ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Διδακτικής της Τεχνολογίας και Ψηφιακών Συστημάτων

Εφαρμογές του πρωτοκόλλου SIP στην πλατφόρμα Asterisk

Ψιαχούλιας Σ. Αργύριος

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ιανουάριος 2008

Περίληψη

Στον κόσμο των υπολογιστών, ένα πρωτόκολλο είναι μια σύμβαση η οποία ενεργοποιεί ή ελέγχει τη σύνδεση, επικοινωνία και μετάδοση δεδομένων μεταξύ δυο σημείων (endpoints) ενός δικτύου. Το SIP είναι ένα application-layer control πρωτόκολλο για δημιουργία, μορφοποίηση και τερματισμό συνόδων με έναν ή περισσότερους χρήστες. Χρησιμοποιείται ευρέως για σηματοδοσία στις Voice over IP (VoIP) εφαρμογές.

Η παρούσα εργασία αποτελείται από τρία κεφάλαια.

Στο πρώτο κεφάλαιο γίνεται μια εισαγωγή στις γενικές αρχές του πρωτοκόλλου. Περιγράφονται η εξέλιξη του μέχρι και την σημερινή του μορφή. Αναλύονται με λεπτομέρεια τα SIP User Agents και οι διάφορες μορφές SIP Servers. Τέλος, αναφέρονται τα προτερήματα του έναντι των υπολοίπων αντιστοίχων πρωτοκόλλων.

Στο δεύτερο κεφάλαιο, περιγράφεται αναλυτικά η λειτουργιά του πρωτοκόλλου SIP. Συγκεκριμένα, γίνεται εκτενής αναφορά των αιτημάτων και των απαντήσεων που χρησιμοποιούνται με πρακτικά παραδείγματα για κάθε ένα από αυτά. Παράλληλα, παρουσιάζονται οι proxy servers του SIP και η αλληλεπίδραση τους με τα μηνύματα του SIP. Τέλος, περιγράφεται η δομή των header και bodies και πως αυτά καθορίζουν τις συναλλαγές μεταξύ των endpoints.

Στο τρίτο κεφάλαιο, γίνεται αναφορά στο Asterisk, ένα λογισμικό ανοιχτού κώδικα το οποίο χρησιμοποιείται για την υλοποίηση ιδιωτικών τηλεφωνικών κέντρων (private branch exchange – PBX). Περιγράφεται το πώς ξεκίνησε, το τι ακριβώς είναι και πως λειτουργεί. Στο τελευταίο μέρος, δίνονται αναλυτικά παραδείγματα χρήσης του, τονίζοντας τη σημασία του πρωτοκόλλου SIP.

Συγκεκριμένα, περιγράφεται η δημιουργία ενός απλού PBX και η πραγματοποίηση κλήσης με ήχο και βίντεο σε πραγματικό χρόνο μεταξύ δυο χρηστών. Ως τελευταίο βήμα, γίνεται ανάλυση μέσω του προγράμματος Wireshark των πακέτων που

ανταλλάσσονται σε μια κλήση όπως η παραπάνω, προκειμένου να γίνει και πρακτικά κατανοητή η λειτουργιά του πρωτοκόλλου SIP.

Πίνακας Περιεχομένων

ΚΕΦΑΛΑΙΟ Α'

ΚΕΦΑΛΑΙΟ Β'

ΚΕΦΑΛΑΙΟ Γ'

ΚΕΦΑΛΑΙΟ Α':

Εισαγωγή και βασικές αρχές του πρωτοκόλλου SIP

1.1 Ιστορική αναδρομή

Είναι γνωστό ότι η αρχιτεκτονική πολυμέσων διαδικτύου περιλαμβάνει πολλά διαφορετικά πρωτόκολλα. Αυτά τα πρωτόκολλα όμως δεν αναπτύχθηκαν το ίδιο χρονικό διάστημα. Αρχίζοντας με τα πρώτα συστήματα πολυμέσων, αυτή η αρχιτεκτονική εξελισσόταν δυναμικά. Νέα πρωτόκολλα σχεδιάστηκαν και υπάρχοντα βελτιώθηκαν. Με το πέρασμα του χρόνου, το Διαδίκτυο έκανε προσπάθειες για να παρέχει όλο και περισσότερες υπηρεσίες πολυμέσων. Παρόλα αυτά, αυτή η αρχιτεκτονική είχε ακόμα ένα ελλείπον κομμάτι: δεν είχε έναν απλό τρόπο να καλέσει τους χρηστές σε μια συγκεκριμένη σύνοδο. Μια πολλαπλής διανομής σύνοδος, παραδείγματος χάριν, μπορούσε να αναγγελθεί χρησιμοποιώντας το πρωτόκολλο ανακοίνωσης συνόδου (SAP), αλλά ήταν στα χέρια του δέκτη να ελέγχει όλες τις αναγγελθείσες συνόδους περιοδικά, για να βρει αυτή που θέλει να συμμετάσχει. Ήταν αδύνατο για έναν χρήστη να ενημερώσει έναν άλλο χρήστη για μια σύνοδο και να τον προσκαλέσει να συμμετέχει σε αυτήν.

Ας υποθέσουμε ότι παρακολουθούμε μια ταινία η οποία εκπέμπεται multicast στο Mbone και θέλουμε να καλέσουμε ένα φίλο να την δει και αυτός. Χρειαζόμαστε ένα απλό μέσο για να ειδοποιήσουμε το φίλο μας, να του στείλουμε μια περιγραφή, και να τον προσκαλέσουμε στη σύνοδο (σχήμα 1-1). Η πρόσκληση των χρηστών στις συνόδους Mbone ήταν ο αρχικός σκοπός του SIP όταν η ομάδα εργασίας εφαρμοσμένης μηχανικής Διαδικτύου (IETF) αρχικά την υιοθέτησε. Το πρωτόκολλο έχει εξελιχθεί σταθερά και το SIP χρησιμοποιείται πλέον για να προσκαλέσει τους χρήστες σε όλους τους τύπους συνόδων, συμπεριλαμβανομένων των πολλαπλής διανομής και από σημείο σε σημείο συνόδων.

Το SIP, όπως το ξέρουμε σήμερα, δεν σχεδιάστηκε από την αρχή, αλλά πρόεκυψε από τη συγχώνευση δύο πρωτοκόλλων IETF. Το SIP πήρε τα καλύτερα χαρακτηριστικά γνωρίσματα από κάθε πρωτόκολλο και έπειτα, όλες οι προσπάθειες εντός της κοινότητας συγκλείνανε σε αυτό.

Σχήμα 1-1: Πρόσκληση σε σύνοδο

1.2 Session Invitation Protocol: SIPv1

Αν και οι πρώτες μεταδόσεις φωνής μέσα από τα packet-switched δίκτυα ξεκίνησαν περίπου το 1974, τα πρώτα συστήματα διασκέψεων πολυμέσων εμφανίστηκαν στις αρχές της δεκαετία του '90. Ο Thierry Turletti ανέπτυξε το σύστημα συνεδριάσεων μέσω video INRIA (IVS), ένα σύστημα για ακουστικές και τηλεοπτικές μεταδόσεις μέσω του Διαδικτύου. Ο χρήστης IVS θα μπορούσε να καλέσει έναν άλλο χρήστη και θα μπορούσαν να ξεκινήσουν μια σύνοδο unicast. Το IVS θα μπορούσε επίσης να χρησιμοποιηθεί στις πολλαπλής διανομής συνόδους. Ως αποτέλεσμα, το IVS

συνέβαλε να αναπτυχθεί το σε πραγματικό χρόνο πρωτόκολλο μεταφορών (RTP) για τα βίντεο με φορμά H.261 [RFC 2032].

Σύντομα έκτοτε, η Eve Schooler ανέπτυξε τον έλεγχο διασκέψεων πολυμέσων (MMCC). Το λογισμικό MMCC παρείχε point-to-point και multipoint τηλεσυνεδριάσεις, με εργαλεία ήχου και εικόνας.

Για να συνδέσει τους διάφορους χρήστες, το MMCC χρησιμοποίησε το πρωτόκολλο ελέγχου σύνδεσης (CCP). Μια χαρακτηριστική συναλλαγή αποτελούταν από μια αίτηση (από το χρήστη) και μια απάντηση (από το μακρινό χρήστη). Για τη μεταφορά, το CCP χρησιμοποιούσε πρωτόκολλο διαγραμμάτων δεδομένων χρηστών (UDP) οπότε εφάρμοζε time-outs και αναμεταδόσεις για να εξασφαλίσει την αξιόπιστη παράδοση των μηνυμάτων πρωτοκόλλου.

Αυτά τα δύο πρώτα συστήματα πολυμέσων έκαναν την αρχή στο σχέδιο του Session Invitation Protocol που δημιουργήθηκε από τον Mark Handley και την Eve Schooler. Η πρώτη έκδοση του SIP, το SIPv1, υποβλήθηκε στο IETF στις 22 Φεβρουαρίου, 1996 [10]. Το SIPv1 χρησιμοποιούσε το πρωτόκολλο περιγραφής συνόδου (SDP) για να περιγράφει τις συνόδους και το UDP για μεταφορά. Ήταν βασισμένο σε κείμενο.

Η έννοια των εγγραφών (registrations) στους server διευθύνσεων διασκέψεων ήταν προεξέχουσα στο SIPv1. Όταν ο χρήστης έκανε register την θέση του, ένας address server ήταν σε θέση να καθοδηγήσει (route) τις προσκλήσεις στον κατάλληλο χρήστη και να παρέχει επίσης ένα ορισμένο επίπεδο κινητικότητας. Εάν κάποιος ήταν μακριά από κανονικό του/της workstation σε επιχειρησιακό ταξίδι, παραδείγματος χάριν, αυτός ο χρήστης θα μπορούσε να επιλέξει να καταχωρήσει τον προσωρινό workstation του/της και να λαμβάνει τις προσκλήσεις στις τοπικές διασκέψεις. Ειδικότερα, το SIPv1 χειριζόταν μόνο το ξεκίνημα της συνόδου. Η σηματοδοσία σταματούσε μόλις ο χρήστης έκανε join.

1.3 Simple Conference Invitation Protocol: SCIP

Επίσης στις 22 Φεβρουαρίου 1996, ο Henning Schulzrinne υπέβαλε ένα σχέδιο στο IETF που διευκρινίζει το απλό πρωτόκολλο πρόσκλησης διασκέψεων (SCIP). Το SCIP ήταν επίσης ένας μηχανισμός για κάλεσμα χρηστών σε συνόδους point-to-point και multicast. Ήταν βασισμένο στο πρωτόκολλο μεταφοράς υπερκειμένων (HTTP) και κατά συνέπεια χρησιμοποιούσε το πρωτόκολλο ελέγχου μετάδοσης (TCP) ως πρωτόκολλο μεταφορών.

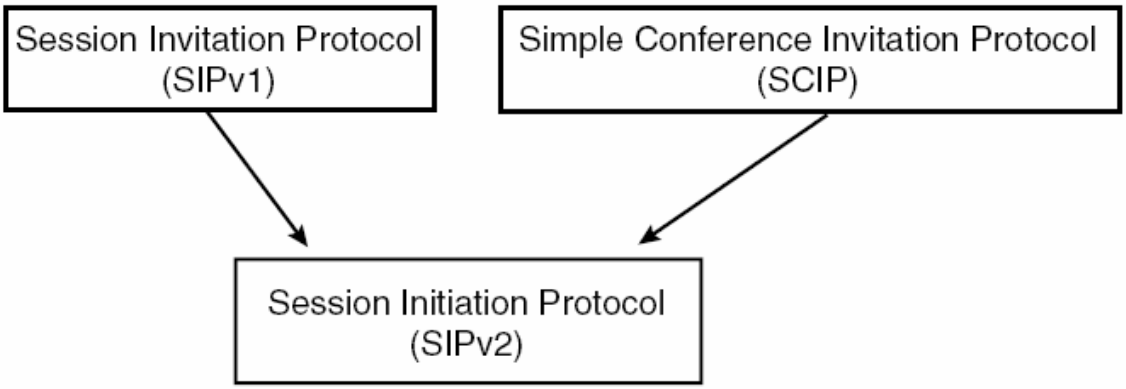
Όπως το SIPv1, ήταν βασισμένο σε κείμενο. Το SCIP χρησιμοποιούσε διευθύνσεις ηλεκτρονικού ταχυδρομείου ως προσδιοριστικά για τους χρήστες και στόχευε να παρέχει ένα καθολικό προσδιοριστικό για σύγχρονες και ασύγχρονες επικοινωνίες. Η σηματοδοσία του SCIP παρέμενε μετά από την αρχή της συνόδου για να επιτρέπει τις αλλαγές παραμέτρων στις τρέχουσες συνόδους και το κλείσιμο υπαρχόντων συνόδων.

1.4 Session Initiation Protocol: SIPv2

Στην 35η συνεδρίαση του IETF στο Λος Άντζελες, η Schooler παρουσίασε το SIP και ο Schulzrinne το SCIP. Κατά τη διάρκεια αυτής της συνεδρίασης και έως την 36η συνεδρίαση, ακολούθησαν έντονες συζητήσεις. Τελικά, αποφασίστηκε να συγχωνευτούν τα δύο πρωτόκολλα.

Το πρωτόκολλο που προέκυψε κράτησε το όνομα SIP, αλλά άλλαξε την έννοια των αρχικών στο πρωτόκολλο έναρξης συνόδου (Session Initiation Protocol) και έτσι πρόεκυψε η έκδοση αριθμός 2 (σχήμα 1-2).

Ένα internet draft του SIPv2, από τον Mark Hanley, Schulzrinne, και την Schooler, υποβλήθηκε στη IETF στο San Jose κατά τη διάρκεια της 37ης συνεδρίασης τον Δεκέμβριο του 1996. Το νέο SIP βασίστηκε στο HTTP, αλλά θα μπορούσε να χρησιμοποιήσει και τα δύο UDP και TCP ως πρωτόκολλα μεταφορών. Χρησιμοποίησε το SDP για να περιγράψει συνόδους πολυμέσων και ήταν text based. Μέχρι σήμερα, παραμένει η παρούσα έκδοση του SIP.



Σχήμα 1-2: Δημιουργία του SIPv2

Οι προσπάθειες ανάπτυξης του SIP ήταν η ουσία της ομάδας εργασίας Multiparty Multimedia Session Control (MMUSIC) που προήδρευσαν οι Joerg Ott και ColinPerkins. Το πρώτο σχέδιο αναπτύχτηκε από την ανατροφοδότηση των συντακτών και συζητήσεις στο mailing list του MMUSIC. Το 1998, ο Jonathan Rosenberg προστέθηκε ως συντάκτης των προδιαγραφών επειδή είχε συμβάλει έντονα στην διαδικασία εξέλιξης, και τον επόμενο Φεβρουάριο (1999), το SIP έφθασε το προτεινόμενο τυποποιημένο επίπεδο και δημοσιεύθηκε ως RFC 2543.

Καθώς ο χρόνος περνούσε, το SIP κέρδιζε σημαντική θέση στο IETF, με συνέπεια το σχηματισμό μιας νέας ομάδα εργασίας SIP τον Σεπτέμβριο του 1999. Αυτή η ομάδα εργασίας προεδρεύθηκε αρχικά από τους Joerg Ott, Jonathan Rosenberg, και Dean Willis.

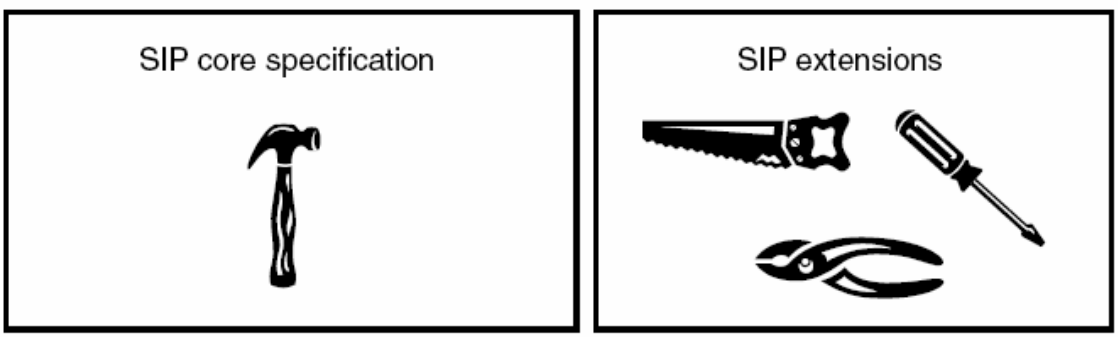
Τον Αύγουστο του 2000, ο Brian Rosen αντικατέστησε τον Rosenberg στην συμπροεδρία. Το Μάρτιο του 2001, η ομάδα του SIP χωρίστηκε στα δύο. Συζητήσεις για τις προδιαγραφές του SIP πραγματοποιούνται τώρα σε μια ομάδα που συνεχίζει να αποκαλείται SIP, ενώ συζητήσεις για τις εφαρμογές του SIP γίνονται από μια ομάδα αποκαλούμενη SIPPING. Αυτή η διαίρεση της εργασίας θα βοηθήσει στη διαχείριση του τεράστιου αριθμού συνεισφορών για το SIP που δέχεται το IETF.

1.5 Η Λειτουργικότητα του SIP

Το RFC 2543 περιγράφει τον πυρήνα του SIP, δηλαδή τη βασική λειτουργία του πρωτοκόλλου. Εκτός από αυτήν την βασική προδιαγραφή, διάφορες επεκτάσεις του SIP έχουν καθοριστεί σε άλλα RFCs και Internet drafts (σχήμα 1-3). Παρακάτω αναφέρεται η λειτουργία που παρέχεται από τη βασική προδιαγραφή.

1.5.1 Εκκίνηση, Τροποποίηση και Τερματισμός Συνόδων

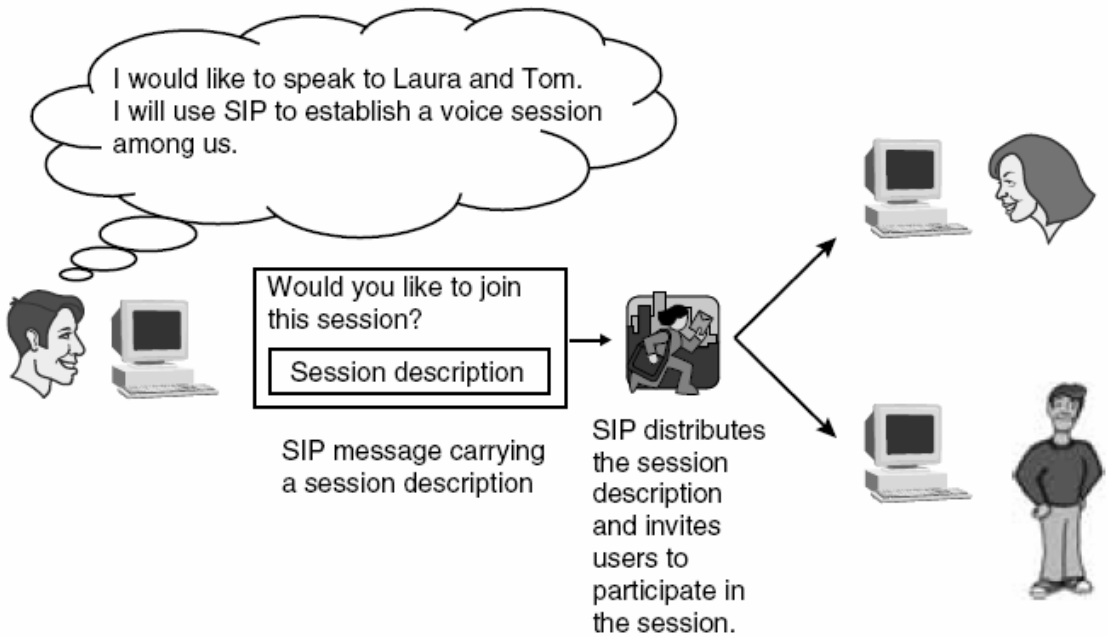
Το SIP ξεκινάει, τροποποιεί, και ολοκληρώνει συνόδους πολυμέσων. Μπορεί να χρησιμοποιηθεί για να προσκαλέσει νέα μέλη σε μια υπάρχουσα σύνοδο ή για να δημιουργήσει καινούργιες συνόδους. Όταν ο χρήστης Bob ενημερώνεται από το SIP ότι μεταδίδεται κάτι ενδιαφέρων μέσω πολλαπλής διανομής στο διαδίκτυο, αναφερόμαστε σε μια υπάρχουσα σύνοδο. Εντούτοις, εάν ο Bob καλεί τη Laura για να το αναφέρει, αυτή η two-party κλήση αποτελεί μια νέα σύνοδο πολυμέσων με ένα ενιαίο άκουσμα. Σε αυτή τη περίπτωση, ο Bob προσκαλεί τη Laura για να συμμετάσχει σε μια σύνοδο που πρόκειται να δημιουργηθεί. Επιπλέον, θα δημιουργηθεί μόνο εάν δύο όροι ικανοποιούνται: (1) Η Laura είναι πρόθυμη να μιλήσει στο Bob και (2) μπορούν να συμφωνήσουν σχετικά με τις παραμέτρους που θα χρησιμοποιηθούν.



Σχήμα 1-3: Επεκτάσεις του SIP

Εν ολίγοις, το SIP χρησιμοποιείται για να διανείμει τις περιγραφές συνόδων μεταξύ πιθανών συμμετεχόντων (σχήμα 1-4). Μόλις διανεμηθεί η περιγραφή συνόδου, το SIP μπορεί να χρησιμοποιηθεί για να διαπραγματευτεί και να τροποποιήσει τις παραμέτρους της συνόδου και να ολοκληρώσει τη σύνοδο.

Το ακόλουθο παράδειγμα εξηγεί όλες αυτές τις λειτουργίες. Ο Bob θέλει να πραγματοποιήσει μια audio-video σύνοδο με τη Laura και σκοπεύει να χρησιμοποιήσει έναν Pulse Code Modulation (PCM) κωδικοποιητή-αποκωδικοποιητή για να κωδικοποιήσει τη φωνή. Σε αυτό το παράδειγμα, η διανομή συνόδου αποτελείται από το Bob που στέλνει στη Laura μια περιγραφή συνόδου με ένα PCM κωδικοποιητή-αποκωδικοποιητή για το τμήμα φωνής της συνόδου. Η Laura προτιμά να χρησιμοποιήσει έναν System for Mobile Communications (GSM) κωδικοποιητή-αποκωδικοποιητή επειδή καταναλώνει λιγότερο εύρος ζώνης. Και οι δύο εγκαθιστούν τελικά έναν GSM κωδικοποιητή-αποκωδικοποιητή, αλλά η σύνοδος δεν μπορεί να αρχίσει μέχρι η διαπραγμάτευση να ολοκληρωθεί.

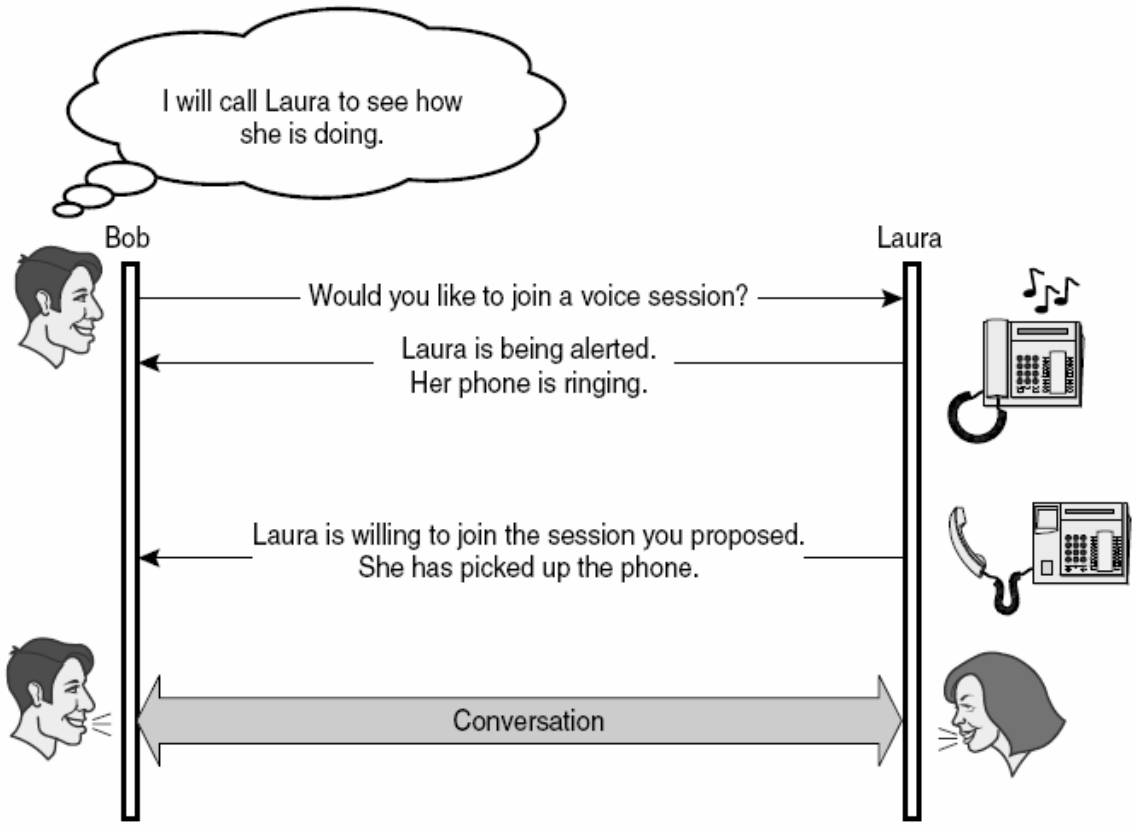


Σχήμα 1-4: Διανομή συνόδων μεταξύ συμμετεχόντων

Ξαφνικά, στη μέση της συνόδου, η Laura αποφασίζει ότι θέλει να διακόψει το τηλεοπτικό σήμα. Τροποποιεί τη σύνοδο για να περιέχει ήχο μόνο. Όταν ο Bob αποφασίζει έπειτα ότι η συνομιλία πρέπει να τελειώσει , η σύνοδος ολοκληρώνεται. Ακριβώς όπως τα τηλεφωνικά συστήματα ενημερώνουν τον καλούντα για τη κατάσταση της κλήσης του/της παίζοντας διαφορετικούς τόνους (busy tones ή ringing tones), το SIP παρέχει σε αυτόν που ξεκινάει μια σύνοδο, πληροφορίες για την πρόοδο της. (σχήμα 1-5).

1.5.2 Κινητικότητα Χρηστών

Το SIP δεν μπορεί να παραδώσει μια περιγραφή συνόδου σε έναν πιθανό συμμετέχοντα μέχρι ότου αυτός εντοπιστεί. Συχνά, ένας χρήστης μπορεί να βρεθεί σε διάφορες θέσεις. Παραδείγματος χάριν, ένας σπουδαστής που χρησιμοποιεί ένα δωμάτιο υπολογιστών στο πανεπιστήμιο συνήθως εργάζεται σε διαφορετικό τερματικό κάθε φορά. Κατά συνέπεια, βρίσκεται σε διαφορετικές διευθύνσεις *πρωτοκόλλου Διαδικτύου* (IP) ανάλογα με ποιος υπολογιστής είναι διαθέσιμος και θέλει να λάβει τις εισερχόμενες προσκλήσεις μόνο στην τρέχουσα θέση του/της. Ένα άλλο πρόσωπο ίσως θελήσει, παραδείγματος χάριν, να λαμβάνει τις προσκλήσεις συνόδου στον τερματικό του/της το πρωί όταν φθάνει στο γραφείο, στον υπολογιστή γραφείου του/της στο σπίτι το βράδυ, και στο κινητό τερματικό του/της όταν ταξιδεύει.

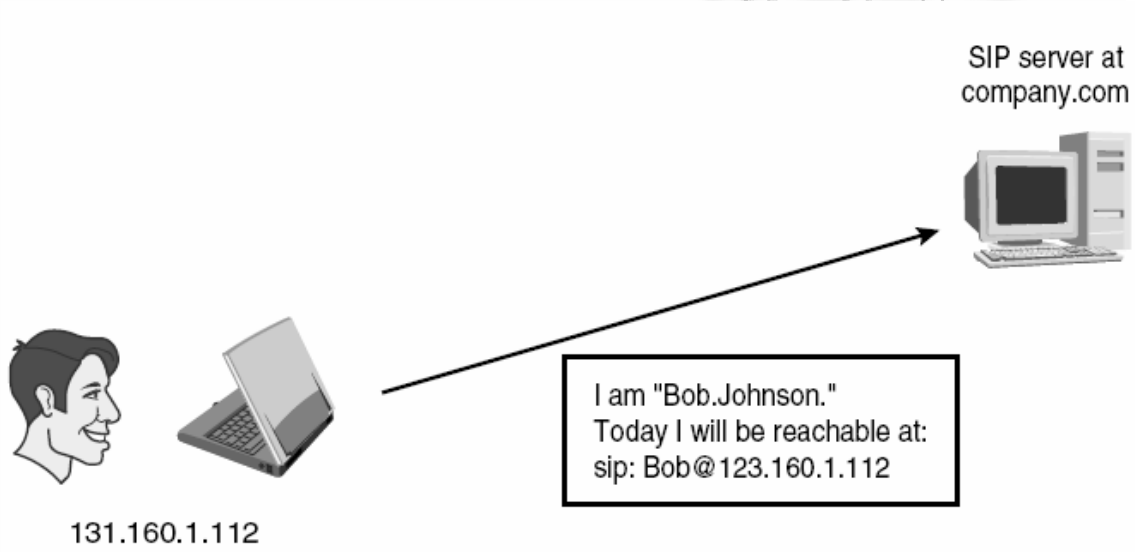


Σχήμα 1-5: Το SIP ενημερωνει για την πρόοδο της συνόδου

SIP URL Ήδη έχουμε αναφέρει ότι το SIP παρέχει κάποια κινητικότητα χρηστών. Οι χρήστες σε ένα περιβάλλον SIP προσδιορίζονται από *το* SIP *Uniform Resource Locators* (URLs). Η μορφή ενός SIP URL είναι παρόμοια με μια διεύθυνση ηλεκτρονικού ταχυδρομείου, γενικά αποτελούμενη από ένα όνομα χρήστη (username) και ένα domain name, δηλαδή όπως: SIP:Bob.Johnson@company.com.
Στο προηγούμενο παράδειγμα, εάν συμβουλευόμαστε τον SIP server που χειρίζεται το domain company.com, θα βρίσκαμε έναν χρήστη του οποίου το username είναι Bob.Johnson. Η URL του Bob ίσως αντ'αυτού να είναι SIP:Bob@131.160.1.112, δείχνοντας ότι ο host του οποίου η διεύθυνση IP είναι 131.160.1.112 έχει έναν χρήστη με το όνομα Bob.

Registrations Έχουμε σημειώσει ότι οι χρήστες καταχωρούν την τρέχουσα θέση τους σε ένα κεντρικό υπολογιστή εάν επιθυμούν να βρεθούν. Σε αυτό το παράδειγμα, ο Bob εργάζεται στο laptop του, του οποίου η διεύθυνση IP είναι

131.160.1.112. Το όνομα της σύνδεσής (login name) του είναι Bob. Καταχωρεί τη τρέχουσα θέση του στον κεντρικό υπολογιστή της επιχείρησης (σχήμα 1-6). Τώρα η Laura θέλει να καλέσει το Bob. Έχει τη δημόσια διεύθυνση SIP του (SIP:Bob.Johnson@company.com) επειδή είναι τυπωμένη στην επιχειρησιακή κάρτα του.



Σχήμα 1-6: Καταχώρηση θέσης του χρήστη

Έτσι όταν ο κεντρικός υπολογιστής του company.com έρχεται σε επαφή και ρωτάται για το Bob. Johnson, ξέρει που μπορεί να βρεθεί και μια σύνδεση είναι δυνατή. Σε αυτήν την κατάσταση, το SIP παρέχει δύο τρόπους λειτουργίας: redirect και proxy. Σε proxy mode, ο κεντρικός υπολογιστής έρχεται σε επαφή με το Bob στη 131.160.1.112 και παραδίδει την περιγραφή συνόδου της Laura σε αυτόν (σχήμα 1-7). Σε redirect mode, ο κεντρικός υπολογιστής λέει στη Laura να δοκιμάσει SIP:Bob@131.160.1.112 (σχήμα 1-8).

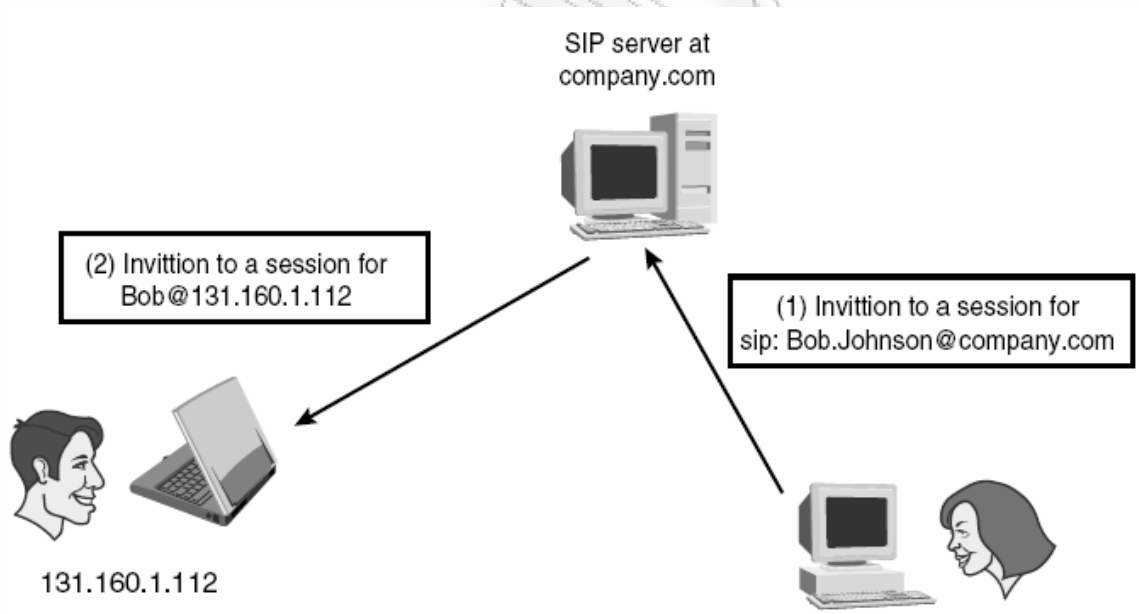
Ένας χρήστης μπορεί να καταχωρήσει διάφορες θέσεις σε έναν κεντρικό υπολογιστή ή ο χρήστης να καταχωρήσει τις θέσεις του/της με διάφορους κεντρικούς υπολογιστές. Δεν είναι ασυνήθιστο διάφοροι κεντρικοί υπολογιστές και θέσεις να έρχονται σε επαφή πριν τελικά ένας χρήστη βρεθεί.

1.6 Διαδικασίες SIP

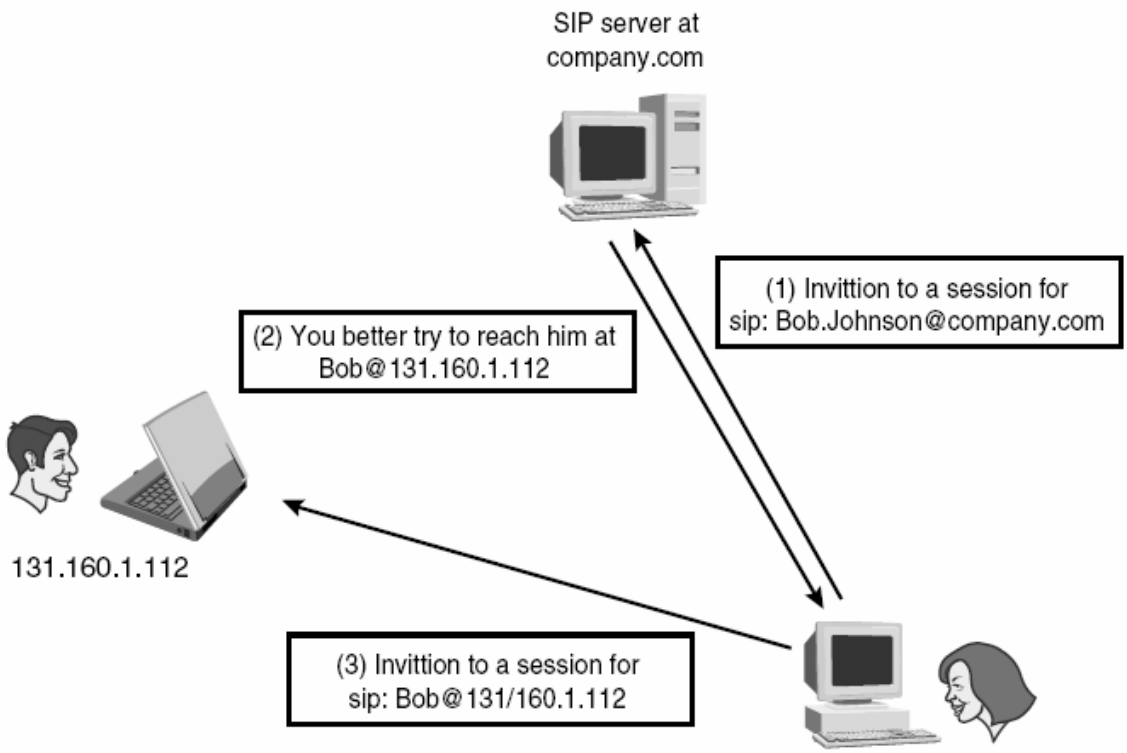
Το πρωτόκολλο SIP καθορίζει διάφορες διαδικασίες και είναι ζωτικής σημασίας να γίνει κατανοητός ο ρόλος τους στην αρχιτεκτονική που χρησιμοποιεί το SIP.

