PROJECTE CENTRALETA PBX

MEMÒRIA

Ricard Sales López
ETIS
Dr. Ignasi Rius Ferrer
13 Gener de 2012

ÍNDEX TEMÀTIC

ÍNDEX DE FIGURES

1. DEFINICIO DEL PROJECTE

1.1. INTRODUCCIÓ

L'actual Treball Final de Carrera es desenvolupa a l'àrea "**Plataforma GNU/Linux**". El projecte proposat té a veure amb la tendència predominant del mercat en quan a les comunicacions en empreses mitjanes i grans. Actualment, la millora de la qualitat dels accessos a internet, ha fet proliferar serveis de telefonia per internet (VoIP).

Els preus reduïts de les trucades IP i la necessitat de majors prestacions (bústies de veu, trucades a grups d'extensions, atenció automàtica –anomenada IVR, etc...) s'ha respòs amb solucions VoIP o mixtes (amb sortides RDSI o analògiques combinades amb VoIP) .El cost de les centraletes i telèfons IP porta a diferents consideracions. En primer lloc, els fabricants de centraletes ofereixen solucions tancades a preus a vegades massa alts. En segon lloc, també ens obliguem a dependre dels seus productes de cara a possibles ampliacions. Finalment, acabem depenent del suport tècnic dels fabricants (un servei a vegades mediocre, en comparació amb els suports de les comunitats OpenSource).

En aquest projecte oferirem un producte de centraleta basat en solucions GNU/Linux existents que integri tot el que es necessita a nivell de xarxa i VoIP. Busquem que sigui de configuració fàcil per a l'administrador final. El hardware necessari no és massa exigent per a un ús de poques extensions, així que podrem virtualitzar el producte per tal de generar el nostre prototip.

1.2. OBJECTIUS

L'objectiu d'aquest projecte és poder crear una centraleta física amb tots els serveis associats integrats (Firewall, accés remot, servidor de serveis de xarxa i centraleta IP). Així, utilitzant solucions contrastades en un únic producte, es poden contenir molt els costos de TI i de comunicacions en les empreses de consum telefònic alt. El producte final podria competir amb solucions propietàries de fabricants de centraletes tradicionals. Tanmateix es poden oferir tots els serveis amb un cost més contingut i una configuració assumible per a tècnics de perfil baix per l'àmplia documentació que es troba a Internet.

1.3. ESTRUCTURA DE LA MEMÒRIA

Bàsicament es pot dividir la memòria en 5 blocs diferenciats, que faig coincidir amb els següents punts de primer nivell:

En el punt 2 tenim la planificació inicial de tot el projecte, que ha servit per a guiar tot el procés. No només hem donat terminis com a dates; també hem creat fites que permetien anar controlant el desenvolupament correcte. Oferim la planificació representada gràficament i una explicació del que significarà cada punt a nivell de feina i objectius de forma exhaustiva.

El punt 3 fa un recull i estudi de requeriments, així com el procés per a decidir quines solucions i infraestructures son de menester per tal de donar solucions òptimes. També fem una ullada als productes que ens ofereix el mercat i les tendències del mateix. Es en aquest punt on es prenen la majoria de decisions que defineixen definitivament el producte que volem crear.

Els punts 4 i 5 son la implementació del projecte i les proves de la maqueta respectivament, documentant la feina més directe sobre el nostre prototip.

En el punt 6 faig una valoració del que ha significat tot el procés, tant a nivell tècnic com organitzatiu. També reviso què he utilitzat dels coneixements adquirits durant els estudis i què he aprés en el procés de portar una primera idea fins a generar un prototip d'un producte que pot arribar a ser viable.

2. PLANIFICACIÓ TEMPORAL
2.1. PLANIFICACIÓ D'ENTREGUES

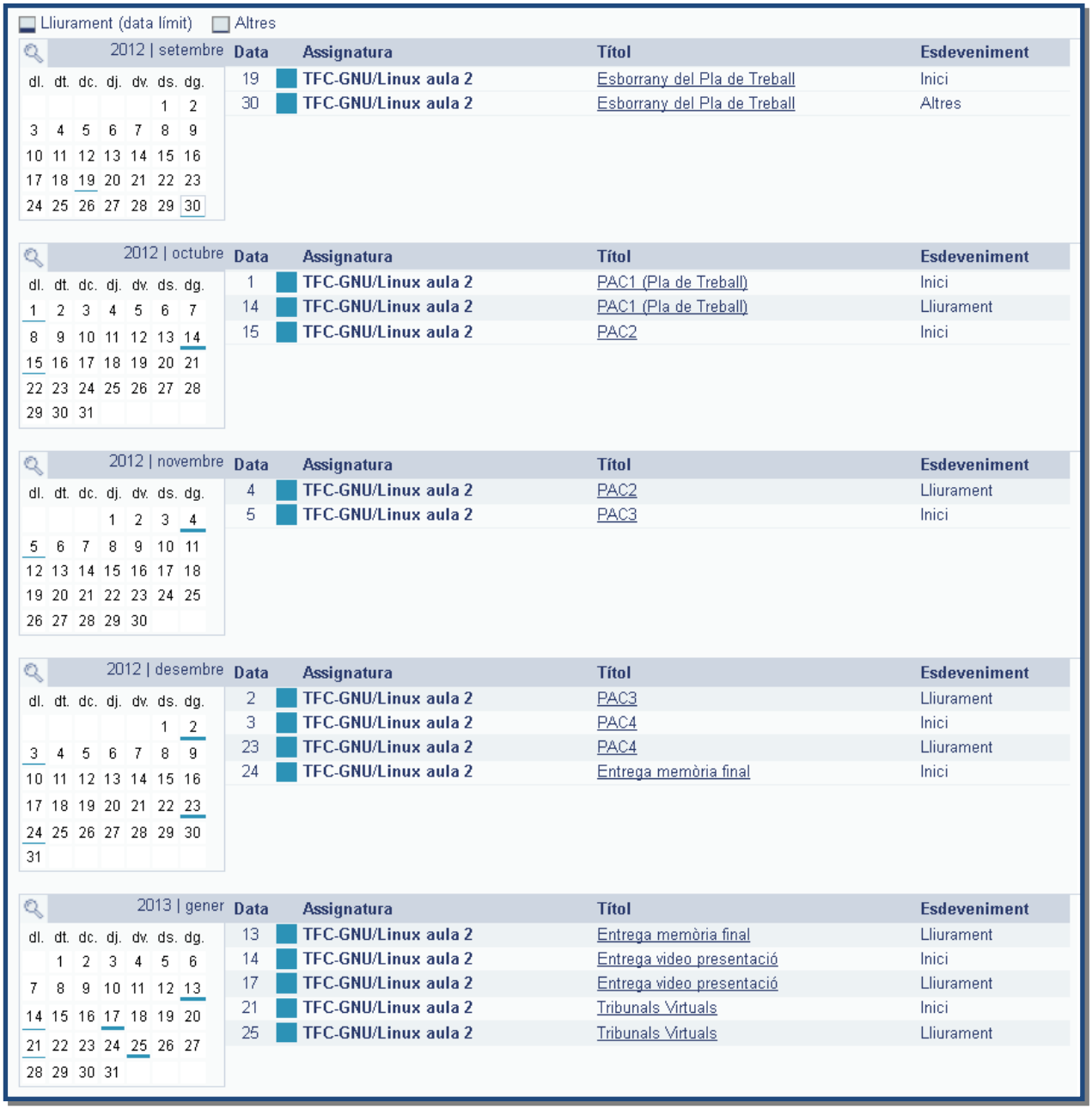

Figura 1: Planificació d'entregues

2.2. PLANIFICACIÓ DETALLADA

Hoja de tareas

	Nombre de tarea	T	Comença	Fi
1	**TFC Plataforma GNU/Linux - Projecte Centraleta PBX**	**129 d**	**19/09/12**	**25/01/13**
2	**01 - Planificació (PAC1)**	**26 d**	**19/09/12**	**14/10/12**
3	01.01 - Realització Esborrany del Pla de Treball	12 d	19/09/12	30/09/12
4	01.02 - Elaboració del Pla de Treball	13 d	01/10/12	13/10/12
5	01.03 - Revisió i Entrega Pla de Treball (PAC1)	1 d	14/10/12	14/10/12
6	**02 - Descripció Tecnologies (PAC2)**	**21 d**	**14/10/12**	**04/11/12**
7	02.01 - Revisió de Requeriments	3 d	15/10/12	17/10/12
8	02.02 - Estudi de les solucions al mercat i tendències	3 d	14/10/12	17/10/12
9	02.03 - Selecció de tecnologies	8 d	18/10/12	25/10/12
10	02.04 - Anàlisis de les tecnologies escollides	6 d	29/10/12	03/11/12
11	02.05 - Revisió i Entrega Descripcions tecnologies (PAC2)	1 d	04/11/12	04/11/12
12	**03 - Disseny i Implementació (PAC3)**	**28 d**	**05/11/12**	**02/12/12**
13	03.01 - Disseny infraestructura de serveis	6 d	05/11/12	10/11/12
14	03.02 - Implementació de tots els serveis	15 d	11/11/12	25/11/12
15	03.03 - Redactat i Entrega Disseny i Implementació (PAC3)	7 d	26/11/12	02/12/12
16	**04 - Resultats i Anàlisis del Treball (PAC4)**	**21 d**	**03/12/12**	**23/12/12**
17	04.01 - Proves i Resultats	14 d	03/12/12	16/12/12
18	04.01 - Anàlisis del treball i redacció (PAC4)	7 d	17/12/12	23/12/12
19	**05 - Memòria, Presentació Virtual i Tribunal (Entrega Final)**	**33 d**	**24/12/12**	**25/01/13**
20	05.01 - Recopilació d'Informació i Redactat Memòria	21 d	24/12/12	13/01/13
21	05.02 - Presentació Virtual	4 d	14/01/13	17/01/13
22	05.03 - Període de Consultes del Tribunal	5 d	21/01/13	25/01/13

Figura 2: Planificació detallada

2.3. DESCRIPCIÓ DE LES DIVISIONS EN ACTIVITATS

2.3.1. REALITZACÓ DE L'ESBORRANY DEL PLA DE TREBALL

L'inici del procés és òbviament una primera aproximació al Planning del projecte. Per tal de poder distribuir les tasques en una línia de temps és necessària la divisió de les mateixes en grups, que siguin coherents (pel que fa a la càrrega de treball) amb les entregues de les diferents PACs (que de per sí ja disposen d' una estructura lògica). Al final d'aquesta fase tindré un esquelet amb una primera proposta.

2.3.2. ELABORACIÓ DEL PLA DE TREBALL

Un cop analitzades les indicacions del tutor, en aquest punt emplenarem aquesta proposta inicial, consistint el primer pas en la redacció dels requeriments i objectius definitius. En segon lloc es farà la descripció de les diferents fites negre sobre blanc, adaptant el diagrama de Gantt a les correccions.

2.3.3. REVISÍO I ENTREGA DLE PLA DE TREBALL (PAC1)

Aquest apartat té per objectiu revisar l'estil i redactat del treball, a fi de respectar les indicacions dels Mòduls 2 i 3. A nivell de maquetació és el moment de revisar l'estètica del document un cop garantits la part formal i de continguts.

2.3.4. REVISIÓ DE REQUERIMENTS

En aquesta fase comencem a aproximar-nos a aspectes de caràcter tecnològic. Entendre els problemes que ens representen els diferents requeriments és el primer pas per tal d'evitar improvisacions quan abordem l'etapa d'implementació.

2.3.5. EL MERCAT I LES SOLUCIONS COMERCIALS

Farem un repàs de la situació del mercat de VoIP al nostre país i de les tendències de futur que s'observen. Tanmateix, revisarem diferents opcions comercials de productes propietaris i el que fan empreses que ofereixen solucions basades en opensource. Finalment justificarem econòmicament el nostre projecte.

2.3.6. SEL·LECCIÓ DE TECNOLOGIES

Un cop estudiat el plec de necessitats a fons, és l'hora d'assignar diferents solucions a cada repte del projecte. I per tant, un cop arribats a aquest punt és important escollir els serveis que millor s'integrin entre ells per tal de no tenir problemes posteriors. I és que, de fet, molts cops les incompatibilitats entre processos servidors estan documentades i es poden evitar amb antelació, escollint les combinacions més adients. Per tant, aspectes tals com les capacitats i limitacions, les futures actualitzacions o el suport tècnic són importants i s'han de tenir en compte en l'elecció que ens ocupa. En conclusió, i per les raons exposades, aquesta fase és estratègicament clau.

Per tal de no perdre el coneixement adquirit durant les consideracions, en aquest apartat intentaré, de forma esquemàtica, justificar la selecció feta d'entre totes les opcions valorades. Així doncs, aquesta elecció haurà de ser raonada, i, per tant, exposaré els motius que m'han dut a descartar cadascuna de les altres opcions.

2.3.7. ANÀLISI DE LES TECNOLOGIES ESCOLLIDES

Un cop escollits tant les tecnologies com els servidors és el moment d'estudiar en profunditat les interaccions i els funcionaments de cadascun d'ells. No n'hi ha prou amb fer una pura enumeració de serveis; hem de poder construir un discurs tecnològicament coherent sobre tot el projecte. Finalment, entendre el funcionament ens ajudarà en la implementació, a banda de justificar la mateixa.

En el cas que ens ocupa, el protocol SIP tindrà probablement un paper destacat. Pel que fa als anàlisis dels serveis de xarxa bàsics, no penso que sigui intel·ligent aprofundir massa, al ser àmpliament coneguts. El criteri en què em basaré serà: "el menys conegut, el més analitzat".

2.3.8. REVISIÓ I ENTREGA DE LES DESCRIPCIONS DE LES TECNOLOGIES (PAC2)

Aquesta entrega és potser la més difícil a nivell de redacció. L'aparició de contingut netament tecnològic ens obliga a mantenir un equilibri entre un llenguatge suficientment entenedor per al lector, però sense menystenir el grau de coneixement tècnic del mateix. No es pot caure en una sobre-explicació ni es pot pretendre assumir que es coneix amb detall tot el que es defineix. El meu objectiu, per tant, girarà entorn la presentació d'un text clar en totes les explicacions però no condescendent en excés. A nivell de maquetació, per una altra banda, no hi trobarem canvis significatius, doncs la pauta haurà vingut donada a la PAC1.

2.3.9. DISSENY DE LA INFRASTRUCTURA DE SERVEIS

Per tal de realitzar la implementació s'ha de planificar quina infraestructura serà necessària. Així, en el nostre projecte virtualitzarem per tal d'oferir una màquina que pugui ser provada pel tribunal si es creu necessari. Per aquest motiu la grandària dels discs virtuals no serà l'òptima per a un funcionament professional, sinó que ens acostarem al mínim per a que es pugui descarregar millor. Això és possible al tractar-se d'un entorn que no és per producció (i per tant ens estalvia espai per als logs, per exemple). De fet, no només hem de pensar en el projecte sinó en l'entorn que necessitarem per a fer les simulacions del funcionament real.

2.3.10. IMPLEMENTACIÓ DE TOTS ELS SERVEIS

En aquest punt intentaré fer un exercici d'honestedat, explicant també els problemes o equivocacions que es poden cometre i com evitar-los. Penso plantejar aquest apartat com un quadern de bitàcola perfecte per tal de poder reproduir el procés per part de qualsevol que llegeixi el projecte.

2.3.11. REDACTAT I ENTREGA DEL DISSENY I IMPLEMENTACIÓ (PAC3)

El redactat de la PAC3 és essencial per tal d'eliminar ambigüitats. I a fi que el procés que s'explica sigui fàcilment reproduïble per un lector davant el seu ordinador, el llenguatge usat serà molt net i clar.

2.3.12. PROVES I RESULTATS

És l'hora de posar en pràctica la feina feta fins ara i crear un bon conjunt de proves és imprescindible per a la qualitat final. El fet de ser exigent en aquest punt fa possible que se'ns mostrin detalls que ens han passat per alt o que han creat problemes en el funcionament real.

Redactar sobre els resultats de les proves pot ser incòmode o molt gratificant. L'auto-exigència és necessària per tal de realitzar les correccions necessàries abans de donar l'aprovació definitiva al projecte final i, per tant, és possible que aquesta fase provoqui correccions en altres punts. S'ha de ser molt curós en prendre bona nota de les correccions si calen de cara a una memòria final adequada.

2.3.13. ANÀLISI DEL TREBALL I REDACCIÓ (PAC4)

Arribats a aquest moment farem referència al procés dut a terme fins ara, explicant com s'ha anat evolucionant respecte al Planning. Tanmateix, en aquest apartat faré una valoració sobre les desviacions de les estimacions fetes a l'inici. La redacció dels punts 2.3.11 i 2.3.12 no sembla que sigui especialment complicada, però ha de respectar un punt de vista realista i objectiu, fugint de sensacions si pretenem valorar realment el mesurable. Abstreure's de les pròpies sensacions és difícil i crec que realitzar el TFC m'aportarà creixement. Malgrat tot, assumeixo que un bon anàlisi és el que compartirien diferents observadors sobre el mateix procés.

2.3.14. RECOPILACIÓ D'INFORMACIÓ I REDACTAT DE LA MEMÒRIA

Si s'han treballat correctament tots els apartats anteriors, arribats a aquest punt gran part de la feina estarà feta. Només caldrà donar cohesió a les diferents entregues, a fi de generar un únic document. En conseqüència, la "unicitat" del redactat és importantíssima i suposarà un gran repte, que portarà a intentar encaixar amb naturalitat i fluïdesa totes les PACs, fent les correccions de continguts i estil que es creguin necessàries. Per tal d'aconseguir aquest objectiu caldrà invertir temps en rellegir el text final, a fi de polir-lo en cada lectura i eliminar possibles canvis d'estil que es puguin observar.

2.3.15. PRESENTACIÓ VISUAL

L'objectiu de la presentació és doble: per una banda mostrar de manera sintetitzada la feina feta i per una altra banda demostrar el seu bon funcionament. Així, a partir de la memòria i les proves fetes es generarà un vídeo que donarà una perspectiva general de tot el que s'ha fet i treballat. Addicionalment es farà un PowerPoint de guia com a material de suport en la presentació de la memòria davant un auditori.

2.3.16. PERÍODE DE CONSULTES DEL TRIBUNAL

Com a fase final del TFC, establirem un període per tal de poder respondre les preguntes que, un cop analitzat en detall el projecte, pugui plantejar el tribunal. L'objectiu serà aconseguir que no siguin necessaris aclariments sobre aspectes que no estiguin a la memòria. Malgrat tot, és també necessari enfrontar les preguntes del tribunal per a estar segurs de que s'han assolit els coneixements tècnics i formals exigibles.

2.4. DIAGRAMA DE GANTT

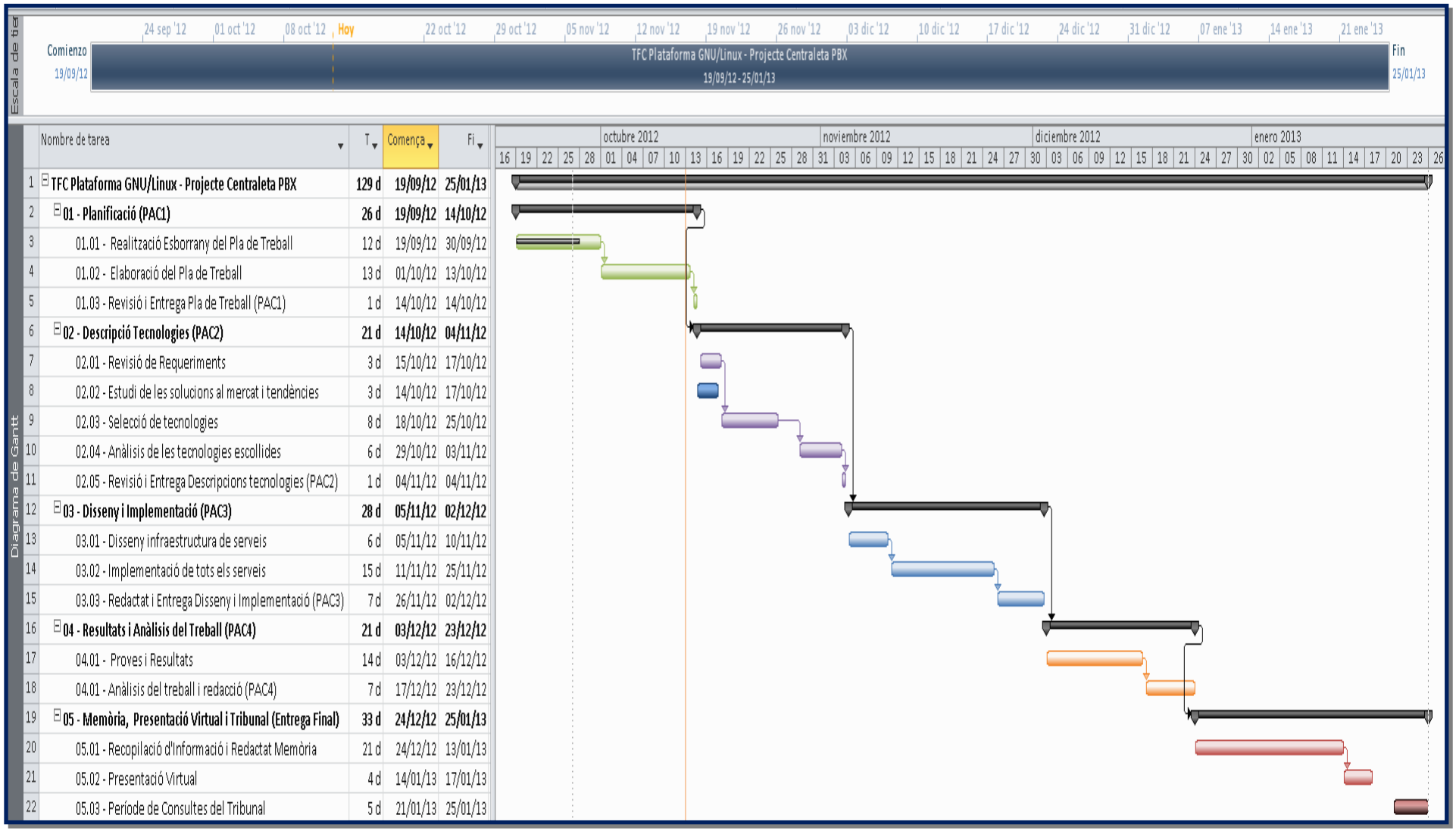

Figura 3: Diagrama de Gannt

3. REVISIÓ DE REQUERIMENTS

3.1. DEFINICIÓ REQUERIMENTS SERVEI CENTRALETA PBX

Els requeriments de la Centraleta PBX són:

- Oferir serveis avançats en les gestions de trucades.
- Aportar una gestió visual per a l'administrador final de la màquina.
- Fer que els terminals telefònics que s'utilitzin es puguin comunicar entre ells sense cost.
- Permetre la interconnexió amb xarxes mòbils, RDSI i analògiques.
- En cas de més d'una seu, fer que dues centraletes es puguin comunicar o si la seu és petita que pugui usar una única centraleta.
- Fer compatible la nostra centraleta amb terminals de diferents fabricants i no dependre d'un de sol.
- Utilitzar una solució que tingui un ampli suport i documentació per donar independència a l'informàtic de l'empresa usuària del nostre producte.
- Per a poder generar una solució integral (però amb un temps de desenvolupament ràpid) s'utilitzarà o reutilitzarà codi lliure amb llicencia GNU.

