλικό ή λογισμικό απρόσκοπτα.

Το Asterisk ξεφεύγει από τα όρια του τηλεπικοινωνιακού προγράμματος και χαρακτηρίζεται σωστότερα από την έννοια της τηλεπικοινωνιακής πλατφόρμας. Δημιουργεί δηλαδή ένα πλαίσιο μέσα στο οποίο θα μπορούσε να αναπτυχθεί το οποιοδήποτε υπάρχον (ή μελλοντικό) τηλεπικοινωνιακό σύστημα. Μπορεί να λειτουργήσει σαν αυτόνομος εξυπηρετητής επεξεργασίας κλήσεων ή ακόμα και σαν προσθήκη σε κάποιο ήδη εγκατεστημένο κέντρο. Μπορεί να χρησιμοποιηθεί μόνο σε επίπεδο λογισμικού, μεταφέροντας φωνή μέσω IP ή να επικοινωνήσει με TDM (Time Division Multiplexing) διεπαφές και να χρησιμοποιήσει το τηλεφωνικό δίκτυο.

Μέσω του Asterisk μπορεί να επιτευχθεί ποικιλία μεταβάσεων όπως:

- Κλειστό (εμπορικό) → Ελεύθερο (δωρεάν)
- Δίκτυα μεταγωγής κυκλώματος → VoIP συστήματα
- Φωνή → Φωνή, Βίντεο, Δεδομένα
- Digital Signal Processing → Host Media Processing
- Κεντρική διαχείριση → Peer to Peer

Ενώ παράλληλα παρέχονται γέφυρες επικοινωνίας με όλες τις προϋπάρχουσες τεχνολογίες και τα πρότυπά τους.

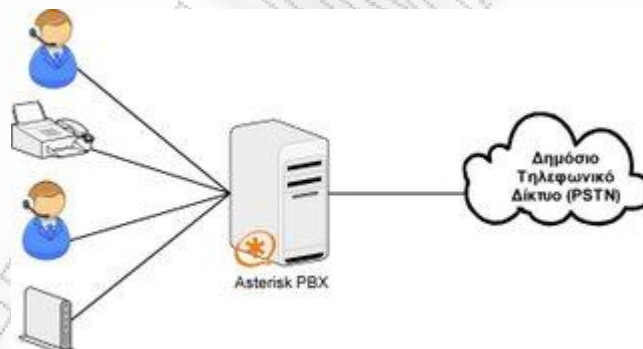
Το Asterisk είναι ελεύθερο λογισμικό

Το Asterisk απολαμβάνει αυτήν τη στιγμή μεγάλη εκτίμηση από την κοινότητα του ελεύθερου λογισμικού, το οποίο στην πράξη σημαίνει ότι εκτός από το γεγονός ότι είναι δωρεάν, ταυτόχρονα σε καθημερινή βάση εκατοντάδες (ή ακόμα και χιλιάδες) προγραμματιστές και σύμβουλοι τηλεπικοινωνιών ασχολούνται με το Asterisk, δημιουργώντας εφαρμογές και προσαρμοσμένες εγκαταστάσεις, με σκοπό να το επεκτείνουν, να το βελτιώσουν, να προβλέψουν τις νέες τεχνολογίες που έρχονται (αν όχι να τις δημιουργήσουν) και να τις ενσωματώσουν έγκαιρα.

Το Asterisk είναι πλατφόρμα πολύπλεξης με διαίρεση χρόνου (TDM) και ανταλλαγής πακέτων φωνής.

Αυτό σημαίνει ότι υποστηρίζει τα υπάρχοντα TDM πρωτόκολλα τηλεπικοινωνιών όπως το Ψηφιακό Δίκτυο Ενοποιημένων Υπηρεσιών (ISDN-BRA-PRI), το Δημόσιο Τηλεπικοινωνιακό Δίκτυο Μεταγωγής (PSTN), το FXS, το FXO, το E1, το T1 και σε γενικές γραμμές οτιδήποτε χρησιμοποιείται στην κλασσική τηλεφωνία όπως τη γνωρίζουμε μέχρι σήμερα. Παράλληλα όμως υποστηρίζει και τα καινούργια VoIP (Voice over Internet Protocol) πρωτόκολλα όπως το SIP, το IAX, το H.323, το MGCP, το SCCP (Cisco-Skinny) και σύντομα το Jingle (Google Talk). Η γεφύρωση των τεχνολογιών προσφέρει στο χρήστη, δυνατότητα μετάβασης στις νέες μορφές επικοινωνίας χωρίς όμως να αναγκαστεί να αποχωριστεί τις παλιές του συνήθειες και εγκαταστάσεις.

Το Asterisk είναι ιδιωτικό σύστημα τηλεφωνικής μεταγωγής (PBX)



Εικόνα 1.1 Το Asterisk ως PBX

Τα συστήματα PBX χρησιμοποιούν στη ζεύξη μεταξύ των τερματικών (τηλέφωνο, fax, modem, κ.α.) που είναι συνδεδεμένα με το PBX και του δημόσιου τηλεπικοινωνιακού δικτύου μεταγωγής (PSTN). Με αυτήν τη συνδεσμολογία είναι δυνατόν να μη χρειάζεται κάθε τερματικό μια αποκλειστική σύνδεση με το PSTN δίκτυο αλλά αντί αυτού να χρησιμοποιείται ένα πλήθος γραμμών ανάλογο με τις απαιτήσεις για επικοινωνία με τον έξω κόσμο (Line Trunking).

Το Asterisk μπορεί να λειτουργήσει σαν ένα παραδοσιακό PBX με τις ευκολίες που αυτά παρέχουν όπως: ενδοεπικοινωνία (extension-to-extension), τριμερής επικοινωνία, τηλεδιάσκεψη, φωνητικό ταχυδρομείο, αναμονή κλήσεων, προώθηση κλήσεων, καταγραφή αρχείου κλήσεων, ηχογράφηση συνομιλιών, κ.α., με τη διαφορά ότι όλα αυτά τα κάνει με πλήρη διαφάνεια και κατά βούληση του χρήστη και έτσι δεν επηρεάζεται από το πώς επικοινωνούν οι χρήστες, δηλαδή με soft phones, παραδοσιακά τηλέφωνα ή VoIP handsets και δεν περιορίζει την τοπολογία του τηλεφωνικού δικτύου.

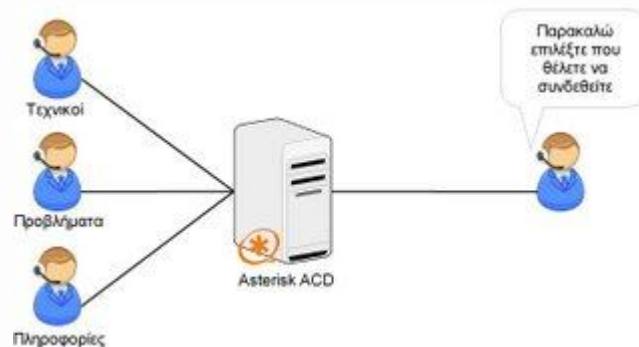
Το Asterisk είναι σύστημα αμφίδρομης φωνητικής απόκρισης (IVR)



Εικόνα 1.2 Το Asterisk ως IVR

Το Asterisk παρέχει τη δυνατότητα δημιουργίας συστήματος αμφίδρομης φωνητικής απόκρισης. Αυτό σημαίνει ότι ο καλών έχει τη δυνατότητα μέσω ενός φωνητικού μενού και με τη χρήση του πληκτρολογίου της τηλεφωνικής του συσκευής να αλληλεπιδράσει με το τηλεφωνικό κέντρο και να αποκτήσει πρόσβαση σε πληροφορίες οι οποίες βρίσκονται στο σύστημά μας. Χαρακτηριστικό παράδειγμα μίας τέτοιας εφαρμογής είναι το σύστημα που χρησιμοποιείται από πολλές τράπεζες, στο οποίο πληκτρολογούμε τον αριθμό της κάρτας μας και κάποιο συνθηματικό και αποκτάμε πρόσβαση σε πληροφορίες όπως το υπόλοιπο του λογαριασμού μας και τις τελευταίες κινήσεις του λογαριασμού.

Το Asterisk είναι σύστημα αυτόματης κατανομής κλήσεων (ACD)



Εικόνα 1.3 Το Asterisk ως ACD

Το Asterisk μπορεί να λειτουργήσει σαν σύστημα ουρών αναμονής των κλήσεων και αυτόματης δρομολόγησης στην κατάλληλη ουρά. Η λειτουργία αυτή, είναι από τις βασικότερες ενός τηλεφωνικού κέντρου, το οποίο θα πρέπει να είναι σε θέση να κάνει σωστή και αποτελεσματική διαχείριση των γραμμών του. Τα κριτήρια δρομολόγησης μπορούν να ποικίλουν, ανάλογα με την ώρα, τη διαθεσιμότητα, τα προσόντα, τα επίπεδα προτεραιοτήτων, κ.α.

1.2.1 Αρχιτεκτονική του Asterisk

Το Asterisk έχει σχεδιαστεί με κύρια προτεραιότητα την ευελιξία και τη συνδεσιμότητα, όπου συγκεκριμένα APIs ορίζουν τον πυρήνα του PBX συστήματος. Η εξελιγμένη αρχιτεκτονική του Asterisk, του επιτρέπει να χειρίζεται τις εσωτερικές διασυνδέσεις με πλήρη διαφάνεια, ανεξαρτήτως πρωτοκόλλων, κωδικοποιήσεων, και τηλεφωνικού υλικού. Με αυτόν τον τρόπο το Asterisk είναι σε θέση να χρησιμοποιήσει όλα τα κατάλληλα υλικά και τις τεχνολογίες που είναι διαθέσιμες σήμερα ή ακόμα και μελλοντικά, για να εκτελέσει τις βασικές του λειτουργίες, συνδέοντας υλικό και λογισμικό.

Ο πυρήνας του Asterisk

- *Μεταγωγέας PBX*: Η πρωταρχική λειτουργία του Asterisk όπως φαίνεται και από την πρώτη ονομασία του (Asterisk the Free PBX) είναι να λειτουργεί σαν σύστημα PBX, συνδέοντας κλήσεις μεταξύ χρηστών και ενεργειών. Ο πυρήνας μεταγωγής συνδέει χρήστες από διάφορες διεπαφές λογισμικού ή υλικού.

- *Εκτελεστής Εφαρμογών*: Εκτελεί εφαρμογές που παρέχουν λειτουργίες όπως, αναπαραγωγή αρχείων, αυτόματος τηλεφωνητής.

- *Μεταφραστής Codec*: Χρησιμοποιεί modules για την κωδικοποίηση και την αποκωδικοποίηση διαφόρων τύπων συμπίεσης ήχου που εφαρμόζονται στην τηλεφωνία. Υποστηρίζονται πολλοί codecs για να μπορέσει να επιτευχθεί μία ισορροπία μεταξύ ποιότητας ήχου και χρήσης του εύρους ζώνης.

- *Χρονοπρογραμματιστής και Ελεγκτής Εισόδου/Εξόδου*: Χειρίζεται λειτουργίες χρονοπρογραμματισμού και εποπτείας σε χαμηλό επίπεδο, επιτρέποντας την επίτευξη της καταλληλότερης επίδοσης σε κάθε περίπτωση φόρτου εργασίας.

APIs Φόρτωσης Modules

Υπάρχουν τέσσερα APIs για να φορτώνονται modules, τα οποία παρέχουν τη διαλειτουργικότητα σε θέματα υλικού και πρωτοκόλλων. Με τη χρήση αυτού του αρθρωτού συστήματος, ο πυρήνας του Asterisk δε χρειάζεται να γνωρίζει λεπτομέρειες για το πώς συνδέεται ο χρήστης, τι codecs χρησιμοποιεί, κ.λπ.

Τα APIs είναι τα:

- *Channel API*: Το channel API διαχειρίζεται τον τύπο της σύνδεσης από την οποία προέρχεται ο χρήστης. Η σύνδεση αυτή μπορεί να είναι VoIP, ISDN, POTS, ή οποιαδήποτε άλλη τεχνολογία. Τα modules φορτώνονται δυναμικά για να χειριστούν τις λεπτομέρειες της σύνδεσης.

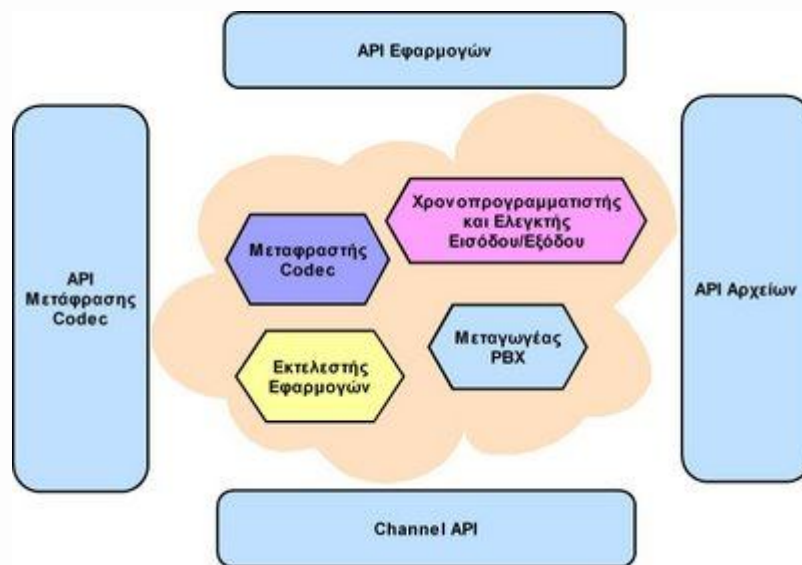
- *API Εφαρμογών*: Το API εφαρμογών, επιτρέπει στα modules εφαρμογών να εκτελεστούν ώστε να παρέχουν διάφορες λειτουργίες. Δυνατότητες όπως τηλεδιάσκεψη, μεταφορά δεδομένων, φωνητικό ταχυδρομείο και οποιαδήποτε άλλη

εργασία μπορεί να εκτελέσει ένα σύγχρονο ή μελλοντικό PBX, χειρίζονται από τα αντίστοιχα ξεχωριστά modules.

- API Μετάφρασης Codec: Το API αυτό φορτώνει ξεχωριστά modules για τον κάθε codec, για να υποστηρίξει τους διάφορους τύπους κωδικοποίησης και αποκωδικοποίησης ήχου που υπάρχουν, όπως: GSM, μLaw, aLaw, ακόμα και mp3.
- API Αρχείων: Το API αρχείων είναι υπεύθυνο για την ανάγνωση και εγγραφή πολλών τύπων αρχείων, και την αποθήκευση δεδομένων στο σύστημα.

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 αρθρωτή δομή του, του επιτρέπει να συνεργάζεται απόλυτα τόσο με τα παραδοσιακά συστήματα, όσο και με τις νέες τεχνολογίες μετάδοσης πακέτων φωνής. Η δυνατότητα που έχει το Asterisk να φορτώνει 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.

Το API εφαρμογών παρέχει τη δυνατότητα στα modules εφαρμογών, να εκτελούν οποιαδήποτε λειτουργία ευέλικτα και κατά βούληση του χρήστη. Επιτρέπει επίσης την εκτέλεση εφαρμογών που έχουν αναπτυχθεί αποκλειστικά για να καλύψουν προσαρμοσμένες ανάγκες και περιπτώσεις χρήσης. Επιπλέον, φορτώνοντας όλες τις εφαρμογές ως modules το Asterisk δίνει τη δυνατότητα στους διαχειριστές να σχεδιάσουν (ευπροσάρμοστα) συστήματα με εύκολη προσαρμογή στις τηλεπικοινωνιακές αλλαγές που μπορεί να προκύψουν.



Εικόνα 2.4 Δομή Asterisk API

Η λέξη *codec* προέρχεται απ' την σύμπτυξη των λέξεων *COder* - *DECoder* και αφορά στην διαδικασία κωδικοποίησης κατά την αποστολή των δεδομένων και αποκωδικοποίησης κατά την λήψη.

Λειτουργίες και Δυνατότητες του Asterisk

Όπως αναφέρθηκε σε προηγούμενος είναι πολύ δύσκολο να περιγραφεί το πλήρες φάσμα των δυνατοτήτων του Asterisk λόγω του πλήθους των περίπλοκων θεμάτων που ενσωματώνει: πολλαπλούς τύπους VoIP καναλιών, υλικά διασύνδεσης, γλώσσα δέσμης ενεργειών (AGI Scripting language), Διασύνδεση Προγράμματος Εφαρμογής (API) και πληθώρα λειτουργιών.

Παρακάτω θα αναφερθούμε σε κάποιες από τις λειτουργίες που κάνουν το Asterisk τόσο ισχυρό και σε επόμενες δημοσιεύσεις θα δούμε πως μπορούμε να υλοποιήσουμε μερικές από αυτές στην πράξη και πού θα μπορούσαν να μας χρησιμεύσουν.

➤ *ADSI On-Screen Menu System*: Εμφάνιση μενού στην οθόνη κατάλληλου τηλεφώνου (screen phone) μέσω του αναλογικού δικτύου για παροχή προσαρμοσμένων λειτουργιών.

➤ *Alarm Receiver*: Δυνατότητα ειδοποίησης ανάλογα με κάποια προσαρμοσμένα όρια που αφορούν την απόδοση του τηλεφωνικού μας κέντρου (π.χ. μεγάλη αναμονή).

➤ *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.λπ.)

➤ *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 με στοιχεία το νούμερο και το όνομα του καλούντα (αν είναι διαθέσιμα).

➤ *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.
- *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 φαξ 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 Dialler*: Αυτόματη κλήση σε τηλεφωνικά νούμερα. Χρησιμοποιείται σε τηλεφωνικά κέντρα (τηλε-μάρκετινγκ) και πραγματοποιεί κλήσεις προς πιθανούς πελάτες με χρήση εξειδικευμένων αλγορίθμων πρόβλεψης.
- *Open Settlement Protocol (OSP)*: Δυνατότητα τιμολόγησης VoIP υπηρεσιών.
- *Roaming Extensions*: Δυνατότητα περιαγωγής της 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 Gateway*: Δυνατότητα διασύνδεσης VoIP τερματικών ανεξαρτήτως πρωτοκόλλου που χρησιμοποιεί το καθένα και μετατροπή των μη συμβατών πρωτοκόλλων.
- *Voicemail*: Φωνητικό ταχυδρομείο 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. τηλε-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-AsteriskExchange), H.323, SIP (SessionInitiationProtocol), MGCP (Media Gateway Control Protocol, SCCP (CiscoSkinny).

2. Πρωτόκολλα Για VoIP^([1],[3],[7],[8])

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα εξηγήσουμε τι ακριβώς είναι το Voice Over IP και τι κάνει τα δίκτυα VoIP τόσο διαφορετικά από τα παραδοσιακά circuit-switched voice δίκτυα. Παρακάτω θα αναφέρουμε τα πρωτόκολλα τα οποία θα αναλύσουμε:

- IAX
- SIP
- H.323

Εκτός από τα πρωτόκολλα , έχουμε και τους *Codecs*. Ένας codec είναι το μέσω από το οποίο το αναλογικό μας σήμα (στην προκείμενη περίπτωση η φωνή) μετατρέπεται σε ψηφιακό σήμα και μεταφέρεται μέσω του Internet. Το εύρος (*bandwidth*) σε οποιαδήποτε θέση , είναι πεπερασμένο με αποτέλεσμα ο αριθμός των ταυτόχρονων συνομιλιών να είναι πεπερασμένος και έτσι κάθε σύνδεση που δημιουργείται να εξαρτάται πλήρως από τον codec που έχει χρησιμοποιηθεί. Παρακάτω θα αναλύσουμε τις διαφορές μεταξύ των παρακάτω codec :

- G.711
- G.726
- G.723.1
- G.729A
- GSM
- iLBC

- Speex
- MP3

Οι διαφορές θα εστιαστούν τόσο σε απαιτήσεις bandwidth(μέγεθος συμπίεσης) , όσο και σε επίπεδο ποιότητας.

Τέλος θα ολοκληρώσουμε την ανάλυση μας με το πώς η κίνηση (*Traffic*) της φωνής δρομολογείται και τι προκαλεί την εμφάνιση των φαινομένων *echo* και πώς εξαλείφεται , καθώς και πως ο Asterisk διαχειρίζεται την πιστοποίηση (*authentication*) των εισερχόμενων και εξερχόμενων κλήσεων.

2.1.1 Η ανάγκη για VoIP πρωτόκολλα

Η βασική προϋπόθεση για να μπορεί να υπάρξει το VoIP είναι η “πακετοποίηση” (*packetization*) της ροής του ήχου (*audio stream*) για την μεταφορά μέσω των δικτύων βασισμένων σε Internet πρωτόκολλα. Η πρόκληση για να μπορέσει να επιτευχθεί κάτι τέτοιο , εξαρτάται από τον τρόπο που επικοινωνούν οι άνθρωποι. Θα πρέπει όχι μόνο το σήμα να φτάνει ακριβώς όπως μεταδίδεται , αλλά θα πρέπει να γίνεται αυτό σε χρόνο μικρότερο από 300 millisecond. Αν τα πακέτα καθούν ή αργήσουν , θα υπάρξει υποβίβαση στην ποιότητα της επικοινωνίας.

Τα πρωτόκολλα μετάδοσης (*transport protocols*) αποτελούν το Internet , δεν σχεδιάστηκαν αρχικά για μετάδοση ροής πραγματικού χρόνου (*real – time streaming*) . Όταν το άλλο “άκρο” της επικοινωνίας δεν λάμβανε κάποιο πακέτο , τότε προσπαθούσε να το λύσει το πρόβλημα αυτό , περιμένοντας περισσότερη ώρα ή ζητώντας επαναμετάδοση του πακέτου αυτού ή σε μερικές περιπτώσεις , ανάλογα με την πληροφορία , την αγνόηση αυτού του πακέτου. Σε επικοινωνία φωνής , αυτές οι μέθοδοι δεν εξυπηρετούν. Η ποιότητα της συνομιλίας μας δεν είναι ικανοποιητική όταν χάνονται γράμματα από τις φράσεις ή όταν υπάρχει καθυστέρηση στην ομιλία.

Το παραδοσιακό PSTN (*Public Switched Telephone Network*) σχεδιάστηκε για να ικανοποιήσει τις ανάγκες για την μετάδοση φωνής , από την τεχνική σκοπιά. Από την σκοπιά της ευελιξίας όμως , το ελάττωμα του είναι εμφανές ακόμα και από κάποιον που δεν έχει ιδιαίτερες τεχνολογικές γνώσεις. Το VoIP κρατάει την

υπόσχεση για ενσωμάτωση των επικοινωνιών φωνής μαζί με τα υπόλοιπα πρωτόκολλα που χρησιμοποιούμε ήδη στα δίκτυα μας. Αλλά εξαιτίας των ειδικών απαιτήσεων που έχει οι επικοινωνία φωνής, θα πρέπει να γίνουν ειδικοί σχεδιασμοί στον σχεδιασμό και ταυτόχρονα ιδιαίτερες συντηρήσεις στα υπάρχοντα δίκτυα , για να μπορέσουν να το υποστηρίξουν.

Το πρόβλημα με τα packet-based πρωτόκολλα φωνής , είναι ότι ο τρόπος της ομιλίας είναι τελείως μη συμβατός με τον τρόπο που χρησιμοποιούν τα IP πρωτόκολλα. Η ομιλία αποτελείται από την αλληλουχία της ροής του ήχου , οπότε εκεί τα Internet πρωτόκολλα είναι σχεδιασμένα να τεμαχίζουν και στην συνέχεια να συμπιέζουν κάθε bit πληροφορίας σε χιλιάδες πακέτα και έπειτα να παραδίδουν με οποιοδήποτε τρόπο το κάθε πακέτο στον προορισμό του. Είναι φανερό ότι χρειάζεται κάποια ενοποίηση μεταξύ αυτών.

2.2 VoIP Πρωτόκολλα

Ο μηχανισμός για να μπορέσει να υπάρξει μια VoIP σύνδεση , συνήθως εμπλέκει διάφορες σηματοδοτικές διεργασίες μεταξύ των άκρων της συνομιλίας , συμπεριλαμβανομένου των μεταξύ *gateways* , διατηρώντας 2 συνεχείς media streams (μία για κάθε κατεύθυνση) , οι οποίες διατηρούν την συνομιλία. Υπάρχουν πολλά πρωτόκολλα που μπορούν να χειριστούν αυτές τις απαιτήσεις. Παρακάτω θα αναφέρουμε μερικά από αυτά , τα οποία είναι απαραίτητα για το VoIP γενικά και ειδικά για τον Asterisk.

2.2.1 IAX (Inter-Asterisk eXchange Protocol)

Το IAX είναι ένα ελεύθερο πρωτόκολλο, το οποίο σημαίνει ότι μπορεί ο οποιοσδήποτε να το κατεβάσει και να αναπτύξει πάνω σε αυτό. Ακόμα το IAX δεν αποτελεί κάποιο standard πρωτόκολλο.