1.6.1 User Agents

User Agent (UA) είναι μια διαδικασία του SIP η οποία αλληλεπιδρά με τον χρήστη. Συνήθως έχει μια διεπαφή (interface) προς τον χρήστη. Ας υποθέσουμε ότι ο Bob θέλει να κάνει μια κλήση μέσω Διαδικτύου με τον υπολογιστή του. Εκκινεί το κατάλληλο πρόγραμμα που περιέχει το SIP User Agent.



Σχήμα 1-7: Proxy SIP server



Σχήμα 1-8: Redirect SIP server

Ο χρήστης αλληλεπιδρά με το UA μέσω του προαναφερθέντος interface, που συχνά είναι ένα παράθυρο με κάποια κουμπιά. Όταν ο Bob επιλέγει το κουμπί "Call Laura", το UA τρέχει τα κατάλληλα SIP μηνύματα που εκκινούν τη κλήση. Η Laura έχει επίσης ένα SIP UA στον υπολογιστή της. Όταν ο δικός της UA της λαμβάνει την πρόσκληση, προειδοποιεί τη Laura με την εμφάνιση ενός pop-up παραθύρου με δύο κουμπιά: "Accept Call" και "Reject Call". Ανάλογα με ποιο κουμπί η Laura επιλέξει, το UA της στέλνει τα κατάλληλα μηνύματα SIP πίσω στο UA του Bob. Όλες οι αλληλεπιδράσεις μεταξύ των χρηστών και του SIP διεκπεραιώνονται από τους UA.

Εντούτοις, μερικά συστήματα που χρησιμοποιούν το SIP δεν συνδέουν άμεσα τους χρήστες μεταξύ τους. Παραδείγματος χάριν, ο Bob μπορεί να κάνει εκτροπή όλες τις προσκλήσεις συνόδου που λαμβάνονται από τα μεσάνυχτα έως τις 7 π.μ. στον SIP αυτόματο τηλεφωνητή του. Η μηχανή θα αρχίσει αυτόματα τις συνόδους προκειμένου να καταγράψει τα μηνύματα. Περιέχει επίσης ένα άλλο UA που δεν

διατηρεί απαραιτήτως την αλληλεπίδραση με το χρήστη, αλλά μπορεί να αποκριθεί στις προσκλήσεις ή να διαβιβάσει τις προσκλήσεις εκ μέρους του Bob.

Media Tools Το SIP παραδίδει μια περιγραφή συνόδου σε ένα SIP UA. Εάν η σύνοδος που περιγράφεται είναι μια σύνοδος φωνής, το UA θα πρέπει να την παραδώσει σε εργαλείο φωνής (voice tool) που θα χειριστεί τον ήχο. Για άλλους τύπους συνόδων, το UA παραδίδει τη σύνοδο στο κατάλληλο εργαλείο.
Τα SIP UA ενσωματώνονται μερικές φορές στο ίδιο interface με τα media tools για τη σύνοδο. Μια ακουστική/τηλεοπτική σύνοδος δεν μπορεί να εκκινήσει χωρίς SIP UA, ένα audio εργαλείο, και ένα video εργαλείο. Εάν αυτά τα τρία συνδυάζονται κάτω από το ίδιο interface, εμφανίζονται ως ενιαία εφαρμογή στο χρήστη: μια εφαρμογή τηλεδιάσκεψης.
Ο διαχωρισμός μεταξύ της διαχείρισης της μεταφοράς της συνόδου από το UA και της διαχείρισης των περιεχομένων της από τα media tools, είναι ισχυρή. Αυτός ο χωρισμός επιτρέπει στο SIP να εκκινήσει οποιοδήποτε τύπο συνόδου.

Πως μοιάζει ένας SIP User Agent; Τα SIP UA εφαρμόζονται πάνω σε πολλά διαφορετικά συστήματα. Μπορούν να τρέξουν, παραδείγματος χάριν, σε έναν υπολογιστή ως εφαρμογή μεταξύ πολλών άλλων, ή μπορούν να εφαρμοστούν σε έναν dedicated server, όπως ένα τηλέφωνο SIP. Ο τύπος συσκευών δεν έχει επιπτώσεις στο SIP. Τα media tools μπορεί να ποικίλουν από συσκευή σε συσκευή ανάλογα με τον τύπο συνόδων που απαιτούνται, αλλά η συμπεριφορά του SIP είναι πάντα η ίδια.

Εν τούτοις, από την άποψη του χρήστη, οι συσκευές SIP μπορούν να φανούν πολύ διαφορετικές από τον έναν στον άλλον. Αυτό συμβαίνει επειδή το interface με τον χρήστη ποικίλλει. Το interface ενός προγράμματος τηλεδιάσκεψης που τρέχει σε έναν υπολογιστή θα είναι πιθανότατα ένα παράθυρο με μια σειρά κουμπιών, αλλά ένα τηλέφωνο SIP πιθανώς θα μοιάζει με ένα παραδοσιακό τηλέφωνο με κουμπιά 0 μέχρι 9, *, και # . Οι συσκευές SIP ποικίλουν από ισχυρούς υπολογιστές με

πρόσβαση στο Διαδίκτυο μέσω μιας σύνδεσης υψηλού-εύρους ζώνης μέχρι μικρές συσκευές με ασύρματες συνδέσεις χαμηλής ταχύτητας. Το σχήμα 1-9 παρουσιάζει μερικά παραδείγματα. Πρέπει να αναφέρουμε ότι η διαδικασία προσαρμογής του SIP σε οικιακές συσκευές βρίσκεται σε εξέλιξη. Επομένως, τα μελλοντικά παραδείγματα συσκευών με SIP User Agents θα μπορούσαν να περιλαμβάνουν ψυγεία, φρυγανιέρες, και λαμπτήρες. Εστιάζουμε στα παραδείγματα τηλεφωνίας επειδή είναι ευκολότερα να κατανοηθούν, εντούτοις, το SIP είναι ισχυρό ακριβώς επειδή μπορεί να χρησιμοποιηθεί για να καθιερώσει οποιοδήποτε είδους σύνοδο. Οι σύνοδοι φωνής είναι μόνο ένα παράδειγμα.

Σχήμα 1-9: Παραδείγματα συσκευών SIP

1.6.2 Redirect Servers

Οι redirect servers βοηθάνε στον εντοπισμό των SIP UA παρέχοντας εναλλακτικές περιοχές όπου ο χρήστης μπορεί να είναι εφικτός. Παραδείγματος χάριν, η Laura θέλει να καλέσει το Bob. Στην οθόνη της, η Laura επιλέγει το κουμπί που λέει "Call Bob". Το UA της προσπαθεί αρχικά τη δημόσια διεύθυνση του Bob, αλλά το domain company.com έχει έναν SIP redirect server που διαχειρίζεται εισερχόμενες προσκλήσεις. Αντ' αυτού λοιπόν, το UA της Laura έρχεται σε επαφή με τον redirect server. Ο server ξέρει ότι ο Bob μπορεί να βρεθεί στο SIP:Bob@131.160.1.112 όταν εργάζεται στο γραφείο του ή στο SIP:Bob@university.com όταν γράφει τη διατριβή του. Κατά συνέπεια, θα συστήσει στο UA της Laura να δοκιμάσει SIP:Bob@131.160.1.112 και SIP:Bob@university.com παρά SIP:Bob.Johnson@company.com. Έχει επίσης την ικανότητα να δώσει προτεραιότητα, οπότε μπορεί να πει στο UA της Laura ότι ο Bob είναι πιθανότερο να βρεθεί στο σχολείο παρά στη δουλειά.

Αφού ενημερωθεί το UA της Laura δοκιμάζει και τις δυο προτεινόμενες SIP διευθύνσεις. Αυτό το παράδειγμα δείχνει ότι ένας redirect server δεν ενεργεί για να εντοπίσει έναν χρήστη, αλλά μόνο επιστρέφει έναν κατάλογο πιθανών θέσεων όπου ο χρήστης μπορεί να είναι. Το UA κάνει προσπάθειες να εντοπίσει τον χρήστη. Σε αυτό το παράδειγμα , το UA της Laura, δοκιμάζει όλες τις πιθανές θέσεις έως ότου βρει τον Bob. Αυτή είναι η κύρια διαφορά μεταξύ ενός redirect server και ενός proxy server. Οι proxy κάνουν συνεχείς προσπάθειες για τον χρήστη παρά να αποστέλλουν νέες πληροφορίες επαφής στον χρήστη.

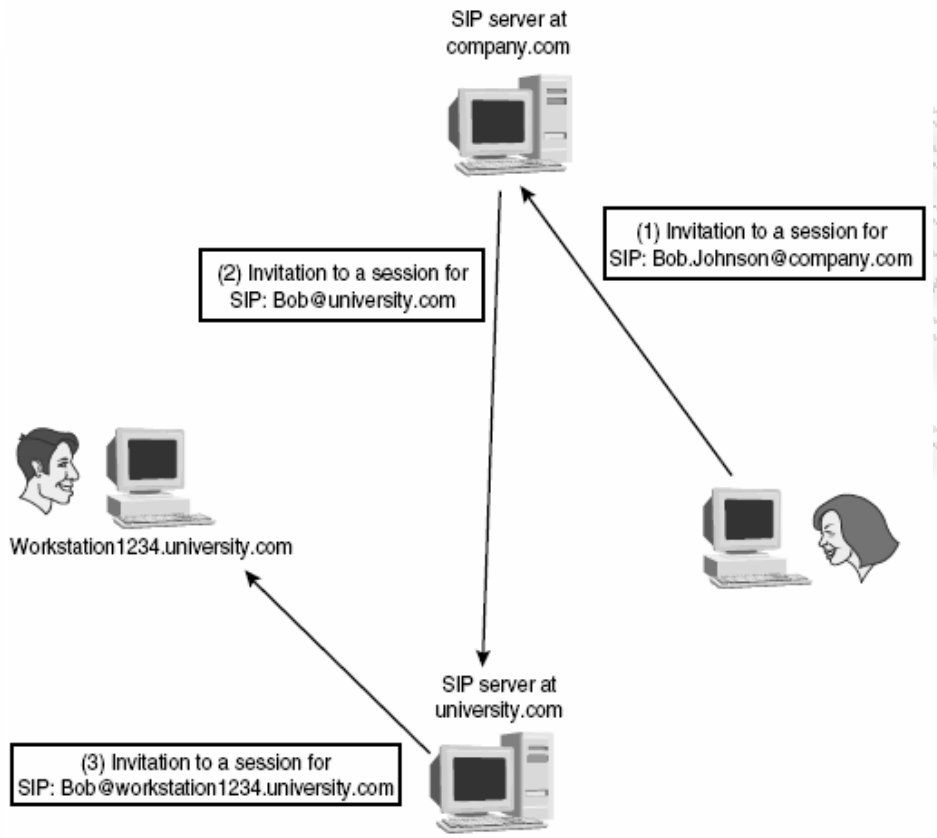
Group Addresses

Οι redirect servers μπορούν επίσης να χρησιμοποιηθούν για να υποστηρίξουν ομαδικές διευθύνσεις. Ας υποθέσουμε ότι η δημόσια διεύθυνση για το τμήμα υποστήριξης μιας εταιρίας Α είναι SIP:support@company.com. Επειδή αυτό το τμήμα πρέπει να δίνει υποστήριξη εικοσιτέσσερις ώρες το εικοσιτετράωρο, διάφοροι άνθρωποι είναι πάντα στην εργασία. Ο Bob δουλεύει από τις 8:00 π.μ. έως τις 4:00 μ.μ, ο Peter από τις 4:00 μ.μ. μέχρι τα μεσάνυχτα, και η Mary από τα

μεσάνυχτα μέχρι τις 8:00 π.μ. Ο redirect server στη company.com είναι σε θέση να επιστρέψει διαφορετικές διευθύνσεις ανάλογα με το χρόνο της ημέρας έτσι ώστε εάν λαμβάνει μια κλήση για το SIP:support@company.com το μεσημέρι, επιστρέφει αυτόματα SIP:Bob.Johnson@company.com.

1.6.3 Proxy Servers

Ας υποθέσουμε ότι το domain company.com έχει έναν proxy server που διαχειρίζεται εισερχόμενες προσκλήσεις. Όταν το UA της Laura δοκιμάζει SIP:Bob. Johnson@company.com, θα φθάσει στον proxy server στο company.com, ο οποίος αμέσως θα δοκιμάσει SIP:Bob@university.com εξ ονόματος του UA της Laura. Εάν το domain του university.com επίσης έχει έναν proxy server, θα δοκιμάσει SIP:Bob@workstation1234.university.com, όπου ο Bob τελικά θα βρεθεί. Σε αυτό το σενάριο, το UA της Laura δοκιμάζει μόνο μια θέση, αλλά αρκετοί proxy είναι ανάμεσα στο εικονικό μονοπάτι (path) μεταξύ των UA (Σχήμα 1-10).



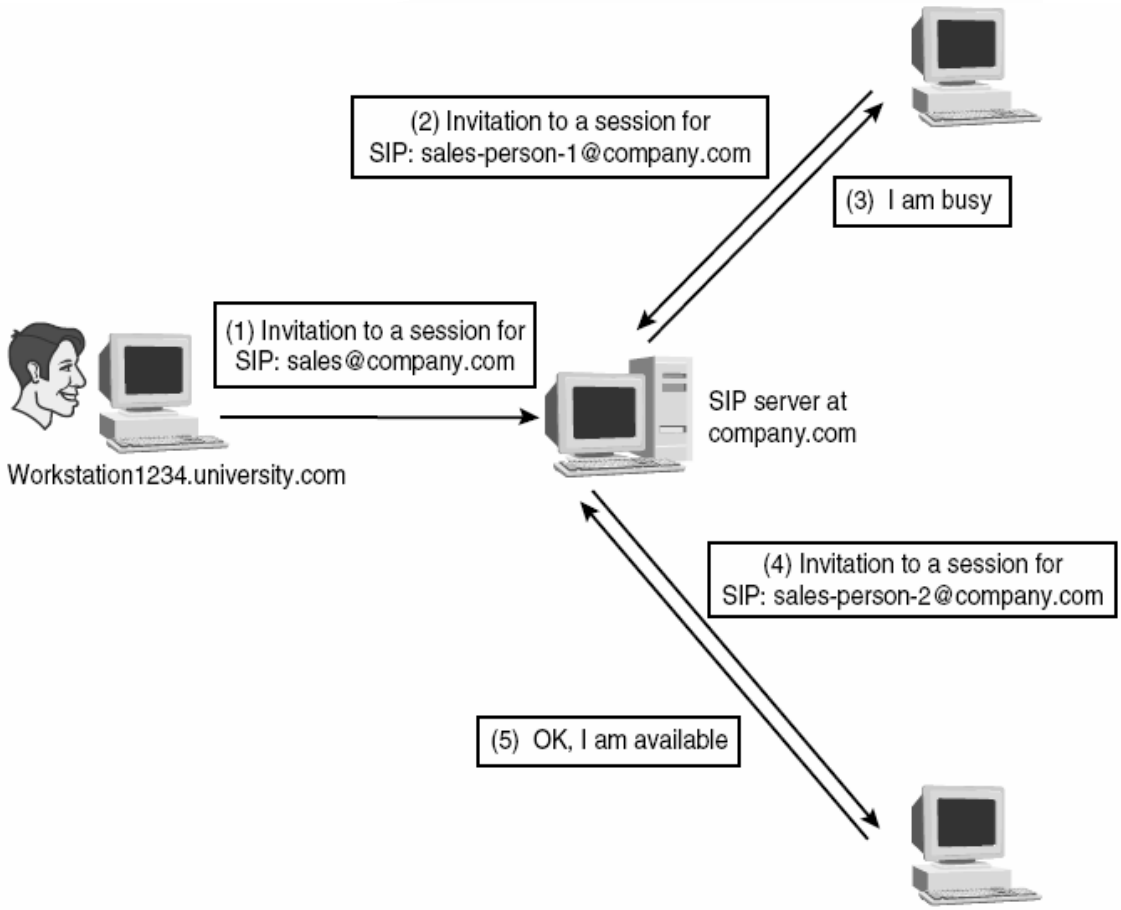
Σχήμα 1-10: Εικονικό μονοπάτι με Proxy Servers

Forking Proxies

Όταν ένας proxy server δοκιμάζει περισσότερες από μια θέσεις για το χρήστη, αυτό λέγεται forking. Οι forking proxies μπορούν να εκτελέσουν παράλληλες ή διαδοχικές αναζητήσεις ανάλογα με τη διαμόρφωσή τους. Μια παράλληλη αναζήτηση αποτελείται από το να δοκιμαστούν όλες τις πιθανές θέσεις συγχρόνως, ενώ μια διαδοχική αναζήτηση αποτελείται από το να δοκιμαστεί κάθε θέση χωριστά.

Group Addresses

Οι proxy servers δημιουργούν επίσης τις διευθύνσεις ομάδας. Το σχήμα 1-11 παρουσιάζει έναν forking server που λαμβάνει μια πρόσκληση για SIP:sales@company.com και δοκιμάζει όλα τα πρόσωπα στο τμήμα πωλήσεων έως ότου βρίσκει ένα που είναι διαθέσιμο.



Σχήμα 1-11: Proxy Servers με Group Addresses

Κατά τη διάρκεια της καθιέρωσης μιας συνόδου, δεν είναι ασυνήθιστο τα δύο είδη κεντρικών υπολογιστών (proxy και redirect) να συνεργάζονται. Ο γενικός όρος SIP server αναφέρεται και στα δύο είδη κεντρικών υπολογιστών χωρίς διαφοροποίηση με βάση τη συμπεριφορά. Στη πραγματικότητα, ο ίδιος κεντρικός υπολογιστής SIP μπορεί να ενεργεί ως redirect ή ως proxy ανάλογα με την περίπτωση. Παραδείγματος χάριν, ένας SIP server μπορεί να κατευθύνει όλες τις προσκλήσεις που παραλαμβάνονται για ορισμένα άτομα και να φερθεί ως proxy για τις υπόλοιπες.

1.7 Χρήσιμα Χαρακτηριστικά

1.7.1 Το SIP Είναι Κομμάτι του IETF Toolkit

Το IETF σχεδίασε το SIP έχοντας υπ' όψιν το μοντέλο του internet. Σαν εργαλείο του IETF toolkit, εκτελεί το ρόλο του και εν συνεχεία εκμεταλλεύεται τους μηχανισμούς του internet για να εκτελέσει πρόσθετες εργασίες. Αυτό παρέχει μεγάλη ευελιξία επειδή τα συστήματα που χρησιμοποιούν το SIP από κοινού με άλλα πρωτόκολλα Διαδικτύου μπορούν να αναβαθμίζονται με έναν ενιαίο (modular) τρόπο.

Παραδείγματος χάριν, εάν ένας νέος μηχανισμός επικύρωσης προταθεί στο IETF, τα συστήματα SIP μπορούν να το χρησιμοποιήσουν χωρίς εφαρμογή τροποποιήσεων πάνω στο SIP. Ένα τέλειο παράδειγμα είναι η εργασία για *το SDP next generation* (SDPng) που το MMUSIC διενεργεί. Όταν το SDPng οριστικοποιηθεί, τα σημερινά συστήματα SIP θα είναι σε θέση να φέρουν τις περιγραφές συνόδου SDPng . Το SIP θα είναι σε θέση να εκμεταλλευθεί τους νέους μηχανισμούς Quality of Service (QoS) το ίδιο καλά. Επομένως το SIP αποτελεί μία δυνατή εγγύηση για το μέλλον [9].

1.7.2 Διαχωρισμός Μεταξύ Εκκίνησης και Περιγραφής μιας Συνόδου

Το SIP κάνει σαφή διάκριση μεταξύ της καθιέρωσης συνόδου και της περιγραφής συνόδου. Ως τμήμα της καθιέρωσης συνόδου, το SIP εντοπίζει τους χρήστες όπως προσφέρονται, αλλά δεν ασχολείται με το τι μπορούν να κάνουν οι χρήστες μόλις καθιερωθεί η σύνοδος. Δεν καθορίζει το πώς μια σύνοδος πρέπει να περιγραφεί ή τύπους συνόδων. Το SIP απλά παρέχει τη συνδεσιμότητα. Τι κάνουν οι χρήστες μετά, είναι έξω από το πεδίο του SIP.

Αυτή η διάκριση κάνει το SIP ουσιαστικά συνεργάσιμο. Μπορεί τώρα να χρησιμοποιηθεί μαζί με SDP, παραδείγματος χάριν, για να καθιερώσει συνόδους *Voice over IP* (VoIP), και θα είναι σύντομα δυνατό να συνδυαστεί με νέα

πρωτόκολλα περιγραφών συνόδων για να καθιερώσει τύπους συνόδων που δεν υπάρχουν ακόμα. Η έννοια που περιγράφουμε δεν είναι ανόμοια με αυτή που χαρακτηρίζει το στρώμα (layer) IP. Είδαμε ότι η πολυτιμότερη υπηρεσία που παρέχεται από το Διαδίκτυο είναι η συνδεσιμότητα IP. Όλα τα υπόλοιπα εφαρμόζονται παίρνοντας τη συνδεσιμότητα IP σαν βάση. Πάλι όμως, το πώς η συνδεσιμότητα IP χρησιμοποιείται, είναι έξω από το πεδίο του ίδιου του IP. Ως εκ τούτου, η δημιουργία υπηρεσιών IP είναι ενιαία, γρήγορη, και αποδοτική.

Το SIP ούτε καν υποθέτει ότι η σύνοδος που έχει καθιερώσει θα πραγματοποιηθεί στο Διαδίκτυο. Παραδείγματος χάριν, εάν ο Bob θέλει να προσκαλέσει τη Laura για να συμμετάσχει σε μια κλήση διασκέψεων που πραγματοποιείται στο *δημόσιο-μεταστρεφόμενο τηλεφωνικό δίκτυο* (PSTN), αυτό που έχει να κάνει είναι να χρησιμοποιήσει το SIP για να παραδώσει τον αριθμό τηλεφώνου που πρέπει να καλέσει για να το κάνει. Σε αυτό το παράδειγμα, η περιγραφή της συνόδου θα περιείχε έναν αριθμό τηλεφώνου αντί των διευθύνσεων IP και των θυρών UDP. Όταν το SIP παραδίδει τη σύνοδο περιγραφή στη Laura, αντιδρά όπως κάνει σε οποιοδήποτε είδος ping με το εργαλείο που της έχει δοθεί. Το SIP παρέχει όσες πληροφορίες είναι απαραίτητες για τον καλούμενο για να δεχτεί την πρόσκληση. Είναι μια υποδειγματική προδιαγραφή IETF δεδομένου ότι εκτελεί το στόχο του. Όταν πρέπει να περιγράψουμε μια σύνοδο που καθιερώνεται από το SIP, πρέπει να χρησιμοποιήσουμε ένα άλλο πρωτόκολλο σχεδιασμένο για αυτό τον σκοπό εκτός από το SIP.

1.7.3 Νοημοσύνη στα End Συστήματα: End-to-End Protocol

Η κοινότητα IETF θεωρεί ότι τα end-to-end πρωτόκολλα είναι καλύτερα για την παροχή end-to-end υπηρεσιών και ότι το IP είναι ένα τέτοιο πρωτόκολλο. Το IP παρέχει συνδεσιμότητα μεταξύ των ακραίων (end) σημείων που χωρίζονται από ένα δίκτυο όπου παρεμβαίνουν δρομολογητές (routers). Οι δρομολογητές εκτελούν τον καθορισμένο με σαφήνεια στόχο της δρομολόγησης δεδομένων όσο το δυνατόν αποτελεσματικότερα. Ομοίως, το SIP παρέχει συνδεσιμότητα μεταξύ χρηστών με SIP

servers. Οι SIP servers έχουν έναν εξίσου καθορισμένο στόχο: δρομολόγηση αιτημάτων και απαντήσεων SIP. Δεν επεξεργάζονται τις περιγραφές συνόδου που φέρονται στα SIP bodies επειδή δεν χρειάζεται προκειμένου να καθοδηγήσουν τα μηνύματα SIP. Αυτό κάνει το SIP ένα αποδοτικό πρωτόκολλο, μαζί με το γεγονός ότι όλη η νοημοσύνη σε ένα δίκτυο SIP βρίσκεται στα άκρα του συστήματος δηλαδή τα UA. Οι κεντρικοί υπολογιστές SIP μπορούν να είναι ουσιαστικά απαθής (stateless) και να "ξεχνούν" τα πάντα για τις συναλλαγές που γίνονται επειδή οι πληροφορίες για να καθοδηγηθεί ένα μήνυμα SIP περιλαμβάνονται στο ίδιο το μήνυμα.

1.7.4 Διαλειτουργικότητα

Το SIP είναι σχεδιασμένο έτσι ώστε οποιαδήποτε εφαρμογή του πρωτοκόλλου να μπορεί να επικοινωνήσει με οποιαδήποτε άλλη εφαρμογή και ενσωματώνει μεθόδους για να βρίσκει επεκτάσεις που θα χρησιμοποιηθούν σε μια σύνοδο. Δύο ιδιαίτερα προηγμένα SIP UA που καθιερώνουν μια σύνοδο είναι πιθανό να χρησιμοποιούν πολλές επεκτάσεις και περίπλοκα χαρακτηριστικά γνωρίσματα. Ακόμα και εάν ένα από αυτά τα προηγμένα UA πρέπει να καθιερώσει μια σύνοδο με ένα στοιχειώδες UA, μπορεί πάντα να το κάνει. Όλες οι επεκτάσεις SIP είναι σχεδιασμένες για να είναι μορφωματικές (modular) έτσι ώστε η χρήση τους να μπορεί να διαπραγματευτεί χωριστά. Μπορούμε να επιλέξουμε ένα συγκεκριμένο σύνολο επεκτάσεων για την πρώτη σύνοδο και ένα απολύτως διαφορετικό σύνολο για την επόμενη.

Η διαπραγμάτευση εγγυάται την πραγματική διαλειτουργικότητα μεταξύ όλων των χρηστών SIP μέσα στο δίκτυο. Αυτό είναι κάτι νέο σε πολλές εφαρμογές φωνής όπου τα πρωτόκολλα (ISDN User Part [ISUP], για παράδειγμα) μπορεί να έχουν πολλές ασυμβατότητες, κάνοντας απαραίτητη τη χρήση gateways μεταξύ των δικτύων που αποτυγχάνουν να "μιλήσουν" στο ίδιο ISUP. Οποιαδήποτε εφαρμογή των gateways για τη μετάφραση πρωτοκόλλων είναι ανεπιθύμητη επειδή σπάζει το μοντέλο end-to-end και επειδή μερικά χαρακτηριστικά γνωρίσματα ενός πρωτοκόλλου μπορεί να χαθούν κατά τη μετάφραση. Αντίθετα, το SIP είναι πραγματικά ένα σφαιρικό πρωτόκολλο.

1.7.5 Ευελιξία

Το SIP προωθεί τη νοημοσύνη στα end συστήματα και προλαμβάνει την ανάγκη να αποθηκευθούν πληροφορίες κατάστασης (state) μέσα στο δίκτυο κατά τη διάρκεια μιας συνόδου. Μόλις ένας χρήστης εντοπισθεί στην διάρκεια της καθιέρωσης συνόδου, η end-to-end επικοινωνία είναι δυνατή μεταξύ των end συστημάτων χωρίς τη βοήθεια του κεντρικού υπολογιστή. Οι κεντρικοί υπολογιστές που δεν είναι αναγκαίο να ελέγχουν τη σηματοδότηση κατά τη διάρκεια της συνόδου μπορούν να χειρίζονται μεγαλύτερο αριθμό συνόδων.

Για αυτό το λόγο, τα δίκτυα SIP είναι ιδιαίτερα ευέλικτα μιας και μεταφέρουν την stateless λειτουργία στα σημεία του δικτύου όπου είναι πιο φορτωμένα (stressed).

ΚΕΦΑΛΑΙΟ Β’:

Αναλυτική λειτουργιά του πρωτοκόλλου SIP

Παρακάτω περιγράφουμε το SIP αναλυτικότερα. Αναφέρουμε τα μηνύματα που ανταλλάσσονται και τη μορφή τους. Παράλληλα, παραθέτουμε και παραδείγματα τα οποία μας δείχνουν στη πράξη το πώς λειτουργεί.

2.1 Συναλλαγές Client/Server

Το SIP είναι βασισμένο στο *Hypertext Transfer Protocol* (HTTP) και όπως το HTTP, έτσι και το SIP είναι ένα πρωτόκολλο αιτήματος/απάντησης. Για να γίνει αντιληπτός ο μηχανισμός αιτήματος/απάντησης που χρησιμοποιείται στο SIP, θα πρέπει να εξετάσουμε τους ορισμούς του πελάτη (client) και του κεντρικού υπολογιστή (server).

Ένας *client* είναι μια οντότητα SIP που παράγει αιτήματα. Ένας server είναι μια οντότητα SIP που λαμβάνει τα αιτήματα και επιστρέφει τις απαντήσεις. Αυτή η ορολογία κληρονομείται από το HTTP, όπου ένας Web Browser περιέχει έναν πελάτη HTTP. Όταν πληκτρολογούμε μια διεύθυνση στον internet explorer, όπως http://www.unipi.gr, στέλνουμε ένα αίτημα σε έναν συγκεκριμένο κεντρικό υπολογιστή δικτύου. Ο κεντρικός υπολογιστής δικτύου στέλνει μια απάντηση με τις πληροφορίες της συγκεκριμένης ιστοσελίδας.

Το SIP ακολουθεί τις ίδιες διαδικασίες. Μέσα από την ίδια ορολογία, όταν δύο UA ανταλλάσσουν μηνύματα SIP, ο UA που αποστέλλει τα αιτήματα είναι ο *User Agent Client* (UAC) και ο UA που επιστρέφει απαντήσεις είναι ο *User Agent Server* (UAS). Ένα αίτημα SIP, μαζί με τις απαντήσεις που προκαλεί, αναφέρεται ως SIP *συναλλαγή (transaction)* .

2.2 Απαντήσεις SIP

Με την αποδοχή ενός αιτήματος, ένας κεντρικός υπολογιστής εκδίδει μια ή περισσότερες απαντήσεις. Κάθε απάντηση έχει έναν κώδικα που δείχνει τη κατάσταση της συναλλαγής. Οι κώδικες κατάστασης (status codes) είναι ακέραιοι που κυμαίνονται από το 100 έως το 699 και ομαδοποιούνται σε κατηγορίες, όπως παρακάτω:

Range	Response Class
100–199	Informational
200–299	Success
300–399	Redirection
400–499	Client error
500–599	Server error
600–699	Global failure

Πίνακας 2-1: Κλάσεις απαντήσεων SIP

Μια απάντηση με έναν κώδικα κατάστασης από 100 έως 199 θεωρείται προσωρινή. Οι απαντήσεις από 200 έως 699 είναι τελικές απαντήσεις. Μια συναλλαγή SIP μεταξύ ενός πελάτη και ενός κεντρικού υπολογιστή περιλαμβάνει ένα αίτημα από τον πελάτη, μια ή περισσότερες προσωρινές απαντήσεις, και μια τελική απάντηση. Μαζί με τον κώδικα κατάστασης, οι απαντήσεις SIP φέρουν και μια φράση.

Η τελευταία περιέχει κατανοήσιμες από τον άνθρωπο πληροφορίες για τον κώδικα κατάστασης. Για παράδειγμα, ένας κώδικας 180 σημαίνει ότι ο χρήστης που προσκαλείται σε μια σύνοδο ειδοποιείται. Επομένως, η φράση ίσως να περιέχει τη λέξη "Ringing". Η φράση μπορεί, φυσικά, να γραφεί σε όποια γλώσσα θέλουμε επειδή θα διαβαστεί από άνθρωπο. Συνεπώς, η επεξεργασία του υπολογιστή αγνοεί τη φράση. Βρίσκει ικανοποιητικές πληροφορίες μέσα στο κώδικα απάντησης.

Εντούτοις, μπορεί να επιδείξει τη φράση στο χρήστη, ο οποίος θα το βρει βεβαίως πιο χρήσιμο να ξέρει ότι το SIP τηλέφωνο είναι σε κατάσταση "Ringing" από το να ξέρει ότι η απάντηση 180 έχει παραληφθεί. Ο πίνακας 2-2 περιέχει όλους τους κώδικες κατάστασης μαζί με τις σχετικές τους φράσεις.

100	Trying	413	Request entity too large
180	Ringing	414	Request-URI too large
181	Call is being forwarded	415	Unsupported media type
182	Queued	420	Bad extension
183	Session progress	480	Temporarily not available
200	OK	481	Call leg/transaction does not exist
202	Accepted	482	Loop detected
300	Multiple choices	483	Too many hops
301	Moved permanently	484	Address incomplete
302	Moved temporarily	485	Ambiguous
305	Use proxy	486	Busy here
380	Alternative service	487	Request cancelled
400	Bad request	488	Not acceptable here
401	Unauthorized	500	Internal servere error
402	Payment required	501	Not implemented
403	Forbidden	502	Bad gateway
404	Not found	503	Service unavailable
405	Method not allowed	504	Gateway time-out
406	Not acceptable	505	SIP version not supported
407	Proxy authentication required	600	Busy everywhere
408	Request time-out	603	Decline
409	Conflict	604	Does not exist anywhere
410	Gone	606	Not acceptable
411	Length required		

Πίνακας 2-2: Κωδικοί απαντήσεων SIP

2.3 Αιτήματα SIP

Η προδιαγραφή SIP καθορίζει έξι τύπους αιτημάτων, κάθε ένας από τους οποίους έχει διαφορετικό σκοπό. Κάθε αίτημα SIP περιέχει έναν τμήμα, αποκαλούμενο method, το οποίο δείχνει το σκοπό του. Ο παρακάτω κατάλογος παρουσιάζει τις έξι μεθόδους.

- INVITE
- ACK
- OPTIONS
- BYE
- CANCEL
- REGISTER

Τα αιτήματα και οι απαντήσεις μπορούν να περιέχουν SIP body. Το body ενός μηνύματος είναι το ωφέλιμο φορτίο του. Το SIP body αποτελείται συνήθως από μια περιγραφή συνόδου.