3.2. DEFINICIÓ REQUERIMENTS SERVEIS ASSOCIATS

La centraleta ha de suportar serveis subsidiaris per a l'administració i accés a la xarxa:

- El trànsit VLAN per a mantenir-se separada de la resta de la xarxa.
- Oferta de DHCP,DNS,NTP a tots els terminals telefònics que en sol·licitin.
- Accés remot per a l'administració.
- Administració de la centraleta amb interfície web.
- Accés de terminals telefònics des de l'exterior de l'empresa de forma segura.
- Integració de la seguretat perimetral a la centraleta amb capacitat d'autoprotegir-se.

3.3. ESTUDI DE REQUERIMENTS

Estudiant la definició de requeriments queda patent que el nostre camí serà el programari lliure o Freeware. I per tal de respectar els terminis establerts per al desenvolupament del producte final (tenint en compte l'abast de la solució que volem proposar) anirem cap a la integració de serveis existents i que no depenguin de llicències cares.

L'objectiu principal és obtenir una centraleta, amb gestió avançada de trucades i que pugui treballar amb equips heterogenis. Per tant, la necessitat d'interconnexió amb les xarxes tradicionals de telefonia és imperativa. Òbviament s'haurà de poder configurar amb un entorn gràfic amable amb l'administrador, per tal d'evitar dependències de tercers per part del client.

Així doncs, i a fi de poder oferir un projecte sense fissures, hem d'incloure tots els serveis de xarxa associats i securitzar la centraleta. Aquest aspecte és MOLT important, donat que, habitualment, existeix un mercat de VoIP fraudulent (especialment ubicat a la Xina i Europa de l'Est). Personalment, he estat testimoni de com s'han produït forats de 6000 € de consum en comptes SIP, en pocs minuts, amb trucades a Cuba (0,90 €/min) o Somàlia (0,85 €/min)[1]. El furt de credencials (degut a una mala implementació) és habitual a causa del desconeixement dels sistemes.

Quobis, consultora especialitzada en VoIP, ha realitzat un Webinar[2] molt interesant sobre la seguretat en VoIP. Les dades que presenta són realment esfereïdores, i posen de manifest que l'accés a les centraletes és un tema de vital importància.

A nivell de proteccions, s'ha de deixar el mínim d'accés necessari i utilitzar comunicacions xifrades (mai en text pla), especialment quan parlem d'accessos exteriors.

Des del punt de vista de l'usuari intern, la separació dels equips telefònics en VLANs diferents de les estacions de treball ens garantiran una mica d'intimitat entre terminals i centraleta (a banda del xifrat i control de trucades que cada tipus d'usuari pot fer).

Pel que fa a l'accés exterior, solucions com SSH o VPN (accessos remots i protegits per encriptació) ens aporten major consistència defensiva. I d'aquí ve la seva importància, primerament per la necessitat de reduir costos en trucades remotes, tractant-les com si fossin trucades entre extensions i en segon lloc per la voluntat de donar mobilitat a baix cost als empleats que ho necessitin.

Tot i així, malgrat totes les consideracions fetes, trobem nombroses solucions que ens permeten, en major o menor grau, assolir les funcions i capacitats demandades. Per tant, un cop estudiats els requeriments, estem en disposició de poder descartar solucions candidates o redefinir requeriments si s'observés que alguns són incompatibles entre sí.

3.4. EL MERCAT I LES SOLUCIONS COMERCIALS

Actualment la telefonia IP gira entorn d'un protocol estàndard que s'ha assumit per part de fabricants i desenvolupadors. Es tracta del **protocol SIP**[3], que va presentar el seu primer esborrany l'any 1996. De fet, totes les solucions que actualment són al mercat busquen la compatibilitat amb aquest protocol per a la senyalització i també amb el **RTP**[3] per l'streaming (enviament d'àudio y/o imatge durant les trucades entre terminals i centraleta). I tot i que aquest fet -a nivell tecnològic- ens podria fer pensar que tots els competidors adopten les mateixes solucions, veurem a continuació com això no és ben bé així. En realitat, podrem observar com, partint del mateix protocol, es plantegen diferents escenaris i aproximacions. En conseqüència, i fent una revisió de les opcions que el mercat ofereix, les podem agrupar en tres grans blocs, com són les solucions hardware, les solucions software amb servidor en propietat i el cloud VoIP.

3.4.1. LES SOLUCIONS HARDWARE

Pel que fa a les solucions hardware, trobem centraletes VoIP, pures o mixtes. I són aquestes últimes les que solen tenir una acceptació i preu més alts. En aquest context, el mercat ofereix molts fabricants de centraletes, que normalment també fabriquen terminals, les quals aconsellen adquirir. Per una altra banda, i malgrat que es tracti de hardware, la majoria de fabricants utilitza uns sistemes de llicències, que encareixen força el producte en cas que les funcions que es busquen les requereixin. Així, la segmentació de les capacitats provoca diferents trams d'ingressos per als fabricants. I en conseqüència, de vegades, amb una mateixa centraleta hardware, es poden anar sumant funcions que s'activen amb un serial del fabricant. Sincerament, en entorns hardware controlats és més fàcil trobar fiabilitat i normalment no es presenten problemes d'interaccions entre processos. En el cas de les empreses grans, aquestes són les més avocades a comprar les solucions propietàries de fabricants punters. En canvi, pel que fa a les empreses mitjanes o en entorns públics, es tendeixen a mirar molt més el costos, tot i comportar un petit peatge en l'estabilitat.

3.4.2. LES SOLUCIONS SOFTWARE

En el cas del Software de VoIP, veiem com aquesta opció està avançant a nivell de mercat, ja que proposa prestacions avançades a preus molt continguts. Aquest avantatge s'assoleix en part gràcies a que (depenent del nombre d'usuaris, les interfícies i la redundància que es vulgui) es pot ajustar el preu del nostre servidor. La majoria de software de telefonia que s'utilitza al món es basa en **Asterisk**, que és una solució GNU/Linux desenvolupada desinteressadament per una gran comunitat. Sobre aquesta base s'han anant sumant funcionalitats, fins a crear distribucions GNU/Linux per telefonia i gestió web amb funcions molt avançades.

Actualment, tot el que sigui necessari en qualsevol entorn es pot trobar ja fet i compartit gratuïtament ... i en cas de no ser així, es pot començar a programar la solució, i compartir-la després. Una de les característiques que converteix Asterisk en una referència mundial és la creació d'unes llibreries de programació, que permeten implementar qualsevol comportament de les trucades de veu o videoconferència i que també suporta la plataforma. Asterisk[4] va néixer l'any 1999 i es tracta d'un projecte madur, fiable en entorns d'alt rendiment i amb una comunitat molt activa que facilita un suport excel·lent.

3.4.3. LES SOLUCIONS CLOUD

Finalment tractarem el tercer gran bloc de solucions i que darrerament s'ha convertit en una opció en creixement. Estem parlant del cloud i el seu impacte en la telefonia no és menor que en d'altres àmbits. Ara per ara, tenim solucions de centraletes virtuals de proveïdors de VoIP força treballades.

Prenent com a base solucions gratuïtes com **Asterisk**, **OpenSIP** o de pagament com **Centile**[5], s'està oferint registre remot d'extensions amb una centraleta penjada al núvol.

El proveïdor ha de mantenir una estructura de suport, per tal d'administrar la centraleta del client a canvi d'unes taxes mensuals. Aquestes garanteixen una quantitat de minuts gratuïts a diferents destins i d'altres avantatges associats, com per exemple: bústies de veu, IVR, anuncis, cues de trucades, grups d'extensions, horaris de desviaments, etc. Com a regla general, permeten adquirir terminals telefònics a preus baixos i/o finançats amb les factures. Encara que no és gaire habitual, a vegades s'utilitzen polítiques de permanència a fi de garantir el pagament dels aparells o es reclamen els terminals en cas de baixa.

El posicionament del producte cloud promet un cost mensual controlat en el manteniment del servei, sense necessitat que el client tingui un administrador de la centraleta. Per una altra banda, el compromís per part del canal de distribució del producte és que el servei tècnic del proveïdor resoldrà tots els problemes. Les centraletes virtuals tenen una gran penetració en la petita i micro empresa, on el cost de les estructures ha de tendir a zero. Per tant, és òbviament la decisió ideal si aquestes empreses busquen reduir costos en telefonia al fer trucades internacionals.

3.4.4. TENDÈNCIA DEL MERCAT

Un cop definits aquests tres blocs de solucions, cal posar en clar que la VoIP s'està implementant, buscant principalment fomentar la reducció de costos. Com ja hem esmentat abans, són les trucades a l'estranger - especialment fora de l'Europa de l'Euro – les que fan pujar més les factures. I en l'actual context de crisi, la necessitat d'obrir mercat nou a l'exterior ha comportat un creixement al sector de la telefonia IP. El que es ven és estalvi, i aquesta proposta, en un moment de recessió econòmica, es fa especialment atractiva i la converteix en un producte estrella. En conseqüència, la ràpida proliferació de proveïdors de VoIP ha portat a les operadores de telefonia tradicional a respondre amb fortes baixades de tarifes a les seves ofertes. Malgrat aquest fet, les trucades internacionals segueixen tenint un cost més baix si es fan a través de SIP. I aquesta situació ha justificat l'increment en quota de mercat que ha guanyat la VoIP.

A Espanya, la tendència és clara, però s'ha de dir que el seu desenvolupament ple serà possibles només gràcies a les millores en velocitat i qualitat de les connexions a Internet. Els consums de banda ampla necessaris per a fer trucades per la xarxa són entre 40 i 100 Kb/s per trucada. El progrés de les infraestructures ha permès la implantació de solucions informàtiques en molts camps. Pel que fa a la telefonia IP, la millora en l'accés a internet ha estat una qüestió cabdal.

A l'actual regulació estatal, s'intenta liberalitzar el màxim possible el mercat de les telecomunicacions. A nivell polític, la UE treballa amb l'axioma de "Neutralitat de la Xarxa", és a dir, la negativa a limitar un tipus de trànsit si no hi ha delicte. Els ISP (proveïdors de serveis d'internet), que majoritàriament també són operadors de telefonia, no han restringit l'ús dels accessos per a VoIP amb terceres companyies. Hi ha hagut intents d'alguns ISP, però no ha tingut gaire repercussió a nivell d'accés fixe a la xarxa[6].

Un tema a banda és el trànsit de VoIP sobre accessos mòbils a internet. Cada cop s'utilitzen més softphones que faciliten comunicacions entre usuaris del programa o registrant un compte contra una centraleta pròpia. Aquest fet permet aprofitar la tarifa de dades, enlloc de carregar una trucada tradicional a la factura. No obstant, molts proveïdors de telefonia mòbil sí limiten aquest trànsit, tot i que, a nivell europeu, s'estan començant a dictar sentències, les quals, poc a poc, van establint jurisprudència al respecte[7], i en un futur afectaran també a les xarxes 3G/4G.

En conclusió podem dir que el que hem exposat fins aquí ens presenta el marc general del mercat de telecomunicacions en relació amb la telefonia IP, i és aquí on el nostre projecte haurà de competir.

3.5. COMPARATIVA D'OPCIONS VALORADES

Nom Solució	Fabricant	Tipus Solució	Cost Aprox.	A Favor	En Contra	
UC 500Series	CISCO	HardWare/ Mixta	>2500 $	Solució Robusta, Seguretat d'Accés	Només configuració per consola. Té una relació dolenta preu/usuari	Hardware
IP Office	Avaya	HardWare/ Mixta	El preu base ja és alt, es fa difícil calcular preu ja que s'ha d'anar sumant llicències	Integra Accés des de l'exterior propietari, Fins 384 usuaris	Té un sistema de llicencies i mòduls d'ampliació para cada prestació.	
3CX Phone System	3CX	SoftWare/ VoIP	FreeWare +Llicencia Windows +Server a partir de 200€	Integra moltes funcions avançades amb una interfície senzill	És inestable en comparació amb d'altres solucions. La integració sobre Windows dels serveis associats implicaria utilitzar servidors freeware y un Windows Server amb les llicències associades.	Software
Asterisk	GNU/ Linux	SoftWare/ VoIP Ampliable a Mixta amb targetes addicionals a partir de 200€	Server a partir de 200€	Gratuït, potent i altament personalitzable. Hi ha software per a alt rendiment per a serveis associats sobre qualsevol Linux. La comunitat de suport d'Asterisk és immensa	No té configuració gràfica. Es necessiten coneixements específics per administrar-ho a nivell de funcionament normal	
FreePBX	GNU/ Linux	SoftWare/ VoIP Ampliable a Mixta amb targetes addicionals a partir de 200€	Server a partir de 200€	Basat en Asterisk, amb interfície gràfica que volca configuracions avançades (amb molta facilitat) sobre els arxius de configuració d'Asterisk. Ofereix serveis avançat de telefonia i backups integrats	La posada en marxa és més complicada que amb solucions propietàries però els canvis de configuració són fàcils. S'han d'implementar a banda els demés serveis associats.	
Elastix	GNU/ Linux	SoftWare/ VoIP Ampliable a Mixta amb targetes addicionals a partir de 200€	Server a partir de 200€	Basat en FreePBX, té funcions avançades de gestió i estadístiques que no té FreePBX	Es tracta d'una distribució Linux CentOS(Basada en RedHAt), que ho té tot llest per engegar, però no et permet escollir una altra distribució. CentOS és un projecte una mica deixat com a distribució.	
Oigaa Office	Voz Telecom	Cloud / VoIP	Pagament mensual	Interfície senzill, terminals preparats per plug&play. Inclou els serveis associats al router ADSL que es facilita. Permet trucades entre extensions sense cost entre terminals remots	No permet gaires funcions avançades, la configuració que si permet s'ha de fer en molts casos a través del suport tècnic de l'empresa.	Cloud
vPBX	Adam VoIP	Cloud/ VoIP	Pagament mensual	Es tracta bàsicament d'una FreePBX i té totes les seves avantatges per a VoIP	No permet un accés VPN per a una connexió segura des d'una xarxa insegura.	

* Les dades del quadre son extretes de les pàgines del fabricant que es recull a la bibliografia

Figura 4: Quadre comparatiu d'opcions

3.6. CONSIDERACIONS TÈCNIQUES

Primerament, el nostre projecte aposta pel grup de solucions de software Lliure o Freeware. De totes formes he estudiat les altres dues opcions i sembla clar que, per als nostres requeriments, no donen una resposta prou eficient a nivell tècnic o econòmic. Les opcions proposades pels fabricants de centraletes, tot i donar respostes harmonitzades i vàlides a nivell de funcionalitat, tenen el gran handicap dels costos. A banda presenten un problema tècnic afegit, ja que s'han de buscar tècnics especialitzats en cada marca, donat que configurar aquestes PBX són un art diferent per a cada sistema. Els fabricants faciliten manuals, però no donen més suport que aquest i algunes FAQ. Així doncs, a vegades costa trobar tècnics de fabricants concrets. Jo he estudiat dos casos molt representatius, CISCO i Avaya.

Mentre que de **CISCO** trobem molts especialistes per temes de switching, routing o security, són pocs els que estan certificats en CCNP Voice. La falta d'un entorn gràfic amigable en les solucions CISCO fa que només uns pocs puguin extreure tot el rendiment.

Respecte al cas d'**Avaya** la situació és diferent. Té un bon entorn gràfic però, per pròpia experiència puc dir que un suport tècnic per part dels distribuïdors molt pobre (tot i que de ben segur hi haurà honroses excepcions). L'arbre web és poblat i trobar com fer determinades accions varia massa depenent del firmware que estiguin fent funcionar. Hem d'admetre que la solució és potent, però també presenta problemes en accions com desviaments amb determinats proveïdors. Malauradament aquests ajustos, de vegades, necessiten que apareguin nous firmwares per a ser definitivament solucionats. A més, al ser una solució tancada, no hi ha possibilitat de canviar el comportament.

L'altre gran bloc de solucions que no dóna la resposta que busquem es basa en el cloud computing. És un problema dependre de que el proveïdor de VoIP faci l'hospedatge també de la centraleta. Això lliga al client amb un proveïdor concret, i en cas d'haver-hi algun tipus de desacord, canviar de proveïdor és un procés costós a nivell de temps i recursos. A nivell de temps s'ha de clonar el comportament de la centraleta en funcionament a un altre de la nova companyia (a vegades amb funcions no equivalents). A nivell de recursos, si es vol minimitzar el temps sense servei durant el canvi, ens veiem obligats a pagar els dos serveis per tal que se solapin fins comprovar el funcionament total. Per tant, una persona de l'empresa client haurà d'invertir hores en la coordinació durant el canvi, cosa que també és un consum de recursos humans. Per empitjorar la situació no solen ser especialistes en el tema i cometen errors típics durant les portabilitats dels números d'un proveïdor a un altre, tals com demanar portabilitats de números associats a la ADSL (a vegades utilitzats per un Fax) del client i quedar-se sense internet a l'executar-la. Per aquests motius, que admeto que no són purament tècnics, no és recomanable la solució cloud.

Entrem una mica més en matèria en el bloc de solucions software amb servidor al client. En aquest punt trobem una antiga disputa entre els administradors de sistemes, "Windows Server o Linux Server?"[8]. Certament estem buscant una centraleta software, però pensem un moment en el fet que serà un sistema que estarà en servei 24/7. Els reinicis de maquinari són un luxe tremendament car, cosa que afecta més a Windows Server que a Linux.
Linux té un consum energètic normalment més petit[9] i la instal·lació sense entorn gràfic maximitza l'estabilitat i optimització (la idea és que si no ho utilitzes, ni consum recursos ni s'espatlla).

Pel que fa a Windows, no ens podem platejar fer córrer una versió que no sigui Server. Els serveis que normalment correran no es poden permetre que ho facin de manera ininterrompuda en un XP per exemple, augmentant encara més els costos de les llicències.

3.7. ANÀLISI DE LES TECNOLOGIES ESCOLLIDES

Dividiré en dues parts l'anàlisi de les tecnologies escollides, els serveis addicionals de xarxa i seguretat i la tecnologia VoIP (basada en SIP). Tal com he indicat a l'inici, al punt 2.3.7, no hi dedicaré gaire temps, doncs ja estan bastament documentats i són molt coneguts. Només faré una breu descripció, indicant el que s'espera de cadascun d'ells.

En canvi, la tecnologia que hi ha darrera la VoIP mereix un estudi més detallat. Per a aquesta tasca aprofitaré bàsicament els coneixements propis i documents generats per mi mateix sobre telefonia IP durant els darrers tres anys de feina. Tot i que el projecte és més ambiciós que el coneixement present, sí que podré aprofitar l'experiència a nivell de funcionaments de protocols i topologies típiques.

3.7.1. SERVEIS ADICIONALS DE XARXA I SEGURETAT

- **Servidor DHCP**: es el servei encarregat d'assignar IPs lliures del rang que es vulgui configurar, la ruta per defecte o gateway, la màscara de xarxa, el servidor DNS, el servidor NTP... En principi és capaç de passar de forma automàtica molts altres paràmetres o configuracions, però nosaltres només ho utilitzarem per al que hem descrit. Farem servir el isc-dhcp-server per a Debian.

- **Servidor DNS**: aquest servei ens ajudarà a resoldre dominis, proporcionant la IP corresponent. En el nostre cas l'utilitzarem per donar la possibilitat de registrar als terminals un domini en comptes d'una IP, i així facilitant un canvi fàcil d'IP de registre si hi ha una centraleta replicant la primera per casos de crash system. Utilitzarem el servidor bind.

- **Servidor VPN**: gràcies a la VPN podem unir dos punts remots de forma segura, encriptant tota la informació de forma que només és comprensible per als dos extrems del túnel VPN amb openVPN. Utilitzem la generació de certificats per a garantir que es reconeguin inequívocament.

- **Servidor SSH**: és un mètode d'accés segur per l'administració remota per consola (Secure Shell). SSH permet fer tunelació de ports concrets un cop dintre de la Shell. D'aquesta manera és que redirigirem els ports 443, 80 i 10.000 per administrar de forma segura la centraleta. Fent això no deixarem exposats els servidors webs d'administració protegits únicament per un user i pass.

- **Servidor NTP**: el seu nom explica la seva funció a la perfecció; es tracta de Network Time Protocol, que utilitzarem per a mantenir en hora la centraleta i els terminals telefònics connectats a la mateixa.

- **Servidor SSL**: em permetrà crear i gestionar certificats per a serveis segurs com el SSH o la VPN. Bàsicament és la implementació de tota la tecnologia criptogràfica sobre la que descansen la resta de serveis segurs.

- **Client Vlan**: per tal de separar a nivell lògic les xarxes de dades i telefonia que comparteixen la xarxa física utilitzem la tècnica Vlan. A través d'un identificador (a nivell ethernet dintre de la capa OSI) es fa una separació lògica i podem crear àmbits de xarxa separats en la mateixa xarxa física. És a dir, dos dispositius amb ID de vlan diferents no seran capaços de comunicar-se encara que un d'ells faci una petició broadcast. Broadcast significa fer una petició a tothom, a tota la xarxa coneguda i desconeguda però accessible per cable o sense fils.

- **Firewall**: un Firewall té la missió de protegir del transit no desitjat, guardant allò que no ha de ser accessible des de dintre o des de fora de la xarxa. Nosaltres implementarem ShoreWall.

- **IDS**: treballa molt en relació amb el Firewall. El segon es preocupa que no es pugui traspassar i el primer mira de llançar alarmes o logs de comportaments que poden implicar que ha succeït una intrusió o un fet que volem documentar. Els IDS tenen la possibilitat de reaccionar de forma intel·ligent als intents d'intrusió (ja que poden revisar el contingut del tràfic a nivell d'aplicació si cal) metre que els Firewalls tenen un comportament rígid que s'espera que eviti la intrusió, això sí, amb un cost computacional més petit. Encara que no el farem servir gaire en el prototip el servidor IDS Snort pot deixar observat aquest requeriment.

3.7.2. SERVEIS VoIP NO BASATS EN SIP

Un cop fet aquest ràpid anàlisi sobre el que són i el que s'espera de les tecnologies addicionals, serem més exhaustius respecte la tecnologia que s'amaga darrere els serveis VoIP.

Per tal d'unir dos punts en telefonia IP[10] ambdós extrems han de parlar un llenguatge comú. Aquest es parla - sovint - sobre el protocol de transport UDP que és el més utilitzat per l'streaming (donat que les capçaleres UDP tenen 8 bytes i les TCP 20 bytes).Per una qüestió d'eficiència, normalment utilitzem UDP. Això fa que l'ample de banda necessari en tasques de capes inferiors sigui menor encara que, precisament, provoca part de la manca de fiabilitat en el control de fluxos. La funció dels protocols de VoIP és autentificar els extrems, intercanviar la informació necessària per a l'streaming, i encaminar de forma inequívoca les trucades a través d'equips intermedis. Actualment per a VoIP els més estesos són SIP, IAX i SCCP (protocol propietari Cisco).

També hi ha empreses, com Skype, que han desenvolupat el seu propi protocol propietari. Aquest tipus de serveis òbviament no tenen la visió de màxima compatibilitat sinó l'interès partidista de crear una posició en el mercat. I és per aquest motiu que no comparteixen el codi dels protocols.