Ιστορία του Πρωτοκόλλου

Το IAX το ανέπτυξε η Digium με σκοπό να επικοινωνούν οι μεταξύ τους Asterisk Servers , γι αυτό τον λόγο ονομαστικέ και Inter Asterisk eXchange Protocol .Το IAX είναι *transport* πρωτόκολλο το οποίο χρησιμοποιεί μία UDP πόρτα (την 4569) για σηματοδότηση (*signaling*) στο κανάλι και ταυτόχρονα για τις ροές πραγματικού χρόνου (*Real Time Protocol streams*) . Όπως θα αναφέρουμε παρακάτω αυτό είναι πολύ χρήσιμο για να μπορέσει να δουλέψει το *Firewall* πίσω από *NAT*.

Το IAX επίσης έχει την ικανότητα να διαχειριστεί πολλαπλά *trunk sessions* σε ένα μόνο *Dataflow* , το οποίο αυτό είναι τεράστιο πλεονέκτημα στο υπάρχον εύρος (*bandwidth*) , όταν στέλνονται ταυτόχρονα κανάλια σε ένα προορισμό. Το *Trunking* επιτρέπει να υπάρχουν πολλαπλές ροές δεδομένων (*Data Streams*) , οι οποίες απεικονίζονται με μία μόνο κεφαλίδα ενός *Datagram* , για να μπορέσουμε να μειώσουμε το *overhead* που σχετίζεται με τα το σύνολο των καναλιών. Αυτό μας βοηθάει σε μείωση του *latency* και στην ελάττωση της απαιτούμενης ενέργειας και του *Bandwidth* , επιτρέποντας στο πρωτόκολλο να προσαρμόσει πολύ πιο εύκολα μεγάλο αριθμό ενεργών καναλιών μεταξύ των *τερματισμών* (*endpoints*)

Το Μέλλον του Πρωτοκόλλου

Από τότε που το IAX βελτιστοποιήθηκε μόνο για την φωνή , υπήρξαν σχόλια για την όχι καλή υποστήριξη του σε βίντεο. Στην πραγματικότητα το IAX μπορεί να υποστηρίξει πολύ καλύτερα οποιοδήποτε *media stream* θελήσουμε. Αυτό γιατί , όπως είπαμε είναι ένα ανοιχτό πρωτόκολλο και μπορεί να ενσωματώσει όποιο τύπο *media* θελήσουμε.

Θέματα Ασφαλείας

Το IAX υποστηρίζει 3 τύπους αυθεντικοποίησης: *plain text* , *MD5 hashing* και ανταλλαγή κλειδιών *RSA*. Αυτά βέβαια δεν κρυπτογραφούν τα δεδομένα ροής που διακινούνται μεταξύ των *endpoints*. Για να μπορέσει να γίνει αυτό χρησιμοποιούμε άλλες λύσεις , όπως ένα *VPN* (*Virtual Private Network*) προσαρμογέα ή κάποιο λογισμικό για να μπορέσει να κρυπτογραφήσει την πληροφορία σε διαφορετικής

τεχνολογίας layer. Στο μέλλον ίσως το IAX να μπορεί να κρυπτογραφήσει της ροές μεταξύ των endpoints με διάφορους τρόπους , όπως για παράδειγμα με έναν διακομιστή RSA κλειδιών.

IAX και NAT

Το IAX σκόπιμα σχεδιάστηκε για να λειτουργεί πίσω από συσκευές που βρίσκονται σε NAT. Η χρήση μία μόνο πόρτας UDP για ταυτόχρονη σηματοδότηση (*signaling*) και αποστολή πληροφορίας κρατάει τον αριθμό των πόρτων που χρειάζεται να είναι ανοιχτές στο Firewall στο ελάχιστο. Αυτοί οι λόγοι καθιστούν το IAX ένα από τα πιο εύκολα (ίσως το πιο εύκολο) πρωτόκολλο για την υλοποίηση του σε ασφαλή δίκτυα.

2.2.2 SIP

Το SIP (Session Initiation Protocol) έχει κατακτήσει τον κόσμο του VoIP. Αρχικά θεωρήθηκε κάτι παραπάνω από μια ενδιαφέρουσα ιδέα. Σήμερα το SIP δείχνει να έτοιμο να εκθρονίσει το εξαιρετικό H.323 , ως πρωτόκολλο της επιλογής μας , ειδικά στα *endpoints* του δικτύου. Η ουσία αυτού του πρωτοκόλλου , είναι ότι κάθε άκρο της σύνδεσης είναι μέρος του δικτύου (*peer*) και έτσι το πρωτόκολλο διαχειρίζεται τις δυνατότητες του μεταξύ αυτών. Αυτό που κάνει το SIP αναγκαίο είναι η απλότητα του πρωτοκόλλου , καθώς έχει πολύ απλή σύνταξη η οποία μοιάζει με τα πολύ γνωστά HTTP και SMTP.

Το SIP υποστηρίζεται από τον ASTERISK με το επιπρόσθετο *chan_sip.so*

Ιστορία του Πρωτοκόλλου

Το SIP παρουσιάστηκε στο Internet Engineering Task Force (IETF) τον Φεβρουάριο του 1996 ως “ draft-ietf-mmusic-sip-00 “. Το αρχικό πρωτόκολλο δεν είχε καμιά σχέση με αυτό που ξέρουμε σήμερα και είχε μόνο μία απαίτηση: απαίτηση για call setup. Μετά από 11 αναθεωρήσεις , φτάσαμε στον Μάρτιο του 1999 , όπου γεννήθηκε το SIP RFC 2443.

Στην αρχή το SIP αγνοήθηκε , γιατί το H.323 θεωρήθηκε ως το πρωτόκολλο για την επικοινωνία μέσω VoIP. Ενώ όμως , η διάδοση του μεγάλωνε , το SIP άρχισε να γίνεται όλο και πιο δημοφιλές και παρόλο τους διαφορετικούς παράγοντες που συνέβαλαν σε αυτό , ο πιο σημαντικός είναι οι ελεύθερες προδιαγραφές του.

Το Μέλλον του Πρωτοκόλλου

Το SIP κέρδισε την θέση του , ως το πρωτόκολλο που ταιριάζει στο VoIP. Όλοι οι νέοι χρήστες και τα επαγγελματικά προϊόντα , περιμένουν να υποστηρίξουν το SIP . Το SIP αναμένεται να υποστηρίξει πολλά παραπάνω από VoIP , όπως μετάδοση βίντεο , μουσικής και διάφορους τύπους πολυμέσων πραγματικού χρόνου. Το SIP είναι ένα ισορροπημένο πρωτόκολλο που μπορεί να διαχειριστεί την πλειοψηφία των καινούργιων εφαρμογών μέσα στα επόμενα χρόνια.

Θέματα Ασφαλείας

Το σύστημα που χρησιμοποιεί το SIP για να αυθεντικοποιεί τους χρήστες είναι ένα challenge/response. Ένα αρχικό *INVITE* στέλνεται στον proxy από την συσκευή που θέλει να επικοινωνήσει. Ο proxy στην συνέχεια στέλνει πίσω ένα *407 Proxy Authorization* μήνυμα , το οποίο περιέχει μία ομάδα τυχαίων χαρακτήρων που αναφέρονται ως “nonce” (μοναδικοί). Αυτοί οι μοναδικοί χαρακτήρες χρησιμοποιούνται μαζί με το password για να δημιουργήσουν ένα MD5 hash , το οποίο έπειτα ξαναστέλνεται πίσω ακολουθούμενο από το *INVITE*. Έπειτα το MD5 hash συγκρίνεται με αυτό που δημιούργησε ο proxy και έτσι αυθεντικοποιείται ο client.

Η Άρνηση Λειτουργίας (*Denial of Service*) είναι μια επίθεση πολύ κοινή στις VoIP επικοινωνίες. Μία επίθεση *DoS* μπορεί να δημιουργηθεί όταν μεγάλος αριθμός λανθασμένων *INVITE* στέλνονται στον proxy server με σκοπό να τον θέσουν εκτός λειτουργίας. Αυτές οι επιθέσεις είναι εύκολο να πραγματοποιηθούν και οι επιπτώσεις στους χρήστες εμφανίζονται άμεσα. Το SIP έχει πολλούς τρόπους για να ελαχιστοποιήσει τις επιδράσεις του *DoS* , αλλά εν τέλη είναι δύσκολο να αποφευχθούν.

Το SIP υλοποιεί ένα σχήμα για να εγγυηθεί ένα ασφαλές μηχανισμό κρυπτογραφημένης μετάδοσης (ονομάζεται *Transport Layer Security* ή αλλιώς *TLS*) , το οποίο χρησιμοποιείται για να πραγματοποιηθεί επικοινωνία μεταξύ του καλούντα και του domain του καλούντα. Πέρα από αυτό, το αίτημα (*request*) στέλνεται ασφαλές στο άλλο άκρο , σύμφωνα με αυτά που ορίζει η τοπική ασφάλεια του δικτύου. Να σημειώσουμε εδώ ότι η κρυπτογράφηση των media (όπως είναι η ροή *RTP*) είναι πέρα από το αντικείμενο του ίδιου του SIP και πρέπει να αναλυθεί ξεχωριστά.

SIP και NAT

Προφανώς το μεγαλύτερο εμπόδιο που πρέπει να κατακτήσει το SIP είναι να εκτελέσει τις διεργασίες του σε επίπεδο NAT. Επειδή το SIP συμπεριλαμβάνει πληροφορίες διευθύνσεων , μέσα στα data frames και επειδή το NAT βρίσκεται σε χαμηλότερο layer δεν έχει την δυνατότητα να αλλάξει αυτές τις διευθύνσεις με αποτέλεσμα οι ροές δεδομένων να μην έχουν την σωστή πληροφορία , ώστε να δημιουργήσουν σύνδεση όταν υπάρχει NAT. Επιπρόσθετα με αυτό όταν υπάρχει firewall μαζί με το NAT , δεν μπορεί να καταλάβει ότι οι εισερχόμενες ροές , είναι μέρος των διαδικασιών του SIP , με αποτέλεσμα να τις απορρίπτει.

2.2.3 H.323

Αυτό το πρωτόκολλο σχεδιάστηκε από την *International Telecommunication Union* (*ITU*) για να μπορέσει να υποστηρίξει την μεταφορά βίντεο κλήσης. Έγινε το στάνταρτ στα συστήματα που έκαναν βίντεο κλήση μέσω IP και έτσι σύντομα έγινε πολύ διάσιμο και στις επικοινωνίες VoIP. Ενώ υπάρχει μεγάλη διαμάχη για το ποιο πρωτόκολλο θα κυριαρχήσει στο VoIP (SIP , H.323 , IAX) , στον ASTERISK το H.323 αποδοκιμαζόταν για χάρη του IAX και του SIP. Το H.323 δεν γνώρισε μεγάλη επιτυχία από τους χρήστες και από τις επιχειρήσεις αν και είναι ακόμα το ποιο ευρέως χρησιμοποιούμενο VoIP πρωτόκολλο μεταξύ των φορέων.

Να πούμε εδώ ότι όλοι μας έχουμε χρησιμοποιήσει το H.323 , χωρίς να το ξέρουμε. Το Microsoft NetMeeting χρησιμοποιεί ευρέως αυτό το πρωτόκολλο.

Οι 2 εκδόσεις του H.323 υποστηρίζονται από τον Asterisk και διαχειρίζονται από τα επιπρόσθετα *chan_h323.so* (παρέχεται από τον ASTERISK) και *chan_oh323.so* (παρέχεται σαν δωρεάν επιπρόσθετο).

Ιστορία του Πρωτοκόλλου

Το H.323 αναπτύχθηκε από την ITU τον Μάιο του 1996 ως ένα μέσο για την μετάδοση , φωνής , βίντεο , δεδομένων και φαξ δια μέσου ενός δικτύου IP , διατηρώντας σύνδεση με το PSTN. Από εκείνη την στιγμή το H.323 άλλαξε πολλές εκδόσεις και ταυτόχρονα δυνατότητες επιτρέποντας του να λειτουργεί σε δίκτυα VoIP και σε ποιο ευρεία διαδεδομένα δίκτυα.

Το Μέλλον του Πρωτοκόλλου

Το μέλλον του H.323 είναι μια θερμή διαμάχη. Το H.323 θεωρείται τεχνικά ανώτερο από το SIP , αλλά καθώς και τόσες άλλες τεχνολογίες , που εν τέλει να μην έχει καμιά σημασία. Ένας από τους παράγοντες που κάνουν το H.323 μη δημοφιλές είναι η πολυπλοκότητα του , αν και πολλοί συμφωνούν ότι το άλλοτε εύκολο SIP , αρχίζει να πάσχει από το ίδιο πρόβλημα.

Το H.323 κρατιέται ακόμα από το πλήθος των παγκόσμιων μέσων VoIP κίνησης , αλλά οι άνθρωποι πλέον γίνονται όλο και πιο λίγοι που εξαρτώνται από παραδοσιακά μέσα για τις τηλεπικοινωνίες τους. Το μέλλον του H.323 έχει γίνει δύσκολο να προβλεφθεί. Εφόσον το H.323 δεν είναι πρωτόκολλο επιλογής για νέες υλοποιήσεις θα περιμένουμε το H.323 να συμφωνήσει να συνεργαστεί με διάφορα θέματα στο μέλλον.

Θέματα Ασφαλείας

Αναφορικά με την ασφάλεια που προσφέρει το H.323 , μπορούμε να πούμε ότι είναι ένα ασφαλές πρωτόκολλο και δεν απαιτεί ιδιαίτερες μελέτες για ασφάλεια παραπάνω από αυτές που είναι σε οποιοδήποτε δίκτυο επικοινωνιών στο Internet. Από την στιγμή που το H.323 χρησιμοποιεί το *RTP* πρωτόκολλο για επικοινωνίες

media , δεν υποστηρίζει φυσικά κρυπτογραφημένες media διαδρομές. Η χρήση *VPN* ή άλλων κρυπτογραφημένων *tunnels* μεταξύ των άκρων , είναι ένας τρόπος για να εξασφαλίσουμε ασφάλεια στις επικοινωνίες μας. Φυσικά και αυτό έχει αρνητικά , καθώς η εγκατάσταση τέτοιων *tunnels* μεταξύ των άκρων είναι δύσκολη και πολλές φορές ακατόρθωτη. Καθώς το VoIP φτάνει σε σημείο να το χρησιμοποιούν οικονομικοί οργανισμοί , είναι πιθανόν να χρειαστούν τα πρωτόκολλα VoIP να προσαρμοστούν ώστε να υποστηρίζουν δυνατές κρυπτογραφικές μεθόδους.

H.323 και NAT

Το H.323 χρησιμοποιεί το *Internet Engineering Task Force* (IETF) RTP πρωτόκολλο για να μεταδώσει media μεταξύ των άκρων. Εξαιτίας αυτού το H.323 έχει τα ίδια θέματα με αυτά του SIP που έχουν να κάνουν σχέση με ο NAT. Το πιο απλό που μπορεί να γίνει είναι να προωθηθούν οι κατάλληλες πόρτες από το NAT στους αντίστοιχους εσωτερικούς clients.

Για να δεχθούμε εισερχόμενη κλήση πρέπει πάντα να προωθούμε την πόρτα *TCP 1720* στον *client*. Επίσης χρειάζεται να προωθήσουμε τις UDP πόρτες για τα RTP media και για τα RTCP control streams.

Εάν έχουμε ένα μεγάλο αριθμό clients πίσω από NAT , θα πρέπει να χρησιμοποιήσουμε ένα *gatekeeper* , ο οποίος θα δουλεύει σε λειτουργία *proxy*. Ο *ASTERISK* δεν μπορεί να λειτουργήσει σαν *gatekeeper* για το H.323. Θα πρέπει να εγκαταστήσουμε μια εξωτερική εφαρμογή για να λειτουργήσει.

2.3 Codecs

Οι *codecs* είναι διάφορα μαθηματικά μοντέλα που αναλαμβάνουν να κωδικοποιήσουν και να συμπιέσουν ψηφιακά , αναλογική πληροφορία ήχου.

Στην πραγματικότητα ο όρος *CODEC* βγαίνει από τους όρους *COder / DECoder* : μία συσκευή που μετατρέπει από αναλογικό σε ψηφιακό σήμα. Σήμερα ο όρος ταιριάζει καλύτερα στο *COmpression / DECOpession*.

Πριν αναλύσουμε με την σειρά τους *codecs* θα δούμε μία σύνοψη όλων των *codec* με τα χαρακτηριστικά τους.

Codec	Data Bitrate (klbps)	Χρειάζονται Άδεια ?
G.711	64 kbps	Όχι
G.726	16 , 24 ή 32 kbps	Όχι
G.723.1	5.3 ή 6.3 kbps	Ναι
G.729A	8 kbps	Ναι
GSM	13 kbps	Όχι

Πίνακας 2.1 Codecs και χαρακτηριστικά τους

2.3.1 G.711

Το G.711 είναι ο θεμελιώδης codec για το PSTN. Στην πραγματικότητα εάν κάποιος αναφερθεί στο PCM σε σχέση με τηλεφωνικό δίκτυο , μπορούμε να σκεφτούμε το G.711. Χρησιμοποιούνται 2 μέθοδοι προσαρμογής (*companding*) : μ-law στην Β. Αμερική και A-law στον υπόλοιπο κόσμο. Και τα δύο δημιουργούν μια λέξη 8-bit η οποία μεταδίδεται 8000 φορές το δευτερόλεπτο. Αν κάνουμε τις πράξεις παρατηρούμε ότι χρειαζόμαστε 64000 bits το δευτερόλεπτο.

Πολλοί νομίζουν ότι το G.711 δεν κάνει συμπίεση. Αυτό δεν είναι τελείως σωστό , καθώς η προσαρμογή θεωρείται ένα είδος συμπίεσης. Αυτό που ισχύει , είναι ότι το G.711 είναι ο βασικός codec από τον οποίο , όλοι οι υπόλοιποι προέρχονται.

2.3.2 G.726

Αυτός ο Codec υπήρξε στο περιθώριο για λίγο καιρό (χρησιμοποιούταν ως G.721 , ο οποίος τώρα είναι απαρχαιωμένος) και είναι ένας από τους πρώτους codec συμπίεσης. Επίσης είναι γνωστός ως *Adaptive Differential Pulse-Code Modulation* (*ADPCM*) και μπορεί να τρέξει σε διάφορους ρυθμούς μετάδοσης. Οι πιο κοινοί ρυθμοί είναι 16 kbps , 24 kbps και 32 kbps. Αυτή την στιγμή ο ASTERISK χρησιμοποιεί μόνο τον ρυθμό ADPCM-32 , ο οποίος είναι μακράν ο πιο δημοφιλής ρυθμός για αυτόν τον codec.

Ο G.726 προσφέρει σχεδόν την ίδια ποιότητα με αυτήν του G.711 , όμως με το μισό εύρος. Αυτό συμβαίνει γιατί , αντί να στέλνει τα αποτελέσματα από την μέτρηση

της κβαντοποίησης , στέλνει μόνο την πληροφορία που χρειάζεται για να μπορέσει να περιγράψει την διαφορά του τρέχον δείγματος με το προηγούμενο. Το G.726 έπεσε από την εύνοια στις αρχές της δεκαετίας του 1990 , εξαιτίας της αδυναμίας του να μεταφέρει σήμα μέσω modem ή fax. Σήμερα αρχίζει να κάνει την επιστροφή του , λόγω του πολύ καλού λόγου απόδοσης “ *ΕΥΡΟΥΣ / ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΙΣΧΥΣ* “. Το G.726 προτιμάται για την ελάχιστη υπολογιστική δουλειά που χρειάζεται για να λειτουργήσει από το σύστημα.

2.3.3 G.723.1

Για να μην υπάρξει σύγχυση με το G.723 , το οποίο είναι και αυτό ένα απαρχαιωμένη έκδοση του ADPCM , αυτός ο codec σχεδιάστηκε για πολύ μικρή ροή μετάδοσης φωνής. Έχει 2 ρυθμίσεις ρυθμού μετάδοσης δεδομένων , μία 5.3 kbps και μία 6.3 kbps. Το G.723.1 είναι ένας codec που απαιτείται για να συνεργαστεί με το πρωτόκολλο H.323 (αν και μπορούν και άλλοι codec να το κάνουν αυτό). Αυτή την στιγμή είναι φορτωμένος με διάφορες πατέντες και γι αυτό απαιτείται άδεια για την χρησιμοποίηση του σε εμπορικές εφαρμογές. Αυτό σημαίνει ότι ενώ μπορούμε να κάνουμε 1 κλήση με 2 codec G.723. 1 με τον ASTERISK , δεν μπορούμε να κάνουμε αποκωδικοποίηση εάν δεν έχουμε άδεια χρήσης.

2.3.4 G.729.A

Το G.729.A προσφέρει εντυπωσιακή ποιότητα ήχου , χρησιμοποιώντας ελάχιστο εύρος. Αυτό επιτυγχάνεται διαμέσου της μεθόδου συμπίεσης φωνής: *Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP)*. Εξαιτίας λοιπόν των ευρεσιτεχνιών που διαθέτει , δεν μπορούμε να τον χρησιμοποιήσουμε , χωρίς να έχουμε πληρώσει την άδεια χρήσης του. Ωστόσο είναι ιδιαίτερα δημοφιλής και γι αυτό υποστηρίζεται από διάφορα συστήματα φωνής.

Για να μπορέσει να επιτύχει αυτόν τον εντυπωσιακό λόγο συμπίεσης , αυτός ο codec χρειάζεται εξίσου ένα μεγάλο ποσό υπολογιστικής ισχύς. Σε ένα σύστημα με ASTERISK εάν χρησιμοποιήσουμε ένα τέτοιο codec , το σύστημα μας θα “γονατίσει” πολύ σύντομα. Το G.729.A χρειάζεται μόνο 8 kbps εύρους.

2.3.5 GSM

Ο GSM είναι ο αγαπημένος codec του ASTERISK. Αυτός ο codec δεν χρειάζεται άδεια χρήσης για να τον χρησιμοποιήσουμε όπως γίνεται με το G.723.1 και το G.729.A. Προσφέρει αξιοσημείωτες επιδόσεις με γνώμονα την ελαχιστοποίηση της ζήτησης σε απαιτήσεις της CPU. Η ποιότητα του ήχου, θεωρείται χαμηλότερου επιπέδου από αυτήν που προσφέρει ο G.729.A, αλλά αυτό εξαρτάται και από την προσωπική γνώμη του καθενός.

Το GSM χρειάζεται για να λειτουργήσει 13 kbps εύρους.

2.3.6 MP3

Το MP3 σίγουρα είναι ένας codec. Συγκεκριμένα ονομάζεται *Moving Picture Experts Group Audio Layer 3 Encoding Standard*. Με τέτοιο όνομα κανείς δεν απορεί γιατί το λέμε MP3. Στον ASTERISK το MP3 χρησιμοποιείται ιδιαίτερα στο *Music on Hold (MoH)* ή αλλιώς Μουσική Κατά την Αναμονή. Το MP3 δεν είναι ένας τηλεφωνικός codec. Έχει βελτιστοποιηθεί για την μουσική και όχι για την φωνή. Ωστόσο είναι πολύ διάσημος στα VoIP συστήματα για τον λόγο που αναφέραμε προηγουμένως, δηλαδή το Music on Hold.

2.4 Quality of Service

Quality of Service (QoS) ή αλλιώς Ποιότητα Υπηρεσίας, αναφέρεται στην πρόκληση να παραδοθεί μια χρονικά ευαίσθητη ροή δεδομένων, μέσα σε ένα δίκτυο το οποίο σχεδιάστηκε να παραδίδει δεδομένα ad hoc, με τη καλύτερα δυνατή προσπάθεια στην ελάχιστη διαδρομή. Αν και δεν υπάρχει κάποιος αυστηρός κανόνας, γενικά αποδεχόμαστε ότι ο ήχος που θα παραχθεί από το μικρόφωνο, θα φτάσει στο αντί του αποδέκτη μέσα σε 300 millisecond και έτσι θα υπάρξει μια ομαλή συνομιλία. Αν η καθυστέρηση του ήχου ξεπεράσει τα 500 millisecond, τότε

είναι δύσκολο να αποφύγεις διακοπές του ήχου μεταξύ της συνομιλίας. Εάν η καθυστέρηση ξεπεράσει το 1 δευτερόλεπτο , τότε η συνομιλία είναι ανυπόφορη.

Επιπλέον για να μπορέσουμε να το επιτύχουμε αυτό στην ώρα του , θα πρέπει να βεβαιωθούμε ότι η μεταδιδόμενη πληροφορία , φθάνει στον προορισμό της , άθικτη. Η μεγάλη απώλεια πακέτων , θα αποκλείσει την ολοκληρωτική αναπαραγωγή των δειγμάτων ήχου και τα κενά που θα δημιουργηθούν στην πληροφορία , θα δημιουργούν στην συνομιλία , χάσιμο λέξεων ή ακόμα και ολόκληρων προτάσεων.