INVITE

Τα αιτήματα INVITE προσκαλούν τους χρήστες να συμμετέχουν σε μια σύνοδο. Το body του INVITE περιέχει την περιγραφή της συνόδου. Για παράδειγμα, όταν ο Bob καλεί τη Laura, το UA του, στέλνει ένα INVITE με μια περιγραφή συνόδου στο UA της Laura. Ας υποθέσουμε ότι το UA του Bob χρησιμοποιεί *το Session Description Protocol (SDP)* για να περιγράψει τη σύνοδο. Το UA της Laura, λαμβάνει το INVITE με την ακόλουθη περιγραφή συνόδου:

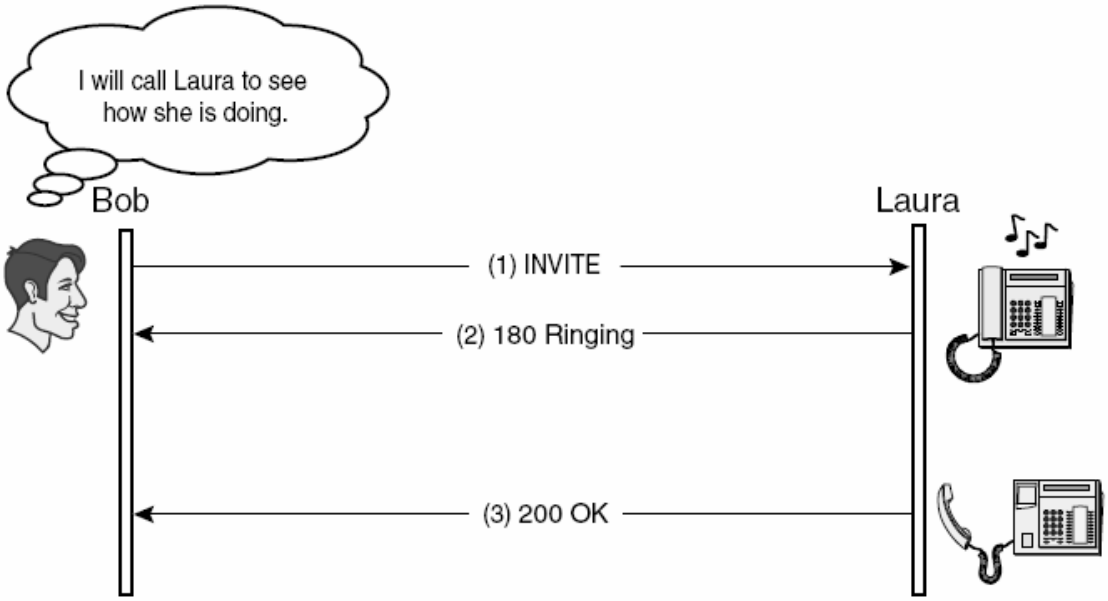
v=0
o=Bob 2890844526 2890842807 IN IP4 131.160.1.112
s=I want to know how you are doing
c=IN IP4 131.160.1.112
t=0 0
m=audio 49170 RTP/AVP 0

Το INVITE που ελήφθηκε από το AU της Laura σημαίνει ότι ο Bob προσκαλεί τη Laura να συμμετάσχει σε μια ακουστική σύνοδο. Από την περιγραφή συνόδου που φέρεται στο INVITE, το UA της Laura ξέρει ότι ο Bob θέλει να λάβει *Real-Time Transport Protocol* (RTP) πακέτα που περιέχουν τη φωνή της Laura στο 131.160.1.112 User Datagram Protocol (UDP) αριθμός θύρας 49170. Το UA της επίσης ξέρει ότι ο Bob μπορεί να λάβει κωδικοποιημένη φωνή *σε Pulse Code Modulation* (PCM). Το UA της Laura αρχίζει να τη ειδοποιεί και επιστρέφει μια απάντηση "180 Ringing"στο UA του Bob. Όταν η Laura δέχεται τελικά την κλήση, το UA της, θα επιστρέψει "200 OK" με μια περιγραφή συνόδου.

v=0
o=Laura 2891234526 2812342807 IN IP4 138.85.27.10
s=I want to know how you are doing
c=IN IP4 138.85.27.10
t=0 0
m=audio 20000 RTP/AVP 0

Σε αυτό το σημείο, η Laura δέχεται την κλήση και ενημερώνει το Bob ότι θα λάβει πακέτα RTP στο 138.85.27.10 UDP θύρα 20000 (σχήμα 2-1). Εάν, όταν είναι η Laura και ο Bob είναι στη μέση της συνόδου, ένας από τους δύο επιθυμεί να τροποποιήσει τη σύνοδο, πρέπει να εκδοθεί έναν νέο INVITE. Αυτός ο τύπος INVITE, αποκαλούμενος re-INVITE, φέρει μια ενημερωμένη περιγραφή συνόδου. Μπορεί να αποτελείται από νέες παραμέτρους όπως αριθμούς θυρών για τα ήδη υπάρχοντα media, ή να προσθέτει νέα media streams. Παραδείγματος χάριν, ο Bob και η Laura μπορούν να προσθέσουν ένα οπτικό stream στη συνομιλία φωνής τους μέσω ενός re-INVITE.

Ουσιαστικά, το SIP χειρίζεται μόνο την πρόσκληση στο χρήστη και την αποδοχή της πρόσκλησης. Όλες οι λεπτομέρειες της συνόδου αντιμετωπίζονται από το πρωτόκολλο περιγραφής συνόδου που χρησιμοποιείται (το SDP σε αυτήν την περίπτωση). Κατά συνέπεια, με μια διαφορετική περιγραφή συνόδου, το SIP μπορεί να προσκαλέσει χρήστες σε οποιοδήποτε τύπο συνόδου.



Σχήμα 2-1: Αίτημα INVITE

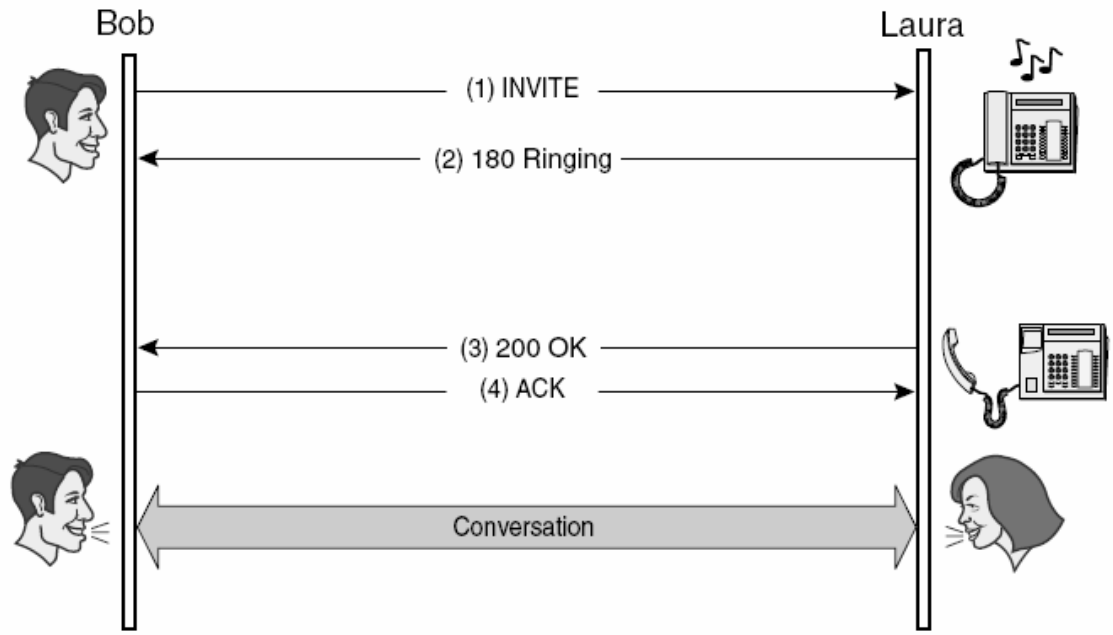
ACK

Τα αιτήματα ACK χρησιμοποιούνται για να αναγνωρίσουν την υποδοχή μιας τελικής απάντησης σε ένα INVITE. Κατά συνέπεια, ένας πελάτης που έχει δημιουργήσει ένα INVITE, εκδίδει ένα αίτημα ACK όταν λαμβάνει μια τελική απάντηση για το INVITE, παρέχοντας μια "τριπλή χειραψία" (three-way handshake): INVITE - τελική απάντηση - ACK (σχήμα 2-2)

Γιατί το SIP χρησιμοποιεί τριπλή χειραψία;

ΤΟ INVITE είναι η μόνη μέθοδος που χρησιμοποιεί τριπλή χειραψία σε αντιδιαστολή με μια διπλής κατεύθυνσης χειραψία (method - τελική απάντηση) [6]. Ορισμένα χαρακτηριστικά κάνουν τη μέθοδο INVITE διαφορετική από τις άλλες μεθόδους. Όταν ένας client εκδίδει ένα αίτημα άλλο από INVITE, αυτό αναμένει μια γρήγορη

απάντηση από τον server. Εντούτοις, η απάντηση από ένα INVITE αίτημα μπορεί να πάρει αρκετό χρόνο. Όταν ο Bob καλεί τη Laura, ίσως αυτή να πρέπει να πιάσει το SIP τηλέφωνο της από την τσάντα της και να πατήσει διάφορα κουμπιά, έτσι η "200 OK" απάντηση που θα έρθει θα καθυστερήσει λίγο ή πολύ. Αποστέλλοντας ένα ACK από τον client στον server ενημερώνουμε τον τελευταίο ότι ο client είναι ακόμα εκεί και ότι η σύνοδος έχει καθιερωθεί επιτυχώς. Η τριπλή χειραψία επιτρέπει επίσης την εφαρμογή των forking servers. Όταν ένας από αυτούς κάνει forking ένα αίτημα, ο πελάτης που εξέδωσε το αίτημα θα λάβει διάφορες απαντήσεις από τους διαφορετικούς κεντρικούς υπολογιστές. Αποστέλλοντας ένα ACK σε κάθε προορισμό που έχει αποκριθεί είναι κρίσιμο για την εξασφάλιση της ομαλής λειτουργίας του SIP σε αναξιόπιστα πρωτόκολλα όπως το UDP.



Σχήμα 2-2: Τριπλή χειραψία

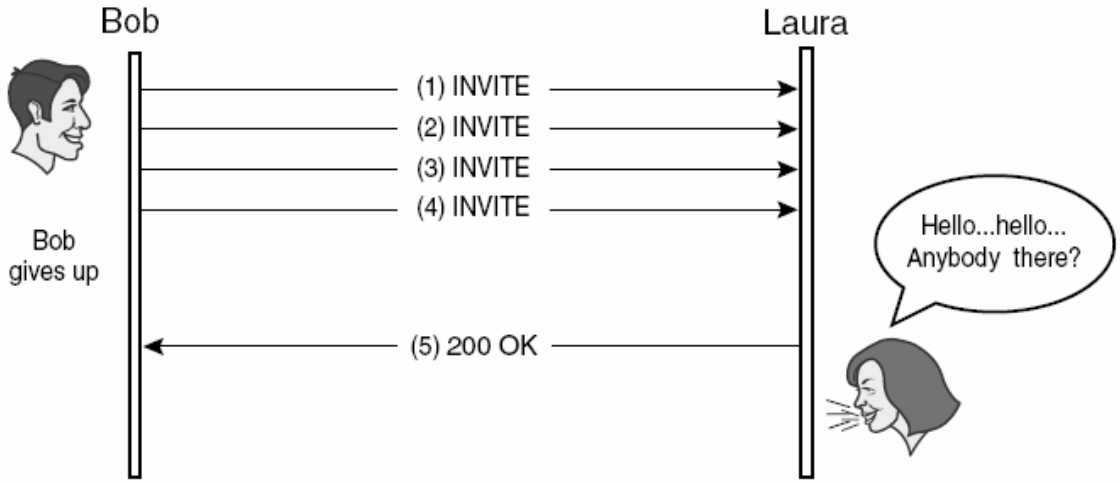
Εκτός από την ταχεία οργάνωση της συνόδου και το forking, η τριπλή χειραψία του INVITE επίσης μας επιτρέπει την αποστολή INVITE χωρίς μια περιγραφή συνόδου, η

οποία αποστέλλεται αργότερα στο ACK. Αυτό το χαρακτηριστικό γνώρισμα είναι χρήσιμο, παραδείγματος χάριν, όταν το SIP αλληλεπιδρά με άλλα πρωτόκολλα σηματοδοσίας που χρησιμοποιούν διαφορετική αλληλουχία μηνυμάτων.

Εντούτοις, το ιστορικό κίνητρο για την κατοχή μιας τριπλής χειραψίας μπορεί να βρεθεί στο παλαιό σχέδιο SIPv1, στο τμήμα για το πώς να παρέχουμε αξιόπιστη παράδοση της πρόσκλησης συνόδου. Το σχέδιο εισήγαγε τη μέθοδο ACK έτσι ώστε να αποφευχθούν μη συγχρονισμένα συμβαλλόμενα μέρη στην καθιέρωση συνόδου, τα οποία μπορεί να εμφανιστούν όταν χρησιμοποιείται μια διπλής κατεύθυνσης χειραψία με ένα αναξιόπιστο πρωτόκολλο μεταφοράς όπως το UDP. Ας εξετάσουμε την ακόλουθη περίπτωση όπου εφαρμόζεται διπλής κατεύθυνσης χειραψία.

Ο Bob στέλνει INVITE στη Laura και θα το αναμεταδίδει έως ότου λάβει μια τελική απάντηση από τη Laura. Έως ότου παραληφτεί αυτή η τελική απάντηση, ο Bob δεν μπορεί να ξέρει εάν η Laura έλαβε το INVITE ή χάθηκε στο δίκτυο. Ο Bob περιμένει για λίγο και επειδή δεν παίρνει καμία απάντηση, κλείνει και σταματά το INVITE. Ο Bob θεωρεί ότι καμία σύνοδος δεν έχει καθιερωθεί.

Κατά προσέγγιση στον ίδιο χρόνο, η Laura δέχεται την κλήση του Bob και στέλνει πίσω "200 OK" απάντηση. Εάν αυτή η απάντηση χαθεί, ο Bob δεν θα την λάβει ποτέ, έτσι ακόμα θεωρεί ότι καμία σύνοδος δεν έχει καθιερωθεί. Επειδή η Laura παρατηρεί ότι ο Bob έχει σταματήσει να κάνει INVITE, υποθέτει ότι έχει λάβει το "200 OK". Επομένως, η Laura σκέπτεται ότι η σύνοδος έχει επιτυχώς καθιερωθεί (Σχήμα 2-3).



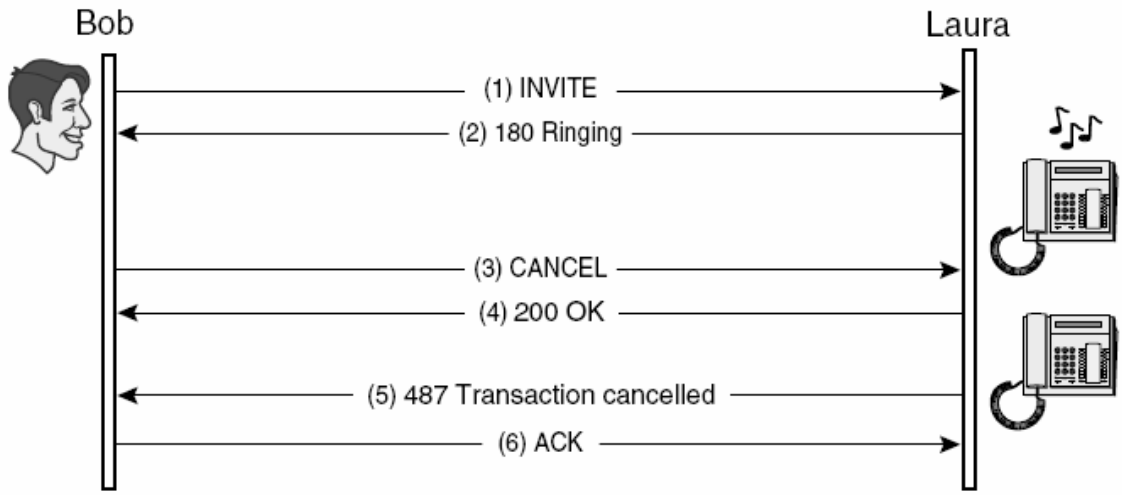
Σχήμα 2-3: Χρησιμότητα του ACK

Εάν μια τριπλή χειραψία ήταν σε ισχύ για αυτό το σενάριο, η Laura δεν θα είχε λάβει ACK για το 200 OK της, δεδομένου ότι ο Bob έχει ήδη αποσυρθεί. Κατά συνέπεια, ορθά θα σκεφτόταν ότι η σύνοδος δεν καθιερώθηκε.

CANCEL Τα αιτήματα CANCEL ακυρώνουν εν αναμονή συναλλαγές. Εάν ένας κεντρικός υπολογιστής SIP έχει λάβει INVITE αλλά δεν έχει ακόμα επιστρέψει μία τελική απάντηση, θα σταματήσει την επεξεργασία του INVITE με την παραλαβή ενός CANCEL. Εάν, εντούτοις, έχει ήδη επιστρέψει μια τελική απάντηση για το INVITE, το CANCEL αίτημα δεν θα έχει καμία επίδραση στη συναλλαγή.

Στο σχήμα 2-4, ο Bob καλεί τη Laura και το SIP τηλέφωνο της αρχίζει να χτυπά, αλλά κανένας δεν το σηκώνει για λίγο. Ο Bob αποφασίζει να κλείσει το τηλέφωνο. Στέλνει ένα αίτημα CANCEL για το προηγούμενό INVITE του. Με την υποδοχή του CANCEL, το SIP τηλέφωνο της Laura σταματά να χτυπά. Ο κεντρικός υπολογιστής στέλνει πίσω ένα 200 OK ως απάντηση για το CANCEL, δείχνοντας ότι η επεξεργασία έγινε επιτυχώς. Είναι σημαντικό να παρατηρηθεί ότι όταν ο κεντρικός υπολογιστής αποκριθεί στο αίτημα CANCEL, αποκρίνεται και στο προηγούμενο INVITE επίσης.

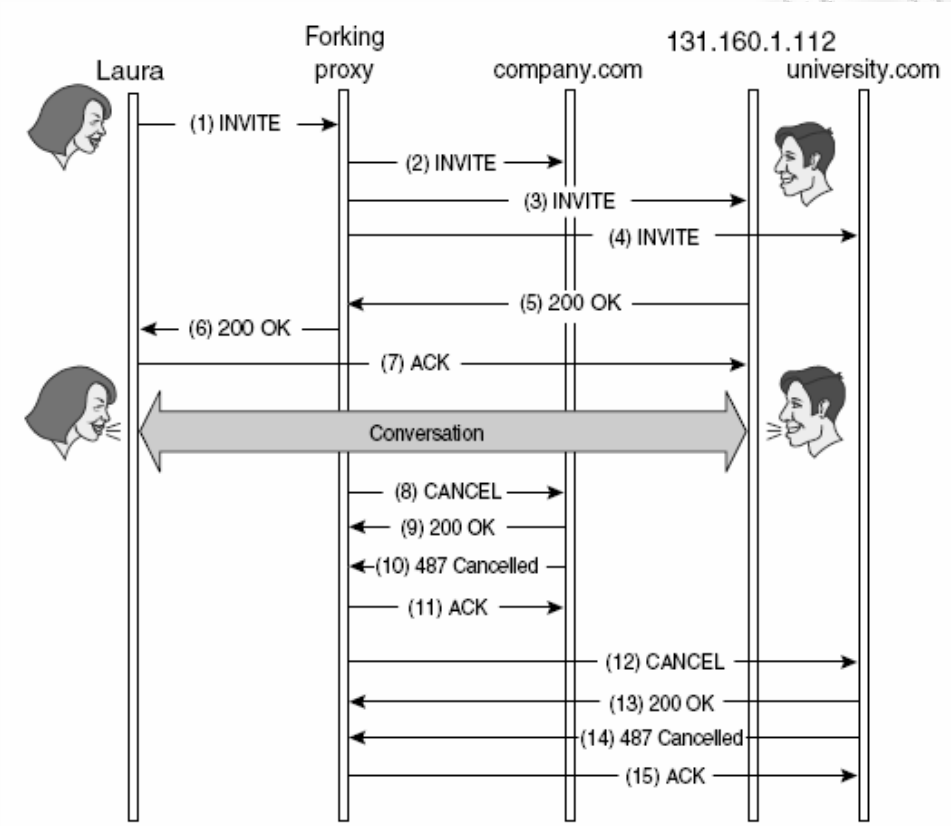
Στέλνει "487 Transaction Cancelled" και ο πελάτης τελειώνει το τριπλής χειραψίας INVITE με την αποστολή ενός ACK (INVITE - 487 Transaction Cancelled - ACK).
Επομένως, το τριπλής χειραψίας INVITE εκτελείται πάντα, ακόμα και όταν η συναλλαγή ακυρώνεται. Τα αιτήματα CANCEL είναι χρήσιμα όταν forking proxy βρίσκονται ενδιάμεσα στην πορεία. Όταν ένας τέτοιος proxy εκτελεί μια παράλληλη αναζήτηση, δοκιμάζει διάφορες θέσεις ταυτόχρονα.



Σχήμα 2-4: Ακύρωση του INVITE

Παραδείγματος χάριν, ένας forking proxy ξέρει τρεις πιθανές θέσεις όπου μπορεί να βρίσκεται ο Bob: SIP:Bob@131.160.1.112, SIP:Bob.Johnson@company.com και SIP:Bob@university.com. Όταν ο proxy λάβει ένα INVITE από τη Laura στο Bob, θα δοκιμάσει αυτές τις τρεις θέσεις παράλληλα (ταυτόχρονα). Ο proxy στέλνει τρία INVITE, ένα σε κάθε θέση. Ο Bob που εργάζεται αυτή τη στιγμή στο 131.160.1.112, απαντά στην κλήση. Ο server λαμβάνει 200 OK από το SIP:Bob@131.160.1.112 και διαβιβάζει αυτή την απάντηση στο UA της Laura. Επειδή η σύνοδος έχει καθιερωθεί πλέον μεταξύ της Laura και του Bob, ο server θέλει να σταματήσει τις άλλες αναζητήσεις, έτσι στέλνει δύο CANCEL, ένα σε κάθε θέση (σχήμα 2-5).
Να υπογραμμίσουμε ότι ένα CANCEL αίτημα δεν έχει επιπτώσεις σε μια συναλλαγή αφότου μια τελική απάντηση έχει σταλεί. Επομένως, στο παράδειγμά μας, ακόμα κι αν ο forking server στέλνει CANCEL στο SIP:Bob@131.160.1.112, η σύνοδος μεταξύ

του Bob και της Laura, εμμένει. Το CANCEL δεν μπορεί να τερματίσει μια τρέχουσα συναλλαγή. Επίσης, αγνοείται από τις ολοκληρωμένες συναλλαγές.



Σχήμα 2-5: Παράδειγμα με forking proxy

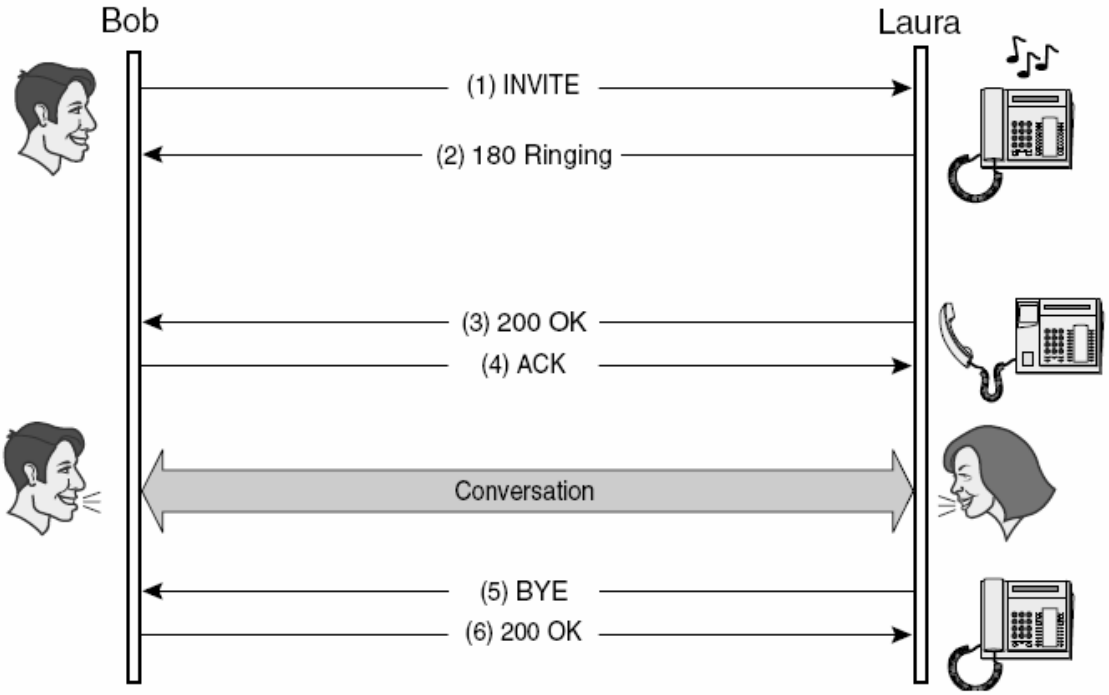
BYE

Τα αιτήματα BYE χρησιμοποιούνται για να εγκαταλείψουμε συνόδους. Στις two-party συνόδους, η εγκατάλειψη από ένα από τα συμβαλλόμενα μέρη υπονοεί ότι η σύνοδος ολοκληρώνεται. Παραδείγματος χάριν, όταν ο Bob στέλνει BYE στη Laura, η σύνοδός τους θεωρείται αυτόματα ολοκληρωμένη (σχήμα 2-6). Στα πολλαπλής διανομής σενάρια, εντούτοις, το αίτημα BYE από έναν από τους συμμετέχοντες απλά σημαίνει ότι ο συγκεκριμένος αποχωρεί από τη διάσκεψη. Η ίδια η σύνοδος δεν επηρεάζεται. Στην πραγματικότητα, είναι κοινή πρακτική στις μεγάλες

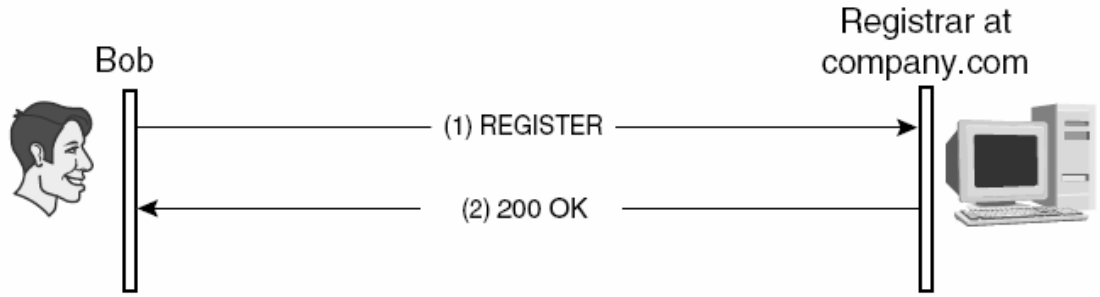
πολλαπλής διανομής συνόδους να μην αποστέλλεται BYE κατά τη αποχώρηση από τη σύνοδο.

REGISTER

Οι χρήστες στέλνουν αιτήματα REGISTER για να ενημερώσουν έναν κεντρικό υπολογιστή (σε αυτή τη περίπτωση, καλείται ως registrar) για την τρέχουσα θέση τους. Ο Bob μπορεί να στείλει ένα REGISTER στον registrar του company.com λέγοντας ότι όλα τα εισερχόμενα αιτήματα για SIP:Bob.Johnson@company.com πρέπει να κατευθύνονται (proxied), στο SIP:Bob@131.160.1.112 (σχήμα 2-7).
Οι κεντρικοί υπολογιστές SIP συνδυάζονται συνήθως με τους SIP registrars. Ένας SIP registrar μπορεί να στείλει όλες τις πληροφορίες που παραλαμβάνονται από διάφορα αιτήματα REGISTER σε έναν ενιαίο κεντρικό υπολογιστή θέσης (location server), μέσω του οποίου γίνεται δυνατό σε οποιοδήποτε κεντρικό υπολογιστή SIP να βρει έναν συγκεκριμένο χρήστη.



Σχήμα 2-6: Αίτημα BYE



Σχήμα 2-7: Αίτημα REGISTER

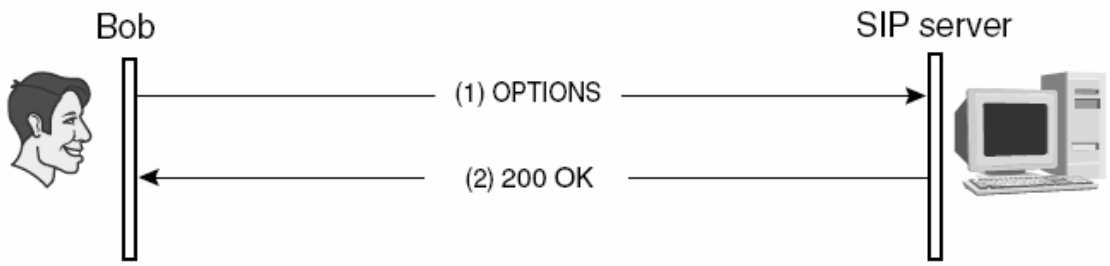
Τα μηνύματα REGISTER περιέχουν επίσης τους χρόνους όταν αναφέρεται η εγγραφή. Παραδείγματος χάριν, ο Bob μπορεί να καταχωρήσει την παρούσα θέση του μέχρι τις τέσσερις η ώρα το απόγευμα επειδή ξέρει ότι τότε θα αφήσει το γραφείο. Ένας χρήστης μπορεί επίσης να καταχωρηθεί σε διάφορες θέσεις

συγχρόνως, λέγοντας ότι ο κεντρικός υπολογιστής πρέπει να τον ψάξει σε όλες τις καταχωρημένες θέσεις μέχρι να τον βρει.

OPTIONS

Τα αιτήματα OPTIONS ρωτούν έναν κεντρικό υπολογιστή για τις ικανότητές του (σχήμα 2-8), συμπεριλαμβανομένων των μεθόδων και των πρωτοκόλλων περιγραφής συνόδου που υποστηρίζει. Ένας κεντρικός υπολογιστής SIP μπορεί να απαντήσει σε ένα αίτημα OPTIONS ότι υποστηρίζει SDP ως πρωτόκολλο περιγραφής συνόδου και πέντε μεθόδους: INVITE, ACK, CANCEL, BYE και OPTIONS. Επειδή ο κεντρικός υπολογιστής δεν υποστηρίζει τη μέθοδο REGISTER, μπορούμε να συμπεράνουμε ότι δεν είναι registrar.

Η μέθοδος OPTIONS ίσως να μην φαίνεται χρήσιμη τώρα, αλλά καθώς νέες επεκτάσεις προσθέτουν νέες μεθόδους στο SIP, η μέθοδος OPTIONS είναι ένας σπουδαίος τρόπος να ανακαλύπτουμε ποιες μεθόδους υποστηρίζει κάθε κεντρικός υπολογιστής. Μια μέθοδος OPTIONS επιστρέφει επίσης δεδομένα που διευκρινίζουν ποιες κωδικοποιήσεις για τα body των μηνυμάτων ο κεντρικός υπολογιστής καταλαβαίνει. Εάν ένας κεντρικός υπολογιστής καταλαβαίνει, παραδείγματος χάριν, ένα ορισμένο κώδικα συμπίεσης, ο πελάτης θα είναι σε θέση να στείλει τις περιγραφές συνόδου συμπιεσμένες, εκμεταλλευόμενος την ευκαιρία να εξοικονομήσει ένα μέρος του εύρους ζώνης.



Σχήμα 2-8: Αίτημα OPTIONS

2.4 Τύποι Proxy Server

Οι proxy server μπορούν να ταξινομηθούν σύμφωνα με το πόσες πληροφορίες θέσης αποθηκεύουν κατά τη διάρκεια μιας συνόδου. Το SIP καθορίζει τρεις τύπους κεντρικών υπολογιστών proxy: call stateful, stateful και stateless.

2.4.1 Call Stateful Proxy

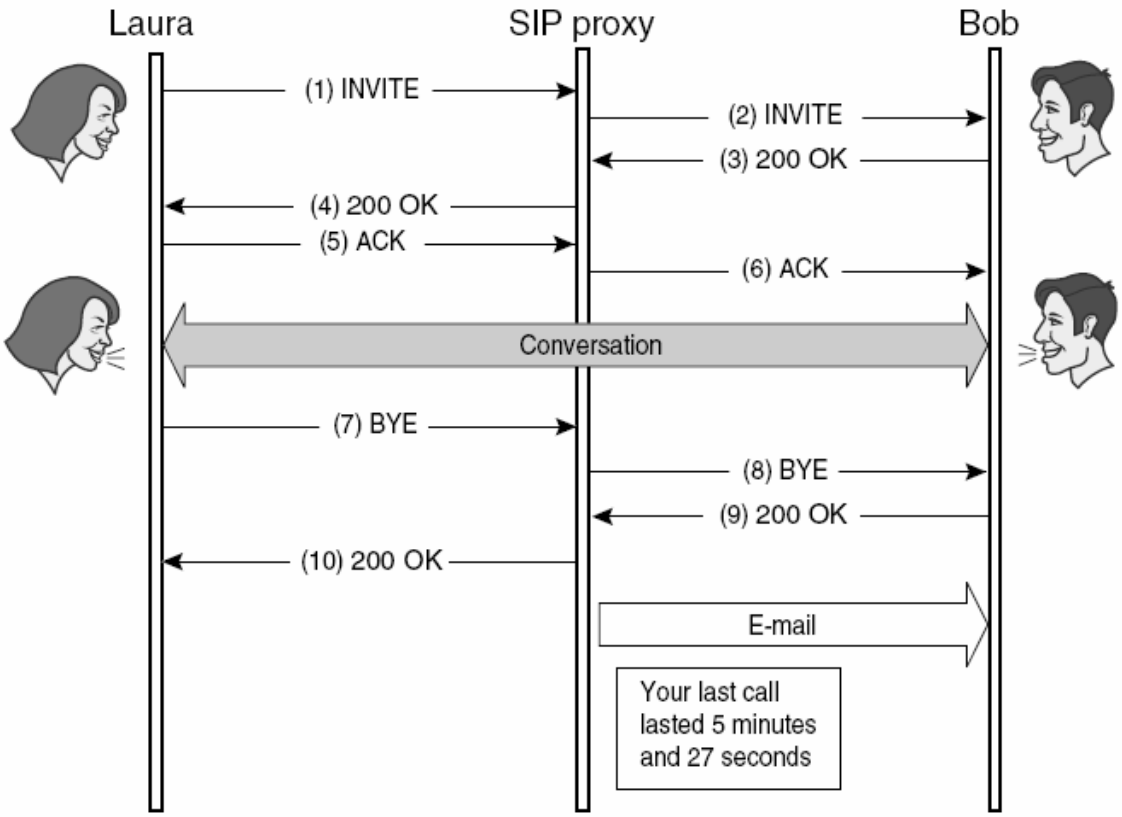
Οι call stateful proxy πρέπει να ενημερώνονται για όλες τις SIP συναλλαγές που συμβαίνουν κατά τη διάρκεια της συνόδου και επομένως, είναι πάντα στην πορεία που ακολουθούν τα μηνύματα SIP ταξιδεύοντας μεταξύ των τελικών χρηστών. Αυτοί οι proxy αποθηκεύουν πληροφορίες από τη στιγμή που η σύνοδος καθιερώνεται μέχρι τη στιγμή που τελειώνει.

Ένα παράδειγμα ενός τέτοιου proxy είναι ένας κεντρικός υπολογιστής που εφαρμόζει μια υπηρεσία κλήσης, όπως η λήψη ενός μηνύματος ηλεκτρονικού ταχυδρομείου στο τελείωμα κάθε κλήσης με πληροφορίες για τη διάρκεια αυτής (σχήμα 2-9). Για να υπολογίσει το μήκος της κλήσης, ο proxy πρέπει να είναι στην πορεία του INVITE που αρχίζει την κλήση και επίσης στην πορεία του BYE που τερματίζει τη κλήση.

2.4.2 Stateful Proxy

Οι stateful proxy καλούνται μερικές φορές transaction stateful proxy επειδή η συναλλαγή είναι η μόνη ανησυχία τους. Ένας stateful proxy αποθηκεύει πληροφορίες κατάστασης (θέσης) για μια δεδομένη συναλλαγή μέχρι αυτή να ολοκληρωθεί. Δεν χρειάζεται να είναι μέσα στη πορεία των μηνυμάτων SIP για τις επόμενες συναλλαγές. Οι forking proxy είναι καλό παράδειγμα των stateful proxy (σχήμα 2-10) Στέλνουν INVITE σε πολλές διαφορετικές θέσεις και πρέπει να αποθηκεύουν τις θέσεις των INVITE συναλλαγών προκειμένου να γνωρίζουν εάν όλες οι θέσεις που δοκιμάζονται έχουν επιστρέψει μια τελική απάντηση ή όχι.

Εντούτοις, μόλις ο χρήστης βρεθεί σε μια συγκεκριμένη θέση, ο proxy δεν χρειάζεται να παραμείνει στο path της σηματοδοσίας περεταίρω.



