



Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή στα Πληροφοριακά Συστήματα

**Υπηρεσίες Επικοινωνίας και Συνεργασίας Πραγματικού
Χρόνου σε Υποδομές Υπολογιστικού Νέφους**

Μιχάλης Καλογήρου

**Επιβλέπων Καθηγητής
Αναστάσιος Νταγιούκλας**

Μάιος 2013



Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Υπηρεσίες Επικοινωνίας και Συνεργασίας Πραγματικού Χρόνου σε Υποδομές Υπολογιστικού Νέφους

Μιχάλης Καλογήρου

**Επιβλέπων Καθηγητής
Αναστάσιος Νταγιούκλας**

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση μεταπτυχιακού τίτλου
σπουδών
στα Πληροφοριακά Συστήματα

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Μάιος 2013

Περίληψη

Η καθολική επικράτηση του πρωτοκόλλου IP δημιούργησε μια νέα πραγματικότητα στις τηλεπικοινωνίες όπου όλες πλέον οι παραδοσιακές υπηρεσίες επικοινωνίας έχουν μετασχηματιστεί και μεταβεί σε δίκτυα IP, με χαρακτηριστικό παράδειγμα την τηλεφωνία μέσω των τεχνολογιών VoIP. Σήμερα βρισκόμαστε στην αρχή μιας άλλης, πολύ σημαντικής, τεχνολογικής αλλαγής, με πολύ μεγάλη δυναμική, αυτής των τεχνολογιών εικονοποίησης (virtualization) και των υπηρεσιών υπολογιστικού νέφους. Η πλειονότητα των επιχειρηματικών εφαρμογών, όλες οι web εφαρμογές καθώς και οι Βάσεις Δεδομένων, έχουν ήδη μεταφερθεί με επιτυχία στα εικονικά περιβάλλοντα, αποδεικνύοντας τα ωφέλημα της εικονοποίησης για εξοικονόμηση και εξορθολογισμό της διαχείρισης των τεχνολογιών πληροφορικής.

Παρ' όλη τη πρόοδο στο τομέα αυτό, η υιοθέτηση της εικονοποίησης από τις εφαρμογές πραγματικού χρόνου, όπως π.χ. VoIP, φαίνεται ότι κινείται με πιο αργούς ρυθμούς. Η παρούσα διατριβή παρουσιάζει, μέσω παραδείγματος, την εφικτότητα και την ωριμότητα της μετάβασης των εφαρμογών της οικογένειας υπηρεσιών πραγματικού χρόνου σε περιβάλλον υπολογιστικού νέφους. Αποδεικνύεται ότι σήμερα υπάρχει η δυνατότητα αξιόπιστης, ασφαλούς και αποδοτικής φιλοξενίας υπηρεσιών πραγματικού χρόνου σε εικονικές μηχανές (virtual machines), που προσφέρονται ως υπηρεσία (Infrastructure as a Service - IaaS) από τις δημόσιες υποδομές υπολογιστικού νέφους (public cloud).

Παράλληλα παρουσιάζει μια κλιμακώσιμη αρχιτεκτονική, χαμηλού κόστους, και υψηλής διαθεσιμότητας που αξιοποιεί πλήρως τα πλεονεκτήματα των υπηρεσιών υπολογιστικού νέφους και είναι κατάλληλη τόσο για οργανισμούς, μικρούς ή μεγάλους, όσο και για πάροχους υπηρεσιών με πολλές χιλιάδες χρήστες. Τα αποτελέσματα της παρούσας έρευνας μπορούν να αποτελέσουν ένα οδηγό ή μια βάση ενός πραγματικού επιχειρηματικού χρόνου που μοντέλου αξιοποιούν ανάπτυξης τα και παροχής πλεονεκτήματα των υπηρεσιών υπολογιστικού νέφους: μικρό αρχικό κόστος, κλιμάκωση σύμφωνα με τις ανάγκες, υψηλή διαθεσιμότητα.

Summary

The universal prevalence of IP protocol created a new reality in telecommunications where all most traditional communication services have been transformed and switched to IP networks. A typical example of such services are VoIP telephony technologies. Today we are at the beginning of another, very important technological change with a huge potential, the Virtualization technologies and cloud computing services. The majority of business applications, all web applications and databases, have already been successfully imported in virtual environments, proving the benefits of Virtualization to save and streamline the management of information technology.

Despite the progress in this area, the adoption of Virtualization of the real-time applications, such as VoIP, seems to be moving more slowly. This thesis presents, through an example, the feasibility and maturity of migration of the family of real-time services in cloud computing environment. It turns out that today there is the possibility of reliable, safe and efficient hosting of real-time services in virtual machines, that offered as a service (Infrastructure as a Service - IaaS) by public cloud computing infrastructure.

Also presenting a scalable architecture, low cost, and high-availability making full use of the advantages of cloud computing services and is suitable for both organizations, small or large, and for delivering services to many thousands of users. The results of this study can be a guide or a basis of a real-time business model leverage the development and provision of service advantages cloud computing services: small initial cost escalation as needed, high availability.

Ευχαριστίες

Ο θεσμός των ανοικτών πανεπιστημίων προσφέρει την δυνατότητα σε εργαζόμενους και άτομα μεγαλύτερης ηλικίας να αποκτήσουν γνώσεις και τυπικά προσόντα παράλληλα με τις επαγγελματικές και οικογενειακές υποχρεώσεις τους. Ευχαριστώ το Ανοικτό Πανεπιστήμιο Κύπρου που μου έδωσε αυτή την ευκαιρία και τον καθηγητή Αναστάσιο Νταγιούκλα, για την εμπιστοσύνη που έδειξε στο αντικείμενο της μεταπτυχιακής διατριβής μου.

Ευχαριστώ την οικογένεια μου, την σύζυγο μου Τζένη και τα παιδιά μου Νικόλα, Πρόδρομο και Ερατώ, για την κατανόηση και την υπομονή που έδειξαν όλο αυτό το διάστημα των μεταπτυχιακών μου σπουδών.

Ευχαριστώ το ΕΔΕΤ για την παραχώρηση της εικονικής μηχανής, πάνω στην οποία έγιναν όλες οι δοκιμές που περιγράφονται στο κείμενο αυτό.

Τέλος, ευχαριστώ το κ. Αντώνη Ψαρά, από την εταιρία Microbase, για την δωρεάν παραχώρηση λογαριασμού VoIP με πρόσβαση στο δημόσιο τηλεφωνικό δίκτυο και την τεχνική βοήθεια που μου πρόσφερε.

Περιεχόμενα

Περίληψη.....	5
Summary.....	6
Ευχαριστίες.....	7
Περιεχόμενα.....	9
Κεφάλαιο 1.....	11
Εισαγωγή.....	11
Περιεχόμενα κεφαλαίων.....	13
Κεφάλαιο 2.....	17
Υπολογιστικό νέφος - Cloud computing.....	17
Θέματα ασφάλειας υπηρεσιών νέφους.....	21
Συστάσεις και σημεία προσοχής.....	22
Κεφάλαιο 3.....	25
Παρουσίαση του SIP (Session Initiation Protocol).....	25
3.1 Λειτουργικές οντότητες του πρωτοκόλλου SIP (SIP Network Elements).....	27
SIP User Agent (UA).....	27
SIP Proxy Server.....	27
SIP Registrar Server.....	28
SIP Redirect Server.....	28
3.2 Μηνύματα και Μέθοδοι SIP.....	29
SIP Requests.....	30
SIP Responses.....	32
3.3 Δομή του Πρωτοκόλλου SIP.....	33
SIP Transactions.....	34
SIP Dialogs.....	37
Επιλογή των Παραμέτρων της Κλήσης.....	38
3.4 Το Πρωτόκολλο SDP.....	38
3.5 Επιλογή των Παραμέτρων Κλήσης μέσω του μοντέλου Offer/Answer.....	40
3.6 Πραγματοποίηση Κλήσεων με το SIP.....	44
Καταχώρηση / Registration.....	44
Επιτυχημένη εγκατάσταση κλήσης.....	45
Κεφάλαιο 4.....	47
Σχεδιασμός ολοκληρωμένου συστήματος επικοινωνίας.....	47
4.1 Παρουσίαση τεχνολογιών αιχμής.....	48
Βασικό σύστημα διαχείρισης κλήσεων.....	48
Media Services.....	51
4.2 Επιλογές κλιμάκωσης και εφεδρείας.....	52
Κεντρικό υποσύστημα μνήμης (memory caching).....	55

Βάσεις Δεδομένων.....	56
4.3 Αρχιτεκτονική συστήματος.....	57
4.4 Θέματα ασφάλειας πιστοποίησης.....	61
Κεφάλαιο 5.....	64
Θέματα απόδοσης και ποιότητας.....	64
5.1 Θέματα εικονοποίησης και εφαρμογών πραγματικού χρόνου.....	65
5.2 Θέματα ποιότητας στην τηλεφωνία με πακέτα	66
Ποιότητα φωνής και jitter (τρέμουλο).....	67
Πηγές του jitter.....	68
Χρονισμός εφαρμογών μέσω.....	69
5.2 Ποιότητα Υπηρεσίας.....	72
5.3 Ποιότητα Εμπειρίας - Quality of Experience (QoE)	74
5.4 Service Level Agreement (SLA)	76
Κεφάλαιο 6.....	78
Δοκιμές - Μετρήσεις.....	78
Υπερφόρτωση λειτουργίας καταχώρησης (registration).....	79
Αποτελέσματα και συμπεράσματα δοκιμών καταχώρησης.....	82
Υπερφόρτωση λειτουργίας εγκαθίδρυσης κλήσεων (Invite Overload).....	83
Αποτελέσματα και συμπεράσματα δοκιμών εγκαθίδρυσης κλήσεων.....	85
Υπερφόρτωση υπηρεσιών (media services).....	85
Αποτελέσματα και συμπεράσματα δοκιμών υπηρεσιών.....	88
Κεφάλαιο 7.....	93
Επίλογος.....	93
Βιβλιογραφία.....	97
Παράρτημα Α.....	98
Α.1 Οδηγίες εγκατάστασης συστήματος.....	99
Α.2 Εγκατάσταση και προσαρμογή SIPP.....	103
Α.3 Οθόνες δοκιμών υπερφόρτωσης.....	104
Α.3.1 Οθόνες δοκιμών “Λειτουργία καταχώρησης (registration)”.....	105
Α.3.2 Οθόνες δοκιμών “Πραγματοποίηση κλήσεων (invite)”.....	113
Α.3.3 Οθόνες δοκιμών “Προηγμένες Υπηρεσίες (media services)”.....	117

Κεφάλαιο 1

Εισαγωγή

Η καθολική επικράτηση του πρωτοκόλλου IP δημιούργησε μια νέα πραγματικότητα στις τηλεπικοινωνίες όπου όλες πλέον οι παραδοσιακές υπηρεσίες επικοινωνίας έχουν μετασηματιστεί και μεταβεί σε δίκτυα IP, με χαρακτηριστικό παράδειγμα την τηλεφωνία μέσω των τεχνολογιών VoIP.

Η αρχή έγινε στην επικοινωνία μεταξύ των δικτύων φωνής των παρόχων, κυρίως για να εκμεταλλευτούν την ανάπτυξη των δικτύων δεδομένων, μειώνοντας έτσι το κόστος από την παράλληλη λειτουργία ξεχωριστών δικτύων φωνής. Στη συνέχεια είχαμε την τεχνολογία VoIP να περνά σε μεγάλους οργανισμούς με τις VoIP ζεύξεις των παραδοσιακών τηλεφωνικών κέντρων. Σήμερα είμαστε πια στην εποχή της IP Τηλεφωνίας όπου τα παραδοσιακά τηλεφωνικά κέντρα έχουν πλέον αντικατασταθεί από υπολογιστικά συστήματα που διαχειρίζονται της VoIP κλήσεις. Για την ανάπτυξη των υπηρεσιών IP Τηλεφωνίας, παρά την αρχική επικράτηση του πρωτοκόλλου H.323 και την χρήση άλλων μη πρότυπων πρωτοκόλλων, έχουμε πλέον την καθολική εδραίωση του πρωτοκόλλου SIP.

Σήμερα βρισκόμαστε στην αρχή μιας άλλης, πολύ σημαντικής, τεχνολογικής αλλαγής, με πολύ μεγάλη δυναμική, αυτής των τεχνολογιών εικονοποίησης (virtualization) και των υπηρεσιών υπολογιστικού νέφους. Η πλειονότητα των επιχειρηματικών εφαρμογών, όλες οι web εφαρμογές καθώς και οι Βάσεις Δεδομένων, έχουν ήδη μεταφερθεί με επιτυχία στα εικονικά περιβάλλοντα, αποδεικνύοντας τα ωφέληματα της εικονοποίησης για εξοικονόμηση και εξορθολογισμό της διαχείρισης των τεχνολογιών πληροφορικής.

Παρ' όλη τη πρόοδο στο τομέα αυτό, η υιοθέτηση της εικονοποίησης από τις εφαρμογές πραγματικού χρόνου φαίνεται ότι κινείται με πιο αργούς ρυθμούς. Οι εφαρμογές αυτού του είδους, όπως π.χ. VoIP, χαρακτηρίζονται από μια ιδιαίτερη ευαισθησία στην καθυστέρηση, απαιτώντας τα πακέτα να παραδίδονται σε τακτά χρονικά διαστήματα με πολύ μικρή χρονική απόκλιση, διαφορετικά μια ακανόνιστη ροή οδηγεί σε απόρριψη πακέτων και επομένως σε κακή ποιότητα υπηρεσίας.

Η παρούσα διατριβή παρουσιάζει, μέσω παραδείγματος, την εφικτότητα και την ωριμότητα της μετάβασης των εφαρμογών της οικογένειας υπηρεσιών πραγματικού χρόνου σε περιβάλλον υπολογιστικού νέφους. Αποδεικνύεται ότι σήμερα υπάρχει η δυνατότητα αξιόπιστης, ασφαλούς και αποδοτικής φιλοξενίας υπηρεσιών

πραγματικού χρόνου σε εικονικές μηχανές (virtual machines), που προσφέρονται σήμερα ως υπηρεσία (Infrastructure as a Service - IaaS) από τις δημόσιες υποδομές υπολογιστικού νέφους (public cloud). Παράλληλα παρουσιάζει μια κλιμακώσιμη αρχιτεκτονική, χαμηλού κόστους, και υψηλής διαθεσιμότητας που αξιοποιεί πλήρως τα πλεονεκτήματα των υπηρεσιών υπολογιστικού νέφους και είναι κατάλληλη τόσο για οργανισμούς, μικρούς ή μεγάλους, όσο και για πάροχους υπηρεσιών με πολλές χιλιάδες χρήστες.

Τα αποτελέσματα της παρούσας έρευνας μπορούν να αποτελέσουν ένα οδηγό ή μια βάση ενός επιχειρηματικού μοντέλου ανάπτυξης και παροχής υπηρεσιών πραγματικού χρόνου που αξιοποιούν τα πλεονεκτήματα των υπηρεσιών υπολογιστικού νέφους: μικρό αρχικό κόστος, κλιμάκωση σύμφωνα με τις ανάγκες, υψηλή διαθεσιμότητα.

Περιεχόμενα κεφαλαίων

Ακολουθεί μια σύντομη παρουσίαση των περιεχομένων των κεφαλαίων που ακολουθούν.

Στο **Κεφάλαιο 2** παρουσιάζονται, αρχικά, τα βασικά χαρακτηριστικά και τα μοντέλα ανάπτυξης και παροχής των υπηρεσιών Υπολογιστικού Νέφους. Στη συνέχεια σημειώνονται κάποια θέματα ασφάλειας και ιδιωτικότητας που χρειάζονται ιδιαίτερη προσοχή κατά την χρήση των υπηρεσιών νέφους, τα οποία πρέπει να λαμβάνονται υπόψη, αλλά να μην αποτρέπουν την υιοθέτηση και αξιοποίηση αυτών των υπηρεσιών. Τέλος δίνονται κάποιες συστάσεις και σημεία προσοχής που αφορούν τους κινδύνους καθώς και νομικά ζητήματα που πρέπει να ορίζονται σε μια σύμβαση μεταξύ του παρόχου των υπηρεσιών νέφους και του πελάτη.

Στο **Κεφάλαιο 3** παρουσιάζουμε με μια μεγαλύτερη λεπτομέρεια το πρωτόκολλο SIP (Session Initiation Protocol), το οποίο αποτελεί την καρδιά του συστήματος μας. Αρχικά παρουσιάζουμε τις τέσσερις λειτουργικές οντότητες του πρωτοκόλλου SIP (SIP Network Elements) : user agent, proxy server, registrar server και redirect server. Στη συνέχεια, με συντομία τα είδη των SIP μηνυμάτων, τη σύνταξή τους καθώς και τα πεδία (header fields) που τα αποτελούν καθώς και η δομή του πρωτοκόλλου SIP . Ακολουθεί μια σύντομη ανασκόπηση του Πρωτοκόλλου SDP (Session Description

Protocol), που χρησιμοποιεί το SIP για την διαπραγμάτευση των παραμέτρων μιας κλήσης, καθώς και του τρόπου επιλογής των παραμέτρων κλήσης μέσω του μοντέλου Offer/Answer. Τέλος παρουσιάζεται ο τρόπος καταχώρησης και πραγματοποίησης κλήσεων με το πρωτόκολλο SIP, συνοδευόμενος από κατάλληλα διαγράμματα ροής των SIP μηνυμάτων.

Στο **4ο Κεφάλαιο** παρουσιάζεται ο σχεδιασμός και ο τρόπος ανάπτυξης του ολοκληρωμένου συστήματος επικοινωνίας πραγματικού χρόνου με την αξιοποίηση πολλών διαφορετικών υποσυστημάτων, τα οποία καλύπτουν διαφορετικές λειτουργίες του συνολικού συστήματος. Το κεφάλαιο ξεκινά με μια σύντομη παρουσίαση των διαθέσιμων τεχνολογιών αιχμής (State-of-the-Art) και κάνει μια επιλογή επιμέρους εφαρμογών ανοικτού λογισμικού που μαζί μπορούν να συγκροτήσουν μια σύγχρονη υπηρεσία επικοινωνίας και συνεργασίας, χωρίς περιορισμούς κλιμάκωσης, με τον μέγιστο βαθμό διαλειτουργικότητας και ομοιογένειας της συνολικής λύσης, καθώς και ευκολία στην καθημερινή λειτουργία και την παρακολούθησή της. Στη συνέχεια δείχνει πως ο σχεδιασμός, που βασίζεται σε εικονικές υποδομές, προσφέρει υψηλή διαθεσιμότητα (μέσω εφεδρειών) και η δυνατότητα γρήγορης και εύκολης κλιμάκωσης, ώστε να εκμεταλλεύεται σωστά τη λογική του pay-as-you-go των υπηρεσιών υπολογιστικού νέφους. Παρουσιάζεται ο τρόπος υλοποίησης μια υποδομής πολλαπλών συστημάτων SIP Proxy και πως αναπτύσσεται ένα σχήμα διαχείρισης κοινόχρηστων πληροφοριών μνήμης, καθώς και πως αντιμετωπίζουμε το ζήτημα της διαθεσιμότητας της Βάσης Δεδομένων. Στόχος είναι να δείξουμε ότι στην τελική αρχιτεκτονική του συστήματος μπορούμε να προσθέτουμε εικονικές μηχανές ώστε να σταδιακά να καλύπτουμε τις απαιτήσεις όποτε αυτές δημιουργούνται, αυξάνοντας ταυτόχρονα την απόδοση του συστήματος. Τέλος δείχνουμε πως αποφεύγεται η χρήση clear text password στη κανονική λειτουργία πιστοποίησης των χρηστών, άρα η μεταφορά τέτοιας πληροφορίας στο δίκτυο, με το μοντέλο πιστοποίησης (authentication) που χρησιμοποιεί το SIP και βασίζεται στο HTTP digest authentication.

Στο **Κεφάλαιο 5** συζητάμε πως η εικονοποίηση επηρεάζει τις εφαρμογές πραγματικού χρόνου. Παρόλο που η πλειονότητα των εφαρμογών έχουν μεταφερθεί με επιτυχία στα εικονικά περιβάλλοντα η υιοθέτηση της εικονοποίησης στην περιοχή εφαρμογών πραγματικού χρόνου φαίνεται ότι κινείται με πιο αργούς ρυθμούς. Για να κατανοήσουμε τους λόγους αυτής της καθυστέρησης μελετάμε τα θέματα

ποιότητας στην τηλεφωνία με πακέτα και την σχέση της ποιότητα και του jitter (τρέμουλο). Συνεχίζοντας την συζήτηση γύρω από τα θέματα ποιότητας, εξετάζουμε πως ο χρονισμός μιας εικονικής μηχανής μπορεί να επηρεάσει την λειτουργία εφαρμογών πραγματικού χρόνου, χρησιμοποιούμε ως παράδειγμα την πολύ γνωστή και διαδεδομένη εφαρμογή Asterisk. Ακολούθως παρουσιάζουμε τη σημερινή πραγματικότητα που δείχνει ότι η Ποιότητα Υπηρεσίας (QoS) γενικά δεν προσφέρεται στους πελάτες από τους παρόχους και ότι οι τεχνικές QoS περιορίζονται εντός των εσωτερικών των δικτύων για μη εμπορικές υπηρεσίες ή λειτουργίες. Αυτή η αδυναμία έχει τροφοδοτήσει τα τελευταία χρόνια μια διαφορετική συζήτηση γύρω από την ποιότητα μιας υπηρεσίας, την αντίληψη του χρήστη - πελάτη για το πόσο καλή είναι μια υπηρεσία και όχι πόσο καλά υποστηρίζεται από ένα δίκτυο, αυτό που καλείται Ποιότητα Εμπειρίας - Quality of Experience (QoE). Κλείνοντας αυτό το κεφάλαιο θα δούμε τι προβλέπεται σήμερα σε επίπεδο συμφωνίας υπηρεσιών - Service Level Agreement (SLA) - για το επίπεδο των παρεχόμενων υπηρεσιών νέφους και των ποινών σε περιπτώσεις παραβίασης των συμφωνηθέντων, καθώς και ένα σύνολο εξαιρέσεων, που τίθενται από τον πάροχο των υπηρεσιών.

Στο **Κεφάλαιο 6** παρουσιάζουμε της δοκιμές και τα αποτελέσματα υπερφόρτωσης των ορίων του συστήματος, αλλά και τον υποσυστημάτων του, στα ακόλουθα τρία επίπεδα βασικών λειτουργιών: καταχώρησης (registration), πραγματοποίησης κλήσεων (invite) και χρήσης προηγμένων υπηρεσιών (media services). Για την πραγματοποίηση των δοκιμών υπερφόρτωσης έγινε χρήση του ανοικτού λογισμικού sipp (<http://sipp.sourceforge.net/>), το οποίο προσομοιώνει έναν SIP User Agent (UA) και μέσω αρχείων xml επιτρέπει την δημιουργία διάφορων σεναρίων για την υπερφόρτωση ενός συστήματος. Ενδεικτικά αναφέρουμε ότι το συγκεκριμένο σύστημα, σε μια εικονική μηχανή με μια vCPU με ένα core κα, 2 GB RAM, μπορούσε να ανταποκριθεί μέχρι 3.000 καταχωρήσεις ανά δευτερόλεπτο, μέχρι 900 κλήσεις ανά δευτερόλεπτο και να υποστηρίξει 40 ταυτόχρονες ροές ζωντανού ραδιοφώνου.

Στο **6ο** και τελευταίο **κεφάλαιο** δηλαδή στον επίλογο, επιλέξαμε να συνοψίσουμε την απάντηση στο βασικότερο ερώτημα της παρούσας διατριβής, κατά πόσο οι σημερινές υπηρεσίες εικονικών μηχανών, που προσφέρονται ως υπηρεσία από τα δημόσια υπολογιστικά νέφη (public cloud IaaS), μπορούν να χρησιμοποιηθούν για την φιλοξενία υπηρεσιών πραγματικού χρόνου. Ο βιαστικός αναγνώστης μπορεί να ανατρέξει κατευθείαν στο κεφάλαιο αυτό για την απάντηση.

Τέλος στο **Παράρτημα Α**, θα βρείτε μια σύνοψη της εγκατάστασης των βασικών εφαρμογών του ολοκληρωμένου συστήματος, με σκοπό ο αναγνώστης να σχηματίσει μια εικόνα των απαιτούμενων υποσυστημάτων και όχι ένα οδηγό εγκατάστασης. Επίσης το παράρτημα περιλαμβάνει ένα οδηγό εγκατάστασης και προσαρμογής της εφαρμογής SIPP που χρησιμοποιήθηκε στις δοκιμές. Τέλος έχουν συμπεριληφθεί αρκετές εικόνες από τις οθόνες των δοκιμών που περιέχουν λεπτομέρειες που δεν έχουν αναφερθεί στο σχετικό κεφάλαιο παρουσίασης των δοκιμών και των αποτελεσμάτων τους και μπορούν να φανούν χρήσιμες σε μια προσεκτικότερη ανάλυση.

Κεφάλαιο 2

Υπολογιστικό νέφος - Cloud computing

Στο κεφάλαιο αυτό θα παρουσιάσουμε τα βασικά χαρακτηριστικά και τα μοντέλα ανάπτυξης και παροχής των υπηρεσιών Υπολογιστικού Νέφους. Καθώς επίσης κάποια θέματα ασφάλειας και ιδιωτικότητας και κάποιες συστάσεις και ζητήματα προσοχής, πριν από την υιοθέτηση των υπηρεσιών αυτών.

Καθώς υπάρχουν σήμερα αρκετές διαφορετικές ερμηνείες του όρου “Cloud computing”, επέλεξα να παρουσιάσω μια περίληψη του ορισμού που δίνει το αμερικανικό ινστιτούτο προτύπων και τεχνολογίας (NIST) [1], που θεωρώ ως τον πλέον ολοκληρωμένο.

Το cloud computing (υπολογιστικό νέφος) είναι ένα μοντέλο για την ενεργοποίηση δικτυακής πρόσβασης, πανταχόθεν, εύκολα και κατ' απαίτηση, σε μια κοινόχρηστη συγκέντρωση ρυθμιζόμενων υπολογιστικών πόρων (π.χ. δίκτυα, εξυπηρετητές, αποθηκευτικά μέσα, εφαρμογές και υπηρεσίες) που μπορεί να τροφοδοτηθεί γρήγορα και να διατεθεί με ελάχιστη προσπάθεια διαχείρισης ή αλληλεπίδρασης ενός πάροχου υπηρεσιών.

Αυτό το μοντέλο αποτελείται από πέντε βασικά χαρακτηριστικά, τρία μοντέλα παροχής υπηρεσιών και τέσσερα μοντέλα ανάπτυξης.

Τα πέντε βασικά χαρακτηριστικά είναι:

1. **Κατ' απαίτηση αυτοεξυπηρέτηση (on-demand self-service).** Ο καταναλωτής μπορεί μονομερώς να κάνει χρήση των υπολογιστικών δυνατοτήτων, όπως π.χ. η ώρα χρήσης του εξυπηρετητή και της δικτυακής αποθήκευσης, όπως αυτός επιθυμεί, αυτόματα, χωρίς απαίτηση ανθρώπινης αλληλεπίδρασης από το φορέα παροχής υπηρεσιών.
2. **Ευρεία δικτυακή πρόσβαση (broad network access).** Οι δυνατότητες είναι διαθέσιμες μέσω του δικτύου και προσβάσιμες μέσω τυποποιημένων μηχανισμών που προωθούν την χρήση από ετερογενείς πλατφόρμες πελάτη (π.χ., κινητά τηλέφωνα, ταμπλέτες, φορητούς υπολογιστές, και σταθμούς εργασίας).
3. **Συγκέντρωση πόρων (resource pooling).** Οι υπολογιστικοί πόροι του παρόχου συγκεντρώνονται για να εξυπηρετήσουν πολλαπλούς καταναλωτές

χρησιμοποιώντας ένα μοντέλο πολυ-ενοικιαστή, με διαφορετικούς φυσικούς και εικονικούς πόρους που δυναμικά ανατίθενται και ξανα-ανατίθενται ανάλογα με τη ζήτηση των καταναλωτών. Υπάρχει μια αίσθηση ανεξαρτησίας της θέσης, υπό την έννοια ότι ο πελάτης γενικά δεν έχει κανέναν έλεγχο ή γνώση σχετικά με την ακριβή τοποθεσία των παρεχόμενων πόρων, αλλά μπορεί να είναι σε θέση να προσδιορίζει τη θέση σε ένα υψηλότερο επίπεδο αφαίρεσης (π.χ. χώρα, κράτος, ή κέντρο δεδομένων - datacenter). Παραδείγματα των πόρων περιλαμβάνουν την αποθήκευση, επεξεργασία, τη μνήμη, και το εύρος ζώνης του δικτύου.

4. **Ταχεία ελαστικότητα (rapid elasticity)**. Οι δυνατότητες μπορούν να τροφοδοτηθεί ελαστικά και κυκλοφορήσουν, σε ορισμένες περιπτώσεις, αυτόματα, να κλιμακωθούν γρήγορα προς τα έξω και προς τα μέσα ανάλογα με τη ζήτηση. Για τον καταναλωτή, οι δυνατότητες για παροχή συχνά φαίνεται να είναι απεριόριστες και μπορούν να θα διατεθούν σε οποιαδήποτε ποσότητα, ανά πάσα στιγμή.
5. **Μετρούμενη υπηρεσία (measured service)**. Τα συστήματα νέφους ελέγχουν και βελτιστοποιούν αυτόματα τη χρήση των πόρων αξιοποιώντας δυνατότητες μέτρησης σε κάποιο επίπεδο αφαίρεσης ανάλογα με το είδος της υπηρεσίας (π.χ. αποθήκευσης, επεξεργασίας, το εύρος ζώνης, ενεργούς λογαριασμούς χρηστών). Η χρήση των πόρων μπορεί να παρακολουθείται, να ελέγχεται και να εκδίδονται αναφορές χρήσης, παρέχοντας διαφάνεια τόσο στον πάροχο όσο στους καταναλωτές των υπηρεσιών.

Αφού είδαμε τα πέντε βασικά χαρακτηριστικά του μοντέλου, ας δούμε τώρα τα τρία, πιο βασικά, μοντέλα παροχής υπηρεσιών:

1. **Λογισμικό ως Υπηρεσία (Software as a Service - SaaS)**. Η δυνατότητα που παρέχεται στον καταναλωτή να χρησιμοποιήσει τις εφαρμογές του παρόχου που τρέχουν σε υποδομές νέφους. Οι εφαρμογές είναι προσβάσιμες από διάφορες συσκευές πελάτη μέσω της διεπαφής ενός web browser (π.χ. web-based e-mail) ή ενός προγράμματος. Ο καταναλωτής δεν διαχειρίζεται ή ελέγχει τη βασική υποδομή νέφους, συμπεριλαμβανομένου του δικτύου, του εξυπηρετητή, του λειτουργικού συστήματος, της αποθήκευσης ή ακόμη και

μεμονωμένων δυνατοτήτων της εφαρμογής, με πιθανή εξαίρεση κάποιων περιορισμένων ρυθμίσεων της εφαρμογής σε επίπεδο χρήστη.

2. **Πλατφόρμα ως Υπηρεσία (Platform as a Service - PaaS)**. Η δυνατότητα που παρέχεται στον καταναλωτή να αναπτύξει πάνω στην υποδομή νέφους δικές του εφαρμογές με τη χρήση γλωσσών προγραμματισμού, βιβλιοθηκών, υπηρεσιών και εργαλείων που υποστηρίζονται από τον πάροχο. Ο καταναλωτής δεν διαχειρίζεται ή ελέγχει τη βασική υποδομή νέφους, συμπεριλαμβανομένου του δικτύου, του εξυπηρετητή, του λειτουργικού συστήματος ή της αποθήκευσης, αλλά έχει τον έλεγχο των εφαρμογών που αναπτύσσει και, ενδεχομένως, των ρυθμίσεων του περιβάλλοντος φιλοξενίας της.

3. **Υποδομή ως Υπηρεσία (Infrastructure as a Service - IAAS)**. Η δυνατότητα που παρέχεται στον καταναλωτή, με την παροχή επεξεργαστικής ισχύς, αποθήκευσης, δικτύου και άλλων θεμελιωδών υπολογιστικών πόρων, για να αναπτύξει και να εκτελέσει οποιοδήποτε λογισμικό της επιλογής του, το οποίο μπορεί να περιλαμβάνει λειτουργικά συστήματα και εφαρμογές. Ο καταναλωτής δεν διαχειρίζεται ή ελέγχει τη βασική υποδομή νέφους αλλά έχει τον έλεγχο των λειτουργικών συστημάτων, της αποθήκευσης και των εφαρμογών, και ενδεχομένως περιορισμένο έλεγχο επιλογής των συστατικών δικτύωσης (π.χ. φιλοξενία firewall).

Τέλος, παρουσιάζουμε τα τέσσερα μοντέλα ανάπτυξης:

1. **Ιδιωτικό νέφος (private cloud)**. Η υποδομή νέφους παρέχεται για αποκλειστική χρήση από ένα μόνο φορέα, ο οποίος αποτελείται από πολλούς καταναλωτές (π.χ. επιχειρηματικές μονάδες). Μπορεί να ανήκει, να διαχειρίζεται και να λειτουργείται από τον ίδιο τον οργανισμό, κάποιο τρίτο ή κάποιο συνδυασμό τους και να βρίσκεται εντός ή εκτός των εγκαταστάσεων του οργανισμού.

2. **Νέφος κοινότητας (community cloud)**. Η υποδομή νέφους παρέχεται για αποκλειστική χρήση από μια συγκεκριμένη κοινότητα καταναλωτών από οργανισμούς που έχουν κοινές ανησυχίες (π.χ. αποστολή, απαιτήσεις

ασφάλειας, πολιτική, ζητήματα συμμόρφωσης). Μπορεί να ανήκει, να διαχειρίζεται και να λειτουργείται από ένα ή περισσότερους από τους οργανισμούς της κοινότητας, από τρίτους ή κάποιο συνδυασμό τους και να βρίσκεται εντός ή εκτός των εγκαταστάσεων των οργανισμών.

3. Δημόσιο

4. **νέφος (public cloud)**. Η υποδομή νέφους παρέχεται για ανοικτή χρήση από το ευρύ κοινό. Μπορεί να ανήκει, να διαχειρίζεται και να λειτουργείται από μια επιχείρηση, ένα ακαδημαϊκό ή κυβερνητικό οργανισμό ή κάποιο συνδυασμό αυτών. Φιλοξενείται στις εγκαταστάσεις του παρόχου - διαχειριστή.
5. **Υβριδικό νέφος (hybrid cloud)**. Η υποδομή νέφους είναι μια σύνθεση από δύο ή περισσότερες διακριτές υποδομές νέφους (ιδιωτική, κοινότητας ή δημόσια) που ενώ παραμένουν μοναδικές οντότητες συνδέονται μεταξύ τους μια τυποποιημένη ή μη τεχνολογία που επιτρέπει την μεταφορά δεδομένων και εφαρμογών (π.χ. για εξισορρόπηση φορτίου μεταξύ των νεφών).

Θέματα ασφάλειας υπηρεσιών νέφους

Η χρήση των υπηρεσιών υπολογιστικού νέφους χρειάζεται ιδιαίτερη προσοχή στα θέματα ασφάλειας, τα οποία πρέπει να λαμβάνονται υπόψη, αλλά να μην αποτρέπουν την υιοθέτηση και αξιοποίηση τους. Δύο είναι τα κυριότερα ζητήματα ασφάλειας και ιδιωτικότητας των υπηρεσιών υπολογιστικού νέφους:

- η απώλεια ελέγχου των δεδομένων,
- η εξάρτηση από τον πάροχο υπολογιστικού νέφους.

Σχετικά με την **απώλεια ελέγχου δεδομένων** οι πελάτες / χρήστες των υπηρεσιών νέφους θα πρέπει να γνωρίζουν και να λαμβάνουν υπόψη τους θέματα όπως τα ακόλουθα:

- Οι πελάτες πρέπει να είναι υποψιασμένοι για τον κίνδυνο που υπάρχει αποθηκεύοντας τα δεδομένα τους στις υποδομές ενός παρόχου. Τα δεδομένα

μπορούν να είναι προσβάσιμα από τον ίδιο τον πάροχο ή ανταγωνιστές του σε περίπτωση κενών ασφάλειας των υποδομών του.

- Επίσης, οι πελάτες πρέπει να γνωρίζουν ότι κάποιοι πάροχοι συνηθίζουν να εφαρμόζουν τεχνικές εξόρυξης (data mining) και ανάλυσης των δεδομένων των χρηστών τους με σκοπό να βελτιώσουν τις υπηρεσίες τους, να προβλέψουν την συμπεριφορά των χρηστών ή να μειώσουν το λειτουργικό τους κόστος.
- Καθώς οι υποδομή, οι εφαρμογές και τα δεδομένα είναι μια υπηρεσία προσβάσιμη μέσω δικτύου, η διασύνδεση μεταξύ του χρήστη και του πάροχου είναι ευάλωτη σε επιθέσεις και επιρρεπής σε υποκλοπές, κυρίως στο σημείο σύνδεσης της δικτυακής συσκευής στο δίκτυο (π.χ. smartphone σε ένα wifi cafe).
- Τυχών απαίτηση για οριστική διαγραφή δεδομένων είναι συνήθως δύσκολη έως αδύνατη υπόθεση, ιδιαίτερα όταν ενσωματώνονται υπηρεσίες λήψης αντιγράφων ασφαλείας.
- Τα δικαιώματα πρόσβασης στα δεδομένα και η νομοθεσία προστασίας των προσωπικών δεδομένων διαφέρουν σημαντικά μεταξύ των χωρών. Η χώρα φιλοξενίας των υποδομών υπολογιστικού νέφους, άρα και η σχετική νομοθεσία, πρέπει να εξετάζονται και να λαμβάνονται υπόψη στην επιλογή του παρόχου των υπηρεσιών.

Σχετικά με την **εξάρτηση από τον πάροχο**, ίσως η πιο κρίσιμη ανησυχία πρέπει να είναι η ευρωστία και η διαθεσιμότητα του, κυρίως ο κίνδυνος για μια ξαφνική καταστροφή ή πτώχευση του που μπορεί να οδηγήσει σε οριστική απώλεια των εφαρμογών και των δεδομένων των πελατών.

Συστάσεις και σημεία προσοχής

Η γνώση και διαχείριση κινδύνου καθώς και τα (νομικά) ζητήματα συμμόρφωσης πρέπει να είναι καλά ορισμένα στη σύμβαση μεταξύ του παρόχου των υπηρεσιών νέφους και των πελατών. Η σύμβαση θα πρέπει να επιτρέπει τη διαφάνεια όσον

αφορά την επεξεργασία και την αποθήκευση των δεδομένων, π.χ. να είναι γνωστή η φυσική τοποθεσία αποθήκευσης των δεδομένων. Με αυτόν τον τρόπο ενισχύεται η εμπιστοσύνη μεταξύ του παρόχου και του πελάτη.

Η παρεχόμενη υπηρεσία θα πρέπει να είναι συμβατή με τους κανονισμούς και τη νομοθεσία που έχει ανάγκη να ακολουθήσει ο πελάτης. Τα προβλήματα και οι κίνδυνοι που επηρεάζουν τους κανόνες προστασίας των δεδομένων στην Ευρώπη θα πρέπει να λαμβάνονται δεόντως υπόψη όταν οι πλατφόρμες υπολογιστικού νέφους βρίσκονται σε υποδομές εκτός ευρωπαϊκών χωρών.

Η επικοινωνία μεταξύ του παρόχου και του πελάτη πρέπει να προστατεύεται επαρκώς για να διασφαλιστεί η εμπιστευτικότητα, η ακεραιότητα, ο έλεγχος ταυτότητας και να ελαχιστοποιηθεί ο κίνδυνος επιθέσεων. Ένας ξεκάθαρος και σαφής προσδιορισμός των μέτρων που ελήφθησαν για να εξασφαλιστεί η ασφάλεια της γραμμής επικοινωνίας θα πρέπει να είναι υποχρεωτικός και θα πρέπει να βασίζεται στην ανοικτή και διαφανή πρότυπα και τεχνολογίες.