2.4.1 TCP , UDP και SCTP

Εάν , πρόκειται να στείλουμε πληροφορία , σε ένα δίκτυο IP , θα πρέπει να χρησιμοποιήσουμε ένα από τα πρωτόκολλα μετάδοσης που αναγράφονται στην επικεφαλίδα της παραγράφου.

Transmission Control Protocol (TCP)

Το TCP , σχεδόν ποτέ δεν έχει χρησιμοποιηθεί για VoIP. Αυτό συμβαίνει γιατί , αν και έχει τους μηχανισμούς για να παρέχει μια ασφαλή παράδοση των πακέτων , δεν έχει την δυνατότητα να βιαστεί για να το πετύχει. Αυτό έχει σαν αποτέλεσμα να δημιουργούνται μεγάλοι χρόνοι υστέρησης και έτσι το TCP δημιουργεί περισσότερα προβλήματα απ' ό τι επιλύει.

Σκοπός του TCP , είναι να παραδίδει τα πακέτα. Για να μπορέσει να το κάνει αυτό , πολλοί μηχανισμοί εκτελούνται , όπως το packet numbering (για επαναδημιουργία των πακέτων) , delivery acknowledgment (επιβεβαίωση παράδοσης) και re-request (επανάληψη ζήτησης) χαμένων πακέτων. Στον κόσμο του VoIP , κυριαρχεί το να φτάνουν τα πακέτα στον προορισμό τους γρήγορα.

Το TCP δουλεύει πολύ καλά στην μετάδοση μεγάλου όγκου πληροφορίας , αλλά δεν είναι αρκετά αποδοτικό για media επικοινωνίες πραγματικού χρόνου (real time).

User Data Protocol

Αντίθετα με το TCP το *User Data Protocol* (*UDP*) δεν παρέχει καμιά ταξινόμηση κατά την παράδοση. Τα πακέτα στέλνονται μέσα από το φυσικό μέσο , όσο το δυνατόν γρηγορότερα και αυτά να αναλαμβάνουν να βρουν τον προορισμό τους , χωρίς να το νοιάζει , εάν θα φτάσουν ή όχι. Καθώς το UDP δεν προσφέρει καμιά εγγύηση , εάν θα φτάσουν τα πακέτα , αυτό που το κάνει αποτελεσματικό , είναι η πολύ μικρή προσπάθει που καταβάλει για να επιτύχει την μεταφορά των πακέτων.

Stream Control Transmission Protocol (SCTP)

Εγκρίθηκε από τη IETF προτείνοντας το σαν πρότυπο στο RCF 2960 , το SCTP είναι σχετικά ένα καινούργιο πρωτόκολλο. Από την αρχή σχεδιάστηκε για να εξαλείψει τις αδυναμίες των TCP και UDP , ιδιαίτερα στους τύπους των υπηρεσιών που σχετίζονται με τα τηλεφωνικά δίκτυα μεταγωγής κυκλώματος (circuit – switch).

Μερικοί από τους στόχους του SCTP είναι :

- Τεχνικές αποφυγής συμφόρησης (congestion avoidance) , ειδικά σε επιθέσεις Άρνησης Λειτουργίας (Denial Of Service)
- Αυστηρή ακολουθία παράδοσης δεδομένων
- Μικρή υστέρηση για βελτίωση μετάδοσης δεδομένων πραγματικού χρόνου.

Παρακάμπτοντας τα μειονεκτήματα των TCP και UDP , αυτοί που αναπτύσσουν το SCTP ελπίζουν να δημιουργήσουν ένα δυνατό πρωτόκολλο για την μετάδοση SS7 και άλλους τύπους PSTN σηματοδοσίας , πάνω από δίκτυα IP.

2.4.2 Διαφορική Υπηρεσία (Differentiated Service)

Οι Διαφορικές Υπηρεσίες (Differentiated Service) ή αλλιώς DiffServ , δεν είναι ακριβώς ένας μηχανισμός QoS που θα παρακολουθεί την κίνηση και θα την διαχειρίζεται αναλόγως. Προφανώς το DiffServ βοηθάει να υπάρχει το QoS ,

επιτρέποντας διάφορους τύπους πακέτων να παίρνουν προτεραιότητα έναντι των άλλων. Αν και αυτό θα αυξήσει σίγουρα την πιθανότητα σε ένα VoIP πακέτο να περάσει γρήγορα , μέσα από κάθε link , δεν εγγυάται τίποτα.

2.4.3 Εγγυημένη Υπηρεσία (Guaranteed Service)

Η απόλυτη εγγύηση του QoS παρέχεται από το PSTN. Για κάθε συνομιλία , ένα κανάλι 64 kbps είναι αφιερωμένο στην κλήση και έτσι το bandwidth που χρειάζεται αυτή είναι εγγυημένο. Παρόμοια πρωτόκολλα που προσφέρουν Guaranteed Service, εξασφαλίζουν ένα απαιτούμενο ποσό εύρους , το οποίο εξασφαλίζεται για να πραγματοποιηθεί η κλήση. Όπως με το οποιοδήποτε δίκτυο που χρησιμοποιεί τεχνολογία πακετοποίησης , έτσι και αυτοί οι μηχανισμοί λειτουργούν καλύτερα όταν η κίνηση είναι λίγο παρακάτω από τα μέγιστα επιτρεπτά επίπεδα. Όταν μία σύνδεση φτάνει στα όρια της , μετά είναι αδύνατο να επέλθει κατάρρευση.

2.4.4 Μέγιστη Προσπάθεια (Best Effort)

Η ευκολότερη και ταυτόχρονα λιγότερη δαπανηρή προσέγγιση στο QoS δεν είναι να τα εξασφαλίσεις όλα από την αρχή , αλλά είναι η μέθοδος Best Effort. Ενώ αυτό θα μπορούσε να είναι κακή ιδέα , στην πραγματικότητα μπορεί να δουλέψει πολύ καλά. Οποιαδήποτε VoIP κλήση που διέρχεται μέσω του Internet , είναι βέβαιο χρειάζεται την μέγιστη προσπάθεια , καθώς οι μηχανισμοί QoS δεν είναι κοινοί σε αυτό το περιβάλλον.

2.5 Echo (ηχώ)

Η ηχώ υπήρξε σοβαρό πρόβλημα για το PSTN. Οι τηλεφωνικές βιομηχανίες επένδυσαν πολλά λεφτά , στο να φτιάξουν πανάκριβα συστήματα τα οποία τα εξαλείφουν το φαινόμενο αυτό.

2.5.1 Γιατί Υπάρχει Η Ηχώ

Εάν ακούσουμε ηχώ , δεν το προκαλεί το τηλέφωνο μας. Είναι το απομακρυσμένο άκρο του κυκλώματος. Αντίθετα , όταν ακουστεί ηχώ στο απομακρυσμένο άκρο έχει δημιουργηθεί από το δικό μας άκρο. Η ηχώ προκαλείται από τον λόγο σε ένα τοπικό αναλογικό βρόγχο θα πρέπει η αποστολή και η λήψη , να πραγματοποιηθούν από το ίδιο ζεύγος καλωδίου. Εάν αυτό το κύκλωμα δεν είναι ηλεκτρικά ισορροπημένο ή συνδεθεί μιας χαμηλής ποιότητας συσκευή στο άκρο του κυκλώματος , τότε τα σήματα που λαμβάνει μπορούν να ανακλώνται πίσω και γίνονται μέρος της μετάδοσης. Μόλις αυτή η ανάκλαση γυρίσει πίσω σε εσένα τότε θα ακούσουμε τα λόγια που μόλις είπαμε. Το ανθρώπινο αυτί , μπορεί να αντιλαμβάνεται μία ηχώ μετά από μια καθυστέρηση της τάξεως των 40 millisecond.

Σε ένα φτηνό τηλέφωνο είναι πιθανό να δημιουργηθεί ηχώ μέσω του ίδιου του ακουστικού. Γι αυτό μπορεί να δημιουργηθεί ηχώ σε μερικά φτηνά IP τηλέφωνα ακόμα και όταν ολόκληρη η διασύνδεση δεν περιέχει αναλογικό κύκλωμα. Στην πραγματικότητα το ακουστικό στα IP τηλέφωνα είναι αναλογικό κύκλωμα. Στον κόσμο του VoIP η ηχώ εμφανίζεται μόνο εάν υπάρχει κάποιο αναλογικό κύκλωμα ανάμεσα στην διασύνδεση ή εάν υπάρχει φθηνός εξοπλισμός στα άκρα , ο οποίος ανακλά το σήμα. Ένας καλός κανόνας είναι να κρατάμε της υστέρηση σε λιγότερο από 250 millisecond.

2.5.2 Διαχείριση Echo Μέσω του ASTERISK

Την ηχώ μπορούμε να την διαχειριστούμε στον Asterisk μέσα από το αρχείο *zconfig.h* , όπου εκεί θα βρούμε πάρα πολλούς αλγόριθμους για ακύρωση ηχούς (*echo cancelation*), με προεπιλεγμένο τον *MARK2*. Στο δικό μας δίκτυο μπορούμε να κάνουμε δοκιμή και να επιλέξουμε τον καταλληλότερο. Επίσης μέσα από αυτό το αρχείο μπορούμε να ρυθμίσουμε πόσο αυστηρός (*aggressive*) θα είναι ο αλγόριθμος. Να σημειώσουμε εδώ ότι το αυστηρό *echo cancelation* , μπορεί να δημιουργηθεί στην ομιλία μας φαινόμενο *walkie-talkie* , δηλαδή μονόπλευρη συνομιλία. Άρα αυτή η επιλογή πρέπει να την δοκιμάζουμε , εφόσον δεν μας έχουν λειτουργήσει οι υπόλοιπες.

Όταν ενεργοποιείται το `echo cancelation` , ο αλγόριθμος διαβάζει την ηχώ της γραμμής , ακούγοντας την όλη την διάρκεια της κλήσης. Συνεπώς η ηχώ , μπορεί να ακουστεί στην αρχή της κλήσης και τελικά να εξαφανιστεί μετά από λίγη ώρα. Για να αποφύγουμε αυτή την κατάσταση , μπορούμε να χρησιμοποιήσουμε μία μέθοδο που ονομάζεται “ `echo training` “. Αυτή η μέθοδος μειώνει τον ήχο στην γραμμή για μικρό χρονικό διάστημα και μετά στέλνει ένα τόνο , από τον οποίο το ποσό της ηχούς που θα εμφανιστεί να αφαιρεθεί πλήρως. Αυτό επιτρέπει στον `ASTERISK` να διαχειριστεί την ηχώ , πολύ πιο γρήγορα. Η μέθοδος `echo training` μπορεί να ενεργοποιηθεί μέσα στο αρχείο `zconfig.h` , σημειώνοντας την γραμμή “ `echotraining = yes` “.

3. Ρύθμιση του Asterisk^([1],[2],[5])

3.1 Εισαγωγή

Αφού ήδη αναφέραμε στο προηγούμενο κεφάλαιο τα πρωτόκολλα που χρησιμοποιεί ο Asterisk , σε αυτό το κεφάλαιο θα αναφέρουμε τα αρχεία που εμπλέκονται και υποδεικνύουν τον τρόπο λειτουργίας αυτών των πρωτοκόλλων. Επίσης θα αναφέρουμε και άλλες ρυθμίσεις που χρειάζεται να προβούμε για να λειτουργήσει το PBX μας , σύμφωνα με τις ανάγκες μας. Η παραμετροποίηση γίνεται εύκολα αλλάζοντας διάφορα configuration files που υπάρχουν για αυτόν τον σκοπό. Τα configuration files υπάρχουν μέσα στον φάκελο “ /etc/asterisk “ και αλλάζονται εύκολα με κάποιον επεξεργαστή κειμένου .

Δεν θα μπούμε σε λεπτομέρειες για τον τρόπο εγκατάστασης του Asterisk ή την δομή των αρχείων μέσα στο σύστημα μας , γιατί κάτι τέτοιο θα ξέφευγε από το πνεύμα αυτής της διπλωματικής εργασίας.

3.2 Ρυθμίζοντας το SIP

Όπως έχουμε ήδη αναφέρει , το SIP χρησιμοποιείται συχνά στα VoIP Phones (είτε στα *Hard Phones* είτε στα *Soft Phones*) , αναλαμβάνοντας την έναρξη και την λήξη μιας κλήσης και ταυτόχρονα όλη την διαδικασία κατά την διάρκεια αυτής. Θα αναλύσουμε το αρχείο που είναι υπεύθυνο για να υποδείξει στον Asterisk τι

ρυθμίσεις πρέπει να ακολουθήσει όταν χρησιμοποιήσει το πρωτόκολλο SIP. Το όνομα του αρχείου που ορίζει το sip είναι το “ *sip.conf* “. Παρακάτω φαίνεται ένα τυπικό αρχείο SIP:

```
[general]
context = default
bindport=5060
disallow=all
allow=g723

[cisco]
type=friend
secret=mysecret
callerid="cisco"<USER1>
host=dynamic
context=trusted
nat=yes
canreinvite=no
quality=yes

[xlite]
type=friend
secret=xlite
host=dynamic
defaultip=10.1.30.17
```

Το αρχείο περιέχει 2 είδη περιεχομένων (sections) : Το Γενικό (general) και το Περιεχόμενο του κάθε client.

I. General Section

Το αρχείο SIP, διαβάζεται από πάνω προς τα κάτω. Το πρώτο περιεχόμενο που πρέπει να ορίσουμε , είναι οι γενικές ρυθμίσεις που πρέπει να ακολουθηθούν. Ορίζεται ως *[general]* και ακολουθούν οι ρυθμίσεις του καναλιού και προκαθορισμένες επιλογές για όλους τους χρήστες που ορίζονται στο αρχείο.

- *context* : Αναφέρεται στη θέση των οδηγιών που χρησιμοποιούνται για τον έλεγχο των εισερχόμενων και εξερχόμενων κλήσεων της συγκεκριμένης extension. Εδώ έχουμε θέση μία γενική τιμή και την έχουμε ονομάσει *default* .
- *bindport* : Αναφερόμαστε στην UDP πόρτα που πρέπει να χρησιμοποιήσει ο Asterisk. Η προεπιλεγμένη είναι η *5060*
- *allow / disallow* : Αναφέρετε στους codec που μπορεί να επιτρέψει ή να αποκλείσει το PBX μας. Με *all* , είναι όλοι , αλλιώς βάζουμε αυτόν που θέλουμε , όπως *g723* , *gsm* , *g711* κτλ.

II. Client Section

Αφού ορίσαμε τις γενικές ρυθμίσεις , ήρθε η ώρα να ορίσουμε τις ρυθμίσεις για τους SIP χρήστες. Το context (περιεχόμενο) ενός χρήστη ορίζεται με το όνομα του μέσα σε τετραγωνικά άγκιστρα [] και τελειώνει εκεί που ξεκινάει το επόμενο. Η έννοια του context είναι αρκετά σημαντική για τη λειτουργία του Asterisk. Θα θεωρήσουμε ότι το όνομα του context είναι και το user name του χρήστη. Άρα στο παράδειγμα μας έχουμε 2 clients με όνομα *cisco* και *xlite*. Οι υπόλοιπες ρυθμίσεις , φαίνονται παρακάτω.

- *type* : Ο τύπος του χρήστη (*type=friend*) μας λέει ότι ο *cisco* χρήστης μπορεί να δεχθεί και να πραγματοποιήσει κλήσεις. Αν θέλουμε ο

χρήστης μόνο να δέχεται κλήσεις τότε αλλάζουμε τον τύπο του σε peer (type=peer) και αν θέλουμε μόνο να πραγματοποιεί σε type=user

- *secret* : Η γραμμή secret=mysecret μας δείχνει ότι το συνθηματικό που θα χρησιμοποιηθεί για την αυθεντικοποίηση του χρήστη cisco χρήστη, είναι η λέξη mysecret
- *callerid* : Εδώ βάζουμε το όνομα που επιθυμούμε οι άλλοι να βλέπουμε όταν τους καλούμε.
- *host* : Σε αυτό το σημείο λέμε στο Asterisk ότι τα τερματικά μας, δεν έχουν σταθερή διεύθυνση IP και ότι χρειάζεται να τη μαθαίνει κάθε φορά που εγγραφόμαστε σε αυτόν. Εναλλακτικά μπορούμε να ορίσουμε μία στατική IP στο χρήστη π.χ. host=10.1.30.17 ή ακόμη και κάποιο domain name.
- *context* : Σε αυτή την ρύθμιση ισχύει ότι έχουμε ήδη πει στο general section.
- *nat* : Αν ο χρήστης είναι πίσω από κάποια συσκευή NAT (Network Address Translation) όπως firewall ή router, η επιλογή αυτή λέει στο Asterisk να αγνοήσει τις πληροφορίες που δέχεται μέσα από το κανάλι της κλήσης (π.χ. source ip) και να χρησιμοποιήσει για επικοινωνία, τη διεύθυνση από την οποία λαμβάνει τα δεδομένα
- *canreinvite* : Η γραμμή canreinvite=no λέει στο Asterisk να μην επιτρέπει στους χρήστες να συνδεθούν απευθείας μεταξύ τους, αλλά πάντα να παρεμβάλετε στην επικοινωνία. Με αλλαγή της επιλογής, σε yes θα συνέβαινε ακριβώς το αντίθετο.
- *quality* : Με την επιλογή qualify=yes μπορούμε να παρακολουθήσουμε την καθυστέρηση της επικοινωνίας και να καταλάβουμε αν το άλλο άκρο είναι προσβάσιμο. Η προεπιλεγμένη τιμή είναι τα 2000 msec, αλλά μπορούμε να την αλλάξουμε αντικαθιστώντας το yes με κάποια αριθμητική τιμή σε msec (π.χ. qualify=3000).

3.3 Ρυθμίζοντας το IAX

Για να μπορέσουμε να επεξεργαστούμε τις ρυθμίσεις για το πρωτόκολλο IAX θα πρέπει να ανοίξουμε το αρχείο “ *iax.conf* “. Παρακάτω φαίνεται ένα τυπικό αρχείο IAX:

```
[general]
bandwidth=low
context=default

[iax-user]
type=friend
secret= mysecret
callerid="iax-user" <USER2>
qualify=yes
host=dynamic
context= trusted
```

Παρόλο που το IAX και το SIP έχουν σημαντικές διαφορές στον τρόπο υλοποίησής τους (έχει ήδη αναφερθεί σε προηγούμενο κεφάλαιο), βλέπουμε ότι στο συγκεκριμένο παράδειγμα οι χρήστες ορίζονται παρόμοια.

Μια σημαντική διαφορά που φαίνεται αμέσως είναι η έλλειψη της γραμμής *nat*. Το IAX έχει σχεδιαστεί με τέτοιο τρόπο ώστε να μην επηρεάζεται σοβαρά από την τοπολογία του δικτύου κάνοντας έτσι πιο εύκολη τη διαδικασία της εγγραφής του πελάτη στον εξυπηρετητή.

Άλλη μία διαφορά που προκύπτει από τα παραπάνω αρχεία είναι η επιλογή *bandwidth=low* που περιέχεται στο [general] context. Η επιλογή αυτή αφορά τους codecs που θα χρησιμοποιηθούν στην επικοινωνία. Επιβάλλουμε δηλαδή στο Asterisk να μη χρησιμοποιήσει codecs που χρειάζονται μεγάλο εύρος ζώνης. Άλλες διαθέσιμες επιλογές είναι οι: *medium* και *high*.

3.4 Ρυθμίζοντας τα Extensions

Στην καρδιά του Asterisk υπάρχει το Dial Plan. Το αρχείο *extension.conf* είναι το μέσο το οποίο υποδεικνύει στο τηλεφωνικό μας κέντρο , πως θα διαχειριστεί τις κλήσεις. Το Dial Plan περιέχει μια λίστα από οδηγίες που είναι πλήρης παραμετροποιήσιμα , κάτι το οποίο δεν συμβαίνει στα παραδοσιακά τηλεφωνικά κέντρα. Επειδή το Dial Plan είναι πάρα πολύ σημαντικό , θα αναλυθεί πλήρως στο επόμενο κεφάλαιο.

3.5 Ρυθμίζοντας το Voice Mail

Από τις πιο δημοφιλείς λειτουργίες ενός σύγχρονου τηλεφωνικού συστήματος είναι αυτή του φωνητικού ταχυδρομείου ή αλλιώς Voice Mail. Το Asterisk μέσω του προγράμματος *Comedian Mail* παρέχει τέτοιες λειτουργίες με αρκετά πιο εξελιγμένες δυνατότητες από τις συνηθισμένες. Μερικές από τις λειτουργίες του *Comedian Mail* είναι οι παρακάτω: ο ορισμός πολλαπλών προστατευμένων θυρίδων φωνητικού ταχυδρομείου και η σύνδεσή τους με πλήθος συσκευών, ο ορισμός της χωρητικότητα της κάθε θυρίδας και ο μέγιστος επιτρεπτός χρόνος του κάθε μηνύματος, η ηχογράφηση προσαρμοσμένων χαιρετισμών ανάλογα με την κατάσταση της κλήσης, η αποστολή των ηχογραφημένων μηνυμάτων μέσω email και η ειδοποίηση βομβητών.

Οι παραπάνω λειτουργίες, ορίζονται στο αρχείο *voicemail.conf*. Το αρχείο *voicemail.conf* ακολουθεί παρόμοια σύνταξη με αυτή του *extensions.conf*. Οι θυρίδες δημιουργούνται μέσα σε contexts στα οποία πρέπει να έχουν πρόσβαση οι αντίστοιχοι χρήστες. Εκτός από τα προσαρμοσμένα contexts τα οποία δημιουργούνται, στο αρχείο *voicemail.conf* υπάρχουν και τα προκαθορισμένα contexts [general] και [zonemessages]. Παρακάτω βλέπουμε το αρχείο *voicemail.conf*:

```

[general]
format=wav49|gsm|wav
maxmessage=300
minmessage=3
maxgreet=45
maxsilence=5
saycid=yes

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

[default]
111 => 123,Sip User,,,tz=Greece
222 => 456,SIP User,,,tz=Greece
333 => 789,SIP User,,,tz=Greece
[zonemessages]
Greece=Europe/Athens|vm-received' Q 'digits/at' R
[default]
111 => 123,Sip User,,,tz=Greece
222 => 456,SIP User,,,tz=Greece
333 => 789,SIP User,,,tz=Greece

```

I. General Section

Στο context [general] ορίζονται γενικές ρυθμίσεις συμπεριφοράς των θυρίδων του φωνητικού ταχυδρομείου όπως:

- Ο τύπος αρχείου που θα ηχογραφηθεί: `format=wav49|gsm|wav`
 - Οι ρυθμίσεις αποστολής email ειδοποιήσεων:
 - *servermail*: Το όνομα του email εξυπηρετητή
 - *fromstring*: Το όνομα που θα εμφανστεί στο πεδίο from του email
 - *attach*: Επισύναψη του ηχογραφημένου μηνύματος στο email
 - *mailcmd*: Η τοποθεσία που λειτουργεί ο εξυπηρετητής ταχυδρομείου στο σύστημα. (π.χ. `/usr/bin/sendmail -t`)
- Τα χρονικά όρια των μηνυμάτων σε δευτερόλεπτα:

- *maxmessage*: Ο μέγιστος αριθμός δευτερολέπτων που θα επιτρέπεται η ηχογράφηση του φωνητικού μηνύματος
 - *minmessage*: Ο ελάχιστος αριθμός δευτερολέπτων που θα επιτρέπεται η ηχογράφηση του φωνητικού μηνύματος
 - *maxgreet*: Ο μέγιστος αριθμός δευτερολέπτων που θα επιτρέπεται η ηχογράφηση φωνητικού χαιρετισμού
 - *maxsilence*: Πόσο δευτερόλεπτα σιωπής θα επιτρέπονται, προτού θεωρηθεί ότι τελείωσε η ηχογράφηση του μηνύματος.
- Πληροφορίες του καλούντα:
- *saycid*: Πριν την αναπαραγωγή του ηχογραφημένου μηνύματος, ακούγεται ο αριθμός του καλούντα αν η επιλογή *saycid=yes*
 - *cdinternalcontexts*: Πριν την αναπαραγωγή του ηχογραφημένου μηνύματος ακούγεται αν ο καλών προήλθε από κάποιο εσωτερικό context ή όχι.