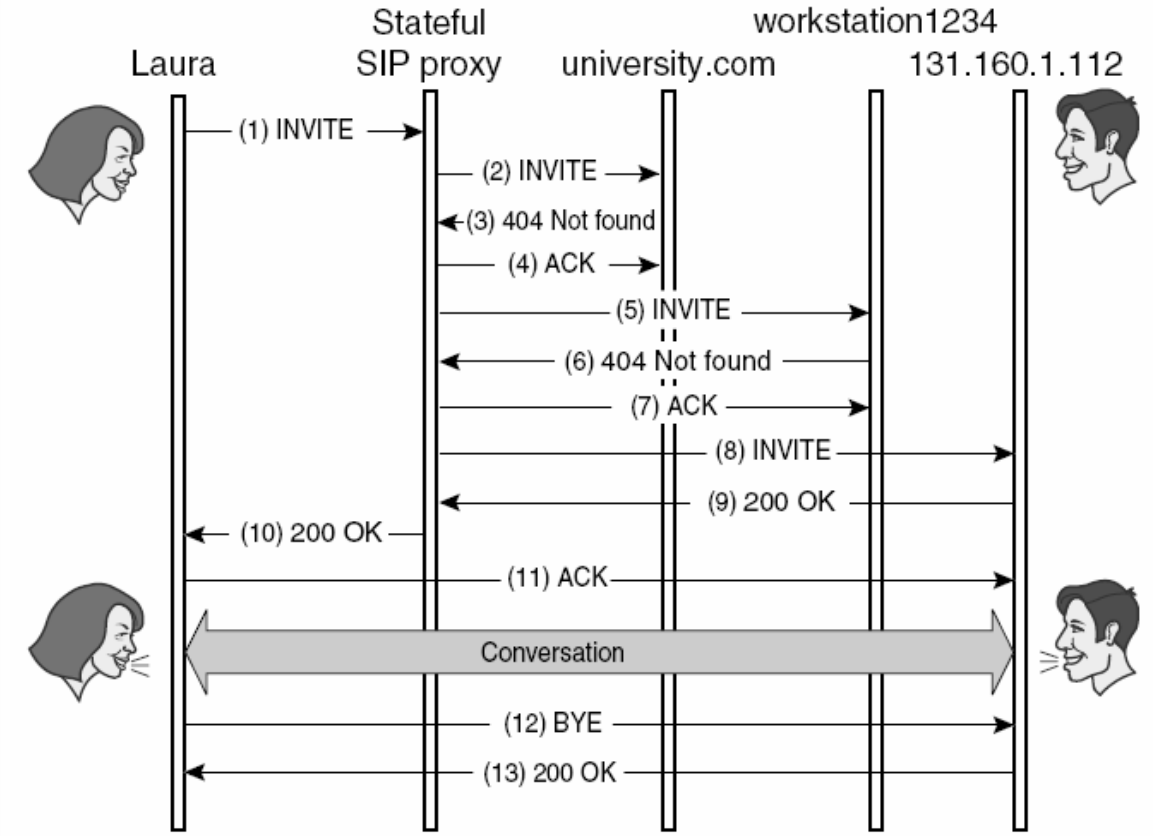
Σχήμα 2-9: Stateful proxy

Δημιουργία των ACK

Το σχήμα 2-10 επιδεικνύει πώς τα ACK παράγονται στο SIP. Είδαμε ότι τα ACK παράγονται ως τελικές απαντήσεις στα INVITE. Είναι μέρος της τριπλής χειραψίας και, ανάλογα με τον τύπο απάντησης που ο κεντρικός υπολογιστής επιστρέφει, αυτοί παράγονται είτε από τους proxy είτε από τα UA. Οι κεντρικοί υπολογιστές proxy μπορούν να δώσουν ACK μόνο σε μη επιτυχείς απαντήσεις, οι οποίες έχουν ένα κώδικα θέσης που είναι μεγαλύτερος από 299. Απαντήσεις επιτυχίας (κώδικας θέσης μεταξύ 200 και 299) παίρνουν πάντα ACK από UA που έχουν αρχίσει ένα INVITE.

Στο σχήμα 2-10, ο proxy server δίνει ACK σε μη επιτυχείς απαντήσεις στα μηνύματα (4) και (7). Εντούτοις, το UA δίνει ACK στο 200 OK στο μήνυμα(11). Αυτό επιτρέπει

σε έναν proxy να δοκιμάσει πολλαπλές θέσεις χωρίς να πρέπει να ενημερώσει το UA για τις ανεπιτυχείς προσπάθειες να βρεθεί ο τελικός χρήστης. Μόλις αποκριθεί θετικά ο χρήστης στο INVITE, το UA πρέπει να λάβει τη περιγραφή συνόδου προκειμένου να καθιερωθεί η σύνοδος.



Σχήμα 2-10: Stateful forking proxy

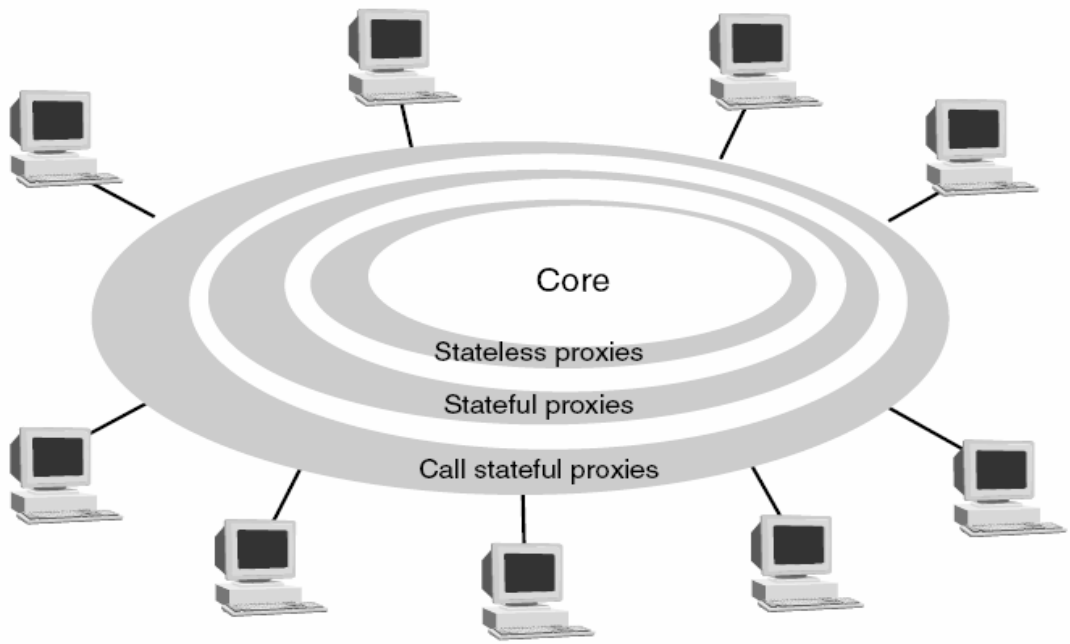
2.4.3 Stateless Proxy

Οι stateless proxy δεν κρατούν θέσεις. Λαμβάνουν ένα αίτημα, το διαβιβάζουν στο επόμενο hop και διαγράφουν αμέσως όλες τις πληροφορίες σχετικές με εκείνο το αίτημα Όταν ένας stateless proxy λαμβάνει μια απάντηση, καθορίζει τη δρομολόγηση βάση του header και δεν διατηρεί τη θέση της.

2.4.4 Διανομή των Proxy

Μια ανάλυση της κυκλοφορίας (traffic) IP σε ένα δίκτυο δείχνει ότι ο πυρήνας (core) είναι περισσότερο φορτωμένος από τις άκρες (edges). Αυτό ισχύει για την κυκλοφορία SIP επίσης. Οι κεντρικοί υπολογιστές SIP στον πυρήνα πρέπει να είναι σε θέση να χειριστούν πολλά μηνύματα, ενώ οι κεντρικοί υπολογιστές SIP στην περιφέρεια δεν είναι απαραίτητο να υποστηρίζουν εξίσου βαριά φορτία. Το SIP είναι σχεδιασμένο για stateless servers στον πυρήνα. Εκτελούν δρομολογήσεις βασιζόμενοι στα headers όσο γρήγορα και αποτελεσματικά γίνεται. Στις άκρες του δικτύου, οι call stateful και stateful server μπορούν να εφαρμοστούν για να εκτελέσουν τη δρομολόγηση βασισμένη σε πιο περίπλοκες μεταβλητές (όπως ο χρόνος της ημέρας ή η ταυτότητα του αποστολέα), ή μπορούν να κάνουν forking τα αιτήματα και να παρέχουν υπηρεσίες στο χρήστη.

Διανέμοντας τους κεντρικούς υπολογιστές με αυτό το τρόπο κάνει το SIP ένα πολύ εξελικτικό πρωτόκολλο εφαρμόσιμο σε όλο και περισσότερο μεγάλα δίκτυα όπως το Διαδίκτυο. Το SIP κρατά τον πυρήνα γρήγορο και απλό και ωθεί τη νοημοσύνη στη περιφέρεια του δικτύου (σχήμα 2-11).



Σχήμα 2-11: Stateless πυρήνας και νοημοσύνη στα άκρα

2.5 Μορφή των SIP Μηνυμάτων

Η εξέλιξη του πρωτοκόλλου πορεύεται σε διακριτά στάδια. Όταν αποφασιστεί ποιες πληροφορίες θα ανταλλαχθούν μεταξύ των διανεμημένων συστημάτων, το επόμενο βήμα είναι να αποφασιστεί πώς αυτές οι πληροφορίες πρέπει να κωδικοποιηθούν. Αυτή η απόφαση έχει βασικά δύο προσεγγίσεις: δυαδική (binary), η οποία χρησιμοποιεί πεδία bit για να κωδικοποιήσει τις πληροφορίες και κειμένου (text), η οποία χρησιμοποιεί σειρές χαρακτήρων. Το ακόλουθο παράδειγμα εξηγεί τις διαφορές μεταξύ των δύο προσεγγίσεων.

Οι χρήστες θέλουν να παρακολουθούν τα τεκταινόμενα του τρέχοντος μήνα στους υπολογιστές τους, και ο κεντρικός υπολογιστής στο δίκτυο έχει αυτές τις πληροφορίες. Χρειαζόμαστε ένα πρωτόκολλο που μπορεί να μεταφέρει αυτές τις πληροφορίες από τον κεντρικό υπολογιστή στον υπολογιστή γραφείου. Το πεδίο των μηνών μπορεί να πάρει ακριβώς 12 πιθανές τιμές: Ιανουαρίου, Φεβρουάριος, Μάρτιος, Απρίλιος, Μάιος, Ιούνιος, Ιούλιος, Αύγουστος, Σεπτέμβριος, Οκτώβριος, Νοέμβριος, και Δεκέμβριος.

Ένα text-based πρωτόκολλο θα διαβιβάσει το όνομα του μήνα μεταξύ των συστημάτων. Ας υποθέσουμε ότι το περιεχόμενο του μηνύματος είναι Ιανουάριος. Κάθε χαρακτήρας (γράμμα) κωδικοποιείται χρησιμοποιώντας ένα byte (8 bit). Κατά συνέπεια, το μήνυμα Ιανουάριος θα κωδικοποιηθεί χρησιμοποιώντας 80 bit.

Ένα δυαδικό πρωτόκολλο, από την άλλη πλευρά, καθορίζει έναν πίνακα με πιθανές τιμές και τις αντίστοιχες κωδικοποιήσεις, όπως φαίνεται στον πίνακα 2-3. Κατά συνέπεια, για να διαβιβάσει τον τρέχοντα μήνα, το δυαδικό πρωτόκολλο θα έστελνε ένα μήνυμα τεσσάρων bit που περιέχει 0000.

Το SIP χρησιμοποιεί text κωδικοποίηση σε αντιδιαστολή με τη binary. Αυτό το ζήτημα έχει δημιουργήσει έντονες συζητήσεις, ενώ φαίνεται να είναι μια σχεδόν φανατική συζήτηση κατά την οποία είναι αδύνατο να διατηρηθεί μια ουδέτερη άποψη [1]. Οι υπερασπιστές του text υποστηρίζουν ότι τα text-based πρωτόκολλα διορθώνονται ευκολότερα επειδή μπορούν άμεσα να κατανοηθούν από έναν

άνθρωπο και ότι είναι πιο εύκαμπτα και ευκολότερο να επεκταθούν με νέα χαρακτηριστικά γνωρίσματα.

Οι "δυαδικοί" οπαδοί υποστηρίζουν ότι τα δυαδικά πρωτόκολλα χρησιμοποιούν το εύρος ζώνης αποτελεσματικότερα και μπορούν επίσης ευκολότερα να διορθωθούν και να επεκταθούν με τα κατάλληλα εργαλεία. Και οι δύο τύποι κωδικοποιήσεων έχουν τα πλεονεκτήματα και τα μειονεκτήματα που δεν θα απαριθμήσουμε σε αυτήν την ανάλυση, αλλά ας λάβουμε υπόψη ότι το SIP είναι ένα text-based πρωτόκολλο και επομένως διαθέτει όλα τα πλεονεκτήματα και τα μειονεκτήματα των text-based πρωτοκόλλων γενικά.

0000	January	0110	July
0001	February	0111	August
0010	March	1000	September
0011	April	1001	October
0100	May	1010	November
0101	June	1011	December

Πίνακας 2-3: Δυαδική κωδικοποίηση μηνών

2.5.1 Μορφή των Αιτημάτων SIP

Ένα αίτημα SIP αποτελείται από μια γραμμή αιτήματος, διάφορα header, μια κενή γραμμή, και ένα body μηνύματος. Ο πίνακας 2-4 παρουσιάζει τη μορφή ενός αιτήματος SIP. Το body message είναι προαιρετικό μιας και μερικά αιτήματα δεν το φέρουν.

Request-line
Several headers
Empty line
Message body

Πίνακας 2-4: Μορφή ενός αιτήματος SIP

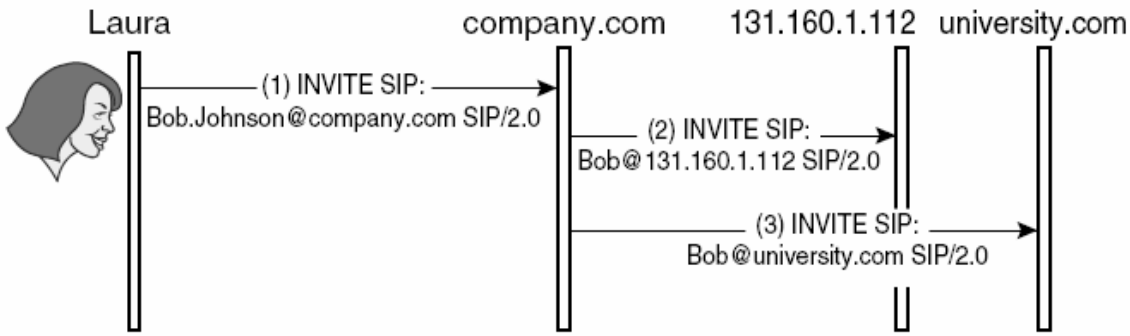
Request Line

Μια γραμμή αιτήματος έχει τρία στοιχεία: μέθοδος, αίτημα-URI, και έκδοση πρωτοκόλλου. Η μέθοδος δείχνει τον τύπο αιτήματος,. Το αίτημα-URI υποδεικνύει το επόμενο hop , εκεί δηλαδή όπου το αίτημα πρέπει να καθοδηγηθεί. Στο σχήμα 2-12, ο SIP proxy στο company.com λαμβάνει ένα INVITE με το αίτημα-URI SIP:Bob.Johnson@company.com. Ο proxy γνωρίζει ότι ο Bob μπορεί να είναι σε δύο θέσεις έτσι παράγει δύο INVITE. Το πρώτο περιέχει SIP:Bob@university.com ως αίτημα-URI και στέλνεται στον κεντρικό υπολογιστή στο university.com. Το δεύτερο INVITE έχει SIP:Bob@131.160.1.112 και στέλνεται στο 131.160.1.112. Ως εκ τούτου, το αίτημα-URI περιέχει τη διεύθυνση του επόμενου hop. Τέλος, ξέρουμε ότι η έκδοση του πρωτοκόλλου είναι SIP/2.0. Επομένως, η γραμμή αιτήματος του INVITE που ελήφθει από τον server του company.com θα ήταν κάπως έτσι:

INVITE sip:Bob.Johnson@company.com SIP/2.0

2.5.2 Μορφή των Απαντήσεων SIP

Μια απάντηση SIP αποτελείται από μια γραμμή θέσης (status), διάφορα header, μια κενή γραμμή, και ένα body μηνύματος. Ο πίνακας 2-5 παρουσιάζει τη μορφή μιας απάντησης SIP. Το body message είναι προαιρετικό μιας και μερικές απαντήσεις δεν το έχουν.



Σχήμα 2-12: Υπόδειξη του επόμενου hop

Status line
Several headers
Empty line
Message body

Πίνακας 2-5: Μορφή μιας απάντησης SIP

Status Line

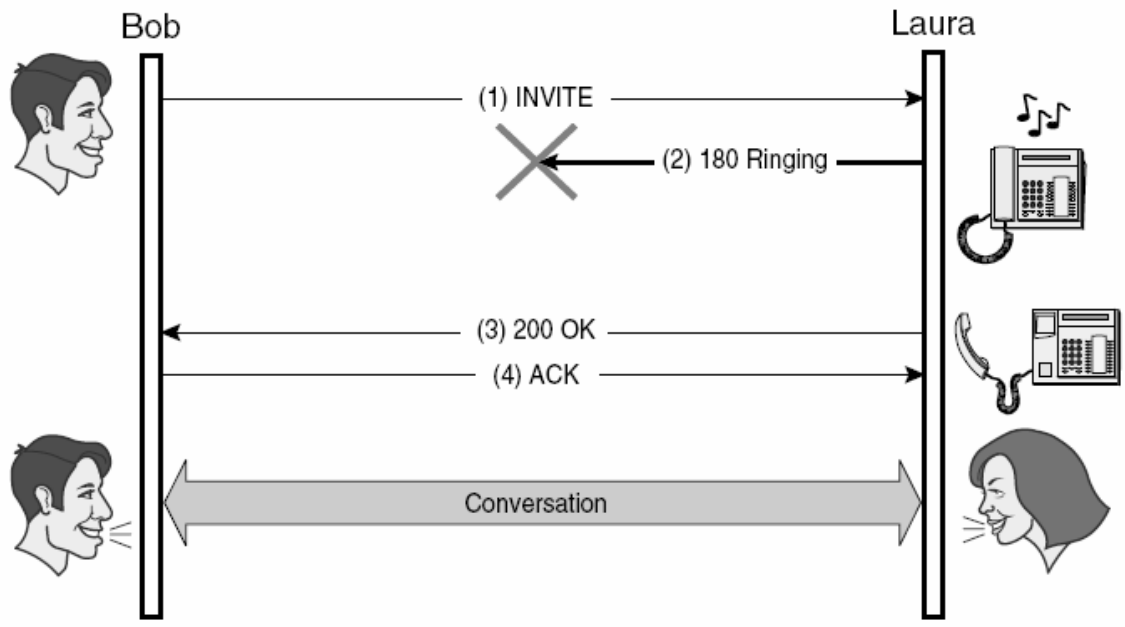
Μια γραμμή θέσης έχει τρία στοιχεία: έκδοση πρωτοκόλλου, κώδικα θέσης, και μια φράση. Η τρέχουσα έκδοση πρωτοκόλλου γράφεται ως SIP/2.0. Ο κώδικας θέσης αναφέρει τη κατάσταση συναλλαγής. Όπως περιγράφεται παραπάνω, οι κώδικες κατάστασης είναι ακέραιοι αριθμοί από 100 έως 699 και ομαδοποιούνται σε έξι διαφορετικές κατηγορίες. Η φράση υπάρχει για τα ανθρώπινα μάτια και μόνο. Κατωτέρω υπάρχει ένα παράδειγμα μιας γραμμής θέσης.

SIP/2.0 180 Ringing

Αξιόπιστη μετάδοση των απαντήσεων

Οι τελικές απαντήσεις διαβιβάζονται αξιόπιστα μεταξύ ενός κεντρικού υπολογιστή και πελάτη, χρησιμοποιώντας αναμεταδόσεις ή ένα αξιόπιστο πρωτόκολλο μεταφοράς για να εξασφαλιστεί η παράδοση. Οι προσωρινές απαντήσεις δεν τυγχάνουν της ίδιας μεταχείρισης. Μπορεί είτε να παραληφθούν από τον πελάτη είτε να χαθούν στο δίκτυο. Το SIP έχει αυτή τη προσέγγιση επειδή ενδιαφέρεται περισσότερο για το εάν μια σύνοδος καθιερώθηκε ή όχι, και τους λόγους για τους οποίους δεν καθιερώθηκε, παρά για το πώς το setup της συνόδου προχωρεί.

Σε μια κλήση SIP, παραδείγματος χάριν, οι καλώντες είναι εγγυημένοι ότι η κλήση έχει γίνει αποδεκτή, αλλά ίσως να μην ξέρουν πότε ο καλούμενος ειδοποιήθηκε. (σχήμα 2-13). Παρόλα αυτά, το SIP μπορεί να επεκταθεί για την αξιόπιστη παράδοση των προσωρινών απαντήσεων εάν αυτό κριθεί απαραίτητο.



Σχήμα 2-13: Προσωρινές απαντήσεις

SIP Headers

Τα αιτήματα SIP περιέχουν μερικά header μετά από τη γραμμή αιτήματος, ενώ οι απαντήσεις SIP τα βάζουν μετά από τη γραμμή θέσης. Τα header παρέχουν πληροφορίες για το αίτημα (ή απάντηση) και για το body που περιέχει. Μερικά header μπορούν να χρησιμοποιηθούν και στα αιτήματα και στις απαντήσεις, αλλά άλλα είναι συγκεκριμένα για αιτήματα (ή απαντήσεις) μόνο. Ένα header αποτελείται από το όνομα του, ακολουθεί μια άνω και κάτω τελεία, και τέλος η τιμή (value) του header. Παραδείγματος χάριν, το header που καλείται "From", το οποίο προσδιορίζει το δημιουργό ενός αιτήματος, μοιάζει με το εξής:

From: Bob Johnson <sip:Bob.Johnson@company.com>

Όπως μπορεί να φανεί σε αυτό το παράδειγμα, το value μπορεί να περιέχει διάφορα πεδία. Σε αυτό το παράδειγμα, το From header έχει δύο πεδία: το όνομα ενός ατόμου και την SIP URL. Ο πίνακας 2-6 περιέχει τα SIP header που καθορίζονται στο πρωτόκολλο. Στη συνέχεια, θα εξηγήσουμε το σκοπό των σημαντικότερων SIP header και θα δώσουμε απλά παραδείγματα χρήσης τους.

Accept	Content-encoding	Max-forwards	Route
Accept-encoding	Content-language	MIME-version	Server
Accept-language	Content-length	Organization	Subject
Alert-info	Content-type	Priority	Supported
Allow	Cseq	Proxy-authenticate	Timestamp
Also	Date	Proxy-authorization	To
Authorization	Encryption	Proxy-require	Unsupported
Call-ID	Error-info	Record-route	User-agent
Call-info	Expires	Require	Via
Contact	From	Response-key	Warning
Content-disposition	In-reply-to	Retry-after	WWW-authenticate

Πίνακας 2-6: SIP Headers

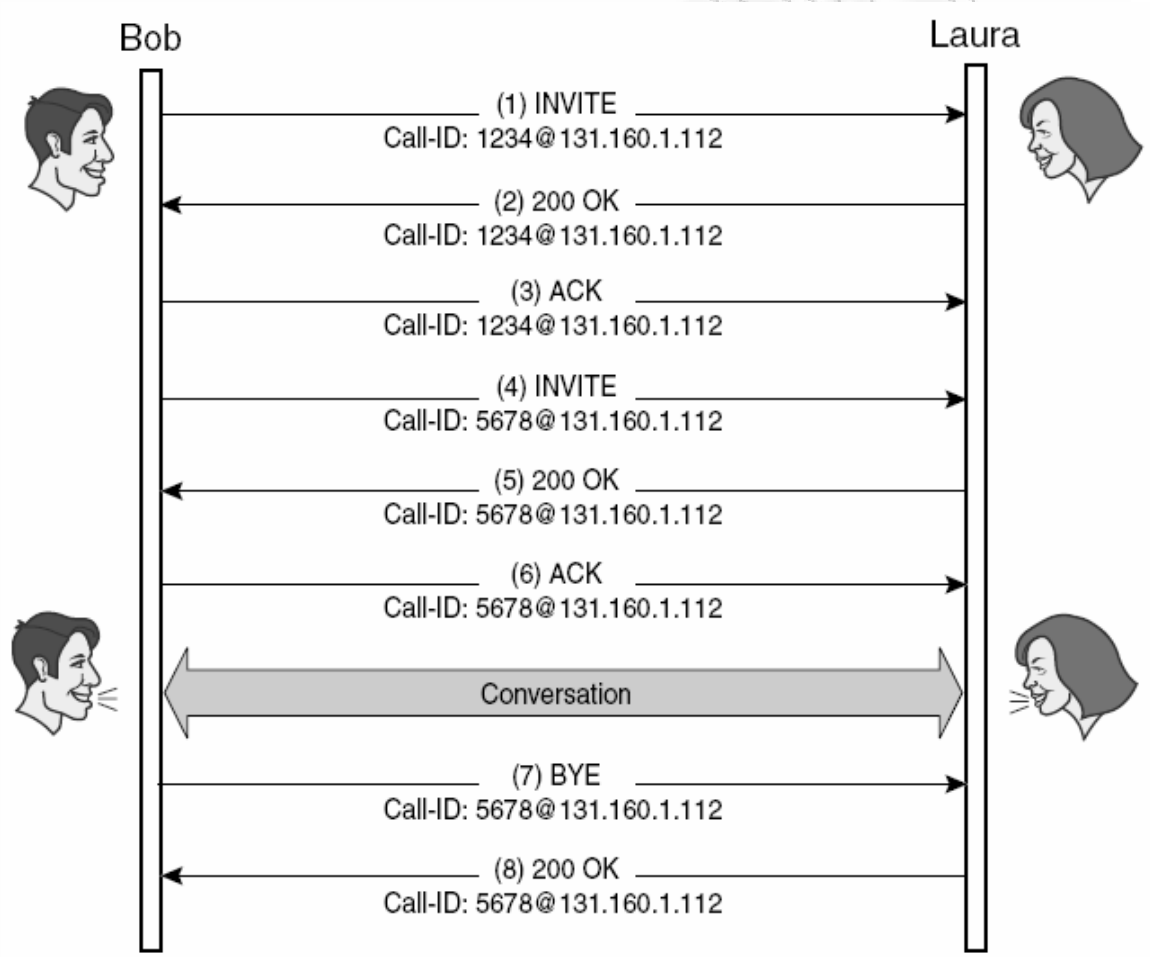
Call-ID

Το Call-ID αντιπροσωπεύει μια σχέση SIP σηματοδοσίας κοινή μεταξύ δύο ή περισσότερων χρηστών. Προσδιορίζει μια ιδιαίτερη πρόσκληση και όλες τις επόμενες συναλλαγές που αφορούν εκείνη την πρόσκληση με μια μορφή όπως την εξής:

Call-ID: ges456fcdw21lkfgte12ax@workstation1234.university.com

Ένας κεντρικός υπολογιστής που ελέγχει SIP σηματοδοσίες για πολλές συνόδους υιοθετεί το Call-ID για να αντιστοιχεί εισερχόμενα μηνύματα στην κατάλληλη σύνοδο. Παραδείγματος χάριν, ο Bob προσκαλεί τη Laura σε μια σύνοδο σκακιού με ένα συγκεκριμένο Call-ID. Το UA της Laura την αποδέχεται και σύντομα το παιχνίδι αρχίζει. Μετά από λίγο, ο Bob καλεί τη Laura να συνομιλήσει μαζί του ενώ παίζουν

ακόμα σκάκι. Αυτό το INVITE από το UA του Bob έχει διαφορετικό Call-ID από το προηγούμενο. Όταν ο Bob και η Laura τελειώσουν, το UA του Bob στέλνει ένα BYE στο UA της Laura για να τελειώσει το τηλεφώνημα. Το UA της Laura χρησιμοποιεί το Call-ID του μηνύματος BYE για να αποφασίσει εάν θα ολοκληρώσει το παιχνίδι σκακιού ή τη συνομιλία (σχήμα 2-14).



Σχήμα 2-14: Χρησιμότητα του Call-ID

Contact

Το header Contact παρέχει ένα URL όπου ο χρήστης μπορεί να βρεθεί άμεσα. Αυτό το χαρακτηριστικό γνώρισμα είναι σημαντικό επειδή αφαιρει φορτίο από τους

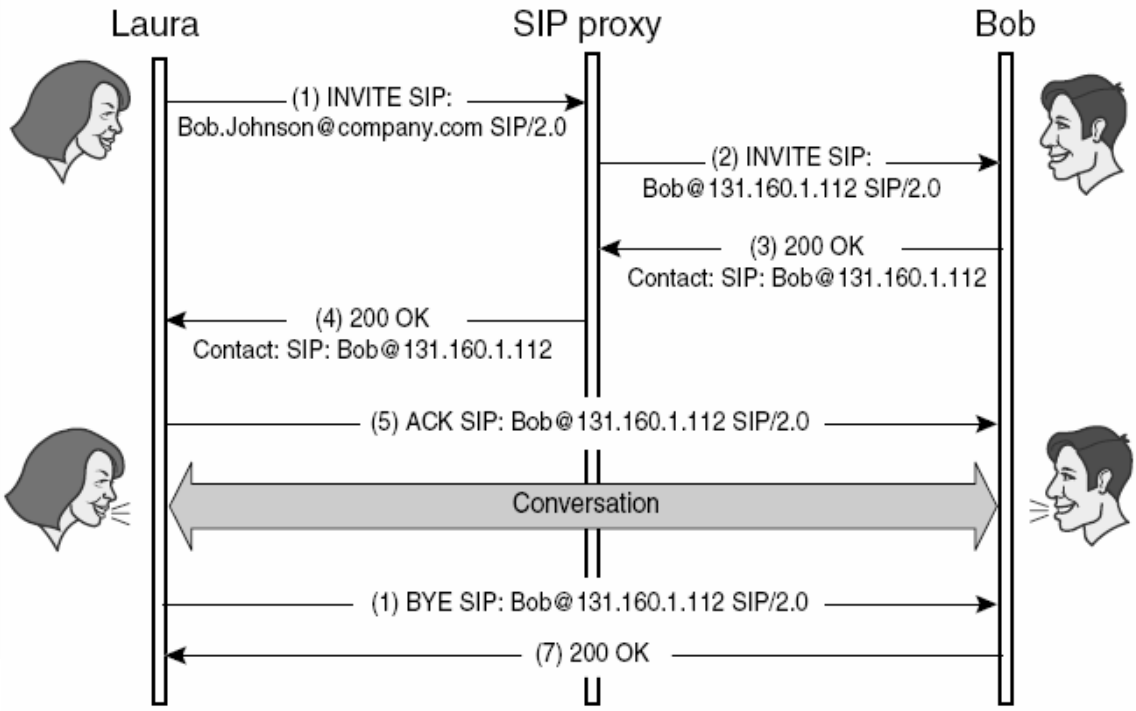
κεντρικούς υπολογιστές SIP που δεν είναι ανάγκη να είναι στο μονοπάτι σηματοδοσίας μετά από τη καθοδήγηση του πρώτου INVITE.

Παραδείγματος χάριν, η Laura καλεί το Bob στο SIP:Bob.Johnson@company.com. Ο proxy του company.com διαβιβάζει το INVITE στο SIP:Bob@131.160.1.112, όπου ο Bob βρίσκεται. Δέχεται την κλήση. Το UA του Bob επιστρέφει απάντηση 200 OK με ένα Contact header:

Contact: Bob Johnson sip:Bob@131.160.1.112

Όταν το UA της Laura λαμβάνει την απάντηση 200 OK, στέλνει ACK στο UA του Bob. Επειδή η θέση του Bob μπορεί να βρεθεί στο Contact header, το ACK στέλνεται άμεσα στο SIP:Bob@131.160.1.112 και το ACK δεν περνά από τον proxy του company.com.

Το σχήμα 2-15 επιδεικνύει το πώς αιτήματα, όπως το BYE, στέλνονται άμεσα μεταξύ των συμμετεχόντων της συνόδου.



Σχήμα 2-15: Χρησιμότητα του Contact header

Cseq

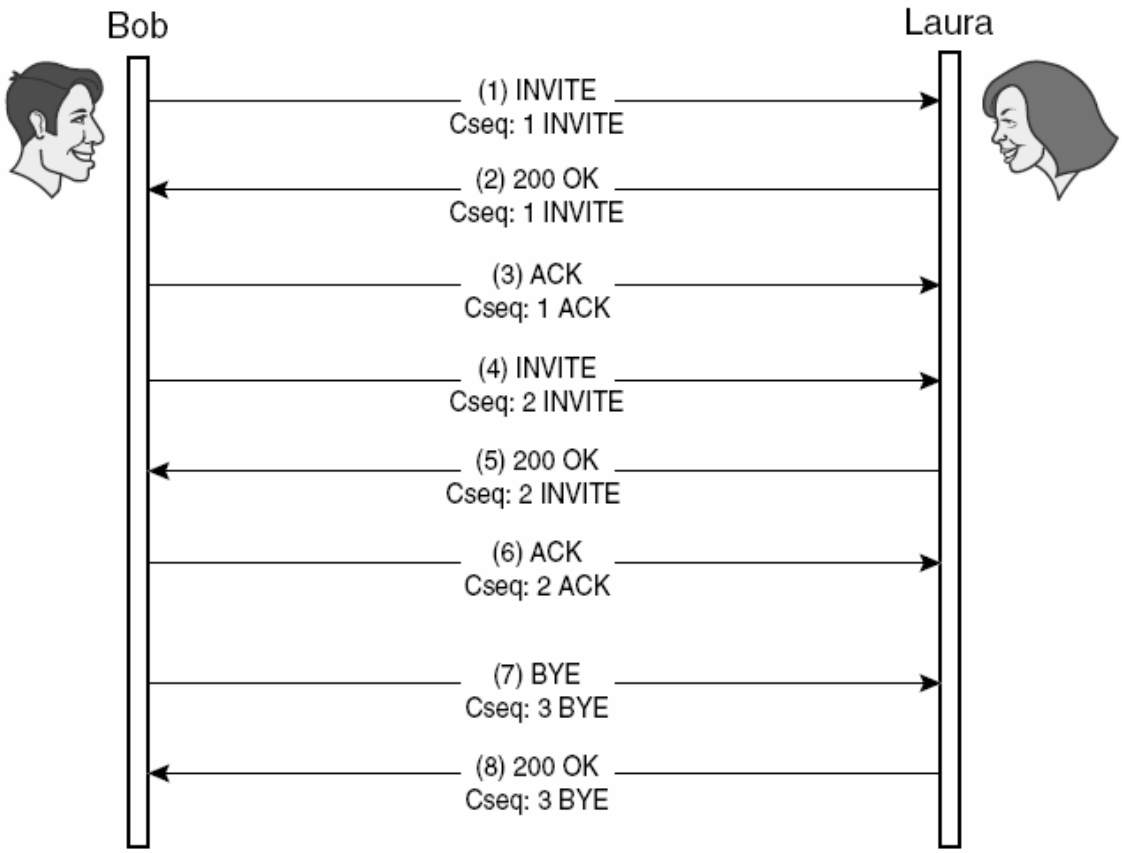
Το header *ακολουθίας εντολής - Command Sequence* (Cseq) έχει δύο πεδία: έναν ακέραιο αριθμό και ένα όνομα μεθόδου. Το αριθμητικό μέρος του Cseq χρησιμοποιείται για να διευθετήσει διαφορετικά αιτήματα μέσα στην ίδια σύνοδο (που καθορίζεται από ένα συγκεκριμένο Call-ID). Χρησιμοποιείται επίσης για να ταιριάξει τα αιτήματα με τις απαντήσεις. Παραδείγματος χάριν, ο Bob στέλνει ένα INVITE στη Laura με το ακόλουθο Cseq:

Cseq: 1 INVITE

Η Laura επιστρέφει μια απάντηση 200 OK με το ίδιο Cseq με το INVITE. Εάν ο Bob θέλει να τροποποιήσει τη σύνοδο που έχει καθιερωθεί ήδη, θα στείλει ένα δεύτερο INVITE (re-INVITE) με το ακόλουθο Cseq:

Cseq: 2 INVITE

Εάν μια αναμετάδοση της απάντησης 200 OK καθυστερήσει από το δίκτυο και φθάσει στο UA του Bob αφού έχει παραγάγει το δεύτερο INVITE, ξέρει ότι αυτό ήταν μια απάντηση για το πρώτο INVITE, χάρη στο header Cseq (σχήμα 2-16).
Μετά από ένα INVITE, όλα τα επόμενα αιτήματα (εκτός από το ACK και CANCEL) περιέχουν ένα Cseq που είναι το αποτέλεσμα της αύξησης κατά ένα από το Cseq του αρχικού αιτήματος.