Οι πάροχοι θα πρέπει να υποχρεούνται να διασφαλίζουν το απόρρητο των δεδομένων. Η υποχρεωτική διαγραφή των δεδομένων, με αίτημα του πελάτη, θα πρέπει να συμπεριληφθεί στη συμφωνία των υπηρεσιών. Το γεγονός ότι **δεν υπάρχει** εγγυημένη πλήρης διαγραφή των δεδομένων πρέπει να ληφθεί υπόψη, όταν τα δεδομένα συλλέγονται και αποθηκεύονται στις υποδομές του παρόχου. Προκειμένου να εξασφαλιστεί η διαθεσιμότητα των δεδομένων, οι πελάτες θα πρέπει να εξετάζουν την δημιουργία τοπικών αντιγράφων ασφαλείας (δηλ. στις δικές τους εγκαταστάσεις).

Το τηλεπικοινωνιακό δίκτυο που υποστηρίζει τις υπηρεσίες υπολογιστικού νέφους θα πρέπει να ασφαλίζεται και να προστατεύεται από κακόβουλο λογισμικό και επιθέσεις DOS (denial-of-service - άρνησης υπηρεσίας).

Πρέπει να παρέχονται επαρκής καταγραφή και έλεγχος συμβάντων. Ένας εξωτερικός έλεγχος μπορεί να είναι επωφελής για τη φήμη των παρόχων καθώς και για την ενίσχυση της εμπιστοσύνης με τον πελάτη.

Οι συνήθεις χρήστες (μη επαγγελματίες) θα πρέπει να εκπαιδευτούν σε σχέση με το νέο πρότυπο. Η εκπαίδευση θα τους προετοιμάσει να πάρουν τις σωστές αποφάσεις σχετικά με τη χρήση των υπηρεσιών υπολογιστικού νέφους, συμπεριλαμβανομένων των πληροφοριών που θα πρέπει να μεταφερθούν στο νέφος και υπό ποιες συνθήκες. Οι επαγγελματίες θα πρέπει να εξειδικευθούν για τη διαχείριση των νέων μορφών κινδύνων.

Κεφάλαιο 3

Παρουσίαση του SIP

(Session Initiation Protocol)

Το SIP (Session Initiation Protocol) είναι ένα πρωτόκολλο σηματοδότησης της IETF με σκοπό τον έλεγχο των κλήσεων για την μεταφορά φωνής και βίντεο πάνω από το Διαδίκτυο (Internet Protocol). Η τελευταία έκδοση είναι η RFC 3261. Σημειώστε ότι το SIP δεν είναι πρότυπο (standard) αλλά RFC. Το SIP αναλαμβάνει την εγκαθίδρυση, την διαχείριση και τον τερματισμό συνόδων μεταξύ δύο ή περισσότερων χρηστών. Εκτός από επικοινωνία φωνής και βίντεο, το SIP υποστηρίζει και άλλες εφαρμογές, όπως Instance Messaging, Presence, conference, μεταφορά αρχείων.

Το SIP είναι ένα πρωτόκολλο του επιπέδου εφαρμογής (OSI model: application layer) και είναι ανεξάρτητο από το πρωτόκολλο του επιπέδου μεταφοράς και μπορεί να λειτουργεί σε TCP ή UDP. Είναι ένα text-based πρωτόκολλο, όμοιο του HTTP και του SMTP, και η λειτουργία του βασίζεται στο μοντέλο πελάτη - εξυπηρετητή (client-server model).

Ειδικότερα οι βασικές λειτουργίες του πρωτοκόλλου SIP είναι:

- User Location: Καθορισμός της θέσης του χρήστη (IP διεύθυνση και port number).
- User Availability: Προσδιορισμός της διαθεσιμότητας του χρήστη.
- Endpoint Capabilities: Προσδιορισμός των δυνατοτήτων του χρήστη (π.χ. codecs).
- Session Setup: Εγκαθίδρυση της συνόδου επικοινωνίας (session) μεταξύ χρηστών.
- Session Management: Διαχείριση της συνόδου επικοινωνίας.

Το SIP δεν καθορίζει τον τύπο της επικοινωνίας που εγκαθίσταται, αλλά μόνο την διαχειρίζεται. Αυτό έχει ως συνέπεια ότι μπορεί να χρησιμοποιηθεί σε έναν τεράστιο αριθμό εφαρμογών και υπηρεσιών.

3.1 Λειτουργικές οντότητες του πρωτοκόλλου SIP (SIP Network Elements)

Το SIP περιλαμβάνει τέσσερις βασικές λειτουργικές οντότητες: user agent, proxy server, registrar server και redirect server. Ορισμένες οντότητες μπορούν να συνυπάρχουν σε ένα σύστημα. Ακολουθεί σύντομη περιγραφή των λογικών οντοτήτων του SIP.

SIP User Agent (UA)

Οι SIP User Agents είναι τα τερματικά σημεία (Endpoints) που χρησιμοποιούν το πρωτόκολλο SIP για την δημιουργία μιας επικοινωνίας μέσω του Διαδικτύου. Ουσιαστικά ο User Agent είναι μια εφαρμογή - ένα λογισμικό - που τρέχει σε μια συσκευή, για παράδειγμα σε μια IP τηλεφωνική συσκευή, σε ένα ATA, σε ένα ηλεκτρονικό υπολογιστή ή ένα κινητό τηλέφωνο ή tablet.

Λειτουργικά ένας SIP UA χωρίζεται σε δύο λογικές οντότητες, σύμφωνα με το μοντέλο πελάτη - εξυπηρετητή σε “πελάτη” User Agent Client (UAC) και σε “εξυπηρετητή” User Agent Server (UAS). Ο πελάτης (UAC) αποστέλλει SIP requests και λαμβάνει SIP responses, ενώ αντίθετα ο εξυπηρετητής (UAS) λαμβάνει SIP requests και αποστέλλει SIP responses. Συχνά λέμε ότι ένας UA συμπεριφέρεται ως UAC ή UAS, ανάλογα με το αν ο αποστέλλει ένα SIP request ή λαμβάνει ένα SIP response, αντίστοιχα.

Σημειώνουμε επίσης ότι το μέρος του SIP Proxy Server που λαμβάνει ένα request ενεργεί ως UAS, ενώ αντίθετα το μέρος που αποστέλλει ένα SIP request λειτουργεί ως UAC.

SIP Proxy Server

Ο SIP Proxy Server είναι η κεντρική λογική οντότητα ενός SIP δικτύου που αναλαμβάνει την δρομολόγηση των SIP μηνυμάτων που ανταλλάσσονται μεταξύ των UAs. Ο κύριος ρόλος ενός SIP Proxy Server είναι η προώθηση των SIP μηνυμάτων όσο το δυνατόν «πιο κοντά» στους αρμόδιους UAs. Τα μηνύματα αυτά θα περάσουν

συνήθως από αρκετούς Proxy Servers πριν φθάσουν στον τελικό τους προορισμό. Υπάρχουν δύο κατηγορίες από SIP Proxy Server:

1. **Stateless Proxy Server:** Είναι ο Proxy Server που εκτελεί απλή διανομή των SIP μηνυμάτων, χωρίς να διατηρεί κάποια πληροφορία και να έχει γνώση για την κατάσταση της συνόδου. Το βασικό πλεονέκτημα των stateless proxy servers είναι η απλότητα τους, η ικανότητα να υποστηρίξουν πολύ μεγάλο αριθμό χρηστών και η δυνατότητα να χρησιμοποιηθούν ως load balancers, message translators και routers.
2. **Statefull Proxy Server:** Ο Proxy Server αποθηκεύει πληροφορίες για τα εισερχόμενα μηνύματα και δημιουργεί και διατηρεί μια εσωτερική κατάσταση για κάθε σύνοδο μέχρι αυτή να ολοκληρωθεί. Οι statefull proxies έχουν μια σειρά από σημαντικά πλεονεκτήματα, όπως η δυνατότητα μετάδοσης SIP μηνυμάτων σε πολλαπλούς αποδέκτες (forking), η δυνατότητα αναμετάδοσης μηνυμάτων, εξασφαλίζοντας την αξιοπιστία σε ένα δίκτυο που χρησιμοποιεί μη αξιόπιστο πρωτόκολλο μεταφοράς, όπως το UDP, η καταγραφή αναλυτικών στοιχείων κλήσεων (σημαντικό για πάροχους), η υποστήριξη πολύπλοκων δικτυακών λειτουργιών, όπως NAT traversal.

SIP Registrar Server

Ένας SIP Registrar Server είναι μια ειδική SIP οντότητα που λαμβάνει τις αιτήσεις εισόδου στο δίκτυο SIP - μέσω SIP REGISTER requests - από τους χρήστες που ανήκουν στο domain που ελέγχει. Ο SIP Registrar εξάγει πληροφορίες για την τρέχουσα θέση των User Agents (IP διεύθυνση, port number και όνομα χρήστη) και δημιουργεί εγγραφές με αυτές τις πληροφορίες σε μια βάση δεδομένων. Έτσι, ένας SIP Proxy Server επικοινωνεί με τον Registrar Server για να ανακτήσει πληροφορίες για την τρέχουσα θέση χρηστών, με σκοπό να τους προωθήσει τα SIP μηνύματα που τους αφορούν.

SIP Redirect Server

Ένας SIP Redirect Server είναι μια SIP οντότητα που ενεργοποιείται όταν ένας χρήστης έχει μετακινηθεί προσωρινά ή μόνιμα σε νέα τοποθεσία και ταυτόχρονα έχει

αλλάξει η λογική του διεύθυνση. Στην περίπτωση που ένας χρήστης προσπαθεί να επικοινωνήσει, μέσω ενός Redirect Server, με ένα άλλο χρήστη που έχει μετακινηθεί, ανακατευθύνετε από τον SIP Redirect Server σε μία λίστα πιθανών εναλλακτικών λογικών διευθύνσεων που μπορεί να βρίσκεται ο καλούμενος χρήστης.

3.2 Μηνύματα και Μέθοδοι SIP

Στην ενότητα αυτή θα παρουσιαστούν με συντομία τα είδη των SIP μηνυμάτων, η σύνταξή τους καθώς και τα πεδία (header fields) που τα αποτελούν. Όπως έχει ήδη αναφερθεί, τα SIP μηνύματα είναι text μηνύματα, με σύνταξη που είναι κοντά στην φυσική αγγλική γλώσσα, γεγονός που προσφέρει απλότητα και εύκολα στην κατανόησή τους.

Ένα μήνυμα SIP αποτελείται από τη γραμμή έναρξης (start line), τις επικεφαλίδες (headers) και ένα σώμα (body). Ένα μήνυμα αιτήματος (request) έχει την ακόλουθη μορφή:

```
<request start line>  
<request headers> (πολλές γραμμές)  
<κενή γραμμή>  
<message body> (μεταφέρει το Session Description Protocol)
```

Για παράδειγμα:

```
INVITE sip:bob@biloxi.com SIP/2.0  
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds  
Max-Forwards: 70  
To: Bob <sip:bob@biloxi.com>  
From: Alice <sip:alice@atlanta.com>;tag=1928301774  
Call-ID: a84b4c76e66710@pc33.atlanta.com  
CSeq: 314159 INVITE  
Contact: <sip:alice@pc33.atlanta.com>  
Content-Type: application/sdp  
Content-Length: 142  
  
(SDP not shown)
```

Όμοια ένα μήνυμα απάντησης (response) είναι ως ακολούθως:

<response status line>
<request headers> (πολλές γραμμές)
<κενή γραμμή>
<message body>

Για παράδειγμα:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP
server10.biloxi.com;branch=z9hG4bKnashds8;received=192.0.2.3
Via: SIP/2.0/UDP
bigbox3.site3.atlanta.com;branch=z9hG4bK77ef4c2312983.1;received=192.0.2
.2
Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds
;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4>
Content-Type: application/sdp
Content-Length: 131

(SDP not shown)
```

Η διευθύνσιδοδότηση των χρηστών ενός SIP δικτύου βασίζεται στο Unified Resource Identifier (URI) syntax notation, που ορίζεται στο RFC 2396. Η γενική σύνταξη μιας SIP URI διεύθυνσης, με αρκετά προαιρετικά πεδία, είναι ως εξής:

sip:user:[password@host](#):port;uri-parameters?headers

Μια τυπική SIP διεύθυνση μοιάζει ως εξής "sip:username@domain". Η διεύθυνση μοιάζει πολύ με τις http διευθύνσεις, καθώς και αυτές ακολουθούν το URI notation.

SIP Requests

Παρακάτω περιγράφονται τα requests που υποστηρίζονται από το πρωτόκολλο SIP:

1. **ACK:** Ένα ACK request αποστέλλετε από έναν UA client για να επιβεβαιώσει την λήψη final response από έναν SIP server ή UA server, όπως 200 OK σε ένα INVITE request.

2. **BYE**: Ένα BYE request αποστέλλεται είτε από τον caller agent είτε από τον callee agent με σκοπό τον τερματισμό της κλήσης.
3. **CANCEL**: Ένα CANCEL request μπορεί να αποσταλεί με σκοπό την ακύρωση της επεξεργασίας ενός request που στάλθηκε προηγουμένως, υπό την προϋπόθεση ότι ο SIP server που έλαβε το request δεν έχει προωθήσει final response στον UAC που το απέστειλε.
4. **INFO**: Το INFO request ορίζεται στο RFC 2976 και χρησιμοποιείται για την μεταφορά πληροφοριών εν μέσω μιας κλήσης, χωρίς όμως να επηρεάζουν την τρέχουσα κατάσταση της.
1. **INVITE**: Ένα INVITE request χρησιμοποιείται για την εγκατάσταση μιας κλήσης, καθώς και την αλλαγή των παραμέτρων μιας εν εξελίξει κλήσης.
2. **MESSAGE**: Το MESSAGE request ορίζεται στο RFC 3428 και χρησιμοποιείται για την αποστολή Instant Messages.
3. **NOTIFY**: Το NOTIFY request ορίζεται στο RFC 3265 και χρησιμοποιείται για την αποστολή event notifications.
4. **OPTIONS**: Ένας UA client αποστέλλει ένα OPTION request σε έναν SIP server με σκοπό να μάθει για τις SIP μεθόδους που υποστηρίζει ο server. Ο SIP server, μέσω ενός SIP response (συνήθως ένα 200 OK) απαντά με μία Η διεθυνσιοδότηση των χρηστών ενός SIP δικτύου βασίζεται στο Unified Resource Identifier (URI) syntax notation, που ορίζεται στο RFC 2396. Η γενική σύνταξη μιας SIP URI διεύθυνσης, με αρκετά προαιρετικά πεδία, είναι ως εξής:
 5. sip:user:[password@host](#):port;uri-parameters?headers
 6. Μια τυπική SIP διεύθυνση μοιάζει ως εξής “sip:username@domain”. λίστα που περιέχει τις μεθόδους που υποστηρίζει. Επιπλέον, ένα OPTIONS request μπορεί να αποσταλεί από ένα UAC προς έναν UAS ζητώντας να μάθει εκτός από τις μεθόδους που υποστηρίζει και τα media capabilities του UAS, καθώς επίσης και τον τρόπο που θα αποκρινόταν στην αποστολή ενός INVITE request.

7. **PRACK:** Το PRACK request ορίζεται στο RFC 3262 και χρησιμοποιείται για την εφαρμογή αξιοπιστίας στην αποστολή provisional responses (1XX responses).
8. **REFER:** Το REFER request ορίζεται στο RFC 3515 και χρησιμοποιείται για session redirection.
9. **REGISTER:** Το REGISTER request αποστέλλεται από έναν UA προς έναν SIP Registrar Server, ώστε να καταχωρηθεί η τρέχουσα θέση του UA στον υπεύθυνο Location Server.
10. **SUBSCRIBE:** Το SUBSCRIBE request ορίζεται στο RFC 3265 και χρησιμοποιείται από έναν UA με σκοπό να αιτηθεί την λήψη συγκεκριμένων event notifications.
11. **UPDATE:** Το UPDATE request ορίζεται στο RFC 3311 και χρησιμοποιείται με σκοπό την αλλαγή των media παραμέτρων μιας κλήσης για την οποία δεν έχει αποσταλεί ή ληφθεί final response (early dialogs).

SIP Responses

Ένας SIP Server ανταποκρίνεται σε ένα SIP request με ένα ή περισσότερα SIP responses. Υπάρχουν έξι διαφορετικές κατηγορίες responses (1XX, 2XX, 3XX, 4XX, 5XX και 6XX). Όλες οι κατηγορίες εκτός από την πρώτη είναι final responses, τα οποία ολοκληρώνουν μια συναλλαγή SIP. Από την άλλη μεριά τα provisional responses (1XX) αποτελούν προσωρινές απαντήσεις, οι οποίες απλά δηλώνουν την πρόοδο της επεξεργασίας ενός SIP request.

Κάθε SIP response αποτελείται από τρία ψηφία, από τα οποία το πρώτο καθορίζει τη κατηγορία της απάντησης, ενώ τα υπόλοιπα δύο είναι συγκεκριμένα για κάθε response. Για παράδειγμα το response 486 Busy Here, το ψηφίο “4” δηλώνει την κατηγορία, η οποία σε αυτή την περίπτωση είναι “Client Failure”, ενώ τα ψηφία 86 είναι μοναδικά για αυτό το response. Επιπλέον, η πρόταση “Busy Here” ονομάζεται reason phrase και αποτελεί μία σύντομη text περιγραφή του SIP response. Τα τρία

ψηφία μαζί με το reason phrase και την έκδοση του πρωτοκόλλου SIP αποτελούν την Start Line ενός SIP response: SIP/2.0 486 Busy Here

Αν εξαιρέσουμε την Start Line, το υπόλοιπο μήνυμα ακολουθεί τη δομή που ακολουθεί ένα SIP request. Μάλιστα, το μεγαλύτερο μέρος των επικεφαλίδων, καθώς και του σώματος ενός SIP response αντιγράφεται από το σχετικό SIP request.

3.3 Δομή του Πρωτοκόλλου SIP

Το SIP είναι ένα δομημένο σε στρώματα πρωτόκολλο σηματοδοσίας, κάτι που σημαίνει ότι η συμπεριφορά του καθορίζεται από την συμπεριφορά κάθε ανεξάρτητου στρώματος που το αποτελεί. Όταν λέμε ότι μία οντότητα SIP περιέχει ένα στρώμα του πρωτοκόλλου, εννοούμε ότι η οντότητα «υπακούει» στο σύνολο των κανόνων που καθορίζονται από εκείνο το στρώμα. Τα στρώματα (layers) του πρωτοκόλλου SIP είναι τα ακόλουθα:

Syntax and Encoding Layer: Είναι το κατώτερο στρώμα του πρωτοκόλλου SIP και είναι υπεύθυνο για τη σύνταξη και την κωδικοποίησή των SIP μηνυμάτων, η οποία ακολουθεί τη σύνταξη των HTTP/1.1 μηνυμάτων.

Transport Layer: Είναι το δεύτερο στρώμα, το οποίο είναι υπεύθυνο για την μεταφορά των SIP μηνυμάτων μέσω του IP δικτύου. Όλες οι οντότητες SIP περιέχουν ένα στρώμα μεταφοράς.

Transaction Layer: Είναι το τρίτο στρώμα του πρωτοκόλλου SIP και είναι απαραίτητο σε κάθε SIP οντότητα, διότι καθορίζει τις απαραίτητες συναλλαγές μεταξύ τους. Μια συναλλαγή είναι ένα αίτημα (request) που στέλνεται από ένα client transaction σε ένα server transaction, μαζί με όλες τις απαντήσεις (responses) που ακολουθούν και αποστέλλονται από το server transaction πίσω προς το client transaction. Οι λειτουργίες που διαχειρίζεται το transaction layer είναι:

- Η πραγματοποίηση των απαραίτητων αναμεταδόσεων μηνυμάτων στην περίπτωση που χρησιμοποιείται ένα μη αξιόπιστο πρωτόκολλο μεταφοράς, όπως το UDP.

- Η διαχείριση των timers οι οποίοι καθορίζουν τα χρονικά πλαίσια των αναμεταδόσεων.
- Η διαδικασία αντιστοίχισης SIP μηνυμάτων που λαμβάνονται από το δίκτυο με τα αντίστοιχα transactions που βρίσκονται σε εξέλιξη.

Όπως αναφέρθηκε και προηγουμένως, ένα SIP transaction αποτελείται από ένα SIP request και όλα τα σχετικά SIP responses που ακολουθούν. Στην περίπτωση που το αρχικό request είναι ένα INVITE request, τότε το transaction που θα ακολουθήσει είναι ένα INVITE transaction, ενώ σε όλες τις άλλες περιπτώσεις (εκτός από το ACK request) το transaction που θα ακολουθήσει είναι ένα NON-INVITE transaction. Το ACK request δεν μπορεί να ξεκινήσει ένα transaction, αλλά χρησιμοποιείται μόνο για να επιβεβαιώσει την λήψη responses σε προηγούμενα requests. Επιπλέον, ανάλογα με το αν ένα transaction αρχικοποιείται λόγω αποστολής ή λήψης SIP request, τα transactions χωρίζονται σε client και server transactions, αντίστοιχα.

Transaction User Layer: Είναι το ανώτερο στρώμα του πρωτοκόλλου SIP το οποίο είναι υπεύθυνο για την διαχείριση των οντοτήτων SIP, εκτός από τους stateless SIP Proxy Servers. Όταν ένα Transaction User (TU) επιθυμεί να στείλει ένα SIP request, δημιουργεί ένα client transaction με παραμέτρους το request, καθώς και την IP διεύθυνση και port number του προορισμού. Στη συνέχεια το transaction και transport layer αναλαμβάνουν την αξιόπιστη μετάδοση του SIP request στον προορισμό του. Να σημειωθεί ότι ένας TU που δημιουργεί ένα client transaction μπορεί επίσης να το ακυρώσει, ζητώντας στην ουσία την ακύρωση της επεξεργασίας του SIP request από το SIP UAS ή τον ενδιαμέσο Proxy που το έλαβε. Κάτι τέτοιο πρακτικά γίνεται με την αποστολή ενός CANCEL request.

SIP Transactions

Όπως αναφέρθηκε, τα SIP transactions χωρίζονται σε κατηγορίες ανάλογα με το αν ένα transaction αρχικοποιείται λόγω αποστολής ή λήψης SIP request (client και server transaction). Επομένως, είναι φανερό πως τα SIP Transactions σχετίζονται με SIP μηνύματα. Έτσι, κάθε transaction αναγνωρίζεται από παραμέτρους που εμφανίζονται σε ένα SIP μήνυμα όπως: η branch παράμετρος του Via Header και το CSeq number

και method. Χρησιμοποιώντας αυτές τις παραμέτρους είναι δυνατόν να γίνει αντιστοίχιση ενός SIP μηνύματος με ένα SIP Transaction.

Μια επιπλέον κατηγοριοποίηση των SIP transactions έχει να κάνει με το αν το αρχικό SIP request που προκαλεί την δημιουργία του SIP transaction είναι ένα INVITE ή non-INVITE, αντίστοιχα. Η κατηγοριοποίηση αυτή έχει γίνει για το λόγο ότι ένα INVITE transaction είναι διαφορετικό από τα non-INVITE transactions, λόγω της εκτεταμένης διάρκειάς του, εφόσον είναι απαραίτητη η αλληλεπίδραση με το χρήστη για την απάντηση σε ένα INVITE request (π.χ. σήκωμα του ακουστικού). Για το λόγο αυτό τα INVITE transactions ακολουθούν μία three-way handshake διαδικασία για να ολοκληρωθούν, σε αντίθεση με τα non-INVITE transactions, τα οποία ολοκληρώνονται άμεσα και αυτόματα, ακολουθώντας μία two-way handshake διαδικασία.

Με βάση τις παραπάνω κατηγοριοποιήσεις υπάρχουν τέσσερα είδη SIP transactions, τα οποία παρουσιάζονται πολύ συνοπτικά παρακάτω.

1. INVITE Client Transactions

Στην κατηγορία αυτή ανήκουν τα SIP transactions τα οποία δημιουργούνται από ένα SIP UAC, όταν επιθυμεί την αποστολή ενός INVITE request, για την εγκατάσταση μιας VoIP κλήσης. Ένα INVITE Client Transaction αποτελείται από μία three-way handshake διαδικασία. Το transaction ξεκινά όταν ο χρήστης δημιουργήσει ένα INVITE request και το μεταβιβάσει στο Transaction Layer του πρωτοκόλλου. Η three-way handshake διαδικασία, που λαμβάνει χώρα στο transaction Layer, περιλαμβάνει την αποστολή του INVITE request, την λήψη ενός τελικού response και την αποστολή ενός ACK request για την επιβεβαίωση της λήψης.

Στη συνέχεια, όταν ληφθεί ένα provisional response (π.χ. 180 Ringing response), οι αναμεταδόσεις του αρχικού INVITE request σταματούν και ο χρήστης αναμένει την λήψη ενός τελικού response, που θα δείχνει την αποδοχή ή την απόρριψη της κλήσης από τον απομακρυσμένο user. Στην περίπτωση που έχουν λάβει χώρα επτά αναμεταδόσεις του αρχικού INVITE και δεν έχει ληφθεί provisional response, τότε το transaction τερματίζει αυτόματα την λειτουργία του, ενώ πιθανά επιπλέον provisional responses που καταφθάνουν στο transport layer και αντιστοιχούνται στο ίδιο transaction, δεν λαμβάνονται υπόψη.

Η λήψη ενός 2XX response προκαλεί τον άμεσο τερματισμό του transaction και η αποστολή του ACK request αναλαμβάνεται από το υψηλότερο στρώμα του πρωτοκόλλου SIP, που είναι το Transaction User Layer.

2. Non-INVITE Client Transactions

Στην κατηγορία αυτή ανήκουν τα SIP transactions τα οποία δημιουργούνται από ένα SIP UAC, όταν επιθυμεί την αποστολή ενός request, εκτός από INVITE και ACK. Τέτοια requests μπορεί να είναι για παράδειγμα BYE ή CANCEL request για την ακύρωση εγκατάστασης ή τον τερματισμό μιας VoIP κλήσης. Η βασική αρχή λειτουργίας των Non-INVITE Client Transactions είναι η χρήση two-way handshake για την ολοκλήρωσή τους, σε αντίθεση με τα INVITE Client Transactions, λόγω του ότι δεν χρησιμοποιείται το ACK request για την επιβεβαίωση των τελικών responses.

3. INVITE Server Transactions

Στην κατηγορία αυτή ανήκουν τα SIP transactions τα οποία δημιουργούνται από ένα SIP UAS, όταν λαμβάνει ένα INVITE request από έναν απομακρυσμένο UAC, με σκοπό την εγκατάσταση μιας VoIP κλήσης. Ένα INVITE Server Transaction αποτελείται από μία three-way handshake διαδικασία και ξεκινά όταν το INVITE request παραδίδεται από το Transport Layer στο Transaction Layer του πρωτοκόλλου SIP.

Στην περίπτωση που χρησιμοποιείται μη αξιόπιστο πρωτόκολλο μεταφοράς, όπως UDP και έχει αποσταλεί provisional response (π.χ. 180 Ringing response, το οποίο σημαίνει ότι το τηλέφωνο του τοπικού user χτυπά), το transaction παραδίδει για αναμετάδοση στο transport layer το τελευταίο provisional response κάθε φορά που λαμβάνεται μία επανάληψη του αρχικού INVITE request.

Ο χειρισμός των τελικών responses από ένα INVITE Server Transaction είναι διαφορετικός από το χειρισμό των provisional responses. Συγκεκριμένα, τα τελικά responses εκτός από τα 2XX, επαναδρομολογούνται κάθε 500 ms, για όσο χρονικό διάστημα δεν λαμβάνεται ένα ACK request. Και σε αυτή την περίπτωση, το χρονικό διάστημα μεταξύ δύο αναμεταδόσεων διπλασιάζεται μετά από κάθε αναμετάδοση, ενώ οι αναμεταδόσεις σταματούν μετά την έβδομη επανάληψη, ή αν ληφθεί ένα matching BYE request, στην περίπτωση του 200 OK response, ή ένα matching

CANCEL request στην περίπτωση των 3XX, 4XX, 5XX, 6XX responses. Να σημειωθεί ότι η αναμετάδοση των 200 OK responses σε INVITE requests δεν πραγματοποιείται από το Transaction Layer, αλλά το αναλαμβάνει το Transaction User Layer.

4. Non-INVITE Server Transactions

Στην κατηγορία αυτή ανήκουν τα SIP transactions τα οποία δημιουργούνται από ένα SIP UAS, όταν λαμβάνει ένα non-INVITE request από έναν απομακρυσμένο UAC, όπως CANCEL ή BYE request.

Όπως και στην περίπτωση των Non-INVITE Client Transactions και σε αυτή την περίπτωση χρησιμοποιείται two-way handshake διαδικασία για την ολοκλήρωσή τους, επειδή δεν λαμβάνεται ACK request για την επιβεβαίωση τελικών responses που στάλθηκαν προηγουμένως. Έτσι, στην περίπτωση που χρησιμοποιείται μη αξιόπιστη σύνδεση για την ανταλλαγή δεδομένων, κατά την διάρκεια ενός non-INVITE Server Transaction αναμεταδίδεται το τελευταίο response- ανεξάρτητα από το αν είναι τελικό ή provisional- κάθε φορά που λαμβάνεται μια αναμετάδοση του αρχικού request.

SIP Dialogs

Ένα SIP Dialog ορίζει μία peer-to-peer επικοινωνία μεταξύ δύο SIP User Agents, η οποία παραμένει ενεργή για το χρονικό διάστημα που οι SIP User Agents επικοινωνούν μεταξύ τους. Η βασικότερη λειτουργία των SIP dialogs είναι η διατήρηση της σωστής αλληλουχίας των SIP μηνυμάτων που ανταλλάσσονται. Σύμφωνα με το RFC 3261, η αρχικοποίηση ενός SIP Dialog μπορεί να πραγματοποιηθεί μόνο μέσω της μεθόδου INVITE. Επιπλέον, η αποστολή ενός SIP request από ένα UAC μέσα στα πλαίσια ενός εγκατεστημένου SIP Dialog, πρέπει να γίνεται ακολουθώντας συγκεκριμένους κανόνες οι οποίοι επίσης καθορίζονται στο RFC 3261.

Ένα SIP Dialog προσδιορίζεται από ένα Dialog ID, το οποίο προκύπτει από το συνδυασμό της τιμής του Call-ID Header του INVITE request που αρχικοποίησε το SIP Dialog, καθώς και των tag παραμέτρων των remote και local User Agents που συμμετέχουν στην peer-to-peer επικοινωνία και οι οποίες εμφανίζονται στα To και From Headers των SIP μηνυμάτων, αντίστοιχα. Με βάση τον παραπάνω συσχετισμό

είναι εύκολο να γίνει αντιστοίχιση των εισερχόμενων SIP μηνυμάτων σε open SIP Dialogs, αλλά και να δημιουργηθούν νέα προς αποστολή μηνύματα στα πλαίσια του διαλόγου αυτού, δίνοντας έτσι την δυνατότητα για ανταλλαγή δεδομένων και πληροφοριών εν μέσω μιας κλήσης. Τέτοια μηνύματα είναι για παράδειγμα το OPTIONS request, το οποίο αποστέλλεται με σκοπό να διαπιστωθούν τα media capabilities του συνομιλητή, ενώ η κλήση είναι σε εξέλιξη.

Ένας UAC αρχικοποιεί ένα SIP Dialog όταν λάβει το πρώτο non-failure response για το INVITE που απέστειλε, ενώ ένας UAS αρχικοποιεί ένα SIP Dialog όταν αποστέλλει το πρώτο non-failure response για το INVITE που έλαβε.

Ο τερματισμός ενός SIP Dialog, πραγματοποιείται είτε με την αποστολή ή λήψη BYE request, είτε με την αποστολή ή λήψη τελικού non-2XX response που αφορά το αρχικό INVITE request.

Επιλογή των Παραμέτρων της Κλήσης

Κατά την εγκατάσταση μιας VoIP κλήσης, μία από τις πιο σημαντικές διαδικασίες είναι η επιλογή των παραμέτρων που θα χρησιμοποιηθούν κατά την διάρκεια της κλήσης. Η επιλογή αυτή γίνεται μέσω της διαδικασίας negotiation κατά την οποία γίνεται προσπάθεια συμφωνίας των συμμετεχόντων στην κλήση, όσον αφορά στους τύπους των δεδομένων που θα ανταλλαχθούν (π.χ. audio, video), την κωδικοποίηση που θα χρησιμοποιηθεί (CODECS), καθώς και πληροφορία για την αποστολή και λήψη των δεδομένων, όπως destination IP addresses και port numbers. Η διαδικασία του negotiation βασίζεται στην ανταλλαγή SDP πακέτων, τα οποία συμπεριλαμβάνονται ως bodies στα SIP messages που ανταλλάσσονται για την εγκατάσταση της VoIP επικοινωνίας, όπως INVITE request και 200 OK response.

Στη συνέχεια γίνεται μια σύντομη παρουσίαση των πρωτοκόλλου SDP, καθώς και της διαδικασίας negotiation για την επιλογή των παραμέτρων μιας κλήσης.

3.4 Το Πρωτόκολλο SDP

Το πρωτόκολλο SIP χρησιμοποιεί το Session Description Protocol (SDP), το οποίο περιγράφεται στο RFC 2237. Το SDP είναι και αυτό ένα προϊόν της ομάδας εργασίας MMUSIC και ο αρχικός σκοπός του ήταν ο καθορισμός μιας τυποποιημένης σύνταξης για την περιγραφή των δεδομένων που ανταλλάσσονται κατά την διάρκεια συνόδων επικοινωνίας. Οι πιο σημαντικές παράμετροι που περιλαμβάνονται σε ένα SDP πακέτο, ώστε να είναι πλήρης η περιγραφή ενός session επικοινωνίας, φαίνεται ακόλουθο πίνακα.

Παράμετρος	Περιγραφή
v	Έκδοση του πρωτοκόλλου SDP
s	Το όνομα του session
t	Ο χρόνος που ένα session παραμένει ενεργό
o	Ο χρήστης που είναι υπεύθυνος για το session (owner)
c	IP Διεύθυνση για την λήψη των media δεδομένων που περιγράφονται στο SDP από την αντίστοιχη m=' line
m	Περιγραφή των δεδομένων media, που θα χρησιμοποιηθούν κατά το session
a	Επιπλέον πληροφορίες για τη λήψη των media δεδομένων, όπως codecs και πρωτόκολλο μεταφοράς

Εκτός από τις παραμέτρους του πίνακα αυτού υπάρχουν κι άλλες για την περιγραφή επιπλέον πληροφοριών που αφορούν ένα session. Τέτοιες πληροφορίες μπορεί να είναι το bandwidth που θα χρησιμοποιηθεί κατά τη διάρκεια του session, επιπλέον contact πληροφορίες για τον owner, όπως e-mail και web site address, κ.τ.λ.

Η περιγραφή των media δεδομένων, που περιλαμβάνεται σε ένα SDP πακέτο, είναι ιδιαίτερα σημαντική, διότι εμπλέκεται άμεσα στην διαδικασία διαπραγμάτευσης και επιλογής του τύπου και της κωδικοποίησης των δεδομένων που θα ανταλλάξουν δύο ή περισσότεροι επλεκόμενοι σε μία κλήση. Πιο συγκεκριμένα, η περιγραφή media δεδομένων σε ένα SDP πακέτο περιλαμβάνει:

1. Τον τύπο των media δεδομένων (video, audio, κλπ).
2. Το πρωτόκολλο μεταφοράς των media δεδομένων (RTP/UDP/IP, H.320, κλπ)
3. Το port λήψης media δεδομένων.

4. Το format κωδικοποίησης των media δεδομένων (H.261 video, MPEG video, PCMU audio, κλπ)

Για παράδειγμα, θεωρήστε το παρακάτω μήνυμα SDP:

```
v = 0
o = bob 1787262837 1787262837 IN IP4 10.0.0.1
s =
t = 0 0
c = IN IP4 10.0.0.1
m = audio 41732 RTP/AVP 0
a = rtpmap:0 PCMU/8000
```

Το μήνυμα αυτό περιγράφει ένα session με τα εξής χαρακτηριστικά:

- Το version number του πρωτοκόλλου είναι 0 (v parameter).
- Ο υπεύθυνος για την διαχείριση του session είναι ο χρήστης bob, που βρίσκεται στην διεύθυνση 10.0.0.1 (o parameter).
- Δεν υπάρχει subject για το session που περιγράφει το πακέτο (s parameter).
- Τα media δεδομένα που περιγράφονται από την γραμμή 'm=' θα αποστέλλονται από την διεύθυνση 10.0.0.1 (c parameter).
- Κατά την διάρκεια του session θα ανταλλάσσονται audio δεδομένα, με port λήψης το 41732, πρωτόκολλο μεταφοράς το RTP και RTP payload value το 0 (m parameter).
- Τα audio δεδομένα που περιγράφονται από την 'm=' γραμμή θα είναι κωδικοποιημένα με μ-law κωδικοποίηση (PCMU) και sampling rate 8000 samples/sec (a parameter).

3.5 Επιλογή των Παραμέτρων Κλήσης μέσω του μοντέλου Offer/Answer

To RFC 3264, "An Offer/Answer Model with the Session Description Protocol (SDP)", παρουσιάζει όλες τις διαδικασίες που είναι απαραίτητες για την διαπραγμάτευση των παραμέτρων μιας κλήσης, χρησιμοποιώντας το πρωτόκολλο SDP. Οι παράμετροι της κλήσης που μπορούν να διαπραγματευτούν είναι οι media codecs που θα χρησιμοποιηθούν για τη κωδικοποίηση των media streams που ανταλλάσσονται κατά τη διάρκεια μιας κλήσης, καθώς και παράμετροι όπως IP address και port numbers για αποστολή και λήψη RTP/RTCP πακέτων.

Η διαδικασία της διαπραγμάτευσης των παραμέτρων μιας κλήσης βασίζεται σε offers και answers. Ένα offer αποτελείται από τους προτεινόμενους media τύπους και codecs μιας επικοινωνίας και είναι στην ουσία το SDP σώμα του αρχικού INVITE request που αποστέλλεται από τον UAC, δηλ. τον καλών (caller). Ένα answer αποτελείται από ένα υποσύνολο των media τύπων και codecs, που πρότεινε ο caller, και ταυτόχρονα υποστηρίζει ο καλούμενος (callee), που δέχθηκε το offer. Το answer είναι στην ουσία το SDP σώμα του 200 OK response που αποστέλλει ο callee πίσω στον caller, αν αποδεχθεί την κλήση. Η απάντηση αυτή προκύπτει μέσω της διαδικασίας SDP negotiation, ώστε και οι δύο πλευρές να έρθουν σε συμφωνία για τα media types και τους codecs της επικοινωνίας.

Ένα offer ή answer περιέχει μία ή περισσότερες περιγραφές media streams, μία σε κάθε γραμμή που ξεκινά με 'm=', ακολουθούμενη από επιπλέον πεδία όπως:

- Ο τύπος των media δεδομένων, π.χ. audio ή video.
- Το RTP port number για τη λήψη των media δεδομένων.
- Οι payload τιμές των codecs (media formats) για την κωδικοποίηση/αποκωδικοποίηση των media δεδομένων.

Για παράδειγμα, μία media line μπορεί να είναι η εξής:

```
m = audio 41732 RTP/AVP 0 1
```

Η παραπάνω media line όταν περιέχεται σε ένα offer προτείνει την χρήση δύο audio codecs, 0 και 1, τα οποία αναμένονται να ληφθούν στο port 41732. Η σειρά 0 και 1

δεν είναι τυχαία αλλά δηλώνει την σειρά προτίμησης του offerer για τους προτεινόμενους codecs.

Μετά από κάθε media line υπάρχει μία σειρά από a-attributes που περιέχουν επιπλέον ιδιότητες για τον media τύπο που περιγράφεται στην media line (ή media stream), καθώς επίσης και για τα media formats. Το πιο σημαντικό a-attribute είναι το 'a=rtpmap', το οποίο υποδεικνύει τους codecs για την κωδικοποίηση/αποκωδικοποίηση των media δεδομένων, ο τύπος των οποίων καθορίζεται στο media line (π.χ. audio). Στην περίπτωση που η μεταφορά δεδομένων γίνεται μέσω RTP, το attribute 'a = rtpmap' περιέχει και την αντιστοίχιση ενός codec με το RTP payload value.

Ένα άλλο σημαντικό a-attribute ενός media stream είναι αυτό που περιγράφει την κατεύθυνση του. Έτσι το attribute 'a = sendonly', υποδεικνύει την υποστήριξη μόνο αποστολής δεδομένων, ενώ το attribute 'a = recvonly' την υποστήριξη μόνο λήψης δεδομένων. Στην περίπτωση που ένα media stream περιέχει το attribute 'a = inactive', τότε υποδεικνύεται ότι το media stream υποστηρίζεται αφενός, αλλά αυτή τη στιγμή δεν μπορεί να γίνει αποστολή ή λήψη του συγκεκριμένου media stream. Σε κάθε άλλη περίπτωση, ένα media stream περιέχει το attribute 'a = sendrecv', ή τίποτα καθώς το τελευταίο attribute είναι το default, υποδεικνύοντας την ικανότητα αποστολής και λήψης του συγκεκριμένου τύπου media.

Με βάση την παραπάνω περιγραφή για τα media lines και τα a-attributes, θεωρήστε για παράδειγμα, το ακόλουθο SDP offer:

```
...  
m = audio 41732 RTP/AVP 0 1 4  
a = rtpmap:0 PCMU/8000  
a = rtpmap:1 PCMA/8000  
a = rtpmap:4 G723/8000  
...
```

προτείνει:

- Ανταλλαγή audio δεδομένων μέσω RTP με payload value 0 ή 1 ή 4.

- Όπου οι τιμές 0, 1 και 4, αντιστοιχούν στους codecs PCMU, PCMA και G723, με προτίμηση στον PCMU.

Η δημιουργία του answer βασίζεται κατά κύριο λόγο στο offer, με τροποποίηση μερικών στοιχείων όπως η IP διεύθυνση και το port number για τη λήψη RTP δεδομένων, η αφαίρεση τύπων media και codecs που δεν υποστηρίζονται από τον answerer, κ.λ.π. Πρέπει να υπάρχει μία ένα προς ένα αντιστοίχιση μεταξύ του αριθμού των 'm=' lines που εμφανίζονται στο offer και στο answer. Εάν ο answerer δεν υποστηρίζει ένα media stream που εμφανίζεται στο offer ως μια 'm=' line, τότε στην απάντησή του γνωστοποιεί το παραπάνω γεγονός, συμπεριλαμβάνοντας αυτή την 'm=' line, αλλά θέτοντας το port number μηδενικό:

```
m = audio 0 RTP/AVP 0 1
```

Κατά την διαδικασία αυτή είναι σημαντικό να αναφερθεί ότι η σειρά εμφάνισης των media formats στο offer αντιπροσωπεύει την προτίμηση του offerer, όσον αφορά τους codecs που επιθυμεί να χρησιμοποιηθούν κατά τη διάρκεια της κλήσης. Από την άλλη μεριά, η σειρά με την οποία θα εμφανίσει ο answerer τα media formats στο answer αντιπροσωπεύει την δική του σειρά προτίμησης. Εντούτοις, προκειμένου να μεγιστοποιηθούν οι πιθανότητες να επέλθει συμφωνία για τους codecs της επικοινωνίας, το answer πρέπει να περιέχει τους codecs που εμφανίζονται στο offer και υποστηρίζονται από τον answerer και μάλιστα με την ίδια σχετική σειρά εμφάνισής τους στο offer. Ομοίως, ο offerer πρέπει να κάνει αποδεκτή την λίστα που περιέχεται στο answer και αποστέλλεται σε αυτόν μέσω του 200 OK μηνύματος που ακολουθεί το αρχικό INVITE. Τελικά, το set των codecs της επικοινωνίας, το οποίο προκύπτει από την διαδικασία της διαπραγμάτευσης είναι αυτό που συμπεριλαμβάνεται στο answer. Από το set αυτό, συνήθως χρησιμοποιείται ο πρώτος στην σειρά codec στην αρχή τουλάχιστον της επικοινωνίας, αλλά και οι δύο participants πρέπει να είναι έτοιμοι να δεχθούν media δεδομένα κωδικοποιημένα σε οποιαδήποτε format συμπεριλαμβάνεται στο set που προέκυψε από την διαδικασία του negotiation.

Σε όλες τις περιπτώσεις, η IP διεύθυνση και το port number ενός media stream υποδεικνύουν τον προορισμό όπου ο offerer ή answerer αναμένει την λήψη του media stream. Για όλους τους τύπους των offers ή answers, εκτός από την περίπτωση που το media stream είναι χαρακτηρισμένο ως inactive, η IP διεύθυνση και το port

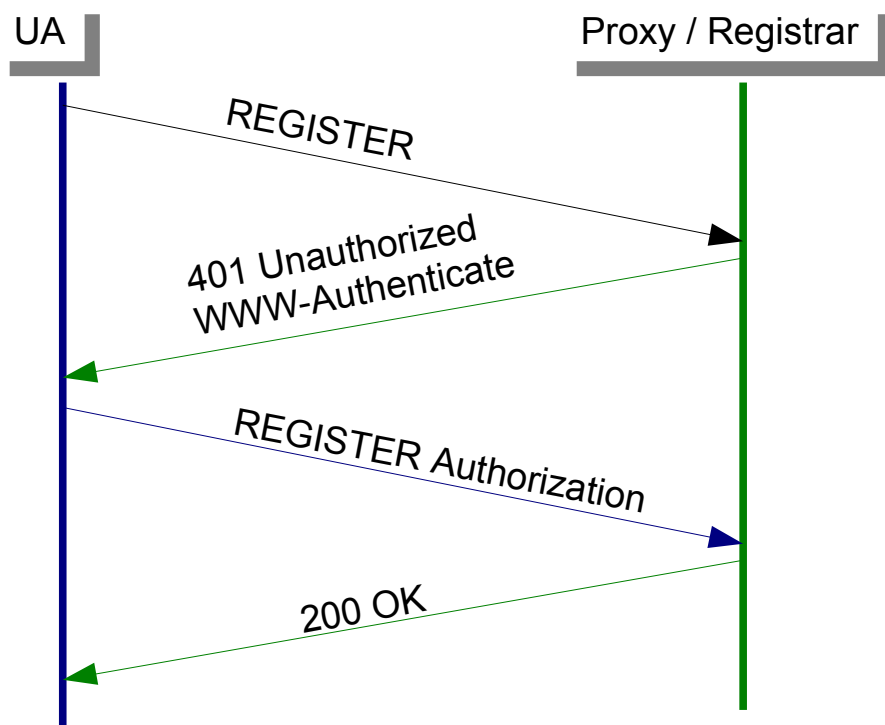
number + 1 αποτελεί τον προορισμό όπου ο offerer ή answerer αναμένει την λήψη RTCP πακέτων. Το παραπάνω δεν ισχύει αν η RTCP IP address και RTCP port number δηλώνονται σαφώς μέσω ενός 'a=rtcp' attribute.

3.6 Πραγματοποίηση Κλήσεων με το SIP

Σε αυτήν την ενότητα θα παρουσιάσουμε το τρόπο καταχώρησης και πραγματοποίησης κλήσεων με το πρωτόκολλο SIP. Οι διαδικασίες συνοδεύονται από κατάλληλα διαγράμματα ροής των SIP μηνυμάτων μεταξύ των συμμετεχόντων.

Καταχώρηση / Registration

Η διαδικασία του Registration είναι μία από τις πιο βασικές και σημαντικές διαδικασίες σε ένα SIP δίκτυο. Με την διαδικασία του Registration, οι users του δικτύου εγγράφονται σε λίστες που διατηρούν οι Registrar Servers του domain που ανήκουν. Οι εγγραφές αυτές είναι στην ουσία μια αντιστοίχιση της λογικής οντότητας του χρήστη, π.χ. bob@anywhere.com, με την πραγματική τρέχουσα φυσική του θέση, όπως αυτή εκφράζεται μέσω μιας συγκεκριμένης IP διεύθυνσης, π.χ. bob@192.10.10.1 (IP address του γραφείου του).

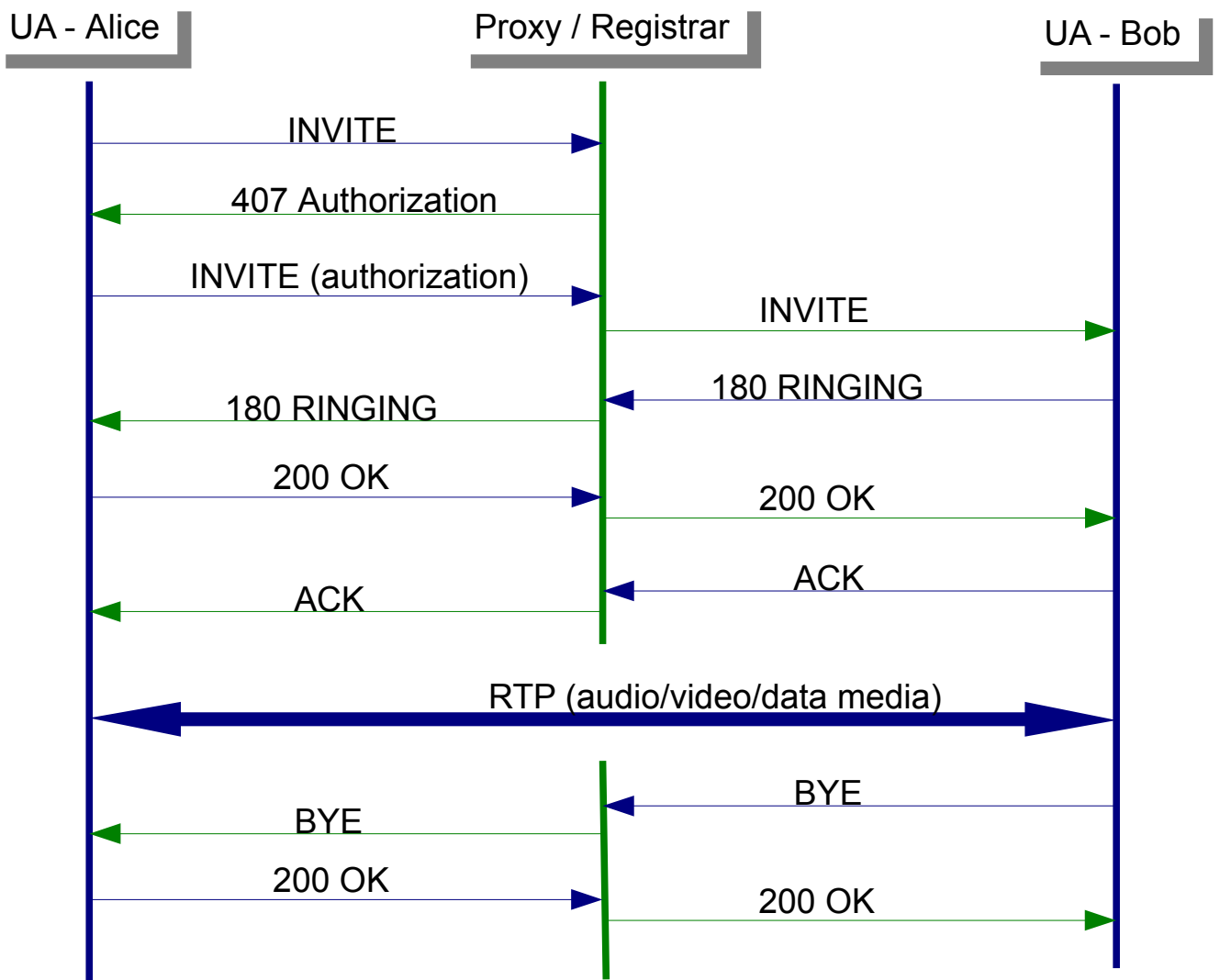


Σχήμα 3.1 Ανταλλαγή SIP μηνυμάτων για επιτυχές Registration

Στο Σχήμα 3.1 φαίνεται το διάγραμμα ανταλλαγής των απαραίτητων SIP μηνυμάτων για την εκτέλεση επιτυχούς registration ενός χρήστη Bob σε ένα domain. Αρχικά ο χρήστης αποστέλλει στον server ένα REGISTER request. Ο server απαντά με 401 Unauthorized response, ζητώντας credentials από τον χρήστη, δίνοντας του ταυτόχρονα τα απαραίτητα στοιχεία για τον υπολογισμό της υπογραφής. Τα στοιχεία αυτά συμπεριλαμβάνονται σε ένα WWW-Authenticate header του 401 response. Στη συνέχεια ο χρήστης υπολογίζει την «υπογραφή» και την συμπεριλαμβάνει σε ένα Authorization Header στο νέο REGISTER request που συντάσσει. Τελικά ο server κάνει αποδεκτό το request και απαντά με 200 OK response, του οποίου το Contact header δηλώνει την εγγραφή που δημιούργησε ο server για την τρέχουσα θέση του χρήστη.

Επιτυχημένη εγκατάσταση κλήσης

Στη παράγραφο αυτή παρουσιάζουμε την διαδικασία και την ανταλλαγή μηνυμάτων που απαιτούνται για την εγκατάσταση μιας VoIP κλήσης, με πιστοποίηση, μεταξύ δύο χρηστών (Bob και Alice), που ανήκουν στο ίδιο domain, του SIP Proxy. Όπως και στην περίπτωση του registration, αρχικά μετά την αποστολή του πρώτου INVITE request, ο SIP Proxy Server ζητά τα κατάλληλα διαπιστευτήρια από τον αποστολέα. Κάτι τέτοιο γνωστοποιείται στο χρήστη μέσω του 407 Proxy Authentication Required response που του αποστέλλει ο Server. Στη συνέχεια, ο χρήστης επανασυντάσσει το INVITE request, συμπεριλαμβάνοντας τα απαραίτητα credentials και το αποστέλλει ξανά, για να συνεχιστεί η διαδικασία εγκατάστασης της κλήσης. Το αίτημα προωθείται από τον SIP Proxy στο καλούμενο (εδώ Bob), ο οποίος απαντά στον Proxy με 180 Ringing. Το Ringing μεταδίδεται στον καλών (Alice), που απαντά με 200 OK, το οποίο ομοίως μεταδίδεται στο καλούμενο. Ο καλούμενος απαντά ACK, αποδέχεται τη κλήση, την απαντήση στέλνει ο Proxy πίσω στον καλών.



Σχήμα 3.2 Ανταλλαγή SIP μηνυμάτων για επιτυχημένη εγκατάσταση κλήση

Από αυτή τη στιγμή οι δύο συνομιλητές ξεκινούν απευθείας επικοινωνία μέσω του πρωτοκόλλου RTP. Η συνομιλία μπορεί να τερματιστεί μετά από ένα αίτημα BYE (από έναν εκ των δύο συνομιλητών) προς τον SIP Proxy. Ο Proxy προωθεί το μήνυμα στον άλλο συνομιλητή, και περιμένει την επιβεβαίωση του (ACK), την οποία προωθεί πίσω στον πρώτο και η διαδικασία ολοκληρώνεται.

Κεφάλαιο 4

Σχεδιασμός ολοκληρωμένου συστήματος επικοινωνίας

Ο σχεδιασμός και η ανάπτυξη ενός ολοκληρωμένου συστήματος επικοινωνίας πραγματικού χρόνου απαιτεί την αξιοποίηση πολλών διαφορετικών υποσυστημάτων, τα οποία καλύπτουν διαφορετικές λειτουργίες του συνολικού συστήματος. Για το κάθε υποσύστημα υπάρχουν πολλές επιλογές λογισμικού ανοικτού κώδικα από τις οποίες θα πρέπει να επιλεγούν οι καταλληλότερες και πλέον αξιόπιστες για τις υπηρεσίες που κάποιος επιθυμεί να αναπτύξει και αυτές που προσφέρουν την καλύτερη δυνατή διαλειτουργικότητα στο πλαίσιο της συνολικής λύσης.

4.1 Παρουσίαση τεχνολογιών αιχμής

Στο υπο-κεφάλαιο αυτό γίνεται μια σύντομη παρουσίαση των διαθέσιμων **τεχνολογιών αιχμής** (State-of-the-Art) και γίνεται μια επιλογή επιμέρους εφαρμογών ανοικτού λογισμικού που μαζί μπορούν να συγκροτήσουν μια σύγχρονη υπηρεσία επικοινωνίας και συνεργασίας, χωρίς περιορισμούς κλιμάκωσης (κατάλληλη από μια μικρή εταιρία μέχρι ένα πάροχο υπηρεσιών), με τον μέγιστο βαθμό διαλειτουργικότητας και ομοιογένειας της συνολικής λύσης, καθώς και ευκολία στην καθημερινή λειτουργία και την παρακολούθησή της.

Βασικό σύστημα διαχείρισης κλήσεων

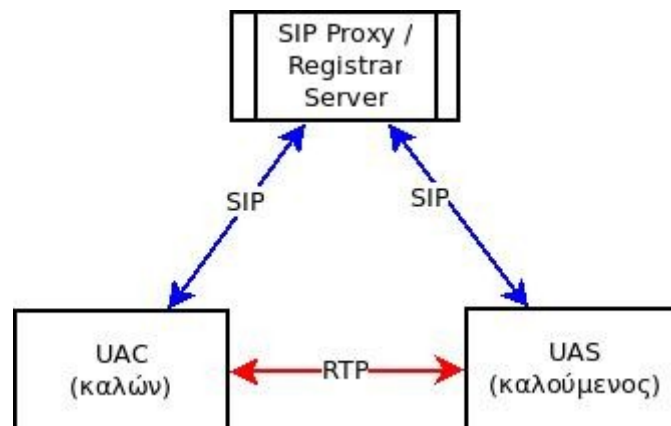
Ξεκινάμε από την παρουσίαση του υποσυστήματος που αποτελεί την καρδιά ενός τέτοιου περιβάλλοντος. Η πρώτη σημαντική απόφαση είναι η επιλογή μεταξύ ενός SIP Proxy / Registrar server ή ενός SIP Back-to-Back User Agent (B2BUA) server που θα αποτελέσουν το βασικό υποσύστημα διαχείρισης των κλήσεων της υποδομής. Οι βασικότερες διαφορές των δύο τεχνολογιών είναι στον τρόπο διαχείρισης της επικοινωνίας των χρηστών, τόσο σε επίπεδο σηματοδοσίας όσο σε επίπεδο ροής δεδομένων.

Ένας SIP Proxy φροντίζει για την επικοινωνία μεταξύ χρηστών ελέγχοντας την σηματοδοσία και προωθώντας την απαραίτητη πληροφορία από τον ένα χρήστη στον άλλο ώστε τελικά να εγκαθιδρυθεί μια κλήση, κάνοντας ελάχιστες μεταβολές και παρεμβάσεις στην επικοινωνία αυτή. Συνήθως, μετά την αρχική επικοινωνία και τη σύνδεση των χρηστών, ο SIP Proxy παραμένει στην διαδρομή της σηματοδοσίας για να έχει την πληροφόρηση της έναρξης και λήξης της επικοινωνίας, κυρίως για

λόγους καταγραφής, ενώ δεν εμπλέκεται στην μεταφορά των δεδομένων, π.χ. της φωνής, μεταξύ των χρηστών, η οποία γίνεται απευθείας. Για παράδειγμα, όταν ο χρήστης A καλεί, μέσω του Proxy, τον χρήστη B, η SIP σηματοδότηση περνά μέσω του Proxy αλλά η φωνή ταξιδεύει απευθείας από τον χρήστη A στον χρήστη B. Η τεχνική αυτή έχει τα εξής πλεονεκτήματα:

- Η λειτουργία ενός SIP Proxy απαιτεί τους λιγότερους δυνατούς υπολογιστικούς και δικτυακούς πόρους.
- Μπορεί να υποστηρίξει ένα πολύ μεγάλο αριθμό χρηστών και ταυτόχρονων συνδέσεων.
- Δεν γνωρίζει ή και δεν ελέγχει το περιεχόμενο των πακέτων που διακινούνται μεταξύ των χρηστών.
- Δεν εισάγει οποιαδήποτε καθυστέρηση ή απώλεια στην μετάδοση των δεδομένων.

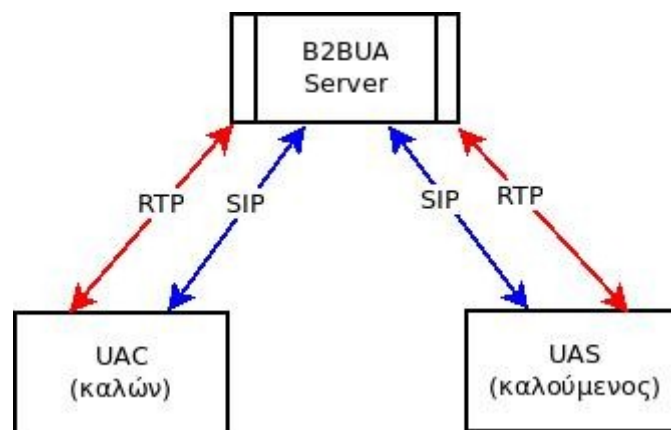
Τα παραπάνω πλεονεκτήματα συνηγορούν υπέρ της αξιοποίησης εικονικών μηχανών για την φιλοξενία τέτοιων συστημάτων.



Σχήμα 4.1 Επικοινωνία UAC-UAS μέσω ενός SIP Proxy

Το σημαντικότερο μειονέκτημα ενός Proxy είναι ότι δεν μπορεί να προσφέρει άμεση υποστήριξη σε υπηρεσίες που απαιτούν εξεργασία των δεδομένων μιας κλήσης, όπως π.χ. η συνδιάσκεψη, η εγγραφή, κ.ά. Για την υλοποίηση τέτοιων υπηρεσιών απαιτείται η παρουσία άλλων υποσυστημάτων, π.χ. το Asterisk είναι ένα από αυτά.

Αντίθετα ένας B2BUA Server λαμβάνει, ελέγχει και επαναπροωθεί όλη την επικοινωνία, σηματοδοσία και δεδομένα, από ένα χρήστη προς ένα άλλο χρήστη. Ο B2BUA προωθεί μόνο τη πληροφορία που πρέπει να περάσει στον χρήστη Β, αποκρύπτοντας π.χ. τα στοιχεία του χρήστη Α. Οι χρήστες γνωρίζουν ως μοναδικό σημείο επικοινωνίας τον B2BUA τόσο σε επίπεδο σηματοδοσίας όσο και σε επίπεδο δεδομένων. Σημαντικό πλεονέκτημα της τεχνικής αυτής είναι ότι μπορεί να υποστηρίξει εγγενώς τις υπηρεσίες που απαιτούν επεξεργασία των δεδομένων, όπως συνδιάσκεψη, εγγραφή, κ.ά., καθώς και χρήστες με διαφορετικά χαρακτηριστικά επικοινωνίας, τόσο σε επίπεδο σηματοδοσίας όσο και σε επίπεδο κωδικοποίησης των δεδομένων. Για παράδειγμα δύο χρήστες χρησιμοποιούν διαφορετικά πρωτόκολλα (SIP και H.323) και διαφορετική κωδικοποίηση (π.χ. G.711 και G.729) για την επικοινωνία τους με τον B2BUA, ο οποίος εφαρμόζει τεχνικές επανα-κωδικοποίησης ώστε να κάνει εφικτή την επικοινωνία.



Σχήμα 4.2 Επικοινωνία UAC-UAS μέσω ενός B2BUA

Σημαντικό μειονέκτημα της τεχνικής B2BUA είναι η απαίτηση για ισχυρούς υπολογιστικούς και δικτυακούς πόρους, γεγονός που περιορίζει σημαντικά τον μέγιστο αριθμό ταυτόχρονων επικοινωνιών. Επίσης η χρήση μιας εικονικής μηχανής για την φιλοξενία ενός τέτοιου συστήματος χρειάζεται μεγάλη προσοχή, τόσο στην επιλογή των χαρακτηριστικών της όσο και στα θέματα ασφάλειας.

Ενδεικτικά, μερικές διαθέσιμες επιλογές για ένα περιβάλλον B2BUA είναι οι παρακάτω:

- *FreeSwitch - B2BUA, με PBX δυνατότητες* (<http://www.freeswitch.org/>)

- SIPpy – B2BUA (<http://www.sippysoft.com/>)
- Asterisk – B2BUA, με PBX δυνατότητες (<http://www.asterisk.org/>)

Είναι αντιληπτό ότι δεν υπάρχει μια βέλτιστη λύση, που να περιλαμβάνει αξιοποίηση των πλεονεκτημάτων των δύο συστημάτων. Η χρήση SIP Proxy είναι όμως καταλληλότερη για μια υλοποίηση σε εικονικές μηχανές καθώς αποδεσμεύει την κίνηση δεδομένων (π.χ. της φωνής) από την εικονική υποδομή, η οποία παραμένει στο δίκτυο του οργανισμού, και παράλληλα προσφέρει όλα τα πλεονεκτήματα των υπηρεσιών νέφους: χαμηλό κόστος, υψηλή διαθεσιμότητα, ευκολία και γρήγορη κλιμάκωση. Οι δύο πιο διαδεδομένοι SIP Proxies είναι ο OpenSIPs (<http://www.opensips.org>) και ο Kamailio (<http://www.kamailio.org>). Έχουν σαν βάση τον OpenSER (<http://en.wikipedia.org/wiki/OpenSER>), ένα από τα πληρέστερα συστήματα ανοικτού κώδικα με πολλές εγκαταστάσεις παγκοσμίως σε μεγάλες και απαιτητικές υλοποιήσεις. Οι δύο υλοποιήσεις δεν έχουν ουσιαστικές διαφορές, για τους σκοπούς του παρόντος συστήματος επέλεξα τον OpenSIPs, λόγω καλύτερης τεκμηρίωσης και της μεγάλης και πιο ενεργής κοινότητας προγραμματιστών.

Media Services

Καλούμε Media Services αυτές τις υπηρεσίες που απαιτούν πρόσβαση και δυνατότητα επεξεργασίας των δεδομένων των κλήσεων, δηλ. της φωνής, του βίντεο. Τέτοιες υπηρεσίες είναι η συνδιάσκεψη (conference), το φωνητικό μήνυμα και το φαξ σε ηλεκτρονικό ταχυδρομείο (voice mail, fax to email), τα συστήματα φωνητικής αλληλεπίδρασης (IVR), η εγγραφή. Για να αντιμετωπίσουμε την αδυναμία υποστήριξης αυτών των υπηρεσιών από ένα SIP Proxy ενσωματώνουμε στο σύστημα μας ένα ή περισσότερους B2BUA servers.

Μια καλή επιλογή B2BUA server είναι αυτή του Asterisk. Η πλατφόρμα Asterisk είναι η πλέον ώριμη, διαθέτει πολύ μεγάλη εγκατεστημένη βάση, μεγάλη και ενεργή κοινότητα προγραμματιστών και πλούσιο υλικό τεκμηρίωσης στο διαδίκτυο. Το Asterisk διαθέτει μια αρκετά απλή δομή διαχείρισης κλήσεων και υλοποίησης των διάφορων media υπηρεσιών, που βασίζονται σε μια λογική τύπου Dial Plan. Το Asterisk διαθέτει επίσης ειδικά προγράμματα εκπαίδευσης και πιστοποίησης

μηχανικών ώστε να είναι δυνατή η υποστήριξη της λειτουργίας από τους μηχανικούς ενός οργανισμού.

4.2 Επιλογές κλιμάκωσης και εφεδρείας

Μια από τις βασικές σχεδιαστικές επιλογές ενός σύγχρονου ολοκληρωμένου συστήματος, που βασίζεται σε εικονικές υποδομές, θα πρέπει να είναι η υψηλή διαθεσιμότητα (μέσω εφεδρειών) και η δυνατότητα γρήγορης και εύκολης κλιμάκωσης, ώστε να εκμεταλλεύεται σωστά τη λογική του pay-as-you-go των υπηρεσιών υπολογιστικού νέφους.

Αυτό πρακτικά σημαίνει ότι για την φιλοξενία του συστήματος ξεκινάμε με μια αρχική εικονική υπολογιστική υποδομή, που να καλύπτει τις αρχικές ανάγκες μας, και στη συνέχεια είτε ενισχύουμε τα χαρακτηριστικά του εξυπηρετητή ή, προτιμότερο, εισάγουμε νέες εικονικές μηχανές. Για να είναι εφικτή η προσθήκη νέων υποσυστημάτων, με σκοπό την ενίσχυση της συνολικής ικανότητας του συστήματος, πρέπει αυτό να έχει προβλεφθεί στην αρχιτεκτονική του συστήματος. Για παράδειγμα, αν για τις υπηρεσίες συνδιάσκεψης έχουμε προβλέψει ένα σύστημα Asterisk, το οποίο ως υποθέσουμε ότι μπορεί να διεκπεραιώσει μέχρι 200 ταυτόχρονες συμμετοχές, τότε η αρχιτεκτονική του συστήματος πρέπει να είναι τέτοια ώστε να επιτρέπει την ομαλή ένταξη ενός δεύτερου συστήματος που θα μπορεί να διπλασιάσει αρχικό πλήθος των συμμετεχόντων. Επομένως ένα χαρακτηριστικό της αρχιτεκτονικής είναι η δυνατότητα κλιμάκωσης κάθε ενός από τα υποσυστήματά της.

Σημαντικό ρόλο στην δυνατότητα κλιμάκωσης του συστήματος έχει ο Proxy, ο οποίος αναλαμβάνει την ισορροπία και διαμοιρασμό του φόρτου (load balancing) των κλήσεων για τις πρόσθετες υπηρεσίες (π.χ. συνδιάσκεψης) μεταξύ των εξυπηρετητών. Όταν για παράδειγμα μια κλήση πρέπει να καταλήξει σε Voicemail, τότε ο Proxy ελέγχει τους διαθέσιμους πόρους και επιλέγει τον καταλληλότερο Voicemail Server για να εξυπηρετήσει την υπηρεσία. Με αυτό τον μηχανισμό, το μόνο που χρειάζεται κάθε φορά είναι η προσθήκη νέων εικονικών εξυπηρετητών και η ενημέρωση του Proxy για την ύπαρξή τους.

Είδαμε ότι η ανάγκη για κλιμάκωση των υπηρεσιών μέσου μπορεί να αντιμετωπιστεί ως μια εγγενής λειτουργία του Proxy. Πως όμως κλιμακώνεται ο ίδιος ο Proxy, ο οποίος διαχειρίζεται τα θέματα των εγγραφών (SIP registration), την εγκαθίδρυση των κλήσεων, την Παρουσία (Presence); Υπάρχουν δύο γνωστές επιλογές για την κλιμάκωση ενός Proxy, η χρήση multi-domain και η τεχνική DNS SRV, που παρουσιάζονται στη συνέχεια.

Η λύση multi-domain βασίζεται στην ιδέα των διαφορετικών domains, όπου κάθε domain είναι μια λογική οντότητα που περιλαμβάνει ένα Proxy και τους χρήστες που εγγράφονται σε αυτόν. Η δημιουργία των domains μπορεί να γίνει εξ αρχής ή αυτά να δημιουργούνται σταδιακά, σύμφωνα με τις ανάγκες. Ένα domain, συνήθως βασίζεται σε κάποια κοινά χαρακτηριστικά των χρηστών του, όπως η γεωγραφική τους θέση, η ιδιότητα τους, ο οργανισμός στον οποίο ανήκουν κλπ. Επιδίωξη μας είναι τα domains να έχουν περίπου το ίδιο μέγεθος, ώστε να υπάρχει ισοκατανομή του φόρτου στους Proxy. Περισσότερα από ένα domains μπορούν έχουν αρχικά τον ίδιο SIP Proxy και ο διαχωρισμός να γίνει σε μεταγενέστερο στάδιο. Είναι σημαντικό, όταν γίνει ένας τέτοιος ο διαχωρισμός, να μην προκύψει ανάγκη για κάποια αλλαγή του ονόματος του SIP Proxy στα συστήματα των χρηστών. Αυτό αποφεύγεται αν ο ίδιος Proxy διαθέτει πολλαπλά DNS ονόματα και οι χρήστες κάθε domain χρησιμοποιούν το όνομα Proxy που αντιστοιχεί στο domain τους, χωρίς βέβαια να γνωρίζουν ότι εξυπηρετούνται από το ίδιο σύστημα. Βέβαια ένα σημαντικό πλεονέκτημα είναι ότι τα domains μπορούν να φιλοξενηθούν πολύ εύκολα σε διαφορετικές εικονικές μηχανές, αυξάνοντας έτσι την ικανότητα του συστήματος.

Στην δεύτερη περίπτωση, της τεχνικής DNS SRV, όλοι οι διαθέσιμοι Proxy εξυπηρετητές διαχειρίζονται όλους τους χρήστες. Η επιλογή του εξυπηρετητή γίνεται με την χρήση DNS SRV Record, [3] RFC2782. Η ιδέα πίσω από την τεχνική αυτή είναι όμοια με αυτή των MX records για την υπηρεσία ηλεκτρονικού ταχυδρομείου. Για το email μια MX δήλωση σε ένα domain λέει που πρέπει να παραδοθεί ένα μήνυμα για ένα χρήστη του domain αυτού. Όμοια, για μια κλήση προς ένα χρήστη σε ένα domain, π.χ. user@domain.gr, το SRV record λέει στο χρήστη σε ποιο Proxy εξυπηρετητή να συνδεθεί. Επιπλέον, μπορούν να οριστούν πολλαπλοί Proxy εξυπηρετητές για το ίδιο domain, με διαφορετικές προτεραιότητες. Η τεχνική αυτή παρουσιάζεται με λίγο μεγαλύτερη λεπτομέρεια καθώς έχει καλύτερη εφαρμογή σε συστήματα μεγάλης κλίμακας.

Στην υλοποίηση αυτή, καταχωρούμε στον DNS Server του domain τους Proxy εξυπηρετητές, σύμφωνα με την ακόλουθη σύνταξη:

```
_Service._Proto.Name TTL Class Priority Weight Port Target
```

Ακολουθεί επεξήγηση των διάφορων πεδίων:

- Service: Το όνομα της υπηρεσίας, εδώ sip (πλήρης κατάλογος εδώ: <http://www.ietf.org/assignments/service-names-port-numbers/service-names-port-numbers.txt>)
- Proto: Το πρωτόκολλο της υπηρεσίας, π.χ. TCP ή UDP
- Name: Το domain στο οποίο αναφέρεται η εγγραφή, π.χ. Domain.gr
- TTL: ένας ακέραιος 32 bit που προσδιορίζει το χρονικό διάστημα που η εγγραφή μπορεί να είναι έγκυρη μετά από μια ανάγνωση. Μηδενική τιμή σημαίνει ότι χρειάζεται νέα ανάγνωση κάθε φορά, δηλ. δεν μπορεί να αποθηκευτεί (cached). Το πεδίο είναι προαιρετικό και η απουσία δηλώνει τιμή μηδέν.
- Class: Έχει DNS σημασία, εγγραφές αυτού του είδους (SRV records) ανήκουν στην κλάση IN SRV.
- Priority: Η προτεραιότητα του συγκεκριμένου εξυπηρετητή. Ένας πελάτης πρέπει να προσπαθεί να συνδεθεί στον εξυπηρετητή με την **μικρότερη** τιμή. Εξυπηρετητές με την ίδια προτεραιότητα, θα δοκιμαστούν με την σειρά που καθορίζεται στην τιμή του πεδίου weight που ακολουθεί. Παίρνει τιμές στο διάστημα [0, 65535].
- Weight: Καθορίζει το βάρος της επιλογής μεταξύ εξυπηρετητών με την **ίδια** προτεραιότητα. Παίρνει τιμές στο διάστημα [0, 65535]. Μεγαλύτερη τιμή σημαίνει, αναλογικά, μεγαλύτερη πιθανότητα επιλογής του εξυπηρετητή.

- Port: Ο αριθμός της πόρτας της υπηρεσίας στην οποία ακούει ο εξυπηρετητής. Παίρνει τιμές στο διάστημα [0, 65535]. Για παράδειγμα 5060 για SIP.
- Target: Το όνομα του εξυπηρετητή. Η παρουσία μίας τελείας, ".", υποδηλώνει ότι η αναφερόμενη υπηρεσία δεν προσφέρεται σε αυτό το domain.

Ακολουθεί ένα πολύ απλό παράδειγμα στο οποίο δηλώνεται η υπηρεσία sip στο domain mydomain.gr με δύο εξυπηρετητές με την ίδια προτεραιότητα (μηδέν) και το ίδιο weight (μηδέν), που σημαίνει ότι ο πελάτης επιλέγει με την ίδια πιθανότητα τον ένα ή τον άλλο εξυπηρετητή:

```
_sip._udp.mydomain.gr IN SRV 0 0 5060 mysipserver1.mydomain.gr.  
_sip._udp.mydomain.gr IN SRV 0 0 5060 mysipserver2.mydomain.gr.
```

Ολοκληρώνοντας θα επιχειρήσουμε μια σύντομη σύγκριση των δύο μεθόδων κλιμάκωσης ενός SIP Proxy, των multi-domain και DNS SRV. Η πρώτη μέθοδος είναι απλούστερη και ξεκάθαρη από πλευράς υλοποίησης και συμβατή με όλες τις εφαρμογές πελάτες. Απαιτείται όμως οι χρήστες να γνωρίζουν τόσο το domain στο οποίο ανήκουν όσο και το domain αυτών που θέλουν να καλέσουν. Επίσης η κλιμάκωση είναι πιο “χειρωνακτική” και απαιτεί αρκετές αλλαγές στα συστήματα, που ενδεχομένως να επηρεάσουν και την λειτουργία των πελατών.

Η δεύτερη λύση (DNS SRV records) είναι πιο σύνθετη στην υλοποίηση και απαιτεί οι πελάτες (soft / hardware phones, ATA κ.τ.λ.) να υποστηρίζουν DNS SRV Records. Όμως, κλιμακώνεται πολύ πιο εύκολα και οι πελάτες χρειάζεται να γνωρίζουν ένα και μοναδικό domain, αυτό της υπηρεσίας στην οποία ανήκουν.

Κεντρικό υποσύστημα μνήμης (memory caching)

Σε μια υποδομή πολλαπλών συστημάτων SIP Proxy είναι αναγκαίο, για λόγους απόδοσης, ένα σχήμα διαχείρισης κοινόχρηστων πληροφοριών για όλα τα Proxy υποσυστήματα. Για παράδειγμα, όταν ένας χρήστης εγγράφεται σε ένα εκ των διαθέσιμων Proxy (registered) καθίσταται άμεσα διαθέσιμος σε όλους τους υπόλοιπους. Ένα τέτοιο περιβάλλον εξασφαλίζει υψηλή απόδοση και επιτρέπει τη

δημιουργία ενός περιβάλλοντος πολύ μεγάλου αριθμού χρηστών, θεωρητικά απεριόριστο.

Μια τεχνική για την αντιμετώπιση του ζητήματος αυτού, που παράλληλα βελτιστοποιεί τους χρόνους αναζήτησης, είναι η εγκατάσταση ενός συστήματος διαχείρισης δεδομένων που βρίσκονται στην μνήμη ενός εξυπηρετητή και γίνονται διαθέσιμα σε διάφορους πελάτες (Memory Cache Server), εδώ οι πελάτες είναι οι εξυπηρετητές Proxy. Ορισμένες από τις διαθέσιμες επιλογές είναι:

- Cassandra - <http://cassandra.apache.org/>
- Redis - <http://www.redis.io/>
- MemCached - <http://www.memcached.org/>

Η λύση MemCached έχει πολύ απλή υλοποίηση και ταιριάζει καλύτερα στο περιβάλλον των υπηρεσιών. Ένας MemCached server (π.χ. σε μια εικονική μηχανή) διαθέτει την μνήμη (RAM) του σε άλλα συστήματα MemCached clients. Οι clients αποθηκεύουν ζευγάρια κλειδιών - τιμών (key-value) στο server τα οποία αργότερα ανακτούν. Οι πληροφορίες είναι διαθέσιμες σε όλους τους clients.

Η ιδέα είναι ότι ένας SIP Proxy, πρώτα εξετάζει αν η πληροφορία που θέλει βρίσκεται στη cache. Αν όχι, ανακτά την πληροφορία από τη Βάση Δεδομένων και, παράλληλα, την αποθηκεύει στη cache για μελλοντική χρήση (από τον ίδιο ή άλλο Proxy).

Σημειώστε ότι, καθώς οι πληροφορίες που τοποθετούνται στη MemCached βρίσκονται αποθηκευμένες σε μια Βάση Δεδομένων, μια διακοπή της υπηρεσίας MemCached δεν προκαλεί διακοπή των υπηρεσιών αλλά μειωμένη απόδοση. Επίσης μπορούν να χρησιμοποιηθούν πολλοί servers, έτσι ώστε η μνήμη είναι σχεδόν απεριόριστη. Τέλος μια επανεκκίνηση του SIP Proxy δεν θα επηρεάσει την cache, τα δεδομένα εξακολουθούν να είναι διαθέσιμα (εφόσον δεν έχουν χρονικό όριο λήξης).

Βάσεις Δεδομένων

Η Βάση Δεδομένων αποτελεί το σημείο αποθήκευσης των στοιχείων των χρηστών, των πληροφοριών, των στοιχείων επικοινωνίας και της παραμετροποίησης των υπηρεσιών και αποτελεί ένα απαραίτητο υποσύστημα. Οι δύο επικρατέστερες επιλογές ΒΔ ανοικτού λογισμικού είναι η MySQL (<http://www.mysql.com>) και η PostgreSQL (<http://www.postgresql.org>).

Για να αντιμετωπίσει το ζήτημα της διαθεσιμότητας της Βάσης Δεδομένων ο openSIPS υποστηρίζει την έννοια της εικονικής βάσης. Η ιδέα είναι ότι στον openSIPS ορίζεται ένας εικονικός δείκτης σε μια ΒΔ, που ουσιαστικά αντιστοιχεί σε πολλές πραγματικές ΒΔ. Αυτό το εικονικό στρώμα επιτρέπει να χρησιμοποιηθούν οι πραγματικές ΒΔ με διάφορους τρόπους, όπως: Παράλληλα, failover (hotswap), round-robin. Έτσι λειτουργίες όπως insert, delete, update, replace μπορούν να γίνονται παράλληλα σε όλες τις πραγματικές ΒΔ ενώ η λειτουργία query με τεχνική round-robin. Με αυτό το τρόπο εξασφαλίζεται η διαθεσιμότητα εφόσον είναι σε λειτουργία μία τουλάχιστον πραγματική ΒΔ.

Σημειώστε ότι η τεχνική εικονικού δείκτη ΒΔ δίνει λύση στην εφεδρεία / διαθεσιμότητα των ΒΔ που χρησιμοποιεί ο SIP Proxy/Registrar, δηλ. ο openSIPS. Η λύση εικονικού δείκτη ΒΔ **δεν** έχει εφαρμογή στα συστήματα Asterisk, που μπορεί να χρησιμοποιούν την ίδια ΒΔ!

Η προτεινόμενη προσέγγιση, που αντιμετωπίζει συνολικά το ζήτημα της εφεδρείας αλλά και της κλιμάκωσης των Βάσεων Δεδομένων, είναι η χρήση της τεχνολογίας MySQL Cluster (έκδοση open source). Για την ελάχιστη υλοποίηση απαιτούνται τουλάχιστον 4 συστήματα: 2 data nodes και 2 SQL/NoSQL Application Nodes & Management Nodes .

4.3 Αρχιτεκτονική συστήματος

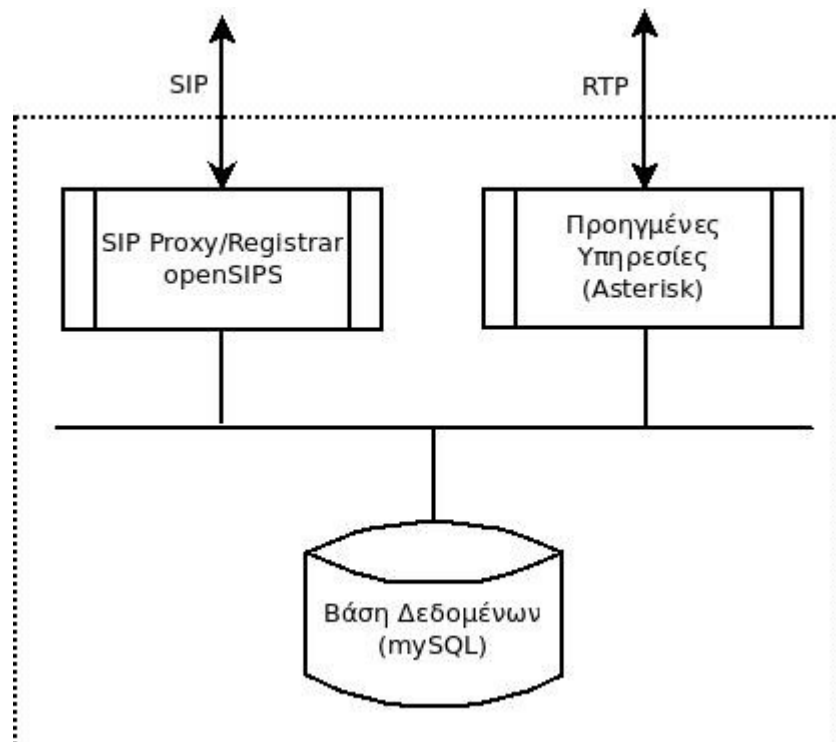
Η πιο απλή αρχιτεκτονική του προτεινόμενου συστήματος επικοινωνίας πραγματικού χρόνου αποτελείται από ακόλουθα τρία βασικά υποσυστήματα:

- ένα SIP Proxy/Registrar, τον openSIPS, για την διαχείριση των κλήσεων,

- ένα B2BUA, το Asterisk, για προηγμένες υπηρεσίες όπως audio conference, audio streaming, voice mail, IVR, fax κλπ,
- μια Βάση Δεδομένων, η MySQL, για τις ανάγκες του SIP registrar, την αποθήκευση των στοιχείων των χρηστών, το σύστημα καταγραφής κλήσεων - cdrs, στοιχεία για τις ανάγκες των προηγμένων υπηρεσιών, κλπ.

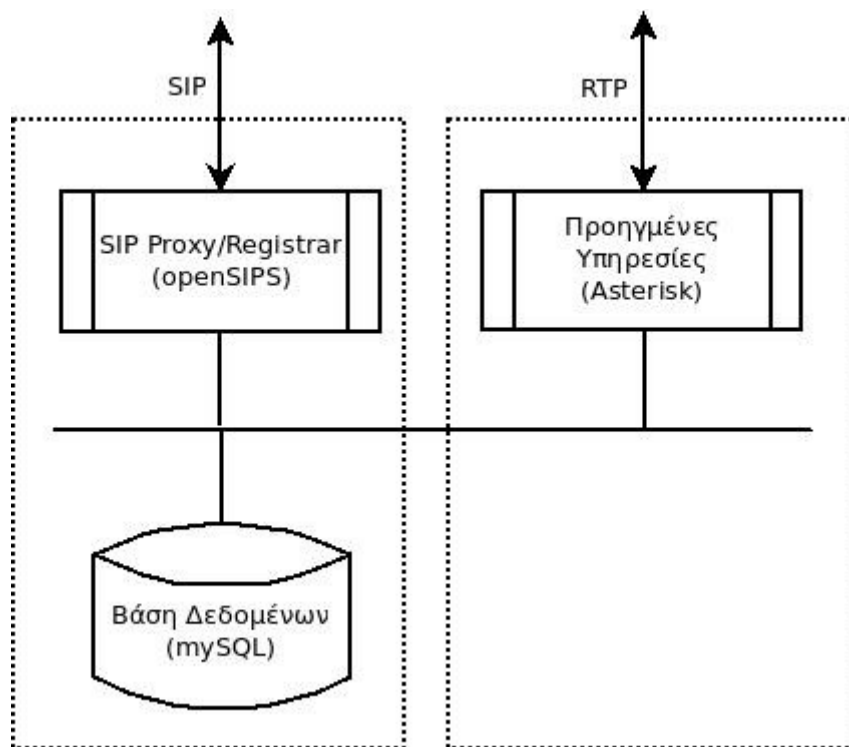
Στο ακόλουθο σχήμα, Σχήμα 4.3, αποτυπώνεται αυτή η βασική αρχιτεκτονική, όπου τα τρία υποσυστήματα είναι δυνατόν να φιλοξενηθούν στην ίδια εικονική μηχανή. Παρατηρήστε πως γίνεται η επικοινωνία το συστήματος προς τον υπόλοιπο κόσμο:

- το σύστημα openSIPS, μόνο μέσω πρωτοκόλλου SIP, για την διαχείριση της σηματοδοσίας,
- το σύστημα Asterisk, μόνο μέσω του πρωτοκόλλου RTP, για τις ανάγκες των προηγμένων υπηρεσιών. Βεβαίως υπάρχει επίσης εσωτερική επικοινωνία του Asterisk με το σύστημα openSIPS μέσω πρωτοκόλλου SIP για την διαχείριση της σηματοδοσίας.



Σχήμα 4.3 Βασική αρχιτεκτονική συστήματος επικοινωνίας πραγματικού χρόνου

Το επόμενο βήμα είναι μια πιο σύνθετη αρχιτεκτονική όπου οι αυξημένες απαιτήσεις των προηγμένων υπηρεσιών σε υπολογιστικούς πόρους οδηγούν στην ανάγκη ενσωμάτωσης μιας πρόσθετης εικονικής μηχανής για την φιλοξενία της εφαρμογής Asterisk, δείτε Σχήμα 4.4. Όπως θα δούμε στα αποτελέσματα των μετρήσεων υπερφόρτωσης του συστήματος, οι υπηρεσίες αυτές έχουν πολύ αυξημένες απαιτήσεις σε υπολογιστική ισχύ και επομένως μπορούν να υποστηρίξουν πολύ περιορισμένο αριθμό ταυτόχρονων κλήσεων.



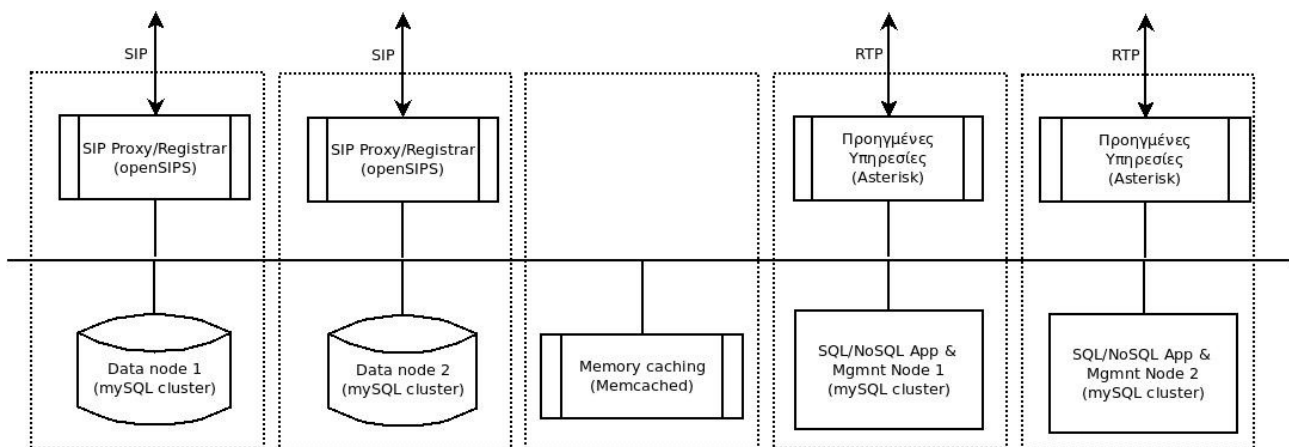
Σχήμα 4.4 Βασική αρχιτεκτονική συστήματος με δύο εικονικές μηχανές

Είναι αντιληπτό ότι πλέον μπορούμε να προσθέτουμε στο σύστημα μας εικονικές μηχανές με Προηγμένες Υπηρεσίες ώστε να σταδιακά να καλύπτουμε τις απαιτήσεις όποτε αυτές δημιουργούνται. Η αντιγραφή μιας εικονικής μηχανής για την δημιουργία μιας δεύτερης είναι υπόθεση τόσο απλή όσο η αντιγραφή ενός αρχείου με μια εντολή!

Το σύστημα openSIPS έχει ενσωματωμένο μηχανισμό ισοκατανομής φόρτου (load balancing) σε επίπεδο διαφορετικών συστημάτων. Με την προσθήκη ενός δεύτερου συστήματος Asterisk, ή παραπάνω, επιτυγχάνουμε επίσης την υψηλή διαθεσιμότητα

των υπηρεσιών, καθώς η διακοπή λειτουργίας ενός εκ των συστημάτων δεν διακόπτει την παροχή των υπηρεσιών.

Στη συνέχεια παρουσιάζουμε μια κλιμακώσιμη αρχιτεκτονική με εφεδρείες και υψηλή διαθεσιμότητα σε όλα τα επίπεδα. Η αρχιτεκτονική βασίζεται στις ιδέες που συζητήσαμε ήδη στη προηγούμενη υπο-ενότητα για τον τρόπο κλιμάκωσης του SIP Proxy, που βασίζεται στην χρήση των τεχνικών memory caching και MySQL cluster για την εσωτερική λειτουργία καθώς και σε DNS SRV records για την επέξω πρόσβαση. Η αρχιτεκτονική αυτή παρουσιάζεται στο ακόλουθο σχήμα, Σχήμα 3. Απαιτούνται τουλάχιστον 5 εικονικές μηχανές, οι 4 από τις οποίες είναι υποχρεωτικές για την διάταξη MySQL cluster.



Σχήμα 4.5 Κλιμακώσιμη αρχιτεκτονική συστήματος με υψηλή διαθεσιμότητα

Με την παραπάνω αρχιτεκτονική οποιοδήποτε από τα τρία κεντρικά υποσυστήματα μπορεί να επεκταθεί, αυξάνοντας ταυτόχρονα την απόδοση του συστήματος, με απλή προσθήκη μιας νέας αντίστοιχης εικονικής μηχανής:

1. Προσθήκη ενός SIP Proxy (openSIPS)

- ο εντάσσεται στο σχήμα DNS SRV records για να δέχεται αιτήματα χρηστών και να διαχειρίζεται κλήσεις
- ο συμμετέχει με client στο σχήμα memory caching
- ο χρησιμοποιεί τη κοινή Βάση Δεδομένων (cluster)

- γνωρίζει και αξιοποιεί τα συστήματα Προηγμένων Υπηρεσιών (Asterisk)
2. Προσθήκη ενός συστήματος για Προηγμένες Υπηρεσίες (Asterisk)
 - εντάσσεται στο σχήμα διαμοιρασμού του φόρτου (load balancing) των κλήσεων των openSIPS, άρα δέχεται κλήσεις από αυτούς
 - χρησιμοποιεί τη κοινή Βάση Δεδομένων (cluster)
 3. Προσθήκη ενός συστήματος data node στη Βάση Δεδομένων.

Υπενθυμίζουμε ότι οι πληροφορίες της MemCached είναι αποθηκευμένες στη Βάση Δεδομένων και μια διακοπή της λειτουργίας memory caching ισοδυναμεί με συνεχόμενες αστοχίες, άρα μείωση της απόδοσης και όχι διακοπή μιας υπηρεσίας. Επίσης υπενθυμίζουμε ότι μπορούν να χρησιμοποιηθούν πολλοί memory caching servers για περισσότερη μνήμη.

4.4 Θέματα ασφάλειας πιστοποίησης

Μια ανησυχία, πολλών διαχειριστών, κυρίως όταν τα συστήματά τους βρίσκονται εκτός του δικτύου τους, είναι να αποφευχθεί η χρήση clear text password στη κανονική λειτουργία πιστοποίησης των χρηστών, άρα η μεταφορά τέτοιας πληροφορίας στο δίκτυο. Το μοντέλο πιστοποίησης (authentication) που χρησιμοποιεί το SIP βασίζεται στο HTTP digest authentication, όπως περιγράφεται RFC 2617, και αντιμετωπίζει ικανοποιητικά το ζήτημα αφού μεταφέρει κρυπτογραφημένα τα στοιχεία του χρήστη στο δίκτυο.

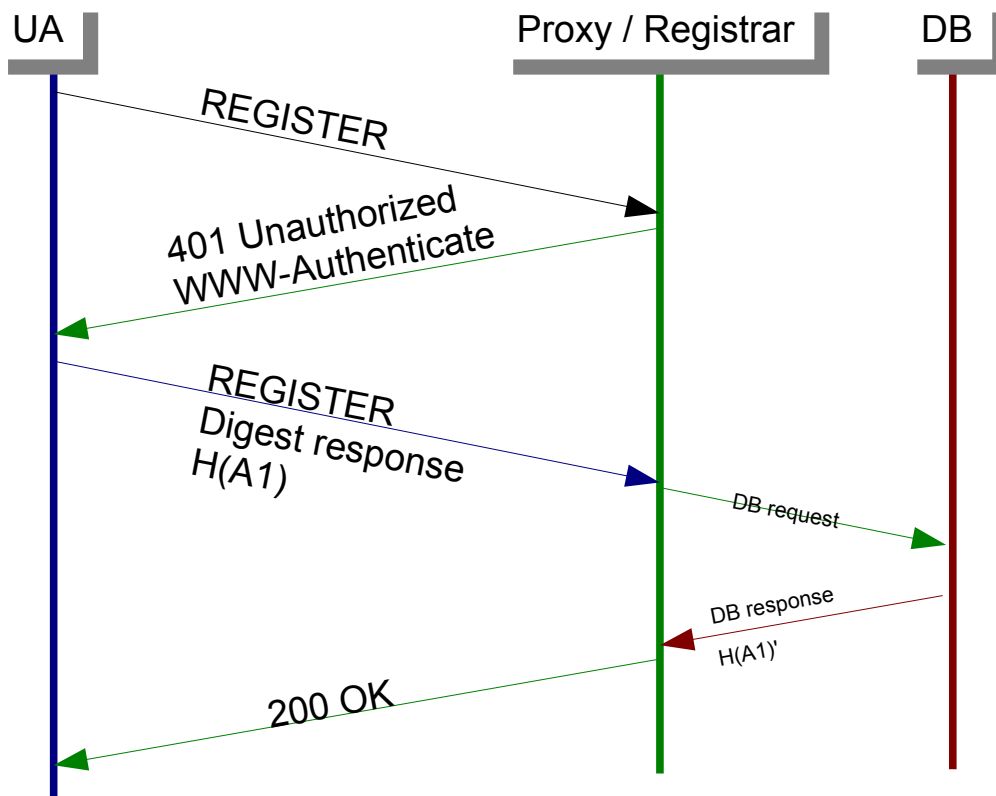
Για την καλύτερη κατανόηση των διάφορων ζητημάτων που προκύπτουν από την εφαρμογή αυτού του μοντέλου, παρουσιάζεται στη συνέχεια η ροή μηνυμάτων μιας επιτυχημένης διαδικασίας καταχώρησης ενός SIP πελάτη σε ένα SIP εξυπηρετητή:

1. Ο SIP πελάτης (UAC) στέλνει αίτημα καταχώρησης, REGISTER, στο SIP Proxy εξυπηρετητή.

2. Καθώς με το αίτημα δεν παρέχονται τα κατάλληλα διαπιστευτήρια, ο εξυπηρετητής το απορρίπτει και απαντά (challenge) με ένα μήνυμα «401 Unauthorized». Το μήνυμα αυτό του εξυπηρετητή περιέχει μια κεφαλίδα WWW-Authenticate που, μεταξύ άλλων, περιλαμβάνει το σχήμα πιστοποίησης που θα ακολουθηθεί (δηλ. Digest) καθώς και άλλα σχετικά πεδία όπως το realm (π.χ. sch.gr) και το nonce. Προαιρετικά ο εξυπηρετητής μπορεί να ορίσει τον αλγόριθμο που θα χρησιμοποιηθεί για την κρυπτογράφηση της απάντησης. Πρακτικά η επιλογή αυτή δεν χρησιμοποιείται, αλλά επιλέγεται ο εξ' ορισμού (default) αλγόριθμος MD5.
3. Ο SIP πελάτης απαντά με ένα νέο αίτημα καταχώρησης, REGISTER, μόνο που τώρα περιλαμβάνει μια κεφαλίδα WWW-Authenticate με τα διαπιστευτήρια του. Η απάντηση αυτή περιέχει, με μια συγκεκριμένη συμφωνημένη δομή, το MD5 checksum των τιμών: username, password, nonce value, HTTP method και URI. Στη δομή αυτή σημαντικό ρόλο έχει το πεδίο H(A1), το οποίο υπολογίζεται ως MD5(username ":" realm ":" password).
4. Ο εξυπηρετητής λαμβάνει την Digest απάντηση (response) του πελάτη και την συγκρίνει με μια αντίστοιχη H(A1)' που ο ίδιος δημιουργεί ή ανακτά από μια Βάση Δεδομένων. Αν η τιμή είναι ίδια (H(A1)=H(A1)') απαντά με 200 OK, αλλιώς ζητά επανάληψη της διαδικασίας

Στην ανωτέρω διαδικασία έχει αναδειχθεί η ανάγκη αποθήκευσης της τιμής H(A1), π.χ. σε μια τοπική Βάση Δεδομένων ή ένα LDAP, ώστε να αποφευχθεί η χρήση clear text password στη κανονική λειτουργία, άρα η μεταφορά τέτοιας πληροφορίας στο δίκτυο. Ο υπολογισμός της τιμής H(A1) γίνεται μια φορά κατά την δημιουργία του λογαριασμού χρήστη ή όποτε γίνεται αλλαγή του κωδικού (password) ως ακολούθως:

$H(A1) = MD5(\text{username}:"\text{realm}:"\text{clear_text_password})$



Σχήμα 4.6 Διαδικασία digest authentication

Για όλα τα αιτήματα που φτάνουν στον εξυπηρετητή και απαιτούν πιστοποίηση, όπως π.χ. η καταχώρηση (REGISTER), ο εξυπηρετητής ζητά την τιμή του πεδίου H(A1), ώστε να την συγκρίνει την τιμή που έλαβε από τον πελάτη. Για λόγους ασφάλειας πολλές άλλες λειτουργίες, εκτός από την καταχώρηση (REGISTER), όπως η πραγματοποίηση μιας κλήσης (INVITE), μπορεί να απαιτούν πιστοποίηση του πελάτη απέναντι στον εξυπηρετητή. Για να περιοριστεί ο αριθμός των αιτημάτων αλλά και για να αυξηθεί η απόδοση του συστήματος, μπορεί να υλοποιηθεί μια τεχνική ενδιάμεσης αποθήκευσης των απαντήσεων στη μνήμη (memory cached) του συστήματος.

Κεφάλαιο 5

Θέματα απόδοσης και ποιότητας

5.1 Θέματα εικονοποίησης και εφαρμογών πραγματικού χρόνου

Η πλειονότητα των κρίσιμων επιχειρηματικών εφαρμογών, όλες οι web και οι Βάσεις Δεδομένων, έχουν μεταφερθεί με επιτυχία στα εικονικά περιβάλλοντα, αποδεικνύοντας τα ωφελήματα της εικονοποίησης (virtualization) για εξοικονόμηση και εξορθολογισμό της διαχείρισης των τεχνολογιών πληροφορικής.

Παρ' όλη τη πρόοδο στο τομέα αυτό, η υιοθέτηση της εικονοποίησης στην περιοχή εφαρμογών πραγματικού χρόνου, με ευαισθησία στη καθυστέρηση, φαίνεται ότι κινείται με πιο αργούς ρυθμούς. Οι υπηρεσίες VoIP είναι το κλασσικό παράδειγμα αυτών των εφαρμογών και μέσω του περιβάλλοντος που αναπτύξαμε διερευνήσαμε κατά πόσο ευσταθεί αυτή η ανησυχία.

Οι εφαρμογές VoIP χαρακτηρίζονται από μια ιδιαίτερη ευαισθησία στην καθυστέρηση, υποχρεώνοντας τα πακέτα φωνής να παραδίδονται σε τακτά χρονικά διαστήματα με πολύ μικρή απόκλιση ώστε να υπάρχει καλή ποιότητα φωνής, διαφορετικά μια ακανόνιστη ροή οδηγεί σε απόρριψη πακέτων με αποτέλεσμα να έχουμε κακή εμπειρία ποιότητας στο τελικό χρήστη.

Επομένως ο σωστός χρονισμός για την επεξεργασία και την έγκαιρη παράδοση των δεδομένων ήχου είναι πολύ καθοριστικός παράγοντας για τις υπηρεσίες VoIP. Ωστόσο σε ένα εικονικό (virtualized) περιβάλλον ενδέχεται η ικανοποίηση αυτής της απαίτησης για τις εφαρμογές VoIP να είναι μια μεγαλύτερη πρόκληση λόγω του επιπλέον στρώματος χρονοπρογραμματισμού των εικονικών μηχανών (VMs) και επεξεργασίας των πακέτων δικτύου.

Είναι φανερό ότι οι εικονικές μηχανές ανταγωνίζονται για τους διαθέσιμους φυσικούς πόρους και το στρώμα εικονοποίησης καλείται να ικανοποιήσει τις διάφορες ανάγκες με ένα δίκαιο τρόπο. Μια λανθασμένη διαστασιολόγηση του πλήθους των εικονικών μηχανών σε μια φυσική υποδομή ή γειτονικές εικονικές μηχανές με αυξημένες απαιτήσεις μπορεί να έχουν ως αποτέλεσμα μείωση των διαθέσιμων πόρων. Υπό αυτή την έννοια μια εικονική μηχανή υστερεί σε απόδοση από μια φυσική μηχανή με τα ίδια χαρακτηριστικά.

Σε ένα ιδιόκτητο περιβάλλον εικονοποίησης, το πλήθος και το είδος των εικονικών μηχανών που ενεργοποιούνται σε μια φυσική υποδομή είναι ελεγχόμενο, όπως ελεγχόμενο είναι το δίκτυο εντός του οργανισμού. Η ανάπτυξη τέτοιων υπηρεσιών σε ένα private datacenter, που φιλοξενείται εντός ενός οργανισμού, δεν αναμένεται να παρουσιάσει κάποια δυσκολία.

Το ενδιαφέρον, και αυτό που εξετάζεται εδώ, είναι η αξιοποίηση εικονικών μηχανών από ένα public datacenter όπου δεν υπάρχει καμία γνώση και έλεγχος των φυσικών υποδομών. Όταν κανείς προμηθεύεται μια εικονική μηχανή από ένα public datacenter επιλέγει συνήθως χαρακτηριστικά όπως ο αριθμός των εικονικών επεξεργαστών (vCPUs), η μνήμη (RAM), το μέγεθος του εικονικού δίσκου, το εύρος του δικτύου. Σε καμία περίπτωση δεν γνωρίζει κάτι για το πλήθος, το είδος και τις ανάγκες των γειτονικών¹ εικονικών μηχανών. Επίσης, μπορεί να καθορίσει την πολιτική ασφάλειας για την πρόσβαση στη μηχανή του. Συνήθως, εξ αρχής επιτρέπεται μόνο πρόσβαση για διαχείριση (SSH), όλες οι άλλες θύρες προς την εικονική μηχανή είναι κλειστές και ο ιδιοκτήτης με αίτημα ή μέσω web εφαρμογής ανοίγει τις πόρτες για τις επιθυμητές υπηρεσίες. Βεβαίως, στην ίδια την εικονική μηχανή μπορεί να εφαρμόσει επιπλέον μια οποιαδήποτε πολιτική ασφάλειας (π.χ. μέσω iptables), δείτε παράδειγμα iptables στο παράρτημα.

Ένα άλλο σημαντικό στοιχείο, εκτός από την σταθερότητα και απόδοση της εικονικής μηχανής, είναι τα ζητήματα δικτύου. Όσο ελεγχόμενο είναι το δίκτυο εντός ενός οργανισμού, τόσο απρόβλεπτο μπορεί να γίνει το δίκτυο από την εικονική μηχανή μέχρι το τελικό χρήστη. Το δίκτυο αυτό συνήθως διέρχεται μέσω πολλών παρόχων, διαφορετικών επιπέδων (isps, carriers,).

5.2 Θέματα ποιότητας στην τηλεφωνία με πακέτα

Υπάρχουν τρία μεγάλα θέματα ποιότητας στην υπηρεσία φωνής, όταν αυτή μεταδίδεται σε πακέτα (ποιότητα, καθυστέρηση και ηχώ):

1. **Ποιότητα φωνής.** Η ποιότητα φωνής που αντιλαμβάνεται ένας χρήστης σε μια συνομιλία είναι ένα γινόμενο της τεχνικής κωδικοποίησης που

¹ Ως γειτονικές εικονικές μηχανή εννοούμε τις μηχανές που φιλοξενούνται στην ίδια φυσική μηχανή, άρα μοιράζονται τους ίδιους υπολογιστικούς πόρους.

χρησιμοποιείται και του ποσοστού των πακέτων που αποτυγχάνουν να φτάσουν στον αποδέκτη για να αποκωδικοποιηθούν. Πακέτα μπορούν να χαθούν με δύο τρόπους σε ένα δίκτυο VoIP:

- Το δίκτυο απορρίπτει πακέτα λόγω λαθών ή υπερχειλίσεων.
- Ο αποδέκτης απορρίπτει πακέτα λόγω καθυστέρησης, μεγαλύτερης από ότι μπορεί να διαχειριστεί ο jitter buffer.

2. **Η καθυστέρηση διάδοσης.** Η καθυστέρηση διάδοσης είναι ο χρόνος που απαιτείται για να διασχίσει ένα πακέτο το δίκτυο (mouth-to-ear delay). Διάφοροι παράγοντες συμβάλουν στην καθυστέρηση διάδοσης, όπως για παράδειγμα η δικτυακή απόσταση μεταξύ των συνομιλητών, το πλήθος και οι διαδικασίες αποθήκευσης - επεξεργασίας στους ενδιάμεσους δρομολογητές. Το σημαντικό στοιχείο εδώ είναι ότι όταν η καθυστέρηση ξεπερνά τα 150 ms αρχίζουν προβλήματα διατήρησης του επιπέδου της συνομιλίας.

3. **Έλεγχος ηχούς.** Όλα τα τηλεφωνικά κυκλώματα εισάγουν ηχώ. Όταν όμως ο χρόνος διαμετακόμισης (round trip) ξεπερνά τα 50 ms η ηχώ αρχίζει να γίνεται αισθητή και σε μεγαλύτερη καθυστέρηση δημιουργεί προβλήματα. Σε αυτή τη περίπτωση απαιτείται η εφαρμογή τεχνικών για την αφαίρεση της. Η πιο διαδεδομένη τεχνική, echo cancellation, χρησιμοποιεί μια μαθηματική προσέγγιση που αφαιρεί τμήμα του μεταδιδόμενου σήματος από το σήμα επιστροφής για να εξαλειφθεί η ηχώ.

Ποιότητα φωνής και jitter (τρέμουλο)

Όταν ένας VoIP media server αποστέλλει μια συνεχόμενη ροή φωνής τα δεδομένα ήχου επεξεργάζονται περιοδικά και στέλλονται πάνω από το δίκτυο σε μορφή διακριτών πακέτων. Τα πακέτα αυτά στη συνέχεια λαμβάνονται από τον χρήστη, σε τακτά χρονικά διαστήματα. Κατά τη διαδικασία αυτή, καθυστερημένα πακέτα μπορεί να χρειαστεί να απορριφθούν έτσι ώστε να μην διαταραχθεί η πραγματικού χρόνου αναπαραγωγή. Σε μια τέτοια περίπτωση τίθεται σε κίνδυνο η ποιότητα φωνής. Για να μετρηθεί σωστά η ακρίβεια του χρόνου άφιξης των πακέτων, ορίστηκε η παράμετρος jitter που εκφράζει τη στατιστική διακύμανση των χρόνων άφιξης των πακέτων. Πιο

συγκεκριμένα, το jitter ορίζεται ως η απόλυτη διαφορά ανάμεσα στην αναμενόμενη (συνήθως 20ms για τυπικά VoIP συστήματα) και την πραγματική ώρα άφιξης μεταξύ των πακέτων.

Πηγές του jitter

Το jitter γενικά προκαλείται από συμφόρηση στο δίκτυο IP. Η συμφόρηση μπορεί να συμβεί είτε στις διεπαφές του δρομολογητή ή στα κυκλώματα του δικτύου ενός παρόχου. Επίσης jitter μπορεί μερικές φορές να προκύπτει κατά την αποστολή. Αυτό συμβαίνει όταν τα δεδομένα ήχου δεν προέρχονται από μια αξιόπιστη πηγή (π.χ. την κάρτα ήχου που έχει ένα πολύ σταθερό ρολόι, πιο ακριβές από ότι του συστήματος) αλλά από μια άλλη πηγή όπως για παράδειγμα η ροή ήχου που παράγεται από ένα κείμενο (text-to-speech λογισμικό) ή από ένα αρχείο, με άλλα λόγια, μιλάμε για εφαρμογές όπως φωνητικό ταχυδρομείο ή συστήματα διαδραστικής φωνητικής απόκρισης (IVR). Το πρόβλημα προκύπτει όταν ο χρονοπρογραμματισμός των διεργασιών από το λειτουργικό σύστημα (kernel processes scheduling) δεν είναι αρκετά λεπτομερής ή ιδιαίτερα ακριβής.

Για παράδειγμα σκεφτείτε ένα λειτουργικό που έχει σταθερό χρονοπρογραμματισμό διεργασιών 10ms και μια εφαρμογή που θέλει να στέλνει ένα πακέτο RTP κάθε 30 χιλιοστά του δευτερολέπτου. Η εφαρμογή ξοδεύει, ας πούμε, 5 χιλιοστά του δευτερολέπτου να κάνει κάποια επεξεργασία (π.χ. σύνθεση ομιλίας από κείμενο). Μετά από αυτό, θα πρέπει να κοιμηθεί ακριβώς για 25 χιλιοστά του δευτερολέπτου, έτσι ώστε το διάστημα μεταξύ των πακέτων να είναι ακριβώς 30 ms. Όμως, λόγω του χρονοπρογραμματισμού των 10ms, το μήκος του ύπνου της εφαρμογής στρογγυλοποιείται στο πλησιέστερο πολλαπλάσιο των 10 ms. Με άλλα λόγια, το χρονικό διάστημα μεταξύ των πακέτων καταλήγει να είναι 35 χιλιοστά του δευτερολέπτου. Αυτό συμβαίνει μεταξύ κάθε ζεύγους πακέτων, και έχουμε πραγματικά κακή ποιότητα ήχου.

Για την άμβλυνση προβλημάτων ποιότητα φωνής που προκαλούνται από το jitter και την απόρριψη πακέτων χρησιμοποιείται συχνά ένας ενταμιευτής (buffer) για την προσωρινή αποθήκευση των πακέτων πριν από τη διαβίβαση τους για περαιτέρω επεξεργασία. Με τους ενταμιευτές jitter η ποιότητα φωνής δεν μπορεί να επηρεαστεί

από λίγα χιλιοστά του δευτερολέπτου με απομονωμένες αιχμές jitter. Το μειονέκτημα είναι ότι εισάγεται μια επιπλέον άκρο-σε-άκρο καθυστέρηση.

Χρονισμός εφαρμογών μέσω

Συνεχίζοντας την συζήτηση γύρω από τα θέματα ποιότητας, σε αυτή την ενότητα θα εξετάσουμε πως ο χρονισμός μιας εικονικής μηχανής μπορεί να επηρεάσει την λειτουργία εφαρμογών πραγματικού χρόνου, χρησιμοποιούμε την πολύ γνωστή και διαδεδομένη εφαρμογή Asterisk ως παράδειγμα.

Το λογισμικό Asterisk χρειάζεται μια πηγή χρονισμού για να καλύψει τις ανάγκες των διάφορων εφαρμογών του, RTP stream και playback (π.χ. conference, voice mail, music on hold, κα). Μια εξωτερική πηγή χρονισμού μπορεί να είναι υλικό ή λογισμικό, για παράδειγμα μια αναλογική ή ψηφιακή κάρτα στον Η/Υ (π.χ. Zaptel), μια συσκευή USB (UHCI-compatible root hub) ή μια διεπαφή με το ρολόι του συστήματος.

Οι πηγές χρονισμού, όταν βασίζονται σε υλικό, έχουν το πλεονέκτημα ότι δεν επηρεάζονται από το φόρτο του συστήματος, που φιλοξενεί την εφαρμογή και παρέχουν σταθερό και αξιόπιστο χρονισμό.

Οι περισσότερες εφαρμογές ικανοποιούνται με διακοπές (interrupts), συνήθως συχνότητας 1KHz (1.000 διακοπές ανά δευτερόλεπτο).

Ως προς την καθαυτό υλοποίηση του Asterisk, στο παρελθόν υπήρχε ένας μοναδικός τρόπος χρονισμού, μέσω Zaptel (τώρα DAHDI), που μπορούσε να λάβει το χρονισμό από κάρτες τηλεφωνίας τύπου Zaptel ή αν δεν υπήρχε κάποια τέτοια κάρτα από το kernel. Από την έκδοση Asterisk 1.6.1 ενσωματώθηκε ένα νέο timing API που δίνει ευελιξία και επιτρέπει τη χρήση διάφορων άλλων τρόπων χρονισμού.

Η παρουσία μιας αξιόπιστης πηγής χρονισμού επιβάλλεται από το πρωτόκολλο RTP, μέσω του πεδίου RTP timestamp (32 bits), ώστε να επιτρέπει στον παραλήπτη συγχρονισμό και υπολογισμό του jitter. Το timestamp αντανακλά τη στιγμή του πρώτου δείγματος (first octet) σε ένα πακέτο RTP. Παράγεται από ένα ρολόι το οποίο αυξάνεται μονοτονικά και γραμμικά στο χρόνο για να επιτρέψει τον συγχρονισμό και

τον υπολογισμό του jitter. Επομένως η ανάλυση και η ποιότητα του ρολογιού πρέπει να είναι ικανή για την επιθυμητή ακρίβεια συγχρονισμού και υπολογισμού του jitter.

Η συχνότητα του ρολογιού εξαρτάται από την μορφή των δεδομένων που μεταφέρονται (payload) και προσδιορίζεται στατικά. Για φωνή σταθερού ρυθμού το timestamp αυξάνεται σύμφωνα με τον (σταθερό) αριθμό των δειγμάτων του πακέτου, για παράδειγμα για πακέτα που περιέχουν 20 ms φωνή με δειγματοληψία 8.000Hz, το timestamp αυξάνεται με συνεχόμενο ρυθμό 160 μονάδων, ακόμα και αν δεν αποστέλλεται κάποιο πακέτο λόγω σιωπής.

Οι εικονικές μηχανές (virtual machines) δεν δουλεύουν με χρονισμό του φυσικού υλικού της πραγματικής μηχανής, αλλά του επιπέδου εικονοποίησης, επομένως δεν μπορούν να πολλαπλασιάσουν την ενέργεια χρονισμού με μεγάλη ακρίβεια. Οι τεχνικές εικονοποίησης εφαρμόζουν διάφορες μεθόδους για να βελτιώσουν την απόδοση - ακρίβεια του χρονισμού, που όμως σε ορισμένες περιπτώσεις δημιουργούνται προβλήματα στις εφαρμογές των εικονικών μηχανών.

Γενικά υπάρχουν δύο τρόποι χρονισμού

1. Tick counting. Στην περίπτωση αυτή υπάρχει ένα υλικό που διακόπτει (interrupt) περιοδικά το Λειτουργικό Σύστημα, π.χ. 1000 φορές το δευτερόλεπτο. Το ΛΣ διαχειρίζεται τις διακοπές (λεγόμενα ticks) για να προσδιορίσει την παρέλευση του χρόνου.
2. Tickless. Ένα υλικό μετρά και αποθηκεύει σε ένα μετρητή πόσες μονάδες χρόνου πέρασαν από την εκκίνηση του συστήματος και το ΛΣ διαβάζει, όταν το χρειάζεται, τον μετρητή αυτό.

Η πρώτη μέθοδος (tick counting) χρησιμοποιείται κυρίως σε προσωπικούς υπολογιστές και είναι δύσκολο να δώσει ακρίβεια χρονισμού σε εικονικές μηχανές. Μια διακοπή πρέπει να μοιραστεί σε όλες τις εφαρμογές του ΛΣ, καθώς και στις εικονικές μηχανές. Είναι πιθανόν μια εικονική μηχανή να μην έχει άμεση πρόσβαση σε μια διακοπή, αλλά να βρεθεί μπροστά σε συσσωρευμένες διακοπές ή κάποιες διακοπές να μην φτάσουν καθόλου.

Η δεύτερη μέθοδος είναι πιο αποδοτική, καθώς δεν απασχολεί την CPU με (αχρείαστες) διακοπές, αλλά διαβάζει τον μετρητή μόνο όταν τον χρειάζεται.

Για την μεταφορά του χρονισμού στις εικονικές μηχανές δεν υπάρχει μια συγκεκριμένη πρότυπη μέθοδος. Κάθε υλοποίηση χρησιμοποιεί μια ή περισσότερες τεχνικές. Για παράδειγμα, η VMware έχει αναπτύξει και χρησιμοποιεί δύο τεχνικές, η πρώτη (apparent time) προσομοιώνει ένα μετρητή χρόνου (timer) για κάθε εικονική μηχανή, ενώ η δεύτερη μέθοδος προσφέρει δυνατότητα άμεσης πρόσβασης της εικονικής μηχανής στο πραγματικό ρολόι του συστήματος, βεβαία υπό την προϋπόθεση ότι η εφαρμογή είναι ενήμερη ότι τρέχει σε VMware και άρα συμπεριλαμβάνει τον κατάλληλο κώδικα.

Στα Linux συστήματα οι τεχνικές timekeeping έχουν αλλάξει δραστικά στις τελευταίες εκδόσεις kernel. Το αρχικό σύστημα χρονισμού ("timer wheel") ήταν βασισμένο στην ιδέα της επαύξησης μιας εσωτερικής μεταβλητής (jiffy) σε κάθε διακοπή χρονισμού.

Αυτή η διακοπή χρονισμού γίνεται η προεπιλεγμένη ελάχιστη μονάδα προγραμματισμού και όλα τα άλλα χρονόμετρα του συστήματος έχουν ως βάση τα jiffies. Ο ρυθμός των διακοπών (χρονισμός) και το βήμα αύξησης καθορίζονται με μια σταθερά (την HZ) κατά την φάση της μεταγλώττισης. Διαφορετικές πλατφόρμες χρησιμοποιούν διαφορετικές τιμές για το HZ. Ιστορικά, στο πρώτο πυρήνα του Linux χρησιμοποιήθηκε το 100 ως τιμή για HZ, αποδίδοντας ένα ρυθμό διακοπών ανά 10 ms. Αργότερα, στη έκδοση 2.4, η τιμή HZ για i386 άλλαξε σε 1000, δίνοντας ένα ρυθμό 1 ms. Πρόσφατα (2.6.13) ο πυρήνας άλλαξε το HZ για i386 σε 250, καθώς το 1000 θεωρήθηκε πολύ υψηλή τιμή.

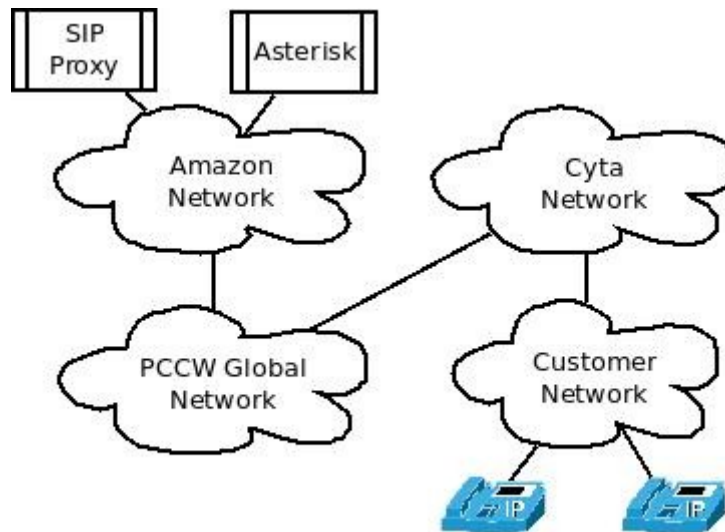
Η πιο πρόσφατη σημαντική αλλαγή ξεκινά στις εκδόσεις 2.6.18 για 32bit και 2.6.21 για 64bit kernels όπου ενσωματώθηκε ένα νέο επίπεδο αφαίρεσης (ονομαζόμενο clocksource) για τον χρονισμό του συστήματος. Έτσι ο kernel μπορεί να επιλέγει το χρονισμό μεταξύ διαφορετικών tickless πηγών κατά την εκκίνησή του. Η δεύτερη μεγάλη αλλαγή έγινε στις εκδόσεις 2.6.21 και 2.6.24 για 32bit και 64bit kernels αντίστοιχα, με την εισαγωγή των "clockevents".

5.2 Ποιότητα Υπηρεσίας

Με τον όρο Ποιότητα Υπηρεσίας (QoS - Quality of Service) εννοούμε τις διάφορες τεχνικές που εφαρμόζονται στα δίκτυα ώστε να ικανοποιήσουν τις απαιτήσεις των διάφορων εφαρμογών, bit rates, delay, packet loss rates. Παρόλο που για αρκετά χρόνια ήταν στο επίκεντρο των προσπαθειών, τα τελικά αποτελέσματα, όσον αφορά γενικότερα στο Διαδίκτυο, δεν είναι τόσο θετικά. Η σημερινή πραγματικότητα δείχνει ότι τέτοιες υπηρεσίες γενικά δεν προσφέρονται στους πελάτες από τους παρόχους, αλλά οι τεχνικές QoS περιορίζονται εντός των ορίων των δικτύων για για εσωτερικές κυρίως υπηρεσίες ή λειτουργίες.

Μια δεύτερη παρατήρηση είναι ότι για να υπάρχει ουσιαστικό αποτέλεσμα στην ποιότητα υπηρεσίας, αυτή πρέπει να είναι εξασφαλισμένη από άκρο-σε-άκρο. Η τελική ποιότητα είναι η χειρότερη μεταξύ των ποιοτήτων που απολαμβάνει η υπηρεσία στα διαφορετικά δίκτυα από όπου διέρχεται. Επομένως θα πρέπει να υπάρχει μια συμφωνία και μια διαλειτουργικότητα ποιότητας μεταξύ όλων των δικτύων από τα οποία διέρχεται μια υπηρεσία για να υπάρχει αποτέλεσμα, γεγονός που σήμερα θεωρείται πρακτικά αδύνατο.

Θα προσπαθήσουμε να γίνει περισσότερο κατανοητό με ένα απλό παράδειγμα. Έστω ότι μια εταιρία αποφασίζει να κάνει χρήση των υπηρεσιών cloud της Amazon (Amazon EC2) με σκοπό να αναπτύξει υπηρεσίες IP Τηλεφωνίας. Προμηθεύεται λοιπόν δύο εικονικούς εξυπηρετητές (virtual servers) για να φιλοξενήσει στον πρώτο ένα SIP Proxy και στο δεύτερο ένα Asterisk για media services. Ας υποθέσουμε επίσης ότι η εταιρία αυτή διαθέτει πρόσβαση στο Διαδίκτυο μέσω της Cyta (τηλεπικοινωνιακός πάροχος σε Κύπρο και Ελλάδα). Μέσω της εντολής traceroute μπορούμε να ανιχνεύσουμε τα ενδιάμεσα δίκτυα και δημιουργήσουμε την ακόλουθη τοπολογία δικτύων από το εσωτερικό δίκτυο της εταιρίας μέχρι τους εικονικούς εξυπηρετητές της Αμαζον:



Σχήμα 5.1 Τοπολογία δικτύων μεταξύ Cyta και Amazon

Βλέπουμε ότι η κίνηση από μια συσκευή του πελάτη προς τους εικονικούς εξυπηρετητές διέρχεται διαμέσου τεσσάρων διαφορετικών δικτύων, άρα για να έχουμε εξασφαλισμένη Ποιότητα Υπηρεσίας στη κίνηση αυτή χρειάζεται να παρέχεται εξίσου και στα τέσσερα αυτά δίκτυα.

Είναι γνωστό ότι σήμερα κανένας ISP - εμπορικός πάροχος διαδικτύου - δεν προσφέρει Ποιότητα Υπηρεσίας στις υπηρεσίες το για το ευρύ κοινό. Αλλά ούτε η Amazon ή κάποιος άλλος πάροχος cloud υπηρεσιών προσφέρει Ποιότητα Υπηρεσίας στις δικές τους υπηρεσίες.

Επομένως η μοναδική δυνατότητα υλοποίησης QoS είναι μόνο εντός του ιδιόκτητου δικτύου της εταιρίας μας. Αλήθεια έχει νόημα κάτι τέτοιο, όταν ήδη έχουμε πει ότι η Ποιότητα Υπηρεσίας αποδίδει καρπούς όταν είναι άκρο-σε-άκρο; Η απάντηση είναι ναι για κάποιες περιπτώσεις, αρκετές μάλιστα. Ακολουθεί μια αιτιολόγηση της απάντησης μας.

Έχουμε ήδη πάρει την πρώτη σημαντική απόφαση του περιβάλλοντος ανάπτυξης της υπηρεσίας που είναι η επιλογή ενός SIP Proxy και όχι ενός B2BUA. Ένας SIP Proxy φροντίζει για την επικοινωνία μεταξύ χρηστών ελέγχοντας την σηματοδοσία και προωθώντας την απαραίτητη πληροφορία από τον ένα χρήστη στον άλλο ώστε τελικά να εγκαθιδρυθεί μια κλήση, ενώ δεν εμπλέκεται στην μεταφορά των

δεδομένων, δηλαδή της φωνής μεταξύ των χρηστών, η οποία γίνεται απευθείας. Για παράδειγμα, όταν ο χρήστης A καλεί, μέσω του Proxy, τον χρήστη B που βρίσκονται στο εσωτερικό δίκτυο της εταιρίας μας, η SIP σηματοδότηση περνά μέσω του Proxy εκτός δικτύου αλλά η φωνή ταξιδεύει απευθείας από τον χρήστη A στον χρήστη B εντός του εταιρικού δικτύου. Επομένως η εγκαθίδρυση κατάλληλων μηχανισμών Ποιότητας Υπηρεσίας, εντός του εταιρικού δικτύου, έχει ουσιαστικό αποτέλεσμα για τις κλήσεις που πραγματοποιούνται εντός του δικτύου αυτού. Για τις υπηρεσίες που τα δεδομένα χρειάζεται να φτάσουν στις εφαρμογές που φιλοξενούνται στις εικονικές μηχανές εκτός δικτύου (π.χ. voice mail, conference), η Ποιότητα Υπηρεσίας εντός του εσωτερικού δικτύου δεν έχει να προσφέρει πολλά καθώς τα δεδομένα σε όλα τα υπόλοιπα δίκτυα αντιμετωπίζονται με βάση την καλύτερη προσπάθεια (best effort).

Βεβαίως, κάποιος θα μπορούσε να σκεφτεί ότι θα μπορούσαν να φιλοξενηθούν αυτές οι εφαρμογές σε φυσικούς ή εικονικούς εξυπηρετητές εντός του εταιρικού δικτύου, ώστε όλη η κίνηση, εκτός της σηματοδότησης, να μένει εντός δικτύου, άρα να λαμβάνει την κατάλληλη Ποιότητα Υπηρεσίας. Αυτό το υβριδικό σχήμα, αφενός δεν αξιοποιεί πλήρως τα πλεονεκτήματα του cloud, καθώς μέρος της υπηρεσίας μένει εκτός, και αφετέρου εξακολουθεί να κληρονομεί τα μειονεκτήματα που έχει η λειτουργία φυσικών εξυπηρετητών από τον ίδιο τον οργανισμό. Συνεπώς η φιλοξενία αυτών των εφαρμογών σε ίδιους εξυπηρετητές δεν είναι λύση.

5.3 Ποιότητα Εμπειρίας - Quality of Experience (QoE)

Η αδυναμία καθορισμού ενός καθολικού πλαισίου παροχής Ποιότητας Υπηρεσίας με ένα απλό και ενιαίο τρόπο έχουν τροφοδοτήσει τα τελευταία χρόνια μια διαφορετική συζήτηση γύρω από την ποιότητα μιας υπηρεσίας ως την αντίληψη του χρήστη - πελάτη για το πόσο καλή είναι μια υπηρεσία και όχι πόσο καλά υποστηρίζεται από ένα δίκτυο.

Η ITU-T εφαρμόζει τον ακόλουθο ορισμό για την Ποιότητα Εμπειρίας - Quality of Experience (QoE): Είναι η συνολική αποδοχή μιας εφαρμογής ή υπηρεσίας όπως αυτή γίνεται υποκειμενικά αντιληπτή από τον τελικό χρήστη.

Επομένως η Ποιότητα Εμπειρίας είναι μια υποκειμενική μέτρηση της εμπειρίας του χρήστη, που πρέπει να είναι τακτική και συνεχής.

Ειδικά για την φωνή, τέτοιες προσπάθειες έχουν γίνει ήδη πολύ παλαιότερα ώστε να ποσοτικοποιηθεί η ανθρώπινη εμπειρία με τη βοήθεια της χρήσης του Mean Opinion Score (MOS) ή απλά η κοινή γνώμη. Το MOS αρχικά χρησιμοποιήθηκε για να βοηθήσει στο σχεδιασμό, την έρευνα και την ανάπτυξη του ψηφιακών τηλεφωνικών συστημάτων, τη μετάφραση και την ανασύσταση αναλογικού σήματος φωνής σε ψηφιακό και αντίστροφα. Μπορούμε να πούμε ότι ο όρος MOS έχει γίνει συνώνυμος με την Ποιότητα Εμπειρίας για τις κλήσεις φωνής. Έτσι το MOS είναι μια ποιοτική μέτρηση που υπολογίζεται όταν πολλοί άνθρωποι ακούνε μια σειρά από τηλεφωνήματα και τα κατατάσσουν σε μια κλίμακα 1 - 5 (5 είναι το καλύτερο) από την άποψη της ποιότητας της εμπειρίας τους. Έχουμε έτσι για παράδειγμα τα αποτελέσματα από την χρήση διαφόρων codecs:

Codec	Data rate [kbit/s]	Mean opinion score (MOS)
iLBC	15.2	4.14
AMR	12.2	4.14
G.711 (ISDN)	64	4.1
G.729	8	3.92
G.723.1 r63	6.3	3.9
G.726 ADPCM	32	3.85
GSM EFR	12.2	3.8
G.729a	8	3.7
G.723.1 r53	5.3	3.65
G.728	16	3.61
GSM FR	12.2	3.5

Η μέτρηση της Ποιότητα Εμπειρίας (Quality of Experience Measurement) είναι η διαδικασία της κατανόησης της πραγματικής απόδοσης των υπηρεσιών, όπως παραδίδονται στον πελάτη, με σκοπό να εξασφαλιστεί ότι οι εν λόγω υπηρεσίες ανταποκρίνονται στις προσδοκίες και τις απαιτήσεις των πελατών. Αυτές οι μετρήσεις γίνονται στο σημείο παράδοσης της υπηρεσίας, δηλαδή στην τερματική συσκευή του χρήστη, π.χ. στον υπολογιστή ή στο smartphone.

Ο ουσιαστικός σκοπός μιας μέτρησης QoE είναι η πρακτική κατανόησης πόσο καλά λειτουργεί μια υπηρεσία, από την άποψη του συνδρομητή, και στη συνέχεια, χρησιμοποιώντας αυτές τις πληροφορίες, να παραδώσει μια υπηρεσία που ικανοποιεί τις απαιτήσεις κατά τρόπο πιο αποτελεσματικό.

Η μέτρηση εξετάζει πόσο καλά εφαρμογές όπως η περιήγηση στο διαδίκτυο, το ηλεκτρονικό ταχυδρομείο, η VoIP τηλεφωνία, το streaming video αποδίδουν στα χέρια του συνδρομητή. Η σύγκριση βασικών δεικτών, όπως η απόδοση του δικτύου, η καθυστέρηση, η απώλεια πακέτων σίγουρα πρέπει να μετρηθούν και να αναλυθούν. Ωστόσο, αυτές οι τιμές αξιολογούνται όσον αφορά την συνδυασμένη επίδραση τους στην εμπειρία του συνδρομητή.

5.4 Service Level Agreement (SLA)

Ένα SLA είναι μια συμφωνία μεταξύ παρόχου και πελάτη για το επίπεδο των παρεχόμενων υπηρεσιών και των ποινών σε περιπτώσεις παραβίασης των συμφωνηθέντων, καθώς και ένα σύνολο εξαιρέσεων, που τίθενται από τον πάροχο των υπηρεσιών.

Μελετώντας διάφορα SLAs στον τομέα των υπηρεσιών υποδομής ως υπηρεσία (IaaS) διαπιστώνει κανείς ότι οι παρεχόμενες εγγυήσεις περιορίζονται σε ποσοστά διαθεσιμότητας της υπηρεσίας και εκφράζονται κατά τρόπο ώστε να κατανέμουν τον κίνδυνο πίσω στον πελάτη. Η συνήθης πρακτική είναι να δίνεται η δυνατότητα στον πελάτη, σε περίπτωση διακοπής μιας εικονικής μηχανής (instance), να μπορεί να "σηκώσει" μια άλλη στην ίδια περιοχή, διαφορετικά, αν αυτό δεν είναι εφικτό, λαμβάνει μια πίστωση στο λογαριασμό του ανάλογα με τη συμφωνία.

Σύμφωνα με την αναλύτρια, Lyndia Leong της Gartner "δυστυχώς, σκοπός των SLA δεν είναι να δώσουν πίσω τα χρήματα των πελατών για την αντιστάθμιση των απωλειών του πελάτη που προέκυψαν από την διακοπή (downtime), αλλά τα SLA αντιπροσωπεύουν μια ψήφο εμπιστοσύνης - πόσο σίγουρος είναι ο πάροχος για την ικανότητά του να υποστηρίξει το SLA και πόσα χρήματα είναι πρόθυμος να στοιχηματίσει σε αυτό". Στην κλίμακα των μεγάλων παρόχων υπάρχουν πολλοί καλοί και εύλογοι λόγοι να φοβούνται τις οικονομικές επιπτώσεις από μαζικές διακοπές. Η φύση των αρχιτεκτονικών cloud IaaS, όπως οι παραδοσιακές υπηρεσίες υποδομής, είναι ευάλωτες σε καταστροφές σε ένα Κέντρο Δεδομένων, αλλά είναι επίσης ακόμα πιο ευάλωτες σε λάθη στο λογικό επίπεδο και το επίπεδο ελέγχου.

Για παράδειγμα, το Amazon EC2 Service Level Agreement αναφέρει ότι "αν το ετήσιο ποσοστό Uptime για έναν πελάτη πέσει κάτω από 99,95% για το έτος υπηρεσίας, τότε ο πελάτης είναι επιλέξιμος να λάβει πίστωση υπηρεσιών ίση με το 10% του λογαριασμού του για την επιλέξιμη περίοδο πίστωσης".

Ένα αντίστοιχο μοντέλο προτείνει και η HP, στο δικό της HP Cloud Compute Service Level Agreement η διαθεσιμότητα μετριέται μηνιαία και η πίστωση είναι κλιμακωτή:

- <99.95% to 99.9% 5%
- <99.9% to 99.5% 10%
- <99.5% to 99% 20%
- <99.0% 30%

Ένας άλλος πάροχος, με μεγαλύτερη αυτοπεποίθηση, στο δικό του OpSource Cloud Hosting SLA, προσφέρει μηνιαία διαθεσιμότητα 100% και κλιμακωτή πίστωση που φτάνει μέχρι 100%, όταν μια διακοπή υπερβεί τα 660 λεπτά σε ένα μήνα.

Συνήθως οι πάροχοι επιδιώκουν να διαφοροποιήσουν το SLA από τους ανταγωνιστές τους με σκοπό να το κάνουν περισσότερο ελκυστικό. Δεν ξεφεύγουν όμως από τον βασικό τους στόχο που είναι να αποφύγουν την ευθύνη της ουσιαστικής ζημίας που προκαλείται στους πελάτες, άρα να δώσουν πίσω χρήματα για την αντιστάθμιση των απωλειών που προέκυψαν από την διακοπή.

Μια άλλη σημαντική διαπίστωση, με ιδιαίτερη σημασία για τις υπηρεσίες πραγματικού χρόνου, είναι ότι σήμερα καμία δημόσια ή ιδιωτική πλατφόρμα υπηρεσιών υπολογιστικού νέφους (private / public cloud) δεν προσφέρεται η δυνατότητα κράτησης υπολογιστικών πόρων (άρα και εγγύηση), αλλά οι πόροι παρέχονται κατ' απαίτηση (on-demand) με βάση την καλύτερη προσπάθεια (best effort) κάθε εικονικής μηχανής. Επομένως ο πάροχος δεν μπορεί να εγγυηθεί συγκεκριμένες τιμές - όρια - στους χρόνους πρόσβασης σε πόρους όπως η CPU, το ρολόι του συστήματος, η μνήμη, το δίκτυο, ο αποθηκευτικός χώρος.

Κεφάλαιο 6

Δοκιμές - Μετρήσεις

Στο κεφάλαιο αυτό παρουσιάζουμε τις δοκιμές υπερφόρτωσης που σε ένα πραγματικό ολοκληρωμένο σύστημα επικοινωνίας που φιλοξενήθηκε σε μια εικονική μηχανή του ΕΔΕΤ (υπηρεσία vima.grnet.gr) που περιλαμβάνει μια (1) εικονική CPU 2.66GHz (δηλαδή ένα core), μνήμη 2.0 GB και Hard disk 50 GB, Λειτουργικό Σύστημα CentOS 5.9 και Linux kernel version 2.6.18-348.1.1.el5.

Οι δοκιμές υπερφόρτωσης έχουν την λογική ελέγχου των ορίων του συστήματος, αλλά και τον υποσυστημάτων του. Έγιναν δοκιμές υπερφόρτωσης και μετρήσεις στα ακόλουθα τρία επίπεδα βασικών λειτουργιών του συστήματος:

1. Λειτουργία καταχώρησης (registration), αφορούσε στον SIP Proxy.
2. Πραγματοποίηση κλήσεων (invite), αφορούσε στον SIP Proxy
3. Προηγμένες Υπηρεσίες (media services), αφορούσε media services.

Για την πραγματοποίηση των δοκιμών υπερφόρτωσης έγινε χρήση του ανοικτού λογισμικού sipp (<http://sipp.sourceforge.net/>). Το λογισμικό sipp προσομοιώνει έναν UA και μέσω αρχείων xml επιτρέπει την δημιουργία διάφορων σεναρίων για την υπερφόρτωση ενός συστήματος.

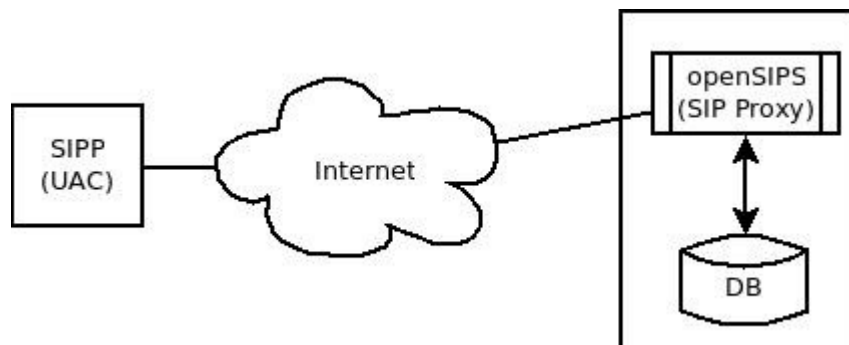
Επίσης έγιναν ορισμένες τροποποιήσεις στα Scripts κυρίως των OpenSIPs ώστε να απομονώνονται τα υποσυστήματα και να γίνονται μετρήσεις προσθέτοντάς τα σταδιακά. Για παράδειγμα σε όλες τις δοκιμές υπερφόρτωσης αίρεται η λειτουργία εκσφαλμάτωσης (debug) στη λειτουργία του OpenSIPs, ώστε να μην αποτελεί περιορισμό το τοπικό ΙΟ σύστημα των δίσκων.

Ακολουθεί αναλυτική περιγραφή των δοκιμών και οι σχετικές μετρήσεις.

Υπερφόρτωση λειτουργίας καταχώρησης (registration)

Η δοκιμή φόρτου καταχώρησης (SIP registration) περιλαμβάνει το σενάριο πραγματοποίησης και καταμέτρησης πολλαπλών καταχωρήσεων ανά δευτερόλεπτο μέχρι να εμφανιστούν στο λογισμικό sipp πολλές επαναλήψεις πακέτων ή λάθη. Οι

καταχωρήσεις γίνονται με πιστοποίηση (username / password), όπως ακριβώς σε μια κανονική λειτουργία του συστήματος. Τα στοιχεία του χρήστη είναι αποθηκευμένα σε μια Βάση Δεδομένων, που βρίσκεται στο ίδιο σύστημα.



Σχήμα 6.1 Τοπολογία σεναρίου υπερφόρτωσης καταχώρησης

Η διαδικασία του Registration περιλαμβάνει τις παρακάτω λειτουργίες OpenSIPs:

1. Η εφαρμογή sipp (UA) αιτείται καταχώρηση στο σύστημα OpenSIPs.
2. Ο OpenSIPs απαντά ζητώντας πιστοποίηση.
3. Η εφαρμογή sipp αιτείται εκ νέου καταχώρηση στο σύστημα OpenSIPs, παρέχοντας τα κατάλληλα διαπιστευτήρια.
4. Ο OpenSIPs ζητά από τη Βάση Δεδομένων το κωδικοποιημένο password που αντιστοιχεί στο χρήστη sipptest, ελέγχει αν συμφωνεί με αυτό που έλαβε και αποθηκεύει τα στοιχεία τοποθεσίας στη Βάση Δεδομένων.

Για την εκτέλεση της δοκιμής υπερφόρτωσης χρησιμοποιείται το Sipp με τις παρακάτω παραμέτρους.