IAX és un protocol obert, fet a mida per Asterisk, però amb certes indefinicions al no tenir un grup de treball tan tancat com SIP. Té problemes amb la majoria de servei IP que es presta, per la qual cosa és SIP qui ha guanyat la partida a IAX. Malgrat tot, IAX s'utilitza molt per a crear troncals entre centraletes Asterisk. SIP s'ha imposat tenint una pròpia RFC, no ambigua, i força estandarditzada. Tots aquests protocols solen tenir el seu punt feble en els problemes amb la NAT, que venen en part heretats de la capa de transport. SIP implementa en la seva senyalització solucions per evitar els problemes d'enrutament darrere un NAT.

Tot això, malgrat tot, és un incís per donar una visió més àmplia, però que no desenvoluparem. Nosaltres utilitzarem bàsicament SIP, el funcionament del qual sí que explicarem en el següent apartat.

3.7.3. SERVEIS VoIP BASATS EN SIP

El protocol SIP va néixer en la seva primera versió l'any 1996, sent acceptat com a protocol de senyalització per als principals grups de desenvolupament de tecnologia mòbil i element permanent d'arquitectura IMS[11] (IP Multimedia Subsystem) l'any 2000.

Em centraré en SIP, ja que és el més estès entre tots els fabricants i és el que ofereixen la immensa majoria de proveïdors de telefonia IP al mercat (tenint en compte les excepcions abans esmentades). A tall d'aclariment, trobareu a l'Annex 1: SIP, la definició d'una estructura bàsica d'un proveïdor SIP, un exemple de registre i finalment una trucada des d'un client a través d'un proveïdor SIP com el que hem definit.

3.7.3.1. ESTRUCTURA D'INTERACCIÓ DE PROTOCOLS

La pila de protocols per a VoIP utilitzant SIP és la següent:

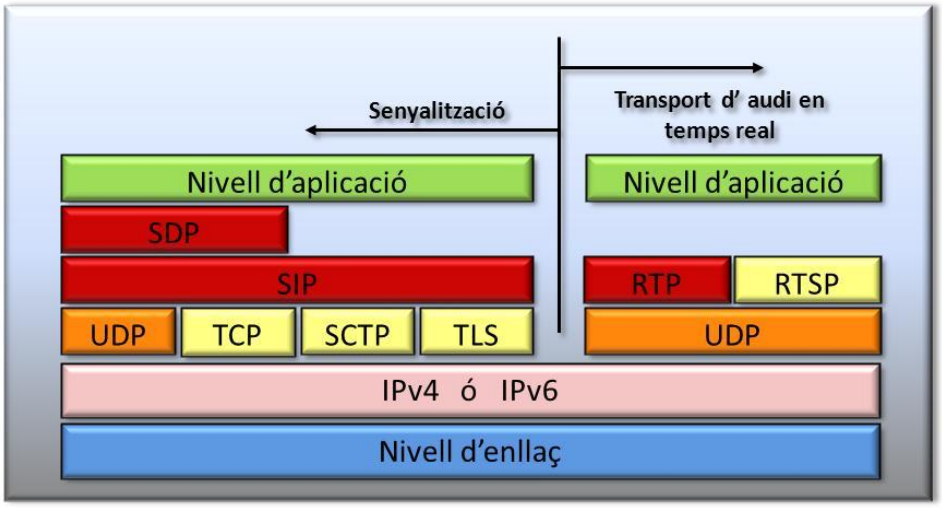

Figura 5: Pila de protocols SIP

Podem veure com en realitat SIP treballa al mateix nivell amb RTP per a transportar l'àudio. La part d' SDP s'implementa a sobre de SIP per controlar valors tals com el temps de sessió activa o diversa informació sobre la connexió multimèdia. Veiem que es pot utilitzar TCP, TLS per al transport però el més habitual és fer-ho sobre UDP. L'RTSP tampoc és necessari per a veu; tota la negociació la fa SIP amb SDP i l'RTP farà directament el transport.

Habitualment, al fer una captura SIP, veurem que l' SDP s'inclou dintre del paquet SIP amb variables de tipus a= , t=,... Normalment, treballant amb Wireshark[12], es veuen aquests valors com a SIP i et marca "(SDP)" quan fas un gràfic de trucada avisant que s'implementa. L'RTP l'identifica com a tal, encara que a vegades s'ha d'agafar el paquet UDP i descodificar-ho com a RTP a partir del número de port UDP. En quan es descodifica es pot observar el codec en el tipus de dades i el nombre de seqüència per ordenar l'streaming.

El protocol SIP[13] es vertebra sobre la idea de query i resposta esperada. Si aquesta resposta no arriba es fan reenviaments, un cert nombre de vegades, durant un temps determinat. En cas de seguir sense resposta, un dels extrems comença a senyalitzar el tall de la trucada amb un error en cas de servidor o un CANCEL. En les sol·licituds s'envien les dades necessàries per tal que la màquina que respon a la sol·licitud ho pugui fer correctament, tant en el contingut de la resposta com en l'enrutament que ha de fer la resposta per arribar a destí.

Bàsicament es tracta d'una negociació en la que, després d'un procés d'autentificació amb un servidor assentarem les bases de l'streaming d'àudio/video, codificació, direccions IP i ports. Quan els dos extrems estiguin preparats, i un cop despenjat el telèfon que rep la trucada, la comunicació serà efectiva. El model que estudiarem a l'annex serà com a clients d'un proveïdor de servei VoIP.

Durant l'streaming utilitzarem normalment el protocol RTP a fi d'enviar l'àudio i serà la senyalització SIP la que enviarà el senyal de penjar la trucada. Bàsicament tota aquesta informació ve codificada en diferents camps definits a l'RFC. Són molts els qui el comparen amb HTTP. En realitat SIP està implementat amb trossos d'altres protocols i està pensat per a col·laborar a la perfecció amb l'RTP, que realment (com hem dit abans) és el que porta l'àudio.

SIP ha unificat la telefonia IP arreu del món, però en el nostre país, a causa de la baixa qualitat dels accessos a Internet, patim un retard important en VoIP respecte d'altres països, sobretot EEUU i el nord d'Europa. L'empenta de la veu IP ha estat molts anys un miratge degut al retard en les infraestructures, tot i que a nivell protocol·lari està provat àmpliament des de fa molts anys. Afortunadament, aquest retard s'està escurçant molt.

3.7.3.2. CODIFICACIÓ DE L'STREAMING D'ÀUDIO

Un cop fet un dibuix general sobre els Protocols ens enfocarem en l'estrella del streaming, el codec. La finalitat és enviar àudio d'un punt a un altre i per a fer-ho hem de transformar el so en bytes. Actualment en la telefonia digital (fixe i mòbil) també transformen el senyal analògic en bits per tal de millorar la qualitat de la veu, igual que en VoIP. La idea general és que, en primer lloc, rebem el so analògic recollit com la oscil·lació de l'aire que movem al parlar en un micròfon. Ara imaginem, conceptualment, que tenim tot l'àudio en forma de gràfica contínua amb els valors de les vibracions o, dit d'un altra manera, freqüències.

El primer que determinen els codecs són els valors màxims i mínims representats, ja que no totes les freqüències són audibles per l'home. Es redueix el marge representat per tal d'aconseguir que l'àudio sigui comprensible sense gaire diferència de qualitat. Per tal d'il·lustrar això, imaginem que tenim una freqüència de 100 Khz (és un valor imaginari qualsevol) com a límit superior. Nosaltres imaginem que podem escoltar fins a 110Khz, però realment si arrodonim valors >100 Khz a 100 entenem el que es parla sense problemes.

Un cop definim els límits, decidim el mostreig, i això ho farem seleccionant una mostra cada X temps. Començant per la primera mesura agafem el valor de la gràfica cada X temps, transformant el so en una seqüència de valors numèrics que podem codificar a bytes. Òbviament, com més grans siguin els límits i, sobretot, si fem mostreig cada menys mil·lisegons, la quantitat d'informació donarà més qualitat, però ocuparà més temps de computació i ample de banda necessari per transmetre l'streaming. Aquesta explicació és simplificada, ja que no és el tema del treball, però és important saber de què parlem. Realment, per transferir la informació de la gràfica de freqüència a bytes s'utilitzen diferents tipus d'algorismes de compressió per tal d'optimitzar encara més. Per tenir una imatge dels codecs més habituals i les seves característiques més importants, facilito una taula comparativa d'una web especialitzada[14].

A la majoria de plataformes s'accepten en la negociació SIP diferents tipus de codecs i per a transportar el senyal des de la telefonia tradicional cap a les plataformes de VoIP s'utilitzen el que anomenem GateWays (terme que abusa una mica del llenguatge empreat a les xarxes).

També hi ha targeteria que es pot implementar a les centraletes directament per a fer aquesta tasca si es vol. Això fa que una trucada provinent de mòbil, que ens arriba en codec GSM, la transformem en G729a per hardware. Aquesta mena de processos tenen uns costos computacionals que poden arribar a ser molt grans en centraletes de servei massiu. A més del cost, la qualitat en la conversió es pot veure compromesa. Per aquests motiu, cada cop és més habitual l'ús de targetes específiques per a fer transcoding i alliberar les CPUs de les centraletes per a l'enrutament de les trucades, les polítiques de marcatge, l'IVRs, el control de canals, el control de diferents interfícies i poc més. En la majoria d'empreses petites i mitjanes s'utilitza només la capacitat de càlcul de la pròpia centraleta. Aquesta tendència es pot invertir si segueixen abaixant el hardware de transcoding com s'està fent.

Els problemes de qualitat d'àudio a vegades tenen a veure amb el fenomen del transcoding. És habitual veure que el volum del so és més baix que l'original quan es fa transcoding del GSM al G729 en alguns GW que s'ha de compensar augmentant el guany en el mateix GW, equips de Proxy, centraletes o fins i tot terminals finals.

Tot això fa que l'art d'afinar la qualitat entre equips heterogenis amb diferents codecs i el guany per defecte que depèn de diferents fabricants no sigui una ciència exacta. Es tracta, més aviat, de la recerca de la virtut i l'equilibri, ja que, per exemple, si ens passem en el guany comencem a tenir problemes per l'acoblament en els terminals.

Així, per totes les qüestions que hem plantejat, l'opció més intel·ligent és unificar el codec utilitzat el millor possible entre els dos extrems de la conversa. Com veurem, la negociació SIP intenta això sempre que és possible i tots els equips comparteixen entre les seves opcions un codec en comú. En SIP no només s'ofereixen els codecs acceptats, el que inicia la trucada, fins i tot indica l'ordre de preferència del que es vol utilitzar. La importància dels codecs en la VoIP és cabdal per a fer possible una telefonia de qualitat depenent dels requeriments i l'ample de banda disponible.

A vegades, a fi d'optimitzar encara més l'ample de banda, es pot fer servir la supressió de silencis, per a no enviar cap streaming quan no es parla, però acostuma a donar problemes en fer el "wake up". S'opta per no utilitzar-lo molt, per evitar problemes que fan que un dels extrems interpreti que la trucada s'ha perdut al no arribar-hi res. En aquest cas es poden observar en les captures ratxes d'RTP sense respostes en un dels extrems.

3.7.3.3. NECESSITATS I CONSIDERACIONS SOBRE L'ACCÉS A INTERNET

Farem una menció especial a les característiques que ha de tenir un accés a Internet per a VoIP. Habitualment quan pensem en l'ús que fem de la nostra connexió, mirem si baixa a molts MB/s per valorar si tenim una bona. Això es un punt de vista molt simplista quan parlem de la VoIP, encara que vàlid per baixar algun programa, navegar o escoltar Spotify(es un streaming en un únic sentit amb un buffering molt importat).

Quan parlem de telefonia IP el primer que hem de pensar és que el consum d'ample de banda és simètric. Necessitarem el mateix ample per parlar que per escoltar.

En les plataformes en les que tinc experiència utilitzem com a codec el G711a/u i el G729 principalment. Els primers per la qualitat i el G729 per la compressió. Com utilitzem un ptime o grandària de frame de 20 ms. trobem que, amb el G729, utilitzem uns 18-19 KB/s i el G711 uns 64 KB/s. Arrodonim (molt a l'alça) a 40-50 i 100 KB/s respectivament pel que suposa la suma de les diferents capçaleres (RTP,UDP,IP, Ethernet).

La percepció d'un usuari no acostumat a VoIP amb la seva ADSL de 3MB és que no entén perquè no pot fer més de tres trucades simultànies. El G729 té llicencia de pagament, i si utilitzem G711 per fer 3 trucades ocupem, arrodonint, uns 300 KB/s. El més habitual en una ADSL 3MB són 320 KB/s de pujada, i aquí observem un cas típic de un dimensionament erroni de l'accés.

Suposant que els 320 teòrics de sincronisme donin el 100% real (normalment el rendiment real sol ser al voltant del 80%) si canviem de codec a G729 podríem fer 6 trucades simultànies en comtes de 3. En VoIP és habitual l'ús de SHDSL. És una connexió d'ampla de banda simètrica (amb mateixa pujada que baixada) per obtenir més pujada encara que la baixada no sigui tan gran. Això s'aconsegueix repartint equitativament les freqüències de la DSL entre pujada i baixada. Així multiplexem els canals 50% i 50% no com l'ADSL que dóna un percentatge molt més gran a la baixada.

Un altre tema de vital importància té lloc quan l'accés a Internet no és exclusiu. Altres ordinadors i dispositius poden estar competint amb la centraleta per l'ample de banda disponible. I això posa en relleu el QoS (Quality of Service). En accessos compartits, les polítiques de QoS garanteixen un ample de banda mínim per a aplicacions determinades. En el nostre cas, si volem garantir que es podran fer 2 trucades en G711, haurem de garantir 200 KB/s simètrics de la nostra connexió (per una mica més de seguretat). Sense QoS, un enviament d'un correu amb un adjunt generós pot deixar sense ample de banda la conversa en curs, donant problemes de qualitat d'àudio i fins i tot talls. Tenint en compte que un streaming de VoIP ideal és un flux constant d'àudio en temps real, hem de tenir molt en compte dues situacions: la latència i la pèrdua de paquets. Es pot fer una prova de las latències i pèrdues al link que referencio[15].

En un escenari d'streaming en temps real, la pèrdua de paquets, tenint en compte que utilitzem UDP (que no espera una confirmació d'entrega) i la pèrdua de trames no se soluciona normalment degut a l'ideal de buffer proper a zero per evitar retards. Com que cada paquet codifica un temps molt petit d'àudio (normalment s'utilitza 20-30 ms) les pèrdues comencen a ser un problema insalvable quan sobrepassem el 2 %. Aquestes normalment es poden monitoritzar amb un programari de tracert continu, com MTR, utilitzant paquets grans propers al MTU (grandària màxima d'una trama sencera completament plena) per a auditar la connexió.

Pel que fa a la latència massa alta, podem tenir un problema de retards en la veu, provocant en la conversa fluida que els dos interlocutors es trepitgin, parlant abans d'escoltar la frase que encara està viatjant cap a ells (i en major grau si és multiconferència). Les latències ideals són per sota de 100 ms i comença a ser un problema greu per sobre de 200 ms. Hem de pensar que les latències que mesurem, igual que les pèrdues, són contra l'altre extrem de la conversa, no cap a un servidor del nostre ISP. Aquest aspecte és especialment delicat en trucades a països llunyans, com Austràlia, o zones subdesenvolupades a nivell d'infraestructures. Per això últim podem pensar en països propers africans, però podem mirar ben a prop, com a segons quins poblets de Catalunya. Un altre fenomen molt estudiat de la latència relacionat amb VoIP és el Jitter (fluctuació)[16].

Si la latència està per sota de 100 però varia de 30 a 100 ms. formant ratxes de paquets amb diferents latències, els paquets acaben arribant desordenats, massa aviat o tard per ser entregats. Es pot minimitzar aquest problema utilitzant un buffer. Si la fluctuació és molt gran i constant, el buffer necessari ens provocarà un retard que no podrem assumir. Estaríem en el mateix cas de latències grans, els interlocutors es trepitjarien i sentirien una frase mentre ells responen a l'anterior. No és assumible un jitter amb fluctuacions de més d'un 20% d'un paquet a un altre.

3.8. ANÀLISI DE LES TECNOLOGIES ESCOLLIDES

El gràfic que descriu el producte final, diferirà una mica del nostre prototip (que descriurem durant la fase de proves) però ens ajudarà a entendre el que estem fent.

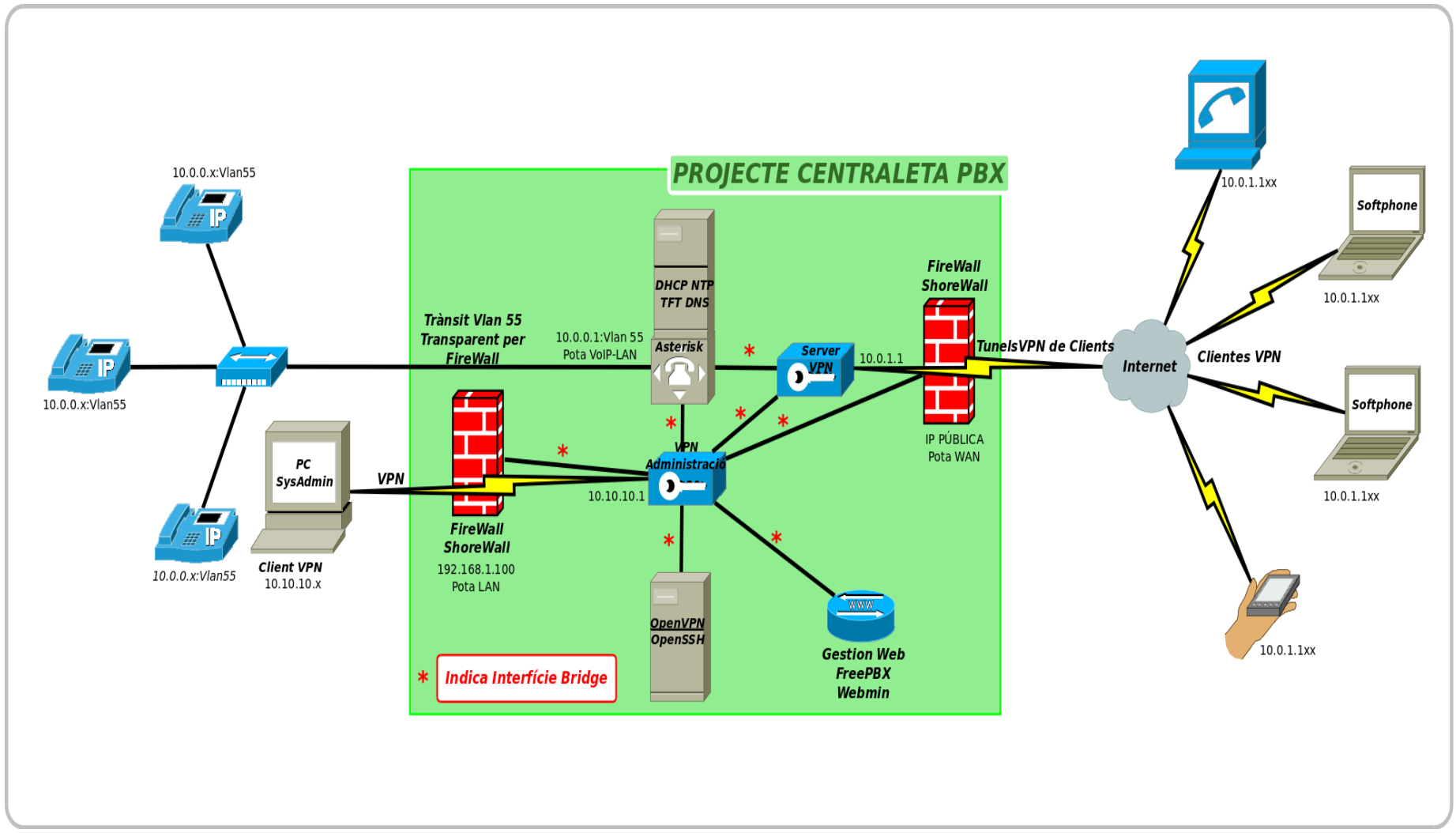

Figura 6: Gràfic del projecte "Centraleta PBX" en una implementació real

En quant al disseny de la infraestructura base és molt senzill. Tot girarà entorn d'una màquina virtual sobre VirtualBox amb una Interface en mode bridge.