II. Zone Messages Section

Στο contexts [*zonemessages*] ορίζονται οι ζώνες τοπικών ωρών που δείχνουν στο Asterisk με ποιόν ακριβώς τρόπο θα ειδοποιεί τους χρήστες για την ώρα που έλαβαν τα μηνύματά τους. Μία ζώνη τοπικής ώρας ορίζεται στο context [*zonemessages*] ως: *ΟΝΟΜΑ_ΖΩΝΗΣ=ΖΩΝΗ_ΤΟΠΙΚΗΣ_ΩΡΑΣ|ΕΠΙΛΟΓΕΣ*, Όπου *ΟΝΟΜΑ_ΖΩΝΗΣ* είναι το όνομα που επιθυμούμε να δώσουμε στη ρύθμιση, *ΖΩΝΗ_ΤΟΠΙΚΗΣ_ΩΡΑΣ* είναι η ζώνη τοπική ώρας για το Linux όπως ορίζεται στο φάκελο */usr/share/zoneinfo* και *ΕΠΙΛΟΓΕΣ* ένας συνδυασμός μεταβλητών ημερομηνίας του Unix και αρχείων ήχου.

III. User Mail Section

Τέλος υπάρχουν τα contexts που περιέχουν τις θυρίδες των χρηστών του τηλεφωνικού συστήματος. Μια θυρίδα που περιέχεται σε ένα προσαρμοσμένο context συντάσσεται ως εξής:

- ❖ Όνομα θυρίδας=> συνθηματικό, ιδιοκτήτης-θυρίδας, ηλεκτρονικό-ταχυδρομείο, βομβητής, εναλλακτικό-ηλεκτρονικό-ταχυδρομείο, προσαρμοσμένες-επιλογές

3.6 Ρυθμίζοντας Ειδικές Λειτουργίες

Το αρχείο *features.conf* περιέχει ρυθμίσεις για ένα πλήθος λειτουργιών που μπορούν να υλοποιηθούν στον Asterisk. Αυτές μπορεί να είναι : στάθμευση κλήσεων , μεταφορά κλήσεων, δυναμική ηχογράφηση κλήσεων, απάντηση μίας απομακρυσμένης extension που δέχεται κλήση, και η δυνατότητα δυναμικής εκτέλεσης μίας εφαρμογής με κάποιο συνδυασμό πλήκτρων. Εμείς για να καταλάβουμε την λειτουργία αυτού του αρχείου θα κάνουμε ένα παράδειγμα για μεταφορά κλήσης.

Η δυνατότητα μεταφοράς κλήσεων από ένα νούμερο σε ένα άλλο είναι από τις πρωταρχικές λειτουργίες που πρέπει να εκτελεί ένα παραδοσιακό PBX. Στον Asterisk μας δίνεται η δυνατότητα να υλοποιήσουμε αυτήν τη λειτουργία με δύο τρόπους: 1) Τυφλή μεταφορά κλήσης (*blindxfer*), 2) Εποπτευόμενη μεταφορά κλήσης (*atxfer*). Στην τυφλή μεταφορά πατάμε το συνδυασμό των πλήκτρων που ενεργοποιούν τη λειτουργία, ακούμε από το Asterisk το μήνυμα “*Transfer*” , εισάγουμε τον αριθμό της γραμμής που επιθυμούμε να μεταφέρουμε την κλήση και κλείνουμε το ακουστικό. Στην εποπτευόμενη μεταφορά μπορούμε να κάνουμε το ίδιο με την τυφλή μεταφορά ή αν επιθυμούμε, να περιμένουμε να απαντηθεί η μεταφορά, να ενημερώσουμε το άτομο που πρόκειται να του μεταφέρουμε την κλήση για τις προθέσεις μας και στη συνέχεια να κλείσουμε τη γραμμή για να επιτευχθεί η μεταφορά. Παρακάτω βλέπουμε το αρχείο *features.conf* με τις ρυθμίσεις για την μεταφορά κλήσης :

```
[general]
transferdigittimeout => 3
xfersound = beep
xferfailsound = beeper
atxfernoanswertimeout = 20

[featuremap]
blindxfer => **
atxfer => ##
```

I. General Section

Σε αυτό το section περιέχονται οι ρυθμίσεις συμπεριφοράς των λειτουργιών που μπορούν να εκτελεστούν μέσω αυτού του αρχείου και περιγράφηκαν προηγουμένως. Αναλυτικότερα:

- ❖ *transferdigittimeout* :Δευτερόλεπτα αναμονής ψηφίων για μεταφορά κλήσης
- ❖ *xfersound* :Αναπαραγωγή ήχου ειδοποίησης επιτυχημένης εποπτευόμενης μεταφοράς.
- ❖ *xferfailsound* :Αναπαραγωγή ήχου ειδοποίησης αποτυχημένης μεταφοράς.
- ❖ *atxfernoanswertimeout* :Χρονικό όριο 20 δευτερολέπτων πριν προκληθεί timeout.

II. Featuremap Section

- ❖ *blindxfer* :Τυφλή μεταφορά κλήσης πατώντας δύο φορές το πλήκτρο “*”.
- ❖ *atxfer*:Εποπτευόμενη μεταφορά κλήσης πατώντας δύο φορές το πλήκτρο “#”

3.7 Ρυθμίζοντας την Απομακρυσμένη Διαχείριση

Υπάρχει η δυνατότητα στον Asterisk να μπορούμε να τον διαχωρίζουμε από απόσταση , με ειδικό λογισμικό. Μέσα από αυτό μπορούμε να έχουμε πλήρη εικόνα του τηλεφωνικού μας κέντρου. Ανάλογα με το λογισμικό , μπορούμε να βλέπουμε

οτιδήποτε γίνεται στ τηλεφωνικό μας κέντρο ή ακόμα να μπορούμε να αλλάζουμε ρυθμίσεις του. Για να γίνει αυτό , εκτός από το λογισμικό θα χρειαστεί να ρυθμίσουμε και τον Asterisk , ώστε να μπορεί να δεχθεί το οποιοδήποτε λογισμικό να επέμβει.

Αυτό γίνεται μέσω του αρχείου `manager.conf` και οι τυπικές παράμετροι που περιέχει και θέλουν ρύθμιση , φαίνονται παρακάτω:

```
[general]
displayssystemname = yes
enabled = yes
webenabled = yes
port = 5038
bindaddr = 0.0.0.0

[mark]
secret = 123
;deny=0.0.0.0/0.0.0.0
permit=10.0.3.0/255.255.255.0
```

I. General Section

- *displayssystemname* : Με αυτή την επιλογή λέμε εάν θέλουμε να εμφανίζεται το όνομα του συστήματος μας ή να είναι κρυφό.
- *enabled* : Εδώ δηλώνουμε , ένα θέλουμε να ενεργοποιήσουμε την απομακρυσμένη διαχείριση από άλλο λογισμικό.
- *webenabled* : Εάν το λογισμικό που χρησιμοποιούμε είναι web based τότε βάζουμε σε αυτή την επιλογή την τιμή “ yes “ .
- *port* : Εδώ ορίζουμε σε ποια πόρτα θα μιλήσει το πρόγραμμα με τον Asterisk. Η default πόρτα είναι η 5038.
- *bindaddr* : Η γραμμή `bindaddr=0.0.0.0` λέει στο Asterisk να ακούει για συνδέσεις και να τις επιτρέπει σε όλες τις διευθύνσεις IP που διαθέτει το σύστημά μας.

II. User section

Σε αυτό το Section ορίζουμε τον user που θα μπορεί να εισέλθει στο σύστημα μας και να το διαχειριστεί. Στην προκειμένη περίπτωση ο χρήστης μας είναι ο “ *mark* “

- *secret* : Εδώ ορίζουμε το συνθηματικό με το οποίο θα μπορεί ο user mark να εισέλθει.
- *deny / permit* : Εδώ ορίζουμε ποιες IP συστημάτων θα δέχεται ή θα απορρίπτει ο Asterisk , κατά την διάρκεια της προσπάθειας τους να εισέλθουν σε αυτόν.

4. Το Dialplan του Asterisk^([1],[2],[5],[6])

4.1 Εισαγωγή

Το Dialplan είναι πραγματικά η καρδιά του Asterisk. Καθορίζει πως ο Asterisk θα διαχειριστεί τις εισερχόμενες και εξερχόμενες κλήσεις. Μέσα σε ένα κέλυφος, υπάρχουν οδηγίες και βήματα που πρέπει να ακολουθήσει ο Asterisk για να μπορέσει να ολοκληρώσει όλες τις διεργασίες που θέλουμε να εκτελεστούν. Αντίθετα με τα παραδοσιακά τηλεφωνικά κέντρα, ο Asterisk είναι πλήρως παραμετροποιήσιμος.

4.2 Σύνταξη του Dialplan

Το Dialplan του Asterisk καθορίζεται από το configuration file, *extension.conf*. Το dialplan αποτελείται από 4 κύρια μέρη: τα *contexts*, τις *extensions*, τους *αριθμούς προτεραιότητας* και τις *εφαρμογές*. Παρακάτω θα εξηγήσουμε ποια είναι η χρησιμότητά τους, πως συντάσσονται, και πως μπορούν να συνδυαστούν για να δημιουργήσουν ένα dialplan.

4.2.1 Contexts

Τα dialplans χωρίζονται σε τομείς που ονομάζονται *contexts*. Αυτά είναι ομάδες ονομαστικών γκρουπ από εσωτερικούς αριθμούς τηλεφώνων (*extensions*). Δηλαδή, μέσω των *context*, μπορούμε να χωρίσουμε το πλήθος των *extension* που έχουμε στο

σύστημά μας και να τις χειριστούμε εντελώς ανεξάρτητα. Οι *extensions* που βρίσκονται μέσα σε ένα *context* μπορούν να αλληλεπιδράσουν μόνο με αυτές 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 άλλες *extensions*.

Ένα *context* ξεκινάει με το όνομά του μέσα σε τετραγωνικά άγκιστρα [] και τελειώνει όταν ξεκινήσει το επόμενο. Να σημειώσουμε εδώ ότι το όνομα του μπορεί να περιέχει γράμματα και αριθμούς. Για παράδειγμα ένα *context* εισερχόμενες κλήσεις θα μπορούσε να είναι έτσι :

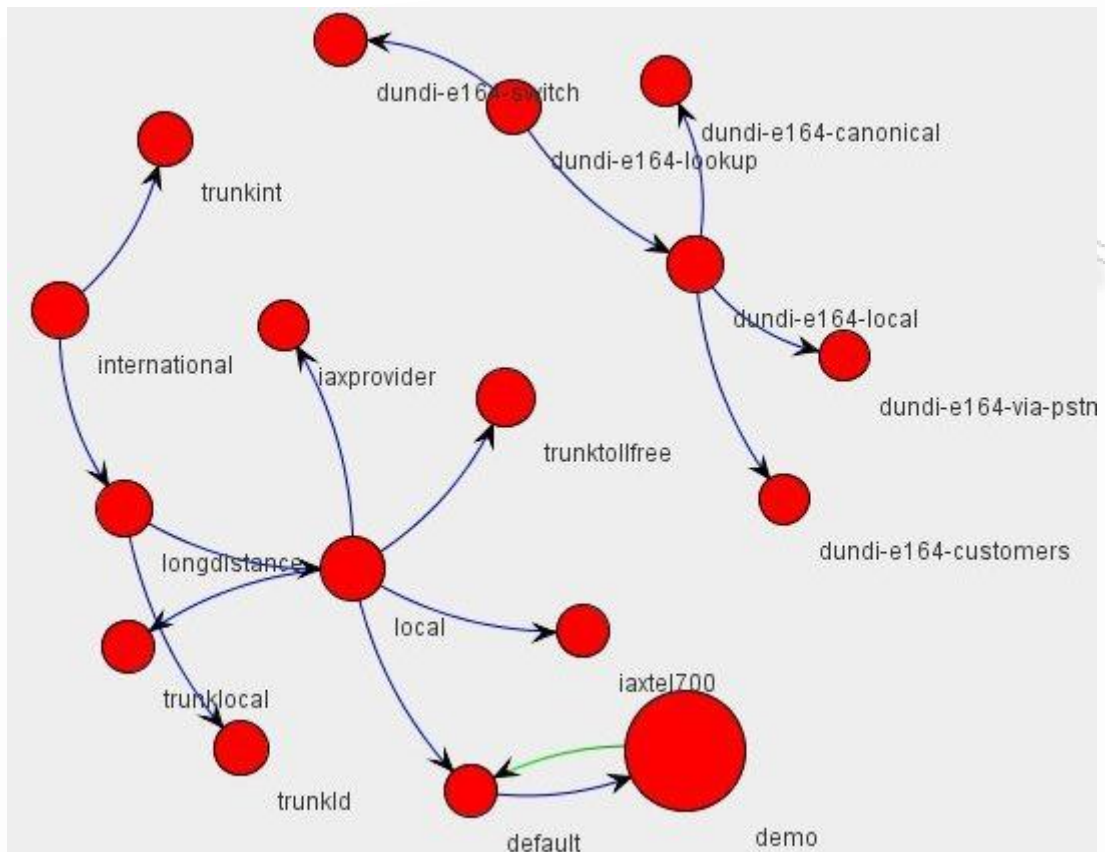
[incoming]

Όλες οι πληροφορίες μετά από αυτό το όνομα είναι μέρος αυτού το *context* έως ότου οριστεί εκ νέου άλλο. Επίσης στην αρχή του Dialplan υπάρχουν 2 ιδιαίτερα *context* που ονομάζονται ως : *[general]* και *[global]*. Με αυτά θα ασχοληθούμε στην συνέχεια.

Στο σημείο αυτό θα αναφέρουμε ένα παράδειγμα για να γίνει κατανοητό το πώς λειτουργούν τα *contexts*. Θα θεωρήσουμε , ότι έχουμε ένα σύστημα Asterisk και το μοιράζονται 2 εταιρίες. Εάν τοποθετήσουμε σε κάθε εταιρία το δικό της εισαγωγικό μενού φωνής , σε διαφορετικό *context* , τότε αμέσως η κάθε εταιρία είναι αυτόνομη. Δηλαδή εάν κάποιος πάρει τηλέφωνο στην *Εταιρία Α* , θα βγει η κλήση εκεί , ενώ εάν πάρει στην *Εταιρία Β* , θα χτυπήσει στην *Β*. Επίσης μέσα στο κάθε *context* της κάθε εταιρίας , θα μπορούσαμε να είχαμε και άλλα *context* τα οποία θα έκαναν άλλες δουλείες. Θα μπορούσε να περιέχει τα ακόλουθα : *[intranet]*, *[agents]*, *[outgoing]*, *[CTO]* και *[CEO]*. Σε αυτά τα *contexts* έχουμε ορίσει τα εξής ότι όλοι οι υπάλληλοι συνδέονται στο *context [intranet]* το οποίο περιλαμβάνει και το *context [operator]*. Στο *context [agents]*, υπάρχουν οι χειριστές που μπορούν να παρέχουν εξωτερικές γραμμές. Τα *contexts [CTO]* και *[CEO]* στα οποία συνδέονται μόνο ο Τεχνικός Διευθυντής και ο Γενικός Διευθυντής της εταιρίας, περιλαμβάνουν το ένα το άλλο καθώς επίσης και το *context [intranet]*. Παρακάτω βλέπουμε όσα είπαμε:

```
[intranet]
include => agents
include => CTO
...
[agents]
...
[outgoing]
...
[CTO]
include => intranet
include => outgoing
include => CEO
include => outgoing
...
[CEO]
include => intranet
include => outgoing
include => CTO
...
```

Με τον παραπάνω τρόπο έχουμε καταφέρει να βάλουμε κάποια ιεραρχία στον τρόπο των κλήσεων. Παρατηρούμε ότι όλοι οι υπάλληλοι μπορούν να καλέσουν ο ένας τον άλλον, να ζητήσουν εξωτερική γραμμή καθώς επίσης και να καλέσουν τον CTO. Μόνο ο CTO μπορεί να καλέσει τον CEO. Ο CEO και ο CTO μπορούν να καλέσουν όλους τους υπαλλήλους του *[intranet]* και κατ' επέκταση και τους χειριστές τους *[agents]*, μπορούν επίσης να καλέσουν απευθείας εξωτερικές γραμμές μέσω του *[outgoing]*. Από τα παραπάνω μπορούμε να συμπεράνουμε για τη μέθοδο *include* ότι, το *context* που κάνει *include* κάποιο άλλο, έχει άμεση πρόσβαση στις *extensions* του άλλου καθώς επίσης και στις *extensions* οποιουδήποτε *context* έχει κάνει *include* αυτό με τη σειρά του, κ.ο.κ.



Εικόνα 3.1 Γραφική αναπαράσταση contexts

[general]

Στο context [general] ορίζονται κάποιες ιδιότητες του extensions.conf και γενικές συμπεριφορές του αρχείου.

- *static*: Αυτή η επιλογή επηρεάζει την εντολή “*save dialplan*”. Η προεπιλεγμένης της τιμή είναι “*no*”
- *writeprotect*: Αν το *static=yes* και *writeprotect=no*, τότε με την εντολή στο CLI του Asterisk, “*save dialplan*”, τότε σώζουμε το αρχείο *extension.conf*.
- *autofallthrough*: Με αυτή την επιλογή μπορούμε να ορίσουμε θέτοντας την με “*yes*” στον Asterisk να κλείνει το κανάλι της κλήσης αν του τελειώσουν οι διαθέσιμες επιλογές που του έχουμε ορίσει μέσω των extensions.
- *clearglobalvars*: Με αυτή την επιλογή μπορούμε να ορίσουμε στον Asterisk με κάθε επανεκκίνηση ή επαναφόρτιση 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`

- *priorityjumping*: Τέλος με την επιλογή αυτή θέτοντας την με “yes”, οι εφαρμογές που υποστηρίζουν άλματα προτεραιότητας ανάλογα με το αποτέλεσμα της λειτουργίας τους (`Dial()`) μπορούν να τα πράξουν.

[globals]

Στο context `[globals]` ορίζονται μεταβλητές οι οποίες ισχύουν 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, μέσα από το `dialplan` με την εφαρμογή `SetGlobalVar(ONOMA_METABΛΗΤΗΣ=ΤΙΜΗ)`

4.2.2 Extensions

Όπως αναφέρθηκε και παραπάνω, μέσα σε κάθε context περιέχεται μία ή περισσότερες `extensions`. Μία `extension`, είναι μία οδηγία που θα ακολουθήσει το Asterisk αν δεχθεί μία κλήση ή αν κάποιος χρήστης πληκτρολογήσει κάποια ψηφία. Οι `extensions` λοιπόν, ορίζουν το πώς η κλήση θα κινηθεί μέσα από μία σειρά αποφάσεων και παραδοχών.

Η σύνταξη μιας extension είναι η λέξη *exten* , ακολουθώντας με ένα ίσον (“=”) και το σύμβολο της ανισότητας (“>”) :

exten=>

Μετά από αυτό ακολουθεί το όνομα του extension. Όταν μιλάμε για τηλεφωνικά συστήματα , τείνουμε να θεωρούμε τα extensions ως αριθμούς που θέλουμε να καλέσουμε και να χτυπήσει το τηλέφωνο τους. Στο Asterisk μπορούμε να θεωρήσουμε , πολλά περισσότερα και αυτό γιατί , μπορούμε εκτός από αριθμούς να βάλουμε και γράμματα και έτσι να σχηματίσουμε λέξεις ή ακόμα και αλφαριθμητικά.

Μία ολοκληρωμένη extension , αποτελείται από 3 στοιχεία :

- Το όνομα ή τον αριθμό της extension
- Την προτεραιότητα. Κάθε extension μπορεί να αποτελείται από πολλά βήματα. Αυτός ο αριθμός του βήματος ονομάζεται προτεραιότητα.
- Την εφαρμογή (ή την εντολή) που πραγματοποιείται κατά την διάρκεια μιας κλήσης.

Αυτά τα 3 στοιχεία διαχωρίζονται μεταξύ τους με ένα κόμα όπως παρακάτω :

exten=>name,priority,application()

Παρακάτω φαίνεται ένα παράδειγμα για το πώς πραγματικά μοιάζει μια extension :

exten=>123,1,Answer()

Το παραπάνω σημαίνει , ότι η extension έχει όνομα 123 , προτεραιότητα 1 και η εφαρμογή που θα εκτελεστεί είναι η *Answer()*.

4.2.3 Προτεραιότητες (Priorities)

Κάθε extension έχει πολλά βήματα , τα οποία καλούνται *προτεραιότητες* (*priorities*). Κάθε προτεραιότητα αριθμείται διαδοχικά , αρχίζοντας από το 1. Κάθε προτεραιότητα , εκτελεί μια εφαρμογή. Στο παράδειγμα που ακολουθεί , στη παρακάτω extension θα απαντηθεί το τηλεφώνημα στην *προτεραιότητα 1* και έπειτα θα κλίσει η γραμμή στην *προτεραιότητα 2*.

exten=>123,1,Answer()

exten=>123,2,Hangup()

Το τι σημαίνει *Answer* και *Hangup* θα το εξηγήσουμε παρακάτω. Αυτό που θα πρέπει να συγκρατήσουμε , είναι ότι η κάθε extension στον Asterisk , ακολουθείται από προτεραιότητα με αριθμητική σειρά.

4.2.4 Εφαρμογές (Applications)

Οι εφαρμογές είναι το σημαντικότερο στοιχείο του dialplan. Κάθε εφαρμογή , εκτελεί μια συγκεκριμένη ενέργεια στο τρέχον κανάλι , όπως να παίζει μουσική , να δεχτεί δεδομένα μέσω των πλήκτρων της τηλεφωνικής συσκευής , ή να τερματίσει την κλήση. Στο προηγούμενο παράδειγμα μας , ήδη αναφέραμε 2 εφαρμογές : την *Answer ()* και την *Hangup ()*.

Εφαρμογές όπως την *Answer ()* και την *Hangup ()* , δεν χρειάζονται επιπρόσθετη πληροφορία για να λειτουργήσουν. Υπάρχουν κα άλλες εφαρμογές που χρειάζονται επιπρόσθετα για να λειτουργήσουν. Αυτές οι επιπρόσθετες πληροφορίες ονομάζονται *arguments* και μπαίνουν επιπρόσθετα στις εφαρμογές για να αλλάξουν τον τρόπο με το οποίο αυτές θα εκτελεστούν. Για να ορίσουμε ένα *argument* σε μια εφαρμογή , αρκεί να το γράψουμε ανάμεσα στις παρενθέσεις “()” της εφαρμογής.

Κάποιες από τις πιο συνήθεις εφαρμογές που χρησιμοποιούνται είναι:

- *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. Τυπώνει μονάχα το αλφαριθμητικό που έλαβε ως όρισμα, στην κονσόλα του Asterisk. Χρησιμεύει όταν ψάχνουμε για σφάλματα στο dialplan.
- *System*: Η εφαρμογή αυτή μας επιτρέπει να τρέξουμε οποιαδήποτε εντολή του λειτουργικού συστήματος (Linux) που έχουμε εγκατεστημένο τον Asterisk. Η εφαρμογή αυτή είναι πάρα πολύ σημαντική γιατί μέσω αυτής μπορούμε μέσω του τηλεφώνου μας να εκτελέσουμε οποιαδήποτε εντολή/εντολές σε κάποιον υπολογιστή κάνοντας τον Asterisk ακόμα πιο ευέλικτο στην δημιουργία εφαρμογών μέσω VoIP .
- *MP3PLAYER*: Με αυτή την εφαρμογή μπορούμε να αναπαράγουμε ένα οποιαδήποτε αρχείο MP3. Ο Asterisk χρησιμοποιεί το πρόγραμμα του λειτουργικού συστήματος *mpeg123*. Αναπαράγοντας το αρχείο , ο καλών αρχίζει να το ακούει και μπορεί ανά πάσα ώρα και στιγμή να το σταματήσει , πατώντας οποιαδήποτε πλήκτρο από τη συσκευή του.
- *DateTime*: Με την αυτή την εφαρμογή, ο Asterisk μας λέει την τρέχουσα ημερομηνία και την ώρα. Η εφαρμογή αυτή χρησιμοποιεί τις βιβλιοθήκες ήχων της Digium για να μπορέσει να αναπαράγει την ημερομηνία και την ώρα.
- *Dial (Τεχνολογία/id, επιλογές, χρόνος)*: Η εφαρμογή Dial() λέει στο Asterisk να καλέσει μέσω της «τεχνολογίας» τον αριθμό «id» και όταν η γραμμή

απαντηθεί να ενώσει τα δύο άκρα της επικοινωνίας ανεξαρτήτου τεχνολογίας που χρησιμοποιούν,

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
- T: επιτρέπει στον καλούντα 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.
- W: Επιτρέπει την ηχογράφηση της κλήσης από τον καλούντα με το πάτημα της αλληλουχίας των πλήκτρων που ενεργοποιούν το automon όπως αυτά έχουν οριστεί στο αρχείο features.conf.