Σχήμα 2-16: Χρησιμότητα του Cseq header

Cseq σε ACK

Ένα αίτημα ACK έχει το ίδιο Cseq με το INVITE που επιβεβαιώνει. Αυτό επιτρέπει στους proxy να παραγάγουν ACK για μη-επιτυχής τελικές απαντήσεις χωρίς τη δημιουργία νέων Cseq. Στην πραγματικότητα, νέα Cseq μπορούν μόνο να δημιουργηθούν από τα UA, γεγονός που εξασφαλίζει τη μοναδικότητα τους.

Cseq σε CANCEL

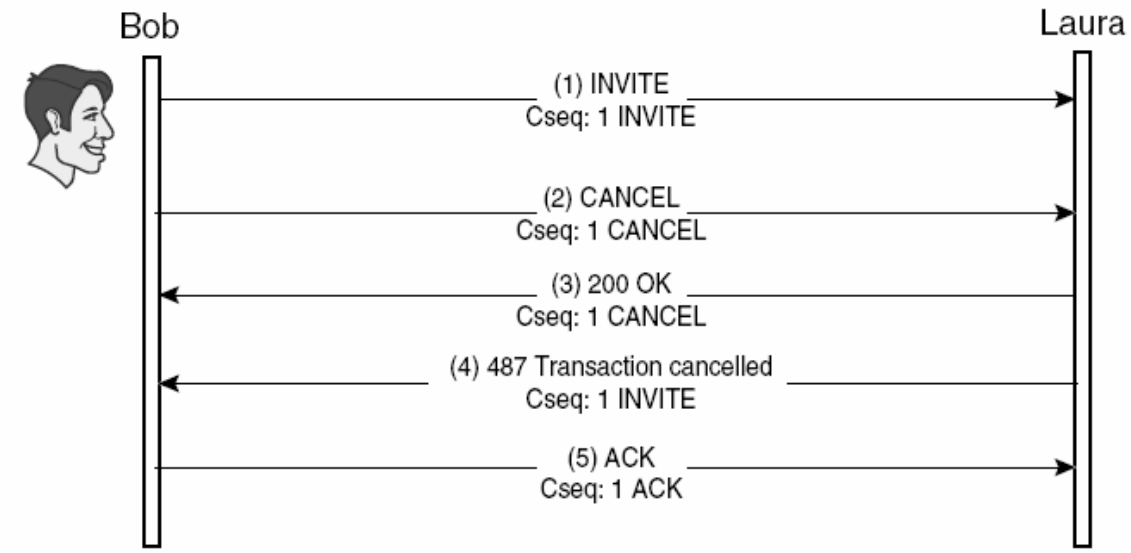
Ένα αίτημα CANCEL έχει το ίδιο Cseq με το αίτημα που ακυρώνει. Αυτό επιτρέπει στους proxy να παραγάγουν CANCEL χωρίς τη δημιουργία νέων Cseq. Επιπλέον, το CANCEL είναι ο λόγος για τον οποίο το Cseq header περιλαμβάνει ένα όνομα της μεθόδου μετά από το αριθμητικό μέρος. Επειδή ο αριθμός Cseq των INVITE και CANCEL είναι ο ίδιος, ένας πελάτης SIP δεν θα μπορούσε να διακρίνει τις

απαντήσεις για CANCEL και τις απαντήσεις για INVITE χωρίς ένα πρόσθετο πεδίο. Το όνομα μεθόδου μέσα στο Cseq λύνει το πρόβλημα (σχήμα 2-17).

From

Το From Header περιέχει αυτόν που εκκίνησε το αίτημα και ένα SIP URL:

From: Bob Johnson sip:Bob.Johnson@company.com



Σχήμα 2-17: Cseq σε CANCEL

Record-Route και Route

Αυτά τα δυο header χρησιμοποιούνται από τους proxy που θέλουν να βρίσκονται στο μονοπάτι σηματοδοσίας για ολόκληρη τη διάρκεια της συνόδου. Είδαμε ότι τα Contact header επιτρέπουν στα UA να στέλνουν αιτήματα άμεσα το ένα στο άλλο. Αυτό αφαιρει φορτίο από τους proxy στο μονοπάτι. Καθοδηγούν το πρώτο INVITE στο κατάλληλο προορισμό και αφήνουν έπειτα τα UA να αρχίσουν να ανταλλάσσουν τη SIP σηματοδοσία.

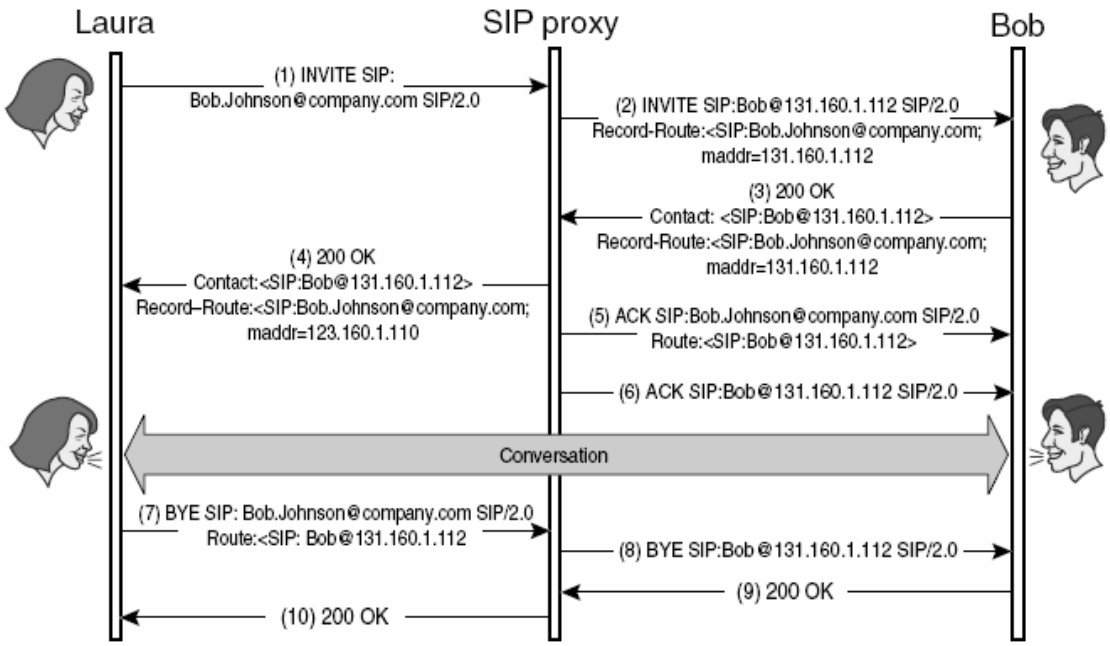
Εντούτοις, μερικές φορές ο proxy πρέπει να μείνει στο μονοπάτι σηματοδοσίας, οπότε ένας μηχανισμός απαιτείται για να κρατήσει τα UA από την ανταλλαγή των

μηνυμάτων SIP μεταξύ τους. Αυτός ο μηχανισμός αποτελείται από δύο header: Route και Record-Route.

Ένας proxy μπορεί να θελήσει να παραμείνει στο μονοπάτι μετά από το πρώτο INVITE για πολλούς λόγους. Ένας από αυτούς είναι η ασφάλεια. Μερικά domain έχουν ένα proxy ασφάλειας, ένα firewall, που φιλτράρει τα εισερχόμενα μηνύματα SIP. Τα μηνύματα SIP που δεν διαπερνούν επιτυχώς το proxy ασφάλειας δεν γίνονται αποδεκτά από το domain. Ένας άλλος λόγος είναι η παροχή υπηρεσιών. Ένας proxy που παρέχει υπηρεσία σχετική με σύνοδο, πρέπει να ξέρει φυσικά πότε η σύνοδος τελειώνει. Τότε είναι που ένα UA στέλνει αίτημα BYE στο άλλο. Είδαμε ήδη ένα παράδειγμα μιας τέτοιας υπηρεσίας στο σχήμα 5-9. Ο call statefull proxy εκείνου του παραδείγματος έπρεπε να δει το BYE από τη Laura στο Bob έτσι ώστε να στείλει email στο Bob με τις πληροφορίες για τη διάρκεια της κλήσης.

Το σχήμα 2-18 επεξηγεί πώς αυτά τα δυο header λειτουργούν. Η Laura στέλνει ένα INVITE στο Bob. Το INVITE διαπερνά έναν SIP proxy που βρίσκεται στο μονοπάτι σηματοδοσίας για τα επόμενα αιτήματα μεταξύ της Laura και του Bob. Ο proxy προσθέτει ένα Record-Route header που περιέχει τη διεύθυνσή του στο INVITE. Το UA του Bob λαμβάνει το INVITE πλήρες με αυτό το Record-Route header και το περιλαμβάνει στην απάντηση 200 OK. Το UA του Bob προσθέτει επίσης το Contact header του στην απάντηση.

Η παράμετρος maddr που εμφανίζεται στο Record-Route απλά περιέχει τη διεύθυνση IP του κεντρικού υπολογιστή, η οποία προστίθεται για να καταγράψει την πραγματική IP του κεντρικού υπολογιστή για μελλοντικά αιτήματα. Το UA της Laura λαμβάνει την απάντηση 200 OK και δημιουργεί ένα Route header το οποίο θα χρησιμοποιηθεί στα επόμενα αιτήματα. Το Route header δημιουργείται καί από το Record-Route, καί από το Contact header της απάντησης. Επειδή μόνο ένας proxy χρειάζεται να είναι στο μονοπάτι, όλα τα επόμενα αιτήματα από τη Laura στο Bob (ACK και BYE σε αυτό το παράδειγμα) θα σταλούν σε αυτόν και θα περιέχουν ένα Route header με τη Contact διεύθυνση του Bob. Με αυτό τον τρόπο, ο proxy στέλνει το αίτημα στη διεύθυνση που περιλαμβάνεται στο Route header.



Σχήμα 2-18: Χρησιμότητα του Route header

Πολλοί Proxies

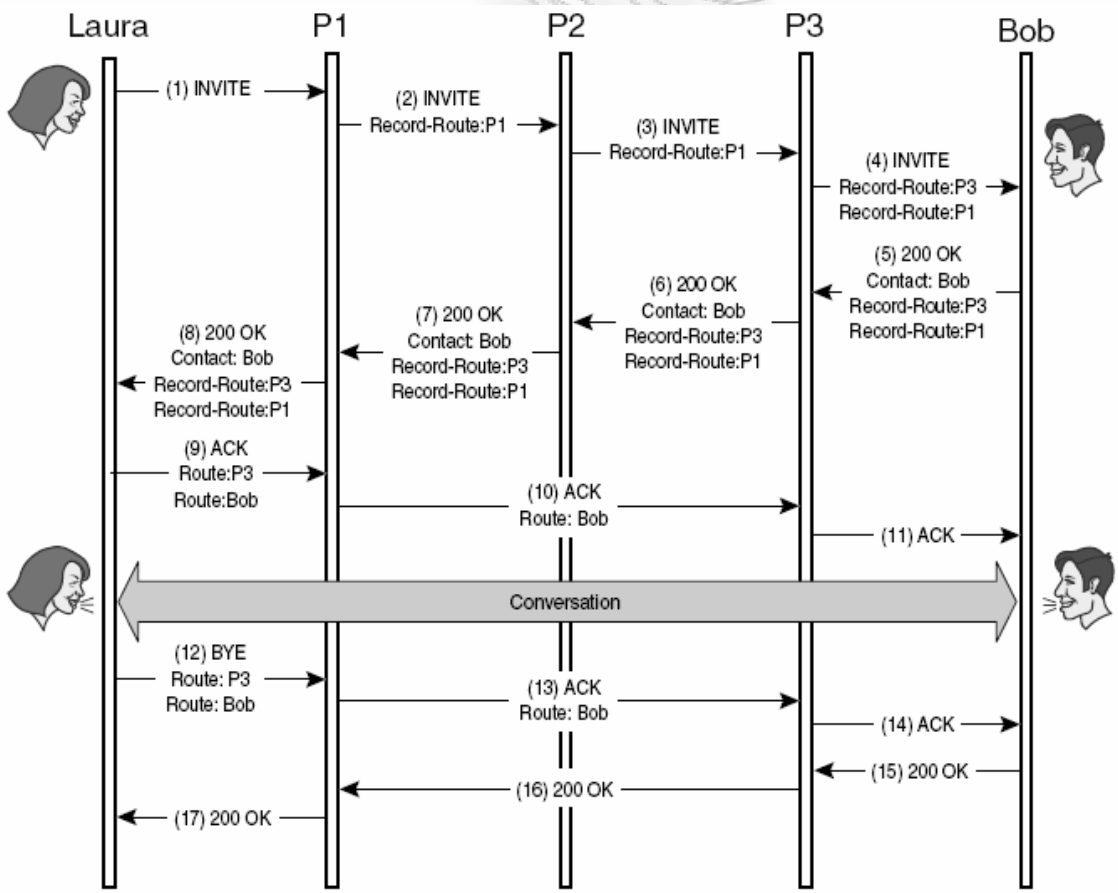
Το προηγούμενο παράδειγμα επιδεικνύει πώς το Record-Route header λειτουργεί για να ενημερώσει το UA της Laura ότι τα επόμενα αιτήματα πρέπει να σταλούν μέσω του proxy παρά άμεσα στο Bob. Εντούτοις, δεν δείχνει γιατί χρειάζεται το Route header. Ένα σενάριο με περισσότερους proxy μπορεί να βοηθήσει στη κατανόηση του σκοπού του Route. Το σχήμα 2-19 περιέχει τρείς proxy: P1, P2, και P3. Οι P1 και P3 πρέπει να είναι στο μονοπάτι σηματοδοσίας, αλλά όχι ο P2. Μπορούμε να δούμε πώς το ACK από τη Laura στο Bob περιέχει ένα Route header που λέει στον P1 να διαβιβάσει το αίτημα στον P3. Η τελευταία διεύθυνση στο Route header είναι η Contact διεύθυνση του Bob.

Να σημειώσουμε ότι το σχήμα 2-19 δεν περιέχει τη πραγματική μορφή των header Contact, Record-Route και Route όπως το σχήμα 2-18. Αντ' αυτού, χρησιμοποιεί μια συμβολική μορφή που απλά προσδιορίζει ποιες διευθύνσεις περιλαμβάνονται σε κάθε header.

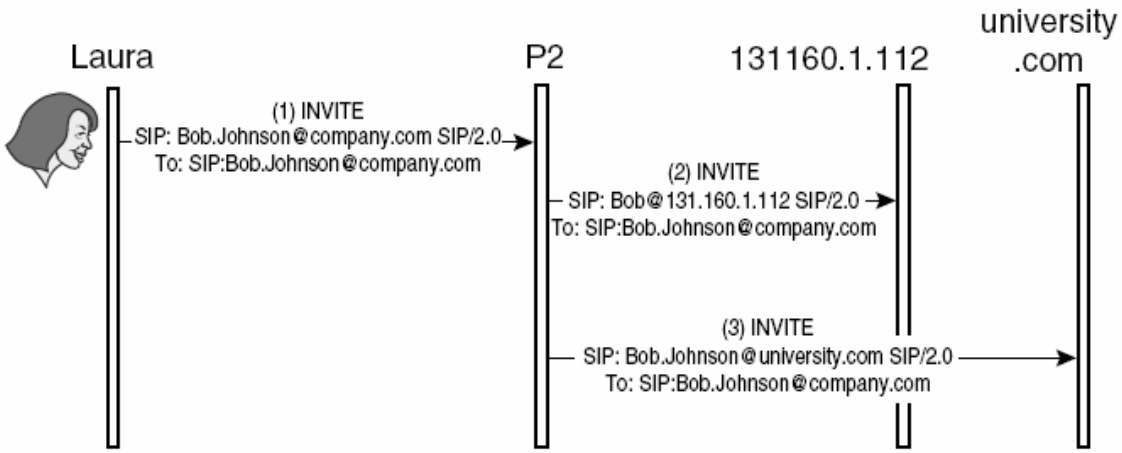
To

To header To περιέχει πάντα τον παραλήπτη του αιτήματος. Συνήθως επίσης περιέχει τη δημόσια διεύθυνση του προορισμού. Είναι σημαντικό να κάνουμε τη διάκριση μεταξύ του To header ενός αιτήματος και του Request-URI. Το To header, που παραμένει το ίδιο καθόλη τη σύνοδο, είναι προορισμένο για απεμακρυσμένο UA. Δεν μπορεί να αλλαχτεί από τους proxy. Το Request-URI περιέχει τη διεύθυνση του επόμενου hop στο μονοπάτι σηματοδοσίας και επομένως αλλάζει από κάθε proxy στην πορεία. Το σχήμα 2-20 επεξηγεί τη χρήση του καθενός.

Η Laura καλεί το Bob χρησιμοποιώντας τη δημόσια διεύθυνσή του: SIP:Bob.Johnson@company.com. Αυτή η SIP URL θα εισαχθεί στο To header και δεν θα αλλάξει κατά τη διάρκεια της συνόδου, δηλαδή όλα τα αιτήματα από τη Laura στο Bob θα έχουν το ίδιο πεδίο To.



Σχήμα 2-19: Παράλειψη του P2 proxy



Σχήμα 2-20: Χρησιμότητα του To header

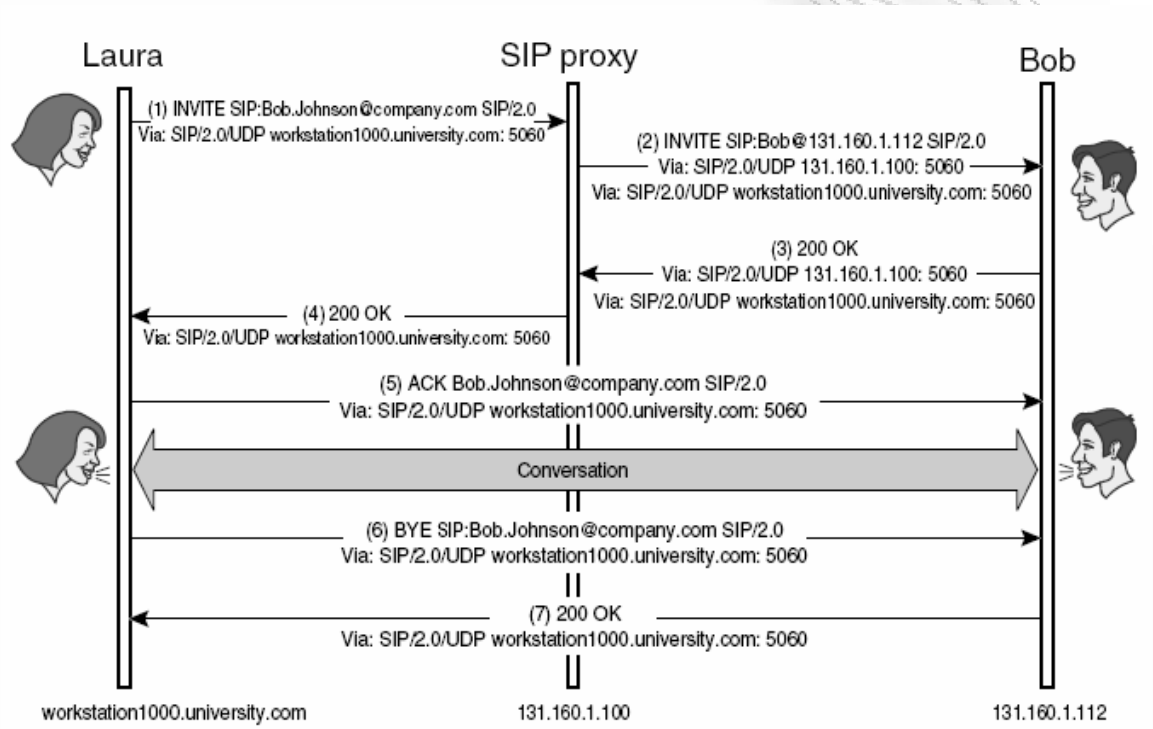
Η Laura τοποθετεί την ίδια SIP URL στο Request-URI έτσι ώστε να στείλει το αίτημα στο SIP στο company.com. Αυτός ο proxy εκτελεί μια παράλληλη αναζήτηση με το να δοκιμάζει δύο διαφορετικές SIP URL: SIP:Bob@131.160.1.112 και SIP:Bob@university.com. Και τα δύο INVITE που στέλνονται από τον proxy στο company.com έχουν το ίδιο To header, αλλά έχουν διαφορετικά Request-URI.

Via

Τα Via header αποθηκεύουν όλους τους proxy που χειρίζονται το αίτημα. Ως εκ τούτου, περιέχουν το μονοπάτι που ακλούθησε το αίτημα (σχήμα 2-21). Αυτές οι πληροφορίες χρησιμοποιούνται για την ανίχνευση βρόχων δρομολόγησης (routing loops). Εάν ένα αίτημα προωθηθεί σε έναν βρόχο, οποιοσδήποτε proxy μπορεί να το παρατηρήσει απλά με την επιθεώρηση των Via header. Εάν βρει τη διεύθυνση του εκεί, ο proxy ξέρει ότι έχει χειριστεί ήδη αυτό το αίτημα. Ένα χαρακτηριστικό Via header είναι το εξής:

Via: SIP/2.0/UDP workstation1234.company.com

Τα Via header χρησιμοποιούνται επίσης για να δρομολογήσουν απαντήσεις προς τον πελάτη που παρήγαγε το αίτημα. Με αυτό το τρόπο, μια απάντηση SIP διαπερνά το ίδιο σύνολο proxy όπως το αίτημα, αλλά με την αντίθετη κατεύθυνση.



Σχήμα 2-21: Χρησιμότητα του Via header

SIP Bodies

Αμφότερα τα αιτήματα και οι απαντήσεις μπορούν να περιέχουν body μηνυμάτων, που χωρίζονται από τα header μηνυμάτων με μια κενή γραμμή. Το body που φέρεται από τα μηνύματα SIP είναι συνήθως μια περιγραφή συνόδου. Επειδή οι SIP proxy δεν χρειάζεται να εξετάζουν τα body, το περιεχόμενο τους είναι αδιάφορο σε αυτούς. Κατά συνέπεια, οι περιγραφές συνόδου μεταφέρονται απευθείας (end to end) μεταξύ των UA. Όλες οι πληροφορίες που χρειάζονται οι proxy προκειμένου να καθοδηγηθούν τα μηνύματα SIP περιλαμβάνονται στις γραμμές αιτήματος και θέσης και στα SIP header. Επειδή τα SIP body είναι σημαντικά μόνο στα UA, τα body

μηνυμάτων μπορεί να είναι κρυπτογραφημένα από end to end χωρίς να χάνουν τη λειτουργικότητα τους.

Μερικοί proxy, εντούτοις, ίσως να θελήσουν να εξετάσουν την περιγραφή συνόδου. Ένα παράδειγμα είναι ένα proxy ασφάλειας (firewall) που θέλει τις πληροφορίες για τα μέσα (media) που ανταλλάσσονται έτσι ώστε μπορεί να αποκλείσει τα μη εξουσιοδοτημένα. Για παράδειγμα, εάν μια επιχείρηση αποφασίσει ότι οι υπάλληλοί της δεν μπορούν να χρησιμοποιούν τηλεδιασκέψεις, το firewall μπορεί να παρεμποδίζει όλα τα τηλεοπτικά streams ενώ αντίθετα να αφήνει τα ακουστικά.

Το ακόλουθο είναι ένα παράδειγμα μιας περιγραφής συνόδου SDP σε ένα SIP body:

v=0

o=Bob 2890844526 2890842807 IN IP4 131.160.1.112

s=I want to know how you are doing

c=IN IP4 131.160.1.112

t=0 0

m=audio 49170 RTP/AVP 0

Ακριβώς όπως τα μηνύματα ηλεκτρονικού ταχυδρομείου μπορούν να φέρουν περισσότερα από ένα επισυναπτόμενα αρχεία (attachment), έτσι και τα μηνύματα SIP μπορούν να φέρουν πολλά body. Παραδείγματος χάριν, η Laura μπορεί να στείλει ένα INVITE με δύο body: μια περιγραφή συνόδου και τη φωτογραφία της. Με αυτό το τρόπο, το UA του Bob μπορεί να δείχνει τη φωτογραφία της στην οθόνη ενώ ο ίδιος ειδοποιείται.

2.6 Transport Layer (Στρώμα Μεταφοράς)

Αναφέραμε προηγουμένως ότι το SIP είναι ένα πρωτόκολλο στρώματος εφαρμογής (application layer). Επομένως, χρησιμοποιεί τα πρωτόκολλα στρώματος μεταφορών (transport layer) για να διαβιβάζει τα αιτήματα και τις απαντήσεις. Η συμπεριφορά οποιουδήποτε πρωτοκόλλου στρώματος εφαρμογής ποικίλλει με το τύπο

μεταφοράς που χρησιμοποιείται. Εάν είναι αξιόπιστο, το πρωτόκολλο στρώματος εφαρμογής δημιουργεί ένα μήνυμα και το παραδίδει στο στρώμα μεταφοράς, αναμένοντας ότι το μήνυμα θα φθάσει στον προορισμό του. Το στρώμα εφαρμογής δεν γνωρίζει το πώς το στρώμα μεταφοράς ολοκληρώνει την παράδοσή του. Απλώς ξέρει ότι η εργασία εκτελείται [11].

Πώς εκτελείται; Συνήθως, το στρώμα μεταφοράς θα αναμεταδώσει το μήνυμα μέχρι το άλλο άκρο το λάβει και στείλει πίσω κάποιο τύπο μηνύματος επιβεβαίωσης. Αυτές οι αναμεταδόσεις είναι αδιάφορες (transparent) για το στρώμα εφαρμογής. Από την άλλη πλευρά, εάν ένα πρωτόκολλο στρώματος εφαρμογής "τρέχει" πάνω από ένα αναξιόπιστο πρωτόκολλο στρώματος μεταφοράς όπως το UDP, δεν μπορεί να υποθέσει την σίγουρη παράδοση των μηνυμάτων. Επομένως, οι αναμεταδόσεις στρώματος εφαρμογής πρέπει να εφαρμοστούν. Αυτό συμβαίνει ως εξής.

Το πρωτόκολλο στρώματος εφαρμογής δημιουργεί ένα μήνυμα και το δίνει στο στρώμα μεταφοράς. Εάν αποτύχει να λάβει μια επιβεβαίωση λήψης από το προορισμό σε μια ορισμένη χρονική περίοδο, θα δημιουργήσει το ίδιο μήνυμα πάλι και θα το περάσει στο στρώμα μεταφοράς ξανά. Με τη χρησιμοποίηση αυτών των time-out στο στρώμα εφαρμογής με τις αναμεταδόσεις του, το πρωτόκολλο στρώματος εφαρμογής μπορεί να εκμεταλλεύεται τους αναξιόπιστους μηχανισμούς μεταφοράς. Ας δούμε τώρα πώς το SIP λειτουργεί στους δύο τύπους μεταφοράς.

2.6.1 Συναλλαγές INVITE

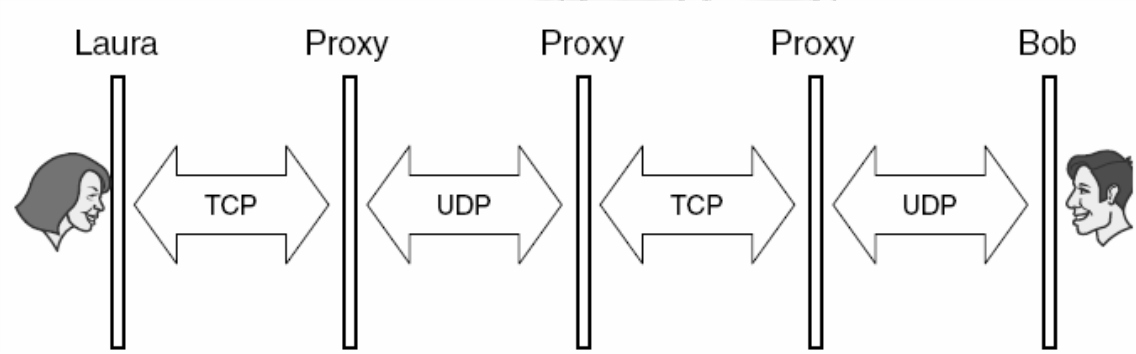
Επειδή οι συναλλαγές INVITE περιλαμβάνουν μια τριπλή χειραψία και ένα ACK αίτημα, απαιτούν διαφορετικό χειρισμό από οποιαδήποτε άλλη συναλλαγή. Επομένως, οι οντότητες SIP μεταχειρίζονται τα INVITE και τα ACK με διαφορετικό τρόπο από τις άλλες μεθόδους.

Αντιμετώπιση Hop-by-Hop

Ας θυμηθούμε ότι όταν ένας proxy βρίσκεται στην πορεία μεταξύ δύο UA, διαφορετικά πρωτόκολλα μεταφορών μπορούν επίσης να είναι μεταξύ τους (σχήμα

2- 22). Ένα UA που χρησιμοποιεί ένα αξιόπιστο πρωτόκολλο μεταφοράς προς ένα proxy δεν μπορεί να είναι σίγουρο ότι η ίδια μεταφορά θα χρησιμοποιηθεί μέχρι τα δεδομένα φτάσουν στο άλλο UA.

Το SIP παρέχει έναν μηχανισμό για να εξασφαλίσει ότι το INVITE τελικά θα παραδοθεί, δηλαδή, καθιστά τους proxy αρμόδιους για τη μεταφορά του INVITE στο επόμενο hop στην πορεία του. Σημειώστε ότι οι stateless proxy δεν μπορούν να αναλάβουν αυτή την ευθύνη επειδή δεν διατηρούν τις πληροφορίες κατάστασης που απαιτούνται για αναμετάδοση του INVITE εάν αυτό χαθεί. Επομένως, το επόμενο hop όσο αφορά την μεταφορά αναφέρεται ως ο επόμενος stateful proxy (ή το τελικό UA).



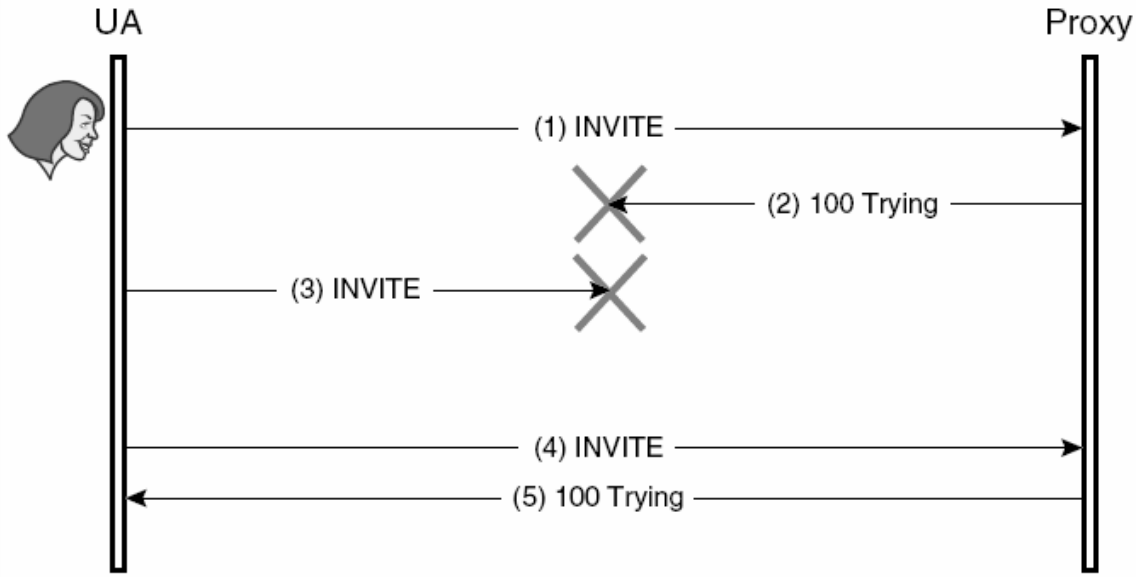
Σχήμα 2-22: Διαφορετικοί τύποι μεταφοράς μεταξύ δύο χρηστών

Μεταδίδοντας ένα INVITE

Επειδή καί ένα UA καί ένας proxy έχουν την ίδια ευθύνη στο να φθάσει το INVITE στο επόμενο hop, οι μηχανισμοί που χρησιμοποιούνται μεταξύ UA και proxy, μεταξύ δύο proxy, και μεταξύ proxy και UA είναι ακριβώς οι ίδιοι. Θα εξηγήσουμε τη συμπεριφορά ενός UA που στέλνει INVITE σε έναν proxy, αλλά αυτός ο proxy θα χρησιμοποιήσει ακριβώς τους ίδιους μηχανισμούς προς το επόμενο proxy στην πορεία. Ένα SIP UA που στέλνει INVITE σε έναν proxy πάνω σε ένα αξιόπιστο πρωτόκολλο μεταφοράς δεν είναι απαραίτητο να εκτελέσει οποιαδήποτε ειδική εργασία, αλλά εάν ένα αναξιόπιστο πρωτόκολλο μεταφοράς όπως το UDP

χρησιμοποιηθεί, πρέπει να είναι έτοιμο για αναμετάδοση, μερικές φορές επανειλημμένα, μέχρι η απάντηση παραληφθεί. (σχήμα 2-23).

Οι proxy server που λαμβάνουν ένα INVITE παράγουν πάντα μια προσωρινή απάντηση "100 Trying". Ένα UA που λαμβάνει INVITE μπορεί να παράγει διάφορες προσωρινές απαντήσεις, όπως "180 Ringing".



Σχήμα 2-23: Αλλεπάλληλη μετάδοση αιτημάτων INVITE

Μεταδίδοντας Απαντήσεις σε ένα INVITE

Είδαμε ότι προσωρινές απαντήσεις χρησιμοποιούνται για να αποτρέψουν τις αναμεταδόσεις INVITE από hop σε hop. Εντούτοις, τίποτα δεν εξασφαλίζει ότι μια προσωρινή απάντηση από το UA του καλούμενου θα φθάσει στο UA του καλώντα. Οι proxy στο μονοπάτι συνήθως θα διαβιβάσουν την απάντηση στο προηγούμενο hop μια φορά, αλλά δεν θα την αναμεταδώσουν εάν αποτύχει να φθάσει (σχήμα 2-24).

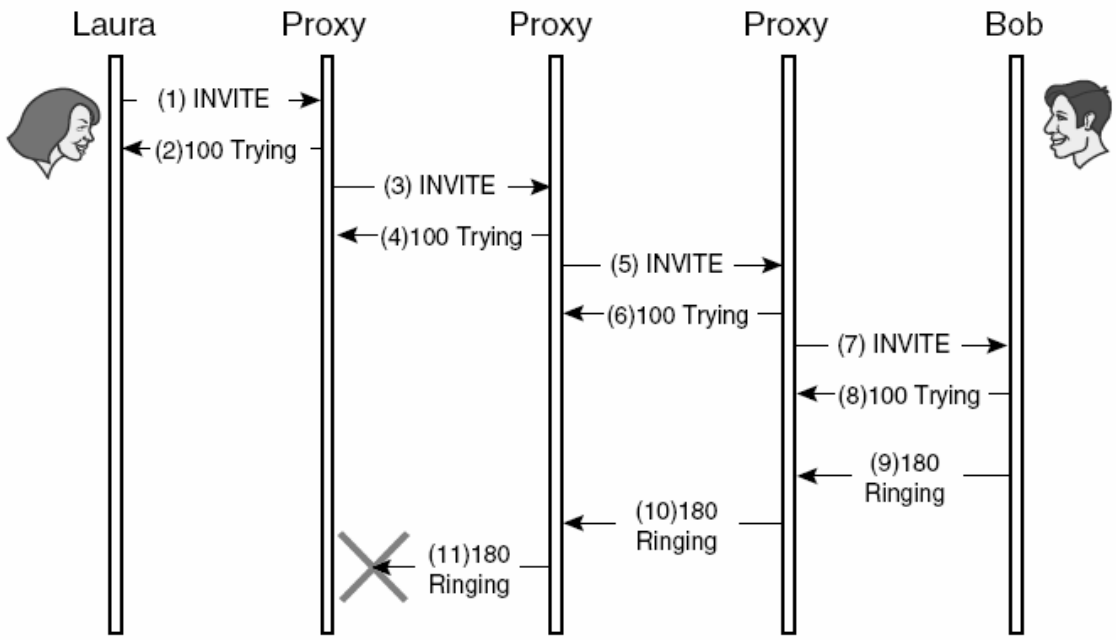
Αντίθετα, το SIP εγγυάται ότι οι τελικές απαντήσεις φθάνουν στον προορισμό τους. Επιτυχείς απαντήσεις (200 έως 299) παραδίδονται αξιόπιστα στο αρχικό UA. Μη-

επιτυχείς τελικές απαντήσεις (300 έως 699) χρησιμοποιούν το ίδιο μηχανισμό hop-by-hop όπως το INVITE.