Partirem d'una distribució Debian 6 de 32 bits (al disposar de poca memòria RAM per a la màquina Virtual – VM en endavant – no podem utilitzar la de 64 bits, a la que li calen 4GB de RAM mínim). Durant la instal·lació ens demanarà password de root i nom de màquina. Jo he utilitzat user= root i pass=toor per facilitat, encara que no és un password segur, per al nostre projecte. Configurarem eth0 amb configuració d'IP fixe. També crearem una Interface etiquetada vlan 55 que utilitzarem per a telefonia LAN. La configuració de xarxa es pot introduir així:

```
apt-get update
apt-get upgrade
apt-get install vlan
echo 'auto lo
iface lo inet loopback
allow-hotplug eth0
iface eth0 inet static
        address 192.168.1.100
        netmask 255.255.255.0
        gateway 192.168.1.1
auto eth0.55
iface eth0.55 inet static
        address 10.0.0.1
        netmask 255.255.255.0' > /etc/network/interfaces
echo 'nameserver 8.8.8.8
nameserver 8.8.4.4' > /etc/resolv.conf
```

Habilitem el Routing entre interfícies per a facilitar el tràfic dintre de la centraleta i reiniciem:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
reboot
```

En quant als serveis de telefonia, l'estructura de software per a VoIP es basarà en els divers DAHDI i el core Asterisk. Com a interfície gràfica, descansarà sobre l'Asterisk una FreePBX.

En realitat, FreePBX només és una interfície; també s'encarrega d'instal·lar mòduls optatius que faciliten l'administració i donen més funcionalitats. FreePBX ens ofereix una plataforma que integra totes les característiques avançades de centraletes hardware de molts milers d'Euros de forma gratuïta.

La implementació, com veurem en el següent punt, té els seus secrets, però el resultat final és realment robust i una solució potent.

Tant Asterisk com FreePPX utilitzen mySQL per a guardar les dades no volàtils com trucades, desviaments, extensions, etc...

Per una altra banda FreePBX-que utilitza llenguatge PHP i PERL per a fer la feina-sobreescriu els arxius de configuració d'Asterisk menys uns concrets anomenats "custom" que permeten a Asterisk configurar lliurement opcions programades. Tots aquests arxius de configuració acaben creant una xarxa a través d' "include" que uneix la configuració sencera.

En paral·lel als serveis de VoIP treballen cooperativament els serveis de xarxa com el DNS, DHCP, NTP, FireWall, SSH, OpenVPN. Els serveis ShoreWall – el nostre Firewall – i OpenVPN tenen el Webmin per sobre com a interfície web, treballant pràcticament igual amb aquest serveis a com ho fa FreePBX amb l'Asterisk.

Conceptualment podem definir-ho com es mostra al següent gràfic:

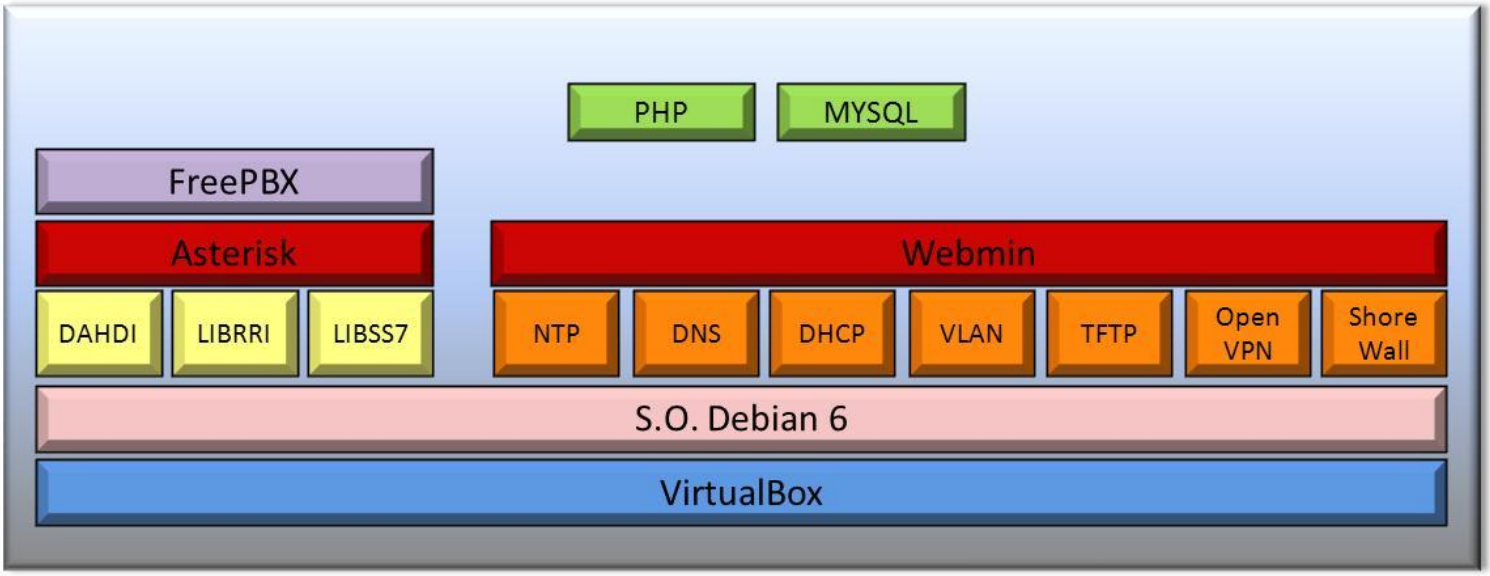

Figura 7: Pila d'interacció del programari

4. IMPLEMENTACIÓ DE TOTS ELS SERVEIS

4.1. INSTAL·LACIÓ DE SERVEIS I FreePBX

En primer lloc instal·lem openssh-server per a poder accedir per consola al servidor que és més còmode per a treballar:

```
apt-get -y install openssh-server
```

Els paquets necessaris per a tota la resta de serveis són:

```
apt-get -y install vim vlan shorewall-perl bind9 openvpn apache2 php5 mysql-server libncurses5-dev make
linux-headers-`uname -r` php5-cli php-pear php-db gcc subversion g++ mpg123 php5-mysql ntp ntpdate
atftpd dhcp3-server libxml2-dev libapache2-mod-php5 php5-gd php-pear sox curl  build-essential libssl-dev
libxml2-dev gawk sudo libapt-pkg-perl libnet-ssleay-perl libauthen-pam-perl libio-pty-perl apt-show-versions
bridge-utils
pear install DB
```

Durant la instal·lació de mySQL configuro quan pregunta l'usuari root password toor (la mateixa del sistema per comoditat, encara que per seguretat haurien de ser diferents).

Tot el llistat de paquets, s'ha hagut de generar a partir de la documentació pròpia d' Asterisk, FreePBX i Webmin. Alguns paquets han estat afegits a l'avisar problemes de dependències durant les instal·lacions. Gràcies al prova/error puc donar una llista suficient per al funcionament dels serveis.

Una vegada instal·lats tots els paquets anem al directori on habitualment es descarreguen els codis font per a compilar després i descarreguem de les webs d'origen:

```
cd /usr/src
# Webmin
wget http://prdownloads.sourceforge.net/webadmin/webmin_1.610_all.deb
# Asterisk 1.8, la última versió certificada.
wget downloads.asterisk.org/pub/telephony/certified-asterisk/certified-asterisk-1.8.11-current.tar.gz
# Dahdi, per a implementar targetes analògiques o rdsi en un futur hipotètic
wget downloads.asterisk.org/pub/telephony/dahdi-linux-complete/dahdi-linux-complete-current.tar.gz
# Libpri, llibreria amb protocols de canals primaris dahdi
wget downloads.asterisk.org/pub/telephony/libpri/libpri-1.4-current.tar.gz
# Libss7, llibreria per Asterisk de protocols
wget downloads.asterisk.org/pub/telephony/libss7/libss7-1.0.2.tar.gz
# FreePBX 2.10, la darrera versió estable.
wget mirror.freepbx.org/freepbx-2.10.0.tar.gz
```

Descomprimim els paquets:

```
tar -zxvf certified-asterisk-1.8.11-current.tar.gz
tar -zxvf dahdi-linux-complete-current.tar.gz
tar -zxvf libpri-1.4-current.tar.gz
tar -zxvf libss7-1.0.2.tar.gz
tar -zxvf freepbx-2.10.0.tar.gz
```

Executem dos scripts que solucionen les dependències requerides

```
/usr/src/certified-asterisk-1.8.11-cert8/contrib/scripts/install_prereq install
/usr/src/certified-asterisk-1.8.11-cert8/contrib/scripts/install_prereq install-unpackaged
```

Ens preguntarà codi de país. Espanya és el 34:

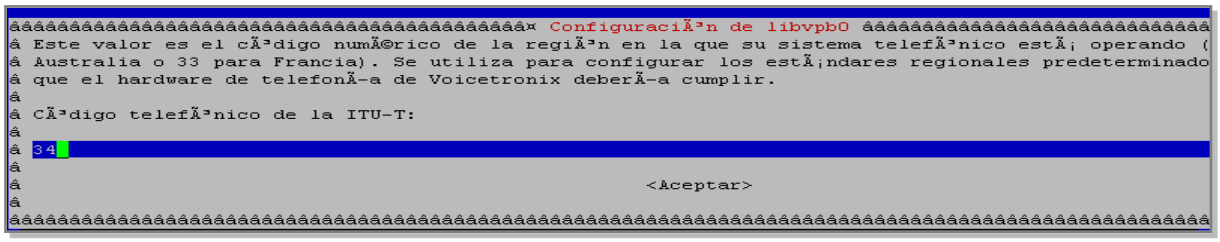

Figura 8: Codi de país per a Asterisk

Instalem Webmin:

```
dpkg --install webmin_1.610_all.deb
apt-get -f install
```

Ara ja tenim funcionant el Webmin, quan acabem d'instal·lar completarem la configuració.

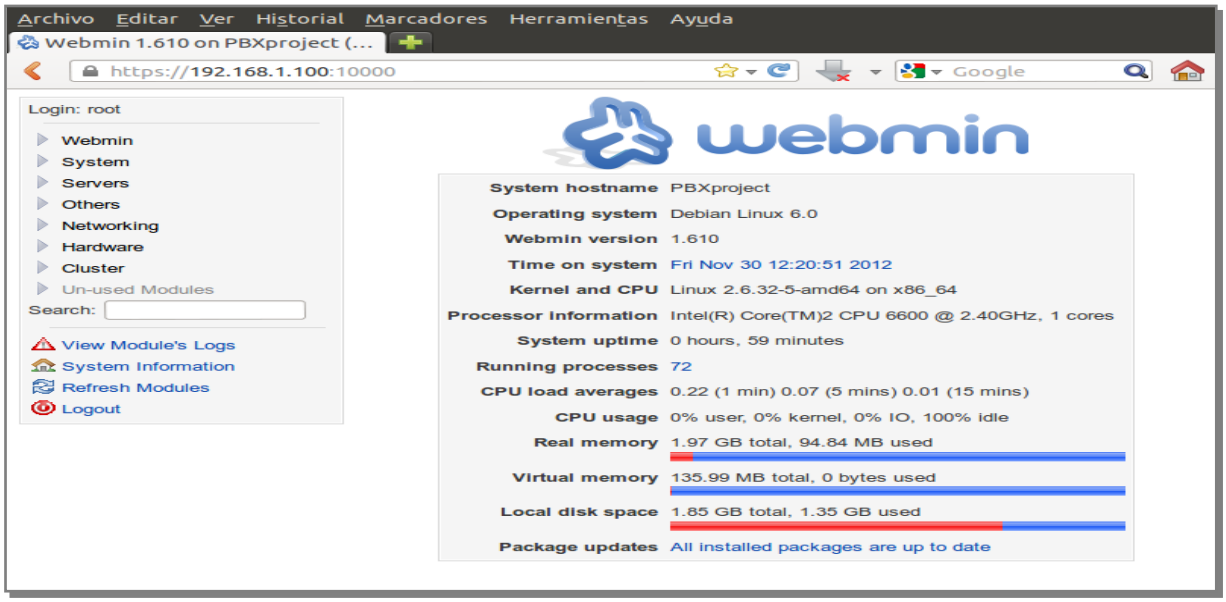

Figura 9: Finestra Webmin

Abans de continuar, farem uns canvis a la configuració del php.ini que ens faran falta per a seguir la instal·lació de Asterisk/FreePBX. Podem editar els canvis a mà, però utilitzo Perl per a automatitzar una mica el procés. Com es pot veure canvio valors de mida d'arxius que puc pujar, ja que tota la configuració d'una FreePBX pot ser molt més gran de 2 MB com a valor per defecte. Faig el mateix amb els temps. Els límits nous són més proporcionals a les mides noves.

L'Últim valor de magic_quotes_gpc és una opció que en la versió 5.3.3 (la que instal·la per defecte dels repositoris Debian Squeeze) encara està en vigor. Si el valor es troba en On (configuració per defecte) cada vegada que es passen els caràcters cometes simples, cometes, barra invertida o NULL , el PHP “escapa” (afegeix la barra invertida al davant, és a dir \' o \” per exemple) els caràcters per que es prenguin com un valor de caràcter normal. El que volem és que prengui valor de caràcter especial per a realitzar funcions al sistema operatiu que ens són necessàries per a guardar la configuració als fitxers del Asterisk des de FreePBX. És per això que canviem el valor a “Off”.

La sintaxi de la crida perl -pi (efectuar una acció sobre un fitxer) és -e per comprovar que existeixi /etc/php5/apache2/php.ini, “s” per dir substituir, /cadena per canviar/cadena per reescriure/, “g” es per que canviï les cadenes en totes les coincidències.

```
perl -pi -e 's/upload_max_filesize = 2M/upload_max_filesize = 40M/g' /etc/php5/apache2/php.ini
perl -pi -e 's/upload_max_filesize = 2M/upload_max_filesize = 40M/g' /etc/php5/cli/php.ini
perl -pi -e 's/max_execution_time = 30/max_execution_time = 120/g' /etc/php5/apache2/php.ini
perl -pi -e 's/max_input_time = 60/max_input_time = 120/g' /etc/php5/apache2/php.ini
perl -pi -e 's/magic_quotes_gpc = On/magic_quotes_gpc = Off/g' /etc/php5/apache2/php.ini
```

Comencem a instal·lar Asterisk:

```
cd /usr/src/libpri-1.4.13/
make
make install
cd /usr/src/libss7-1.0.2
make
make install
cd /usr/src/dahdi-linux-complete-2.6.1+2.6.1/
./configure
make
make install
make config
cd /usr/src/certified-asterisk-1.8.11-cert8/
./configure
contrib/scripts/get_mp3_source.sh
make menuconfig
```

En aquest punt podem habilitar mòduls i opcions de compilació. Nosaltres escollim:

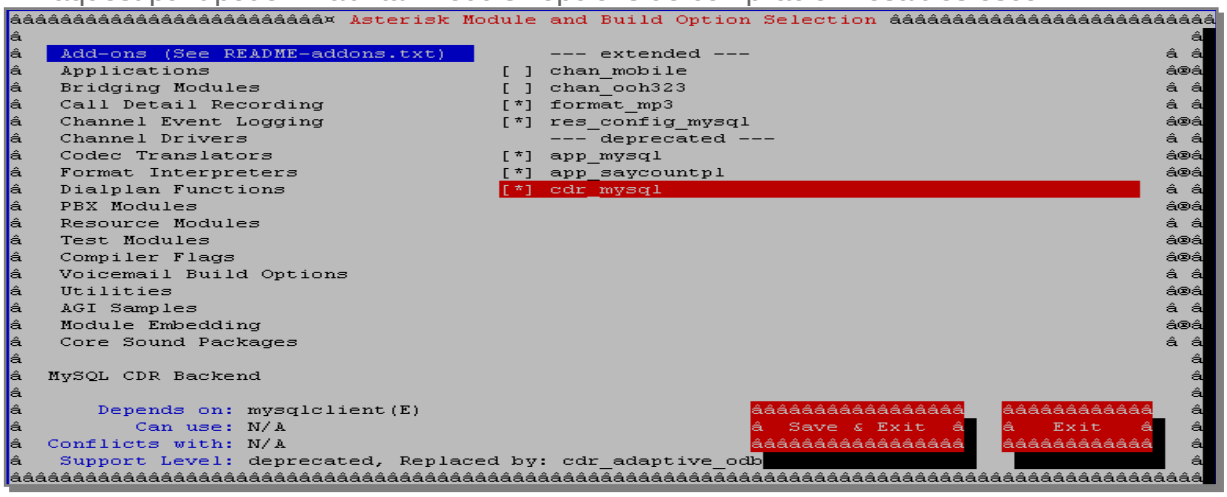

Figura 10: Add-ones Asterisk

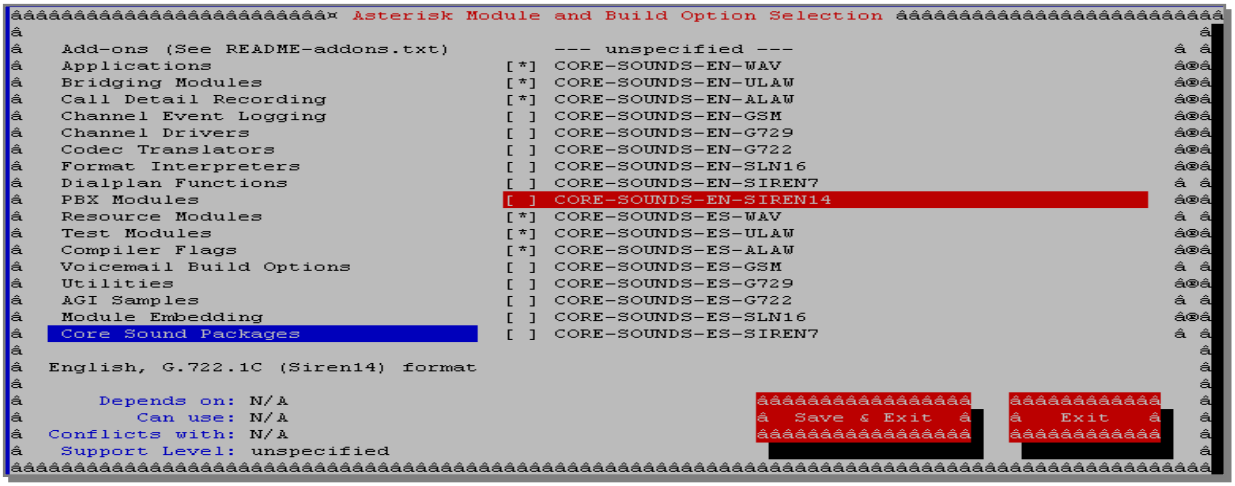

Figura 11: Codecs Asterisk

```
make
make install
make samples
```

Ara ja tenim instal·lat Asterisk, encara que encara hem de fer ajustos amb els arxius de configuració del servei, sistema operatiu i mySQL :

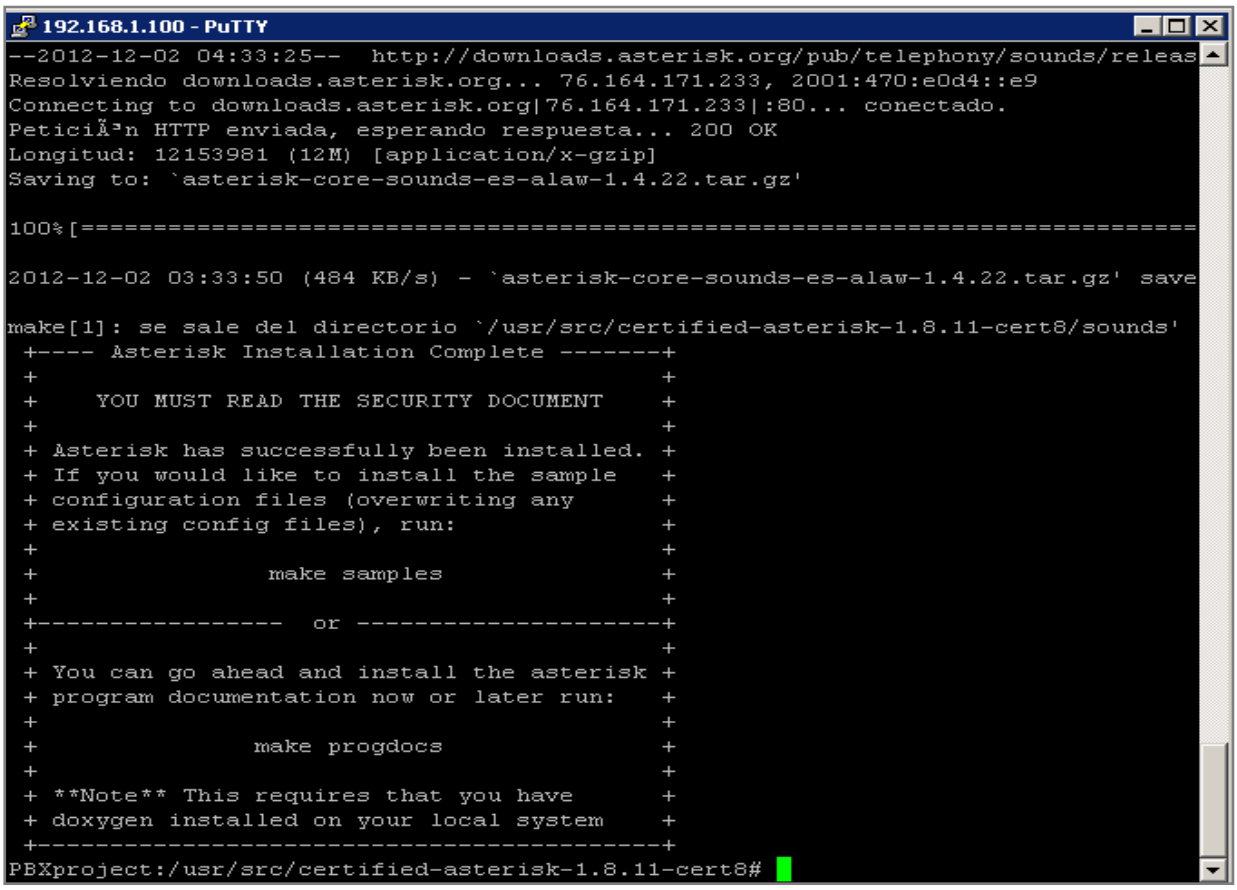

Figura 12: Pantalla final d'instal·lació d'Asterisk

Fem el “make samples” per a generar els arxius de configuració per defecte i ja estem preparats per a començar:

Activem l'arxiu de configuració retirant “(!)” de l’asterisk.conf

```
perl -pi -e 's/\(!\)//g' /etc/asterisk/asterisk.conf
```

Afegim grup i usuari d'asterisk per a poder arrancar els processos propis i canviem la configuració del servidor web per a que l'usuari asterisk pugui gestionar-lo (cosa que necessitarem per a FreePBX).

```
groupadd asterisk
useradd -g asterisk -d /var/lib/asterisk -s /bin/bash asterisk
perl -pi -e 's/www-data:x:33:/www-data:x:33:asterisk/g' /etc/group
perl -pi -e 's/User \${APACHE_RUN_USER}/User asterisk/g' /etc/apache2/apache2.conf
perl -pi -e 's/Group \${APACHE_RUN_GROUP}/Group asterisk/g' /etc/apache2/apache2.conf
```

Anem al directori que hem descomprimit de FreePBX I creem les basses de dades a partir de les plantilles de la freePBX de la subcarpeta ./SQL

```
cd /usr/src/freepbx-2.10.0/
mysql --defaults-file=/etc/mysql/debian.cnf -e "CREATE DATABASE asteriskcdrdb;"
mysql --defaults-file=/etc/mysql/debian.cnf asteriskcdrdb < SQL/cdr_mysql_table.sql
mysql --defaults-file=/etc/mysql/debian.cnf -e "CREATE DATABASE asterisk;"
mysql --defaults-file=/etc/mysql/debian.cnf asterisk < SQL/newinstall.sql
```

Donem permisos a les bases de dades per a asterisk creant password amp109 per a mySQL

```
mysql -uroot -ptoor -e "GRANT ALL PRIVILEGES ON asterisk.* TO asteriskuser@localhost IDENTIFIED BY 'amp109'; FLUSH PRIVILEGES;"
mysql -uroot -ptoor -e "GRANT ALL PRIVILEGES ON asteriskcdrdb.* TO asteriskuser@localhost IDENTIFIED BY 'amp109'; FLUSH PRIVILEGES;"
```

Creem els directoris necessaris I donem permisos per al servidor web I per a Asterisk/FreePBX:

```
mkdir -p /var/www/admin/cgi-bin
mkdir -p /var/www/admin/htdocs
mkdir -p /var/lib/asterisk/sounds/custom
chown -R asterisk: /var/run/asterisk/
chown -R asterisk: /etc/asterisk
chown -R asterisk: /var/{lib,log,spool}/asterisk
chown -R asterisk: /var/www/
chown –R asterisk: /var/lib/php5
```

Arranquem Asterisk i instal·lem FreePBX (Asterisk Manager Interface user admin pass amp109)

```
asterisk start
./install_amp --username=asteriskuser --password=amp109
```

Permisos després de la instalació a /var/lib/asterisk ja que la instal·lació el modifica.
chown -R asterisk:asterisk /var/lib/asterisk/

Donem la ruta al servidor web Apache per a trobar continguts de FreePBX

```
perl -pi -e 's/DocumentRoot \/var\/www/DocumentRoot \/var\/www\/html/g' /etc/apache2/sites-enabled/000-default
perl -pi -e 's/Directory \/var\/www/Directory \/var\/www\/html/g' /etc/apache2/sites-enabled/000-default
perl -pi -e 's/<a href="index.php">/<a href="admin\/index.php">/g' /var/www/html/index.html
```

Aquest pas és per a configurar targetes que utilitzin el driver Dahdi per a unificar comunicacions tradicionals i analògiques. En el nostre cas, no tinc hardware i el podem estalviar, però com és imperatiu al plec de condicions col·loco la configuració bàsica.