4.3 Το Πρώτο Μας Dialplan

Μετά από όλα όσα είπαμε θα προχωρήσουμε στον σχεδιασμό του πρώτου μας dialplan. Για να γίνει αυτό το πρώτο πράγμα στο οποίο πρέπει να συμφωνήσουμε, είναι τι όνομα θα έχει το extension μας. Θα θεωρήσουμε ένα απλό τριψήφιο αριθμό, για παράδειγμα το “100”. Αφού αποφασίσαμε ποιο το όνομα του extension, θα πρέπει να ορίσουμε τώρα τι ακριβώς θα γίνεται, όταν καλούμε τον εν λόγω αριθμό, από την τηλεφωνική μας συσκευή. Στο παράδειγμα μας θα θεωρήσουμε ότι όταν καλούμε τον αριθμό 100 απλά να χτυπάει για λίγη ώρα και εάν δεν το σηκώσει ο χρήστης που είναι πίσω από το 100, μετά από λίγα δευτερόλεπτα, να κλείνει το

κανάλι επικοινωνίας. Τέλος για να το πετύχουμε αυτό θα πρέπει να ορίσουμε και τις κατάλληλες προτεραιότητες.

Έχοντας όλα αυτά υπ όψη , συντάσσουμε το παρακάτω dialplan.

[example]

```
exten=>100,1,Noop()
```

```
exten=>100,2,Dial(Sip/sip-user,20)
```

```
exten=>100,3,Hangup()
```

Το context το ονομάσαμε “example” . Έπειτα ορίζουμε τις 3 προτεραιότητες. Στην 1^η προτεραιότητα βάζουμε την εφαρμογή Noop() , που δεν κάνει ουσιαστικά τίποτα , αλλά την βάζουμε για να μπορούμε να κάνουμε debug σε περίπτωση προβλήματος. Έπειτα με την 2^η προτεραιότητα ορίζουμε καλώντας το νούμερο 100 , να γίνει κλήση με το πρωτόκολλο SIP στον user sip-user. Η προσπάθεια κλήσης θα διαρκέσει 20 δευτερόλεπτα. Εάν μέσα σε αυτά ο sip-user το σηκώσει τότε θα συνεχιστεί η κλήση μέχρι ως ότου τερματιστεί από τους χρήστες , αλλιώς μετά το πέρας των 20 δευτερολέπτων το σύστημα θα μας πάει στην 3^η προτεραιότητα και θα τερματιστεί η κλήση. Αυτό το πολύ απλό dialplan μπορούμε να το χρησιμοποιήσουμε εάν μόνο θέλουμε να βγάζουμε κλήσεις προς διάφορους χρήστες. Στην πραγματικότητα θέλουμε πολύ περισσότερα , γι αυτό το dialplan μας είναι πολύ ποιο πολύπλοκο.

Θα κάνουμε άλλο ένα παράδειγμα χρησιμοποιώντας την εφαρμογή playback. Έστω ότι έχουμε ηχογραφήσει ένα αρχείο wav ή gsm , το οποίο μας ενημερώνει για κάτι. Εμείς θέλουμε όταν κάποιος χρήστης καλεί έναν συγκεκριμένο αριθμό , να ακούγεται αυτό το αρχείο ήχου. Για να γίνει αυτό θα γράψουμε το παρακάτω dialplan:

[example2]

```
exten=>200,1,Answer()
```

```
exten=>200,2,Playback(/test)
```

```
exten=>200,3,Hangup()
```

Περνώντας λοιπόν το 200 , θα απαντηθεί αυτόματα η κλήση και θα ανοίξει το σύστημα ένα κανάλι επικοινωνίας. Στην δεύτερη προτεραιότητα ενεργοποιείται η εφαρμογή Playback και της ορίζουμε μέσα στην παρένθεση την ακριβή θέση και το όνομα του αρχείου ήχου που θα αναπαράγει. Αφού λοιπόν τελειώσει το αρχείο ήχου με την τελευταία προτεραιότητα τερματίζουμε την κλήση.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΡΑΙΑ

5. Προηγμένες Τεχνικές Ανάπτυξης Υπηρεσιών VoIP^([4],[5],[8])

Στον Asterisk, λειτουργίες ή προγράμματα μπορούν να υλοποιηθούν είτε εξωτερικά, μέσω ενός AGI script, είτε εσωτερικά, μέσω συναρτήσεων, μεταβλητών και εφαρμογών στο dialplan. Σε αυτό το κεφάλαιο θα ασχοληθούμε με την προσθήκη εσωτερικής και εξωτερικής λογικής στο dialplan χρησιμοποιώντας εργαλεία που αναφέραμε στο προηγούμενο κεφάλαιο, καθώς επίσης και με την AGI λογική.

5.1 Εργαλεία Για Την Ανάπτυξη Εσωτερικής Λογικής

5.1.1 Μεταβλητές

Προηγουμένως, αναφερθήκαμε στις global μεταβλητές οι οποίες δηλώνονται στο context [globals] του Asterisk dialplan, ισχύουν καθολικά στο Asterisk και καλούνται με το συμβολισμό $\{ONOMA_METABΛΗΤΗΣ\}$.

Εκτός από τις global μεταβλητές το Asterisk έχει τη δυνατότητα αποθήκευσης και ανάκτησης τιμών και από έναν άλλον τύπο μεταβλητών, τις channel μεταβλητές. Μια channel μεταβλητή, είναι μια μεταβλητή η οποία έχει ισχύ για τη διάρκεια της κλήσης στην οποία δημιουργήθηκε, μετά το τέλος της οποίας παύει να υπάρχει.

Μία channel μεταβλητή ορίζεται μέσα στο dialplan, μέσω της εφαρμογής Set(),

exten => extension, προτεραιότητα, Set(ONOMA_ΜΕΤΑΒΛΗΤΗΣ=Τιμή)

Και η τιμή της ανακτάται στο dialplan με τη χρήση του συμβολισμού $\{ONOMA_METABΛΗΤΗΣ\}$, όπως είδαμε προηγουμένως για τις global μεταβλητές.

Υπάρχουν επίσης πολλές προκαθορισμένες channel μεταβλητές οι οποίες μπορούν να χρησιμοποιηθούν στο Asterisk. Κάποιες από αυτές είναι:

- $\{ANSWEREDTIME\}$: Κρατάει την τιμή των δευτερολέπτων που έχουν περάσει απ' τη στιγμή που ξεκίνησε η κλήση.
- $\{EPOCH\}$: Η τρέχουσα ώρα σε μορφή UNIX (δευτερόλεπτα που έχουν περάσει από την 1^η Ιανουαρίου 1970)
- $\{CHANNEL\}$: Το όνομα του τρέχοντος καναλιού κλήσης
- $\{CONTEXT\}$: Το όνομα του τρέχοντος context
- $\{PRIORITY\}$: Το νούμερο της τρέχουσας προτεραιότητας
- $\{EXTEN\}$: Η extension που πληκτρολογήθηκε από το χρήστη
- $\{INVALID_EXTEN\}$: Χρησιμοποιείται σε συνδυασμό με την ειδική extension i και περιέχει την εσφαλμένη extension που πληκτρολογήθηκε από το χρήστη.
- $\{ENV(VARIABLENAME)\}$: Περιέχει την τιμή της μεταβλητής συστήματος (λειτουργικού) VARIABLENAME.
- $\{DIALSTATUS\}$: Ανάλογα με την κατάσταση της κλήσης, η μεταβλητή παίρνει τις τιμές ANSWER, NOANSWER, BUSY, CONGESTED, κ.λ.π.

Παρόλο που στην τρέχουσα έκδοση του Asterisk υπάρχουν πολλές προκαθορισμένες channel μεταβλητές οι οποίες κρατάνε σημαντικά στοιχεία όπως: το νούμερο του καλούντα ($\{CALLERIDNUMBER\}$), το όνομα του καλούντα ($\{CALLERIDNAME\}$), την τρέχουσα ώρα, ημερομηνία ($\{TIMESTAMP\}$, $\{DATETIME\}$) κ.α. Η εξέλιξη του Asterisk δείχνει ότι οι δημιουργοί του θα σταματήσουν να τις αναπτύσσουν και τελικά θα αποσύρουν τη χρήση τους. Η νέα τάση που υπάρχει στο Asterisk είναι η χρήση συναρτήσεων τις οποίες θα εξηγήσουμε παρακάτω.

5.1.2 Η εφαρμογή Goto() και Ετικέτες

Ήδη είδαμε ότι κάποιιοι κανόνες ακολουθούνται κατά τη διάρκεια μίας κλήσης, και κάποιες αποφάσεις παίρνονται ανάλογα με συγκεκριμένα κριτήρια. Για να μπορέσει το σύστημά να αλληλεπιδράσει με τους χρήστες θα πρέπει να είναι προγραμματισμένο με τέτοιο τρόπο ώστε να μπορεί να παίρνει κάποιες λογικές αποφάσεις ανάλογα με τις εισόδους του χρήστη. Συντάσσουμε το παρακάτω παράδειγμα για να καταλάβουμε τι ακριβώς εννοούμε. Ονομάζουμε το *context* , *mini-pbx* ως:

```
[mini-pbx]
exten => ${LINE},1,Answer()
exten => ${LINE},2,Background(custom/my-menu)
exten => ${LINE},3,WaitExten(5)

exten => 1,1,Dial(${SIP-USER},20,r)
exten => 1,2,Playback(vm-nobodyavail)
exten => 1,3,Goto(mini-pbx,${LINE},2)

exten => 2,1,Dial(${IAX-USER},20,r)
exten => 2,2,Playback(vm-nobodyavail)
exten => 2,3,Goto(mini-pbx,${LINE},2)

exten => i,1,Playback(pbx-invalid)
exten => i,2,Goto(mini-pbx,${LINE},2)
exten => t,1,Playback(vm-goodbye)
exten => t,2,Hangup()
```

Παραπάνω χρησιμοποιείται ένας προσαρμοσμένος χαιρετισμός που δημιουργήθηκε και τοποθετήθηκε στο φάκελο */var/lib/asterisk/sounds/custom*. Με την εφαρμογή *Background()*, το Asterisk περιμένει να δεχτεί είσοδο από το χρήστη και αναπαραγάγει το αρχείο ήχου *my-menu*, το οποίο ζητάει από τον καλούντα να πληκτρολογήσει 1 αν επιθυμεί να συνδεθεί με τον *sip-user* ή 2 αν επιθυμεί να

συνδεθεί με τον `iax-user`. Στη συνέχεια περιμένει με την `WaitExten()` άλλα 5 δευτερόλεπτα για είσοδο.

Ανάλογα με την είσοδο (1 ή 2), μέσω της `Dial()` καλείται ο αντίστοιχος χρήστης. Αν ο χρήστης απαντήσει στην κλήση τότε δημιουργείται το κανάλι επικοινωνίας και τα δύο άκρα αρχίζουν και ανταλλάζουν πακέτα φωνής. Αν περάσουν 20 δευτερόλεπτα και ο χρήστης δεν απαντήσει την κλήση, τότε μέσω της εφαρμογής `Goto()` η κλήση πάει στον context `[mini-pbx-incoming]`, στην extension `#{LINE}`, στη δεύτερη προτεραιότητα, δηλαδή στην εφαρμογή `Background()` για να επαναληφθεί η παραπάνω διαδικασία.

Η εφαρμογή `Goto()` συντάσσεται ως: `Goto(προορισμός)`

Ο προορισμός μπορεί να είναι:

- Μία προτεραιότητα μέσα στην ίδια extension (π.χ. 2)
- Μία extension και μία προτεραιότητα μέσα στο ίδιο context (π.χ. 111,2)
- Ένα context, μια extension και μία προτεραιότητα (π.χ. internal,111,2)
- Μία ετικέτα (π.χ. (Menu))

Τέλος, αν ο χρήστης εισάγει κάτι διαφορετικό από 1 ή 2 τότε μεταφέρεται στην ειδική extension `i` όπου ακούει το μήνυμα `invalid pbx` και μεταφέρεται πάλι στην `Background()`, ενώ αν δεν εισάγει τίποτα κατά τη διάρκεια της `Background()` και της `WaitExten()`, τότε θα μεταφερθεί στην ειδική extension `t` όπου θα ακούσει το μήνυμα `goodbye`, και θα κλείσει η γραμμή.

Αν ακολουθήσουμε τη συμβολική n αρίθμηση των προτεραιοτήτων, τότε τα πράγματα περιπλέκονται κάπως για την `Goto()`. Εφόσον δεν γνωρίζουμε το ακριβές νούμερο της κάθε προτεραιότητας, θα πρέπει με κάποιο τρόπο να μπορέσουμε να σηματοδοτήσουμε τα σημεία στα οποία επιθυμούμε να μεταφερθούμε. Η σηματοδότηση των σημείων στο `dialplan` επιτυγχάνεται με τη χρήση των ετικετών

(labels). Άρα ξαναγράφουμε το προηγούμενο παράδειγμα εκφρασμένο με την ειδική *n* προτεραιότητα και τη χρήση ετικετών:

```
[mini-pbx]
exten => ${LINE},1,Answer
exten => ${LINE},n(Menu),Background(custom/my-menu)
exten => ${LINE},n,WaitExten(5)

exten => 1,1,Dial(${SIP-USER},20,r)
exten => 1,n,Playback(vm-nobodyavail)
exten => 1,n,Goto(mini-pbx-incoming,${LINE},Menu)

exten => 2,1,Dial(${IAX-USER},20,r)
exten => 2,n,Playback(vm-nobodyavail)
exten => 2,n,Goto(mini-pbx-incoming,${LINE},Menu)

exten => i,1,Playback(pbx-invalid)
exten => i,n,Goto(mini-pbx-incoming,${LINE},Menu)
exten => t,1,Playback(vm-goodbye)
exten => t,n,Hangup
```

Το αποτέλεσμα είναι ακριβώς το ίδιο και για τις 2 μεθόδους. Ο τρόπος με τον οποίο σηματοδοτήθηκε το σημείο που επιθυμούμε να μεταφερθούμε ήταν με την προσθήκη της ετικέτας (*Menu*). Αντί να χρησιμοποιούμε συγκεκριμένους αριθμούς λοιπόν, μπορούμε να χρησιμοποιήσουμε λέξεις οι οποίες βοηθάνε πολύ στην αναγνωσιμότητα του dialplan και τη διαδικασία του debug . Η μέθοδος αυτή, είναι επίσης ανεκτική σε αλλαγές χωρίς να χρειαστεί να αλλάζουμε κάθε φορά αριθμήσεις στις προτεραιότητες.

5.1.3 Η εφαρμογή Gotoif()

Ας υποθέσουμε ότι στο παραπάνω παράδειγμα δε θέλουμε ο χρήστης να μπορεί να επιστρέψει πάνω από τέσσερις φορές στο ηχητικό menu. Αυτό θα μπορούσε να

υλοποιηθεί με τη χρήση της εφαρμογής *GotoIf()*. Με την *GotoIf()* έχουμε τη δυνατότητα να εξετάσουμε την ορθότητα μίας έκφρασης και ανάλογα με το αποτέλεσμα (αληθές ή ψευδές) να κατευθυνθούμε στον ανάλογο προορισμό. Μια έκφραση αποτελείται από δύο συγκρίσιμα μεγέθη (αριθμούς, αλφαριθμητικά, εκφράσεις) και έναν τελεστή σύγκρισης(=, >, <, >=, <=, !=, |, &).

Η *GotoIf()* συντάσσεται ως: *GotoIf(έκφραση:προορισμός1,προορισμός2)*

Για να μπορέσουμε λοιπόν να ορίσουμε ένα μέγιστο αριθμό προσπαθειών που θα μπορεί ο χρήστης να επαναλάβει τη διαδικασία του *Menu*, θα ορίσουμε μία μεταβλητή *RETRIES=0*, την οποία θα αυξάνουμε κατά 1, κάθε φορά που θα επιχειρείται σύνδεση με τις extensions *1,2,i*. Όσο η μεταβλητή *RETRIES* είναι μικρότερη ή ίση του 4, τότε ο καλών θα επιστρέφει στο κεντρικό menu, ενώ σε αντίθετη περίπτωση θα κατευθυνθεί στην ειδική extension *i* όπου θα τερματιστεί η κλήση, όπως φαίνεται παρακάτω:

```
[mini-pbx]
exten => ${LINE},1,Answer
exten => ${LINE},n,Set(RETRIES=0)
exten => ${LINE},n(Menu),Background(custom/my-menu)
exten => ${LINE},n,WaitExten(5)

exten => 1,1,Dial(${SIP-USER},20,r)
exten => 1,n,Set(RETRIES=${RETRIES} + 1)
exten => 1,n,Playback(vm-nobodyavail)
exten => 1,n,GotoIf(${RETRIES} <= 4)?${LINE},Menu:t,1)

exten => 2,1,Dial(${IAX-USER},20,r)
exten => 2,n,Set(RETRIES=${RETRIES} + 1)
exten => 2,n,Playback(vm-nobodyavail)
exten => 2,n,GotoIf(${RETRIES} <= 4)?${LINE},Menu:t,1)

exten => i,1,Playback(pbx-invalid)
exten => i,n,Set(RETRIES=${RETRIES} + 1)
```

```
exten => i,n,GotoIf($[${RETRIES} <= 4]?${LINE},Menu:t,1)
exten => t,1,Playback(vm-goodbye)
exten => t,n,Hangup
```

5.1.4 Συναρτήσεις (Functions)

Εκτός από τις εφαρμογές, οι δημιουργοί του Asterisk έχουν προσθέσει και τις συναρτήσεις. Η προσθήκη των συναρτήσεων και ο τρόπος χρήσης τους, δίνουν σαφείς ενδείξεις πως η σύνταξη του dialplan κατευθύνεται προς μία προγραμματιστική λογική.

Οι συναρτήσεις τροφοδοτούν το Asterisk με δυνατότητες εξελιγμένων πράξεων και υπολογισμών όπως είναι για παράδειγμα η εξαγωγή MD5 checksums, η χρήση ανεπτυγμένων μαθηματικών εκφράσεων όπως η ύψωση δύναμης, ο υπολογισμός μήκους αλφαριθμητικών, η επεξεργασία των στοιχείων αναγνώρισης κλήσεων, χρονικοί υπολογισμοί, κ.α. Όλα τα παραπάνω είναι δυνατόν να συνδυαστούν με σκοπό να παρέχουν ακόμα πιο εξελιγμένες μεθόδους επεξεργασίας.

Σε αντίθεση με τις εφαρμογές, οι συναρτήσεις δεν δηλώνονται απευθείας στο dialplan. Δηλώνονται μέσα σε μία εφαρμογή (π.χ. Set()) και επιστρέφουν κάποια τιμή (π.χ. Read(), NoOp()).

Μια συνάρτηση στο dialplan δηλώνεται ως:

ΟΝΟΜΑ_ΣΥΝΑΡΤΗΣΗΣ(όρισμα)

π.χ. exten => 123, 1, Set(CALLERID(name)="George Duke")

Και καλείται με παρόμοιο τρόπο με αυτόν των μεταβλητών, ως:

\${ΟΝΟΜΑ_ΣΥΝΑΡΤΗΣΗΣ(όρισμα)}

π.χ. exten => 123, 1, NoOp(\${CALLERID(all)})

Οι συναρτήσεις παρέχουν επίσης δυνατότητες ενθυλάκωσης:

```
{ONOMA_ΣΥΝΑΡΤΗΣΗΣ(${ONOMA_ΣΥΝΑΡΤΗΣΗΣ(όρισμα)})}
```

π.χ. exten => 123, 1, NoOp(\${LEN(\${CALLERID(all)})})

Στα παραπάνω παραδείγματα χρησιμοποιήσαμε τις συναρτήσεις LEN() και CALLERID(). Στο πρώτο παράδειγμα, δηλώθηκε στο Asterisk ότι το όνομα George Duke θα χρησιμοποιηθεί στην αναγνώριση κλήσης. Στο δεύτερο παράδειγμα κατά τη διάρκεια της κλήσης, θα τυπωθεί στην κονσόλα του Asterisk το όνομα και το νούμερο του καλούντα, με το νούμερο κλεισμένο μέσα στα σύμβολα < > (π.χ. George Duke <123456789>). Τέλος στο τρίτο παράδειγμα, η τιμή της συνάρτησης CALLERID(all) ενθυλακώνετε στη συνάρτηση LEN() και τυπώνεται στην κονσόλα του Asterisk το μήκος (ακέραιος αριθμός) του αλφαριθμητικού CALLERID(all).

5.2 Δημιουργία Εφαρμογών Με εσωτερική Λογική

5.2.1 Φωνητικό ταχυδρομείο

Όλες οι λειτουργίες φωνητικού ταχυδρομείου στο Asterisk περιέχονται σε δύο εφαρμογές: στην Voicemail(), η οποία χειρίζεται τη διαδικασία καθοδήγησης των καλούντων και καταγραφής των μηνυμάτων τους, και την VoicemailMain() η οποία χειρίζεται τη διαδικασία σύνδεσης των χρηστών στο σύστημα προκειμένου να ακούσουν τα μηνύματα τους και να διαχειριστούν το λογαριασμό τους. Πληροφορίες για τις δύο αυτές εφαρμογές μπορούν να βρεθούν στο κεφάλαιο Το dialplan του Asterisk, στην παράγραφο εφαρμογές.

Παρακάτω θα φτιάξουμε ένα παράδειγμα και θα συνδέσουμε 3 extensions (την 111 , 222 , 333) με τη θυρίδα των φωνητικών ταχυδρομείων τους προκειμένου να μπορούν να αφήνουν μηνύματα ο ένας στον άλλον και να έχουν πρόσβαση στη θυρίδα τους. Τέλος θα χρησιμοποιηθεί η ειδική channel μεταβλητή `{DIALSTATUS}` σε συνδυασμό με την εφαρμογή `Goto()` προκειμένου να μπορεί να γίνει διαχείριση της κλήσης χωρίς να εξαρτάται το σύστημα από τα άλματα προτεραιοτήτων τα οποία σε επόμενες εκδόσεις του Asterisk θα σταματήσουν να υποστηρίζονται. Σύμφωνα με τα παραπάνω, το αρχείο `extensions.conf` γίνεται:

```
[general]
autofallthrough=yes
clearglobalvars=yes
;priorityjumping=yes

[mini-pbx-internal]
exten => 111,1,Dial({SIP-USER},20,r)
exten => 111,n,Goto(111-{DIALSTATUS},1)
exten => 111-NOANSWER,1,Voicemail(u111)
exten => 111-BUSY,1,Voicemail(b111)
exten => 111-ANSWER,1,Hangup()
exten => _111-,1,Goto(111-NOANSWER,1)
exten => 222,1,Noop()
exten => 222,2,Dial(SIP/222,20,tr)
exten => 222,n,Goto(111-{DIALSTATUS},1)
exten => 222-NOANSWER,1,Voicemail(u111)
exten => 222-BUSY,1,Voicemail(b111)
exten => 222-ANSWER,1,Hangup()
exten => _222-,1,Goto(111-NOANSWER,1)

exten => 333,1,Noop()
exten => 333,2,Dial(SIP/333,20,tr)
exten => 333,n,Goto(111-{DIALSTATUS},1)
exten => 333-NOANSWER,1,Voicemail(u111)
exten => 333-BUSY,1,Voicemail(b111)
exten => 333-ANSWER,1,Hangup()
```

```
exten => _333-.,1,Goto(111-NOANSWER,1)
```

Πρόσβαση στις θυρίδες φωνητικού ταχυδρομείου

```
exten => 500,1,GotoIf($[${CALLERID(number)} = 111]?yes:no)
```

```
exten => 500,n(yes),VoicemailMain(${CALLERID(number)})
```

```
exten => 500,n(no),VoicemailMain()
```

Εξετάζοντας ενδεικτικά τη ροή της κλήσης για το χρήστη sip-user βλέπουμε ότι μόλις κληθεί η extension 111 επιχειρείται μέσω της εφαρμογής *Dial()* η δημιουργία καναλιού επικοινωνίας μεταξύ του sip-user και του καλούντα. Ανάλογα με το αποτέλεσμα της *Dial()* η channel μεταβλητή *DIALSTATUS* θα πάρει μία από τις τιμές *ANSWER*, *NOANSWER*, *BUSY*, *CONSESTION*, *CANCEL*, *CHANUNAVAIL*, *DONTCALL*, *TORTURE*, *INVALIDARGS*. Το αποτέλεσμα της *DIALSTATUS* σε συνδυασμό με την εφαρμογή *Goto()* θα στείλουν την κλήση είτε προς το φωνητικό ταχυδρομείο του 111 είτε προς την εφαρμογή *Hangup()* προκειμένου τερματιστεί η κλήση. Αξίζει επίσης να σημειωθεί πως παρόλο που συγκεκριμένα έχουν προβλεφθεί οι περιπτώσεις *BUSY* και *NOANSWER* με τους ανάλογους χαιρετισμούς (b,u), μέσω της pattern *_111-* προβλέπονται και όλες οι άλλες περιπτώσεις της *DIALSTATUS* και δρομολογούνται με πανομοιότυπο τρόπο όπως στην περίπτωση της *NOANSWER*. Αν υποθέσουμε ότι ο χρήστης 222 δεν είναι συνδεδεμένος στο σύστημα (*CHANUNAVAIL*), τότε η *GotoIf()* θα στείλει τον 111 στην περίπτωση *NOANSWER* και στη συνέχεια στο φωνητικό ταχυδρομείο του 222, όπου το Asterisk θα τον ενημερώσει ότι ο χρήστης δεν είναι διαθέσιμος και θα τον προτρέψει να αφήσει κάποιο φωνητικό μήνυμα. Το μήνυμα θα εγγραφεί στο φάκελο */var/spool/asterisk/voicemail/default/222/tmp* σε τρεις εκδόσεις (wav,wav49,gsm) και θα μεταφερθεί όταν ολοκληρωθεί στο φάκελο *../INBOX*.