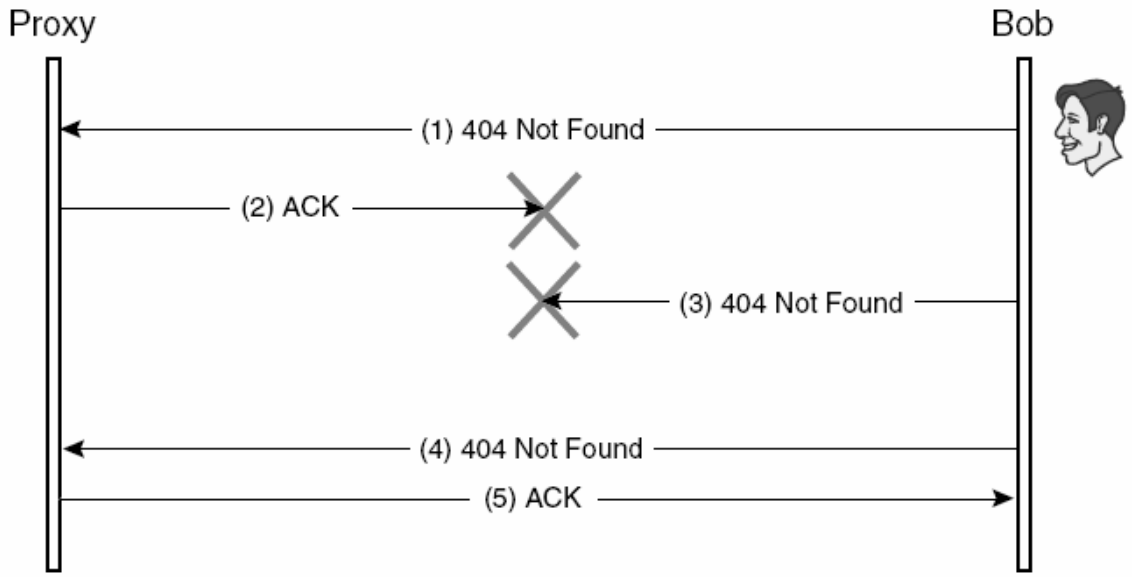
Μη επιτυχείς τελικές απαντήσεις

Η ιδέα πίσω από τη μετάδοση μη-επιτυχών τελικών απαντήσεων είναι η ίδια με αυτήν πίσω από τη μετάδοση των INVITE. Κάθε κεντρικός υπολογιστής εξασφαλίζει ότι το προηγούμενο hop λαμβάνει την απάντηση και ότι αναλαμβάνει την ευθύνη για την διαχείριση της. Ένα UA που χρησιμοποιεί ένα αναξιόπιστο πρωτόκολλο μεταφορών αναμεταδίδει τη μη-επιτυχή τελική απάντηση μέχρι ένα ACK φθάσει (σχήμα 2-25).

Θεωρητικά, ένα UA που χρησιμοποιεί ένα αξιόπιστο πρωτόκολλο μεταφοράς δεν θα έπρεπε να χρησιμοποιεί το ACK. Εντούτοις, προκειμένου το πρωτόκολλο να φαίνεται ομοιογενές σε αξιόπιστες και αναξιόπιστες μεταφορές, τα ACK χρησιμοποιούνται πάντα.

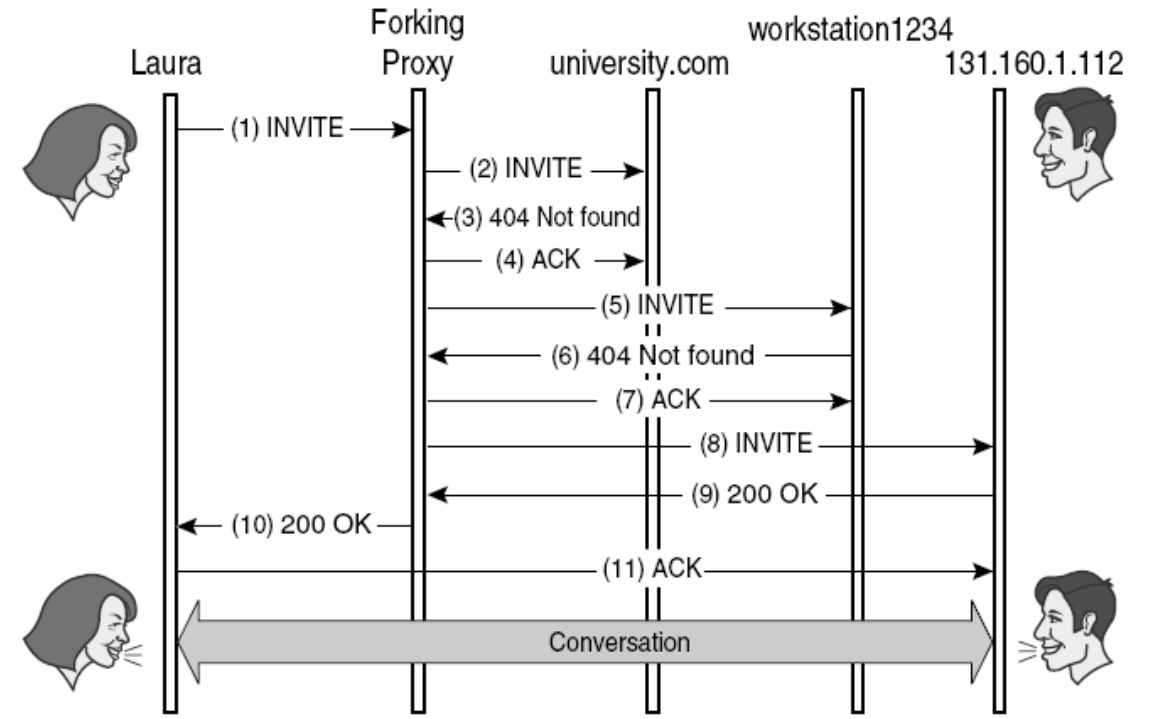


Σχήμα 2-24: Προσωρινές απαντήσεις



Σχήμα 2-25: Αλλεπάλληλη μετάδοση απάντησης “404 Not Found”

Έχουμε αναφέρει μερικές περιπτώσεις στις οποίες μη-επιτυχείς τελικές απαντήσεις δεν μεταδίδονται στο αρχικό UA. Το σχήμα 2-26 επιδεικνύει πώς ο forking server στο company.com λαμβάνει απαντήσεις "404 Not Found" τις οποίες δεν μεταδίδει στο UA της Laura, χάρη στο μηχανισμό μεταφορών hop-by-hop που χρησιμοποιείται για αυτό το είδος απάντησης.



Σχήμα 2-26: Διαχείριση προσωρινών απαντήσεων από τον forking proxy

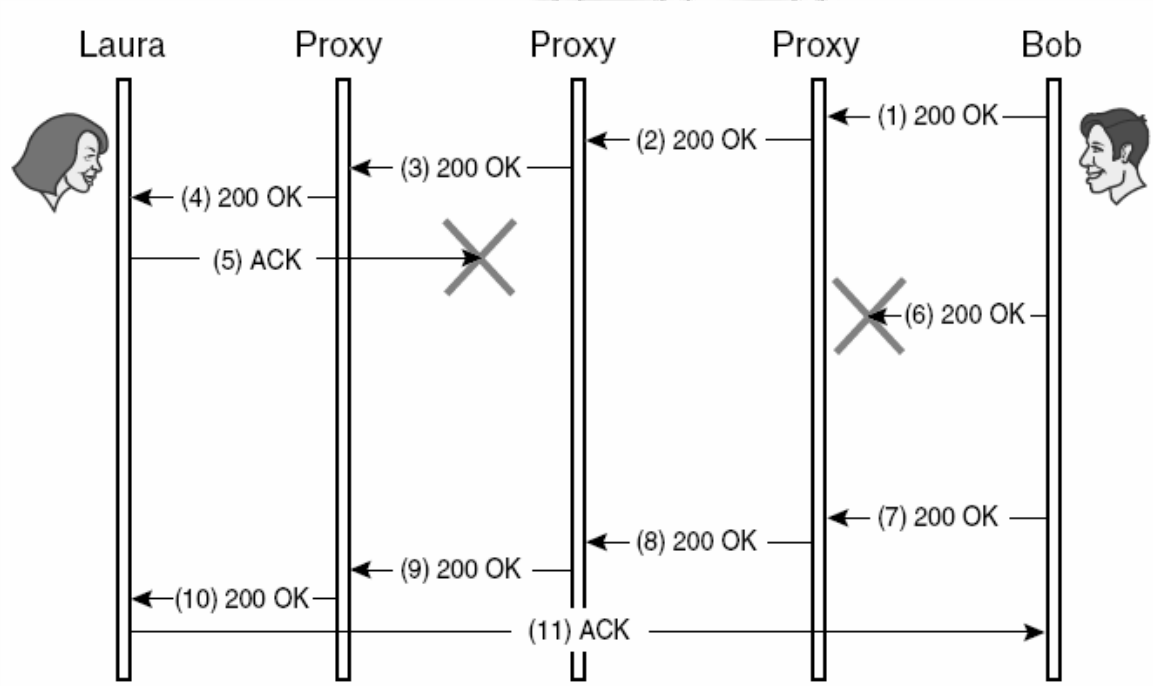
Επιτυχείς τελικές απαντήσεις

Οι επιτυχείς τελικές απαντήσεις διαβιβάζονται αξιόπιστα από end σε end μεταξύ των UA και δεν χρειάζονται το μηχανισμό hop-by-hop που χρησιμοποιείται από άλλες τελικές απαντήσεις (σχήμα 2-27). Μόνο τα UA που δημιούργησαν ένα INVITE μπορούν να στείλουν ένα ACK ως μια τελική επιτυχή απάντηση. Επομένως, άσχετα από το πρωτόκολλο μεταφορών που χρησιμοποιείται (αξιόπιστο ή αναξιόπιστο), ένα UA αναμεταδίδει επιτυχείς τελικές απαντήσεις έως ότου λάβει ένα ACK από το αρχικό UA. Οι proxy στο μονοπάτι διαβιβάζουν απλά τις επιτυχείς τελικές απαντήσεις και τα ACK τους. Δεν εμπλέκονται στην αξιοπιστία.

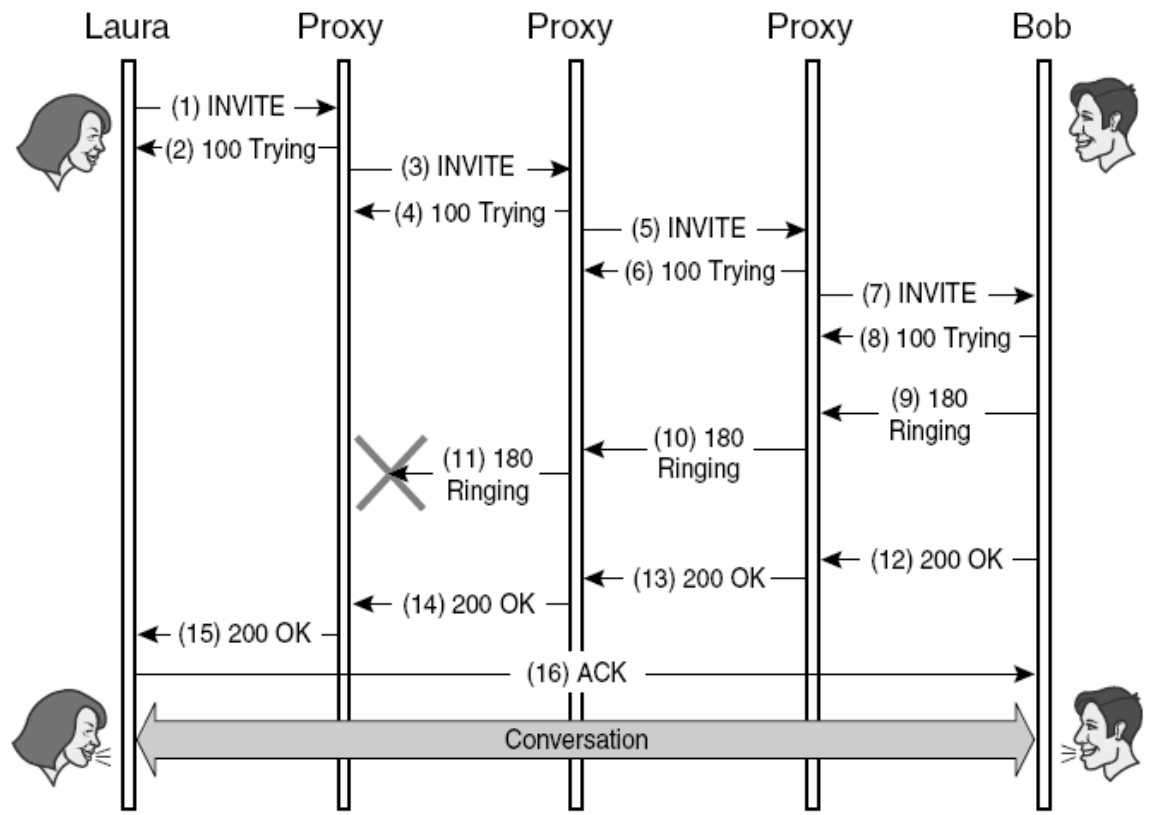
Το σχήμα 2-28 παρουσιάζει ολόκληρη τη διαδικασία καθιέρωσης συνόδου από το INVITE έως ότου πραγματοποιηθεί η πραγματική συνομιλία.

2.6.2 Συναλλαγές CANCEL

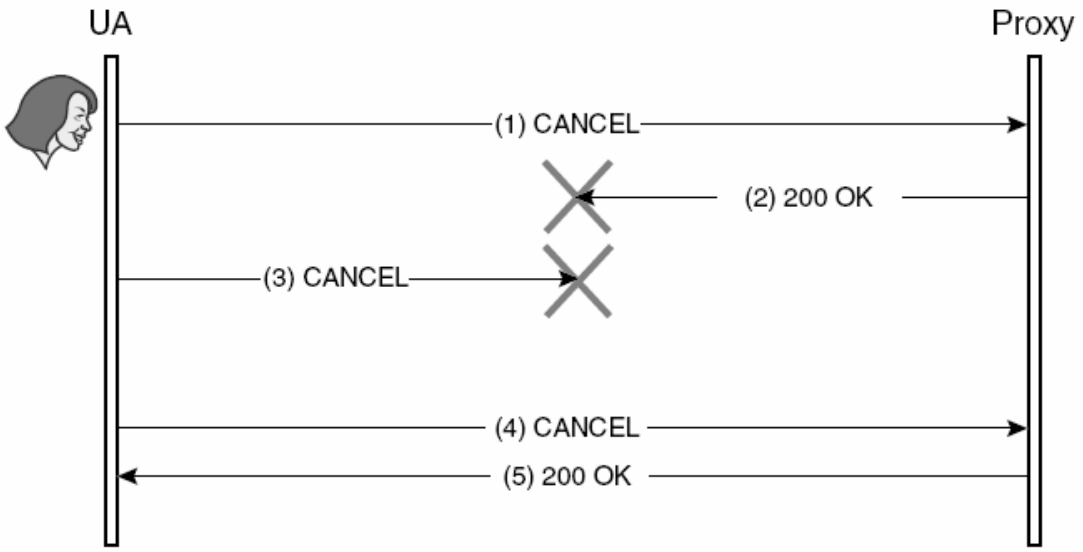
Όπως οι συναλλαγές hop-by-hop, έτσι και οι συναλλαγές INVITE αντιμετωπίζονται με έναν ειδικό τρόπο. Όταν ένα UA στέλνει CANCEL σε έναν proxy, αυτός αποκρίνεται με μια τελική απάντηση. Σε εκείνο το σημείο, η CANCEL συναλλαγή θεωρείται τελειωμένη για το UA. Έπειτα, ο proxy θα στείλει ένα άλλο CANCEL στο επόμενο hop, και επίσης θα λάβει μια τελική απάντηση. Είναι αντιληπτό, ότι η αξιοπιστία για τα αιτήματα CANCEL είναι εύκολο να επιτευχθεί με την αναμετάδοση (σχήμα 2-29).



Σχήμα 2-27: Επιτυχείς τελικές απαντήσεις



Σχήμα 2-28: Ολοκληρωμένο παράδειγμα



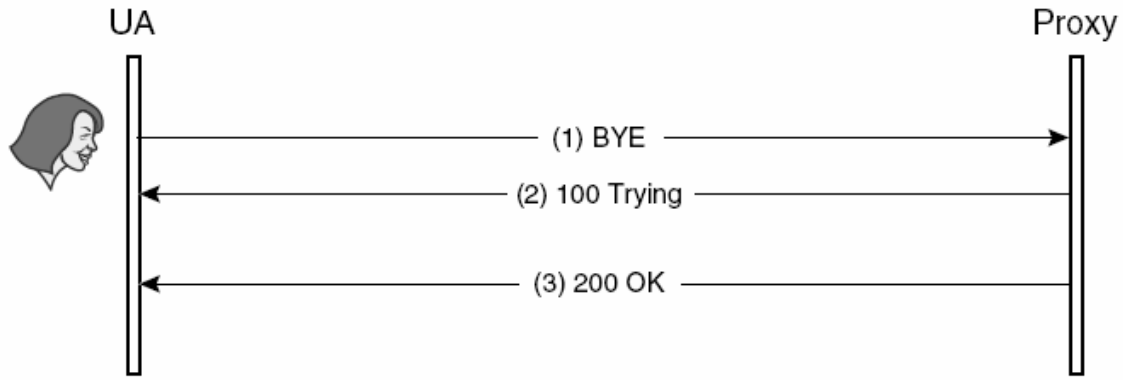
Σχήμα 2-29: Αλλεπάλληλη μετάδοση αιτήματος CANCEL

2.6.3 Άλλες Συναλλαγές

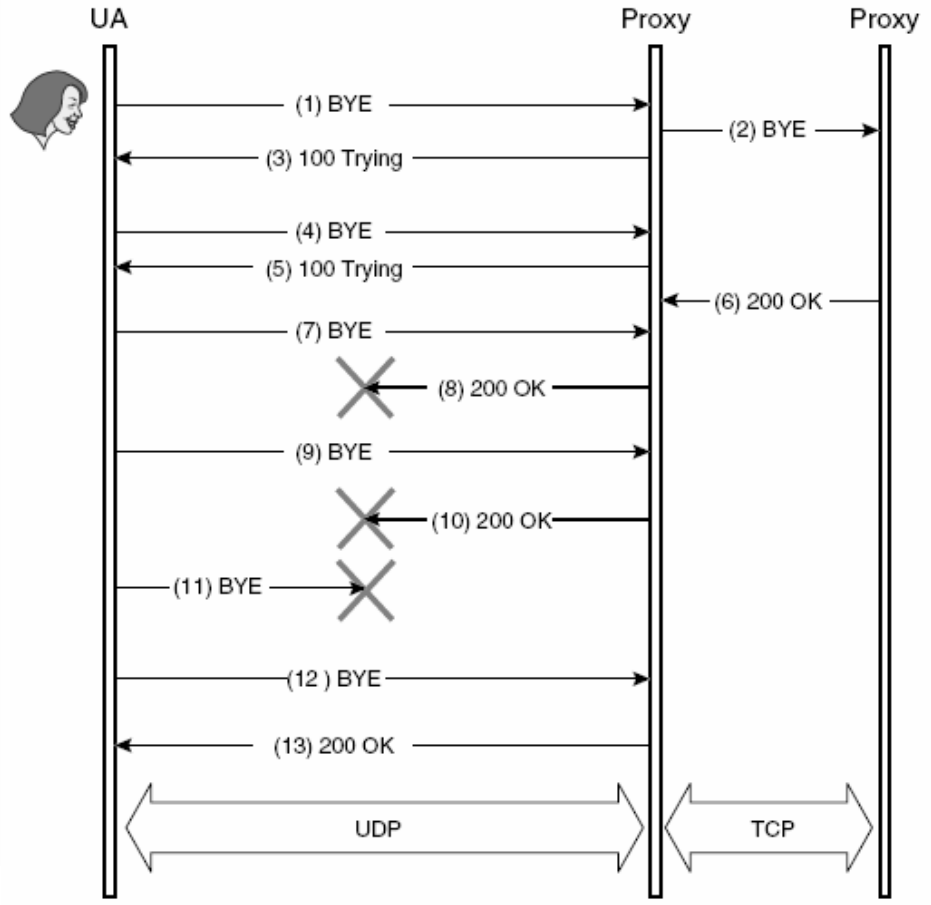
Στην περίπτωση των INVITE, ACK, και CANCEL αιτημάτων, το SIP παρέχει μηχανισμούς αξιοπιστίας κατάλληλους για τα χαρακτηριστικά που παρουσιάζουν και τα τρία παραπάνω. Τα υπόλοιπα SIP αιτήματα ακολουθούν συνηθισμένους κανόνες. Τα OPTIONS, BYE και REGISTER αντιμετωπίζονται με τον ίδιο τρόπο όσον αφορά την αξιοπιστία. Αυτή η κοινή αντιμετώπιση επιτρέπει στο πρωτόκολλο να επεκτείνεται με νέες μεθόδους. Ένας proxy μπορεί να εφαρμόσει τους κοινούς κανόνες αξιοπιστίας σε οποιαδήποτε άγνωστη μέθοδο. Επομένως, σχετικά με την αξιοπιστία, καμία διαφορά δεν εμφανίζεται μεταξύ ενός BYE και μιας νέας μεθόδου [12].

Οι κοινοί κανόνες αξιοπιστίας χρησιμοποιούν επίσης το μηχανισμό hop-by-hop που χρησιμοποιείται για τα INVITE. Ένα UA σιγουρεύεται ότι το αίτημα παραλαμβάνεται από τον επόμενο proxy, και έπειτα ο επόμενος proxy εξασφαλίζει ότι ο ακόλουθος proxy στο μονοπάτι το λαμβάνει, και ούτω καθεξής. Όταν η τελική απάντηση προέρχεται από το απεμακρυσμένο end του δικτύου, ο proxy θα εξασφαλίσει ότι παραδίδεται στο αρχικό UA που δημιούργησε το αίτημα.

Για την αξιόπιστη μεταφορά, το UA στέλνει το αίτημα στον proxy. Όταν η τελική απάντηση φθάσει, ο proxy θα την προωθήσει στο UA επίσης χρησιμοποιώντας αξιόπιστη μεταφορά (σχήμα 2-30). Οποιαδήποτε προσωρινή απάντηση που φθάνει πριν από τη τελική απάντηση στέλνεται επίσης στο UA μέσω του αξιόπιστου πρωτοκόλλου μεταφορών. Για την αναξιόπιστη μεταφορά, το UA πρέπει να εξακριβώσει ότι ο proxy έχει λάβει το αίτημα. Όταν ο proxy λάβει μια απάντηση από το απεμακρυσμένο άκρο (remote end), πρέπει να εξασφαλίσει ότι το UA θα το λάβει επίσης. Το UA αναμεταδίδει το αίτημα μέχρι ο proxy να του δώσει μια τελική απάντηση. Ο proxy αναμεταδίδει την τελική απάντησή του εφ' όσον συνεχίζει να λαμβάνει αιτήματα για αναμετάδοση. Όταν οι αναμεταδόσεις σταματήσουν, ο proxy καταλαβαίνει ότι το UA έχει λάβει την τελική απάντηση (σχήμα 2-31).



Σχήμα 2-30: Αξιόπιστη μετάδοση



Σχήμα 2-31: Οι μεταδόσεις αιτημάτων γίνονται αλλεπάλληλα, ανεξάρτητα από τις προσωρινές απαντήσεις

ΚΕΦΑΛΑΙΟ Γ':

Εφαρμογές του SIP στη πλατφόρμα Asterisk

Το Asterisk όπως πολλά σπουδαία πράγματα προέκυψε από μία ανάγκη: την έλλειψη ευέλικτων και φτηνών τηλεπικοινωνιακών συστημάτων στην αγορά. Ο Marc Spencer, όταν επιχείρησε να δημιουργήσει ένα τηλεφωνικό κέντρο για την παροχή τηλεφωνικών υπηρεσιών υποστήριξης σε Linux συνάντησε το οικονομικό εμπόδιο που θέτουν όλα τα εμπορικά τηλεφωνικά κέντρα. Οι ανάγκες του Spencer απαιτούσαν 24ωρη λειτουργία κατά την οποία ο πελάτης θα έπαιρνε τηλέφωνο στην εταιρία, ένα ηχογραφημένο μήνυμα θα τον προέτρεπε να δώσει τον κωδικό πελάτη μέσω του πληκτρολογίου της τηλεφωνικής του συσκευής και στη συνέχεια θα έλεγε το πρόβλημα του. Το τηλεφωνικό κέντρο θα ηχογραφούσε το πρόβλημα του πελάτη και θα το προωθούσε σε κάποιον τεχνικό, ο οποίος θα έβλεπε τον κωδικό, θα άκουγε το πρόβλημα και θα επικοινωνούσε με τον πελάτη για να του λύσει την απορία. Ο Spencer θεώρησε πως όλα τα υπάρχοντα εμπορικά τηλεφωνικά συστήματα που θα μπορούσαν να καλύψουν τις ανάγκες του ήταν πάρα πολύ ακριβά ή δεν προσέφεραν ακριβώς αυτό που ήθελε. Έτσι προέκυψε η ιδέα του Asterisk. Θα δημιουργούσε ένα πρόγραμμα το οποίο με τη χρήση του κατάλληλου υλικού θα δεχόταν τις εισερχόμενες κλήσεις και θα τις επεξεργαζόταν ανάλογα με τις ανάγκες του [5].

Όμως ακόμα και έτσι, το υλικό που θα μπορούσε να του παρέχει σύνδεση με το δημόσιο τηλεφωνικό δίκτυο ήταν εξοπλισμένο με ακριβούς και εξειδικευμένους DSP μικροεπεξεργαστές. Η απαραίτητη ώθηση για την εξέλιξη του Asterisk ήρθε από το Zapata Telephony Project του Jim Dixon. Ο Dixon πίστευε πως λόγω της εκθετικής ανάπτυξης της υπολογιστικής ισχύος των επεξεργαστών και σε συνδυασμό με την πτώση των τιμών τους, θα μπορούσαν να κατασκευαστούν οικονομικές διεπαφές (PCI κάρτες) χωρίς τα ακριβά DSP κυκλώματα, αφήνοντας έτσι την κωδικό-αποκωδικοποίηση εξ'ολοκλήρου στον επεξεργαστή. Παρόλο που το φορτίο επεξεργασίας που θα έπεφτε στον επεξεργαστή θα ήταν τεράστιο, ο Dixon ήξερε ότι

η αναλογία τιμής/απόδοσης θα έγερνε συνεχώς προς όφελος του. Από το συγκερασμό των παραπάνω καινοτόμων ιδεών, προέκυψε η Digium, μία τηλεφωνική εταιρία η οποία θα βασιζόταν στις κάρτες Zaptel (του Zapata Telephony Project) και το Asterisk για να παρέχει τηλεφωνικές λύσεις ανοιχτού κώδικα.

3.1 Τι είναι το Asterisk

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

Το Asterisk ξεφεύγει από τα όρια του τηλεπικοινωνιακού προγράμματος και χαρακτηρίζεται σωστότερα από την έννοια της τηλεπικοινωνιακής πλατφόρμας. Δημιουργεί δηλαδή ένα πλαίσιο μέσα στο οποίο θα μπορούσε να αναπτυχθεί το οποιοδήποτε υπάρχον (ή μελλοντικό) τηλεπικοινωνιακό σύστημα. Μπορεί να λειτουργήσει ως αυτόνομος εξυπηρετητής επεξεργασίας κλήσεων ή ακόμα και ως μία προσθήκη σε κάποιο ήδη εγκατεστημένο κέντρο.

Το Asterisk είναι δυνατό να χρησιμοποιηθεί μόνο σε επίπεδο λογισμικού, μεταφέροντας φωνή μέσω IP ή να επικοινωνήσει με TDM (Time Division Multiplexing) διεπαφές και να χρησιμοποιήσει το τηλεφωνικό δίκτυο.

Μέσω του Asterisk μπορεί να επιτευχθεί μία ποικιλία μεταβάσεων όπως:

• Κλειστό (εμπορικό) → Ελεύθερο (δωρεάν)

• Δίκτυα μεταγωγής κυκλώματος → VoIP συστήματα

• Φωνή → Φωνή, Βίντεο, Δεδομένα

• Digital Signal Processing → Host Media Processing

• Κεντρική διαχείριση → Peer to Peer

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

Επισήμως το Asterisk ορίζεται ως: *μία πλατφόρμα ελεύθερου λογισμικού, "υβριδικής" πολύπλεξης με διαίρεση χρόνου (hybrid TDM), με δυνατότητες ιδιωτικού συστήματος μεταγωγής τηλεφωνίας - πακέτων φωνής (packet voice PBX) και συστήματος αμφίδρομης φωνητικής απόκρισης (IVR) με λειτουργίες αυτόματης κατανομής κλήσεων (ACD)* [Mark Spencer, Fosdem 2006]. Ο παραπάνω ορισμός αν και δεν καλύπτει το σύνολο των δυνατοτήτων του Asterisk -πράγμα αδύνατο να επιτευχθεί μέσα σε τρεις γραμμές- είναι μία καλή αρχή για να αρχίσει ξετυλίγεται το κουβάρι των τεχνολογιών που το συνθέτουν. Παρακάτω θα επιχειρήσουμε να αποσαφηνίσουμε τις έννοιες του ορισμού και να ξεκινήσουμε σιγά-σιγά να εξετάζουμε επιμέρους δυνατότητες της πλατφόρμας.

3.1.1 Το Asterisk είναι ελεύθερο λογισμικό

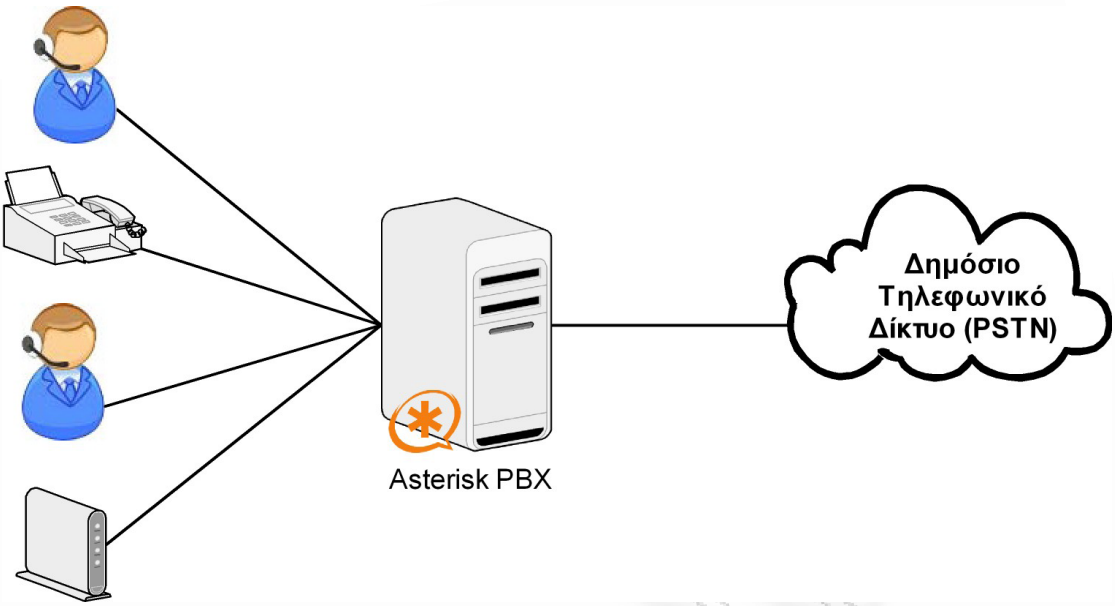
Ξεκινώντας λοιπόν με την έννοια του ελεύθερου λογισμικού, το Asterisk λειτουργεί υπό τη Γενική Άδεια Δημόσιας Χρήσης GNU (GPL), αλλά για λόγους διαλειτουργικότητας (π.χ. υποστήριξη του ιδιωτικού codec G.729) υπάρχει επίσης και η εμπορική διανομή του Asterisk (όπως συμβαίνει και με τη MySQL). Το Asterisk απολαμβάνει αυτήν τη στιγμή μεγάλη εκτίμηση από την κοινότητα του ελεύθερου λογισμικού, το οποίο στην πράξη σημαίνει ότι εκτός από το γεγονός ότι είναι δωρεάν, ταυτόχρονα σε καθημερινή βάση εκατοντάδες (ή ακόμα και χιλιάδες) προγραμματιστές και σύμβουλοι τηλεπικοινωνιών ασχολούνται με το Asterisk, δημιουργώντας εφαρμογές και προσαρμοσμένες εγκαταστάσεις, με σκοπό να το επεκτείνουν, να το βελτιώσουν, να προβλέψουν τις νέες τεχνολογίες που έρχονται (αν όχι να τις δημιουργήσουν) και να τις ενσωματώσουν έγκαιρα.

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

Αυτό σημαίνει ότι υποστηρίζει τα υπάρχοντα TDM πρωτόκολλα τηλεπικοινωνιών όπως το Ψηφιακό Δίκτυο Ενοποιημένων Υπηρεσιών (ISDNBRA-PRI), το Δημόσιο Τηλεπικοινωνιακό Δίκτυο Μεταγωγής (PSTN), το FXS, το FXO, το E1, το T1 και σε γενικές γραμμές οτιδήποτε χρησιμοποιείται στην κλασσική τηλεφωνία όπως τη γνωρίζουμε μέχρι σήμερα. Παράλληλα όμως υποστηρίζει και τα καινούργια VoIP (Voice over Internet Protocol) πρωτόκολλα όπως το SIP, το IAX, το H.323, το MGCP, το SCCP (Sisco-Skinny) και σύντομα το Jingle (Google Talk). Η γεφύρωση των τεχνολογιών προσφέρει στο χρήστη, δυνατότητα μετάβασης στις νέες μορφές επικοινωνίας χωρίς όμως να αναγκαστεί να αποχωριστεί τις παλιές του συνήθειες και εγκαταστάσεις [7].

3.1.3 Το Asterisk είναι ιδιωτικό σύστημα τηλεφωνικής μεταγωγής (PBX)

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



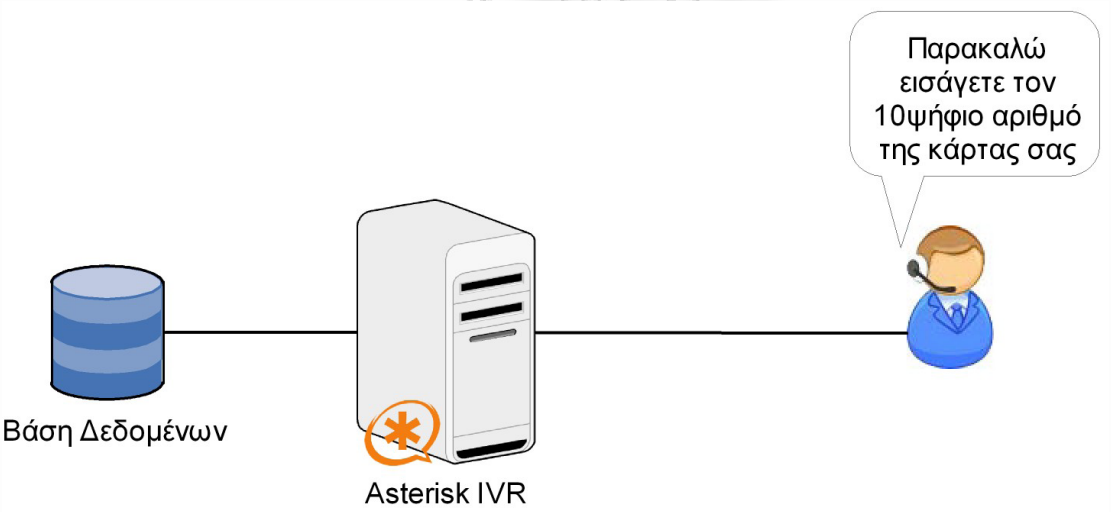
Το Asterisk μπορεί να λειτουργήσει σαν ένα παραδοσιακό PBX με τις ευκολίες που αυτά παρέχουν όπως:

- ενδοεπικοινωνία (extension-to-extension),
- τριμερής επικοινωνία
- τηλεδιάσκεψη
- φωνητικό ταχυδρομείο
- αναμονή κλήσεων
- προώθηση κλήσεων
- καταγραφή αρχείου κλήσεων
- ηχογράφηση συνομιλιών

με τη διαφορά ότι όλα αυτά τα κάνει με πλήρη διαφάνεια και κατά βούληση του χρήστη και έτσι δεν επηρεάζεται από το πώς επικοινωνούν οι χρήστες, δηλαδή με softphones, παραδοσιακά τηλέφωνα ή VoIP handsets και δεν περιορίζει την τοπολογία του τηλεφωνικού δικτύου.