```
/etc/init.d/dahdi start
dahdi_cfg
dahdi_genconf
echo "#include dahdi-channels.conf" >> /etc/asterisk/chan_dahdi.conf
echo "#include chan_dahdi_additional.conf" >> /etc/asterisk/chan_dahdi.conf
```

Reiniciem servidor web i preparem arxiu amportal per a iniciar el servei FreePBX:

```
etc/init.d/apache2 restart
cp start_asterisk /etc/init.d/amportal
```

Afegim al amportal informació personalitzada:

```
touch intermedi
echo "### BEGIN INIT INFO
# Provides:        amportal
# Required-Start:  $local_fs $remote_fs dahdi
# Required-Stop:   $local_fs $remote_fs dahdi
# Should-Start:    $network $syslog
# Should-Stop:     $network $syslog
# Default-Start:   2 3 4 5
# Default-Stop:    0 1 6
# Description:     amportal by R.Sales Project
### END INIT INFO">./intermedi
cat /etc/init.d/amportal >> ./intermedi
mv ./intermedi /etc/init.d/amportal
chmod 754 /etc/init.d/amportal

perl -pi -e 's/stop gracefully/core stop gracefully/g' /etc/init.d/amportal
```

Actualitzem i iniciem el servei:

```
update-rc.d amportal defaults
/etc/init.d/amportal start
```

Preparem el logs per a rotar, per tal que no ens quedem sense lloc:

```
echo "/var/log/asterisk/queue_log /var/log/asterisk/full /var/log/asterisk/messages /var/log/asterisk/cdr-csv/Master.csv{
daily
missingok
rotate 7
sharedscripts
postrotate
/usr/sbin/asterisk -rx 'logger reload' > /dev/null 2> /dev/null
endscript
}" > /etc/logrotate.d/asterisk
```

Augmentem el nombre d'arxius que pot tenir obert root I asterisk a /etc/security/limits.conf:

```
perl -pi -e 's/# End of file//g' /etc/security/limits.conf
echo "# root - nofile 1000000
# asterisk - nofile 1000000
# End of file" >> /etc/security/limits.conf
```

Revisem efectivament el funcionament de la centraleta operativa que ens permetrà efectuar proves de funcionament per a la següent fase. Aprofito per a canviar password per “admin/admin” amb l’entorn de configuració web i ho comprovo.

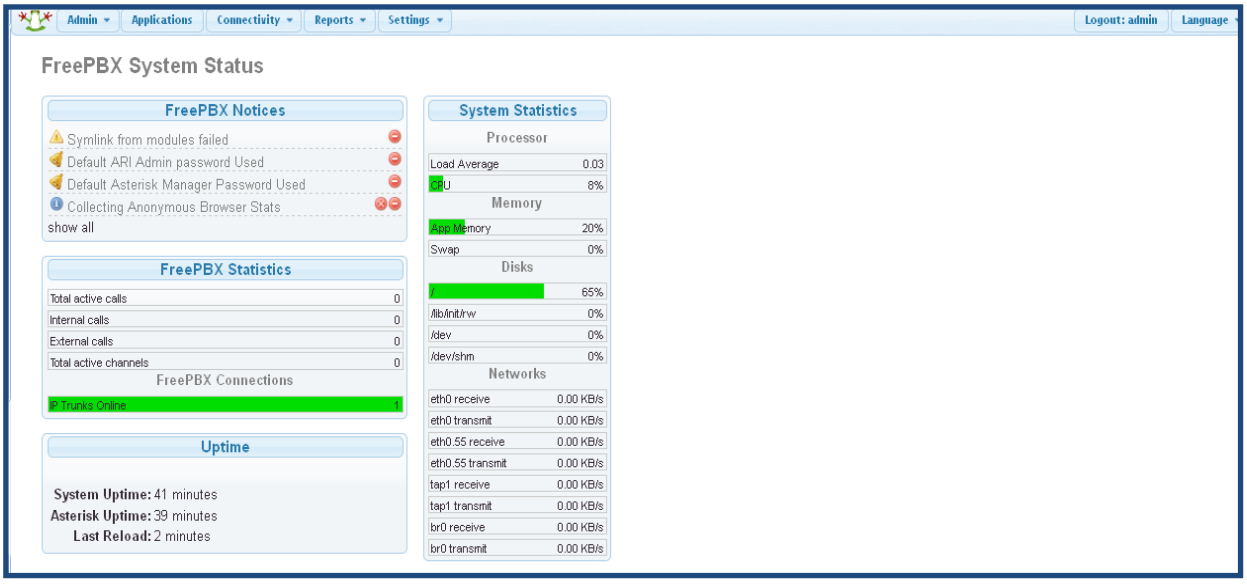

Figura 13: Status de la FreePBX

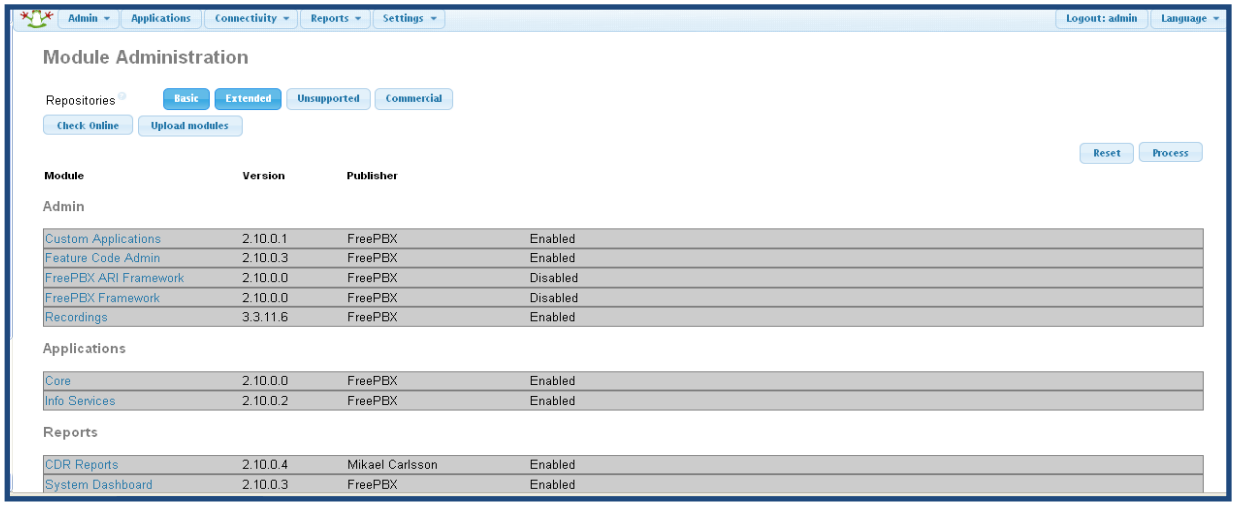

Figura 14: Instal·lació de mòduls FreePBX

4.2. CONFIGURACIÓ DE SERVEIS ASSOCIATS

4.2.1. NTP

Comencem amb ntpdate –u hora.rediris.es per a posar a hora el servidor. Configurem el servidor d'hora.rediris.es al /etc/ntp.conf. Restrigeixo a la ip de broadcast de la Vlan el servei ntp. Inserto al /etc/ntp.conf:

```
broadcast 10.0.0.255
disable auth
broadcastclient
```

La seguretat la gestionaré amb el Shorewall enfora y la Vlan de telefonia la consideraré una xarxa segura i controlada a través dels switchs.

4.2.2. DHCP

Editem /etc/dhcp/dhcpd.conf I afegim:

```
subnet 10.0.0.0 netmask 255.255.255.0
{
range 10.0.0.30 10.0.0.50;
option subnet-mask 255.2255.255.0;
option broadcast-address 10.0.0.255;
option domain-name "PBXproject.com";
option domain-name-servers 10.0.0.1;
option routers 10.0.0.1;
}
```

4.2.3. DNS

Editem /etc/bind9/named.conf.options per a configurar el reenviament de solicituts.

```
forwarders {
        8.8.8.8;
        8.8.4.4;
}
```

En el /etc/resolv.conf poso com a servidor localhost: “nameserver 127.0.0.1”. Després afegeixo l’entrada 192.168.1.100 PBXproject.com a /etc/host.conf.

4.2.4. TFTP

Editem /etc/default/atftpd.conf i modifiquem un valor:

```
--mcast-addr 10.0.0.0-255
```

De moment deixo el directori arrel del servidor al directori /srv/tftp que és el que té per defecte. En principi ens serveix per a carregar configuracions i firmware de telèfons físics, però en el nostre projecte no comprarem terminals.

4.2.5. SHOREWALL FIREWALL

Editem l’arxiu /etc/sysctl.conf i descomentem la línia que permet enrutament entre interfícies per al Shorewall. Es crea una configuració provisional que s’haurà d’afinar en el període de proves per a donar el punt òptim d’usabilitat/seguretat.

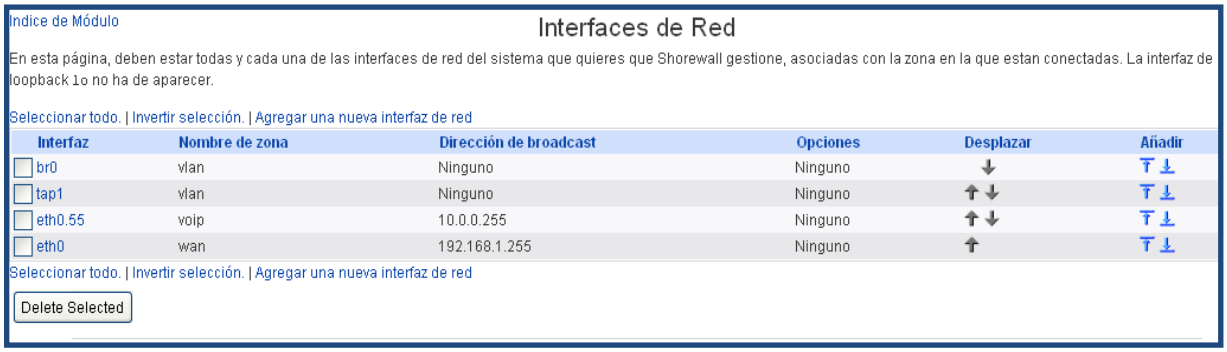

Figura 15: Interfícies de xarxa

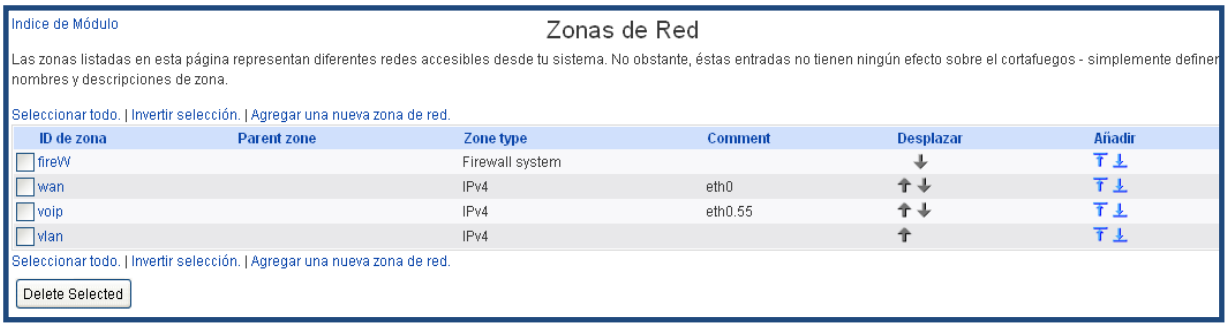

Figura 16: Zones de xarxa

Figura 17: Polítiques per defecte

Figura 18: Regles de Firewall

4.2.6. OpenVPN

Per a carregar al webmin el mòdul OpenVPN (si no el carrega sol a la instal·lació), buscar en Webmin/WebminConfiguration/Webmin Modules/Third party modules per a carregar el mòdul. El primer que s'ha de fer es crear una Autoritat Certificadora (CA en endavant):

Figura 19: Configuració de l'autoritat certificadora (CA)

Triga molta estona en generar la clau i he optat per una de 1024 per estalviar temps, però és mes apropiada la de 2048 com a mínim per a un model en producció.

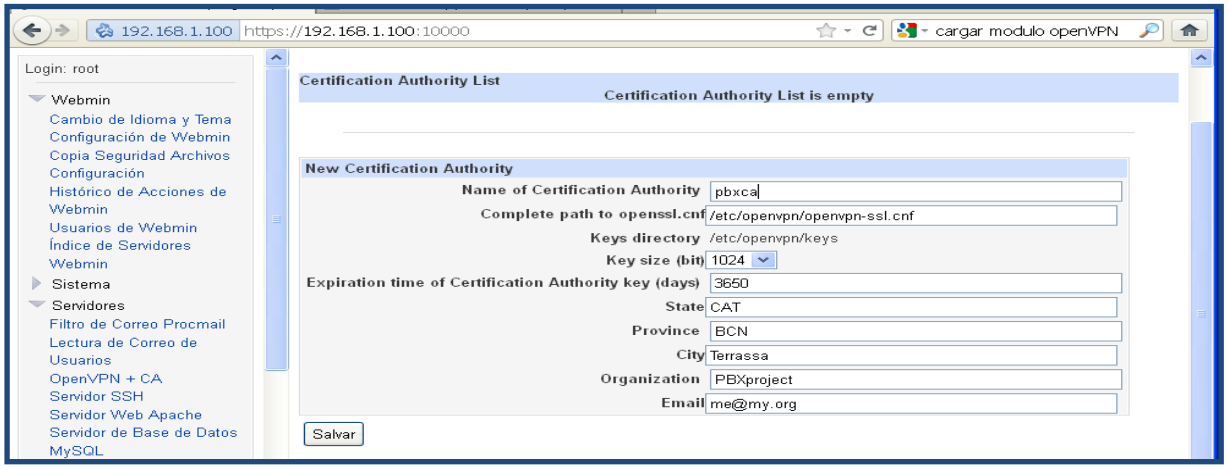

Figura 20: Generació de les claus de la CA

Creada la CA, generem amb la CA las Keys per a servidors i clients:

Figura 21: Generació de claus per a servidors i clients

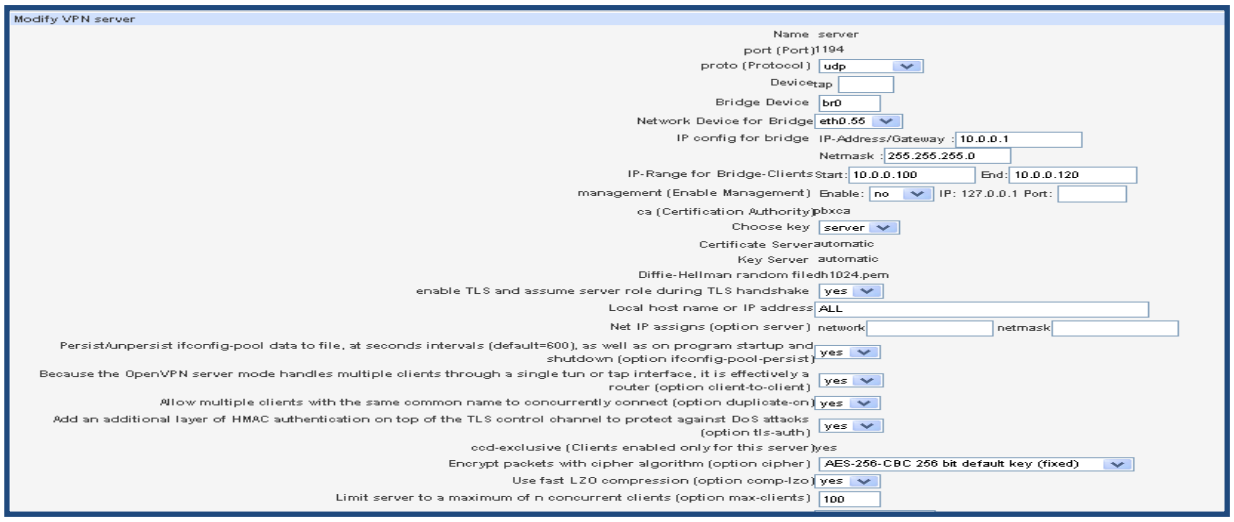

Figura 22: Configuració del Server VPN

5. PROVES I RESULTATS

5.1. MAQUETA DE PROVES

5.1.1. LLICÈNCIES RESPECTE EL MODEL INICIAL I L'ENTORN DE PROVES

En el cas del nostre projecte, en una implementació real, necessitaríem un servidor amb un parell de targetes Ethernet per a separar físicament LAN i WAN. Respecte als requeriments d'electrònica de xarxa i dispositius, tots haurien de suportar la configuració de Vlan (estàndard 802,1q) . La majoria de serveis auxiliars només estarien disponibles des de la interfície LAN amb etiquetatge Vlan.

L'únic servei que és accessible desde la pota WAN hauria de ser la VPN. D'aquesta manera la configuració i l'accés al nostre Servidor PBX només es podria fer físicament en local desde dintre de l'empresa que el custodia o a través d'un túnel segur amb xifratge fort.

La seguretat desitjable per a ús en producció és un ShoreWall (el nostre servidor de Firewall) que deixa passar únicament sol·licituds per a connexions VPN des de la WAN. Tota la resta de trànsit per als nostres clients a la LAN, exceptuant el trànsit a les aplicacions Web de configuració (port 80 i 10000) i el SSH, es permetrà. D'aquesta manera des de dintre la LAN tampoc es podrà modificar la configuració a no ser que es tingui un client OpenVPN per a poder accedir-hi de forma autoritzada i segura.

En el nostre entorn de proves, he relaxat molt la seguretat. Els continus canvis i proves no fan òptim el bloquejar tot el trànsit, així que per a les proves he parat el Shorewall. Els gràfics físics i lògics de la maqueta ajudaran a entendre el perquè d'aquestes limitacions. També explicaré el perquè de prescindir de la part de configuració Vlan, ja que amb les eines que he pogut utilitzar no ha estat possible.

5.1.2. GRÀFIC DE LA MAQUETA

La maqueta física sobre la que treballo és la següent:

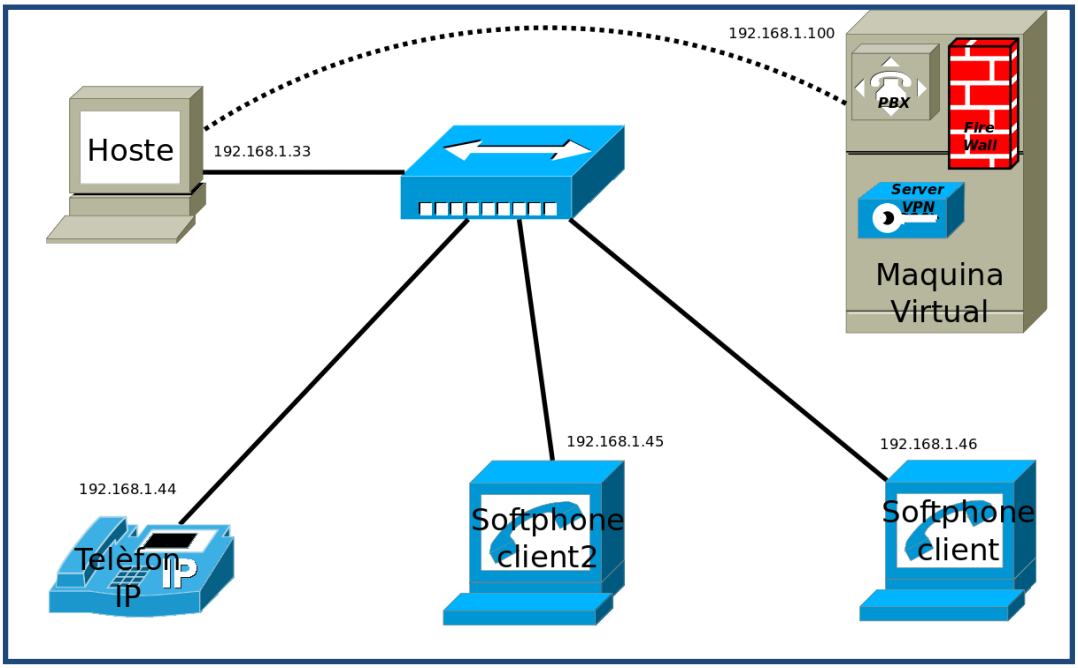

Figura 23: Representació de la maqueta de proves físiques

Malgrat tot, per tal de simplificar les proves i la comprensió de les mateixes, aturo el servei Shorewall (és el nostre Firewall) per tal d'evitar els problemes de tenir només una interfície virtualitzada en comptes de 2 físiques per a LAN y WAN al nostre Servidor. Aquesta limitació, juntament a la que es té en interfícies virtuals amb VirtualBox m'impedeix treballar amb Vlans. De fet el telèfon que he aconseguit per a fer les proves és molt vell i no suporta configuració Vlan a la interfície del terminal. El switch que utilitzo és en realitat un router amb 4 ports per a la LAN que no accepta etiquetatge Vlan.

El problema per a tenir una maqueta més semblant a una situació real és de pressupost. Treballo amb el que he pogut aconseguir i amb el préstec de dos portàtils sobre els que executaré un softphone anomenat Linphone que és gratuït.

Finalment el dibuix lògic de la maqueta és el següent:

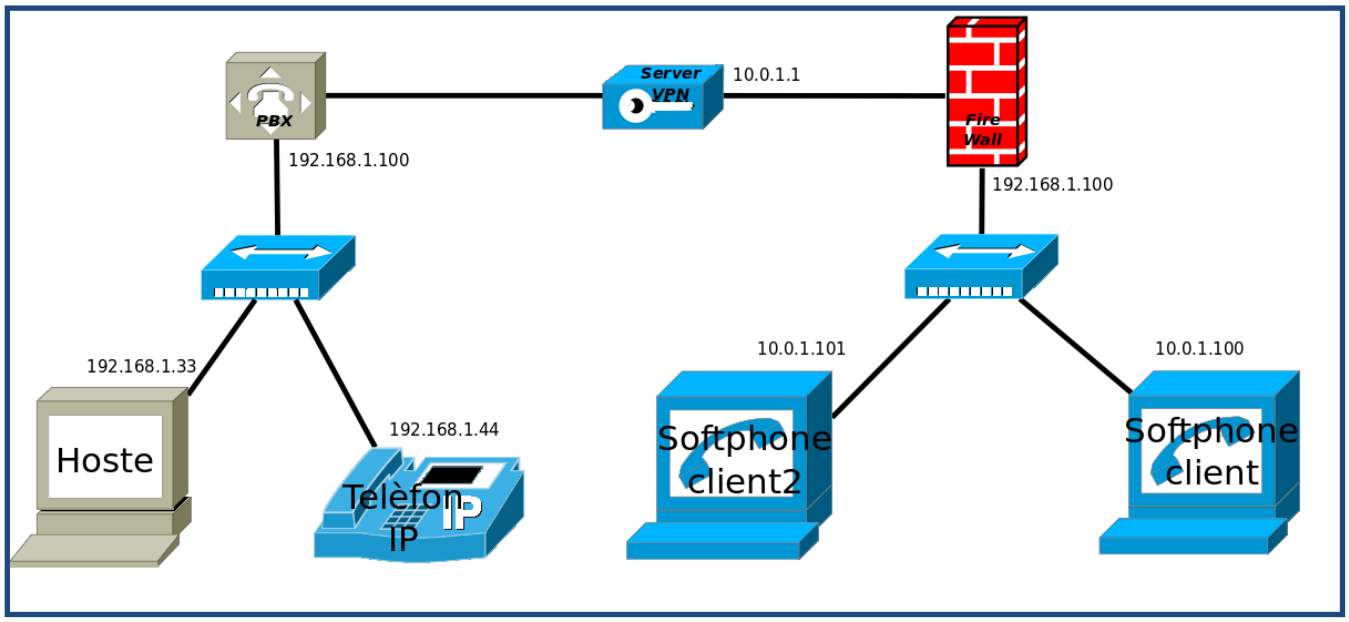

Figura 24: Representació de la maqueta de proves a nivell de funcionament lògic

5.2. METODOLOGIA I EINES PER LES PROVES

La metodologia utilitzada és bàsicament un "divideix i venceràs". Vaig servei per servei utilitzant els programes client adequats, a fi de garantir el correcte funcionament. Per tal de documentar-ho, faig captures de pantalla de les respostes del servidor, així com dels logs del serveis, que són els arxius en els que el sistema guarda l'activitat de cada servei per a auditar l'activitat si fos necessari.

Per a provar de forma eficient alguns dels serveis faig petites configuracions del mateixos que he documentat en el punt escaient. Algunes vegades es pot veure que les proves d'aquests serveis es fan utilitzant configuracions que no tenen a veure amb el projecte global, però com he dit es tracta d'una llicència per tal de simplificar la demostració del bon funcionament.

Malgrat que fem la comprovació a "trossos" de molts serveis, les proves de telefonia obligaran a fer-los treballar plegats.

Durant les comprovacions, les captures de tot el trànsit amb el programa Wireshark (software d'anàlisis de trànsit de xarxa) ha estat la meva guia per comprovar la correcció de la comunicació.

5.3. PROVES DE SERVEIS ASSOCIATS

Recordem que per a realitzar les proves parem el firewall fent un “shorewall stop” per tal de facilitar les proves pels motius que vam exposar en el punt 5.1.1.

5.3.1 NTP

La prova d'aquest serveis és senzilla. Utilitzem el client ntpdate per a fer una consulta directa al servidor. Podem veure a la dreta el servidor corrents amb la IP pública en el nostre cas 192.168.1.100. A l'esquerra podem veure la màquina hoste 192.168.1.33 fent la consulta.

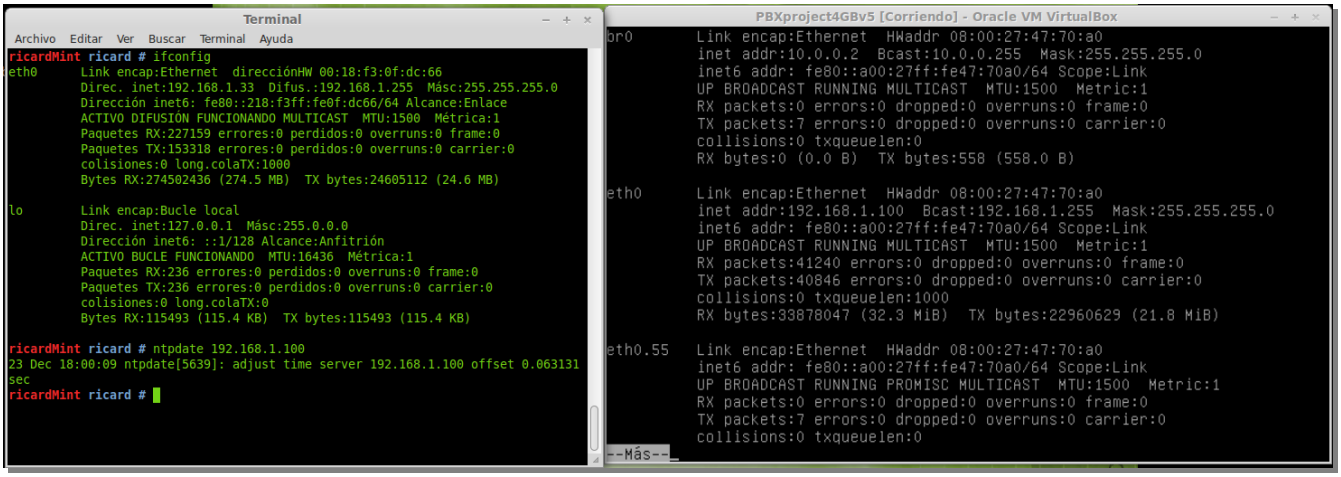

Figura 25: Prova NTP

5.3.2. DHCP

La comprovació del DHCP ha de tenir en compte que no puc utilitzar Vlan per la forma com tracta els broadcast el meu switch i la Màquina Virtual paquets etiquetats amb Vlan. Per aquest motiu faig una configuració expressa per a provar aquest servei que després no s'utilitzarà.