```
# cd /usr/src/sipp/sipp-3.3
# ./sipp iptel.no-ip.org:5060 -r 1500 -i 147.52.82.82 -s sipptest -sf
test_registration.xml
```

Όπου

iptel.no-ip.org:5060 η ip διεύθυνση του OpenSIPs στον οποίο θα σταλούν τα Registrations

-r 1500 η εφαρμογή ξεκινά με 1500 Registration ανά δευτερόλεπτο **(κατά την διάρκεια της δοκιμής, πατώντας το “+ ή *” ο ρυθμός μπορεί να αυξηθεί κατά 1 ή 10)**

-i 147.52.82.82 η διεύθυνση του τοπικού συστήματος

-s sipptest το όνομα χρήστη που ζητά καταχώρηση

-sf test_registration το σενάριο (αρχείο xml) για την εκτέλεση της δοκιμής.

Τα σενάρια του Sipr είναι XML αρχεία τα οποία περιέχουν την δομή των πακέτων που θα πρέπει να αποστείλουν καθώς και τις απαντήσεις που θα πρέπει να δεχθούν για να προχωρήσουν στην επόμενη αποστολή. Το σενάριο test_registration.xml είναι ένα σενάριο το οποίο παρουσιάζεται στον ακόλουθο πίνακα.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<scenario name="Basic Sipstone UAC">
  <send retrans="500">
    <![CDATA[
      REGISTER sip:[remote_ip] SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:[local_port]
      To: <sip:[service]@iptel.no-ip.org:[remote_port]>
      From: <sip:[service]@[remote_ip]:[remote_port]>
      Contact: <sip:[service]@[local_ip]:
[local_port]>;transport=[transport]
      Expires: 300
      Call-ID: [call_id]
      CSeq: 2 REGISTER
      Content-Length: 0
    ]]>
  </send>

  <recv response="401" auth="true">
  </recv>

  <send retrans="500">
    <![CDATA[
      REGISTER sip:[remote_ip] SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:[local_port]
      To: <sip:[service]@iptel.no-ip.org:[remote_port]>
      From: <sip:[service]@[remote_ip]:[remote_port]>
      Contact: <sip:[service]@[local_ip]:
[local_port]>;transport=[transport]
      [authentication username=michalis password=m1ch@11s]
      Expires: 300
      Call-ID: [call_id]
      CSeq: 2 REGISTER
      Content-Length: 0
    ]]>
  </send>
</scenario>
```

```

]]>
</send>

<recv response="200" rtd="true">
</recv>
</scenario>

```

Αποτελέσματα και συμπεράσματα δοκιμών καταχώρησης

Καταχωρήσεις ανά δευτερόλεπτο	Φόρτος επεξεργαστή εφαρμογής sipr	Φόρτος επεξεργαστή εφαρμογής openSIPS	Επαναλήψεις πακέτων ή λάθη
1.500	13,4	27,4	0
2.000	11,1	37,0	0
2.500	18,7	47,8	0
2.750	14,0	52,2	0
3.000	14,1	57,7	3
3.200	22,3	54,7	7
3.270	26,0	54,2	>638

Ο openSIPS χρησιμοποιεί 4 διεργασίες (child processes) για ταυτόχρονη εξυπηρέτηση των αιτημάτων, οι οποίες αθροιστικά καταναλώνουν το μεγαλύτερο ποσοστό υπολογιστικών πόρων (CPU time). Η χρήση της CPU του συστήματος openSIPS έφτασε μέχρι το 54,2%, γεγονός που δείχνει ότι το όριο δεν οφείλεται στην υψηλή χρήση της CPU.

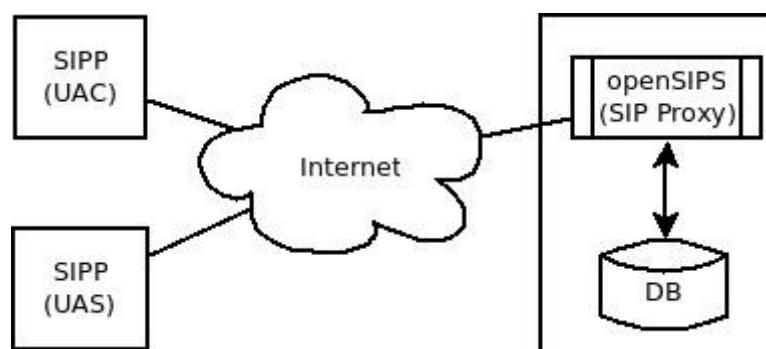
Η εφαρμογή mysql χρησιμοποιεί σε όλες τις μετρήσεις το δεύτερο μεγαλύτερο ποσοστό υπολογιστικών πόρων, μετά τον openSIPS.

Η χρήση της κύριας μνήμης (RAM) του συστήματος openSIPS ήταν γύρω στο 50%, άρα το όριο δεν οφείλεται στην χρήση της μνήμης.

Πάνω από τις 3.200 καταχωρήσεις ανά δευτερόλεπτο το σύστημα αδυνατεί να ανταποκριθεί στα αιτήματα και έχουμε πολλές επαναλήψεις.

Υπερφόρτωση λειτουργίας εγκαθίδρυσης κλήσεων (Invite Overload)

Η δοκιμή υπερφόρτωσης κλήσεων (Invite Overload) περιλαμβάνει το σενάριο πραγματοποίησης πολλαπλών κλήσεων προς συγκεκριμένο προορισμό ώστε να μετρηθούν το πόσες νέες κλήσης ανά δευτερόλεπτο μπορεί ο OpenSIPS να υποστηρίξει. Η κλήση θα ξεκινάει από έναν Sipp που θα παίζει τον ρόλο του SIP Client (UAC) και θα απαντιούνται από έναν άλλο Sipp που θα παίζει το ρόλου του Server (UAS).



Σχήμα 6.2 Τοπολογία σεναρίου υπερφόρτωσης κλήσεων

Για την εκτέλεση της δοκιμής χρησιμοποιήθηκαν δύο υπολογιστές με εγκατεστημένο το Sipp. Ο πρώτος, παίζει τον ρόλο του UAS, και θα ξεκινήσει το Sipp με την παρακάτω εντολή:

```
# ./sipp -sn uas -p 5060 -i 147.52.82.82
```

Όπου

-sn το ενσωματωμένο σενάριο SIP user agent server το οποίο θα εκτελεστεί

-p η πόρτα στην οποία ακούει ο Sipp UA Server.

-i 147.52.82.82 η τοπική διεύθυνση του UA Server.

Για να είναι ο UAS προσβάσιμος από τον openSIPS (SIP server) δημιουργούμε μια χειρωνακτική καταχώρηση, δηλαδή στη βάση δεδομένων “location” του SIP registrar εισάγουμε μια εγγραφή με τη IP διεύθυνση του συστήματος του UAS (εδώ 147.52.82.82). Χρησιμοποιούμε την παρακάτω εντολή για αυτό το σκοπό:

```
#opensipsctl ul add sipptest sip:sipptest@147.52.82.82:5060
```

Ενώ ο δεύτερος Server ο οποίος θα παίζει τον ρόλο του UAC θα πρέπει να ξεκινήσει το Sipp με την παρακάτω εντολή

```
# ./sipp -sn uac -s sipptest -r 1000 iptel.no-ip.org -i 147.52.3.5
```

Όπου

-sn το ενσωματωμένο σενάριο το οποίο θα εκτελεστεί

-s sipptest είναι ο χρήστης που θα κληθεί

-r ο αριθμός των κλήσεων ανά δευτερόλεπτο με την οποία θα ξεκινήσει η δοκιμή (κατά την διάρκεια της δοκιμής, πατώντας το “+” ή “*” ο ρυθμός αυξάνεται κατά 1 ή δέκα κλήσεις αντίστοιχα)

iptel.no-ip.org δηλώνει τον openSIPS, SIP Server

-i 147.52.82.82 η τοπική διεύθυνση του UA client.

Σημ.: Καθώς αναμένεται μεγάλος αριθμός ταυτόχρονων κλήσεων, και για τις “ζωντανές” κλήσεις διατηρούνται κάποιες πληροφορίες, χρειάζεται να αποδοθεί στην εφαρμογή openSIPS μεγαλύτερο όριο διαθέσιμης μνήμης, αυτό γίνεται μέσω της παραμέτρου -m, δηλαδή δίνουμε 246 MB με την εντολή: opensips -m 256

Αποτελέσματα και συμπεράσματα δοκιμών εγκαθίδρυσης κλήσεων

Κλήσεις ανά δευτερόλεπτο	Φόρτος επεξεργαστή εφαρμογής openSIPS	Επαναλήψεις πακέτων ή λάθη
500	17,3%	0
900	42,0%	0
930	52,5%	>111

Η χρήση της CPU του συστήματος openSIPS έφτασε μέχρι το 52,5%, γεγονός που δείχνει ότι το όριο δεν οφείλεται στην υψηλή χρήση της CPU.

Η χρήση της κύριας μνήμης (RAM) του συστήματος openSIPS ήταν σταθερή γύρω στο 50%, γεγονός που δείχνει ότι το όριο δεν οφείλεται στην υψηλή χρήση της μνήμης.

Πάνω από τις 900 εγκαθιδρύσεις κλήσεων ανά δευτερόλεπτο το σύστημα αδυνατεί να ανταποκριθεί και έχουμε πολλές επαναλήψεις.

Υπερφόρτωση υπηρεσιών (media services)

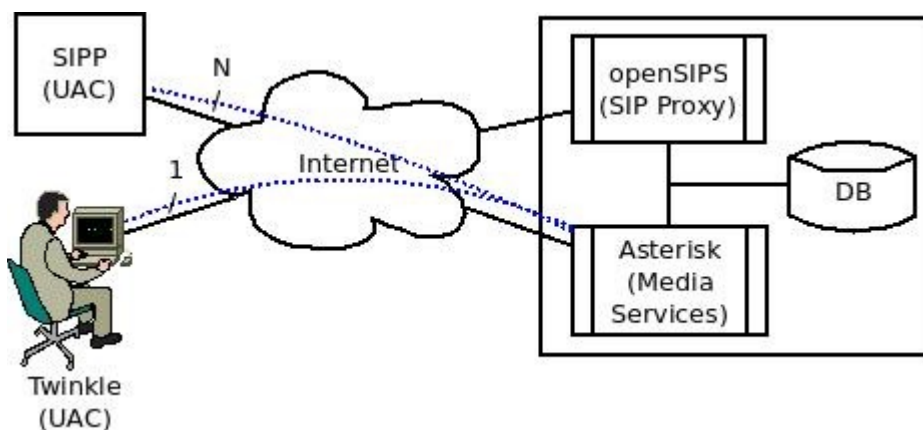
Όπως έχουμε ήδη αναφέρει, για την υποστήριξη συμπληρωματικών και πολυμεσικών υπηρεσιών χρησιμοποιείται ένας εξυπηρετητής Asterisk. Σε αυτή την ενότητα δοκιμών υπερφόρτωσης υλοποιούμε ένα σενάριο ελέγχου των αυτών ικανοτήτων του συστήματος μας:

- Ο εξυπηρετητής Asterisk έχει συνδεθεί με ζωντανή ραδιοφωνική ροή, η οποία αναμεταδίδεται, υπό μορφή μουσικής στην αναμονή, σε ανεξάρτητες ροές όταν δέχεται κλήση στον αριθμό 700.
- Το σύστημα openSIPS δρομολογεί κάθε κλήση που δέχεται προς τον αριθμό '*700' στον Asterisk, στο εσωτερικό αριθμό 700.
- Σε ένα υπολογιστή, με ρόλο UAC, ξεκινάμε το Sipp με ένα σενάριο σύμφωνα με το οποίο πραγματοποιείται μία (1) κλήση ανά δευτερόλεπτο προς τον αριθμό

'*700' (ο openSIPS δρομολογεί την κλήση στον Asterisk). Κάθε κλήση θα έχει διάρκεια 5 δευτερολέπτων, επομένως μετά από 5 δευτερόλεπτα θα έχουμε μόνιμα 5 ταυτόχρονες κλήσεις στο σύστημα, καθώς για κάθε μια που ολοκληρώνεται μια άλλη θα ξεκινά. Για να αυξήσουμε τον αριθμό των ταυτόχρονων κλήσεων, αυξάνουμε τον ρυθμό κλήσεων ανά δευτερόλεπτο - πατώντας το "+" αυξάνουμε τον ρυθμό κατά ένα. Για κάθε μονάδα αύξησης του ρυθμού κλήσεων αυξάνουμε κατά 5 τις ταυτόχρονες κλήσεις, δηλαδή για 2 κλήσεις ανά δευτερόλεπτο θα έχουμε 10 ταυτόχρονες κλήσεις στο σύστημα, για 3 θα έχουμε 15, για 4 θα έχουμε 20, κοκ.

```
# ./sipp -sf uac_pcap.xml -s *700 -r 1 iptel.no-ip.org -i 147.52.82.82
```

- Τέλος, για να ελέγχουμε την ποιότητα του ήχου (ποιότητα εμπειρίας - QoE) που μεταδίδεται από το σύστημα, καθώς θα αυξάνουμε τον αριθμό των ενεργών κλήσεων, συνδεόμαστε στο σύστημα και από ένα άλλο υπολογιστή, μέσω μιας εφαρμογής sip phone (εδώ επιλέξαμε το Twinkle).