3.1.4 Το Asterisk είναι σύστημα αμφίδρομης φωνητικής απόκρισης

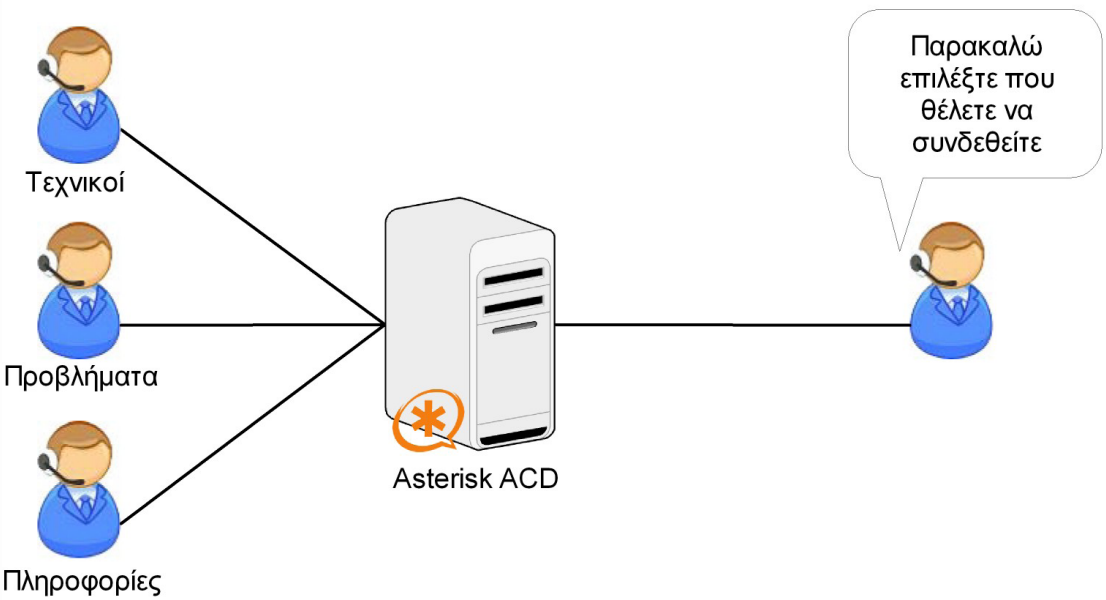
Το Asterisk παρέχει τη δυνατότητα δημιουργίας συστήματος αμφίδρομης φωνητικής απόκρισης. Αυτό σημαίνει ότι ο καλών έχει τη δυνατότητα μέσω ενός φωνητικού μενού και με τη χρήση του πληκτρολογίου της τηλεφωνικής του συσκευής να αλληλεπιδράσει με το τηλεφωνικό κέντρο και να αποκτήσει πρόσβαση σε πληροφορίες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



3.1.5 Το Asterisk είναι σύστημα αυτόματης κατανομής κλήσεων (ACD)

Το Asterisk μπορεί να λειτουργήσει σαν σύστημα ουρών αναμονής των κλήσεων και αυτόματης δρομολόγησης στην κατάλληλη ουρά. Η λειτουργία αυτή, είναι από τις βασικότερες ενός τηλεφωνικού κέντρου, το οποίο θα πρέπει να είναι σε θέση να κάνει σωστή και αποτελεσματική διαχείριση των γραμμών του. Τα κριτήρια

δρομολόγησης μπορούν να ποικίλουν, ανάλογα με την ώρα, τη διαθεσιμότητα, τα προσόντα, τα επίπεδα προτεραιοτήτων, κ.α.



3.2 Αρχιτεκτονική του Asterisk

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

3.2.1 Ο πυρήνας του Asterisk

• Μεταγωγέας PBX: Η πρωταρχική λειτουργία του Asterisk όπως φαίνεται και από την πρώτη ονομασία του (Asterisk the Free PBX) είναι να λειτουργεί σαν σύστημα

PBX, συνδέοντας κλήσεις μεταξύ χρηστών και ενεργειών. Ο πυρήνας μεταγωγής συνδέει χρήστες από διάφορες διεπαφές λογισμικού ή υλικού.

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

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

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

3.2.2 APIs Φόρτωσης Modules

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

Τα APIs είναι τα εξής:

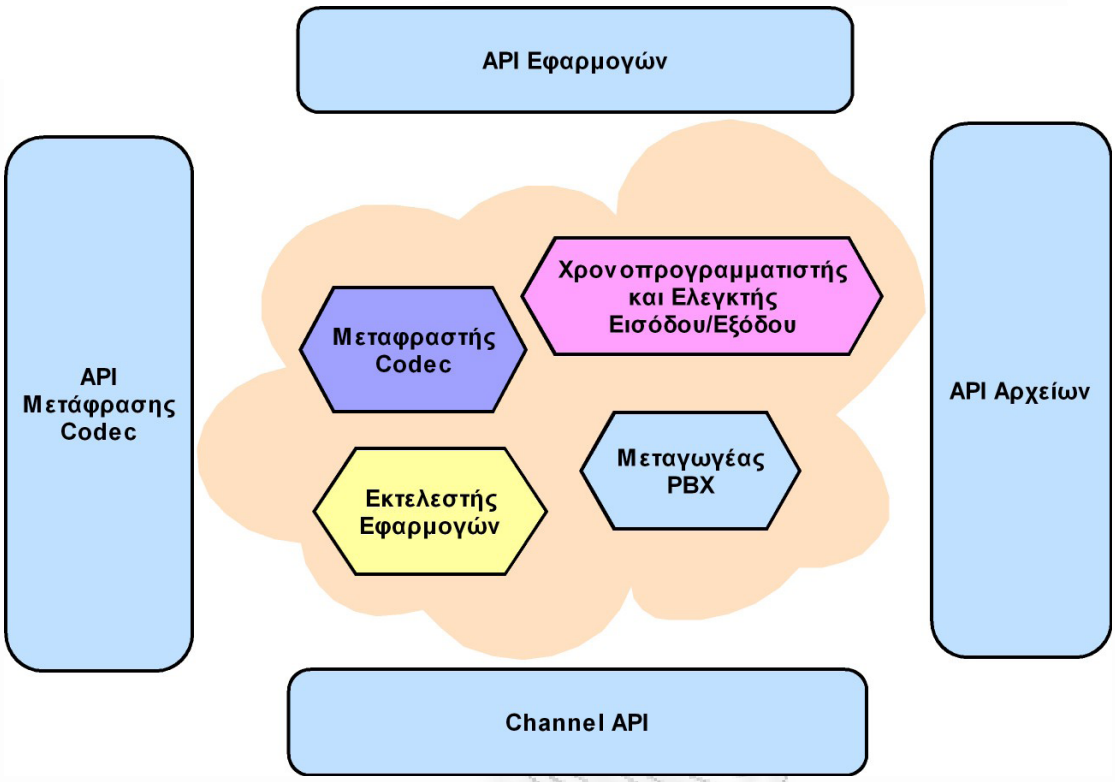
• Channel API: Το channel API διαχειρίζεται τον τύπο της σύνδεσης από την οποία προέρχεται ο χρήστης. Η σύνδεση αυτή μπορεί να είναι VoIP, ISDN, POTS, ή οποιαδήποτε άλλη τεχνολογία. Τα modules φορτώνονται δυναμικά για να χειριστούν τις λεπτομέρειες της σύνδεσης.

• API Εφαρμογών: Το API εφαρμογών, επιτρέπει στα modules εφαρμογών να εκτελεστούν ώστε να παρέχουν διάφορες λειτουργίες. Δυνατότητες όπως τηλεδιάσκεψη, μεταφορά δεδομένων, φωνητικό ταχυδρομείο και οποιαδήποτε άλλη εργασία μπορεί να εκτελέσει ένα σύγχρονο ή μελλοντικό PBX, χειρίζονται από τα αντίστοιχα ξεχωριστά modules.

• API Μετάφρασης Codec: Το API αυτό φορτώνει ξεχωριστά modules για τον κάθε codec, για να υποστηρίξει τους διάφορους τύπους κωδικοποίησης και αποκωδικοποίησης ήχου που υπάρχουν, όπως: GSM, μLaw, aLaw, ακόμα και mp3.

• API Αρχείων: Το API αρχείων είναι υπεύθυνο για την ανάγνωση και εγγραφή πολλών τύπων αρχείων, και την αποθήκευση δεδομένων στο σύστημα.

Με τη χρήση αυτών των APIs, το Asterisk επιτυγχάνει πλήρη ανεξαρτησία μεταξύ της βασικής λειτουργίας του ως PBX και της πληθώρας των τεχνολογιών που υπάρχουν στο χώρο της τηλεφωνίας. Η αρθρωτή δομή του, του επιτρέπει να συνεργάζεται απόλυτα τόσο με τα παραδοσιακά συστήματα, όσο και με τις νέες τεχνολογίες μετάδοσης πακέτων φωνής. Η δυνατότητα που έχει το Asterisk να φορτώνει module για κάθε codec, του επιτρέπει να πραγματοποιεί μετάδοση πακέτων φωνής τόσο σε δίκτυα με μικρό εύρος ζώνης (σύνδεση μέσω modem) με χρήση codec υψηλής συμπίεσης, όσο και σε ευρυζωνικά δίκτυα, παρέχοντας υψηλής ποιότητας ήχο.
Το API εφαρμογών παρέχει τη δυνατότητα στα modules εφαρμογών να εκτελούν οποιαδήποτε λειτουργία ευέλικτα και κατά βούληση του χρήστη. Επιτρέπει επίσης την εκτέλεση εφαρμογών που έχουν αναπτυχθεί αποκλειστικά για να καλύψουν προσαρμοσμένες ανάγκες και περιπτώσεις χρήσης. Επιπλέον, φορτώνοντας όλες τις εφαρμογές ως modules το Asterisk δίνει τη δυνατότητα στους διαχειριστές να σχεδιάσουν (ευπροσάρμοστα) συστήματα με εύκολη προσαρμογή στις τηλεπικοινωνιακές αλλαγές που μπορεί να προκύψουν.



3.3 Λειτουργίες και Δυνατότητες του Asterisk

Όπως αναφέρθηκε προηγουμένως είναι πολύ δύσκολο να περιγραφεί το πλήρες φάσμα των δυνατοτήτων του Asterisk λόγω του πλήθους των περίπλοκων θεμάτων που ενσωματώνει: πολλαπλούς τύπους VoIP καναλιών, υλικά διασύνδεσης, γλώσσα δέσμης ενεργειών (AGI Scripting language), Διασύνδεση Προγράμματος Εφαρμογής (API) και πληθώρα λειτουργιών. Παρακάτω θα αναφερθούμε σε κάποιες από τις λειτουργίες που κάνουν το Asterisk τόσο ισχυρό και σε επόμενα κεφάλαια θα δούμε πως μπορούμε να υλοποιήσουμε μερικές από αυτές στην πράξη και πού θα μπορούσαν να μας χρησιμεύσουν.

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

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

• *Automated Attendant*: Επιτρέπει σε κάποιον να πληκτρολογήσει έναν κεντρικό αριθμό και στη συνέχεια να πληκτρολογήσει τον κωδικό κάποιας υπηρεσίας ή κάποιας extension. Μπορεί να χρησιμοποιηθεί σε συνδυασμό με το Dial by Name (βλ. παρακάτω) για να παρέχει π.χ. τη δυνατότητα κλήσης με χρήση ονόματος.

• *Blacklists*: Δημιουργία μαύρης λίστας εισερχομένων κλήσεων (συνήθως με χρήση caller id) και ξεχωριστή διαχείριση της ανάλογα με προσαρμοσμένους κανόνες.

• *Call Detail Records*: Αρχείο καταγραφής κλήσεων με στοιχεία όπως η ώρα έναρξη της κλήσης, η διάρκεια της κλήσης, το νούμερο του καλούντα, την κατάσταση της κλήσης, κ.α.

• *Call Forward*: Προώθηση κλήσεων κατά βούληση ή ανάλογα με την κατάσταση (Κατειλημμένο, Δεν απαντά, κ.λ.π.)

• *Call Monitoring*: Παρακολούθηση κλήσεων σε πραγματικό χρόνο ή καταγραφή τους για διασφάλιση ποιότητας υπηρεσιών.

• *Call Parking*: Στάθμευση της κλήσης σε ένα εικονικό νούμερο το οποίο χρησιμοποιείται σαν χώρος στάθμευσης των κλήσεων και επανάκτηση της κλήσης κατά βούληση

• *Call Queuing*: Ουρές αναμονής κλήσεων με δυνατότητα αναπαραγωγής μουσικής ή ανακοινώσεων κατά τη διάρκεια αναμονής.

• *Call Recording*: Ηχογράφηση κλήσεων σε πραγματικό χρόνο.

• *Call Transfer*: Μεταφορά κλήσεων από ένα νούμερο σε ένα άλλο.

• *Call Waiting*: Αναμονή κλήσεων με δυνατότητα αναγνώρισης κλήσης της δεύτερης γραμμής, μουσική κατά τη διάρκεια της αναμονής και προώθηση κλήσης που βρίσκεται στην αναμονή.

• *Caller ID*: Αναγνώριση κλήσης με στοιχεία το νούμερο και το όνομα του καλούντα (αν είναι διαθέσιμα).

• *Calling Cards*: Δυνατότητα παροχής υπηρεσιών τηλεφωνίας με χρήση προπληρωμένων καρτών ή γενικότερα προπληρωμένων λογαριασμών.

• *Dial by Name*: Δυνατότητα κλήσης με χρήση ονόματος αντί για νούμερο.

• *Direct Inward System Access*: Δυνατότητα απομακρυσμένης σύνδεσης σε λειτουργίες που είναι διαθέσιμες μόνο σε τοπικές extensions.

• *Distinctive Ring*: Δυνατότητα αλλαγής του ρυθμού κουδουνίσματος του τηλεφώνου.

• *Distributed Universal Number Discovery (DUNDI)*: Χρήση του DUNDI γιαεύρεση τηλεφώνου μέσω ερώτησης σε κάποιον γνωστό μας σύνδεσμο.

• *ENUM*: Χρήση του ENUM για ενοποίηση του τηλεφωνικού συστήματος αριθμοδότησης (E.164) με το σύστημα διευθυνσιοδότησης του διαδικτύου (DNS) και έμμεση αναζήτηση.

• *Fax Transmit and Receive*: Αποστολή/Λήψη φαξ και προώθηση στο email.

• *Flexible Extension Logic*: Ευέλικτη και παραμετροποιήσιμη αριθμοδότηση και διαχείριση των κλήσεων.

• *Macros*: Αυτόματη εκτέλεση πολύπλοκων πολλαπλών ενεργειών που εκτελούνται συχνά για εξοικονόμηση χρόνου και αποφυγή λαθών.

• *Predictive Dialler*: Αυτόματη κλήση σε τηλεφωνικά νούμερα. Χρησιμοποιείται σε τηλεφωνικά κέντρα (τηλε-μάρκετινγκ) και πραγματοποιεί κλήσεις προς πιθανούς πελάτες με χρήση εξειδικευμένων αλγορίθμων πρόβλεψης.

• *Open Settlement Protocol (OSP)*: Δυνατότητα τιμολόγησης VoIP υπηρεσιών.

• *Roaming Extensions*: Δυνατότητα περιαγωγής της extension σε οποιοδήποτε σημείο του κόσμου με πρόσβαση σε τηλεφωνικό δίκτυο ή στο internet.

• *Call Routing*: Δρομολόγηση της κλήσης ανάλογα με τον αριθμό αυτού που καλεί, την ώρα κλήσης, το κόστος κλήσης, κ.α.

• *SMS Messaging*: Αποστολή γραπτών μηνυμάτων

• *Streaming Media Access*: Δυνατότητα βιντεοκλήσης.

• *Talk Detection*: Αναγνώριση ομιλίας με χρήση του sphinx.

• *Text-to-Speech*: Εκφώνηση κειμένου μέσω του Festival, Cepstral κ.α.

• *VoIP Gateway*: Δυνατότητα διασύνδεσης VoIP τερματικών ανεξαρτήτως πρωτοκόλλου που χρησιμοποιεί το καθένα και μετατροπή των μη συμβατών πρωτοκόλλων.

• *Voicemail*: Φωνητικό ταχυδρομείο με δυνατότητες ειδοποίησης νέων μηνυμάτων μέσω email, αποστολής του μηνύματος ως προσάρτηση σε email, οπτικής απεικόνισης νέων μηνυμάτων στα τερματικά, οργάνωσης σε φακέλους, ομαδικής αποστολής και απομακρυσμένης διαχείρισης.

• *Zapateller*: Χρήση ειδικού ήχου (*Special Information Tone*) για αποφυγή τηλεφωνημάτων από αυτόματες κλήσεις π.χ. τηλε-μάρκετινγκ.

• *AGI (Asterisk Gateway Interface)*: Δυνατότητα αλληλεπίδρασης εξωτερικών προγραμμάτων με το Asterisk. Πλήρης διαχείριση του συστήματος σε επίπεδο λειτουργιών και κονσόλας.

• *Echo cancellation*: Δυνατότητα εξάλειψης του φαινόμενου της ηχώ με χρήση εξελιγμένων αλγορίθμων.

• *Codecs*: Υποστήριξη πληθώρας μεθόδων κωδικο-αποκωδικοποίησης και συμπίεσης όπως: ADPCM, G.711 (A-Law & μ-Law), G.722, G.723.1 (pass through), G.726, G.729(με αγορά άδειας χρήσης), GSM, iLBC, Linear, LPC-10, Speex.

• *Protocols*: Υποστήριξη πληθώρας πρωτοκόλλων όπως: IAX(Inter-Asterisk Exchange), H.323, SIP (Session Initiation Protocol), MGCP (Media Gateway Control Protocol, SCCP (Cisco Skinny).

3.4 Παραδείγματα χρήσης πρωτοκόλλου SIP στη πλατφόρμα Asterisk

Αφού έχουμε εγκαταστήσει και εκκινήσει τη λειτουργία του Asterisk, μπορούμε να αρχίσουμε να το διαμορφώνουμε έτσι ώστε να του προσθέσουμε λειτουργίες. Η ακριβής διαδικασία της διαμόρφωσης του Asterisk εξαρτάται κάθε φορά από τις λειτουργίες που επιθυμούμε να παρέχει το τηλεφωνικό μας σύστημα στους χρήστες του. Όπως ισχύει σε αρκετά UNIX προγράμματα, έτσι και το Asterisk διαμορφώνεται μέσω πολλών αρχείων (.conf) που συνεργάζονται εσωτερικά μεταξύ τους. Τα .conf αρχεία του Asterisk βρίσκονται στο φάκελο /etc/asterisk

Τα αρχεία αυτά είναι:

```
# cd /etc/asterisk
# ls
adsi.conf           cdr_tds.conf     indications.conf  privacy.conf
adtranvofr.conf     codecs.conf      logger.conf       queues.conf
agents.conf         dnsmgr.conf      manager.conf      res_odbc.conf
alarmreceiver.conf  dundi.conf       meetme.conf       rpt.conf
alsa.conf           enum.conf        mgcp.conf         rtp.conf
asterisk.adsi       extconfig.conf   misdn.conf        sip.conf
asterisk.conf       extensions.ael   modem.conf        sip_notify.conf
cdr.conf            extensions.conf  modules.conf      skinny.conf
cdr_custom.conf     features.conf    musiconhold.conf  telcordia-1.adsi
cdr_manager.conf    festival.conf    osp.conf          voicemail.conf
cdr_odbc.conf       iax.conf         oss.conf          vpb.conf
cdr_pgsql.conf      iaxprov.conf     phone.conf        zapata.conf
```

Όλες οι λειτουργίες του Asterisk καθορίζονται μέσα σε αυτά τα αρχεία με μοναδική εξαίρεση να αποτελεί το αρχείο zaptel.conf που βρίσκεται στο φάκελο /etc και παρέχει ρυθμίσεις για το υλικό Zaptel.

Αρχικές ενέργειες

Τα παραπάνω πρότυπα .conf αρχεία δημιουργήθηκαν με την εντολή make samples κατά την εγκατάσταση του πακέτου asterisk [2]. Τα σχόλια που περιέχονται σε αυτά τα αρχεία καθώς επίσης και τα σενάρια χρήσης που υλοποιούνται, θα μας χρησιμεύσουν ως σημείο αναφοράς για την πληθώρα των ρυθμίσεων που μπορούμε να υλοποιήσουμε στο Asterisk. Επειδή στα παραδείγματα που θα αναπτύξουμε παρακάτω, θα αναλύσουμε κάποιες έννοιες και λειτουργίες του Asterisk, οι οποίες θα πρέπει να αναπτυχθούν απ' την αρχή, είναι καλή πρακτική να κρατήσουμε αντίτυπα των αρχείων που θα αλλάζουμε στην πορεία. Για το λόγο αυτό θα δημιουργήσουμε ένα φάκελο /backup όπου θα αντιγράψουμε όλα τα πρότυπα αρχεία.

```
# cd /etc/asterisk
# mkdir backup
# cp * backup/
```

Το πιο απλό PBX

Για να κατανοήσουμε στην πράξη το πως ακριβώς συνδέονται τα .conf αρχεία και πώς συντάσσονται θα δημιουργήσουμε μία απλή υλοποίηση ενός PBX γενικής χρήσεως. Στο συγκεκριμένο παράδειγμα θα διαμορφώσουμε το Asterisk και δύο softphones έτσι ώστε να μπορέσουν να μιλήσουν μεταξύ τους, μέσω του Asterisk. Τα softphones που επιλέχτηκαν είναι το kiax και το x-lite. Το kiax χρησιμοποιεί το πρωτόκολλο IAX2 ενώ το x-lite χρησιμοποιεί το SIP, το Asterisk χειρίζεται και τα δύο αυτά πρωτόκολλα λειτουργώντας ως VoIP Gateway μεταξύ των δύο.

Τα αρχεία με τα οποία θα ασχοληθούμε είναι τα sip.conf, iax.conf και extensions.conf. Στα αρχεία sip.conf και iax.conf θα ορίσουμε τους χρήστες που θα επιτρέπουμε να συνδεθούν στο Asterisk. Στη δική μας περίπτωση δηλαδή, στο αρχείο sip.conf θα ορίσουμε το χρήστη sip-user και στο iax.conf θα ορίσουμε το χρήστη iax-user. Τέλος, στο αρχείο extensions.conf θα ορίσουμε τον τρόπο με τον οποίο θα συνδεθούν οι χρήστες.

sip.conf

Διαμορφώνουμε το αρχείο etc/asterisk/sip.conf με κάποιον επεξεργαστή κειμένου της επιλογής μας έτσι ώστε να περιέχει τις παρακάτω γραμμές:

```
[general]
bindport=5060
bindaddr=0.0.0.0
context=default

[sip-user]
type=friend
secret=unipi
callerid=”sip-user” <111>
qualify=yes
nat=yes
host=dynamic
canreinvite=no
context=mini-pbx-internal
```

Το αρχείο ξεκινάει με το γενικό περιεχόμενο [general], το οποίο περιέχει ρυθμίσεις του καναλιού και προκαθορισμένες επιλογές για όλους τους χρήστες που ορίζονται στο αρχείο. Οι ρυθμίσεις αυτές μπορούν να παρακαμφθούν ανά χρήστη, αρκεί να δηλωθεί στον ορισμό του χρήστη. Συγκεκριμένα στο context [general], ορίσαμε τον αριθμό θύρας για τις SIP συνδέσεις στο 5060 (προεπιλεγμένη τιμή). Η γραμμή bindaddr=0.0.0.0 λέει στο Asterisk να ακούει για SIP συνδέσεις και να τις επιτρέπει σε όλες τις διευθύνσεις IP που διαθέτει το σύστημά μας. Το context (περιεχόμενο) ενός χρήστη ορίζεται με το όνομα του μέσα σε τετραγωνικά άγκιστρα [ ] και τελειώνει εκεί που ξεκινάει το επόμενο. Η έννοια του context είναι αρκετά σημαντική για τη λειτουργία του Asterisk. Στο παράδειγμα του mini-PBX θα μπορούσαμε να πούμε ότι το [sip-user] αντιστοιχεί με το όνομα του χρήστη του οποίου θα ορίσουμε τις παραμέτρους.

Ο τύπος του χρήστη (type=friend) μας λέει ότι ο sip-user μπορεί να δεχθεί και να πραγματοποιήσει κλήσεις. Αν θέλουμε ο χρήστης μόνο να δέχεται κλήσεις τότε αλλάζουμε τον τύπο του σε peer (type=peer) και αν θέλουμε μόνο να πραγματοποιεί σε type=user.

Η γραμμή secret=unipi μας δείχνει ότι το συνθηματικό που θα χρησιμοποιηθεί για την αυθεντικοποίηση του χρήστη sip-user, είναι η λέξη unipi. Στο callerid βάζουμε το όνομα που επιθυμούμε οι άλλοι να βλέπουμε όταν τους καλούμε. Θεωρείται καλό να περιέχεται και το extension (π.χ. <111>) του χρήστη έκτος του ονόματός του, για ευκολότερη αναγνώριση.

Με την επιλογή qualify=yes μπορούμε να παρακολουθήσουμε την καθυστέρηση της επικοινωνίας και να καταλάβουμε αν το άλλο άκρο είναι προσβάσιμο. Η προεπιλεγμένη τιμή είναι τα 2000 msec, αλλά μπορούμε να την αλλάξουμε αντικαθιστώντας το yes με κάποια αριθμητική τιμή σε msec (π.χ. qualify=3000). Αν ο χρήστης είναι πίσω από κάποια συσκευή NAT (Network Address Translation) όπως firewall ή router, η επιλογή αυτή λέει στο Asterisk να αγνοήσει τις πληροφορίες που δέχεται μέσα από το κανάλι της κλήσης (π.χ. source ip) και να χρησιμοποιήσει για επικοινωνία, τη διεύθυνση από την οποία λαμβάνει τα δεδομένα.

Στη γραμμή host=dynamic λέμε στο Asterisk ότι τα τερματικά μας, δεν έχουν σταθερή διεύθυνση IP και ότι χρειάζεται να τη μαθαίνει κάθε φορά που εγγραφόμαστε σε αυτόν. Εναλλακτικά μπορούμε να ορίσουμε μία στατική IP στο χρήση π.χ. host=192.168.1.7 ή ακόμη και κάποιο domain name.

Η γραμμή canreinvite=no λέει στο Asterisk να μην επιτρέπει στους χρήστες να συνδεθούν απευθείας μεταξύ τους, αλλά πάντα να παρεμβάλεται στην επικοινωνία.

Τέλος, το context=mini-pbx-internal αναφέρεται στη θέση των οδηγιών του αρχείου extensions.conf που θα χρησιμοποιηθούν για τον έλεγχο των εισερχόμενων και εξερχόμενων κλήσεων του συγκεκριμένου χρήστη ή ακόμη σωστότερα της συγκεκριμένης extension. Για το συγκεκριμένο χρήστη, παρακάμψαμε τη γενική τιμή του context=default που θέσαμε στο generals και ορίσαμε κάποια νέα που θα ισχύσει για το συγκεκριμένο χρήστη.

iax.conf

Παρακάτω ακολουθεί η διαμόρφωση του αρχείου iax.conf όπου ορίζεται ο χρήστης iax-user. Με κάποιον επεξεργαστή κειμένου, διαμορφώνουμε το αρχείο /etc/asterisk/iax.conf ώστε να περιέχει τα παρακάτω:

```
[general]
bandwidth=low
context=default
[iax-user]
type=friend
secret=unipi
callerid=”iax-user” <222>
qualify=yes
host=dynamic
context=mini-pbx-internal
```

Παρόλο που το IAX και το SIP έχουν σημαντικές διαφορές στον τρόπο υλοποίησης τους, βλέπουμε ότι στο συγκεκριμένο παράδειγμα οι χρήστες ορίζονται παρόμοια. Μία σημαντική διαφορά που φαίνεται αμέσως είναι η έλλειψη της γραμμής nat. Αυτό συμβαίνει επειδή το πρωτόκολλο IAX έχει σχεδιαστεί με τέτοιο τρόπο ώστε να μην επηρεάζεται σοβαρά από την τοπολογία του δικτύου κάνοντας έτσι πιο εύκολη τη διαδικασία της εγγραφής του πελάτη στον εξυπηρετητή. Άλλη μία διαφορά που προκύπτει από τα παραπάνω αρχεία είναι η επιλογή bandwidth=low που περιέχεται στο [general] context. Η επιλογή αυτή αφορά τους codecs που θα χρησιμοποιηθούν στην επικοινωνία. Επιβάλουμε δηλαδή στο Asterisk να μη χρησιμοποιήσει codecs που χρειάζονται μεγάλο εύρος ζώνης. Άλλες διαθέσιμες επιλογές είναι οι: medium και high. Αφού τελειώσουμε την επεξεργασία των δύο παραπάνω αρχείων, θα πρέπει να φορτώσουμε τις καινούργιες ρυθμίσεις στο Asterisk. Αυτό επιτυγχάνεται μέσω της εντολής reload στην κονσόλα του Asterisk.

```
*CLI> reload
```

Το Asterisk είναι πλέον έτοιμο να επιτρέψει στους δύο πελάτες να συνδεθούν.

Διαμόρφωση των προγραμμάτων πελατών

Για τις ανάγκες του παραδείγματος χρησιμοποιήθηκαν ενδεικτικά δύο softphones από τα πολλά διαθέσιμα δωρεάν στο διαδίκτυο. Θα πρέπει να τα παραμετροποιήσουμε έτσι ώστε να μπορούν να συνδεθούν με το Asterisk.

Διαμόρφωση x-lite

Το x-lite είναι ένα SIP softphone για το λειτουργικό MS Windows και είναι διαθέσιμο από την ιστοσελίδα της counterpath (http://www.counterpath.com). Αφού κατεβάσουμε και εγκαταστήσουμε το x-lite, προχωράμε στη διαμόρφωση του προγράμματος με τις απαραίτητες ρυθμίσεις. Με δεξί click πάνω στην επιφάνεια του προγράμματος και επιλογή του “Sip Accounts Settings…” βλέπουμε το παράθυρο διαχείρισης των SIP λογαριασμών μας. Επιλέγουμε το “Add…” και συμπληρώνουμε τα πεδία:
User name: *sip-user*
Password: *unipi*
Domain: *Η IP που βρίσκεται το Asterisk (local ή external) ή το πλήρες Domain Name, για παράδειγμα 192.168.1.5. Αν δεν γνωρίζουμε την IP του Asterisk server μας, μπορούμε να γράψουμε στη γραμμή εντολών του λειτουργικού την εντολή* ifconfig. Αφήνουμε όλα τα υπόλοιπα πεδία κενά και πατάμε OK. Βεβαιωνόμαστε ότι ο λογαριασμός είναι ενεργοποιημένος (enabled tickbox) και κλείνουμε το παράθυρο διαχείρισης των SIP λογαριασμών.
Μετά από λίγη ώρα θα πρέπει να δούμε στο x-lite το μήνυμα:

Ready

Your username is: sip-user

όπως φαίνεται στην παρακάτω εικόνα.

Διαμόρφωση kiax

Το kiax είναι ένα softphone που χρησιμοποιεί το πρωτόκολλο IAX2 για να πραγματοποιεί VoIP κλήσεις. Είναι διαθέσιμο δωρεάν (άδεια χρήσης GPL) για τα λειτουργικά συστήματα Ms Windows, Linux, FreeBSD, netBSD και μπορούμε να το βρούμε στην ιστοσελίδα http://sourceforge.net/projects/kiax. Αφού κατεβάσουμε το πρόγραμμα και το εγκαταστήσουμε προχωράμε στην παραμετροποίηση του. Στην κεντρική οθόνη του προγράμματος επιλέγουμε File→Settings και στο παράθυρο που μας εμφανίστηκε πατάμε το κουμπί New Account και συμπληρώνουμε τα πεδία:

Account Name: *iax-user*

IAX Server: *Η IP που βρίσκεται το Asterisk (local ή external) ή το πλήρες Domain Name, για παράδειγμα 192.168.1.5.*

Username: *iax-user*

Password: *unipi*

CallerID Name: *<κενό>*

CallerID Number: *<κενό>*

Πατάμε το κουμπί save&close και μετά από λίγη ώρα θα πρέπει να δούμε το μήνυμα Registered όπως φαίνεται στην παρακάτω εικόνα.

extensions.conf

Μέχρι στιγμής τα δύο τερματικά μας μπορούν να εγγραφούν στο Asterisk. Δεν μπορούν ακόμα όμως ούτε να δεχθούν, ούτε να πραγματοποιήσουν κλήσεις. Ο χειρισμός και η δρομολόγηση των κλήσεων που περνάνε από το Asterisk

υλοποιείται στο αρχείο etc/asterisk/extensions.conf. Για συνδεθούν οι δύο πελάτες μεταξύ τους μέσω του Asterisk διαμορφώνουμε το αρχείο extensions.conf έτσι ώστε να περιέχει μόνο τις παρακάτω γραμμές:

```
[mini-pbx-internal]
exten => 111,1,Dial(SIP/sip-user)
exten => 222,1,Dial(IAX2/iax-user)
```

Για να διαβάσει το Asterisk τις αλλαγές μας, γράφουμε στην κονσόλα του, την εντολή reload.

```
*CLI> reload
```

Το απλό μας PBX είναι πλέον έτοιμο και λειτουργικό μέσα σε λίγες μονάχα γραμμές. Οι αλλαγές που κάναμε στο extensions.conf είναι στοιχειώδεις, και ενω μας δίνουν ένα λειτουργικό σύστημα δεν παρέχουν κανέναν μηχανισμό χειρισμού των κλήσεων, αλλά ούτε και σύνδεση με το δημόσιο τηλεφωνικό δίκτυο. Παρόλα αυτά, για λόγους ευκολίας θα παραμείνουμε σε αυτήν την υλοποίηση.

Στο αρχείο extensions.conf λοιπόν, δημιουργήσαμε το context [mini-pbxinternal], που χρησιμοποιήθηκε στον ορισμό των πελατών sip-user (στο sip.conf) και iax-user (στο iax.conf). Μέσα στο context [mini-pbx-internal], ορίσαμε δύο extensions, την 111 και την 222. Όταν κάποιος πελάτης συνδεδεμένος στο Asterisk με πρόσβαση στο [mini-pbx-internal] πληκτρολογήσει 111, τότε η εφαρμογή Dial(), θα καλέσει τον πελάτη sip-user χρησιμοποιώντας τον ορισμό του στο αρχείο sip.conf. Με την ίδια λογική, όταν κάποιος πελάτης συνδεδεμένος στο Asterisk πληκτρολογήσει 222, τότε μέσω της Dial(), θα κληθεί ο πελάτης iax-user που έχει οριστεί στο αρχείο iax.conf. Μπορούμε να παρατηρήσουμε ότι στην Dial() υπάρχει το πρωτόκολλο IAX2 και όχι IAX όπως το αναφέρουμε έως τώρα. Η σωστή ονομασία είναι η IAX2 αφού

βρίσκεται πλέον στη δεύτερη έκδοσή του. Παρόλα αυτά, για λόγους συντομίας, στη βιβλιογραφία αναφέρεται ως IAX.