```
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.200 192.168.1.210;
  option domain-name-servers 192.168.1.100;
  option domain-name "pbxproject.com";
  option routers 192.168.1.100;
  option nntp-server 192.168.1.100;
  option tftp-server-name "192.168.1.100";
  option broadcast-address 192.168.1.255;
  default-lease-time 600;
  max-lease-time 7200;
}
```

A continuació connecto directament el telèfon IP que tinc per a fer les proves i com es pot veure rep IP per DHCP sense problemes. Es poden observar les fotografies durant el procés i el resultat després de la consulta DHCP feta pel terminal. Enganxo la seqüència complerta a continuació.

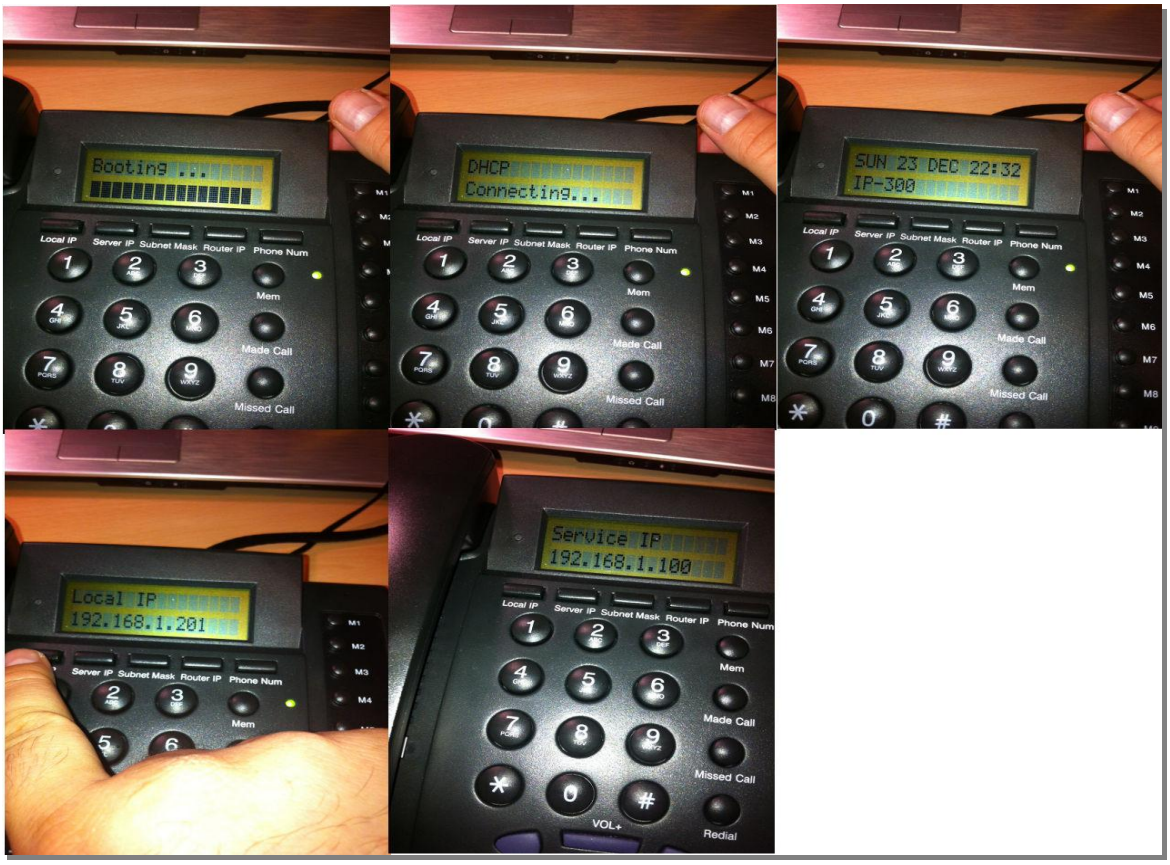

Figura 26: Prova DHCP

5.3.3. DNS

Per tal de poder confirmar el bon funcionament del DNS configurarem el domini pbxproject.com per tal que el resolgui com a 192.168.1.100. En primer lloc crearem un arxiu a "/etc/bind/" que es digui db.pbxproject amb el contingut següent:

```
$TTL    604800
@       IN      SOA     pbxproject.com. root.pbxproject.com (
                              2             ; Serial
                         604800                     ; Refresh
                          86400             ; Retry
                        2419200                     ; Expire
                         604800 )           ; Negative Cache TTL
@       IN      NS      localhost.
@       IN      A       192.168.1.100
@       IN      AAAA    ::1
```

Després afegim a /etc/bind/named.conf.local les següents línies:

```
zone "pbxproject.com" {
        type master;
        file "/etc/bind/db.pbxproject";
};
```

Ara només hem de reiniciar el servei amb un “/etc/init.d/bind9 restart” i com podem veure a la captura ja resol el domini tal com hem configurat.

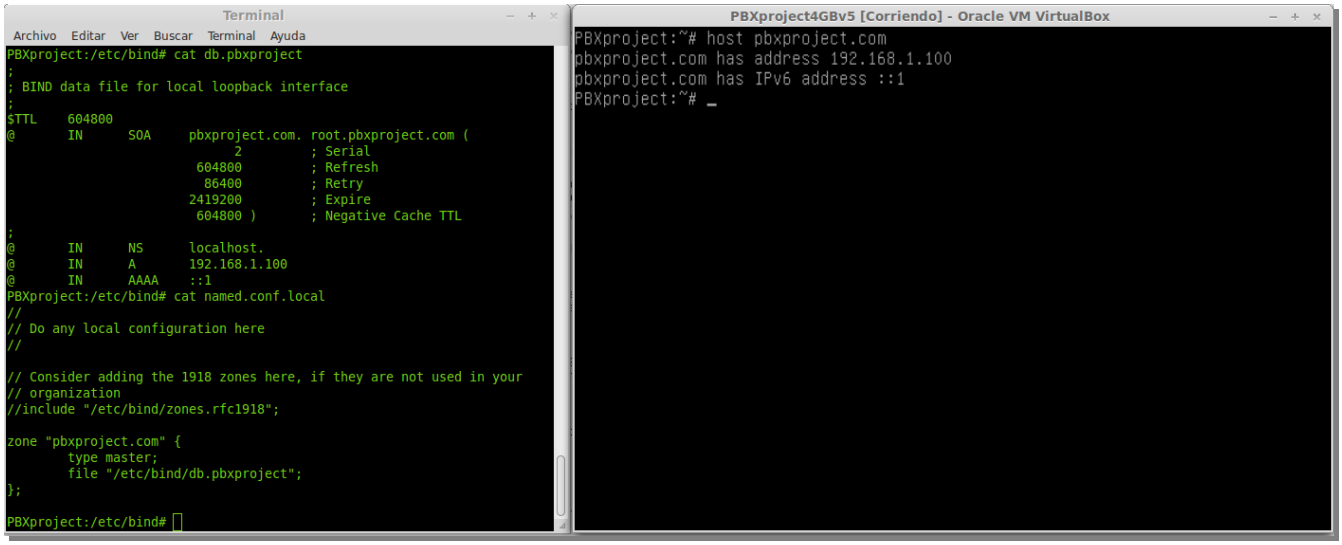

Figura 27: Prova DNS

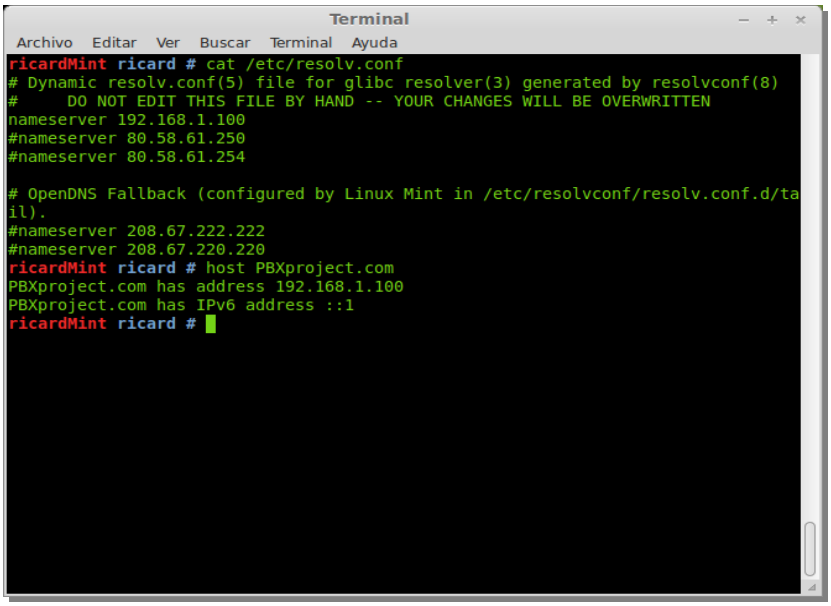

Figura 28: Prova des d’hoste DNS

5.3.4 TFTP

Per a la prova de tftp, seguim una dinàmica semblant a l'anterior prova. Creem un arxiu que és el que baixarem amb la següent comanda executada des de servidor.

```
echo "Això es una prova de tftp"> /srv/tftp/prova.txt
```

Utilitzarem per baixar un arxiu un client tftp executat a la màquina hoste com es veu a la captura.

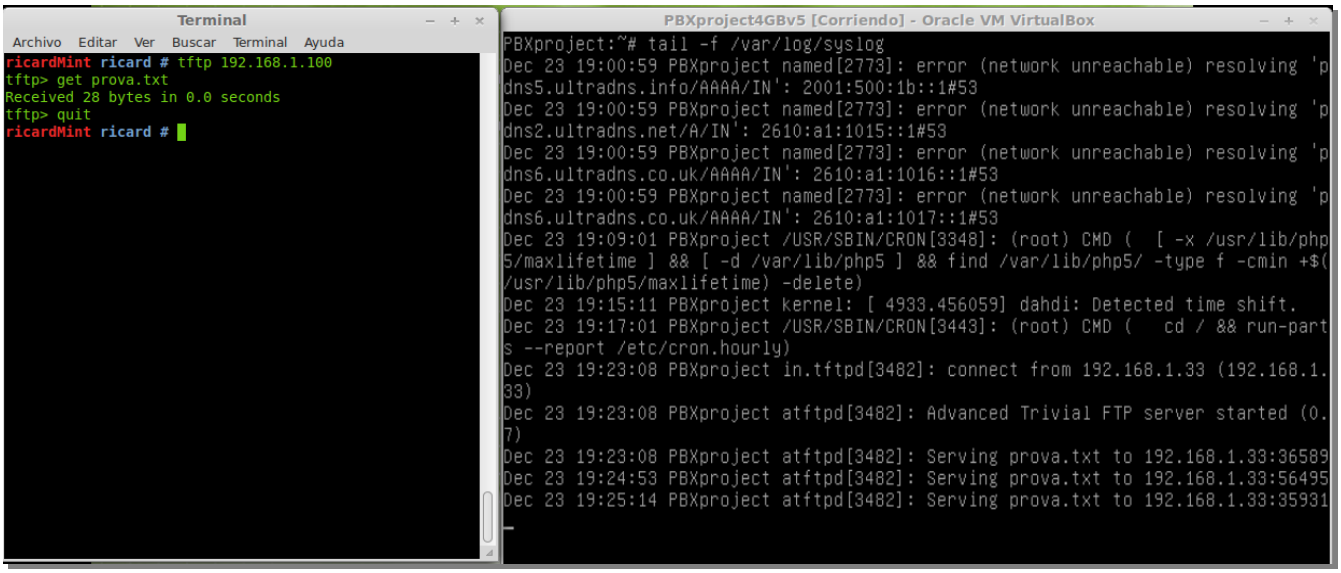

Figura 29: Prova TFTP

5.3.5. SHOREWALL FIREWALL

El funcionament del Shorewall és fàcilment comprovable. Partim de la configuració feta en la implementació que permet trànsit TCP als ports 22-SSH,80-FreePBX HTTP,10000-Webmin HTTPS i el UDP número 1194 per a la VPN. Fem un escaneig de ports amb l'eina nmap – aplicació d'exploració de ports – amb el Shorewall parat i després en fem un altre amb el servei engegat. La diferència es podrà observar automàticament en comparar els dos escanejos de ports TCP i UDP.

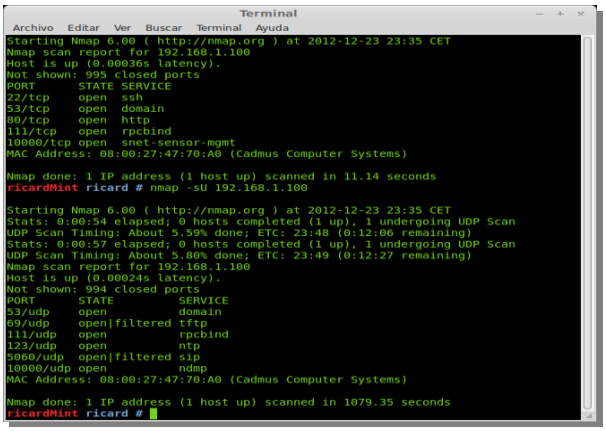

Figura 30: Escaneig sense Shorewall

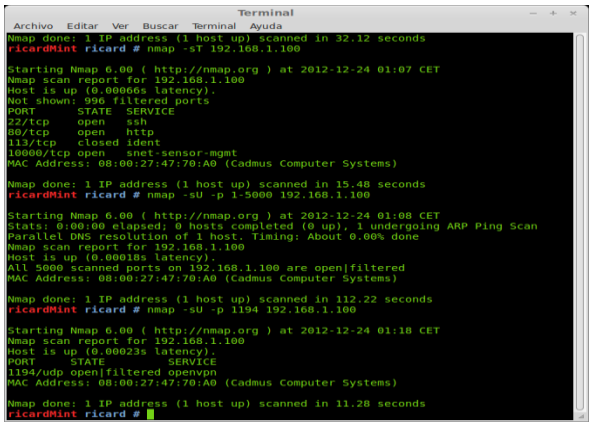

Figura 31: Escaneig amb Shorewall

5.3.6 OpenVPN

Aprofitant el client que vaig crear durant la implementació faig les comprovacions per a confirmar el funcionament. A la cantonada superior esquerra podem veure el missatge de confirmació que rep el client. A la cantonada superior dreta podem observar el log del Servidor que recull la connexió. A la part baixa podem veure a l'esquerra la interfície que ha creat la VPN, mentre que a la dreta es pot veure com arribo sense problemes a la interfície remota a l'altre costat del túnel VPN.

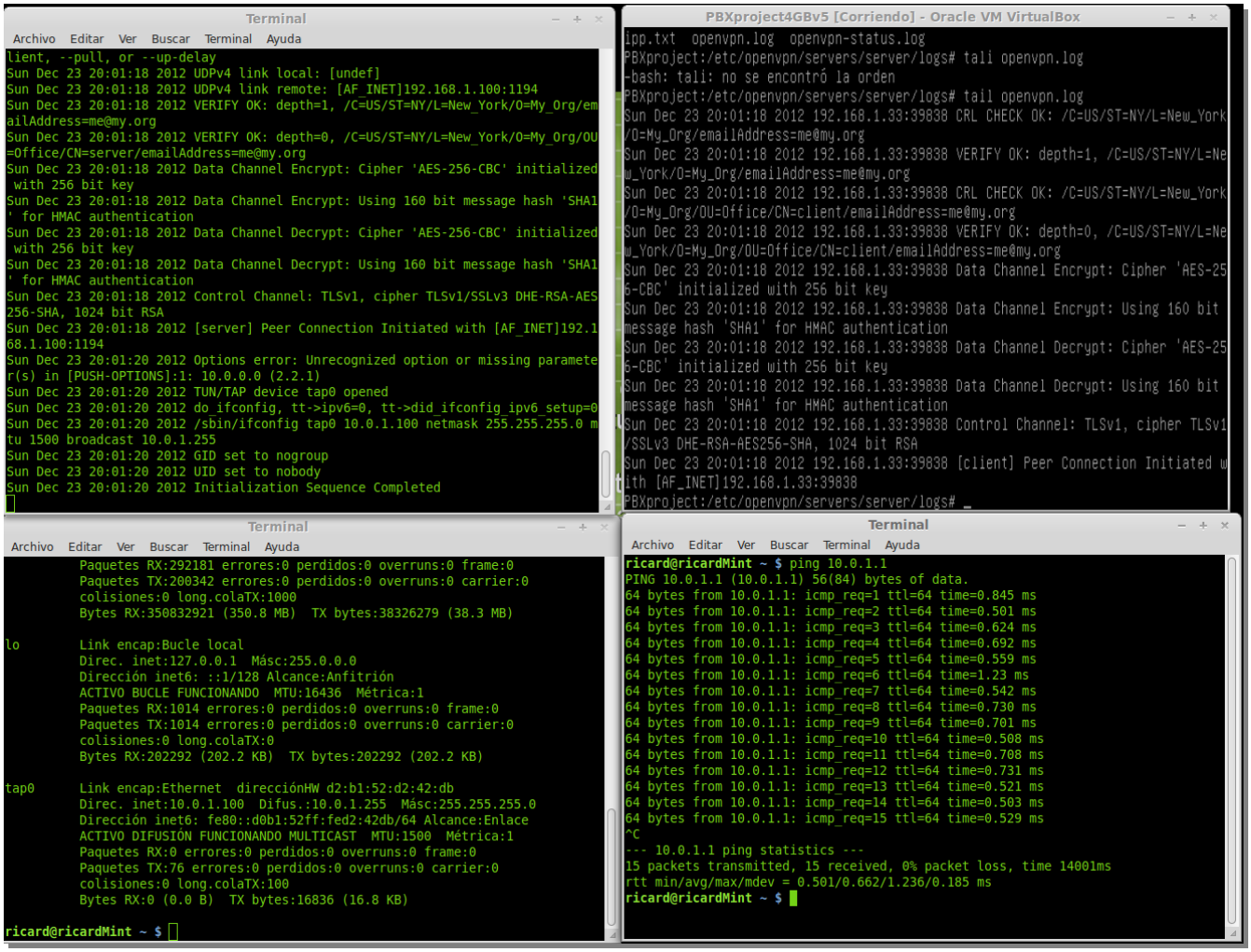

Figura 32: Prova OpenVPN

5.4. PROVES DE TELEFONIA

A l'iniciar les proves de telefonia hem de crear una configuració bàsica per a poder treballar. En primer lloc hem d'afegir uns "includes" – per a incloure el contingut d'altres arxius- dintre del context "[general]" a l'arxiu /etc/asterisk/sip.conf per a poder treballar els arxius generats per FreePBX.

```
[...]  [general]
#include sip_general_additional.conf
#include sip_registrations.conf
#include sip_additional.conf
```

També hem de modificar el /etc/asterisk/extensions.conf amb uns “includes” que poden anar al principi de la configuració i donant d'alta dos contextes nous importants per a poder generar trucades com a mínim entre extensions.