Τέλος δημιουργήθηκε η extension 500 στην οποία οι χρήστες sip-user και iax-user έχουν πρόσβαση στη θυρίδα του φωνητικού τους ταχυδρομείου. Για το χρήστη sip-user έχει προβλεφθεί (μέσω της συνάρτησης *CALLERID(number)*) και την εφαρμογής *GotoIf()* αν καλεί απ' το δικό του λογαριασμό την extension 500 τότε να έχει άμεση πρόσβαση στη θυρίδα του, ενώ σε κάθε άλλη περίπτωση (όπως ισχύει για το χρήστη 222 και 333) θα ζητηθεί από το χρήστη να εισάγει σε ποια συγκεκριμένη θυρίδα επιθυμεί να συνδεθεί. Σε περίπτωση που επιθυμούμε να παρέχουμε σε

επιπλέον χρήστες αυτήν τη δυνατότητα αρκεί να αλλάξουμε την έκφραση που υπάρχει στην *GotoIf()* χρησιμοποιώντας τον τελεστή | (ή) όπως φαίνεται παρακάτω :

```
exten => 500,1,GotoIf($[${CALLERID(number)} = 111] | $[${CALLERID(number)} = 222])?yes:no)
```

```
exten => 500,n(yes),VoicemailMain(${CALLERID(number)})
```

```
exten => 500,n(no),VoicemailMain()
```

5.2.2 Πρόγνωση Καιρού

Έχοντας υπόψη μας ότι η πλατφόρμα του Asterisk , μπορεί να κάνει τα πάντα αρκεί να τη προγραμματίσουμε για να το κάνει , σκεφτήκαμε να δημιουργήσουμε ένα extension το οποίο όταν το καλούμε αυτό θα κατεβάζει από το Internet την πρόγνωση του καιρού και θα μας την αναφωνούσε. Επειδή ακριβώς πρόκειται για τηλεφώνημα , η πρόγνωση του καιρού θα πρέπει να αναπαράγεται. Δυστυχώς στην Ελλάδα δεν υπάρχει κάποιο site πρόγνωσης καιρού , που να έχει την πρόγνωση σε μορφή ήχου για να μπορείς να την κατεβάσεις. Για το λόγο αυτό καταφύγαμε σε Αμερικάνικη υπηρεσία πρόγνωσης καιρού την NOAA (National Oceanic and Atmospheric Administration). Από εκεί θα κατεβάζουμε το αρχείο ήχου το οποίο θα αναπαράγεται από τον Asterisk , όταν θα καλούμε ένα συγκεκριμένο extension.

Για να φανεί ποιο εντυπωσιακό , έως ότου να κατέβει το αρχείο του καιρού , δημιουργήσαμε ένα αρχείο που μας ενημερώνει ότι το αρχείο κατεβαίνει. Χρησιμοποιήσαμε την υπηρεσία της AT&T , *Text-To-Speech* (TTS). Εκεί γράψαμε ένα μήνυμα , έπειτα επιλέξαμε με ποια φωνή θέλουμε να το ακούσουμε και τέλος αποθηκεύσαμε τον ήχο στο υπολογιστή μας.



Εικόνα 5.4 Demo εφαρμογή TTS της AT&T

Δεν μένει τώρα παρά να γράψουμε το extension. Φτιάχνουμε το extension “865” και γράφουμε τις παρακάτω γραμμές :

```

exten => 865,1,Answer
exten => 865,2,Playback(latest-forecast)
exten => 865,3,System(/usr/bin/curl -s
http://www.wrh.noaa.gov/sgx/data/audio/LAXSAFC42.mp3 > /tmp/LAXSAFC42.mp3)
exten => 865,4,Wait(2)
exten => 865,5,MP3Player(/tmp/LAXSAFC42.mp3)
exten => 865,6,System(rm /tmp/LAXSAFC42.mp3 -f)
exten => 865,7,Hangup

```

Με το που καλούμε το “865” , ανοίγει το κανάλι επικοινωνίας με την προτεραιότητα “1” και την εφαρμογή “Answer”. Στην προτεραιότητα “2” ξεκινάει να αναπαράγεται το μήνυμα Text-To-Speech που δημιουργήσαμε. Έπειτα στην επόμενη προτεραιότητα , κάνουμε χρήση της εφαρμογής System , όπου μας επιτρέπει να γράψουμε εντολές του λειτουργικού μας συστήματος και ο Asterisk με τη σειρά να τις εκτελέσει. Η εντολή , λέει στο λειτουργικό μας σύστημα να συνδεθεί σε

συγκεκριμένη διεύθυνση και το αρχείο που προκύπτει από αυτή να πάει και να το αποθηκεύσει τοπικά στον υπολογιστή μας σε συγκεκριμένο path. Αφού γίνει αυτό έρχεται η προτεραιότητα “5” και αρχίζει να αναπαράγει τον ήχο με την πρόγνωση του καιρού που μόλις κατεβάσαμε. Μόλις τελειώσει η αναπαραγωγή θα πρέπει το αρχείο που αναπαράχθηκε να διαγραφεί από το σύστημα μας για τον λόγο ότι το καινούργιο που θα κατέβει θα έχει το ίδιο όνομα και θα παρουσιάσει σφάλμα ο ASTERISK. Αυτό γίνεται πάλι με την εφαρμογή System , γράφοντας την αντίστοιχη εντολή που θα πρέπει να εκτελεστεί από το λειτουργικό μας σύστημα. Και βέβαια με την τελευταία προτεραιότητα κλείνει το κανάλι επικοινωνίας.

5.2.3 Date Time

Μια ακόμη διάσημη υπηρεσία των δημοσίων τηλεφωνικών δικτύων , είναι η αναγγελία της ώρας. Θα δημιουργήσουμε μία extension ή οποία θα διαβάζει το real time clock του συστήματος και θα μας αναπαράγει την ημερομηνία και την ώρα. Για να το κάνουμε ποιο πολύπλοκο , θα προσθέσουμε μία συνθήκη ή οποία θα λέει πως , εάν δεν κλείσει το κανάλι επικοινωνίας , ο χρήστης που κάλεσε το extension , τότε το σύστημα να συνεχίσει να διαβάζει την ημερομηνία μέχρι 4 φορές. Έπειτα θα κλείσει το κανάλι επικοινωνίας. Παραθέτουμε το extension , το οποίο ονομάσαμε “965”.

```
exten => 965,1,Answer
exten => 965,n,Set(Retries=${Retries}+1)
exten => 965,n,DateTime
exten => 965,n,Gotof($[${Retries} <= 4]965,3)
exten => 965,3,Hangup
```

Παρατηρούμε αυτά που ήδη είπαμε. Στην 1^η προτεραιότητα , ανοίγει το κανάλι επικοινωνίας αφού καλέσουμε τον αριθμό “965”. Στην 2^η προτεραιότητα ορίζουμε έναν μετρητή για να μπορέσει να μετρήσει τις φορές που θα μας πει συνεχόμενα το σύστημα την ώρα. Μετά από αυτό μέσω της εφαρμογής “DateTime” , το σύστημα , αρχίζει και διαβάζει το εσωτερικό του ρολοί. Έπειτα μέσα από τα sound files του Asterisk , διαλέγει αυτά που χρειάζεται για να σχηματίσει την ώρα που διάβασε από

το ρολόι. Αφότου κάνει αυτό , προχωράει στην επόμενη προτεραιότητα και ελέγχει εάν ο μετρητής ξεπέρασε τον αριθμό 4. Εάν όχι , προσθέτει 1 στον μετρητή και αναγγέλλει ξανά την ώρα έως ότου φτάσει της 4 φορές , όπου εκεί προχωράει στην τελευταία προτεραιότητα και κλείνει το κανάλι επικοινωνίας.

5.3 Προγραμματισμός AGI

Μία από τις πιο εντυπωσιακές δυνατότητες του Asterisk, είναι αυτή της εκτέλεσης εξωτερικών προγραμμάτων, για την υλοποίηση πολύπλοκων ενεργειών και ανεπτυγμένης λογικής που με τη χρήση μόνο του dialplan θα ήταν αδύνατη. Το AGI έρχεται να υποκαταστήσει την ανάγκη για μία διεπαφή προγραμματισμού (API) παρέχοντας σε εξωτερικές εφαρμογές, τις απαραίτητες μεθόδους επικοινωνίας με το dialplan του Asterisk και κατά συνέπεια με το χειρισμό των κλήσεων.

5.3.1 Τι είναι το AGI

Τα αρχικά AGI προκύπτουν από τη σύμπτυξη των λέξεων Asterisk Gateway Interface. Το AGI είναι μία διεπαφή επικοινωνίας μεταξύ του Asterisk (dialplan) και εξωτερικών προγραμμάτων, παρόμοια με αυτήν που παρέχει το CGI στον HTTP εξυπηρετητή Apache. Όταν μία κλήση ανατεθεί σε κάποιο AGI script, τότε ο πλήρης έλεγχος της κλήσης αυτής, περνάει στο εξωτερικό πρόγραμμα που δείχνει το AGI στο dialplan και επανέρχεται στο Asterisk μετά το πέρας της λειτουργίας του.

Η επικοινωνία μεταξύ του Asterisk και της εξωτερικής εφαρμογής, επιτυγχάνεται μέσω ανταλλαγής μηνυμάτων απλού κειμένου (εντολή ↔ κωδικός επιστροφής) στις τυπικές ροές εισόδου, εξόδου και σφάλματος (stdin, stdout, stderr) του λειτουργικού UNIX. Αυτό σημαίνει πως οποιαδήποτε γλώσσα προγραμματισμού μπορεί να χρησιμοποιήσει τις παραπάνω ροές μπορεί και να αλληλεπιδράσει με το Asterisk. Όταν εκτελεστεί ένα AGI script, το Asterisk γράφει δεδομένα στο stdin του

προγράμματος και διαβάζει τα αποτελέσματα από τα stdout και stderr του προγράμματος.

Στον παρακάτω πίνακα αναφέρονται κάποιες από τις πιο συνηθισμένες διαθέσιμες AGI εντολές που μπορούν οι εξωτερικές εφαρμογές να στείλουν στο Asterisk.

AGI Εντολή	Περιγραφή
ANSWER	Απαντάει την κλήση, αν δεν έχει ήδη απαντηθεί.
CHANNEL STATUS <όνομα_καναλιου>	Επιστρέφει την κατάσταση του καναλιού.
GET DATA <αρχείο> [χρόνος] [ψηφία]	Αναπαραγάγει το αρχείο ήχου για συγκεκριμένο χρόνο και περιμένει μέχρι μέγιστο αριθμό ψηφίων.
GET VARIABLE <μεταβλητή>	Επιστρέφει την τιμή της channel μεταβλητής <μεταβλητή>.
HANGUP <όνομα_καναλιού>	Κλείνει το κανάλι.
SAY DIGITS <αριθμός>	Αναπαραγάγει τον <αριθμό> ψηφίο-ψηφίο.
SAY NUMBER <αριθμός>	Αναπαραγάγει τον <αριθμό>.
SAY ALPHA <λέξη>	Αναπαραγάγει την <λέξη>, γράμμα-γράμμα.
SAY DATE <ημερομηνία>	Αναπαραγάγει την ημερομηνία
SET CALLERID <αλφαριθμητικό>	Ορίζει τα στοιχεία αναγνώρισης κλήσεως.
SET CONTEXT <context>	Θέτει το τρέχων context όταν σταματήσει η λειτουργία του script.
SET EXTENSION <extension>	Θέτει την τρέχουσα extension όταν σταματήσει η λειτουργία του script.
SET PRIORITY <προτεραιότητα>	Θέτει την τρέχουσα προτεραιότητα όταν σταματήσει η λειτουργία του script.

<code>SET VARIABLE <μεταβλητή> <τιμή></code>	Ορίζει στην channel μεταβλητή <μεταβλητή> την <τιμή>.
<code>STREAM FILE <όνομα_αρχείου></code>	Αναπαραγάγει το αρχείο ήχου <όνομα_αρχείου>.
<code>WAIT FOR DIGIT <δευτερόλεπτα></code>	Περιμένει είσοδο για συγκεκριμένα <δευτερόλεπτα>.

Πίνακας 5.1 Εντολές AGI

Για κάθε εντολή που τροφοδοτείται στο Asterisk, επιστρέφεται στην εφαρμογή ένας κωδικός επιστροφής. Παρόλο που οι διαθέσιμοι κωδικοί επιστροφής είναι μόνο τρεις, ο κωδικός της επιτυχημένης εκτέλεσης εμφωλεύει πολλές περιπτώσεις όπως φαίνεται στον παρακάτω πίνακα:

Κωδικός Επιστροφής	Ορίσματα	Περιγραφή
200	"return=<τιμή>"	Επιστρέφεται αν το Asterisk εκτελέσει την εντολή. Στο πεδίο <τιμή> επιστρέφεται 0 ή 1 ανάλογα με το αν εκτελέστηκε με επιτυχία η εντολή ή εναλλακτικά αποστέλλεται η τιμή η οποία ζητήθηκε.
510	"invalid or unknown command"	Επιστέφεται αν το Asterisk δεν αναγνωρίσει την εντολή που του στάλθηκε.
520	"Διευκρίνιση Συνταξης"	Επιστρέφεται αν δεν έχει συνταχθεί σωστά η εντολή που στάλθηκε και συνοδεύεται από διευκρινήσεις σωστής σύνταξης της εντολής.

Πίνακας 5.2 Κωδικοί Επιστροφής

Υπάρχουν τέσσερις κατηγορίες AGI, η καθεμία από τις οποίες παρέχει ξεχωριστές δυνατότητες στο εξωτερικό πρόγραμμα που καλούν. Οι κατηγορίες αυτές είναι:

- AGI: Παρέχει πλήρη έλεγχο του dialplan σε εξωτερικές εφαρμογές χρησιμοποιώντας τα stdin, stdout, stderr. Καλείται στο dialplan ως:
exten => extension,προτεραιότητα,AGI(Script)
- EAGI: Λειτουργεί όπως το AGI αλλά παρέχει και έλεγχο του ηχητικού καναλιού επικοινωνίας. Καλείται στο dialplan ως:
exten => extension,προτεραιότητα,EAGI(Script)
- DeadAGI: Παρέχει πρόσβαση σε ένα κανάλι επικοινωνίας που έχει τερματιστεί. Καλείται στο dialplan ως:
exten => extension,προτεραιότητα,DeadAGI(Script)
- FastAGI: Παρέχει πλήρη έλεγχο του dialplan μέσω δικτύου με τη χρήση TCP/IP συνδέσεων. Καλείται στο dialplan ως:
exten=>extension,προτεραιότητα,AGI(agi://IP/Script)

5.3.2 Επιπλέον AGI Βιβλιοθήκες και Ανάπτυξη Αλγορίθμων

Εξαιτίας των ευκολιών σύνδεσης που παρέχει η διεπαφή AGI μεταξύ του Asterisk και των εξωτερικών εφαρμογών, πολλοί άνθρωποι ενδιαφέρθηκαν για το πώς θα μπορούσαν να απλοποιήσουν τη διαδικασία της σύνδεσης, προκειμένου ο προγραμματιστής να επικεντρωθεί στην εφαρμογή που θέλει να δημιουργήσει και όχι στον τρόπο που θα αυτή θα ανταλλάζει μηνύματα με το Asterisk. Από το ενδιαφέρον αυτό, έχουν προκύψει AGI βιβλιοθήκες για σχεδόν οποιαδήποτε γλώσσα προγραμματισμού μπορεί κάποιος να σκεφτεί: Perl, PHP, C, Java, Python, .NET, κ.α.

Έχοντας αυτές τις βιβλιοθήκες το μόνο που έχουμε να κάνουμε είναι να γράψουμε ένα αλγόριθμο στην επιθυμητή γλώσσα προγραμματισμού που θέλουμε και έπειτα απλά φορτώνουμε στο Asterisk τις βιβλιοθήκες αυτής της γλώσσας και καλούμε τον αλγόριθμο.

Για να γίνει αυτό κατανοητό συντάχθηκε ένας αλγόριθμος σε γλώσσα Perl , ο οποίος υλοποιεί μία εφαρμογή “Wake Up Call(αφύπνιση) “ στο τηλεφωνικό μας

κέντρο. Σκοπός μας δεν είναι να αναλύσουμε τον κώδικα που αναπτύχθηκε , αλλά να δείξουμε πόση ευελιξία υπάρχει στον Asterisk και ταυτόχρονα τις δυνατότητες που έχουμε για να υλοποιήσουμε το οτιδήποτε χρειαστούμε.

5.3.3 AGI Wake-Up Call Perl

Αφού γράψουμε τον αλγόριθμο μας σε Perl , τον τοποθετούμε στο παρακάτω path : `/var/lib/asterisk/agi-bin/` . Επίσης πρέπει να προσέξουμε τα παρακάτω paths να υπάρχουν και να έχουμε τα δικαιώματα στο σύστημα με τον Asterisk , ώστε να μπορούμε να εγγράψουμε μέσα σε αυτά : `/var/spool/asterisk/outgoing` και `/var/spool/asterisk/wakeups` .

Ο αλγόριθμος μας έχει όλες τις δυνατότητες που μπορεί να έχει ένα ξυπνητήρι. Μερικές από αυτές είναι :

- ❖ Προγραμματισμός απεριόριστων αφυπνίσεων
- ❖ Ανάγνωση και διαγραφή επιμέρους αφυπνίσεων
- ❖ Επανάληψη της αφύπνισης , εάν δεν απαντηθεί στην αρχική του κλήση.

Ο αλγόριθμος χρησιμοποιεί τους ήχους του Asterisk για να δημιουργήσει προτάσεις και να τις αναπαράγει για να δώσει τις αντίστοιχες οδηγίες για την εισαγωγή των δεδομένων. Η εισαγωγή των δεδομένων γίνεται μέσω των πλήκτρων του τηλεφώνου μας. Με την βοήθεια των *touch commands* του Unix το πάτημα των πλήκτρων μετατρέπεται σε δεδομένα , τα οποία καταχωρούνται σε κάποιο temporary αρχείο. Αφού οριστούν όλα τα στοιχεία , δημιουργείται ένα τελικό αρχείο , το οποίο έχει όλη την πληροφορία που χρειάζεται το σύστημα του Asterisk , ώστε να πραγματοποιήσει την προγραμματισμένη κλήση , ως ξυπνητήρι. Όταν έρθει η ώρα που έχουμε προγραμματίσει την αφύπνιση , ανοίγει αυτόματα ένα κανάλι επικοινωνίας και πραγματοποιεί μία κλήση στο extension από το οποίο προγραμματίσαμε το ξυπνητήρι. Εάν απαντηθεί η κλήση , ο asterisk μας ενημερώνει ότι αυτή η κλήση είναι τι ξυπνητήρι μας. Εάν δεν απαντηθεί τότε μετά από λίγο χρονικό διάστημα η κλήση ξανά πραγματοποιείται μέχρι να απαντηθεί η κλήση.

Τέλος το μόνο που μας μένει είναι πως θα καλούμε αυτόν τον αλγόριθμο. Προσθέτουμε στο dialplan μας τις παρακάτω γραμμές :

```
exten=>*63,1,Answer
```

```
exten=>*63,2,AGI(jwakeup.agi)
```

```
exten =>*63,3,Hangup
```

Όπως βλέπουμε η διαδικασία για να τρέξουμε ένα AGI script είναι πολύ απλή. Ανοίγει ένα κανάλι επικοινωνίας και μετά καλούμε τον αλγόριθμο μας με το ακριβές όνομα του αρχείου που θα εκτελεστεί. Όταν αυτός εκτελεστεί, το κανάλι επικοινωνίας κλείνει.

6. Προσομοίωση Λειτουργιών Asterisk

6.1 Εισαγωγή

Αφού λοιπόν περιγράψαμε αναλυτικά όλες τις παραμέτρους που χρειάστηκε να εισάγουμε στην πλατφόρμα του Asterisk , προχωρήσαμε στην ανάπτυξη διαφόρων υπηρεσιών και ευκολιών , οι οποίες είναι αντιπροσωπευτικές για να δείξουμε τις δυνατότητες την πλατφόρμας. Αυτό που θα παρουσιάσουμε σε αυτό το κεφάλαιο , είναι η προσημείωση όλων αυτών των υπηρεσιών , για να διαπιστωθεί η καλή λειτουργία τους. Ακολουθούν όλα τα screen shots μαζί με τον σχολιασμό τους για τις διάφορες λειτουργίες.

6.2 Dial To All Users

Παρακάτω θα πραγματοποιήσουμε απλές κλήσεις μεταξύ των χρηστών μας. Θα χρησιμοποιήσουμε όλους τους δυνατούς συνδυασμούς για να διαπιστωθεί η καλή λειτουργία και των τριών χρηστών μας.

Βλέπουμε τις εντολές που εκτελέστηκαν με την σειρά. Στην αρχή εκτελείτε το “*Noop*” μιας και είναι η πρώτη εντολή που έχουμε εισάγει στο dial plan μας. Έπειτα από αυτό εκτελείτε η εντολή “*Dial*” η οποία μας λέει από ποιον χρήστη εκτελείται και σε ποιος είναι ο προορισμός του. Μετά από αυτό , αφού βρει τον χρήστη προορισμού , μας αναφέρει το σύστημα ότι το τηλεφώνω του “*is ringing* (χτυπάει)”. Αφότου το χρήστης σηκώσει το τηλέφωνο του το σύστημα μας αναφέρει ότι η κλήση απαντήθηκε (“*answered*”). Τέλος , όταν ολοκληρωθεί η κλήση , τότε μας επιστρέφει το αντίστοιχο μήνυμα.