Γενικά το Asterisk ελέγχει σε ποιο context (π.χ. [mini-pbx-internal]) ανήκει αυτός που πραγματοποιεί την κλήση από το .conf (SIP, IAX, κ.λ.π.) αρχείο που αυτός έχει οριστεί και χρησιμοποιεί αυτό το context για να καταλάβει ποιους κανόνες θα ακολουθήσει στο αρχείο extensions.conf. Τα παραπάνω φαίνονται και στην πράξη στις εικόνες που ακολουθούν

Κλήση: sip-user → iax-user

```
   -- Executing Dial("SIP/sip-user-0818f810", "IAX2/iax-user") in new stack
   -- Called iax-user
   -- Call accepted by 192.168.1.33 (format gsm)
   -- Format for call is gsm
   -- IAX2/iax-user-3 is ringing
   -- IAX2/iax-user-3 answered SIP/sip-user-0818f810
   -- Hungup 'IAX2/iax-user-3'
 == Spawn extension (mini-pbx-internal, 222, 1) exited non-zero on 'SIP/sip-user-0818f810'
```

Κλήση: iax-user → sip-user

```
   -- Accepting AUTHENTICATED call from 192.168.1.33:
      > requested format = gsm,
      > requested prefs = (),
      > actual format = gsm,
      > host prefs = (),
      > priority = mine
   -- Executing Dial("IAX2/iax-user-3", "SIP/sip-user") in new stack
   -- Called sip-user
   -- SIP/sip-user-08194ef0 is ringing
   -- SIP/sip-user-08194ef0 answered IAX2/iax-user-3
 == Spawn extension (mini-pbx-internal, 111, 1) exited non-zero on 'IAX2/iax-user-3'
   -- Hungup 'IAX2/iax-user-3'
```

Παράδειγμα κλήσης video και ήχου μεταξύ δύο χρηστών

Η χρησιμότητα του Asterisk είναι ότι μπορούμε να ξεκινήσουμε μια σύνοδο SIP μεταξύ δυο χρηστών που να περιλαμβάνει εικόνα και ήχο με πολύ απλό τρόπο. Όπως και παραπάνω, αρκεί να διαμορφώσουμε κατάλληλα τα αρχεία sip.conf και extensions.conf [4]

Sip.conf

Διαμορφώνουμε το αρχείο etc/asterisk/sip.conf με κάποιον επεξεργαστή κειμένου της επιλογής μας έτσι ώστε να περιέχει τις παρακάτω γραμμές:

[general]

context=default

srvlookup=yes

videosupport=yes

[101]

type=friend

secret=101

qualify=yes

nat=yes

host=dynamic

canreinvite=no

context=internal

context=internal

disallow=all

allow=alaw

allow=ulaw

allow=gsm

allow=h263

allow=h263p

[102]

type=friend

secret=102

qualify=yes

nat=yes

host=dynamic

canreinvite=no

context=internal

context=internal

disallow=all

allow=alaw

allow=ulaw

allow=gsm

allow=h263

allow=h263p

Με την εντολή “disallow=all” απενεργοποιούμε όλους τους διαθέσιμους codecs έτσι ώστε στις επόμενες γραμμές να ενεργοποιήσουμε αυτούς που θέλουμε.

Οι codecs alaw, ulaw, gsm είναι για φωνή, ενώ οι h263, h263p είναι για βίντεο.

Αυτή τη φορά δημιουργήσαμε δυο χρήστες, τους 101 και 102. Στη θέση αυτών των ονομάτων θα μπορούσε να είναι οποιοδήποτε αλφαριθμητικό.

Να τονίσουμε εδώ ότι και οι δυο χρήστες θα χρησιμοποιήσουν το πρωτόκολλο SIP, γι’ αυτό άλλωστε ορίζονται μέσα στο αρχείο sip.conf

Διαμόρφωση των προγραμμάτων πελατών

Και για τους δύο χρήστες, θα χρησιμοποιήσουμε ένα εμπορικό πρόγραμμα με πολλές δυνατότητες. Είναι το eyeBeam

(http://www.inphonex.com/support/eyebeam-configuration.php)

Αφού το εγκαταστήσουμε, προχωράμε στη διαμόρφωση του προγράμματος με τις απαραίτητες ρυθμίσεις. Με δεξί click πάνω στην επιφάνεια του προγράμματος και επιλογή του “Sip Accounts Settings…” βλέπουμε το παράθυρο διαχείρισης των SIP λογαριασμών μας. Επιλέγουμε το “Add…” και συμπληρώνουμε τα πεδία:

User name: *101*

Password: *101*

Domain: *Η IP που βρίσκεται το Asterisk (local ή external) :*

192.168.199.128

Αφήνουμε όλα τα υπόλοιπα πεδία κενά και πατάμε OK. Βεβαιωνόμαστε ότι ο λογαριασμός είναι ενεργοποιημένος (enabled tickbox) και κλείνουμε το παράθυρο διαχείρισης των SIP λογαριασμών. Μετά από λίγη ώρα θα πρέπει να δούμε στο eyeBeam το μήνυμα:

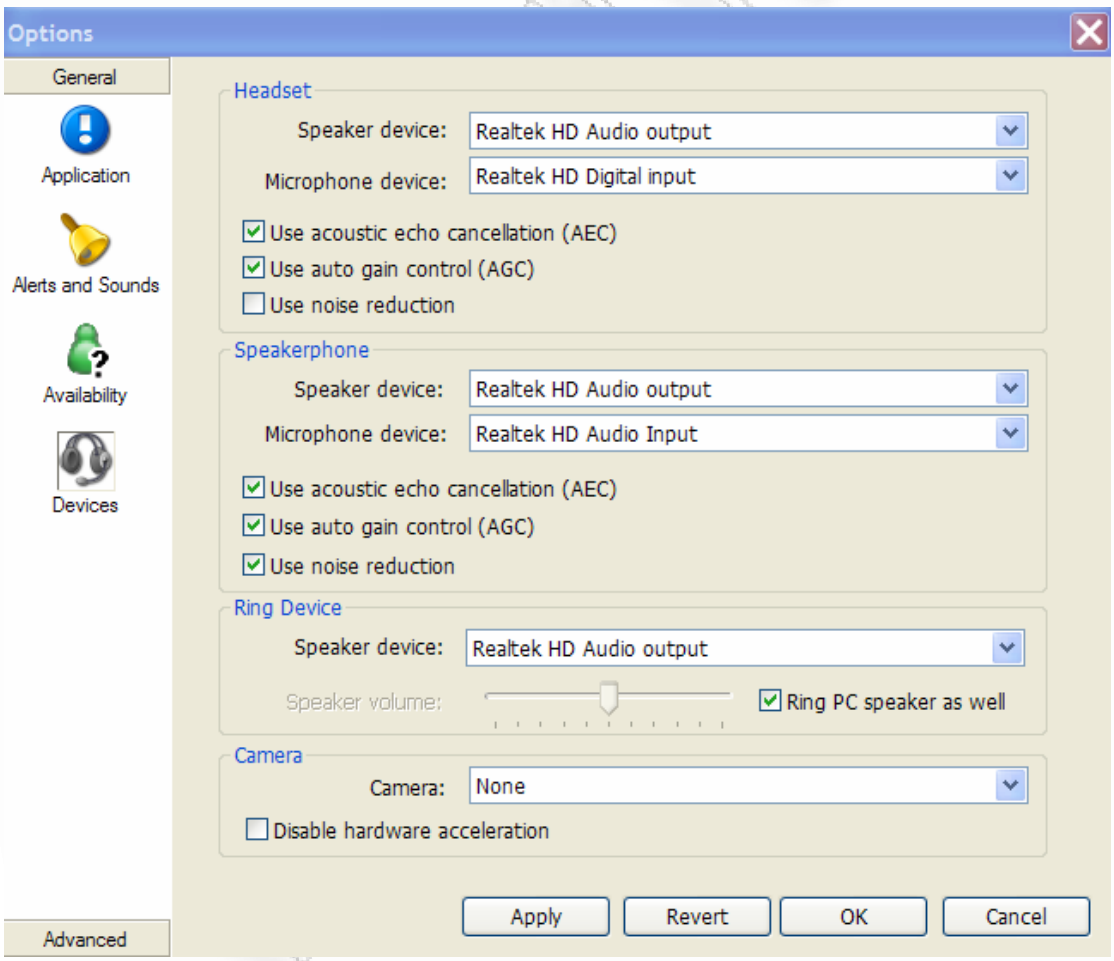
Ready

Your username is: 101

Το επόμενο βήμα είναι να ρυθμίσουμε τις οπτικοακουστικές συσκευές που θα χρησιμοποιήσει το eyeBeam.

Πάμε Options -> Devices

Σε αυτή τη καρτέλα μπορούμε να δούμε τις συσκευές audio-video του συστήματος μας. Στο παράδειγμα μας ενεργοποιούμε το μικρόφωνο (Realtek HD Digital Input) και την κάμερα (Creative WebCam NX Pro) και πατάμε OK.



extensions.conf

Ο χειρισμός και η δρομολόγηση των κλήσεων όπως αναφέραμε και παραπάνω πραγματοποιείται μέσω του extensions.conf. Για να συνδεθούν οι δύο πελάτες μεταξύ τους μέσω του Asterisk διαμορφώνουμε το αρχείο extensions.conf έτσι ώστε να περιέχει μόνο τις παρακάτω γραμμές:

```
[internal]
exten => 101,1,Dial(SIP/101,10)
exten => 102,1,Dial(SIP/102,10)
```

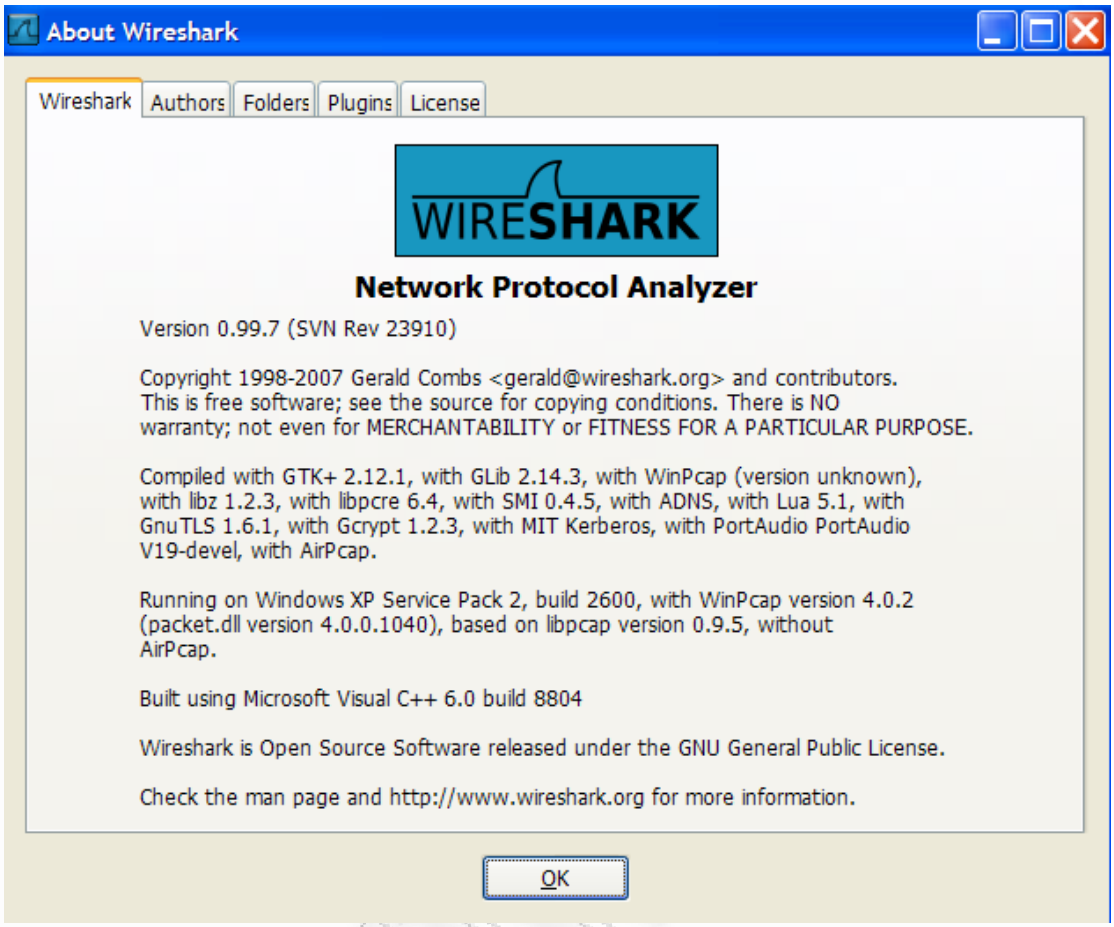
Η λογική είναι όπως στο προηγούμενο παράδειγμα. Και για τους δυο χρήστες ορίζονται οι αριθμοί που μπορούν να πάρουν μέσω του context με ονομασία “internal”. Είναι προφανές ότι μπορούν να πάρουν στη συγκεκριμένη περίπτωση μόνο ο ένας τον άλλον.
Το “10” που τίθεται ως παράμετρος είναι τα δευτερόλεπτα που θα περιμένει το Asterisk για να δεχθεί ο καλούμενος την κλήση. Μετά το πέρας των δέκα δευτερολέπτων η προσπάθεια έναρξης συνόδου σταματά.

Τα soft phones τώρα είναι έτοιμα για κλήση. Καλούμε από το soft phone του χρήστη 101 τον χρήστη 102 και το τηλέφωνο του “χτυπάει”. Ο 102 πατώντας το πράσινο κουμπί δέχεται την κλήση και η συνομιλία με video και ήχο ξεκινάει κανονικά.

Ανάλυση πακέτων σε μια σύνοδο SIP

Είναι ιδιαίτερα σημαντικό να δούμε στη πράξη τί γίνεται σε μία σύνοδο SIP που πραγματοποιείται μέσω της πλατφόρμας Asterisk. Με αυτό το τρόπο θα επιβεβαιώσουμε τα πακέτα που θεωρητικά είπαμε πως ανταλλάσουν τα δυο end σε μια σύνοδο.

Για αυτή την εργασία θα χρησιμοποιήσουμε το πρόγραμμα Wireshark. Είναι ένα ελεύθερο λογισμικό με απεριόριστες δυνατότητες ανάλυσης δικτύων.

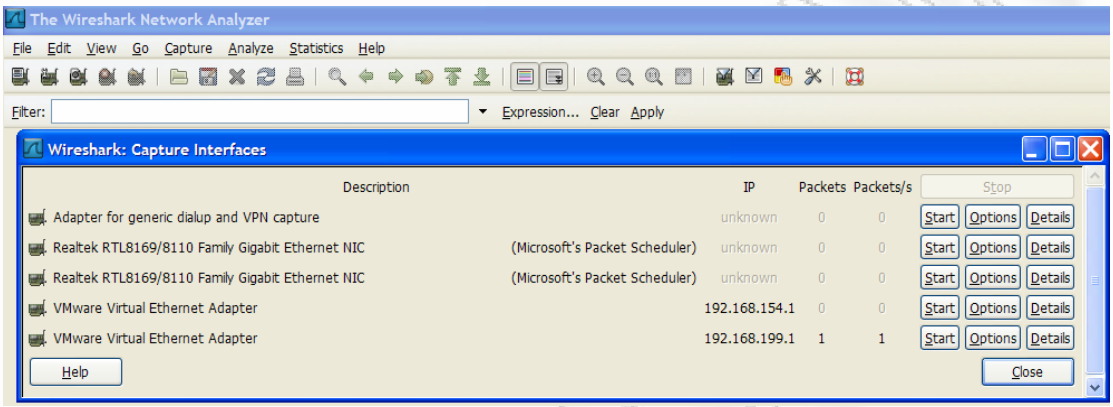


Η διαδικασία του capture και της ανάλυσης των πακέτων που ανταλλάσσονται είναι αρκετά απλή μέσω του Wireshark.

Πρώτα απ' όλα έχουμε ετοιμάσει τα δυο softphone και τα έχουμε κάνει register στο Asterisk με τον τρόπο που περιγράψαμε στο προηγούμενο παράδειγμα.

Να τονίσουμε εδώ, ότι η όλη διαδικασία για αυτό το παράδειγμα έγινε σε έναν μόνο υπολογιστή. Το Asterisk έτρεξε μέσω VMware σε περιβάλλον Fedora και τα δυο softphone στα WindowsXP. Γι' αυτό τον λόγο άλλωστε παρακάτω θα φανεί ότι τα softphone βρίσκονται στην ίδια IP αλλά σε διαφορετική θύρα.

Ανοίγουμε αρχικά το Wireshark και πάμε στη καρτέλα “Capture” επιλέγοντας “Interface”.

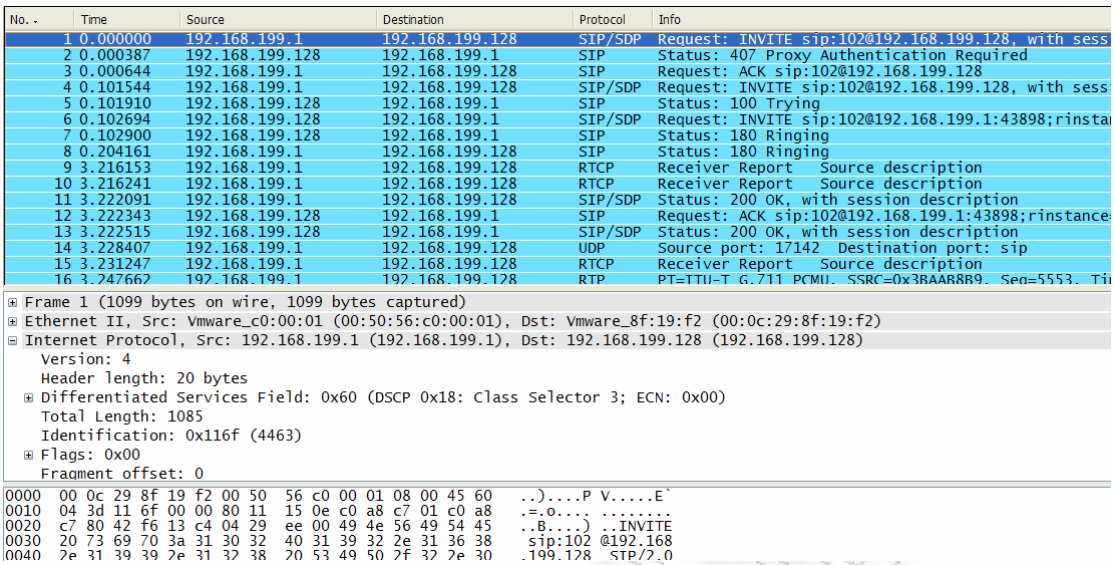


Η IP της εικονικής μηχανής οπού είναι εγκατεστημένος ο Asterisk είναι 192.168.199.1

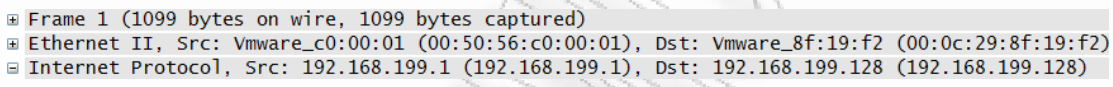
Επιλεγούμε Start και το Wireshark αρχίζει να “ακούει” πακέτα στη συγκεκριμένη διεύθυνση.

Έπειτα πραγματοποιούμε κανονικά την κλήση μεταξύ των χρηστών 101 και 102 ακριβώς όπως περιγράψαμε στο προηγούμενο παράδειγμα. Οι χρήστες συνομιλούν και κάποια στιγμή ο 102 κλείνει και αυτό σημαίνει την λήξη της συνόδου.

Αυτόματα κατά την διάρκεια της συνόδου το Wireshark καταγράφει τα πακέτα και τα εμφανίζει στο κεντρικό παράθυρο όπως παρακάτω:

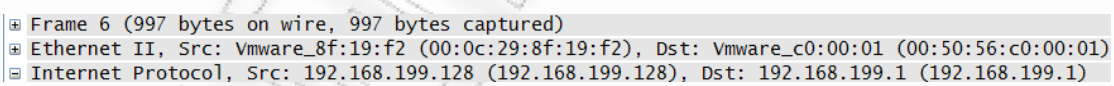


Στη συγκεκριμένη εικόνα έχουμε επιλέξει να παρουσιάσουμε μόνο το πρώτο frame (No.1)



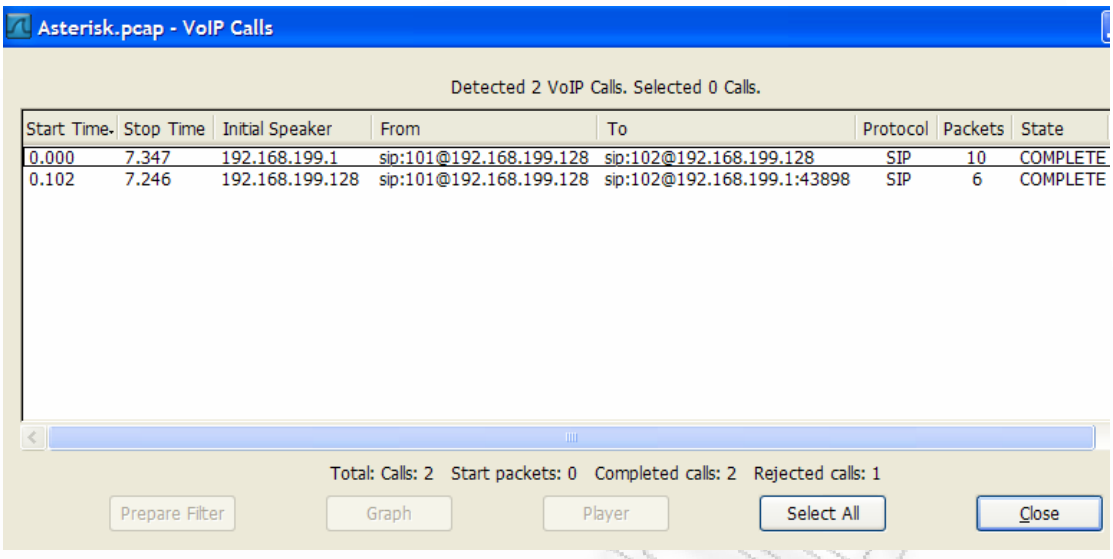
Έχει σταλθεί την χρονική στιγμή 0.0 και είναι ένα request. Είναι φυσικά το αίτημα invite που έκανε ο καλών (χρήστης 101) με IP 192.168.199.1 στον Asterisk με IP 192.168.199.128 με σκοπό να καλέσει τον χρηστή 102, ο οποίος εδώ έχει ιδία IP (192.168.199.1) μιας και όπως είπαμε βρισκόμαστε στον ίδιο υπολογιστή.

Αυτό φαίνεται στο frame No.6 όπου ο Asterisk στέλνει στον χρηστή 102 το αίτημα invite. Ο χρήστης έχει ίδιο IP με του χρηστή 101 αλλά σε διαφορετική θύρα.



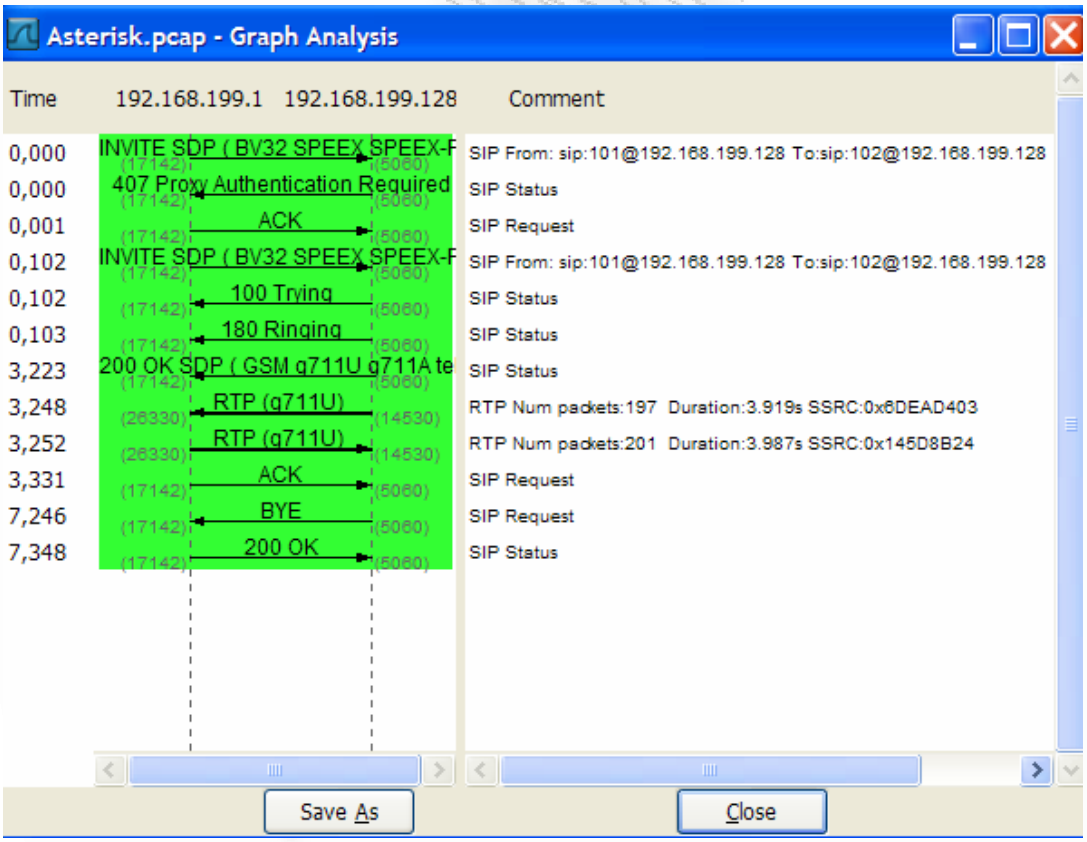
Για να δούμε συνολικά σε γράφημα την ανταλλαγή των αιτημάτων και απαντήσεων μπορούμε να επιλέξουμε από την καρτέλα Statistics την επιλογή VoIP Calls.

Θα μας εμφανίσει τις ανταλλαγές που έχουν καταγραφεί:



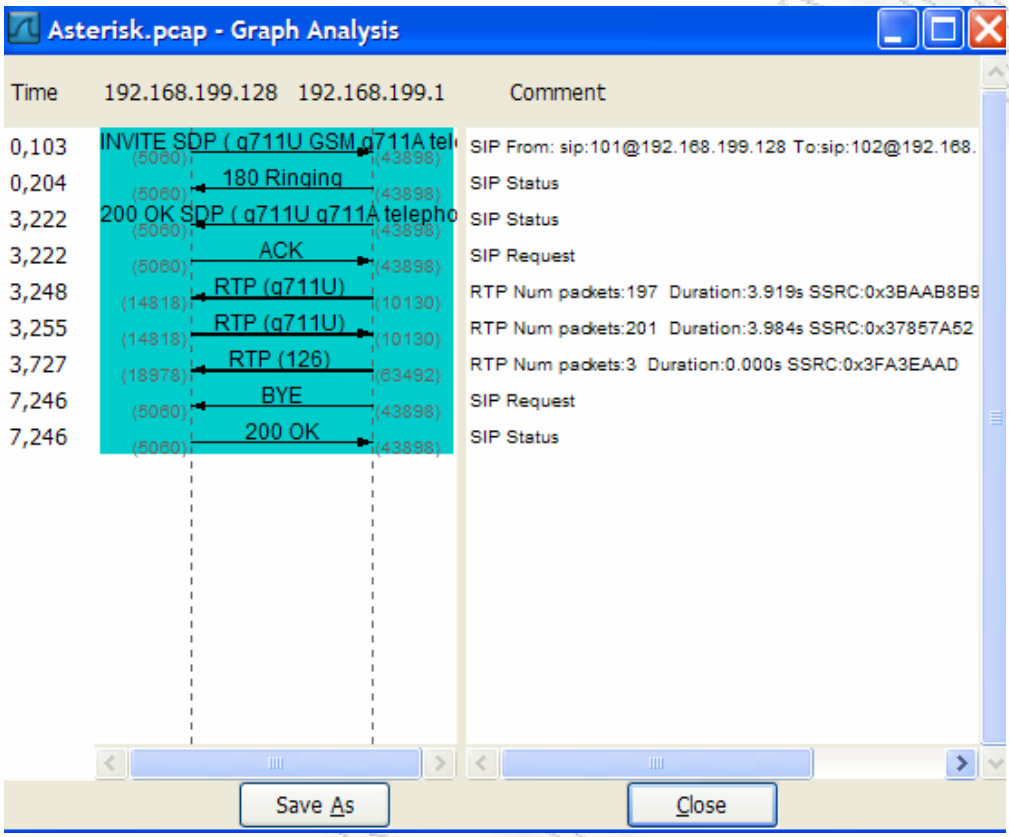
Η πρώτη είναι η ανταλλαγή μεταξύ του χρήστη 101 με τον Asterisk (IP 192.168.199.128) και η δεύτερη του χρήστη 102 με τον Asterisk.

Επιλεγούμε την πρώτη και πατάμε Graph. Μας εμφανίζει την ανταλλαγή σε γράφημα:



Αριστερά βρίσκεται ο χρήστης 101 με IP 192.168.199.1 και θύρα 17142, ενώ δεξιά ο Asterisk με IP 192.168.199.128 και θύρα 5060.

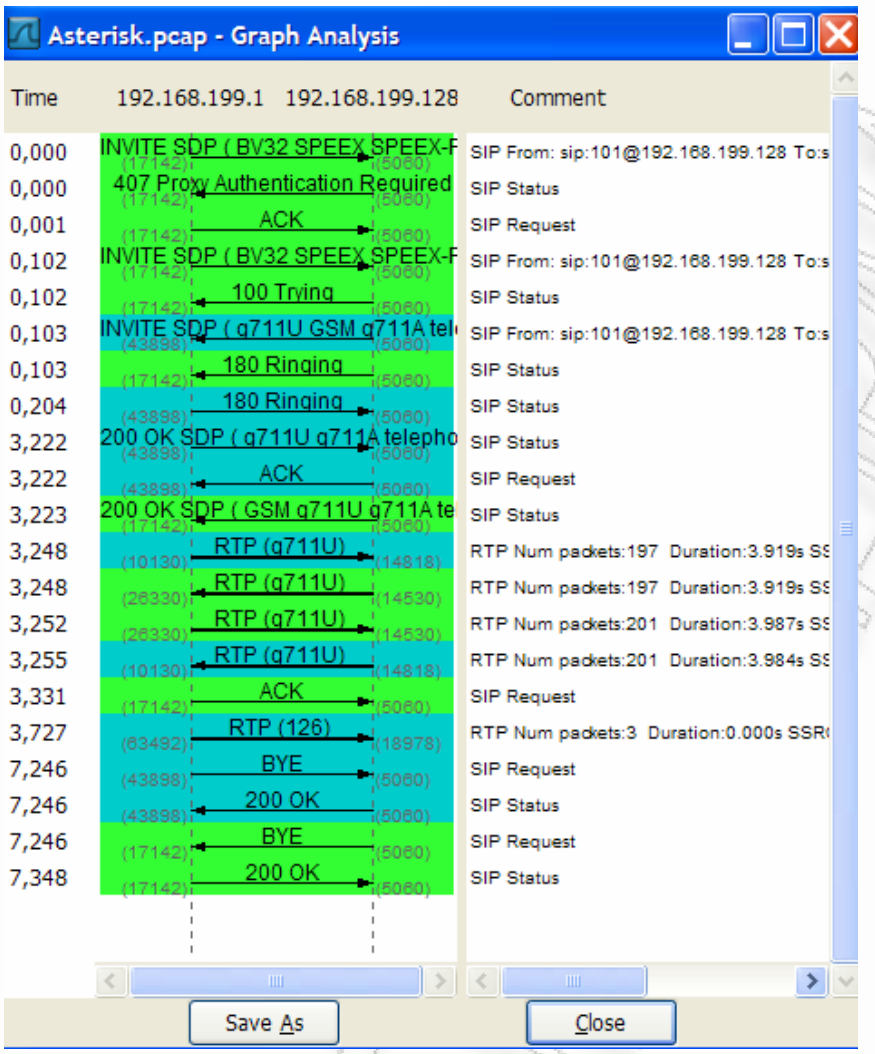
Επιλέγοντας την δεύτερη συναλλαγή μας εμφανίζει το παρακάτω γράφημα:



Όμοια με παραπάνω, αριστερά βρίσκεται ο Asterisk με IP 192.168.199.128 και θύρα 5060, ενώ δεξιά ο χρήστης 102 με IP 192.168.199.1 και θύρα 43898.

Τονίζουμε τις θύρες μιας και θα μας φανούν χρήσιμες στην τελευταία εικόνα όπου θα εμφανίσουμε όλες τις ανταλλαγές ταυτόχρονα, μεταξύ των δύο χρηστών και του Asterisk.

Για να το κάνουμε αυτό, επιλεγούμε “Select All” από το παράθυρο και έπειτα “Graph”.



Εξαιτίας των χρωμάτων οι ταυτότητες των δυο μερών είναι ευκρινής.

Με ανοιχτό πράσινο χρώμα είναι οι ανταλλαγές μεταξύ του χρήστη 101 και του Asterisk, ενώ με γαλάζιο οι ανταλλαγές μεταξύ του χρήστη 102 και του Asterisk.

1) Time 0.000: Ο 101 κάνει αίτηση INVITE στον Asterisk για να συνδεθεί με τον 102
2) Time 0.102: Ο Asterisk τον ενημερώνει με Trying
3) Time 0.103: Ο Asterisk ενημερώνει τον 102 για το INVITE του 101 και στέλνει Ringing στον 101
4) Time 3.223: Ο 102 δέχεται την κλήση στέλνοντας OK

5) Time 3.331: Ο 101 στέλνει ACK ως απάντηση στο OK του 102
6) Ανταλλαγή πακέτων RTP μεταξύ 101 και 102
7) Time 7.246: Ο 102 στέλνει BYE και αποχωρεί. Ο Asterisk στέλνει το BYE στον 101
8) Time 7.246: Ο 101 στέλνει πίσω ένα OK το οποίο φθάνει στον 102

Συμπεράσματα

Στη σημερινή εποχή, ο κόσμος του διαδικτύου έχει μπει για τα καλά στη ζωή μας παρέχοντας μας πρωτόγνωρες υπηρεσίες. Μια από τις τελευταίες και πιο διαδεδομένες εξελίξεις είναι και η τηλεφωνία μέσω internet (VoIP). Το SIP έρχεται να συμπληρώσει αυτή την τεχνολογία ως ένα από τα δημοφιλεστέρα πρωτόκολλα σηματοδοσίας. Δίνει τη δυνατότητα δημιουργίας, μορφοποίησης και τερματισμού συνόδων ήχου και εικόνας μεταξύ δύο ή και περισσότερων συμμετεχόντων. Το μεγάλο του πλεονέκτημα έναντι των άλλων πρωτοκόλλων είναι ότι κάνει σαφή διάκριση μεταξύ της καθιέρωσης συνόδου και της περιγραφής συνόδου. Αυτή η διάκριση κάνει το SIP ουσιαστικά συνεργάσιμο καταστώντας το συμβατό με τα πρωτόκολλα του μέλλοντος.

Το SIP χρησιμοποιείται από πολλές πλατφόρμες. Το Asterisk είναι μια από αυτές, κερδίζοντας καθημερινά όλο και περισσότερους χρήστες. Είναι ένα λογισμικό ανοιχτού κώδικα με πολλά πλεονεκτήματα. Τα σημαντικότερα είναι η ευελιξία, λειτουργικότητα, αξιοπιστία του και πάνω από όλα ότι παρέχεται δωρεάν σε όλους. Μέσω του Asterisk έχει πλέον καταστεί δυνατό για τον καθένα να κάνει το δικό του ιδιωτικό τηλεφωνικό δίκτυο. Είτε πρόκειται για ένα μικρό οικογενειακό δίκτυο, είτε για μια μεγάλη εταιρία με εκατοντάδες τηλεφωνικές συσκευές, το Asterisk δίνει την λύση. Η εποχή της επικοινωνίας μέσω του διαδικτύου μόλις έχει ξεκινήσει.

Βιβλιογραφία – Πηγές

[1] Andrew S. Tanenbaum. 2000. *Δίκτυα Υπολογιστών*. 3η Έκδοση. Εκδόσεις Παπασωτηρίου.

[2] “Asterisk Guru - Tutorials and howto's for the asterisk PBX and voip in general” . http://www.asteriskguru.com

[3] Ben Jackson and Champ Clark. 2007. *Asterisk Hacking*. Pap/Com ed. Syngress.

[4] David Gomillion and Barrie Dempster. 2005. *Building Telephony Systems with Asterisk*. Packt Publishing.

[5] Jim Van Meggelen, Jared Smith, and Leif Madsen. 2005. *Asterisk: The Future of Telephony*. O'Reilly Media, Inc.

[6] Alan B. Johnston . 2003. SIP: Understanding the Session Initiation Protocol, Second Edition

[7] Paul Mahler. 2004. VoIP Telephony with Asterisk

[8] Henry Sinnreich. 2006. Internet Communications Using SIP: Delivering VoIP and Multimedia Services with Session Initiation Protocol

[9] Olivier Hersent. 2005. Beyond VoIP Protocols: Understanding Voice Technology and Networking Techniques for IP Telephony

[10] http://www.voip-info.org/wiki

[11] http://www.wikipedia.org

[12] http://www.sipforum.org

[13] http://www.ietf.org