```
[...]
; include extension contexts generated from AMP
#include extensions_additional.conf
#include extensions_minivm.conf
; Customizations to this dialplan should be made in extensions_custom.conf
; See extensions_custom.conf.sample for an example
;#include extensions_custom.conf
[...]
[from-internal-xfer]
include => from-internal-additional ; auto-generated
exten => s,1,Macro(hangupcall)
exten => h,1,Macro(hangupcall)
[from-internal]
include => from-internal-xfer
include => bad-number
```

Amb això ja tenim la base per a les nostres proves. Realment FreePBX pot realitzar tot tipus de gestió de les trucades, que ja estan documentades a la pròpia pàgina de “FreePBX.org”. Les funcionalitats avançades d’aquesta solució són necessàries per a l'èxit del nostre projecte, però no serà objecte d'estudi. El que sí farem a continuació és donar d'alta 4 extensions, de la 100 a la 103. Facilito el mapa per a interpretar després què fem quan truquem a una extensió o una altra.

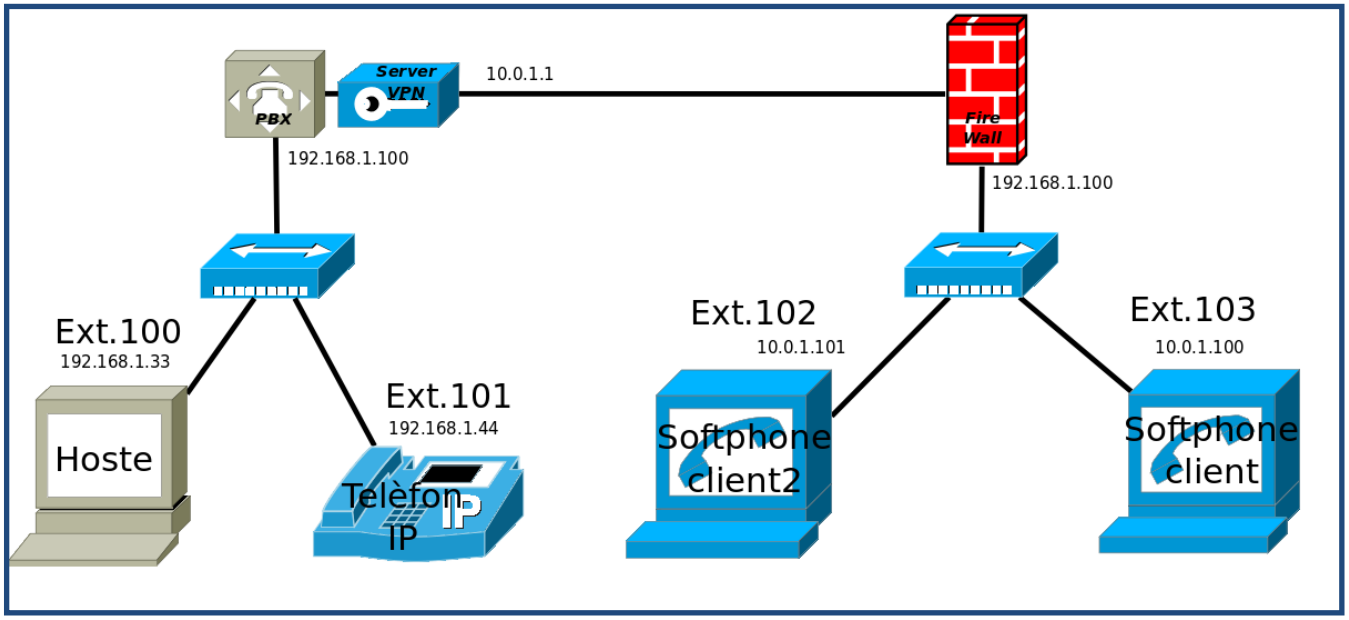

Figura 33: Assignació d'extensions en la maqueta

La captura mostra com s'han donat d'alta les 4 extensions a la FreePBX. Per a donar d'alta cal anar a Applications/Extensions. Després de fer Submit de totes les extensions fem un Apply per a que recarregui la configuració l'Asterisk.

Figura 34: Alta extensions FreePBX

Per a seguir les proves amb aquest esquema és imprescindible parar el Shorewall, cosa que podem fer a través del Webmin o fent "shorewall stop" des de la consola. A continuació aixeco els dos túnels VPN, utilitzant "openvpn client.conf" que rep la IP 10.0.1.100 i a l'altre equip "openvpn client2.conf" que rep la IP 10.0.1.101. Adjunto la captura del log openvpn:

Figura 35: Log del servei OpenVpn on es veuen aixecats els dos clients

Comprovo que tots dos equips arriben a la 10.0.1.1 que és la IP del costat del túnel que està dintre de la centraleta i registro dos softphones Linphone. Utilitzo aquest softphone ja que em permet enrutar el trànsit per dintre de la VPN (d'altres com l'Ekiga dóna problemes) contra aquesta IP.

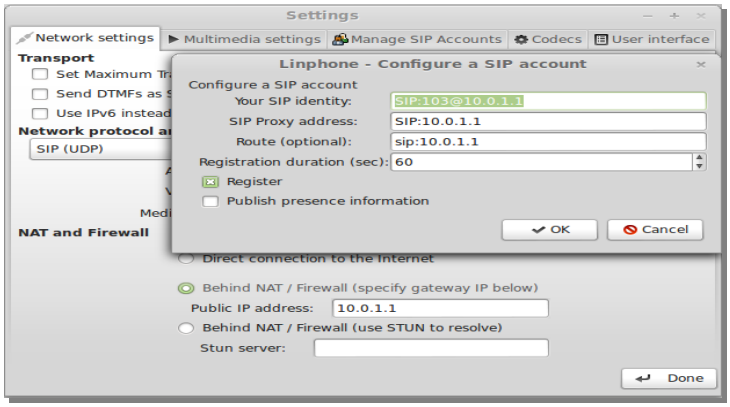

Figura 36: Configuració Linphone a través de VPN

Registro les 4 extensions. Si ens fixem en el registre d'extensions (ordre "sip show peers") es veu clar quines extensions es registren a través de la VPN. Aquestes utilitzen IPs del rang 10.0.1.x que pertanyen als túnels. Enganxo la captura amb el registre:

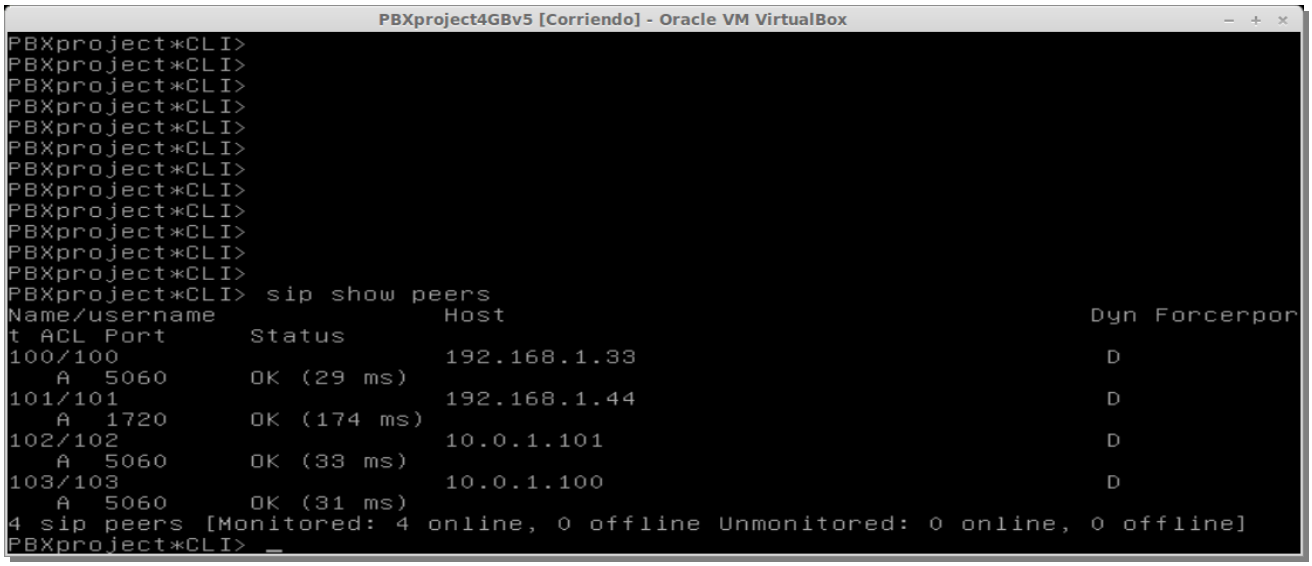

Figura 37: Registre de les 4 extensions en la consola de l'Asterisk

5.4.1. PROVES BÀSIQUES IMITANT FUNCIONAMENT LAN

Faig una trucada des de l'extensió 101 cap a la 100 i faig captura per a mostrar el funcionament. L'extensió 101 és un telèfon IP antic que no suporta vídeo. Enganxo l'entrada de la trucada juntament amb el codi de la consola Asterisk en la que es veu la trucada:

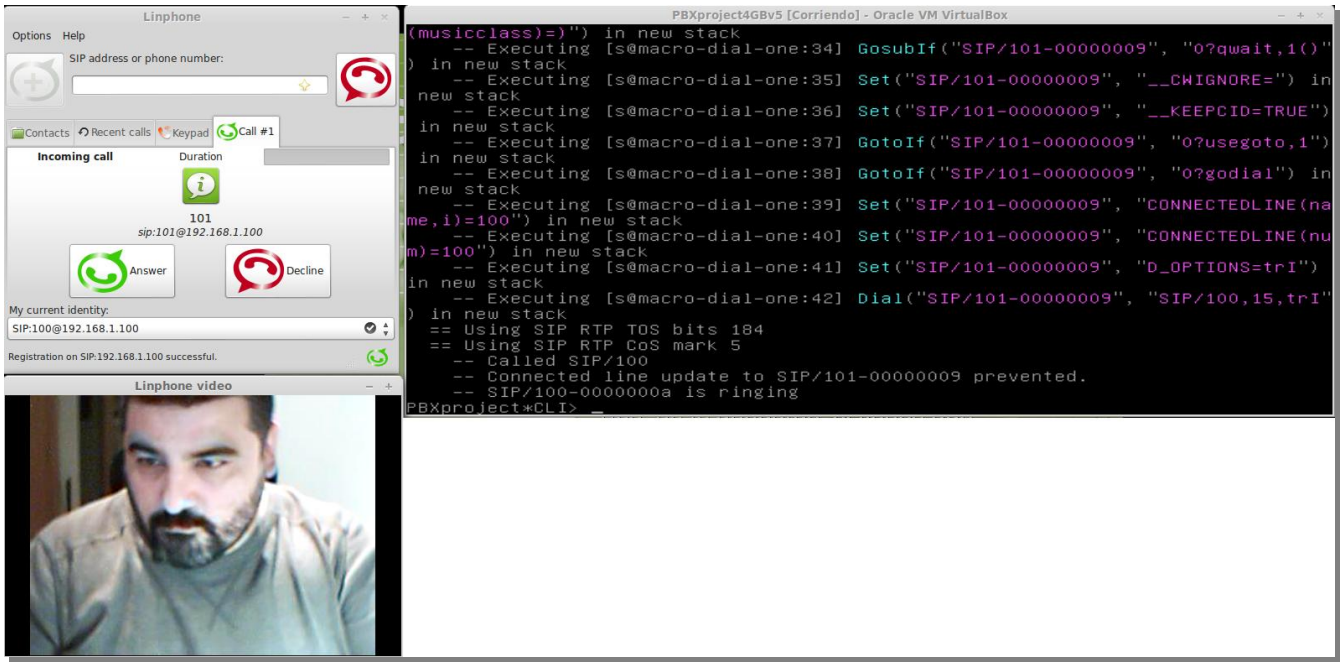

Figura 38: Trucada entre extensions que hi són a la LAN

5.4.2. PROVES D'USUARIS REMOTS A TRAVÉS DE VPN

Truquem entre la 102 i la 103 amb èxit. Fixem-nos que quan truquem, ho fem a 103@10.0.1.1 que és la IP del túnel. Si passéssim pel túnel al nostre dns i registréssim contra domini no tindríem el problema. La trucada la inicia la 102. Despenjo per a confirmar l'àudio que com es pot veure a l'indicador de barra passa sense problemes:

Figura 39: Trucada de la 102 a 103 del costat de la 102

Figura 40: Trucada de la 102 a 103 del costat de la 103

5.4.3. PROVES D'USUARIS VPN TRUCANT A EXTENSIÓ DE LA LAN

Provem a trucar des de l'extensió 102 a la 100 i obtenim bon resultat:

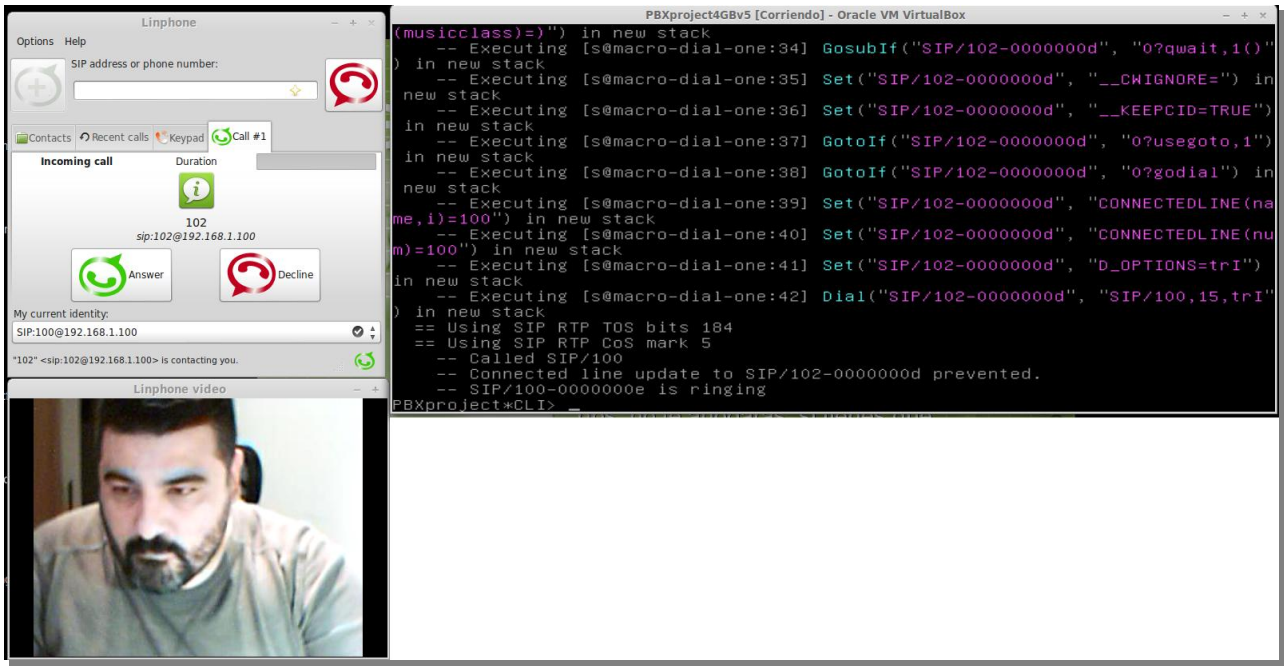

Figura 41: Trucada de la 102 a 100 del costat de la 100

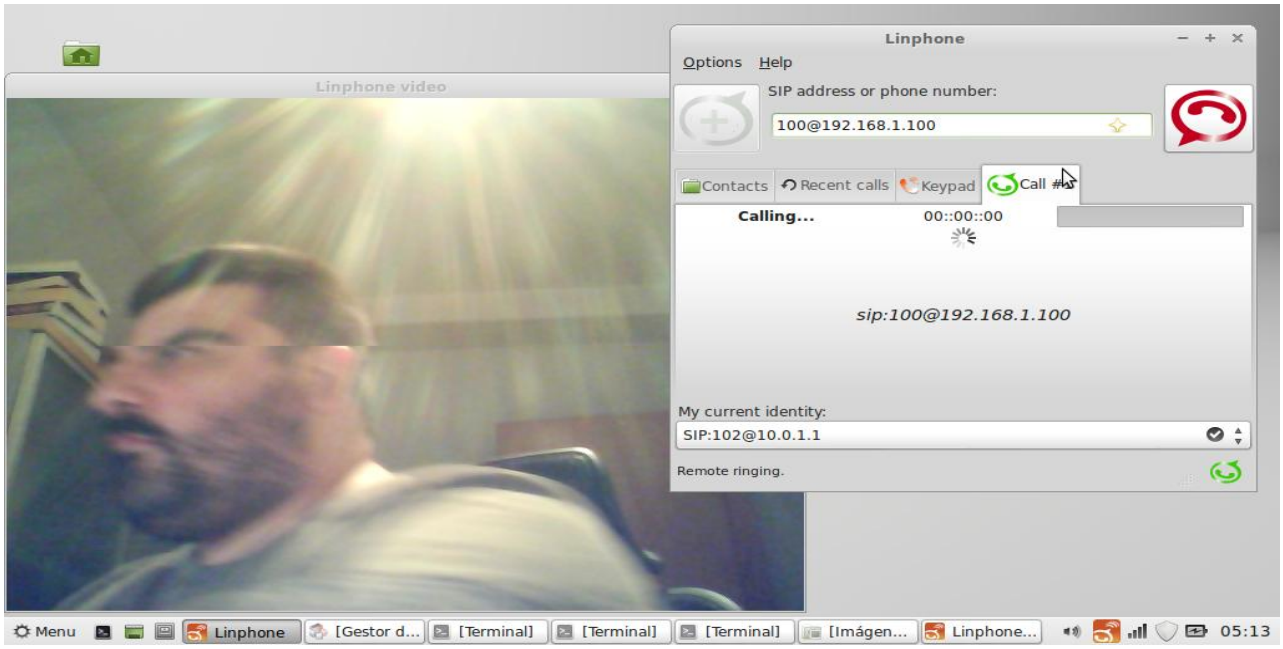

Figura 42: Trucada de la 102 a 100 del costat de la 102

6. ANÀLISIS DEL TREBALL

A l'hora d'analitzar el projecte, m'he estimat més fer petites subdivisions, ja que les diferents fases m'han suposat reptes diferents.

6.1. LA PRIMERA IDEA

A l'inici del projecte, la meva primera feina va ser somiar en un món ple de fantàstiques possibilitats. Poc a poc, de l'abstracció ben intencionada van anar perfilant-se quines necessitats pot tenir una empresa que volgués fer una migració cap a una nova centraleta IP.

En aquest punt és on comença a prendre forma realment el projecte. En un primer acostament vaig redactar una llista de requeriments irrenunciables així com l'estudi del mercat per a veure si es podia justificar econòmicament el projecte.

El següent pas lògic, després de fer l'acostament, era buscar les tecnologies que donaven millor resposta per a les necessitats d'implementació. Escollir quina era la millor manera de fer-ho va suposar una tasca enorme de documentació que ha resultat poc visible. Probablement les hores dedicades han estat exageradament grans, però malgrat tot, crec que la recerca ha estat prou exitosa.

6.2. LA PLANIFICACIÓ

La veritat sobre aquest apartat és que es basava en un munt d'elucubracions. És força complicat posar terminis a tasques que no s'han fet mai. Si la planificació fos un contracte amb un client podria dir que l'he respectat escrupolosament, però si es tracta de la comptabilització de les hores dedicades al projecte per a facturar-ho crec que he fracassat.

La forma de poder respectar la planificació ha estat dedicar-hi moltes més hores en els mateixos dies que tenia comptats. I per tant, ha estat en l'observança de la planificació que m'he adonat que el meu projecte era massa ambiciós per al temps que preveia dedicar-hi.

Malgrat que ha estat un repte organitzatiu important, estic content del que he après. He entès que en els petits detalls d'implementació i els processos de proves es perden una quantitat enorme d'hores.

Penso que l'experiència de l'enginyer en projectes semblants és clau per a poder fer aproximacions correctes de temps. Malgrat tot sempre s'ha de guardar un marge d'imponderables en les últimes fases del projecte.

6.3. ARRIBEN ELS PROBLEMES D'IMPLEMENTACIÓ

En principi, he partit d'un axioma que al final ha resultat no ser tant cert com m'esperava. Vaig fer una feina de documentació important i vaig intentar fer valdre la meva experiència per a fer la implementació. Realment, algunes de les coses del projecte les he fet abans, encara que fa temps.

Així que després de trobar la millor combinació de serveis, ve el moment en el que començo a compilar i instal·lar serveis que està documentat que funcionen bé... però la realitat no és tant amable. Noves versions amb diferències i problemes nous han deixat tremolant les hores que tenia previstes. Petits problemes a vegades han suposat dies de proves i navegació desesperada per Internet. Finalment, sempre era una línia de codi, el valor d'una variable, una petita modificació en un arxiu...però les hores han estat incomptables. Penso que finalment he corregit totes aquestes petites coses i he facilitat la manera de fer-ho per esquivar-les.

6.4. LES PROVES DE FUNCIONAMENT

En aquest punt he tingut més problemes al no tenir suficients recursos, que per qualsevol problema d'implementació anterior. Per tal de poder treballar amb el que tinc he hagut de deixar a banda la part de Vlan. Crear una maqueta que pogués representar el funcionament teòric ha estat un repte d'imaginació i feina. És per aquest motiu que no em quedo satisfet del tot amb les proves que he realitzat.

Malgrat la impossibilitat de provar-ho amb Vlans, crec que les proves han estat suficients per a demostrar el bon funcionament global del projecte. Si hagués tingut recursos per a comprar un switch amb 802.1q amb un ordinador dedicat per això, en comptes de virtualitzat, el resultat hauria estat l'òptim. Les proves fetes em fan estar segur d'això.

6.5. CONCLUSIONS

Crec finalment que he aconseguit assolir els objectius que em vaig proposar. En la introducció i objectius del projecte volia presentar un producte capaç de donar resposta a les empreses. Tal i com em vaig proposar, he obtingut un prototip que justifica el llançament d'un nou producte a partir de solucions de Programari Lliure implementades en aquest projecte.

Totes les solucions implementades són àmpliament esteses, però podem dir que la implementació creuada de totes aquestes solucions ens donaria una quota de mercat suficient per a fer negoci.

Finalment crec que el projecte m'ha ajudat a entendre millor la dinàmica d'un projecte complet. La part inicial de generació d'objectius, justificació i planificació ha estat totalment nova. En aquest punt he pogut utilitzar moltes eines apreses d'assignatures com Multimèdia i Comunicació o Administració de Xarxes i Sistemes Operatius.

A nivell tècnic, he aprés molt sobre els serveis que he implementat i encara més sobre els errors que poden donar i perquè. He pogut d'utilitzar molts coneixements adquirits durant la carrera i m'he sentit pagat per les moltes hores d'estudi sobre molts temes (especialment xarxes i sistemes operatius).

Penso que tot el que he après és totalment exportable al món laboral i acosta la formació acadèmica rebuda a la UOC cap a l'empresa. Ara entenc millor la diferència entre ser un enginyer i ser un bon tècnic de sistemes, entre el que s'espera del primer i del segon.

7. BIBLIOGRAFIA I RECURSOS UTILITZATS

FONTS D'INFORMACIÓ

- Materials Web UOC: Treball Final de Carrera -Aula de TFC
- Pla docent de l'assignatura TFC GNU/Linux

PÀGINES WEB

3CX *Caracteristiques generals 3CX* [en línia] http://www.3cx.com/ip-pbx/index.html (01/11/2012)

AdamVozIP *Caracteristiques generals vPBX* [en línia] http://www.adamvozip.es/index.php?option=com_content&view=article&id=132&Itemid=221&lang=es (01/11/2012)

ASTERISK *Caracteristiques generals ASTERISK PBX* [en línia] http://www.asterisk.org/ (01/11/2012)

AVAYA *Caracteristiques generals OXO* [en línia] http://www.avaya.com/es/resource/assets/brochures/SB4821SE.pdf (01/11/2012)

CISCO *Caracteristiques generals UC500 Series* [en línia] http://www.ciscocignal.com/products/category/6-telefonia-ip/17-pbx-cisco-unified-communications-serie-500-.html (01/11/2012)

ELASTIX *Caracteristiques generals ELASTIX PBX* [en línia] http://www.elastix.org/index.php/es/inicio.html (01/11/2012)

FreePBX *Caracteristiques generals FreePBX* [en línia] ww.freepbx.org (01/11/2012)

VozTelecom *Caracteristiques generals Oigaa Office* [en línia] http://www.voztele.com/centralita-virtual-oigaa/modalidades-centralita-virtual-oigaa/oigaa-office/oigaa-office.htm (01/11/2012)

VOIP PARA NOTAVOS Pagina principal [en línia] www.voipnovatos.es (01/12/2012)

VOIP INFO Pagina principal [en línia] www.voip-info.org (01/12/2012)

WEBMIN Pagina principal [en línia] www.webmin.com (01/12/2012)

PROGRAMARI UTILITZAT

- Dia: Eina per a diagramar xarxes i UML.
- LibreOffice: ofimàtica.
- Linphone: Softphone amb llicencia GNU
- Microsoft Office: ofimàtica.
- Microsoft Project: ofimàtica, gestió de projectes.
- VirtualBox: Software de virtualització de maquines.
- Wireshark: Anàlisi de transit de xarxa.

PEUS DE PÀGINA:

1 Segons tarifes de VozTelecom: http://www.voztele.com/tarifas-llamadas-voip/docs/Tarifas_web.pdf
2 Webminar Seguretat VoIP:http://www.slideshare.net/Quobis/webinar-seguridad-voip
3 Informació Protocols VoIP: http://es.wikipedia.org/wiki/SIP _http://es.wikipedia.org/wiki/Real-time_Transport_Protocol
4 Informació Asterisk: http://www.asterisk.org/
5 Informació Centile: http://www.centile.com/
6 http://blogcmt.com/2012/06/11/asi-tratan-el-p2p-y-la-voip-las-operadoras-europeas/
7 http://www.aciem.org/home/index.php/component/k2/item/104-holanda-aprueba-una-ley-que-proh%C3%ADbe-que-los-isp-bloqueen-voip
8. http://www.infoworld.com/d/networking/windows-versus-linux-server-face-262?page=0,0
9. http://www.infoworld.com/d/developer-world/linux-beats-windows-2008-power-saving-measures-595
10.http://es.wikipedia.org/wiki/Protocolos_de_VoIP
11 http://es.wikipedia.org/wiki/Subsistema_Multimedia_IP
12 .Anàlisi del trànsit de xarxa, que permet estudiar el contingut de tots els nivells de xarxa.
13 http://www.voipforo.com/SIP/SIPmensajes.php
14 http://www.voipforo.com/codec/codecs.php
15 http://pingtest.net/
16 http://es.wikipedia.org/wiki/Jitter

LINK DESCÀRREGA PROTOTIP:

https://www.dropbox.com/s/t2y6ggz8lhlkypr/PBXproject.ova

El arxiu que es descarrega PBXproject.ova es directament importable desde el VirtualBox, programari gratuït que podeu trobar a:

- https://www.virtualbox.org/wiki/Downloads

ANNEX I: SIP

I-A. ARQUITECTURA D'UN PROVEÏDOR SIP

L'arquitectura que pressento és molt general, sense entrar en detall del funcionament intern. La descripció general de Servidors ens ajudarà a adonar-nos fins a quin punt es pot complicar l'estructura des del punt de partida de dos punts units per SIP directament. Tot seguit faig una relació descriptiva dels elements de l'arquitectura i mes tard explico la interrelació.

- **User-Agent**: en l'estructura Client-Servidor, poden ser el client o servidor (UAC o UAS) i són els extrems de la trucada. Són els que poden fer o rebre una trucada.

- **SIP Servers**: són les màquines que reben peticions, i donen valor afegit a la trucada o col·laboren en balancejos de càrrega.

  D'aquesta última classe m'agradaria fer una separació en tres sub-tipus, que ens acosten a la realitat una mica més. Encara que totes les funcions es poden implementar en una única màquina és necessari repartir funcions entre diferents servidors per a no col·lapsar un Servidor en concret. Partim de la idea que estem centrats en l'Arquitectura d'un proveïdor VoIP i que tenim requeriments alts.

- **SIP Proxy**: són les màquines que autentifiquen a l'UA i encaminen el SIP de la trucada supervisant tota la senyalització.

- **RTP Proxy**: com que d'una trucada el volum d'informació SIP és petit respecte el flux d'àudio RTP, el SIP Proxy durant la negociació va oferint als extrems diferents IP dels RTP Proxy disponibles. Segons el tipus de trucada es pot pactar en la trucada que l'àudio no passi per RTP Proxy i viatgi d'extrem a extrem. Això últim, donat l'heterogeni del nostre món SIP no és l'opció més segura, encara que és la més eficient. Un problema habitual que es troba és que la senyalització SIP directament entre extrems sigui deficient i algun dels extrems perdi àudio per un marcatge incorrecte del RTP i sobretot UDP amb NATs de per mig.

- **App-Server**: els serveis afegits que es poden donar tenen com a límit la imaginació. La forma més fàcil d'entendre del que estem parlant és posar exemples concrets. Un App-Server pot canviar el número que es mostra en fer una trucada, pot cancel·lar una trucada a un 806 de pagament, perquè el client ho ha restringit, pot enviar la trucada a una bústia de veu o desviar-la a un altre número. A nivell del proveïdor pot controlar el consum de la trucada per a facturar-la o per tallar-la si s'ha excedit el risc de facturació assignat al client. Una tasca important consisteix en aplicar unes regles d'enrutament de les trucades a GW per a un client concret i un tipus de destins. Això últim és cabdal per a optimitzar al màxim els costos de les trucades i poder oferir millors preus als clients.

- **GateWay IP/PSTN**: son les màquines que interconnexionen la xarxa SIP amb la telefonia tradicional. Normalment estan connectats a canals primaris d'operadores tradicionals, per una banda, i Internet per l'altra. Fa la conversió d'RTB a IP i d'IP a RTB en temps real. Els GW fan totes les conversions majoritàriament per hardware, encara que són configurables per software algunes funcions.

Són solucions predominantment hardware per a aconseguir un rendiment molt alt en temps real i no tenir retards majors, com en el cas de les solucions software.

Aquesta afirmació té a veure amb el temps de computació de la lògica de qualsevol programari. Amb circuits sempre és més ràpid d'implementar tasques repetitives i simples.

- **Registre de UA**: normalment les estructures dels proveïdors VoIP tenen adreça IP fixes en tots els servidors i Gateways. En el seu defecte tenen dominis que van actualitzant la seva resolució, encara que aquesta opció no resulta gaire bona per a resoldre IP dinàmiques, per qüestions de propagació DNS i memòries cau.

  Els que sí són habituals són les IP dinàmiques o més d'un registre d'un compte SIP per part d'un UA (client). En les trucades sortints no hi ha problema en localitzar i validar la identitat d'un client, ja que tota aquesta informació s'inclou en la negociació de la trucada. Quan hi ha una trucada entrant cap a un client és on necessito conèixer per endavant en quina IP o IPs s'han d'entregar la trucada i en quin port espera la trucada entrant. No només s'ha de conèixer a on enviar la trucada sinó validar que qui la rep estigui degudament autoritzat. Tot aquest procés s'ha d'anar repetint cada cert temps per tal que la informació estigui actualitzada.

I-B. REGISTRE D'UN CLIENT SIP

El procés de negociació per a registrar un client és :

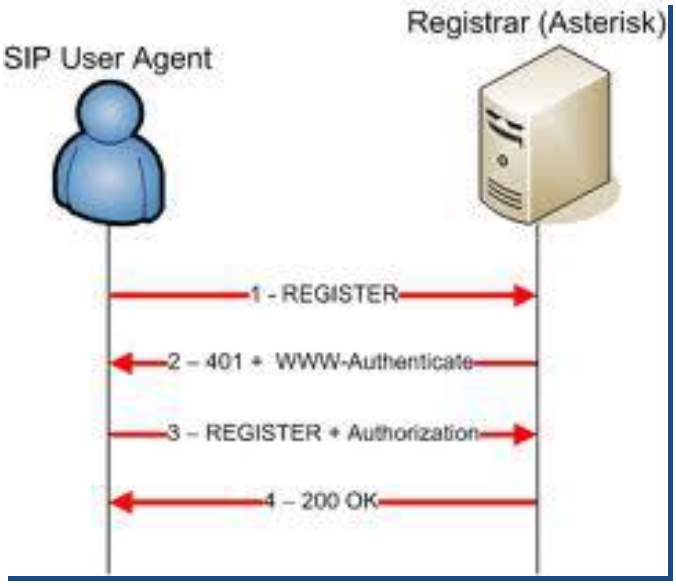

Figura 43: Procés de registre d'un client

El client fa el pas de demanar el Registre. En el nostre cas el client vol registrar-se a 217.18.xxx.xxx en el port 61000. A mes envia la IP local 192.168.1.101 en el Contact per a facilitar ser trobat a darrere de NAT. Per una altra banda “Expires: 3600” fa referència a que el registre caducarà en una hora = 3600 segons. Dintre d'un hora, ambdós equips negociaran de nou el registre.