Σχήμα 6.3 Τοπολογία σεναρίου υπερφόρτωσης υπηρεσίας ροής μέσου

Ακολουθεί το σενάριο της εφαρμογής SIPP.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE scenario SYSTEM "sipp.dtd">
<!-- -->
<!-- Sipp 'uac' scenario with pcap (rtp) play -->
<!-- -->
```

```

<scenario name="UAC with media">
  <!-- In client mode (sipp placing calls), the Call-ID MUST be      -->
  <!-- generated by sipp. To do so, use [call_id] keyword.          -->
  <send retrans="500">
    <![CDATA[

      INVITE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
      From: sipp <sip:sipp@[local_ip]:
[local_port]>;tag=[pid]SIPpTag09[call_number]
      To: [service] <sip:[service]@[remote_ip]:[remote_port]>
      Call-ID: [call_id]
      CSeq: 1 INVITE
      Contact: sip:sipp@[local_ip]:[local_port]
      Max-Forwards: 70
      Subject: Performance Test
      Content-Type: application/sdp
      Content-Length: [len]

      v=0
      o=user1 53655765 2353687637 IN IP[local_ip_type] [local_ip]
      s=-
      c=IN IP[local_ip_type] [local_ip]
      t=0 0
      m=audio [auto_media_port] RTP/AVP 8 101
      a=rtpmap:8 PCMA/8000
      a=rtpmap:101 telephone-event/8000
      a=fmtp:101 0-11,16
    ]]>
  </send>

  <recv response="100" optional="true">
  </recv>

  <recv response="180" optional="true">
  </recv>

  <!-- By adding rrs="true" (Record Route Sets), the route sets    -->
  <!-- are saved and used for following messages sent. Useful to test-->
  <!-- against stateful SIP proxies/B2BUAs.                        -->
  <recv response="200" rtd="true" crlf="true">
  </recv>

  <!-- Packet lost can be simulated in any send/recv message by    -->
  <!-- by adding the 'lost = "10"'. Value can be [1-100] percent. -->
  <send>
    <![CDATA[
      ACK sip:[service]@[remote_ip]:[remote_port] SIP/2.0
      Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
      From: sipp <sip:sipp@[local_ip]:
[local_port]>;tag=[pid]SIPpTag09[call_number]
      To: [service] <sip:[service]@[remote_ip]:
[remote_port]>[peer_tag_param]
      Call-ID: [call_id]
      CSeq: 1 ACK
      Contact: sip:sipp@[local_ip]:[local_port]
      Max-Forwards: 70
    ]]>
  </send>

```

```

    Subject: Performance Test
    Content-Length: 0
  ]]>
</send>

<pause milliseconds="5000"/>

<!-- The 'crlf' option inserts a blank line in the statistics report.
-->
<send retrans="500">
  <![CDATA[

    BYE sip:[service]@[remote_ip]:[remote_port] SIP/2.0
    Via: SIP/2.0/[transport] [local_ip]:[local_port];branch=[branch]
    From: sipp <sip:sipp@[local_ip]:
[local_port]>;tag=[pid]SIPpTag09[call_number]
    To: [service] <sip:[service]@[remote_ip]:
[remote_port]>[peer_tag_param]
    Call-ID: [call_id]
    CSeq: 2 BYE
    Contact: sip:sipp@[local_ip]:[local_port]
    Max-Forwards: 70
    Subject: Performance Test
    Content-Length: 0

  ]]>
</send>

<recv response="200" crlf="true">
</recv>

<!-- definition of the response time repartition table (unit is ms)-->
<ResponseTimeRepartition value="10, 20, 30, 40, 50, 100, 150, 200"/>

<!-- definition of the call length repartition table (unit is ms) -->
<CallLengthRepartition value="10, 50, 100, 500, 1000, 5000, 10000"/>
</scenario>

```

Αποτελέσματα και συμπεράσματα δοκιμών υπηρεσιών

Ρυθμός κλήσεων ανά δευτερόλεπτο	Αριθμός ταυτόχρονων κλήσεων	Φόρτος CPU συστήματος (%)	Λάθη - επαναλήψεις πακέτων	Προσέγγιση Ποιότητας Εμπειρίας (QoE) [1]
0	1	1,3	0	5
1	6	8,1	0	5
2	11	17,4	0	5
3	16	24,5	0	5

4	21	33,4	0	5
5	26	45,8	0	5
6	31	53,1	0	5
7	36	62,6	0	4
8	41	75,8	0	4
9	46	85,7	>100	3

[1] Η παραπάνω Ποιότητα Εμπειρίας είναι εντελώς υποκειμενική, αποτελεί απλά μια εκτίμηση της ποιότητας φωνής κατά την διάρκεια της δοκιμής.

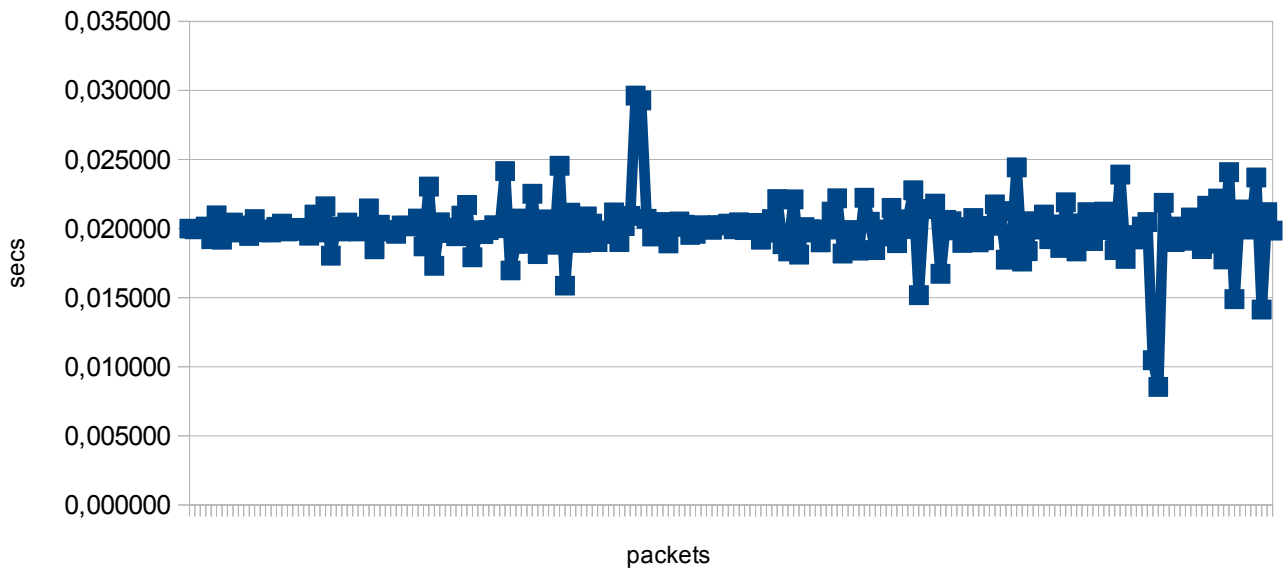
Για να αναλύσουμε λίγο περισσότερο το λόγους της κακής ποιότητας εμπειρίας σε συνθήκες υψηλού φόρτου του συστήματος χρησιμοποιήσαμε την εφαρμογή Wireshark (www.wireshark.org) για να καταγράψουμε τους χρόνους άφιξης των πακέτων στον υπολογιστή μέτρησης του ελέγχου ποιότητας (QoE). Παρουσιάζουμε τα αποτελέσματα σε τρεις φάσεις, την αρχική όπου η ποιότητα εμπειρίας ήταν άριστη και το σύστημα χωρίς φόρτο, μια ενδιάμεση όπου η ποιότητα εξακολουθούσε να είναι πολύ καλή με το σύστημα να έχει περίπου 50% του φόρτου και τέλος με υψηλό φόρτο όταν η ποιότητα εμπειρίας έχει πέσει κάτω από αποδεκτά όρια.

Δεδομένου ότι για την επικοινωνία επιλέξαμε κωδικοποίηση G711 alaw, είναι γνωστό ότι αυτός ο codec έχει δειγματοληψία 8.000 Hz και κάθε πακέτο περιέχει 20ms φωνής, άρα έχουμε ένα σταθερό ρυθμός αποστολής ένα πακέτο κάθε 20ms. Βεβαίως ο φόρτος του συστήματος αποστολής (CPU, network, disk IO), η κατάσταση του δικτύου και τυχών buffering του αποδέκτη επηρεάζουν το τελικό χρόνο άφιξης των πακέτων, που σε ιδανικές συνθήκες πρέπει να είναι ένα πακέτο κάθε 20ms. Στα διαγράμματα που ακολουθούν ο κατακόρυφος άξονας Y αποτυπώνει τη διαφορά του χρόνου (ΔX) άφιξης ενός πακέτου από το προηγούμενο του:

$$\Delta X_i = X_i - X_{i-1} \quad \text{όπου } X_i \text{ ο χρόνος άφιξης του πακέτου } i, \text{ με } i = 2, \dots, 200.$$

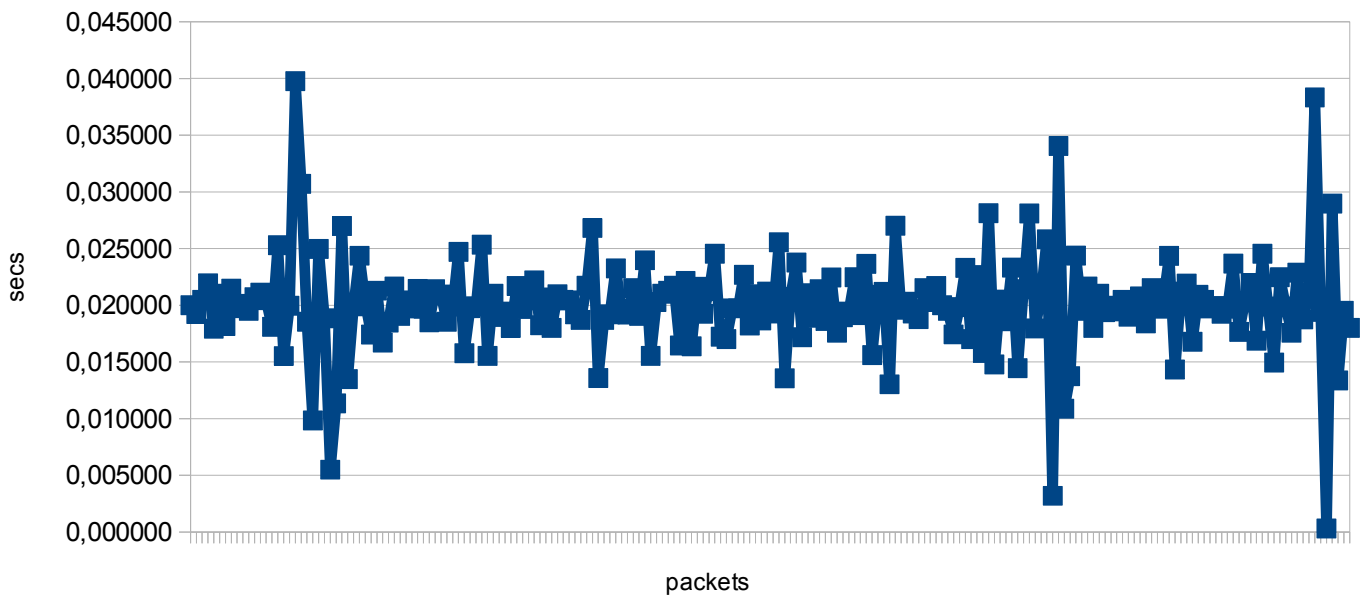
Για το πρώτο πακέτο θέσαμε $\Delta X_1 = 0,0020 \text{ secs}$ (20ms), δηλαδή όσος ο αναμενόμενος ρυθμός αποστολής των πακέτων.

Στα διαγράμματα αποτυπώνεται πολύ παραστατικά η άμεση σχέση της διαφοράς του χρόνου άφιξης με την ποιότητα, όσο μεγαλώνουν και πληθαίνουν οι τιμές του ΔΧ τόσο χειροτερεύει η ποιότητα της υπηρεσίας μας.

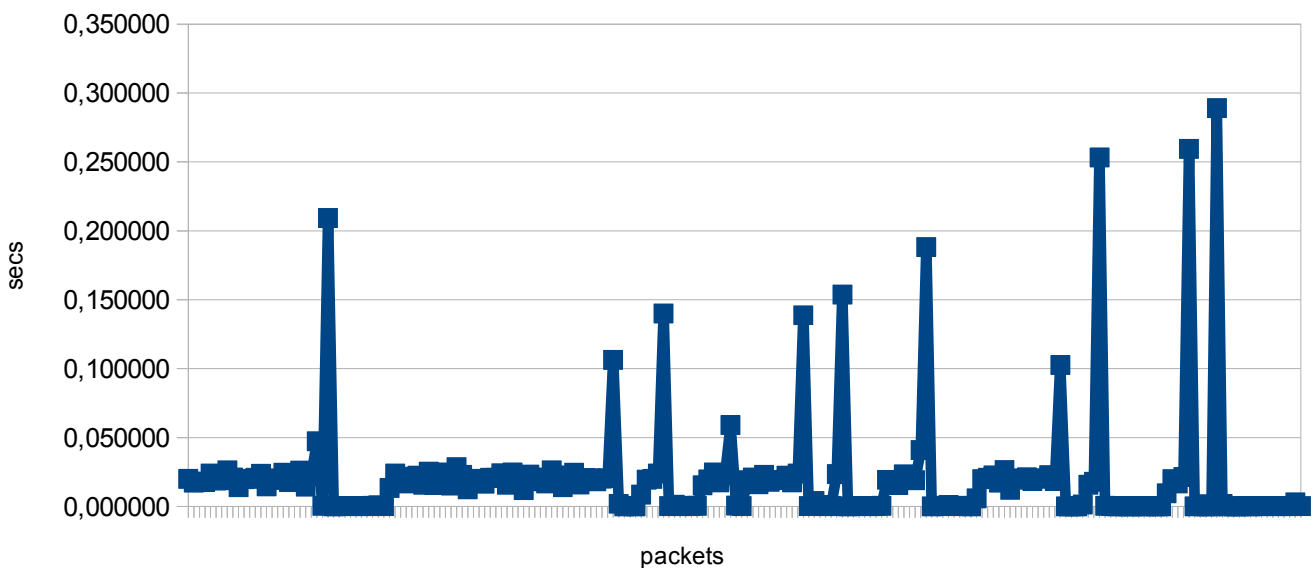


Στο πρώτο διάγραμμα βλέπουμε μια σχεδόν άριστη συγκέντρωση των τιμών ΔΧ γύρω από τον νοητό οριζόντιο άξονα των 20ms. Μερικά πακέτα (περίπου 2%) παρουσιάζουν μεγάλες αποκλίσεις, που όμως δεν υπερβαίνουν τα 10ms.

Στο δεύτερο διάγραμμα, με μέσο φόρτο συστήματος, βλέπουμε επίσης μια μεγάλη συγκέντρωση των τιμών ΔΧ γύρω από τον νοητό οριζόντιο άξονα των 20ms, με μικρότερη όμως πυκνότητα. Έχουμε δηλαδή τις περισσότερες τιμές ΔΧ να αποκλίνουν περίπου 5ms από την ιδανική τιμή των 20ms. Μερικά πακέτα (περίπου 4%) παρουσιάζουν συγκριτικά πολύ μεγαλύτερες αποκλίσεις, προς τα πάνω μέχρι τα 40ms ή προς τα κάτω, δηλαδή φτάνουν σχεδόν ταυτόχρονα.

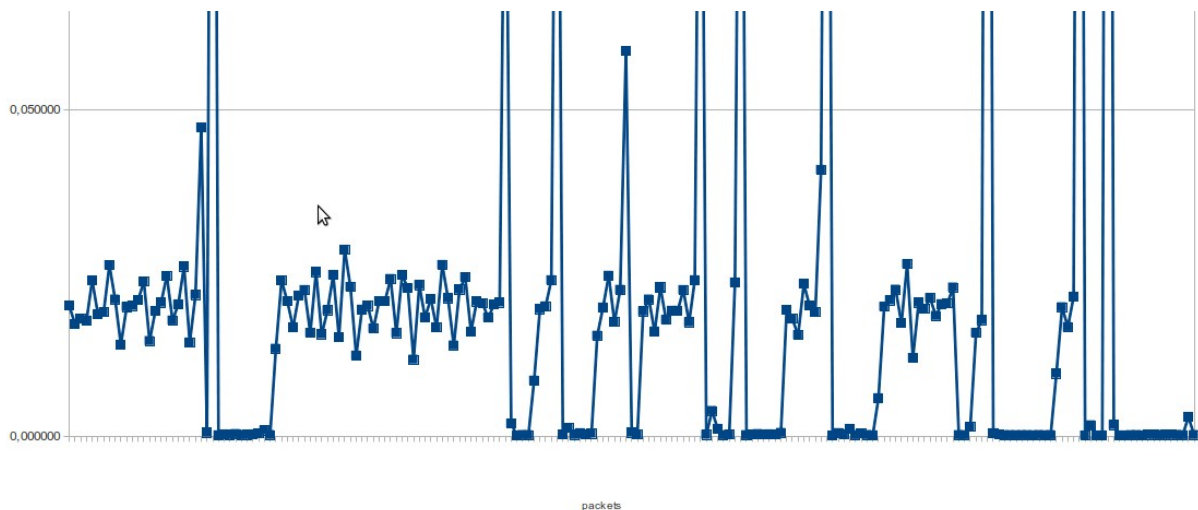


Στο τρίτο και τελευταίο διάγραμμα, όπου έχουμε υψηλό φόρτο συστήματος, βλέπουμε επίσης μια μεγάλη συγκέντρωση των τιμών ΔX γύρω από τον νοητό οριζόντιο άξονα των 20ms.



Παρατηρούμε όμως αρκετά πακέτα (συνολικά 10, άρα περίπου 5%) με χρόνους απόκλισης πάνω από 100ms, τα οποία προφανώς απορρίπτονται, επομένως είναι ένας λόγος κακής ποιότητας. Σημειώνουμε πάντως ότι επειδή είναι σχετικά μικρός ο αριθμός τους δεν επιφέρουν σημαντικό πρόβλημα στην ποιότητα.

Θα συνεχίσουμε την όμως την ανάλυση μεγεθύνοντας το κάτω μέρος του διαγράμματος για να δούμε τι συμβαίνει με τα πακέτα που έχουν σχεδόν μηδενική διαφορά χρόνου άφιξης, φτάνουν σχεδόν ταυτόχρονα.



Έχουμε υπολογίσει ότι περίπου 68 πακέτα, ή 34%, έχουν διαφορά χρόνου άφιξης μικρότερο από 1ms. Είναι προφανές ότι αυτός είναι ο σημαντικότερος λόγος της κακής ποιότητα, καθώς ο αποδέκτης αναγκάζεται να απορρίψει πολλά, αν όχι όλα, από αυτά τα πακέτα ώστε να διατηρήσει σταθερό το ρυθμό μετατροπής τους σε φωνή. Παρατηρούμε επίσης ότι αυτά τα πακέτα φτάνουν σε ομάδες, σε αντίθεση με τα πακέτα με πολύ μεγάλη διαφορά χρονικής καθυστέρησης που είναι μεμονωμένα. Η αιτία βρίσκεται στο στο σύστημα που δημιουργεί και εκπέμπει τα πακέτα, το οποίο αδυνατεί να ανταποκριθεί, λόγω υπερφόρτωσης της CPU, στην απαίτηση για σταθερό ρυθμό παραγωγής των πακέτων.

Κεφάλαιο 7

Επίλογος

Στον επίλογο αυτό επιλέξαμε να συνοψίσουμε μια απάντηση στο βασικότερο ερώτημα της παρούσας διατριβής: μπορούν οι σημερινές (1ο εξάμηνο του 2013) εικονικές μηχανές, που προσφέρονται ως υπηρεσία από τα δημόσια υπολογιστικά νέφη (public cloud IaaS), να χρησιμοποιηθούν για την φιλοξενία υπηρεσιών πραγματικού χρόνου;

Η απάντηση μας είναι κατηγορηματικά θετική, και η πρόβλεψη μας είναι ότι σύντομα θα αποτελεί και μοναδική επιλογή. Η απάντηση που δίνουμε είναι τεκμηριωμένη, βασισμένη τόσο στην θεωρητική ανάλυση των ζητημάτων όσο και στην ανάπτυξη και δοκιμή ενός πραγματικού περιβάλλοντος προηγμένων υπηρεσιών τηλεφωνίας VoIP. Το περιβάλλον που αναπτύξαμε φιλοξενήθηκε εξολοκλήρου σε εικονική μηχανή και, παράλληλα, μπορεί να εκμεταλλευτεί με τον καλύτερο δυνατό τρόπο τα πλεονεκτήματα του υπολογιστικού νέφους.

Καταρχήν διαπιστώσαμε ότι οι σύγχρονες τεχνολογίες υλικού των υπολογιστικών υποδομών - επιπέδου Κέντρων Δεδομένων -, σε συνδυασμό με την εξέλιξη των λειτουργικών συστημάτων (kernels) και των λογισμικών εικονοποίησης (hypervisors), έχουν ξεπεράσει τόσο τα πρώιμα προβλήματα αξιόπιστου πολλαπλασιασμού του χρονισμού του συστήματος όσο και τα ζητήματα ανταγωνισμού στην χρήση των φυσικών πόρων και μπορούν να αντιμετωπίσουν τις διαφορετικές ανάγκες των εικονικών μηχανών. Επομένως δεν υφίστανται ζητήματα καταλληλότητας ή απόδοσης των υπολογιστικών δυνατοτήτων των εικονικών μηχανών για τις υπηρεσίες πραγματικού χρόνου.

Τι γίνεται όμως με την ανάγκη για υψηλή διαθεσιμότητα των υπηρεσιών πραγματικού χρόνου; Είναι σαφές ότι τα σύγχρονα Κέντρα Δεδομένων (Data centers) μπορούν να προσφέρουν διαθεσιμότητα που πλησιάζει το 100%. Υπάρχουν μάλιστα πάροχοι που “τολμούν” να υπογράφουν SLA με διαθεσιμότητα 100%. Άρα δεν τίθεται θέμα διαθεσιμότητας, είναι μάλιστα βέβαιο ότι κανένας οργανισμός με ιδιόκτητη παραδοσιακή υποδομή δεν μπορεί να ισχυριστεί ότι είναι σε θέση να εξασφαλίσει 100% διαθεσιμότητα στις δικές του υποδομές.

Ανασταλτικός παράγοντας, για την υιοθέτηση τέτοιων λύσεων, είναι σήμερα η περιορισμένη προσφορά, άρα και το πολύ υψηλό κόστος, πραγματικά ευρυζωνικών υπηρεσιών, δηλαδή υπηρεσιών δικτύου πρόσβασης σύγχρονης μετάδοσης τουλάχιστον

10Mbps (download και upload). Είναι φανερό ότι οι υπηρεσίες ADSL είναι εντελώς ακατάλληλες για αυτό το σκοπό. Το περιορισμένο εύρος ζώνης προς το δίκτυο παρόχου (upload 1 έως 2 Mbps) δεν μπορεί να υποστηρίξει ικανοποιητικά μεγάλο αριθμό ταυτόχρονων κλήσεων, όταν μάλιστα απουσία μηχανισμών QoS, οι υπηρεσίες πραγματικού χρόνου ανταγωνίζονται για το διαθέσιμο εύρος του δικτύου, με τους ίδιους όρους best effort, με όλες τις άλλες υπηρεσίες.

Σχετικά με τα θέματα ασφάλειας, καταρχήν θα πρέπει να αποδεχθούμε την ιδέα φιλοξενίας των δεδομένων, της οποιασδήποτε υπηρεσίας μας, στις υποδομές υπολογιστικού νέφους ενός παρόχου. Να αποδεχθούμε δηλαδή την απώλεια ελέγχου των δεδομένων, αντισταθμίζοντάς την όμως με σχετικές ρυθμίσεις στο συμβόλαιο παροχής των υπηρεσιών και, έχοντας επίγνωση των κινδύνων, με τεχνικές προστασίας των δεδομένων μας. Είδαμε για παράδειγμα πως το πρωτόκολλο SIP μας επιτρέπει να προστατεύσουμε τους κωδικούς των χρηστών με ένα πολύ απλό τρόπο: αποθηκεύοντας και διακινώντας τους σε κρυπτογραφημένη μορφή (MD5 authentication). Για ακόμα περισσότερη ασφάλεια, μπορούμε να χρησιμοποιήσουμε το Secure SIP (SIPS, RFC 3261) όταν διαθέτουμε τερματικές συσκευές με SIP user agents που το υποστηρίζουν, η ιδέα της επέκτασης του πρωτοκόλλου είναι αντίστοιχη με αυτή του HTTPS.

Όσον αφορά στην Ποιότητα Υπηρεσίας (QoS), δυστυχώς δεν υπάρχει σήμερα τεχνο-οικονομικά αποδεκτός τρόπος που να μας επιτρέπει να την συμπεριλάβουμε σε μια υλοποίηση των υπηρεσιών μας. Ούτε οι πάροχοι υπηρεσιών υπολογιστικού νέφους, αλλά ούτε οι πάροχοι δικτύου πρόσβασης προσφέρουν Ποιότητα Υπηρεσίας (QoS). Συζητήσαμε όμως ότι η εφαρμογή τεχνικών Ποιότητα Υπηρεσίας (QoS) στο εσωτερικό δίκτυο ενός οργανισμού εξακολουθεί να έχει αξία για το σύνολο των κλήσεων που παραμένουν εντός του δικτύου του οργανισμού (intranet). Άσχετα αν, για αυτές τις κλήσεις, το SIP ταξιδεύει στο Διαδίκτυο, το RTP τους, που ευθύνεται για την ποιότητα της υπηρεσίας, παραμένει εντός του εσωτερικού δικτύου και μπορούμε να του εξασφαλίσουμε το απαραίτητο εύρος ζώνης και μικρή καθυστέρηση.

Τέλος, παρουσιάσαμε μια αρχιτεκτονική ενός ολοκληρωμένου συστήματος επικοινωνίας που διαχώρισε την σηματοδότηση (SIP Proxy) από τις υπηρεσίες (Media Services) και μπορεί να προσφέρει το σύνολο των γνωστών τηλεφωνικών υπηρεσιών:

τηλεφωνικές κλήσεις, Παρουσία (presence), Άμεσο Μήνυμα (Instant Messaging), Συνδιάσκεψη (voice conference), Φωνητικό Μήνυμα (Voice mail), Fax σε Email, IVR.

Η σπουδαιότητα της αρχιτεκτονικής αυτού του συστήματος είναι ότι μπορεί να υλοποιηθεί σε μία εικονική μηχανή, για τις ανάγκες ενός μικρού οργανισμού π.χ. μιας Μικρομεσαίας Επιχείρισης αλλά εξίσου καλά να υλοποιηθεί σε ένα πολύ μεγάλο αριθμό εικονικών μηχανών για τις ανάγκες μεγάλων οργανισμών ή παρόχων, με εκατοντάδες χιλιάδες χρήστες. Δείξαμε με πιο τρόπο ακριβώς κλιμακώνεται κάθε επιμέρους υποσύστημα, προσθέτοντας εικονικές μηχανές, ώστε να ικανοποιεί σταδιακά τις νέες ανάγκες καθώς θα αυξάνεται η ζήτηση των υπηρεσιών.

Κλείνουμε αυτό τον επίλογο με μερικά αριθμητικά στοιχεία. Σε μια εικονική μηχανή του ΕΔΕΤ (vima.gnet.gr) με μια vCPU (ένα core), 2 GB RAM και 40 GB δίσκο και μια υλοποίηση του συστήματος εξολοκλήρου με χρήση λογισμικού ανοικτού κώδικα (CentOS, openSIPS, Asterisk, MySQL, κά) διαπιστώσαμε ότι το συγκεκριμένο σύστημα μπορούσε να ανταποκριθεί μέχρι 3.000 καταχωρήσεις (SIP REGISTER) ανά δευτερόλεπτο, να διεκπεραιώσει μέχρι 900 κλήσεις ανά δευτερόλεπτο (SIP INVITES) και να υποστηρίξει μέχρι 40 ταυτόχρονες κλήσεις με διαχείριση του μέσου (RTP). Μπορούμε να πούμε ότι ένα τέτοιο σύστημα είναι ικανό να ανταποκριθεί στις ανάγκες ενός μικρού οργανισμού, με λίγες εκατοντάδες χρήστες, και το κόστος μίσθωσης μιας τέτοιας μηχανής σε πάροχο υπηρεσιών υπολογιστικού νέφους δεν θα υπερβαίνει σήμερα τα 500 Ευρώ το χρόνο!

Βιβλιογραφία

1. <http://www.opensips.org/>
2. <http://www.asterisk.org/>
3. The NIST (National Institute of Standards and Technology) Definition of Cloud Computing, <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
4. Cloud Computing Security and Privacy Issues, http://www.hepis.gr/el/activities/Documents/CEPIS_Cloud_Computing_Security_v172.pdf
5. RFC2782, A DNS RR for specifying the location of services (DNS SRV)
6. <http://www.voip-info.org/wiki/view/DNS+SRV>
7. <http://www.voip-info.org/wiki/view/QoS>
8. <http://www.voip-info.org/wiki/view/Asterisk+and+Virtual+Private+Servers>
9. <http://www.telephreak.org/papers/vpa/>
10. Timekeeping best practices for Linux guests, http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1006427
11. <https://wiki.asterisk.org/wiki/display/AST/Timing+Interfaces>
12. Grdner, <http://www.gartner.com/technology/reprints.do?id=1-1B2IRYF&ct=120626&st=sg>
13. Cloud IaaS SLAs can be meaningless, by Lydia Leong, http://blogs.gartner.com/lydia_leong/2012/12/05/cloud-iaas-slas-can-be-meaningless/
14. <http://aws.amazon.com/ec2-sla/>
15. <https://www.hpcloud.com/sl>
16. <http://www.opsource.net/Services/Cloud-Hosting/Service-Level-Agreement>
17. Wireshark. www.wireshark.org

Παράρτημα Α

A.1 Οδηγίες εγκατάστασης συστήματος

1. Εγκατάσταση του Λειτουργικού Συστήματος CentOS 5.9 64bit
2. Εγκατάσταση βασικών πακέτων λογισμικού για το ολοκληρωμένο σύστημα

2.1 Εγκατάσταση PHP

```
# yum install php
```

2.2 Εγκατάσταση MySQL

```
# yum install php-mysql mysql mysql-server  
# /sbin/chkconfig --levels 235 mysqld on
```

2.3 Εγκατάσταση phpMyAdmin

```
# rpm -Uvh  
http://ftp.ntua.gr/pub/linux/fedora-epel/5/i386/epel-release-5-4.noarch.  
rpm  
# yum install phpmyadmin
```

Επιβολή σύνδεσης SSL

```
# yum -y install mod_ssl  
# vi /etc/phpMyAdmin/config.inc.php  
! add a configuration line at the bottom of the file  
$cfg['ForceSSL'] = true;
```

3. Εγκατάσταση openSIPS

3.1 Εγκατάσταση εξαρτήσεων (dependencies)

```
# yum install bison curl curl-devel db4.i386 db4-devel expat-devel flex
gcc gcc-c++ kernel-devel libtool libtool-ltdl libtool-ltdl-devel libxml2
libxml2-devel lynx mod_dav_svn mysql mysql-devel ncurses ncurses-devel
ntp openldap-devel openssl openssl-devel pcre-devel sqlite sqlite-devel
subversion zlib-devel
```

3.2 Εγκατάσταση εφαρμογής openSIPS

```
# mkdir -p opensips
# cd opensips/
# wget
http://opensips.org/pub/opensips/latest/src/opensips-1.8.2-svn9715_src.t
ar.gz
# tar -zxvf opensips-1.8.2-svn9715_src.tar.gz
# cd opensips-1.8.2-tls/
```

Πως να αποκλείσουμε πακέτα που δεν σκοπεύουμε να χρησιμοποιήσουμε από την εγκατάσταση

```
# vi Makefile.conf
exclude_modules?= aaa_radius b2b_logic cachedb_cassandra
cachedb_memcached cachedb_redis carrierroute cpl-c db_berkeley db_http
db_oracle db_postgres db_unixodbc event_rabbitmq h350 regex identity
jabber json ldap lua httpd mi_http mi_xmlrpc mmgeoip osp perl perlvdb
python rls snmpstats

# make prefix=/ all
# make prefix=/ install
```

3.3 Παραμετροποίηση openSIPS

```
# vi /etc/opensips/opensipsctlrc
SIP_DOMAIN=iptel.no-ip.org
DBENGINE=MYSQL
DBHOST=localhost
DBNAME=opensips
DBRWUSER=opensips
DBRWPW=<password>
DBROOTUSER="root"
INSTALL_EXTRA_TABLES=yes
INSTALL_PRESENCE_TABLES=yes
```

Δημιουργία της Βάσης Δεδομένων του openSIPS

```
# opensipsdbctl create
```

Για αυτόματη εκκίνηση των Services

```
# cd /usr/src/opensips/opensips-1.8.2-tls/packaging/fedora
# cp opensips.init /etc/init.d/opensips
# chkconfig opensips on
```

Καθορισμός path εκτελέσιμων openSIPS

```
# vi /etc/init.d/opensips
osser=/sbin/$prog
```

Καθορισμός παραμέτρων λειτουργίας του openSIPS γίνεται στο αρχείο διαμόρφωσης

```
# vi /etc/opensips/opensips.cfg
```

4. Εγκατάσταση εφαρμογής Asterisk

```
# cd /usr/src
# mkdir asterisk
# cd asterisk
# wget
http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-11-current
.tar.gz
# wget
http://downloads.asterisk.org/pub/telephony/dahdi-linux/dahdi-linux-2.6.
2-rc1.tar.gz
# wget
http://downloads.asterisk.org/pub/telephony/dahdi-tools/dahdi-tools-curr
ent.tar.gz
# tar -zxvf dahdi-linux-2.6.2-rc1.tar.gz
# tar -zxvf asterisk-11-current.tar.gz
# tar -zxvf dahdi-tools-current.tar.gz

# cd dahdi-linux-2.6.2-rc1
# make
# make install

# cd ../dahdi-tools-2.6.1/
# ./configure
# make menuconfig
# make
# make install
```

```
# make config

# cd ../asterisk-11.2.1/
./configure --libdir=/usr/lib64

# ./configure
# make
# make install
# make config
# make samples
```

4.1 Παραμετροποίηση υπηρεσιών SIP (/etc/asterisk/sip.conf)

SIP Trunk με τον OpenSIPs

SIP trunk με υπηρεσία ePhone για κλήσεις προς/από PSTN

4.2 Παραμετροποίηση υπηρεσιών (IVR, Voicemail, FAX, κλπ), χρησιμοποιούμε τεχνική Real time

```
# vi /etc/asterisk/extensions.conf
[<context-name>]
switch => Realtime/<service-name>
```

5. Ασφάλεια πρόσβασης στο σύστημα

1. Επιτρέπουμε τα ακόλουθα το σύστημα μας

- ping (icmp) για έλεγχο
- οτιδήποτε έχει ξεκινήσει από το σύστημα μας (ESTABLISHED)
- SSH (port 22) για απομακρυσμένη διαχείριση
- HTTPS (port 443) για την εφαρμογή phpMyAdmin (διαχείριση ΒΔ)
- UDP & TCP SIP (port 5060) για την εφαρμογή openSIPS
- UDP & TCP SIP (port 5062) για την εφαρμογή Asterisk (σύνδεση με PSTN)

2. Απαγορεύουμε οτιδήποτε άλλο

```
# vi /etc/sysconfig/iptables
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j
ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j
ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 5060
-j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 5060
-j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m udp -p udp --dport 5062
-j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 5062
-j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

A.2 Εγκατάσταση και προσαρμογή SIPP

Για την εγκατάστασή του θα πρέπει να ακολουθηθούν τα παρακάτω βήματα. Αρχικά εγκαθιστούμε από το package repository του λειτουργικού συστήματος τα Dependencies για την εγκατάσταση του Sipp όπως φαίνεται παρακάτω.

```
# yum install -y openssl openssl-devel gcc gcc-c++ ncurses
ncurses-devel libpcap libpcap-devel
```

Έπειτα δημιουργούμε έναν φάκελο στο /usr/src με όνομα sipp.

```
# mkdir /usr/src/sipp
```

Κατεβάζουμε τα sources του sipp και το κάνουμε Compile με το switch pcapplay_oss1 ώστε να υποστηρίζει playback αρχείων ήχου pcap και digest authentication, όπως φαίνεται παρακάτω.

```
# cd /usr/src/sipp
# wget
http://sourceforge.net/projects/sipp/files/sipp/3.3/sipp-3.3.tar.gz/download
# tar -zxvf sipp-3.3.tar.gz
# cd sipp-3.3
# make pcapplay_oss1
```

Σημ. Για εγκατάσταση των εργαλείων

```
yum groupinstall "Development Tools"
ubuntu
apt-get install libssl-dev libpcap-dev libncurses5-dev
mkdir /usr/src/sipp
cd /usr/src/sipp
wget http://sourceforge.net/projects/sipp/files/sipp/3.3/sipp-3.3.tar.gz
tar -zxvf sipp-3.3.tar.gz
cd sipp-3.3
make pcapplay_oss1
```

A.3 Οθόνες δοκιμών υπερφόρτωσης

Στη συνέχεια παραθέτουμε τις βασικές οθόνες των δοκιμών των τριών κατηγοριών

1. Λειτουργία καταχώρησης (registration).
2. Πραγματοποίηση κλήσεων (invite).
3. Προηγμένες Υπηρεσίες (media services).

Οι οθόνες περιέχουν λεπτομέρειες των δοκιμών που δεν έχουν αναφερθεί στο σχετικό κεφάλαιο παρουσίασης των δοκιμών και των αποτελεσμάτων τους και μπορούν να φανούν χρήσιμες σε μια προσεκτικότερη ανάλυση.