6.3 Voice Mail

Αφού διαπιστώσαμε ότι βγαίνουν όλες οι κλήσεις προς του χρήστες , θα δοκιμάσουμε τώρα το voicemail. Για αρχή θα καλέσουμε έναν από τους χρήστες μας και δεν θα απαντήσουμε την κλήση για να δούμε εάν μετά από 20 δευτερόλεπτα (όπως έχουμε ορίσει) θα μας προωθήσει στο voicemail. Παρακάτω φαίνεται το screen shot της διαδικασίας.

```

root@localhost:~
File Edit View Terminal Tabs Help
localhost*CLI>
localhost*CLI>
localhost*CLI>
localhost*CLI>
-- Executing [111@asterisk-pbx:1] NoOp("SIP/222-09c0e0a0", "") in new stack
-- Executing [111@asterisk-pbx:2] Dial("SIP/222-09c0e0a0", "SIP/111|20|tr")
in new stack
-- Called 111
-- SIP/111-09c0e0a0 is ringing
Nobody picked up in 20000 ms
-- Executing [111@asterisk-pbx:3] Goto("SIP/222-09c0e0a0", "111-NOANSWER|1")
in new stack
-- Goto (asterisk-pbx,111-NOANSWER,1)
-- Executing [111-NOANSWER@asterisk-pbx:1] VoiceMail("SIP/222-09c0e0a0", "u1
11") in new stack
-- <SIP/222-09c0e0a0> Playing 'vm-theperson' (language 'en')
-- <SIP/222-09c0e0a0> Playing 'digits/1' (language 'en')
-- <SIP/222-09c0e0a0> Playing 'digits/1' (language 'en')
-- <SIP/222-09c0e0a0> Playing 'digits/1' (language 'en')
-- <SIP/222-09c0e0a0> Playing 'vm-isunavail' (language 'en')
-- <SIP/222-09c0e0a0> Playing 'vm-intro' (language 'en')
-- <SIP/222-09c0e0a0> Playing 'beep' (language 'en')
Recording the message
-- x=0, open writing: /var/spool/asterisk/voicemail/default/111/tmp/Z4zT3e
format: wav49, 0x9c0e0a0
-- x=1, open writing: /var/spool/asterisk/voicemail/default/111/tmp/Z4zT3e
format: gsm, 0x9c0e0a0
-- x=2, open writing: /var/spool/asterisk/voicemail/default/111/tmp/Z4zT3e
format: wav, 0x9c0e0a0
-- User hung up
== Spawn extension (asterisk-pbx, 111-NOANSWER, 1) exited non-zero on 'SIP/222
-09c0e0a0'
localhost*CLI>

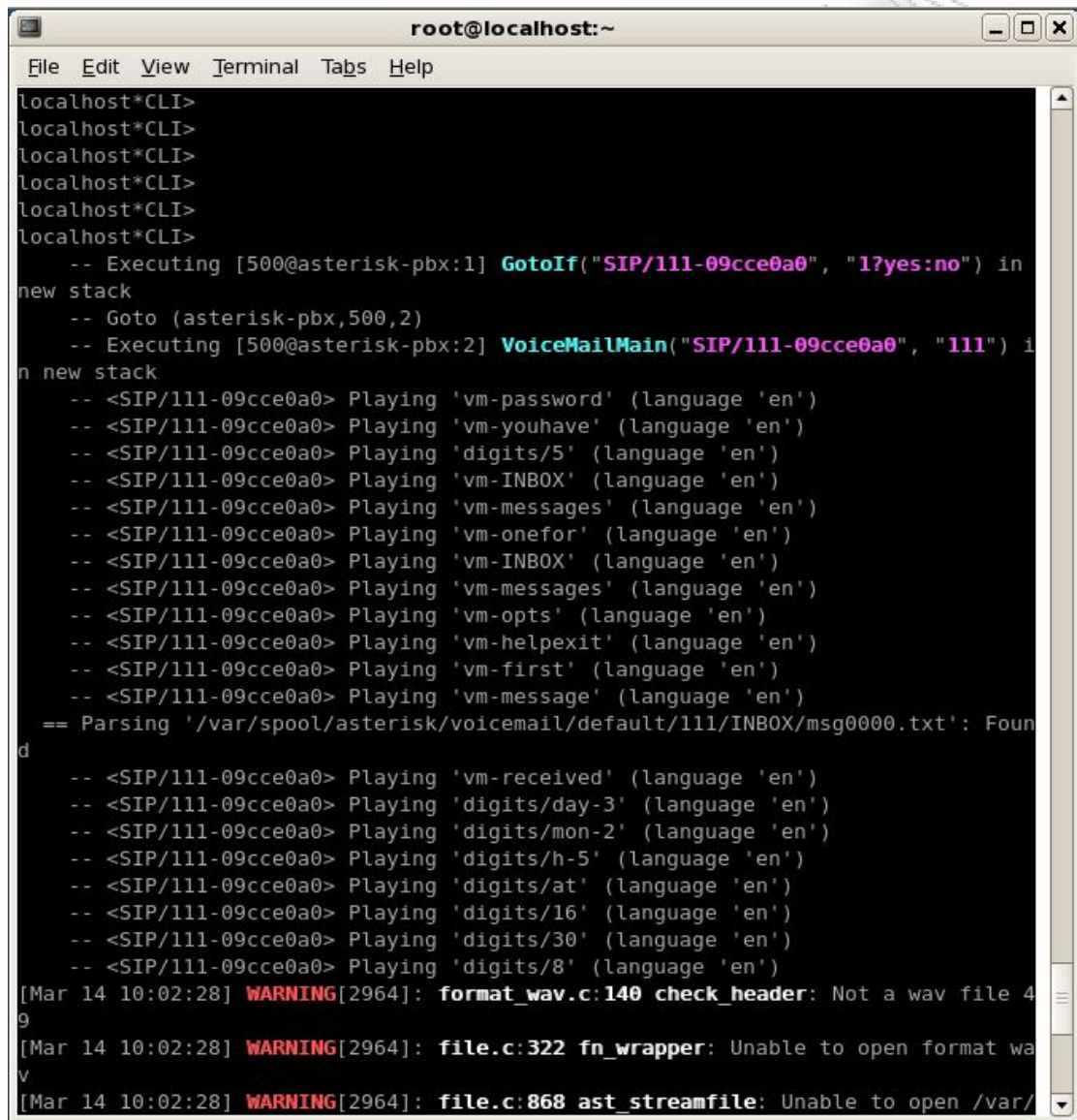
```

Εικόνα 6.9 Προώθηση σε Voicemail μετά από κλήση σε χρήστη

Παρατηρούμε ότι ο χρήστης “222” επιχειρεί να κάνει κλήση στον χρήστη “111”. Το κανάλι επικοινωνίας ανοίγει και αρχίζει να χτυπάει η συσκευή του 111. Αφού περάσουν 20 δευτερόλεπτα, ενεργοποιείται η προώθηση η οποία μας στέλνει στον τηλεφωνητή (voicemail). Αφού μας ενημερώνει ότι ο χρήστης δεν είναι διαθέσιμος, μας παροτρύνει να αφήσουμε κάποιο μήνυμα. Αφήνουμε το μήνυμα μας και κλείνουμε την γραμμή.

Η επόμενη κίνηση είναι να πάμε να ακούσουμε το μήνυμα που μόλις αφήσαμε με 2 τρόπους. Ο πρώτος τρόπος είναι να πάμε να ακούσουμε το μήνυμα από την συσκευή του χρήστη “111” και μετά να ακούσουμε τα μηνύματα αυτά από την συσκευή άλλου χρήστη κάνοντας χρήση της απομακρυσμένης πρόσβασης που μας διατίθεται. Επειδή οι εντολές που εκτελούνται από τον Asterisk είναι πολλές και δεν

χώρεσαν σε ένα Screenshot , αναγκαστήκαμε να βάλουμε 2 (“α” και “β”) , στα οποία το “β” είναι συνέχεια του “α”.



```
root@localhost:~
File Edit View Terminal Tabs Help
localhost*CLI>
localhost*CLI>
localhost*CLI>
localhost*CLI>
localhost*CLI>
localhost*CLI>
-- Executing [500@asterisk-pbx:1] GotoIf("SIP/111-09cce0a0", "1?yes:no") in
new stack
-- Goto (asterisk-pbx,500,2)
-- Executing [500@asterisk-pbx:2] VoiceMailMain("SIP/111-09cce0a0", "111") i
n new stack
-- <SIP/111-09cce0a0> Playing 'vm-password' (language 'en')
-- <SIP/111-09cce0a0> Playing 'vm-youhave' (language 'en')
-- <SIP/111-09cce0a0> Playing 'digits/5' (language 'en')
-- <SIP/111-09cce0a0> Playing 'vm-INBOX' (language 'en')
-- <SIP/111-09cce0a0> Playing 'vm-messages' (language 'en')
-- <SIP/111-09cce0a0> Playing 'vm-onefor' (language 'en')
-- <SIP/111-09cce0a0> Playing 'vm-INBOX' (language 'en')
-- <SIP/111-09cce0a0> Playing 'vm-messages' (language 'en')
-- <SIP/111-09cce0a0> Playing 'vm-opts' (language 'en')
-- <SIP/111-09cce0a0> Playing 'vm-helpexit' (language 'en')
-- <SIP/111-09cce0a0> Playing 'vm-first' (language 'en')
-- <SIP/111-09cce0a0> Playing 'vm-message' (language 'en')
== Parsing '/var/spool/asterisk/voicemail/default/111/INBOX/msg0000.txt': Foun
d
-- <SIP/111-09cce0a0> Playing 'vm-received' (language 'en')
-- <SIP/111-09cce0a0> Playing 'digits/day-3' (language 'en')
-- <SIP/111-09cce0a0> Playing 'digits/mon-2' (language 'en')
-- <SIP/111-09cce0a0> Playing 'digits/h-5' (language 'en')
-- <SIP/111-09cce0a0> Playing 'digits/at' (language 'en')
-- <SIP/111-09cce0a0> Playing 'digits/16' (language 'en')
-- <SIP/111-09cce0a0> Playing 'digits/30' (language 'en')
-- <SIP/111-09cce0a0> Playing 'digits/8' (language 'en')
[Mar 14 10:02:28] WARNING[2964]: format_wav.c:140 check_header: Not a wav file 4
9
[Mar 14 10:02:28] WARNING[2964]: file.c:322 fn_wrapper: Unable to open format wa
v
[Mar 14 10:02:28] WARNING[2964]: file.c:868 ast_streamfile: Unable to open /var/
```

Εικόνα 6.10 Ανάκτηση μηνυμάτων (α)

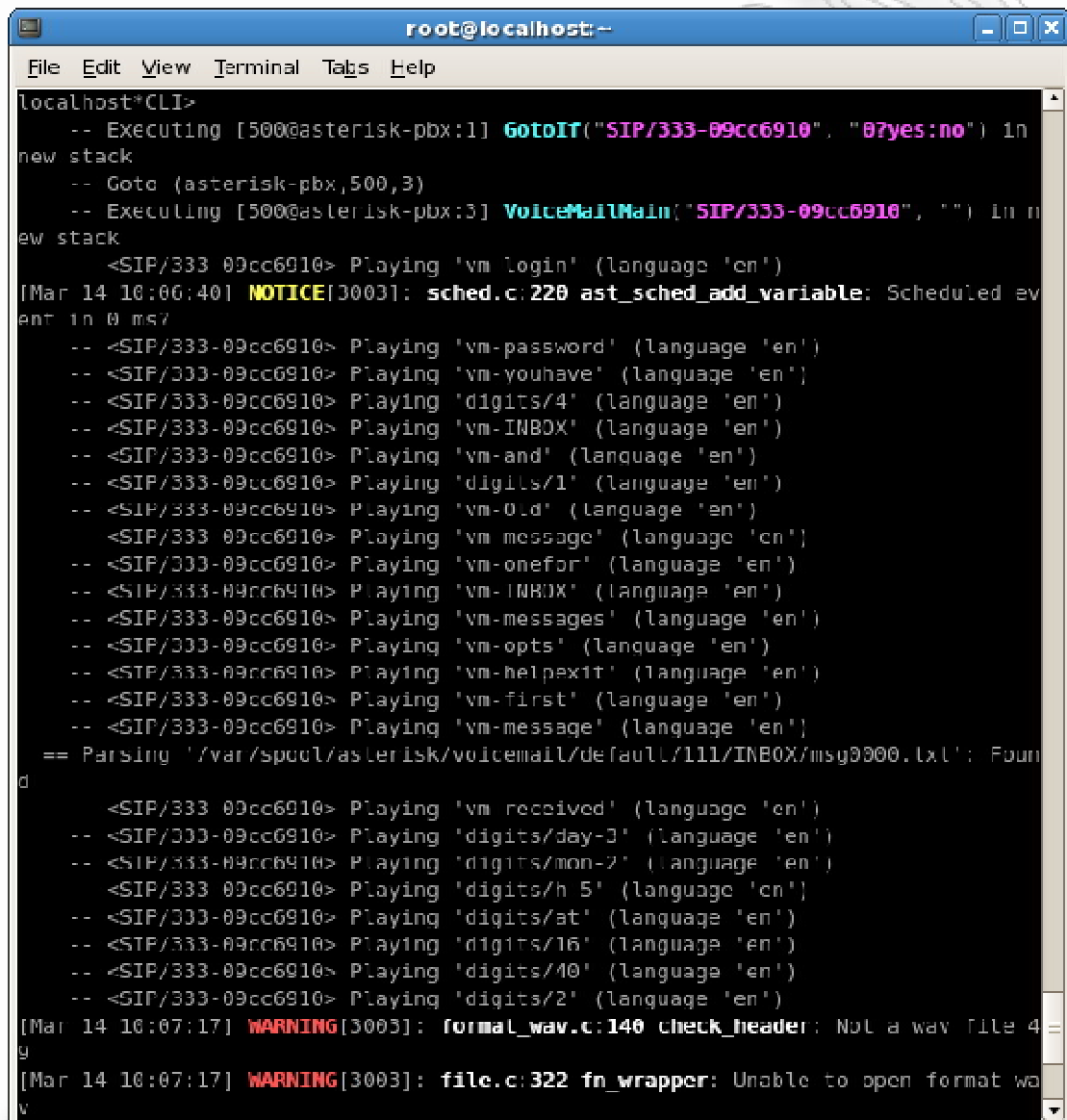
```
root@localhost:~
File Edit View Terminal Tabs Help
-- <SIP/111-09c0e0a0> Playing 'digits/at' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'digits/16' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'digits/30' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'digits/8' (language 'en')
[Mar 14 10:02:28] WARNING[2964]: format wav.c:140 check_header: Not a wav file 4
9
[Mar 14 10:02:28] WARNING[2964]: file.c:322 fn_wrapper: Unable to open format wa
v
[Mar 14 10:02:28] WARNING[2964]: file.c:868 ast_streamfile: Unable to open /var/
spool/asterisk/voicemail/default/111/INBOX/msg0000 (format 0x4 (ulaw)): No such
file or directory
[Mar 14 10:02:28] WARNING[2964]: app_voicemail.c:4662 play_message: Playback of
message /var/spool/asterisk/voicemail/default/111/INBOX/msg0000 failed
-- <SIP/111-09c0e0a0> Playing 'vm-advopt's' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm-repeat' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm-next' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm-delete' (language 'en')
[Mar 14 10:02:38] NOTICE[2954]: sched.c:220 ast_sched_add_variable: Scheduled ev
ent in 0 ms7
-- <SIP/111-09c0e0a0> Playing 'vm-toforward' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm-savemessage' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm-helpexit' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm-deleted' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm-advopts' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm-repeat' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm-next' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm-undelete' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm-toforward' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm-savemessage' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm undeleted' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm-advopts' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm repeat' (language 'en')
-- <SIP/111-09c0e0a0> Playing 'vm-next' (language 'en')
[Mar 14 10:03:09] WARNING[2964]: file.c:644 ast_readaudio_callback: Failed to wr
ite frame
-- Spawn extension (asterisk-pbx, 500, 2) exited non-zero on 'SIP/111-09c0e0a0
localhost*CLI>
```

Εικόνα 6.11 Ανάκτηση μηνυμάτων (β)

Αφού λοιπόν καλέσει ο χρήστης το νούμερο του τηλεφωνητή (500) , θα παροτρύνει το σύστημα να εισάγουμε τον αριθμό της θυρίδας μας , ο οποίος πρέπει συμπίπτει με τον αριθμό της συσκευή μας. Στην προκειμένη περίπτωση συμπίπτει γι αυτό προχωράει η διαδικασία αναγνώρισης , στην οποία πρέπει να εισάγουμε το password μας. Αφού το εισάγουμε , μπαίνουμε στο μενού του τηλεφωνητή , όπου εκεί μπορούμε να κάνουμε ένα πλήθος από λειτουργίες , όπως να ακούσουμε τα μηνύματα μας , να τα σβήσουμε , να αλλάξουμε ρυθμίσει και άλλα. Στην προκειμένη

περίπτωση αλλά περιηγηθήκαμε στα μενού του τηλεφωνητή για να διαπιστωθεί η καλή λειτουργία αυτής της δυνατότητας.

Παρακάτω θα ακολουθηθεί η ίδια διαδικασία , όμως θα προσπαθήσουμε να μπούμε στη θυρίδα “111” από την συσκευή “333”.



```
root@localhost:~#
File Edit View Terminal Tabs Help
localhost*CLI>
-- Executing [500@asterisk-pbx:1] GotoIT("SIP/333-09cc6910", "0?yes:no") in
new stack
-- Goto (asterisk-pbx,500,3)
-- Executing [500@asterisk-pbx:3] VoiceMailMain("SIP/333-09cc6910", "") in n
ew stack
<SIP/333-09cc6910> Playing 'vm login' (language 'en')
[Mar 14 10:06:40] NOTICE[3003]: sched.c:220 ast_sched_add_variable: Scheduled ev
ent in 0 ms?
-- <SIP/333-09cc6910> Playing 'vm-password' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-youhave' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/4' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-INBOX' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-and' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/1' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-0ld' (language 'en')
<SIP/333-09cc6910> Playing 'vm message' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-onefor' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-INBOX' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-messages' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-opts' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-helpexit' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-first' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-message' (language 'en')
== Parsing '/var/spool/asterisk/voicemail/default/111/INBOX/msg0000.txt': Four
d
<SIP/333-09cc6910> Playing 'vm received' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/day-3' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/mon-2' (language 'en')
<SIP/333-09cc6910> Playing 'digits/h 5' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/at' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/16' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/40' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/2' (language 'en')
[Mar 14 10:07:17] WARNING[3003]: format_wav.c:140 check_header: Not a wav file 4
y
[Mar 14 10:07:17] WARNING[3003]: file.c:322 fn_wrapper: Unable to open format wav
```

Εικόνα 6.12 Ανάκτηση μηνυμάτων της θυρίδας "111" από άλλη συσκευή (α)

```
root@localhost:~
File Edit View Terminal Tabs Help
-- <SIP/333-09cc6910> Playing 'vm-onefor' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-INBOX' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-messages' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-opts' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-helpexit' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm first' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-message' (language 'en')
== Parsing '/var/spool/asterisk/voicemail/default/111/INBOX/msg0000.txt': Found
d
-- <SIP/333-09cc6910> Playing 'vm-received' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/day-3' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/mon-2' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/h-5' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/al' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/16' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/40' (language 'en')
-- <SIP/333-09cc6910> Playing 'digits/2' (language 'en')
[Mar 14 18:07:17] WARNING[3003]: format_wav.c:140 check_header: Not a wav file 4
0
[Mar 14 18:07:17] WARNING[3003]: file.c:322 fn_wrapper: Unable to open format wa
v
[Mar 14 18:07:17] WARNING[3003]: file.c:868 ast_streamfile: Unable to open /var/
spool/asterisk/voicemail/default/111/INBOX/msg0000 {format 0x4 {ulaw}}: No such
file or directory
[Mar 14 18:07:17] WARNING[3003]: app_voicemail.c:4662 play_message: Playback of
message /var/spool/asterisk/voicemail/default/111/INBOX/msg0000 failed
-- <SIP/333-09cc6910> Playing 'vm-advopts' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-repeat' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm next' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-delete' (language 'en')
[Mar 14 18:07:26] NOTICE[3003]: sched.c:220 ast_sched_add_variable: Scheduled ev
ent in 0 ms?
-- <SIP/333-09cc6910> Playing 'vm-toforward' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-savemessage' (language 'en')
-- <SIP/333-09cc6910> Playing 'vm-helpexit' (language 'en')
-- Spawn extension (asterisk-pbx, 500, 3) exited non-zero on 'SIP/333-09cc6910
'
localhost*CLI>
```

Εικόνα 6.13 Ανάκτηση μηνυμάτων της θυρίδας "111" από άλλη συσκευή (β)

Σε αυτή την περίπτωση δοκιμάζουμε να εισέλθουμε στην θυρίδα "111" από τον χρήστη "333". Αφού καλούμε το 500, η υπηρεσία του τηλεφωνητή μας παροτρύνει να εισάγουμε τον αριθμό της θυρίδας που θέτουμε να εισέλθουμε. Εμείς βάζουμε το 111. Γίνεται ένας έλεγχος στο εάν ο αριθμός αυτός μπορεί να ανακτήσει τη θυρίδα του απομακρυσμένα ή όχι. Φυσικά ο αριθμός 111, βρίσκεται μέσα στην λίστα και έτσι η διαδικασία προχωράει όπως προηγουμένως. Εάν δεν υπήρχε στην λίστα θα μας έβγαζε ένα μήνυμα λάθους και θα μας έλεγε να εισάγουμε ξανά την σωστή θυρίδα.

πληκτρολογεί τον αριθμό που θα μεταφερθεί ο “111”. Αφού γίνει αυτό , η κλήση μεταφέρετε στον χρήστη “222” , όπου και αρχίζει να χτυπάει η συσκευή του. Αφού το σηκώσει πραγματοποιείται σύνδεση με τον χρήστη “111”.

Επίσης το σύστημα μας είναι ρυθμισμένο για να εκτελεί και εποπτευόμενη μεταφορά κλήσης (atxfer). Εάν πραγματοποιούσαμε αυτή την μεταφορά κλήσης , τότε η μόνη διαφορά θα ήταν ότι θα περίμενε ο χρήστης “333” να απαντηθεί η κλήση από τον “222” και μετά να μεταφέρει την κλήση στον “111”.

6.5 Date and Time

Η ανακοίνωση της ημερομηνίας και της ώρας , είναι κάτι πολύ γνωστό , σαν υπηρεσία από τα χρόνια , της αναλογικής τηλεφωνίας. Η πλατφόρμα του Asterisk , δεν θα μπορούσε να μην το κάνει αυτό. Η διαδικασία είναι απλή , στην πραγματοποίησή της. Το μόνο που θα πρέπει να προσέξουμε , είναι ότι η εντολή αυτή χρησιμοποιεί τα standard sound files της εγκατάστασης του Asterisk και δεν μπορεί να χρησιμοποιηθεί με βιβλιοθήκες ήχων. Αφού προσέξουμε αυτή την λεπτομέρεια τότε μπορούμε να προχωρήσουμε στην εκτέλεση της εφαρμογής.

```

root@localhost:~
File Edit View Terminal Tabs Help
localhost*CLI>
localhost*CLI>
localhost*CLI>
localhost*CLI>
localhost*CLI>
localhost*CLI>
-- Executing [100@asterisk-pbx:1] Answer("SIP/333-09cd3830", "") in new stack
-- Executing [100@asterisk-pbx:2] Background("SIP/333-09cd3830", "Intro_") in new stack
-- <SIP/333-09cd3830> Playing 'Intro_' (language 'en')
== CDR updated on SIP/333-09cd3830
-- Executing [965@asterisk-pbx:1] Answer("SIP/333-09cd3830", "") in new stack
-- Executing [965@asterisk-pbx:2] Set("SIP/333-09cd3830", "Retries==+1") in new stack
-- Executing [965@asterisk-pbx:3] DateTime("SIP/333-09cd3830", "") in new stack
-- <SIP/333-09cd3830> Playing 'digits/day-5' (language 'en')
-- <SIP/333-09cd3830> Playing 'digits/mon-2' (language 'en')
-- <SIP/333-09cd3830> Playing 'digits/h-14' (language 'en')
-- <SIP/333-09cd3830> Playing 'digits/2' (language 'en')
-- <SIP/333-09cd3830> Playing 'digits/thousand' (language 'en')
-- <SIP/333-09cd3830> Playing 'digits/8' (language 'en')
-- <SIP/333-09cd3830> Playing 'digits/at' (language 'en')
-- <SIP/333-09cd3830> Playing 'digits/10' (language 'en')
-- <SIP/333-09cd3830> Playing 'digits/17' (language 'en')
-- <SIP/333-09cd3830> Playing 'digits/a-m' (language 'en')
[Mar 14 10:17:48] WARNING[3063]: ast_expr2.fl:407 ast_yyerror: ast_yyerror(): syntax error: syntax error, unexpected '+', expecting $end; Input:
+1 <= 4
^
[Mar 14 10:17:48] WARNING[3063]: ast_expr2.fl:411 ast_yyerror: If you have questions, please refer to doc/channelvariables.txt in the asterisk source.
-- Executing [965@asterisk-pbx:4] GotoIf("SIP/333-09cd3830", "965|3") in new stack
== Auto fallback, channel: 'SIP/333-09cd3830' status is 'UNKNOWN'
localhost*CLI>

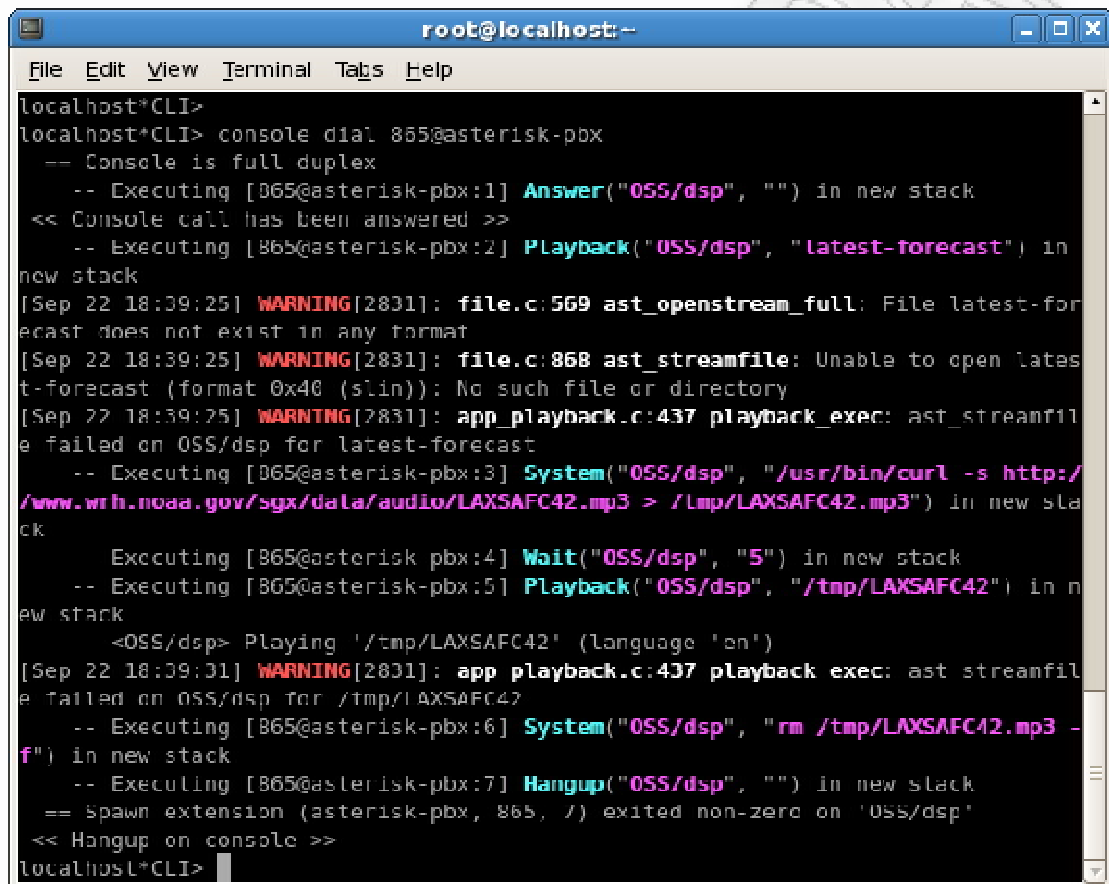
```

Εικόνα 6.11 Ανάγνωση Ημερομηνίας και Ώρας

Ο χρήστης “333” καλεί το κέντρο (100), όπου εκεί ενημερώνεται ότι για να ακούσει την ημερομηνία και την ώρα θα , πρέπει να καλέσει τον αριθμό “965”. Αφού το κάνει αυτό , το κανάλι ανοίγει αυτόματα και το σύστημα ξεκινά να αναφωνεί την ημερομηνία και την ώρα που διαβάζει εκείνη την στιγμή από το εσωτερικό του ρολόι. Για να μπορέσουμε να το κάνουμε ποιο ρεαλιστικό , προσθέσαμε μία συνθήκη η οποία επαναλαμβάνει την ανάγνωση της ημερομηνίας και ώρας για 4 φορές , εφ’ όσον ο χρήστης δεν κλείσει το ακουστικό νωρίτερα. Εάν δεν το κάνει στην 4^η επανάληψη , η γραμμή κλείνει από μόνη της.