A.3.1 Οθόνες δοκιμών “Λειτουργία καταχώρησης (registration)”

```

root@mk-vm:~
File Edit View Search Terminal Help
top - 21:44:07 up 25 days, 7:42, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 93 total, 2 running, 91 sleeping, 0 stopped, 0 zombie
Cpu(s): 13.4%us, 6.3%sy, 0.0%ni, 80.0%id, 0.0%wa, 0.2%hi, 0.2%si,
Mem: 1922464k total, 1229924k used, 692540k free, 167764k buffers
Swap: 4128760k total, 0k used, 4128760k free, 869440k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 28661 root        20   0 61620 9704 2620 R 40.0  0.5   0:04.51 sipp
    1 root        20   0 19228 1496 1220 S  0.0  0.1   0:02.78 init
    2 root        20   0   0     0   0   S  0.0  0.0   0:00.00 kthreadd
    3 root        RT   0   0     0   0   S  0.0  0.0   0:00.19 migration/0
    4 root        20   0   0     0   0   S  0.0  0.0   0:00.46 ksoftirqd/0
    5 root        RT   0   0     0   0   S  0.0  0.0   0:00.00 migration/0
    6 root        RT   0   0     0   0   S  0.0  0.0   0:04.19 watchdog/0
    7 root        RT   0   0     0   0   S  0.0  0.0   0:01.73 migration/1
    8 root        RT   0   0     0   0   S  0.0  0.0   0:00.00 migration/0
    9 root        20   0   0     0   0   S  0.0  0.0   0:00.00 migration/0
   10 root        RT   0   0     0   0   S  0.0  0.0   0:00.00 migration/0
   11 root        20   0   0     0   0   S  0.0  0.0   0:00.00 migration/0
   12 root        20   0   0     0   0   S  0.0  0.0   0:00.39 events/0
   13 root        20   0   0     0   0   S  0.0  0.0   0:00.00 khelper
   14 root        20   0   0     0   0   S  0.0  0.0   0:00.00 kthread
   15 root        20   0   0     0   0   S  0.0  0.0   0:03.13 kblockd/0
   16 root        20   0   0     0   0   S  0.0  0.0   0:00.00 kacpid
    
```

```

root@kalogirou:/etc/opensips
File Edit View Search Terminal Help
top - 21:44:23 up 23 days, 48 min, 2 users, load average: 0.22, 0.07, 0.07
Tasks: 87 total, 1 running, 86 sleeping, 0 stopped, 0 zombie
Cpu(s): 27.4%us, 12.7%sy, 0.0%ni, 46.8%id, 0.0%wa, 1.3%hi, 11.7%si,
Mem: 2058808k total, 1013760k used, 1045048k free, 212732k buffers
Swap: 2064376k total, 0k used, 2064376k free, 558828k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 2147 mysql      18   0 269m  48m 4472 S 18.6  2.4 287:24.93 mysqld
 7622 root       15   0 142m 4744 3844 S  7.7  0.2  0:18.25 opensips
 7626 root       15   0 142m 4744 3844 S  7.7  0.2  0:18.67 opensips
 7624 root       15   0 142m 4744 3844 S  7.0  0.2  0:18.97 opensips
 7620 root       15   0 142m 4948 4044 S  6.0  0.2  0:17.70 opensips
 175 root       15   0   0     0   0   S  0.3  0.0  2:14.79 pdflush
 465 root       15   0 592m  32m 9704 S  0.3  1.6  2:44.25 asterisk
18256 root       15   0 18226 1264 1080 S  0.3  0.1 20:55.14 mpg123
    0 root        0.0  0.0 0:02.95 init
    0 root        0.0  0.0 0:00.00 migration/0
    0 root        0.0  0.0 0:00.00 ksoftirqd/0
    0 root        0.0  0.0 0:00.00 watchdog/0
    0 root        0.0  0.0 0:00.39 events/0
    0 root        0.0  0.0 0:00.00 khelper
    0 root        0.0  0.0 0:00.00 kthread
    0 root        0.0  0.0 0:03.13 kblockd/0
    0 root        0.0  0.0 0:00.00 kacpid
    
```

```

root@mk-vm:/usr/src/sipp/sipp-3.3
File Edit View Search Terminal Help

----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
1500.0(0 ms)/1.000s 5060 11.02 s 16530 62.217.124.222:5060(UDP)

1503 new calls during 1.001 s period 0 ms scheduler resolution
26 calls (limit 4500) Peak was 41 calls, after 2 s
0 Running, 16528 Paused, 494 Woken up
0 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets

Messages Retrans Timeout Unexpected-Msg
REGISTER -----> 16527 0 0
401 <----- 16515 0 0
REGISTER -----> 16515 0 0
200 <----- E-RTD1 16504 0 0
----- [+]*[/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
    
```

```

root@mk-vm:~
File Edit View Search Terminal Help
top - 21:44:28 up 25 days, 7:42, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 93 total, 1 running, 92 sleeping, 0 stopped, 0 zombie
Cpu(s): 11.1%us, 3.2%sy, 0.0%ni, 85.6%id, 0.0%wa, 0.0%hi, 0.2%si,
Mem: 1922464k total, 1244628k used, 677836k free, 167764k buffers
Swap: 4128760k total, 0k used, 4128760k free, 869440k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 28661 root        20   0 75764 23m 2648 S 32.6  1.2   0:14.96 sipp
    1 root        20   0 19228 1496 1220 S  0.0  0.1   0:02.78 init
    2 root        20   0   0     0   0   S  0.0  0.0   0:00.00 kthreadd
    3 root        RT   0   0     0   0   S  0.0  0.0   0:00.19 migration/0
    4 root        20   0   0     0   0   S  0.0  0.0   0:00.46 ksoftirqd/0
    5 root        RT   0   0     0   0   S  0.0  0.0   0:00.00 migration/0
    6 root        RT   0   0     0   0   S  0.0  0.0   0:04.19 watchdog/0
    7 root        RT   0   0     0   0   S  0.0  0.0   0:01.73 migration/0
    8 root        RT   0   0     0   0   S  0.0  0.0   0:00.00 ksoftirqd/0
    9 root        20   0   0     0   0   S  0.0  0.0   0:00.00 kthreadd
   10 root        RT   0   0     0   0   S  0.0  0.0   0:00.00 ksoftirqd/0
   11 root        20   0   0     0   0   S  0.0  0.0   0:00.00 kthreadd
   12 root        20   0   0     0   0   S  0.0  0.0   0:00.00 ksoftirqd/0
   13 root        20   0   0     0   0   S  0.0  0.0   0:00.00 kthreadd
   14 root        20   0   0     0   0   S  0.0  0.0   0:00.00 ksoftirqd/0
   15 root        20   0   0     0   0   S  0.0  0.0   0:00.00 kthreadd
   16 root        20   0   0     0   0   S  0.0  0.0   0:00.00 ksoftirqd/0
    
```

```

root@kalogirou:/etc/opensips
File Edit View Search Terminal Help
top - 21:44:44 up 23 days, 48 min, 2 users, load average: 0.24, 0.08, 0.08
Tasks: 87 total, 6 running, 81 sleeping, 0 stopped, 0 zombie
Cpu(s): 37.0%us, 16.0%sy, 0.0%ni, 36.7%id, 0.0%wa, 1.3%hi, 9.0%si,
Mem: 2058808k total, 1013760k used, 1045048k free, 212732k buffers
Swap: 2064376k total, 0k used, 2064376k free, 558828k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 2147 mysql     18   0 269m  48m 4472 S 26.0  2.4 287:28.87 mysqld
 7622 root       15   0 142m 4744 3844 R  8.7  0.2  0:20.13 opensips
 7626 root       15   0 142m 4744 3844 R  8.7  0.2  0:20.43 opensips
 7620 root       15   0 142m 4948 4044 R  8.3  0.2  0:19.47 opensips
 7624 root       15   0 142m 4744 3844 R  8.0  0.2  0:20.79 opensips
18256 root       15   0 19336 1364 1080 R  0.3  0.1 20:55.20 mpg123
    1 root       15   0 10372  700  588 S  0.0  0.0  0:02.95 init
    2 root       RT   5   0   0     0   0   S  0.0  0.0  0:00.00 migration/0
    3 root       RT   0   0   0     0   0   S  0.0  0.0  0:00.00 ksoftirqd/0
    4 root       RT   0   0   0     0   0   S  0.0  0.0  0:00.00 watchdog/0
    5 root       RT   0   0   0     0   0   S  0.0  0.0  0:00.39 events/0
    6 root       RT   0   0   0     0   0   S  0.0  0.0  0:00.00 khelper
    7 root       RT   0   0   0     0   0   S  0.0  0.0  0:00.00 kthread
    8 root       RT   0   0   0     0   0   S  0.0  0.0  0:03.13 kblockd/0
    9 root       RT   0   0   0     0   0   S  0.0  0.0  0:00.00 kacpid
   10 root       RT   0   0   0     0   0   S  0.0  0.0  0:00.00 cqueue/0
   11 root       RT   0   0   0     0   0   S  0.0  0.0  0:00.00 khubd
    
```

```

root@mk-vm:/usr/src/sipp/sipp-3.3
File Edit View Search Terminal Help

----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
2000.0(0 ms)/1.000s 5060 33.05 s 55010 62.217.124.222:5060(UDP)

2006 new calls during 1.001 s period 0 ms scheduler resolution
38 calls (limit 6000) Peak was 49 calls, after 27 s
0 Running, 54956 Paused, 545 Woken up
0 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets

Messages Retrans Timeout Unexpected-Msg
REGISTER -----> 55006 0 0
401 <----- 54991 0 0
REGISTER -----> 54991 0 0
200 <----- E-RTD1 54972 0 0
----- [+]*[/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
    
```

```

root@mk-vm:~
File Edit View Search Terminal Help
top - 21:44:52 up 25 days, 7:43, 2 users, load average: 0.00, 0.00, 0
Tasks: 93 total, 2 running, 91 sleeping, 0 stopped, 0 zombie
Cpu(s): 18.7%us, 9.3%sy, 0.0%ni, 71.8%id, 0.0%wa, 0.0%hi, 0.2%si,
Mem: 1922464k total, 1252392k used, 670072k free, 167764k buffers
Swap: 4128760k total, 0k used, 4128760k free, 869440k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  COMMAND
28661 root        20   0 83436 30m 2648  R 64.5  1.6   0:27.51 sipp
   1 root        20   0 19228 1496 1220  S  0.0  0.1   0:02.78 init
   2 root        20   0   0     0   0   S  0.0  0.0   0:00.00 kthreadd
   3 root        RT   0   0     0   0   S  0.0  0.0   0:00.19 migration/0
   4 root        20   0   0     0   0   S  0.0  0.0   0:00.46 ksoftirqd/0
   5 root        RT   0   0     0   0   S  0.0  0.0   0:00.00 migration/0
   6 root        RT   0   0     0   0   S  0.0  0.0   0:04.19 watchdog/0
   7 root        RT   0   0     0   0   S  0.0  0.0   0:01.73 migration/0
   8 root        RT   0   0     0   0   S  0.0  0.0   0:00.00 migration/0
   9 root        20   0   0     0   0   S  0.0  0.0   0:00.00 ksoftirqd/0
  10 root        RT   0   0     0   0   S  0.0  0.0   0:00.00 watchdog/0
  11 root        20   0   0     0   0   S  0.0  0.0   0:00.39 events/0
  12 root        20   0   0     0   0   S  0.0  0.0   0:00.00 khelper
  13 root        20   0   0     0   0   S  0.0  0.0   0:00.00 kthread
  14 root        20   0   0     0   0   S  0.0  0.0   0:03.13 kblockd/0
  15 root        20   0   0     0   0   S  0.0  0.0   0:00.00 kacpid
  16 root        20   0   0     0   0   S  0.0  0.0   0:00.00 cqueue/0
    
```

```

root@kalogirou:/etc/opensips
File Edit View Search Terminal Help
top - 21:45:11 up 23 days, 49 min, 2 users, load average: 1.05, 0.28,
Tasks: 87 total, 5 running, 82 sleeping, 0 stopped, 0 zombie
Cpu(s): 47.8%us, 22.9%sy, 0.0%ni, 7.0%id, 0.0%wa, 1.7%hi, 20.6%si,
Mem: 2058808k total, 1013636k used, 1045172k free, 212732k buffers
Swap: 2064376k total, 0k used, 2064376k free, 558828k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  COMMAND
2147 mysql      18   0 269m  48m 4472  S 26.6  2.4 287:36.34 mysqld
7622 root       15   0 142m 4744 3844  R 15.0  0.2  0:23.15 opensips
7626 root       15   0 142m 4744 3844  R 14.6  0.2  0:23.53 opensips
7620 root       15   0 142m 4948 4044  S 14.3  0.2  0:22.38 opensips
7624 root       15   0 142m 4744 3844  R 14.3  0.2  0:23.80 opensips
 465 root       15   0 592m  32m 9704  S  0.7  1.6  2:44.33 asterisk
2241 root       18   0 286m  11m 7084  S  0.3  0.6  0:01.11 httpd
 1 root       15   0 10272 700 588  S  0.0  0.0  0:02.95 init
 0 root       0   0   0     0   0   S  0.0  0.0  0:00.00 migration/0
 0 root       0   0   0     0   0   S  0.0  0.0  0:00.00 ksoftirqd/0
 0 root       0   0   0     0   0   S  0.0  0.0  0:00.00 watchdog/0
 0 root       0   0   0     0   0   S  0.0  0.0  0:00.39 events/0
 0 root       0   0   0     0   0   S  0.0  0.0  0:00.00 khelper
 0 root       0   0   0     0   0   S  0.0  0.0  0:00.00 kthread
 0 root       0   0   0     0   0   S  0.0  0.0  0:03.13 kblockd/0
 0 root       0   0   0     0   0   S  0.0  0.0  0:00.00 kacpid
 0 root       0   0   0     0   0   S  0.0  0.0  0:00.00 cqueue/0
    
```

```

root@mk-vm:/usr/src/sipp/sipp-3.3
File Edit View Search Terminal Help

----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
2500.0(0 ms)/1.000s 5060 57.09 s 109767 62.217.124.222:5060(UDP)

2505 new calls during 1.002 s period 0 ms scheduler resolution
44 calls (limit 7500) Peak was 150 calls, after 48 s
0 Running, 72550 Paused, 2322 Woken up
0 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets

Messages Retrans Timeout Unexpected-Msg
REGISTER -----> 109767 0 0
401 <----- 109747 0 0
REGISTER -----> 109747 0 0
200 <----- E-RTD1 109723 0 0
----- [+]*[/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
    
```

```

root@mk-vm:~
File Edit View Search Terminal Help
top - 21:45:13 up 25 days, 7:43, 2 users, load average: 0.00, 0.00, 0
Tasks: 93 total, 1 running, 92 sleeping, 0 stopped, 0 zombie
Cpu(s): 14.0%us, 6.7%sy, 0.0%ni, 79.2%id, 0.0%wa, 0.0%hi, 0.0%si,
Mem: 1922464k total, 1257724k used, 664740k free, 167764k buffers
Swap: 4128760k total, 0k used, 4128760k free, 869440k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  COMMAND
 28661 root        20   0  88832 36m 2648 S 54.6  1.9   0:41.00 sipp
 28659 root        20   0  15020 1288 1008 R  0.3  0.1   0:00.13 top
   1 root        20   0  19228 1496 1220 S  0.0  0.1   0:02.78 init
   2 root        20   0   0     0     0 S  0.0  0.0   0:00.00 kthreadd
   3 root        RT   0   0     0     0 S  0.0  0.0   0:00.19 migration/0
   4 root        20   0   0     0     0 S  0.0  0.0   0:00.46 ksoftirqd/0
   5 root        RT   0   0     0     0 S  0.0  0.0   0:00.00 migration/0
   6 root        RT   0   0     0     0 S  0.0  0.0   0:00.19 watchdog/0
   7 root        RT   0   0     0     0 S  0.0  0.0   0:00.00 ksoftirqd/0
   8 root        RT   0   0     0     0 S  0.0  0.0   0:00.00 watchdog/0
   9 root        20   0   0     0     0 S  0.0  0.0   0:00.00 events/0
  10 root        RT   0   0     0     0 S  0.0  0.0   0:00.00 khelper
  11 root        20   0   0     0     0 S  0.0  0.0   0:00.00 kthread
  12 root        20   0   0     0     0 S  0.0  0.0   0:03.13 kblockd/0
  13 root        20   0   0     0     0 S  0.0  0.0   0:00.00 kacpid
  14 root        20   0   0     0     0 S  0.0  0.0   0:00.00 cqueue/0
  15 root        20   0   0     0     0 S  0.0  0.0   0:00.00 khubd
    
```

```

root@kalogirou:/etc/opensips
File Edit View Search Terminal Help
top - 21:45:30 up 23 days, 49 min, 2 users, load average: 1.27, 0.37,
Tasks: 87 total, 6 running, 81 sleeping, 0 stopped, 0 zombie
Cpu(s): 52.2%us, 23.6%sy, 0.0%ni, 14.6%id, 0.0%wa, 0.7%hi, 9.0%si,
Mem: 2058808k total, 1013636k used, 1045172k free, 212732k buffers
Swap: 2064376k total, 0k used, 2064376k free, 558828k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  COMMAND
 2147 mysql      18   0  269m  48m 4472 S 47.2  2.4  287:42.99 mysqld
 7620 root        15   0  142m 4948 4044 R  9.3  0.2   0:24.32 opensips
 7624 root        15   0  142m 4744 3844 R  9.3  0.2   0:25.81 opensips
 7626 root        15   0  142m 4744 3844 R  9.3  0.2   0:25.50 opensips
 7622 root        15   0  142m 4744 3844 R  9.0  0.2   0:25.20 opensips
18256 root        15   0  19336 1364 1080 S  0.7  0.1  20:55.34 mpg123
   1 root        15   0  10372  700  588 S  0.0  0.0   0:02.95 init
   2 root        RT   5   0     0     0 S  0.0  0.0   0:00.00 migration/0
   3 root        RT   0   0     0     0 S  0.0  0.0   0:00.00 ksoftirqd/0
   4 root        RT   0   0     0     0 S  0.0  0.0   0:00.00 watchdog/0
   5 root        RT   0   0     0     0 S  0.0  0.0   0:00.39 events/0
   6 root        RT   0   0     0     0 S  0.0  0.0   0:00.00 khelper
   7 root        RT   0   0     0     0 S  0.0  0.0   0:00.00 kthread
   8 root        RT   0   0     0     0 S  0.0  0.0   0:03.13 kblockd/0
   9 root        RT   0   0     0     0 S  0.0  0.0   0:00.00 kacpid
  10 root        RT   0   0     0     0 S  0.0  0.0   0:00.00 cqueue/0
  11 root        RT   0   0     0     0 S  0.0  0.0   0:00.00 khubd
    
```

```

root@mk-vm:/usr/src/sipp/sipp-3.3
File Edit View Search Terminal Help

----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
2750.0(0 ms)/1.000s 5060 77.12 s 162925 62.217.124.222:5060(UDP)

2753 new calls during 1.002 s period 0 ms scheduler resolution
55 calls (limit 8250) Peak was 150 calls, after 48 s
0 Running, 85376 Paused, 2825 Woken up
0 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets

Messages Retrans Timeout Unexpected-Msg
REGISTER -----> 162925 0 0
401 <----- 162903 0 0
REGISTER -----> 162903 0 0
200 <----- E-RTD1 162870 0 0
----- [+]*[/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
    
```



```

root@mk-vm:~
File Edit View Search Terminal Help
top - 21:45:34 up 25 days, 7:44, 2 users, load average: 0.00, 0.00, 0
Tasks: 93 total, 1 running, 92 sleeping, 0 stopped, 0 zombie
Cpu(s): 14.1%us, 5.4%sy, 0.0%ni, 80.5%id, 0.0%wa, 0.0%hi, 0.0%si,
Mem: 1922464k total, 1261328k used, 661136k free, 167764k buffers
Swap: 4128760k total, 0k used, 4128760k free, 869440k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  COMMAND
 28661 root        20   0 92456  39m 2664  S 45.9   2.1   0:54.36 sipp
    1 root        20   0 19228 1496 1220  S  0.0   0.1   0:02.78 init
    2 root        20   0   0     0   0    S  0.0   0.0   0:00.00 kthreadd
    3 root        RT   0   0     0   0    S  0.0   0.0   0:00.19 migration/0
    4 root        20   0   0     0   0    S  0.0   0.0   0:00.46 ksoftirqd/0
    5 root        RT   0   0     0   0    S  0.0   0.0   0:00.00 migration/0
    6 root        RT   0   0     0   0    S  0.0   0.0   0:04.19 watchdog/0
    7 root        RT   0   0     0   0    S  0.0   0.0   0:01.73 migration/0
    8 root        RT   0   0     0   0    S  0.0   0.0   0:00.00 migration/0
    9 root        20   0   0     0   0    S  0.0   0.0   0:00.00 ksoftirqd/0
   10 root        RT   0   0     0   0    S  0.0   0.0   0:00.00 watchdog/0
   11 root        20   0   0     0   0    S  0.0   0.0   0:00.39 events/0
   12 root        20   0   0     0   0    S  0.0   0.0   0:00.00 khelper
   13 root        20   0   0     0   0    S  0.0   0.0   0:00.00 kthread
   14 root        20   0   0     0   0    S  0.0   0.0   0:03.13 kblockd/0
   15 root        20   0   0     0   0    S  0.0   0.0   0:00.00 kacpid
   16 root        20   0   0     0   0    S  0.0   0.0   0:00.00 cqueue/0
    
```

```

root@kalogirou:/etc/opensips
File Edit View Search Terminal Help
top - 21:45:51 up 23 days, 49 min, 2 users, load average: 2.35, 0.68,
Tasks: 87 total, 2 running, 85 sleeping, 0 stopped, 0 zombie
Cpu(s): 57.7%us, 24.3%sy, 0.0%ni, 1.0%id, 0.0%wa, 2.0%hi, 15.0%si,
Mem: 2058808k total, 1013884k used, 1044924k free, 212732k buffers
Swap: 2064376k total, 0k used, 2064376k free, 558828k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU  %MEM    TIME+  COMMAND
 2147 mysql      18   0 269m  48m 4472  S 41.6   2.4 287:50.97 mysqld
 7620 root        15   0 142m 4948 4044  S 14.6   0.2  0:27.31 opensips
 7622 root        15   0 142m 4744 3844  S 14.0   0.2  0:27.92 opensips
 7626 root        15   0 142m 4744 3844  S 13.3   0.2  0:28.48 opensips
 7624 root        16   0 142m 4744 3844  R 13.0   0.2  0:28.88 opensips
 465 root        15   0 592m  32m 9704  S  0.7   1.6  2:44.48 asterisk
18256 root        15   0 19336 1364 1080  S  0.3   0.1 20:55.41 mpg123
 1 root        15   0 10272  700  588  S  0.0   0.0  0:02.95 init
 0.0 0.0 0:00.00 migration/0
 0.0 0.0 0:00.00 ksoftirqd/0
 0.0 0.0 0:00.00 watchdog/0
 0.0 0.0 0:00.39 events/0
 0.0 0.0 0:00.00 khelper
 0.0 0.0 0:00.00 kthread
 0.0 0.0 0:03.13 kblockd/0
 0.0 0.0 0:00.00 kacpid
 0.0 0.0 0:00.00 cqueue/0
    
```

```

root@mk-vm:/usr/src/sipp/sipp-3.3
File Edit View Search Terminal Help

----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length)  Port  Total-time  Total-calls  Remote-host
3000.0(0 ms)/1.000s 5060    98.15 s      224145      62.217.124.222:5060(UDP)

3009 new calls during 1.001 s period  0 ms scheduler resolution
55 calls (limit 9000)                  Peak was 267 calls, after 89 s
0 Running, 94002 Paused, 3131 Woken up
8 dead call msg (discarded)           0 out-of-call msg (discarded)
3 open sockets

Messages  Retrans  Timeout  Unexpected-Msg
REGISTER -----> 224139  3        0          0
401 <----- 224115  0        0          0
REGISTER -----> 224115  8        0          0
200 <----- E-RTD1 224082  0        0          8

----- [+|-|*|/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----

Last Error: Dead call 193624-28661@147.52.82.82 (aborted at index 3), re...
    
```

```

root@mk-vm:~
File Edit View Search Terminal Help
top - 21:46:44 up 25 days, 7:45, 2 users, load average: 0.00, 0.00, 0
Tasks: 93 total, 1 running, 92 sleeping, 0 stopped, 0 zombie
Cpu(s): 22.3%us, 9.3%sy, 0.0%ni, 68.0%id, 0.0%wa, 0.2%hi, 0.2%si,
Mem: 1922464k total, 1265672k used, 656792k free, 167764k buffers
Swap: 4128760k total, 0k used, 4128760k free, 869440k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU %MEM    TIME+  COMMAND
 28661 root        20   0 96812  43m 2664  S 69.5  2.3   1:43.42  sipp
    1 root        20   0 19228 1496 1220  S  0.0  0.1    0:02.78  init
    2 root        20   0   0     0   0    S  0.0  0.0    0:00.00  kthreadd
    3 root        RT   0   0     0   0    S  0.0  0.0    0:00.19  migration/0
    4 root        20   0   0     0   0    S  0.0  0.0    0:00.47  ksoftirqd/0
    5 root        RT   0   0     0   0    S  0.0  0.0    0:00.00  migration/0
    6 root        RT   0   0     0   0    S  0.0  0.0    0:04.19  watchdog/0
    7 root        RT   0   0     0   0    S  0.0  0.0    0:01.72  migration/0
    8 root        RT   0   0     0   0    S  0.0  0.0    0:00.00  migration/0
    9 root        20   0   0     0   0    S  0.0  0.0    0:00.00  ksoftirqd/0
   10 root        RT   0   0     0   0    S  0.0  0.0    0:00.00  watchdog/0
   11 root        20   0   0     0   0    S  0.0  0.0    0:00.39  events/0
   12 root        20   0   0     0   0    S  0.0  0.0    0:00.00  khelper
   13 root        20   0   0     0   0    S  0.0  0.0    0:00.00  kthread
   14 root        20   0   0     0   0    S  0.0  0.0    0:03.13  kblockd/0
   15 root        20   0   0     0   0    S  0.0  0.0    0:00.00  kacpid
   16 root        20   0   0     0   0    S  0.0  0.0    0:00.00  cqueue/0
    
```

```

root@kalogirou:/etc/opensips
File Edit View Search Terminal Help
top - 21:47:00 up 23 days, 50 min, 2 users, load average: 3.21, 1.27,
Tasks: 87 total, 1 running, 86 sleeping, 0 stopped, 0 zombie
Cpu(s): 54.7%us, 22.3%sy, 0.0%ni, 7.3%id, 0.0%wa, 1.0%hi, 14.7%si,
Mem: 2058808k total, 1013636k used, 1045172k free, 212732k buffers
Swap: 2064376k total, 0k used, 2064376k free, 558828k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU %MEM    TIME+  COMMAND
 2147 mysql      18   0 269m  48m 4472  S 40.6  2.4 288:16.51  mysqld
 7622 root       15   0 142m 4744 3844  S 13.0  0.2   0:36.60  opensips
 7626 root       15   0 142m 4744 3844  S 13.0  0.2   0:37.32  opensips
 7624 root       15   0 142m 4744 3844  S 12.6  0.2   0:37.66  opensips
 7620 root       15   0 142m 4948 4044  S 12.3  0.2   0:35.90  opensips
 1862 root       25   0 21068 1332  560  S  0.3  0.1   0:03.15  pcsd
18256 root       15   0 19336 1364 1080  S  0.3  0.1 20:55.61  mpg123
 1 root       15   0 10272  700  588  S  0.0  0.0   0:02.95  init
 0 root       0   0   0     0   0    S  0.0  0.0   0:00.00  migration/0
 0 root       0   0   0     0   0    S  0.0  0.0   0:00.00  ksoftirqd/0
 0 root       0   0   0     0   0    S  0.0  0.0   0:00.00  watchdog/0
 0 root       0   0   0     0   0    S  0.0  0.0   0:00.39  events/0
 0 root       0   0   0     0   0    S  0.0  0.0   0:00.00  khelper
 0 root       0   0   0     0   0    S  0.0  0.0   0:00.00  kthread
 0 root       0   0   0     0   0    S  0.0  0.0   0:03.13  kblockd/0
 0 root       0   0   0     0   0    S  0.0  0.0   0:00.00  kacpid
 0 root       0   0   0     0   0    S  0.0  0.0   0:00.00  cqueue/0
    
```

```

root@mk-vm:/usr/src/sipp/sipp-3.3
File Edit View Search Terminal Help

----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length)  Port  Total-time  Total-calls  Remote-host
3200.0(0 ms)/1.000s 5060   166.26 s     436187      62.217.124.222:5060(UDP)

3206 new calls during 1.002 s period  0 ms scheduler resolution
57 calls (limit 9600)                 Peak was 267 calls, after 89 s
0 Running, 104751 Paused, 3630 Woken up
14 dead call msg (discarded)          0 out-of-call msg (discarded)
3 open sockets

Messages  Retrans  Timeout  Unexpected-Msg
REGISTER -----> 436187  7        0          0
 401 <----- 436159  0        0          0
REGISTER -----> 436159 14        0          0
 200 <----- E-RTD1 436116  0        0          14

----- [+|-|*|/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----

Last Error: Dead call 428399-28661@147.52.82.82 (aborted at index 3), re...
    
```

```

root@mk-vm:~
File Edit View Search Terminal Help
top - 21:47:08 up 25 days, 7:45, 2 users, load average: 0.00, 0.00, 0
Tasks: 93 total, 1 running, 92 sleeping, 0 stopped, 0 zombie
Cpu(s): 26.0%us, 11.4%sy, 0.0%ni, 62.6%id, 0.0%wa, 0.0%hi, 0.0%si,
Mem: 1922464k total, 1267968k used, 654496k free, 167764k buffers
Swap: 4128760k total, 0k used, 4128760k free, 869440k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU %MEM    TIME+  COMMAND
 28661 root        20   0 98968  45m 2664  S 77.9  2.4   2:01.04 sipp
 28659 root        20   0 15020 1288 1008  R  0.3  0.1   0:00.23 top
   1 root        20   0 19228 1496 1220  S  0.0  0.1   0:02.78 init
   2 root        20   0 0        0 0      S  0.0  0.0   0:00.00 kthreadd
   3 root        RT   0 0        0 0      S  0.0  0.0   0:00.19 migration/0
   4 root        20   0 0        0 0      S  0.0  0.0   0:00.47 ksoftirqd/0
   5 root        RT   0 0        0 0      S  0.0  0.0   0:00.00 migration/0
   6 root        RT   0 0        0 0      S  0.0  0.0   0:00.10 watchdog/0
   7 root        RT   0 0        0 0      S  0.0  0.0   0:00.00 kthreadd
   8 root        RT   0 0        0 0      S  0.0  0.0   0:00.00 kthreadd
   9 root        20   0 0        0 0      S  0.0  0.0   0:00.00 kthreadd
  10 root        RT   0 0        0 0      S  0.0  0.0   0:00.00 kthreadd
  11 root        20   0 0        0 0      S  0.0  0.0   0:00.00 kthreadd
  12 root        20   0 0        0 0      S  0.0  0.0   0:00.00 kthreadd
  13 root        20   0 0        0 0      S  0.0  0.0   0:00.00 kthreadd
  14 root        20   0 0        0 0      S  0.0  0.0   0:00.00 kthreadd
  15 root        20   0 0        0 0      S  0.0  0.0   0:00.00 kthreadd
    
```

```

root@kalogirou:/etc/opensips
File Edit View Search Terminal Help
top - 21:47:24 up 23 days, 51 min, 2 users, load average: 3.23, 1.44,
Tasks: 87 total, 4 running, 83 sleeping, 0 stopped, 0 zombie
Cpu(s): 54.2%us, 26.6%sy, 0.0%ni, 0.0%id, 3.7%wa, 1.3%hi, 14.3%si,
Mem: 2058808k total, 1015372k used, 1043436k free, 212732k buffers
Swap: 2064376k total, 0k used, 2064376k free, 558836k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU %MEM    TIME+  COMMAND
 2147 mysql      18   0 269m  48m 4472  S 38.3  2.4 288:25.36 mysqld
 7620 root        16   0 142m 4984 4072  R 14.3  0.2   0:39.22 opensips
 7624 root        15   0 142m 4844 3936  S 14.3  0.2   0:40.80 opensips
 7622 root        15   0 142m 4844 3936  S 13.6  0.2   0:39.83 opensips
 7626 root        15   0 142m 4844 3936  S 13.6  0.2   0:40.49 opensips
   376 root        10  -5 0        0 0      S  1.7  0.0   7:49.58 kjournald
 1675 root        16   0 5932  628 508  S  0.3  0.0   0:47.16 syslogd
 7675 root        15   0 12752 1080 822  R  0.3  0.1   0:00.32 top
   0.3  0.1 20:55.67 mpg123
   0.0  0.0 0:02.95 init
   0.0  0.0 0:00.00 migration/0
   0.0  0.0 0:00.00 ksoftirqd/0
   0.0  0.0 0:00.00 watchdog/0
   0.0  0.0 0:00.39 events/0
   0.0  0.0 0:00.00 khelper
   0.0  0.0 0:00.00 kthread
   0.0  0.0 0:03.13 kblockd/0
    
```

```

root@mk-vm:/usr/src/sipp/sipp-3.3
File Edit View Search Terminal Help

----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
3270.0(0 ms)/1.000s 5060 191.30 s 517211 62.217.124.222:5060(UDP)

3276 new calls during 1.002 s period 0 ms scheduler resolution
822 calls (limit 9810) Peak was 978 calls, after 190 s
0 Running, 106549 Paused, 4400 Woken up
20 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets

Messages Retrans Timeout Unexpected-Msg
REGISTER -----> 517211 638 0
401 <----- 516814 0 0 0
REGISTER -----> 516814 989 0
200 <----- E-RTD1 516369 0 0 20

----- [+|-|*|/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----

Last Error: Dead call 513551-28661@147.52.82.82 (aborted at index 3), re...
    
```


A.3.2 Οθόνες δοκιμών “Πραγματοποίηση κλήσεων (invite)”

Applications Places en [Battery] [Wi-Fi] [Speaker] Δευ 29 Απρ 9:59:15 MM Michalis Kalogirou

root@mk: /usr/src/sipp/sipp-3.3

```

File Edit View Search Terminal Help
Call-rate(length)  Port  Total-time  Total-calls  Remote-host
500.0(0 ms)/1.000s  5060      25.21 s      12608      62.217.124.222:5060(UDP)

501 new calls during 1.002 s period    0 ms scheduler resolution
18 calls (limit 1500)                  Peak was 139 calls, after 0 s
1 Running, 12607 Paused, 505 Woken up
5 dead call msg (discarded)            0 out-of-call msg (discarded)
3 open sockets

Messages  Retrans  Timeout  Unexpected-Msg
INVITE ----->      12607    0        0
100 <-----      12603    0        0
180 <-----      12574    0        0
183 <-----         0        0        0
200 <-----      E-RTD1 12599    0        0
ACK ----->      12599    0
Pause [ 0ms]      12599    5
BYE ----->      12593    0        0
200 <-----      12585    0        0

----- [+|-|*|/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
Last Error: Dead call 12069-6886@147.52.3.5 (aborted at index 6), receiv...

```

root@mk-vm: /usr/src/sipp/sipp-3.3

```

File Edit View Search Terminal Help
----- Scenario Screen ----- [1-9]: Change Screen --
Port  Total-time  Total-calls  Transport
5060      30.05 s      12660      UDP

501 new calls during 1.002 s period    0 ms scheduler resolution
2002 calls                              Peak was 2104 calls, after 8 s
0 Running, 12660 Paused, 497 Woken up
0 dead call msg (discarded)
3 open sockets

Messages  Retrans  Timeout  Unexpected-Msg
-----> INVITE      12660    0        0
<----- 180      12660    0
<----- 200      12660    0
-----> ACK          E-RTD1 12610    0
-----> BYE          12651    0
<----- 200      12651    0
[ 4000ms] Pause      12651
----- Sipp Server Mode -----
Last Error: Aborting call on unexpected message for Call-Id '12507-6886@...

```

```

top - 21:59:14 up 23:22,  1 user,  load average: 0.95, 1.21, 0.87
Tasks:  85 total,  1 running,  84 sleeping,  0 stopped,  0 zombie
Cpu(s): 17.3%us,  7.0%sy,  0.0%ni, 59.7%id,  0.0%wa,  2.7%hi, 13.3%si,  0.0%st
Mem:   2058808k total, 1140984k used,  917824k free,  201956k buffers
Swap:  2064376k total,   0k used,  2064376k free,  666608k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 8336 root        15   0  366m 242m 242m  S   7.0  12.1   0:29.05 opensips
 8337 root        15   0  366m 239m 238m  S   7.0  11.9   0:34.37 opensips
 8334 root        15   0  366m 243m 242m  S   6.7  12.1   0:35.76 opensips
 8335 root        15   0  366m 243m 242m  S   6.7  12.1   0:35.05 opensips
 2149 mysql       18   0  254m  26m 4340  S   1.7   1.3   3:25.30 mysqld
 7527 root        15   0 19336 1360 1080  S   0.7   0.1   0:39.37 mpg123
    1 root        15   0 10372  700  588  S   0.0   0.0   0:00.54 init
    2 root         RT  -5    0    0    0  S   0.0   0.0   0:00.00 migration/0
    3 root         34  19    0    0    0  S   0.0   0.0   0:00.00 ksoftirqd/0
    4 root         RT  -5    0    0    0  S   0.0   0.0   0:00.00 watchdog/0

```

[SIPp - Googl...] [Update Ma...] [Metantuxia...] [Diatrivi] [tmm] root@mk:/u... root@mk-v... root@kalogi... [michalis@p...]

Applications Places

en [network icons] Δευ 29 Απρ 9:59:56 MM Michalis Kalogirou

root@mk: /usr/src/sipp/sipp-3.3

File Edit View Search Terminal Help

Call-rate(length)	Port	Total-time	Total-calls	Remote-host
900.0(0 ms)/1.000s	5060	66.29 s	42710	62.217.124.222:5060(UDP)

902 new calls during 1.002 s period 0 ms scheduler resolution
37 calls (limit 2700) Peak was 139 calls, after 0 s
1 Running, 26071 Paused, 1416 Woken up
18 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets

	Messages	Retrans	Timeout	Unexpected-Msg
INVITE ----->	42709	0	0	
100 <-----	42700	0	0	0
180 <-----	42599	0	0	0
183 <-----	0	0	0	0
200 <-----	E-RTD1 42692	0	0	0
ACK ----->	42692	0		
Pause [0ms]	42692			18
BYE ----->	42671	0	0	
200 <-----	42655	0	0	0

----- [+ | - | * | /] : Adjust rate - - - - [q] : Soft exit - - - - [p] : Pause traffic - - - -

Last Error: Dead call 36543-6886@147.52.3.5 (aborted at index 6), receiv...

root@mk-vm: /usr/src/sipp/sipp-3.3

File Edit View Search Terminal Help

----- Scenario Screen ----- [1-9]: Change Screen --

Port	Total-time	Total-calls	Transport
5060	71.12 s	42795	UDP

900 new calls during 1.002 s period 0 ms scheduler resolution
3485 calls Peak was 3532 calls, after 69 s
0 Running, 28102 Paused, 1385 Woken up
0 dead call msg (discarded)
3 open sockets

	Messages	Retrans	Timeout	Unexpected-Msg
-----> INVITE	42795	0	0	0
<----- 180	42795	0		
<----- 200	42795	0	0	
-----> ACK	E-RTD1 41770	0	0	0
-----> BYE	42779	0	0	0
<----- 200	42779	0		
[4000ms] Pause	42779			1009

----- Sipp Server Mode -----

Last Error: Aborting call on unexpected message for Call-Id '42630-6886@...

```
top - 21:59:56 up 23:23,  1 user,  load average: 2.38, 1.58, 1.01
Tasks: 85 total,  5 running, 80 sleeping,  0 stopped,  0 zombie
Cpu(s): 42.0%us, 18.0%sy,  0.0%ni,  7.0%id,  0.0%wa,  3.7%hi, 29.3%si,  0.0%st
Mem:  2058808k total, 1140984k used,  917824k free,  201956k buffers
Swap: 2064376k total,    0k used, 2064376k free,  666608k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2149	mysql	18	0	254m	26m	4340	S	23.3	1.3	3:35.67	mysqld
8337	root	15	0	366m	240m	239m	R	17.3	12.0	0:39.52	opensips
8336	root	15	0	366m	243m	242m	R	17.0	12.1	0:34.30	opensips
8334	root	15	0	366m	244m	243m	R	16.0	12.1	0:41.11	opensips
8335	root	15	0	366m	244m	243m	S	15.3	12.1	0:40.40	opensips
8339	root	15	0	366m	202m	201m	S	0.7	10.1	0:01.73	opensips
8373	root	15	0	12752	1080	832	R	0.7	0.1	0:01.83	top
2348	root	15	0	619m	61m	9404	S	0.3	3.0	1:01.39	asterisk
1	root	15	0	10372	700	588	S	0.0	0.0	0:00.54	init
2	root	BT	-5	0	0	0	S	0.0	0.0	0:00.00	migration/0

[SIPp - Googl...] [Update Ma...] [Metantuxia...] [Diatrivi] [tmm] root@mk:/u... root@mk-v... root@kalogi... [michalis@p...]

Applications Places

en 10:00:23 MM Michalis Kalogirou

root@mk: /usr/src/sipp/sipp-3.3

File Edit View Search Terminal Help

```

Call-rate(length)  Port  Total-time  Total-calls  Remote-host
930.0(0 ms)/1.000s  5060    93.34 s     67527       62.217.124.222:5060(UDP)

932 new calls during 1.002 s period    0 ms scheduler resolution
168 calls (limit 2790)                  Peak was 447 calls, after 92 s
1 Running, 30228 Paused, 2105 Woken up
34 dead call msg (discarded)           0 out-of-call msg (discarded)
3 open sockets

Messages  Retrans  Timeout  Unexpected-Msg
INVITE -----> 67526 111 0
100 <----- 67498 1 0 0
180 <----- 67307 0 0 0
183 <----- 0 0 0 0
200 <----- E-RTD1 67445 0 0 0
ACK -----> 67445 0
Pause [ 0ms] 67445 22
BYE -----> 67419 154 0
200 <----- 67337 0 0 0

----- [ + | * | / ] : Adjust rate ---- [ q ] : Soft exit ---- [ p ] : Pause traffic -----
Last Error: Dead call 65668-6886@147.52.3.5 (successful), received 'SIP/...
```

root@mk-vm: /usr/src/sipp/sipp-3.3

File Edit View Search Terminal Help

```

----- Scenario Screen ----- [1-9]: Change Screen --
Port  Total-time  Total-calls  Transport
5060    98.16 s     67591  UDP

1024 new calls during 1.002 s period    0 ms scheduler resolution
3416 calls                               Peak was 3636 calls, after 90 s
0 Running, 33589 Paused, 1768 Woken up
55 dead call msg (discarded)
3 open sockets

Messages  Retrans  Timeout  Unexpected-Msg
-----> INVITE 67591 0 0 0
<----- 180 67591 0
<----- 200 67591 117 0
-----> ACK E-RTD1 65251 0 0 37
-----> BYE 67493 45 0 0
<----- 200 67493 45
[ 4000ms] Pause 67493 2214

----- Sipp Server Mode -----
Last Error: Aborting call on unexpected message for Call-Id '67542-6886@...
```

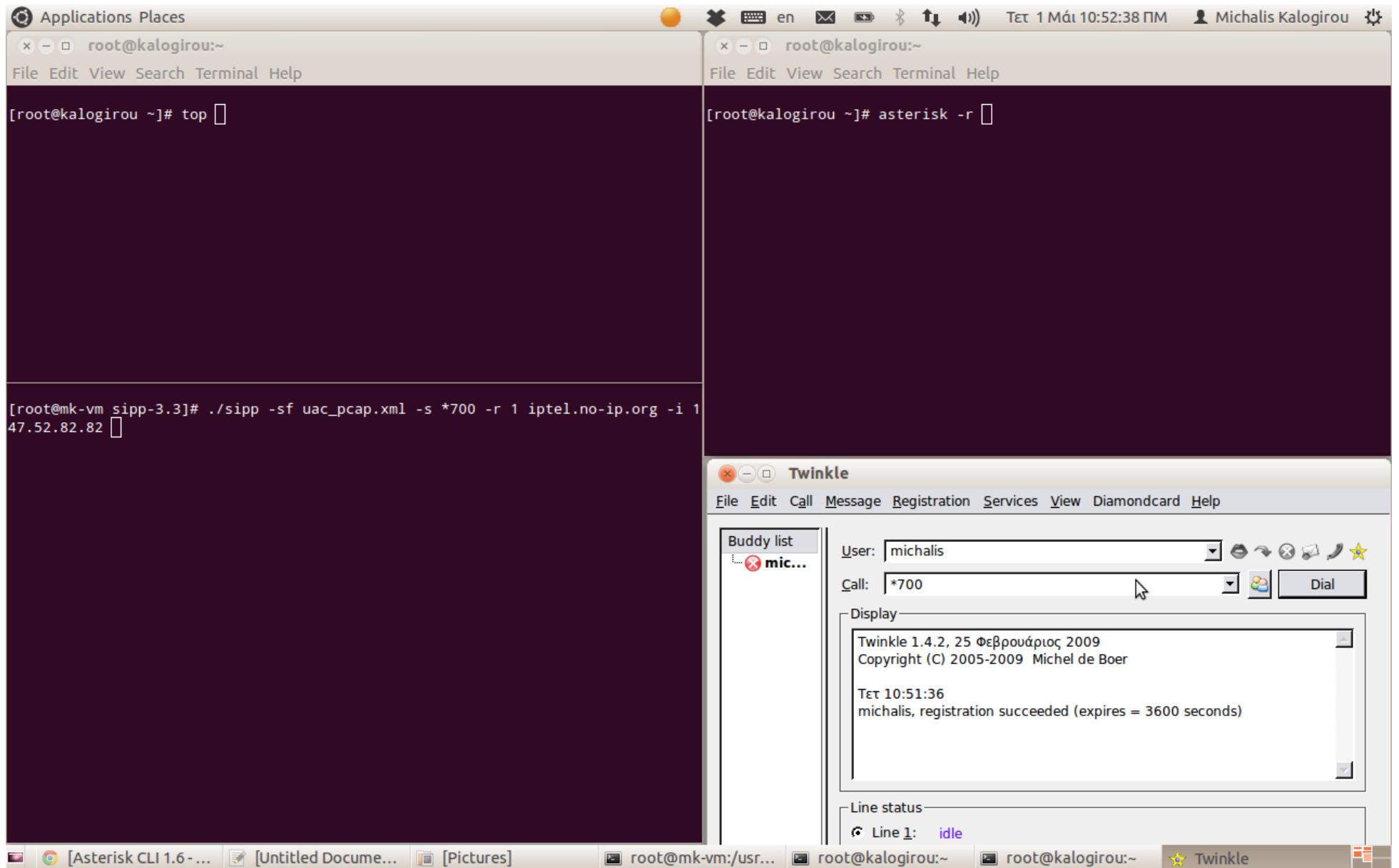
```

top - 22:00:23 up 23:24, 1 user, load average: 3.96, 2.01, 1.17
Tasks: 85 total, 8 running, 77 sleeping, 0 stopped, 0 zombie
Cpu(s): 52.5%us, 20.1%sy, 0.0%ni, 0.0%id, 0.0%wa, 3.0%hi, 24.4%si, 0.0%st
Mem: 2058808k total, 1141976k used, 916832k free, 201956k buffers
Swap: 2064376k total, 0k used, 2064376k free, 666612k cached

  PID USER   PR  NI  VIRT  RES  SHR  S  %CPU  %MEM  TIME+  COMMAND
 2149 mysql  18   0 254m 26m  4340 S 35.3  1.3  3:44.82 mysqld
 8335 root   16   0 366m 244m 243m R 16.0 12.1  0:44.45 opensips
 8334 root   15   0 366m 244m 243m R 15.6 12.1  0:45.50 opensips
 8336 root   16   0 366m 243m 243m R 15.6 12.1  0:38.14 opensips
 8337 root   16   0 366m 240m 239m R 15.6 12.0  0:43.78 opensips
 8339 root   15   0 366m 202m 201m S  0.7 10.1  0:01.86 opensips
 7527 root   15   0 19336 1360 1080 S  0.3  0.1  0:39.61 mpg123
 8373 root   15   0 12752 1080 832 R  0.3  0.1  0:01.87 top
    1 root   15   0 10372 700 588 S  0.0  0.0  0:00.54 init
    2 root   BT   5   0   0   0   0 S  0.0  0.0  0:00.00 migration/0
```

[SIPp - Googl...] [Update Ma...] [Metantuxia...] [Diatrivi] [tmm] root@mk:/u... root@mk-v... root@kalogi... [michalis@p...]

A.3.3 Οθόνες δοκιμών “Προηγμένες Υπηρεσίες (media services)”



Applications Places

root@kalogirou:~

File Edit View Search Terminal Help

```
top - 10:52:52 up 2 days, 12:16, 3 users, load average: 2.76, 6.15, 4.55
Tasks: 90 total, 2 running, 88 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.3%us, 0.3%sy, 0.0%ni, 98.0%id, 0.0%wa, 0.0%hi, 0.3%si, 0.0%st
Mem: 2058808k total, 1996720k used, 62088k free, 48004k buffers
Swap: 2064376k total, 88k used, 2064288k free, 1673648k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19920	root	15	0	593m	28m	9588	S	1.0	1.4	0:02.25	asterisk
19936	root	15	0	19336	1360	1080	S	0.3	0.1	0:00.32	mpg123
1	root	15	0	10372	700	588	S	0.0	0.0	0:00.93	init
2	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	migration/0
3	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
4	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
5	root	10	-5	0	0	0	S	0.0	0.0	0:00.07	events/0
6	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	khelper
23	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	kthread

```
[root@mk-vm sipp-3.3]# ./sipp -sf uac_pcap.xml -s *700 -r 1 iptel.no-ip.org -i 147.52.82.82
```

root@kalogirou:~

File Edit View Search Terminal Help

```
Asterisk 11.2.1, Copyright (C) 1999 - 2012 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 11.2.1 currently running on kalogirou (pid = 19920)
== Using SIP RTP TOS bits 184
== Using SIP RTP CoS mark 5
-- Executing [700@DialIn:1] GoTo("SIP/opensips-00000000", "MusicOnHold,700,1")
-- Goto (MusicOnHold,700,1)
-- Executing [700@MusicOnHold:1] Answer("SIP/opensips-00000000", "") in new
stack
-- Executing [700@MusicOnHold:2] MusicOnHold("SIP/opensips-00000000", "radio")
in new stack
-- Started music on hold, class 'radio', on channel 'SIP/opensips-00000000'
kalogirou*CLI>
```

Twinkle

File Edit Call Message Registration Services View Diamondcard Help

Buddy list

- mic...

User: michalis

Call: Dial

Display

```
Twinkle 1.4.2, 25 Φεβρουάριος 2009
Copyright (C) 2005-2009 Michel de Boer

Τετ 10:51:36
michalis, registration succeeded (expires = 3600 seconds)

Τετ 10:52:49
Line 1: far end answered call.
```

Line status

Line 1: established g711a 0:00:05

[Asterisk CLI 1.6- ...] [Untitled Docume... [Pictures] root@mk-vm:/usr... root@kalogirou:~ root@kalogirou:~ Twinkle

Applications Places

root@kalogirou:~

File Edit View Search Terminal Help

```
top - 10:53:25 up 2 days, 12:17, 3 users, load average: 1.54, 5.46, 4.38
Tasks: 90 total, 2 running, 88 sleeping, 0 stopped, 0 zombie
Cpu(s): 8.1%us, 1.7%sy, 0.0%ni, 89.3%id, 0.3%wa, 0.0%hi, 0.7%si, 0.0%st
Mem: 2058808k total, 1999744k used, 59064k free, 48044k buffers
Swap: 2064376k total, 88k used, 2064288k free, 1673660k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19920	root	15	0	599m	31m	9632	S	9.0	1.5	0:03.08	asterisk
19936	root	15	0	19336	1360	1080	S	0.7	0.1	0:00.42	mpg123
2149	mysql	18	0	254m	26m	4412	S	0.3	1.3	19:55.34	mysqld
1	root	15	0	10372	700	588	S	0.0	0.0	0:00.93	init
2	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	migration/0
3	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0

root@mk-vm:/usr/src/sipp/sipp-3.3

File Edit View Search Terminal Help

```
----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
1.0(0 ms)/1.000s 5060 12.50 s 12 62.217.124.222:5060(UDP)

1 new calls during 1.002 s period 0 ms scheduler resolution
5 calls (limit 15) Peak was 6 calls, after 6 s
1 Running, 13 Paused, 4 Woken up
0 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets

Messages Retrans Timeout Unexpected-Msg
INVITE -----> 12 0 0 0
100 <----- 12 0 0 0
180 <----- 0 0 0 0
200 <----- E-RTD1 12 0 0 0

ACK -----> 12 0
Pause [ 5000ms] 12 0
BYE -----> 7 0 0 0
200 <----- 7 0 0 0

----- [+-|*[/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
```

root@kalogirou:~

File Edit View Search Terminal Help

```
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 11.2.1 currently running on kalogirou (pid = 19920)
== Using SIP RTP TOS bits 184
== Using SIP RTP CoS mark 5
-- Executing [700@DialIn:1] GoTo("SIP/opensips-00000000", "MusicOnHold,700,1")
-- Goto (MusicOnHold,700,1)
-- Executing [700@MusicOnHold:1] Answer("SIP/opensips-00000000", "") in new stack
-- Executing [700@MusicOnHold:2] MusicOnHold("SIP/opensips-00000000", "radio") in new stack
-- Started music on hold, class 'radio', on channel 'SIP/opensips-00000000'
kalogirou*CLI> logger mute
Console is muted.
kalogirou*CLI> core show channels count
6 active channels
6 active calls
10 calls processed
kalogirou*CLI>
```

Twinkle

File Edit Call Message Registration Services View Diamondcard Help

Buddy list

- mic...

User: michalis

Call: Dial

Display

```
Twinkle 1.4.2, 25 Φεβρουάριος 2009
Copyright (C) 2005-2009 Michel de Boer

Τετ 10:51:36
michalis, registration succeeded (expires = 3600 seconds)

Τετ 10:52:49
Line 1: far end answered call.
```

Line status

Line 1: established g711a 0:00:38

[Asterisk CLI 1.6-... [Untitled Docume... [Pictures] root@mk-vm:/usr... root@kalogirou:~ root@kalogirou:~ Twinkle

Applications Places

root@kalogirou:~

```
File Edit View Search Terminal Help
top - 10:53:43 up 2 days, 12:17, 3 users, load average: 1.20, 5.20, 4.31
Tasks: 90 total, 2 running, 88 sleeping, 0 stopped, 0 zombie
Cpu(s): 17.4%us, 2.7%sy, 0.0%ni, 78.3%id, 0.0%wa, 0.3%hi, 1.3%si, 0.0%st
Mem: 2058808k total, 2005116k used, 53692k free, 48060k buffers
Swap: 2064376k total, 88k used, 2064288k free, 1673700k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19920	root	15	0	602m	36m	9632	S	20.0	1.8	0:05.69	asterisk
8476	root	15	0	366m	23m	22m	S	0.7	1.2	0:01.53	opensips
2149	mysql	18	0	254m	26m	4412	S	0.3	1.3	19:55.38	mysqld
19936	root	15	0	19336	1360	1080	S	0.3	0.1	0:00.48	mpg123
1	root	15	0	10372	700	588	S	0.0	0.0	0:00.93	init
2	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	migration/0

root@mk-vm:/usr/src/sipp/sipp-3.3

```
File Edit View Search Terminal Help
----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
2.0(0 ms)/1.000s 5060 30.53 s 43 62.217.124.222:5060(UDP)

2 new calls during 1.002 s period 0 ms scheduler resolution
10 calls (limit 30) Peak was 11 calls, after 22 s
1 Running, 44 Paused, 5 Woken up
0 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets

Messages Retrans Timeout Unexpected-Msg
INVITE -----> 43 0 0
100 <----- 43 0 0
180 <----- 0 0 0
200 <----- E-RTD1 43 0 0

ACK -----> 43 0
Pause [ 5000ms] 43 0
BYE -----> 33 0 0
200 <----- 33 0 0

----- [+-|*[/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
```

root@kalogirou:~

```
File Edit View Search Terminal Help
== Using SIP RTP CoS mark 5
-- Executing [700@DialIn:1] GoTo("SIP/opensips-00000000", "MusicOnHold,700,1")
-- Goto (MusicOnHold,700,1)
-- Executing [700@MusicOnHold:1] Answer("SIP/opensips-00000000", "") in new stack
-- Executing [700@MusicOnHold:2] MusicOnHold("SIP/opensips-00000000", "radio") in new stack
-- Started music on hold, class 'radio', on channel 'SIP/opensips-00000000'
kalogirou*CLI> logger mute
Console is muted.
kalogirou*CLI> core show channels count
6 active channels
6 active calls
10 calls processed
kalogirou*CLI> core show channels count
11 active channels
11 active calls
33 calls processed
kalogirou*CLI>
```

Twinkle

File Edit Call Message Registration Services View Diamondcard Help

Buddy list
mic...

User: michalis

Call: Dial

Display

Twinkle 1.4.2, 25 Φεβρουάριος 2009
Copyright (C) 2005-2009 Michel de Boer

Τετ 10:51:36
michalis, registration succeeded (expires = 3600 seconds)

Τετ 10:52:49
Line 1: far end answered call.

Line status

Line 1: established g711a 0:00:56

[Asterisk CLI 1.6- ...] [Untitled Docume... [Pictures] root@mk-vm:/usr... root@kalogirou:~ root@kalogirou:~ Twinkle

Applications Places

root@kalogirou:~

```
File Edit View Search Terminal Help
top - 10:53:58 up 2 days, 12:17, 3 users, load average: 0.93, 4.94, 4.24
Tasks: 90 total, 2 running, 88 sleeping, 0 stopped, 0 zombie
Cpu(s): 24.5%us, 4.0%sy, 0.0%ni, 68.5%id, 0.0%wa, 0.7%hi, 2.3%si, 0.0%st
Mem: 2058808k total, 2009124k used, 49684k free, 48084k buffers
Swap: 2064376k total, 88k used, 2064288k free, 1673748k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19920	root	15	0	604m	39m	9632	S	29.6	2.0	0:09.12	asterisk
19936	root	15	0	19336	1360	1080	S	0.7	0.1	0:00.52	mpg123
8475	root	15	0	366m	23m	22m	S	0.3	1.2	0:01.27	opensips
8476	root	15	0	366m	23m	22m	S	0.3	1.2	0:01.54	opensips
19960	root	15	0	12752	1080	832	R	0.3	0.1	0:00.12	top
1	root	15	0	10372	700	588	S	0.0	0.0	0:00.93	init

root@mk-vm:/usr/src/sipp/sipp-3.3

```
File Edit View Search Terminal Help
----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
3.0(0 ms)/1.000s 5060 45.56 s 83 62.217.124.222:5060(UDP)

3 new calls during 1.002 s period 0 ms scheduler resolution
15 calls (limit 45) Peak was 16 calls, after 39 s
1 Running, 77 Paused, 8 Woken up
0 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets

Messages Retrans Timeout Unexpected-Msg
INVITE -----> 83 0 0 0
100 <----- 83 0 0 0
180 <----- 0 0 0 0
200 <----- E-RTD1 83 0 0 0

ACK -----> 83 0
Pause [ 5000ms] 83 0
BYE -----> 68 0 0 0
200 <----- 68 0 0 0

----- [ + ] - [ * ] [ / ] : Adjust rate ---- [ q ] : Soft exit ---- [ p ] : Pause traffic -----
```

root@kalogirou:~

```
File Edit View Search Terminal Help
-- Executing [700@MusicOnHold:1] Answer("SIP/opensips-00000000", "") in new
stack
-- Executing [700@MusicOnHold:2] MusicOnHold("SIP/opensips-00000000", "radi
o") in new stack
-- Started music on hold, class 'radio', on channel 'SIP/opensips-00000000'
kalogirou*CLI> logger mute
Console is muted.
kalogirou*CLI> core show channels count
6 active channels
6 active calls
10 calls processed
kalogirou*CLI> core show channels count
11 active channels
11 active calls
33 calls processed
kalogirou*CLI> core show channels count
16 active channels
16 active calls
78 calls processed
kalogirou*CLI>
```

Twinkle

File Edit Call Message Registration Services View Diamondcard Help

Buddy list
mic...

User: michalis

Call: Dial

Display

Twinkle 1.4.2, 25 Φεβρουάριος 2009
Copyright (C) 2005-2009 Michel de Boer

Τετ 10:51:36
michalis, registration succeeded (expires = 3600 seconds)

Τετ 10:52:49
Line 1: far end answered call.

Line status

Line 1: established g711a 0:01:11

[Asterisk CLI 1.6- ...] [Untitled Docume... [Pictures] root@mk-vm:/usr... root@kalogirou:~ root@kalogirou:~ Twinkle

Applications Places

root@kalogirou:~

File Edit View Search Terminal Help

```
top - 10:54:16 up 2 days, 12:17, 3 users, load average: 0.66, 4.62, 4.15
Tasks: 90 total, 2 running, 88 sleeping, 0 stopped, 0 zombie
Cpu(s): 33.4%us, 5.3%sy, 0.0%ni, 57.6%id, 0.0%wa, 0.7%hi, 3.0%si, 0.0%st
Mem: 2058808k total, 2016232k used, 42576k free, 48116k buffers
Swap: 2064376k total, 88k used, 2064288k free, 1673848k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19920	root	15	0	607m	46m	9632	S	40.6	2.3	0:15.37	asterisk
2149	mysql	18	0	254m	26m	4412	S	0.7	1.3	19:55.55	mysqld
8476	root	15	0	366m	23m	22m	S	0.3	1.2	0:01.57	opensips
19960	root	15	0	12752	1080	832	R	0.3	0.1	0:00.15	top
1	root	15	0	10372	700	588	S	0.0	0.0	0:00.93	init
2	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	migration/0

root@mk-vm:/usr/src/sipp/sipp-3.3

File Edit View Search Terminal Help

```
----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
4.0(0 ms)/1.000s 5060 61.59 s 142 62.217.124.222:5060(UDP)

4 new calls during 1.002 s period 0 ms scheduler resolution
20 calls (limit 60) Peak was 21 calls, after 55 s
1 Running, 114 Paused, 10 Woken up
0 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets
```

	Messages	Retrans	Timeout	Unexpected-Msg
INVITE ----->	142	0	0	
100 <-----	142	0	0	0
180 <-----	0	0	0	0
200 <-----	E-RTD1 142	0	0	0
ACK ----->	142	0		
Pause [5000ms]	142			0
BYE ----->	122	0	0	
200 <-----	122	0	0	0

```
----- [+-|*[/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
```

root@kalogirou:~

```
-- Started music on hold, class 'radio', on channel 'SIP/opensips-00000000'
kalogirou*CLI> logger mute
Console is muted.
kalogirou*CLI> core show channels count
6 active channels
6 active calls
10 calls processed
kalogirou*CLI> core show channels count
11 active channels
11 active calls
33 calls processed
kalogirou*CLI> core show channels count
16 active channels
16 active calls
78 calls processed
kalogirou*CLI> core show channels count
21 active channels
21 active calls
125 calls processed
kalogirou*CLI>
```

Twinkle

File Edit Call Message Registration Services View Diamondcard Help

Buddy list

- mic...

User:

Call: Dial

Display

```
Twinkle 1.4.2, 25 Φεβρουάριος 2009
Copyright (C) 2005-2009 Michel de Boer

Τετ 10:51:36
michalis, registration succeeded (expires = 3600 seconds)

Τετ 10:52:49
Line 1: far end answered call.
```

Line status

Line 1: established g711a 0:01:27

[Asterisk CLI 1.6- ...] [Untitled Docume... [Pictures] root@mk-vm:/usr... root@kalogirou:~ root@kalogirou:~ Twinkle

Applications Places

en 10:54:30 ΠΜ Michalis Kalogirou

```

root@kalogirou:~
File Edit View Search Terminal Help
top - 10:54:28 up 2 days, 12:18, 3 users, load average: 0.56, 4.47, 4.10
Tasks: 90 total, 3 running, 87 sleeping, 0 stopped, 0 zombie
Cpu(s): 45.8%us, 6.6%sy, 0.0%ni, 42.5%id, 0.3%wa, 0.3%hi, 4.3%si, 0.0%st
Mem: 2058808k total, 2020240k used, 38568k free, 48132k buffers
Swap: 2064376k total, 88k used, 2064288k free, 1673936k cached

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19920	root	15	0	609m	50m	9632	S	54.9	2.5	0:20.88	asterisk
2149	mysql	18	0	254m	26m	4412	S	1.0	1.3	19:55.63	mysqld
8474	root	15	0	366m	23m	22m	S	0.7	1.2	0:01.75	opensips
8476	root	15	0	366m	23m	22m	S	0.3	1.2	0:01.59	opensips
19936	root	15	0	19336	1360	1080	R	0.3	0.1	0:00.59	mpg123
1	root	15	0	10372	700	588	S	0.0	0.0	0:00.93	init

```

root@mk-vm:/usr/src/sipp/sipp-3.3
File Edit View Search Terminal Help
----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
5.0(0 ms)/1.000s 5060 75.62 s 208 62.217.124.222:5060(UDP)

5 new calls during 1.002 s period 0 ms scheduler resolution
25 calls (limit 75) Peak was 26 calls, after 70 s
1 Running, 150 Paused, 11 Woken up
0 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets

Messages Retrans Timeout Unexpected-Msg
INVITE -----> 208 0 0 0
100 <----- 208 0 0 0
180 <----- 0 0 0 0
200 <----- E-RTD1 207 0 0 0

ACK -----> 207 0
Pause [ 5000ms] 207 0
BYE -----> 183 0 0 0
200 <----- 183 0 0 0

----- [+-|*[/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----

```

```

root@kalogirou:~
File Edit View Search Terminal Help
6 active channels
6 active calls
10 calls processed
kalogirou*CLI> core show channels count
11 active channels
11 active calls
33 calls processed
kalogirou*CLI> core show channels count
16 active channels
16 active calls
78 calls processed
kalogirou*CLI> core show channels count
21 active channels
21 active calls
125 calls processed
kalogirou*CLI> core show channels count
26 active channels
26 active calls
195 calls processed
kalogirou*CLI>

```

Twinkle

File Edit Call Message Registration Services View Diamondcard Help

Buddy list
mic...

User:

Call:

Display

Twinkle 1.4.2, 25 Φεβρουάριος 2009
Copyright (C) 2005-2009 Michel de Boer

Τετ 10:51:36
michalis, registration succeeded (expires = 3600 seconds)

Τετ 10:52:49
Line 1: far end answered call.

Line status

Line 1: established g711a 0:01:41

[Asterisk CLI 1.6- ...] [Untitled Docume... [Pictures] root@mk-vm:/usr... root@kalogirou:~ root@kalogirou:~ Twinkle

Applications Places

en 10:54:43 PM Michalis Kalogirou

root@kalogirou:~

File Edit View Search Terminal Help

```
top - 10:54:43 up 2 days, 12:18, 3 users, load average: 0.44, 4.25, 4.04
Tasks: 90 total, 2 running, 88 sleeping, 0 stopped, 0 zombie
Cpu(s): 53.1%us, 7.9%sy, 0.0%ni, 32.7%id, 0.0%wa, 0.7%hi, 5.6%si, 0.0%st
Mem: 2058808k total, 2024620k used, 34188k free, 48152k buffers
Swap: 2064376k total, 88k used, 2064288k free, 1674092k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19920	root	15	0	612m	53m	9632	S	64.9	2.7	0:29.60	asterisk
2149	mysql	18	0	254m	26m	4412	S	0.7	1.3	19:55.76	mysqld
8476	root	15	0	366m	23m	22m	S	0.3	1.2	0:01.62	opensips
15305	root	15	0	90196	3456	2688	S	0.3	0.2	0:00.40	sshd
19936	root	15	0	19336	1360	1080	S	0.3	0.1	0:00.64	mpg123
19960	root	15	0	12752	1080	832	R	0.3	0.1	0:00.19	top

root@mk-vm:/usr/src/sipp/sipp-3.3

File Edit View Search Terminal Help

```
----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
6.0(0 ms)/1.000s 5060 87.64 s 276 62.217.124.222:5060(UDP)

6 new calls during 1.002 s period 0 ms scheduler resolution
30 calls (limit 90) Peak was 31 calls, after 84 s
1 Running, 182 Paused, 12 Woken up
0 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets

Messages Retrans Timeout Unexpected-Msg
INVITE -----> 276 0 0 0
100 <----- 276 0 0 0
180 <----- 0 0 0 0
200 <----- E-RTD1 276 0 0 0

ACK -----> 276 0
Pause [ 5000ms] 276 0
BYE -----> 246 0 0 0
200 <----- 246 0 0 0

----- [+-|*[/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
```

root@kalogirou:~

File Edit View Search Terminal Help

```
11 active channels
11 active calls
33 calls processed
kalogirou*CLI> core show channels count
16 active channels
16 active calls
78 calls processed
kalogirou*CLI> core show channels count
21 active channels
21 active calls
125 calls processed
kalogirou*CLI> core show channels count
26 active channels
26 active calls
195 calls processed
kalogirou*CLI> core show channels count
31 active channels
31 active calls
270 calls processed
kalogirou*CLI>
```

Twinkle

File Edit Call Message Registration Services View Diamondcard Help

Buddy list
mic...

User:

Call: Dial

Display

```
Twinkle 1.4.2, 25 Φεβρουάριος 2009
Copyright (C) 2005-2009 Michel de Boer

Τετ 10:51:36
michalis, registration succeeded (expires = 3600 seconds)

Τετ 10:52:49
Line 1: far end answered call.
```

Line status

Line 1: established g711a 0:01:54

[Asterisk CLI 1.6- ...] [Untitled Docume...] [Pictures] root@mk-vm:/usr... root@kalogirou:~ root@kalogirou:~ Twinkle

Applications Places

en 10:54:59 ΠΜ Michalis Kalogirou

root@kalogirou:~

File Edit View Search Terminal Help

```
top - 10:54:58 up 2 days, 12:18, 3 users, load average: 1.38, 4.25, 4.04
Tasks: 90 total, 2 running, 88 sleeping, 0 stopped, 0 zombie
Cpu(s): 62.6%us, 8.8%sy, 0.0%ni, 22.9%id, 0.0%wa, 0.7%hi, 5.1%si, 0.0%st
Mem: 2058808k total, 2030488k used, 28320k free, 48176k buffers
Swap: 2064376k total, 88k used, 2064288k free, 1674252k cached
```

PID	USER	PR	NI	VRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19920	root	15	0	614m	59m	9632	S	73.5	2.9	0:39.58	asterisk
2149	mysql	18	0	254m	26m	4412	S	1.3	1.3	19:55.92	mysqld
8475	root	15	0	366m	23m	22m	S	0.7	1.2	0:01.32	opensips
376	root	10	-5	0	0	0	S	0.3	0.0	0:45.55	kjournald
8473	root	15	0	367m	26m	25m	S	0.3	1.3	0:01.70	opensips
8474	root	15	0	366m	23m	22m	S	0.3	1.2	0:01.79	opensips

root@mk-vm:/usr/src/sipp/sipp-3.3

File Edit View Search Terminal Help

```
----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
7.0(0 ms)/1.000s 5060 103.67 s 382 62.217.124.222:5060(UDP)

7 new calls during 1.002 s period 0 ms scheduler resolution
35 calls (limit 105) Peak was 36 calls, after 98 s
1 Running, 225 Paused, 14 Woken up
0 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets

Messages Retrans Timeout Unexpected-Msg
INVITE -----> 382 0 0
100 <----- 382 0 0
180 <----- 0 0 0
200 <----- E-RTD1 382 0 0

ACK -----> 382 0
Pause [ 5000ms] 382 0
BYE -----> 347 0 0
200 <----- 347 0 0

----- [+-|*[/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
```

root@kalogirou:~

File Edit View Search Terminal Help

```
16 active channels
16 active calls
78 calls processed
kalogirou*CLI> core show channels count
21 active channels
21 active calls
125 calls processed
kalogirou*CLI> core show channels count
26 active channels
26 active calls
195 calls processed
kalogirou*CLI> core show channels count
31 active channels
31 active calls
270 calls processed
kalogirou*CLI> core show channels count
36 active channels
36 active calls
367 calls processed
kalogirou*CLI>
```

Twinkle

File Edit Call Message Registration Services View Diamondcard Help

Buddy list
mic...

User:

Call: Dial

Display

```
Twinkle 1.4.2, 25 Φεβρουάριος 2009
Copyright (C) 2005-2009 Michel de Boer

Τετ 10:51:36
michalis, registration succeeded (expires = 3600 seconds)

Τετ 10:52:49
Line 1: far end answered call.
```

Line status

Line 1: established g711a 0:02:09

[Asterisk CLI 1.6- ...] [Untitled Docume... [Pictures] root@mk-vm:/usr... root@kalogirou:~ root@kalogirou:~ Twinkle

Applications Places

en 10:55:20 ΠΜ Michalis Kalogirou

```

root@kalogirou:~
File Edit View Search Terminal Help
top - 10:55:19 up 2 days, 12:18, 3 users, load average: 2.02, 4.22, 4.03
Tasks: 90 total, 1 running, 89 sleeping, 0 stopped, 0 zombie
Cpu(s): 75.8%us, 9.1%sy, 0.0%ni, 9.7%id, 0.0%wa, 0.3%hi, 5.0%si, 0.0%st
Mem: 2058808k total, 2036976k used, 21832k free, 48200k buffers
Swap: 2064376k total, 88k used, 2064288k free, 1674568k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM   TIME+  COMMAND
 19920 root        15   0  618m  64m  9632 S  87.9   3.2   0:57.12 asterisk
  2149 mysql      18   0   254m  26m  4412 S   1.0   1.3  19:56.17 mysqld
  8475 root        15   0   366m  23m  22m S   0.7   1.2   0:01.38 opensips
   376 root        10  -5    0    0    0 S   0.3   0.0   0:45.56 kjournald
19936 root        15   0 19336 1360 1080 S   0.3   0.1   0:00.75 mpg123
    1 root        15   0 10372  700  588 S   0.0   0.0   0:00.93 init

```

```

root@mk-vm:/usr/src/sipp/sipp-3.3
File Edit View Search Terminal Help

----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length)  Port  Total-time  Total-calls  Remote-host
8.0(0 ms)/1.000s  5060  124.72 s    543          62.217.124.222:5060(UDP)

8 new calls during 1.002 s period      0 ms scheduler resolution
41 calls (limit 120)                   Peak was 43 calls, after 122 s
1 Running, 274 Paused, 17 Woken up
0 dead call msg (discarded)           0 out-of-call msg (discarded)
3 open sockets

Messages  Retrans  Timeout  Unexpected-Msg
INVITE -----> 543      0        0            0
 100 <----- 543      0        0            0
 180 <----- 0        0        0            0
 200 <----- E-RTD1 542      0        0            0

ACK -----> 542      0
Pause [ 5000ms] 542      0
BYE -----> 502      0        0            0
 200 <----- 502      0        0            0

----- [+-|*[/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----

```

```

root@kalogirou:~
File Edit View Search Terminal Help
21 active channels
21 active calls
125 calls processed
kalogirou*CLI> core show channels count
26 active channels
26 active calls
195 calls processed
kalogirou*CLI> core show channels count
31 active channels
31 active calls
270 calls processed
kalogirou*CLI> core show channels count
36 active channels
36 active calls
367 calls processed
kalogirou*CLI> core show channels count
41 active channels
41 active calls
514 calls processed
kalogirou*CLI>

```

```

Twinkle
File Edit Call Message Registration Services View Diamondcard Help

Buddy list
mic...

User: michalis
Call:
Dial

Display
Twinkle 1.4.2, 25 Φεβρουάριος 2009
Copyright (C) 2005-2009 Michel de Boer

Τετ 10:51:36
michalis, registration succeeded (expires = 3600 seconds)

Τετ 10:52:49
Line 1: far end answered call.

Line status
Line 1: established g711a 0:02:30

```

[Asterisk CLI 1.6- ...] [Untitled Docume... [Pictures] root@mk-vm:/usr... root@kalogirou:~ root@kalogirou:~ Twinkle

Applications Places

en 10:55:51 ΠΜ Michalis Kalogirou

root@kalogirou:~

File Edit View Search Terminal Help

```
top - 10:55:49 up 2 days, 12:19, 3 users, load average: 9.04, 5.70, 4.53
Tasks: 90 total, 3 running, 87 sleeping, 0 stopped, 0 zombie
Cpu(s): 85.7%us, 10.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.7%hi, 3.7%si, 0.0%st
Mem: 2058808k total, 2038040k used, 20768k free, 48224k buffers
Swap: 2064376k total, 88k used, 2064288k free, 1675000k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19920	root	15	0	621m	65m	9632	S	97.4	3.2	1:25.17	asterisk
2149	mysql	18	0	254m	26m	4412	S	0.7	1.3	19:56.46	mysqld
8474	root	15	0	366m	23m	22m	S	0.3	1.2	0:01.85	opensips
8475	root	15	0	366m	23m	22m	S	0.3	1.2	0:01.43	opensips
8476	root	15	0	366m	23m	22m	S	0.3	1.2	0:01.69	opensips
19960	root	15	0	12752	1080	832	R	0.3	0.1	0:00.28	top

root@mk-vm:/usr/src/sipp/sipp-3.3

File Edit View Search Terminal Help

```
----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
 9.0(0 ms)/1.000s 5060 156.78 s 818 62.217.124.222:5060(UDP)

9 new calls during 1.002 s period 0 ms scheduler resolution
98 calls (limit 135) Peak was 102 calls, after 150 s
1 Running, 324 Paused, 12 Woken up
54 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets
```

	Messages	Retrans	Timeout	Unexpected-Msg
INVITE ----->	818	0	0	
100 <-----	818	0	0	0
180 <-----	0	0	0	26
200 <-----	E-RTD1 706	5	0	0
ACK ----->	706	5		
Pause [5000ms]	706			0
BYE ----->	694	138	0	
200 <-----	648	0	0	46

```
----- [+]/*|/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
Last Error: Dead call 673-9480@147.52.82.82 (aborted at index 7), receiv...
```

root@kalogirou:~

File Edit View Search Terminal Help

```
36 active channels
36 active calls
367 calls processed
kalogirou*CLI> core show channels count
41 active channels
41 active calls
514 calls processed
kalogirou*CLI> core show channels count
48 active channels
48 active calls
696 calls processed
kalogirou*CLI> core show channels count
48 active channels
48 active calls
696 calls processed
kalogirou*CLI> core show channels count
46 active channels
46 active calls
698 calls processed
kalogirou*CLI>
```

Twinkle

File Edit Call Message Registration Services View Diamondcard Help

Buddy list
mic...

User:

Call:

Display

```
Twinkle 1.4.2, 25 Φεβρουάριος 2009
Copyright (C) 2005-2009 Michel de Boer

Τετ 10:51:36
michalis, registration succeeded (expires = 3600 seconds)

Τετ 10:52:49
Line 1: far end answered call.
```

Line status

Line 1: established g711a 0:03:02

[Asterisk CLI 1.6- ...] [Untitled Docume...] [Pictures] root@mk-vm:/usr... root@kalogirou:~ root@kalogirou:~ Twinkle

Applications Places

root@kalogirou:~

```
File Edit View Search Terminal Help
top - 10:56:01 up 2 days, 12:19, 3 users, load average: 15.09, 7.17, 5.03
Tasks: 90 total, 6 running, 84 sleeping, 0 stopped, 0 zombie
Cpu(s): 82.9%us, 11.4%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.7%hi, 5.0%si, 0.0%st
Mem: 2058808k total, 2039360k used, 19448k free, 48236k buffers
Swap: 2064376k total, 88k used, 2064288k free, 1675064k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19920	root	15	0	626m	65m	9632	S	98.8	3.2	1:37.06	asterisk
8478	root	15	0	366m	29m	28m	S	0.3	1.5	0:04.70	opensips
19936	root	15	0	19336	1360	1080	S	0.3	0.1	0:00.87	mpg123
1	root	15	0	10372	700	588	S	0.0	0.0	0:00.93	init
2	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	migration/0
3	root	34	19	0	0	0	R	0.0	0.0	0:00.00	ksoftirqd/0

root@mk-vm:/usr/src/sipp/sipp-3.3

```
File Edit View Search Terminal Help
----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
9.0(0 ms)/1.000s 5060 168.80 s 927 62.217.124.222:5060(UDP)

10 new calls during 1.002 s period 0 ms scheduler resolution
97 calls (limit 135) Peak was 111 calls, after 165 s
1 Running, 336 Paused, 11 Woken up
75 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets
```

	Messages	Retrans	Timeout	Unexpected-Msg
INVITE ----->	926	0	0	
100 <-----	926	0	0	0
180 <-----	0	0	0	124
200 <-----	E-RTD1 706	5	0	0
ACK ----->	706	5		
Pause [5000ms]	706			0
BYE ----->	706	174	0	
200 <-----	648	0	0	58

```
----- [+]/*|/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
Last Error: Aborting call on unexpected message for Call-Id '830-9480@14...
```

root@kalogirou:~

```
File Edit View Search Terminal Help
41 active channels
41 active calls
514 calls processed
kalogirou*CLI> core show channels count
48 active channels
48 active calls
696 calls processed
kalogirou*CLI> core show channels count
48 active channels
48 active calls
696 calls processed
kalogirou*CLI> core show channels count
46 active channels
46 active calls
698 calls processed
kalogirou*CLI> core show channels count
60 active channels
60 active calls
736 calls processed
kalogirou*CLI>
```

Twinkle

File Edit Call Message Registration Services View Diamondcard Help

Buddy list
mic...

User: michalis

Call: Dial

Display

```
Twinkle 1.4.2, 25 Φεβρουάριος 2009
Copyright (C) 2005-2009 Michel de Boer

Τετ 10:51:36
michalis, registration succeeded (expires = 3600 seconds)

Τετ 10:52:49
Line 1: far end answered call.
```

Line status

Line 1: established g711a 0:03:14

[Asterisk CLI 1.6-... [Untitled Docume... [Pictures] root@mk-vm:/usr... root@kalogirou:~ root@kalogirou:~ Twinkle

Applications Places

en 10:56:21 ΠΜ Michalis Kalogirou

root@kalogirou:~

File Edit View Search Terminal Help

```
top - 10:56:19 up 2 days, 12:19, 3 users, load average: 16.71, 7.94, 5.31
Tasks: 90 total, 7 running, 83 sleeping, 0 stopped, 0 zombie
Cpu(s): 87.1%us, 6.9%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.7%hi, 5.3%si, 0.0%st
Mem: 2058808k total, 2041344k used, 17464k free, 48252k buffers
Swap: 2064376k total, 88k used, 2064288k free, 1675064k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19920	root	15	0	626m	65m	9632	S	99.7	3.2	1:55.03	asterisk
19960	root	15	0	12752	1080	832	R	0.3	0.1	0:00.32	top
1	root	15	0	10372	700	588	S	0.0	0.0	0:00.93	init
2	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	migration/0
3	root	34	19	0	0	0	R	0.0	0.0	0:00.00	ksoftirqd/0
4	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	watchdog/0

root@mk-vm:/usr/src/sipp/sipp-3.3

File Edit View Search Terminal Help

```
----- Scenario Screen ----- [1-9]: Change Screen --
Call-rate(length) Port Total-time Total-calls Remote-host
9.0(0 ms)/1.000s 5060 185.83 s 1049 62.217.124.222:5060(UDP)

9 new calls during 1.002 s period 0 ms scheduler resolution
96 calls (limit 135) Peak was 111 calls, after 165 s
1 Running, 402 Paused, 4 Woken up
87 dead call msg (discarded) 0 out-of-call msg (discarded)
3 open sockets
```

	Messages	Retrans	Timeout	Unexpected-Msg
INVITE ----->	1049	0	0	
100 <-----	1049	0	0	0
180 <-----	0	0	0	247
200 <-----	E-RTD1 706	5	0	0
ACK ----->	706	5		
Pause [5000ms]	706			0
BYE ----->	706	174	0	
200 <-----	648	0	0	58

```
----- [+]/*|/]: Adjust rate ---- [q]: Soft exit ---- [p]: Pause traffic -----
Last Error: Aborting call on unexpected message for Call-Id '953-9480@14...
```

root@kalogirou:~

File Edit View Search Terminal Help

```
48 active channels
48 active calls
696 calls processed
kalogirou*CLI> core show channels count
48 active channels
48 active calls
696 calls processed
kalogirou*CLI> core show channels count
46 active channels
46 active calls
698 calls processed
kalogirou*CLI> core show channels count
60 active channels
60 active calls
736 calls processed
kalogirou*CLI> core show channels count
60 active channels
60 active calls
736 calls processed
kalogirou*CLI>
```

Twinkle

File Edit Call Message Registration Services View Diamondcard Help

Buddy list
mic...

User: michalis

Call: Dial

Display

```
Twinkle 1.4.2, 25 Φεβρουάριος 2009
Copyright (C) 2005-2009 Michel de Boer

Τετ 10:51:36
michalis, registration succeeded (expires = 3600 seconds)

Τετ 10:52:49
Line 1: far end answered call.
```

Line status

Line 1: established g711a 0:03:31

[Asterisk CLI 1.6-... [Untitled Docume... [Pictures] root@mk-vm:/usr... root@kalogirou:~ root@kalogirou:~ Twinkle