6.6 Weather Forecast

Με την εφαρμογή αυτή , διαπιστώνουμε ότι με τον Asterisk , μπορούμε να κάνουμε σχεδόν τα πάντα. Η προσημείωση αυτής της εφαρμογής φαίνεται παρακάτω.



```
root@localhost:~
File Edit View Terminal Tabs Help
localhost*CLI>
localhost*CLI> console dial 865@asterisk-pbx
-- Console is full duplex
-- Executing [065@asterisk-pbx:1] Answer("OSS/dsp", "") in new stack
<< Console call has been answered >>
-- Executing [865@asterisk-pbx:2] Playback("OSS/dsp", "latest-forecast") in
new stack
[Sep 22 18:39:25] WARNING[2831]: file.c:569 ast_openstream_full: File latest-for
ecast does not exist in any format
[Sep 22 18:39:25] WARNING[2831]: file.c:868 ast_streamfile: Unable to open lates
t-forecast (format 0x40 (slin)): No such file or directory
[Sep 22 18:39:25] WARNING[2831]: app_playback.c:437 playback_exec: ast_streamfil
e failed on OSS/dsp for latest-forecast
-- Executing [065@asterisk-pbx:3] System("OSS/dsp", "/usr/bin/curl -s http://
/www.wrh.noaa.gov/sgx/data/audio/LAXSAFC42.mp3 > /tmp/LAXSAFC42.mp3") in new sta
ck
-- Executing [865@asterisk-pbx:4] Wait("OSS/dsp", "5") in new stack
-- Executing [865@asterisk-pbx:5] Playback("OSS/dsp", "/tmp/LAXSAFC42") in n
ew stack
<OSS/dsp> Playing '/tmp/LAXSAFC42' (language 'en')
[Sep 22 18:39:31] WARNING[2831]: app_playback.c:437 playback_exec: ast streamfil
e failed on OSS/dsp for /tmp/LAXSAFC42
-- Executing [865@asterisk-pbx:6] System("OSS/dsp", "rm /tmp/LAXSAFC42.mp3 -
f") in new stack
-- Executing [865@asterisk-pbx:7] Hangup("OSS/dsp", "") in new stack
== Spawn extension (asterisk-pbx, 865, /) exited non-zero on 'OSS/dsp'
<< Hangup on console >>
localhost*CLI>
```

Εικόνα 6.12 Πρόγνωση Καιρού

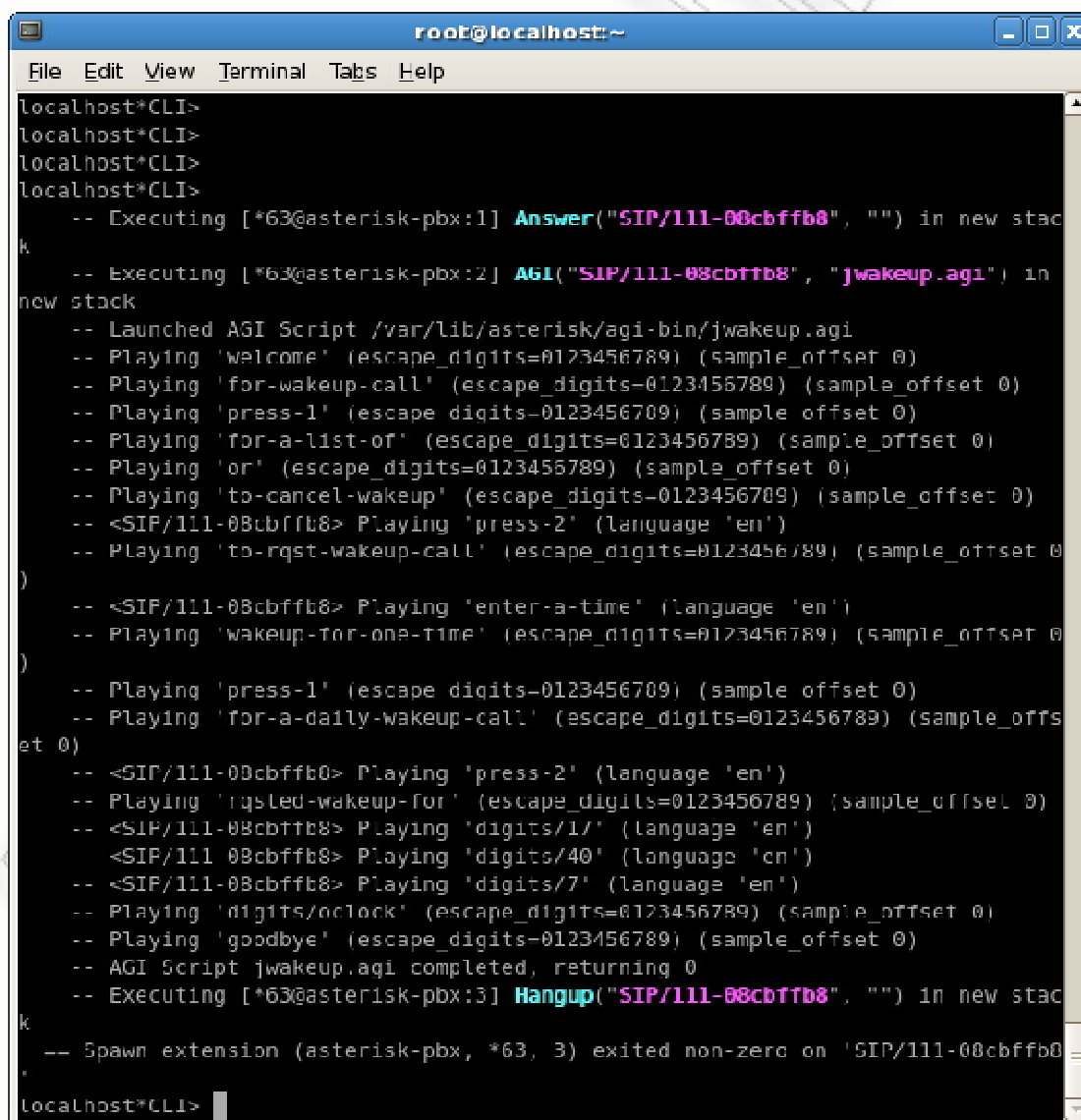
Όπως βλέπουμε , με το που καλούμε το καλούμε το κέντρο μας (100) , ενημερωνόμαστε ότι για την πρόγνωση του καιρού πρέπει να πληκτρολογήσουμε το “865”.

Αφού λοιπόν το πληκτρολογήσουμε αναπαράγεται το αρχείο που μας ενημερώνει να περιμένουμε έως ότου κατέβει το αρχείο της πρόγνωσης. Αφού λοιπόν κατέβει η πρόγνωση αρχίζει και αναπαράγεται. Μόλις αυτή ολοκληρωθεί , τότε το αρχείο , διαγράφεται από το σύστημα μας , για να μην γίνει μπέρδεμα την επόμενη φορά που θα ξανακαλέσουμε το “865” και η γραμμή μας κλείνει.

6.7 Wakeup Call

Όπως αναφέραμε και στο προηγούμενο μας κεφάλαιο , το AGI , είναι μία από τις πιο εντυπωσιακές δυνατότητες του Asterisk και αυτό γιατί σου δίνει την δυνατότητα να εκτελέσεις κάποιον αλγόριθμό , τον οποίο τον έχεις γράψει σε μια από τις διάφορες γνώστες γλώσσες προγραμματισμού.

Παρακάτω θα δούμε την προσημείωση της λειτουργίας αφύπνισης (wakeup call) , η οποία αναπτύχθηκε σε γλώσσα Perl. Επειδή οι λειτουργίες του wakeup call είναι αρκετές , εμείς θα προσημειώσουμε μία απλή αφύπνιση.



```
root@localhost:~  
File Edit View Terminal Tabs Help  
localhost*CLI>  
localhost*CLI>  
localhost*CLI>  
localhost*CLI>  
-- Executing [*63@asterisk-pbx:1] Answer("SIP/111-08cbffb8", "") in new stack  
-- Executing [*63@asterisk-pbx:2] AGI("SIP/111-08cbffb8", "jwakeup.agi") in new stack  
-- Launched AGI Script /var/lib/asterisk/agi-bin/jwakeup.agi  
-- Playing 'welcome' (escape_digits=0123456789) (sample_offset 0)  
-- Playing 'for-wakeup-call' (escape_digits=0123456789) (sample_offset 0)  
-- Playing 'press-1' (escape_digits=0123456709) (sample_offset 0)  
-- Playing 'for-a-list-of' (escape_digits=0123456789) (sample_offset 0)  
-- Playing 'or' (escape_digits=0123456789) (sample_offset 0)  
-- Playing 'to-cancel-wakeup' (escape_digits=0123456709) (sample_offset 0)  
-- <SIP/111-08cbffb8> Playing 'press-2' (language 'en')  
-- Playing 'to-rqst-wakeup-call' (escape_digits=0123456789) (sample_offset 0)  
-- <SIP/111-08cbffb8> Playing 'enter-a-time' (language 'en')  
-- Playing 'wakeup-for-one-time' (escape_digits=0123456789) (sample_offset 0)  
-- Playing 'press-1' (escape_digits=0123456709) (sample_offset 0)  
-- Playing 'for-a-daily-wakeup-call' (escape_digits=0123456789) (sample_offset 0)  
-- <SIP/111-08cbffb8> Playing 'press-2' (language 'en')  
-- Playing 'requested-wakeup-for' (escape_digits=0123456789) (sample_offset 0)  
-- <SIP/111-08cbffb8> Playing 'digits/1/' (language 'en')  
-- <SIP/111-08cbffb8> Playing 'digits/40' (language 'en')  
-- <SIP/111-08cbffb8> Playing 'digits/7' (language 'en')  
-- Playing 'digits/oclock' (escape_digits=0123456789) (sample_offset 0)  
-- Playing 'goodbye' (escape_digits=0123456789) (sample_offset 0)  
-- AGI Script jwakeup.agi completed, returning 0  
-- Executing [*63@asterisk-pbx:3] Hangup("SIP/111-08cbffb8", "") in new stack  
-- Spawn extension (asterisk-pbx, *63, 3) exited non-zero on 'SIP/111-08cbffb8'  
localhost*CLI>
```

Εικόνα 6.13 Εισαγωγή Αφύπνισης

Η λειτουργία της αφύπνισης , ενεργοποιείται με την πληκτρολόγηση του αριθμού “*63”. Να σημειώσουμε εδώ πως η αφύπνιση θα πραγματοποιηθεί στον αριθμό από τον οποίο θα καλέσουμε. Αφού καλέσουμε τον αριθμό αυτό , μπαίνουμε στο κεντρικό μενού της αφύπνισης. Το σύστημα ,αφότου μας καλωσορίσει , μας ενημερώνει πως εάν θέλουμε να εισάγουμε μία αφύπνιση , τότε να πατήσουμε το “1”. Εάν θέλουμε να ακούσουμε ή να ακυρώσουμε , τις αφυπνίσεις που ήδη υπάρχουν , να πατήσουμε το “2”.

Εμείς θα πατήσουμε το “1” , μιας και όπως ήδη είπαμε θα προσημειώσουμε μια απλή αφύπνιση. Πατώντας το “1” , θα ενημερωθούμε , εάν θέλουμε 1 αφύπνιση ή θέλουμε επαναλαμβανόμενες αφυπνίσεις. Επιλέγουμε πάλι το “1”. Τώρα το σύστημα , μας ζητάει να εισάγουμε την ώρα αφύπνισης που επιθυμούμε. Αφού λοιπόν την εισάγουμε πατώντας τα αντίστοιχα πλήκτρα του τηλεφώνου μας (εισάγαμε 17:47) , το σύστημα αποθηκεύει την αφύπνιση και μας κλείνει την γραμμή , αφότου μας αποχαιρετήσει.

Αυτό που πρέπει να κάνουμε τώρα είναι να περιμένουμε ως την ώρα που ορίσαμε για να δούμε εάν θα επαληθευτεί η αφύπνιση. Φυσικά , αυτό πραγματοποιείται και παρακάτω βλέπουμε την επαλήθευση.

```
root@localhost:~  
File Edit View Terminal Tabs Help  
localhost*CLI>  
localhost*CLI>  
localhost*CLI>  
localhost*CLI>  
localhost*CLI>  
localhost*CLI>  
localhost*CLI>  
localhost*CLI>  
localhost*CLI>  
localhost*CLI>  
localhost*CLI>  
localhost*CLI>  
-- Attempting call on Local/111@asterisk-pbx for application AGI(jwakeup.agi  
[1747.temp.1.111.call) (Retry 1)  
-- Executing [111@asterisk-pbx:1] NoOp("Local/111@asterisk-pbx-b03a,2", "")  
in new stack  
-- Executing [111@asterisk-pbx:2] Dial("Local/111@asterisk-pbx-b03a,2", "SIP  
/111]20|tr") in new stack  
-- Called 111  
-- SIP/111-08cbffb8 is ringing  
SIP/111-08cbffb8 answered Local/111@asterisk pbx b03a,2  
[Apr 22 17:47:04] WARNING[2928]: pbx.c:5170 ast pbx outgoing app: Local/111@aste  
risk-pbx-b03a,1 already has a call record??  
-- Launched AGI Script /var/lib/asterisk/agi-bin/jwakeup.agi  
-- Spawn extension (asterisk-pbx, 111, 2) exited non-zero on 'Local/111@asteri  
sk-pbx-b03a,2'  
-- Playing 'this-is-yr-wakeup-call' (escape_digits=0123456789) (sample_offset  
t 0)  
-- Playing 'lu-confirm-wakeup' (escape_digits=0123456789) (sample_offset 0)  
-- Playing 'press-1' (escape_digits=0123456789) (sample_offset 0)  
Playing 'to snooze for' (escape_digits=0123456789) (sample_offset 0)  
-- Playing 'digits/10' (escape_digits=0123456789) (sample_offset 0)  
-- Playing 'minutes' (escape_digits=0123456789) (sample_offset 0)  
-- <SIP/111-08cbffb8> Playing 'press-2' (language 'en')  
-- Playing 'goodbye' (escape_digits=0123456709) (sample_offset 0)  
-- AGI Script jwakeup.agi completed, returning 0  
[Apr 22 17:47:14] NOTICE[2928]: pbx_spool.c:351 attempt_thread: Call completed t  
o Local/111@asterisk pbx  
localhost*CLI>
```

Εικόνα 6.14 Εκτέλεση Αφύπνισης

Όπως παρατηρούμε από την παραπάνω εικόνα , το σύστημα , ανοίγει από μόνο του κανάλι επικοινωνίας την ώρα που του είχαμε ορίσει και αρχίζει και καλεί στον αριθμό από τον οποίο ορίσαμε την αφύπνιση. Εμείς με το που απαντήσουμε την κλήση , το σύστημα , μας ενημερώνει ότι αυτή η κλήση είναι η αφύπνιση μας και συνεχίζει λέγοντας εάν θέλουμε να την ακυρώσουμε ή να ξανακαλέσει μετά από 10 λεπτά (λειτουργία snooze). Εάν πατήσουμε το “1” επιβεβαιώνεται η αφύπνιση και πάυει να υφίσταται , ενώ πατώντας το “2” (ή κλείνοντας το ακουστικό) η αφύπνιση , θα επαναληφθεί σε 10 λεπτά. Φυσικά εμείς πατάμε το “1” και η κλήση τερματίζεται , αφότου διαγραφεί από το σύστημα η συγκεκριμένη αφύπνιση.

7. Επίλογος και Μελλοντικές Προτάσεις

7.1 Επίλογος

Ο Asterisk έγινε το καθιερωμένο σύστημα για μεταφορά φωνής και λειτουργίες PBX , στην κοινότητα του ανοιχτού κώδικα. Η φιλοξενία διαδικασιών όπως TDM και DSP , έκαναν δυνατόν να χρησιμοποιήσουμε φτηνό hardware ώστε να μπορέσουμε να υλοποιήσουμε μία ευέλικτη πλατφόρμα PBX. Υποστηρίζοντας πολλά πρωτόκολλα φωνής και προσφέροντας υψηλού επιπέδου εφαρμογές PBX , ο Asterisk έγινε μία αποτελεσματική πλατφόρμα μέσα σε ένα εικονικό τηλεφωνικό κέντρο. Μεγαλώνοντας η υποστήριξη του από την κοινότητα ανοιχτού λογισμικού , αυτό βοηθάει τον Asterisk να μένει αξιόπιστος και ενημερωμένος με τις νέες τεχνολογίες που κυκλοφορούν. Ταυτόχρονα όμως η έλλειψη εμπορικού ενδιαφέροντος , προκαλεί στον Asterisk το μειωμένο ενδιαφέρον με αποτέλεσμα να υπάρχουν σημεία στην πλατφόρμα που χρίζουν μειωμένη ευκολία στην χρήση τους και ταυτόχρονα διάφορες λύσεις που δίνονται για αυτόν τον λόγο είναι ξεπερασμένες.

7.2 Μελλοντικές Προτάσεις

Σε αυτή την διπλωματική εργασία πραγματοποιήθηκε η ανάπτυξη προηγμένων εφαρμογών VoIP στην πλατφόρμα του Asterisk οι οποίες αναδεικνύουν ένα μέρος των δυνατοτήτων της πλατφόρμας. Με αφετηρία αυτή την εργασία θα μπορούσαν να εισαχθούν μια σειρά βελτιώσεων και αναβαθμίσεων οι οποίες θα καθιστούσαν τον Asterisk

πιο ρεαλιστικό και ποιο ευέλικτο. Οι χρήσιμες εφαρμογές που θα μπορούσαμε να αναπτύξουμε είναι πάρα πολλές. Παρακάτω θα παρουσιάσουμε μερικές από αυτές :

- Αναγνώριση φωνής και εκτέλεση διαφόρων εντολών και διαδικασιών μέσω αυτής.
- Εισαγωγή κατάλληλου Hardware και σύνδεση με εξωτερικό ενσύρματο ή κινητής τηλεφωνίας δίκτυο.
- Εισαγωγή διαφόρων προηγμένων εφαρμογών μέσω του AGI , όπως λήψη e-mail και ειδοποίηση στο κινητό μας τηλέφωνο , αναγνώριση κλήσης μέσω on-screen display(φωτογραφίες και διάφορες άλλες πληροφορίες) , αποστολή και λήψη προσωπικών αρχείων μέσω εσωτερικού δικτύου.
- Υπηρεσίες Billing μέσω προπληρωμένων υπηρεσιών ή ακόμα χρεώσεις πραγματικού χρόνου μέσω διαφόρων on-line υπηρεσιών (π.χ. PayPal)

Παράρτημα

Asterisk's Configuration Files

Sip.conf

```
[general]
port=5060
bindaddr=0.0.0.0
context=default
srvlookup=yes
videosupport=yes
```

```
[111]
type=friend
secret=123
callerid="sip-user"<KOSTAS>
quality=yes
nat=yes
host=dynamic
canreinvite=no
context=asterisk-pbx
```

```
[222]
type=friend
secret=123
callerid="sip-user"<Andrew>
quality=yes
nat=yes
host=dynamic
```

```
canreinvite=no
context=asterisk-pbx
```

```
[333]
```

```
type=friend
secret=123
callerid="Other"<333>
quality=yes
nat=yes
host=dynamic
canreinvite=no
context=asterisk-pbx
```

```
*****
```

Extensions.conf

```
*****
```

```
[general]
```

```
autofallthrough=yes
clearglobalvars=yes
;priorityjumping=yes
```

```
[asterisk-pbx]
```

```
exten => 100,1,Answer
exten => 100,4,Background(/) ; lipei to minima pou prepei na valoume
exten => 100,2,WaitExten(5)
exten => 100,3,Goto(2)
```

```
exten => 111,1,Noop()
exten => 111,2,Dial(SIP/111,20,tr)
exten => 111,n,Goto(111- $\{DIALSTATUS\}$ ,1)
exten => 111-NOANSWER,1,Voicemail(u111)
exten => 111-BUSY,1,Voicemail(b111)
exten => 111-ANSWER,1,Hangup()
exten => _111-.,1,Goto(111-NOANSWER,1)
```

```
exten => 222,1,Noop()
```

```

exten => 222,2,Dial(SIP/222,20,tr)
exten => 222,n,Goto(111- $\{\text{DIALSTATUS}\}$ ,1)
exten => 222-NOANSWER,1,Voicemail(u111)
exten => 222-BUSY,1,Voicemail(b111)
exten => 222-ANSWER,1,Hangup()
exten => _222-.,1,Goto(111-NOANSWER,1)

exten => 333,1,Noop()
exten => 333,2,Dial(SIP/333,20,tr)
exten => 333,n,Goto(111- $\{\text{DIALSTATUS}\}$ ,1)
exten => 333-NOANSWER,1,Voicemail(u111)
exten => 333-BUSY,1,Voicemail(b111)
exten => 333-ANSWER,1,Hangup()
exten => _333-.,1,Goto(111-NOANSWER,1)

exten => 500,1,GotoIf( $\{\text{CALLERID(number)}\} = 111\]?yes:no$ )
exten => 500,n(yes),VoicemailMain( $\{\text{CALLERID(number)}\}$ )
exten => 500,n(no),VoicemailMain()

exten => 865,1,Answer
exten => 865,2,Playback(latest-forecast)
exten => 865,3,System(/usr/bin/curl -s
http://www.wrh.noaa.gov/sgx/data/audio/LAXSAFC42.mp3 > /tmp/LAXSAFC42.mp3)
exten => 865,4,Wait(2)
exten => 865,5,MP3Player(/tmp/LAXSAFC42.mp3)
exten => 865,6,System(rm /tmp/LAXSAFC42.mp3 -f)
exten => 865,7,Hangup

exten => 965,1,Answer
exten => 965,n,Set(Retries= $\{\text{Retries}\}+1$ )
exten => 965,n,DateTime
exten => 965,n,GotoIf( $\{\text{Retries}\} <= 4\]965,3$ )
exten => 965,3,Hangup

exten => i,1,Goto(100,1)

```

Voicemail.conf

[general]

format=wav49|gsm|wav

maxmessage=300

minmessage=3

maxgreet=45

maxsilence=5

saycid=yes

[zonemessages]

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

[default]

111 => 123,Sip User,,,tz=Greece

222 => 456,SIP User,,,tz=Greece

333 => 789,SIP User,,,tz=Greece

Features.conf

[general]

transferdigittimeout => 3

xfersound = beep

xferfailsound = beeper

atxfernoanswertimeout = 20

[featuremap]

blindxfer => **

atxfer => ##

Βιβλιογραφία

- [1] Asterisk The Future Of Telephony. James Van Maggelen , Jared Smith , Leif Madsen. September 2005
- [2] Configuration Guide For Asterisk PBX , How to build and configured a PBX with open Source Software. Flavio E. Concalves. March 2007
- [3] Asterisk : A non Technical Review. Nasser K. Manesh. December 2004
- [4] Building Telephony Systems With Asterisk. David Gomillion, Barrie Dempster . September 2005
- [5] <http://www.voip-info.org/wiki-Asterisk>
- [6] <http://www.asteriskguru.com/>
- [7] VoIP Overview For Operators. Gene Lew. May 2005
- [8] VoIP For Business. Tim Frichtel, Scale 4X. February 2006