```
U 2011/06/06 19:54:52.766515 217.18.xxx.xxx:61000 -> 193.22.119.20:5060
REGISTER sip:voztele.com SIP/2.0..
From: <sip:34000xxxxxx@voztele.com>;tag=df 9088-c0a80165-13c4-50029-16-10314d1d-16..
To: <sip:34000xxxxxx@voztele.com>..
Call-ID: e16268-c0a80165-13c4-50029-16-16b35e87-16..CSeq: 219 REGISTER..
Via : SIP/2.0/UDP 192.168.1.101:5060;branch=z9hG4bK-5fb9e-175ee162-3a25e103..Ma x-Forwards: 70..
Expires: 3600..Authorization: Digest username="34000xxxxxx", realm="voztele.com",uri="s
ip:voztele.com",algorithm=MD5..
Contact: <sip:34000xxxxxx@192.168.1.101>..Content-Length: 0....
```

La resposta natural del Servidor es Unauthorized i envia una clau al camp “nonce”.

```
U 2011/06/06 19:54:52.768155 193.22.119.20:5060 -> 217.18.xxx.xxx:61000
SIP/2.0 401 Unauthorized..
From: <sip:34000xxxxxx@voztele.com>;tag=df9088-c0a 80165-13c4-50029-16-10314d1d-16..
To: <sip:34000xxxxxx@voztele.com>;tag=ba3d5 1acad53eeb51d56ab2459dbff7b.430c..
Call-ID: e16268-c0a80165-13c4-50029-16-16 b35e87-16..CSeq: 219 REGISTER..
Via: SIP/2.0/UDP 192.168.1.101:5060;branch=z 9hG4bK-5fb9e-175ee162-3a25e103;
rport=61000;received=217.18.xxx.xxx..WWW-Aut henticate: Digest realm="voztele.com",
nonce="4ded31b8c04bc3bf6e56431996528 e0c38bf596e", stale=true..
Server: OpenSER (1.2.1-notls (i386/linux))..Conte nt-Length: 0....
```

El client torna a enviar el REGISTER però aquesta vegada inclou el “nonce” del servidor i una resposta “response” calculada amb el “nonce” i una contrasenya que valida el compte.

```
U 2011/06/06 19:54:54.762387 217.18.xxx.xxx:61000 -> 193.22.119.20:5060
REGISTER sip:voztele.com SIP/2.0..
From: <sip:34000xxxxxx@voztele.com>;tag=df 9088-c0a80165-13c4-50029-16-10314d1d-16..
To: <sip:34000xxxxxx@voztele.com>..
Call-ID: e16268-c0a80165-13c4-50029-16-16b35e87-16..CSeq: 220 REGISTER..
Via : SIP/2.0/UDP 192.168.1.101:5060;branch=z9hG4bK-5fba0-175ee932-196c3bd1..
Ma x-Forwards: 70..Expires: 3600..Authorization: Digest username="34000xxxxxx", realm="voztele.com",
nonce="4ded31b8c04bc3bf6e56431996528e0c38bf596e",uri="s ip:voztele.com",
response="6bef91b0xxxxxxxxxxxxxxxxxxxxxxxxxxxx",algorithm=MD5..
Contact: <sip:34000xxxxxx@192.168.1.101>..Content-Length: 0....
```

El Servidor valida la contrasenya, el user 34000xxxxxx i retorna OK. Ja està registrat.

```
U 2011/06/06 19:54:54.764354 193.22.119.20:5060 -> 217.18.xxx.xxx:61000
SIP/2.0 200 OK..
From: <sip:34000xxxxxx@voztele.com>;tag=df9088-c0a80165-13c4 -50029-16-10314d1d-16..
To: <sip:34000xxxxxx@voztele.com>;tag=ba3d51acad53eeb 51d56ab2459dbff7b.6462..
Call-ID: e16268-c0a80165-13c4-50029-16-16b35e87-16. .CSeq: 220 REGISTER..
Via: SIP/2.0/UDP 192.168.1.101:5060;branch=z9hG4bK-5fba0-175ee932-196c3bd1;
rport=61000;received=217.18.xxx.xxx..
Contact: <sip:340 0024321@192.168.1.101>;expires=3600;received="sip:217.18.xxx.xxx:61000"..
Se rver: OpenSER (1.2.1-notls (i386/linux))..Content-Length: 0....
```

Durant el temps que està registrat (el valor Expires) és possible que les sessions NAT obertes als routers es tanquin. Per aquest motiu hi ha un sistema de “keep-alive” per a mantenir la connexió. Realment és més important mantenir el tràfic que realment la informació que s'intercanvia. Aquí podem observar aquest procés que es fa mitjançant els OPTIONS d'un adaptador VoIP/analògic SPA 2102 des de la IP 84.124.xxx.xxx en el port 5070 i amb IP local 192.168.1.186.

```
U 2011/06/06 19:54:11.676232 193.22.119.20:5062 -> 84.124.xxx.xxx:5070
OPTIONS sip:34000xxxxxx@192.168.1.186:5070 SIP/2.0..
Via: SIP/2.0/UDP 193.22. 119.20:5062;branch=0..
From: sip:xxxxx@voztele.com;tag=e56131c4..
To: 34000xxxxxx@voztele.com..
Call-ID: 338f5133-b9b5acb5-0a30bd@193.22.119.20..CSeq : 1 OPTIONS..Content-Length: 0....
U 2011/06/06 19:54:11.710687 84.124.xxx.xxx:5070 -> 193.22.119.20:5062
SIP/2.0 200 OK..To: sip:34000xxxxxx@voztele.com;tag=23380d7a2af427i0..
From: sip:xxxxx@voztele.com;tag=e56131c4..
Call-ID: 338f5133-b9b5acb5-0a30bd@193. 22.119.20..CSeq: 1 OPTIONS..
Via: SIP/2.0/UDP 193.22.119.20:5062;branch=0..
S erver: Linksys/SPA2102-5.1.9..Content-Length: 0..Allow: ACK, BYE, CANCEL, I NFO, INVITE,
NOTIFY, OPTIONS, REFER..Supported: x-sipura, replaces....
```

Gràcies al registre i al manteniment dels OPTIONS de la connexió estem preparats per a rebre trucades en el nostre client o centraleta. S'ha d'entendre que és un procés fractal, és a dir, que si tenim una centraleta registrada contra un SIP Server, els terminals telefònics han d'estar a la seva vegada registrats contra la seva centraleta per tal de poder ringar quan hi arribi una trucada que s'ha d'entregar a una extensió determinada. Dintre de la LAN (o des de fora si es tenen extensions remotes) s'ha de portar a terme la mateixa política amb totes les extensions que volen rebre trucades.

I-C. ESTUDI D'UNA TRUCADA (EXEMPLE COMENTAT)

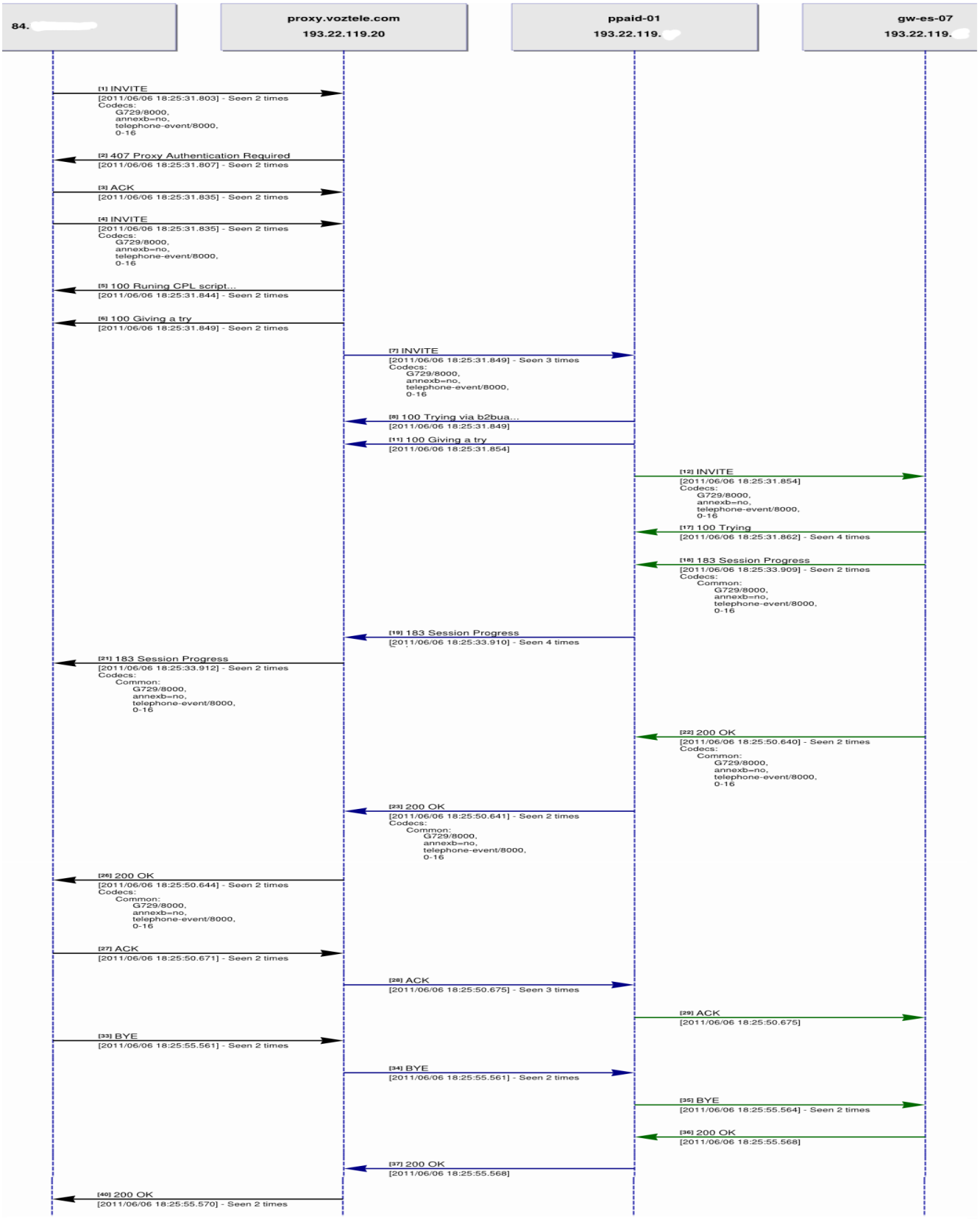

Figura 44: Captura de trucada

Desenvolupo a partir del número de paquet que es pot veure entre claudàtors de la captura.

```
[1] INVITE sip:9.....@voztele.com SIP/2.0 Volem marcar el 9.....
Via: SIP/2.0/UDP 84.124.xxx.xxx:7123;branch=z9hG4bK1558e699;rport
Max-Forwards: 70
From: "asterisk" <sip:34000xxxxxx@voztele.com>;tag=as5805d795
```

Estic marcant des de l'usuari 34000xxxxxx de vozTelecom.

```
To: <sip:9.....@voztele.com> Volem marcar al 9....., To: es cap a on volem anar
Contact: <sip:34000xxxxxx@84.124.xxx.xxx:7123>
```

En el Contact tenim la nostra IP 84.124.xxx.xxx i el nostre port 7123. En aquest cas haurem de tenir una bona política NAT per a arribar a l'Asterisk que tenim al darrere.

```
Call-ID: 69cd18e66b996cf8215b62577977bfb6@voztele.com
```

Call-ID és un identificació única d'aquesta trucada per a l'Asterisk.

```
CSeq: 102 INVITE Portem un numero de seqüència per a reconèixer les respostes
User-Agent: Asterisk PBX 1.6.2.18 Veiem que es tracta d'un Asterisk 1.6
Date: Mon, 06 Jun 2011 16:24:58 GMT Data
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO Els mètodes acceptats
Supported: replaces, timer
Content-Type: application/sdp Fixar-se que anunciem que tenim sdp dintre d'aquest paquet SIP
Content-Length: 260 Longitud del paquet SIP
```

La majoria de valors següents ja són SDP i es poden consultar en el link que vaig facilitar abans d'aquest protocol. Només comento el principal.

```
v=0
o=root 169170159 169170159 IN IP4 84.124.xxx.xxx
s=Asterisk PBX 1.6.2.18
c=IN IP4 84.124.xxx.xxx Aquesta es la IP on s'espera que arribi el RTP
t=0 0
m=audio 13684 RTP/AVP 18 101
```

L'àudio s'espera per a aquesta trucada al port 13684. Asterisk utilitza habitualment del 10000 al 20000. L'ordre de codecs utilitzat és el 18 i el 101. La llista és ben curta però pot ser molt llarga. En realitat només el 18 es destina a l'àudio com es pot veure i es tracta del codec G729. El 101 és el telephone-event, és a dir, els DTMF per poder passar opcions a les ordres per marcació com els IVR (les operadores de resposta automàtica) que per exemple et diu que marquis 1 per l'opció x, 2 per l'opció i etc.

```
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendrecv
```

Aquest valor diu que els paquets d'àudio que passarem com hem vist abans en G729 seran de 20 ms. de durada.

Com en el cas del Register, enviem un error, en aquest cas el 407, amb el nonce per a sol·licitar la autenticació.

```
[2]SIP/2.0 407 Proxy Authentication Required
Via: SIP/2.0/UDP 84.124.xxx.xxx:7123;branch=z9hG4bK1558e699;rport=7123;received=84.127.xxx.xxx
From: "asterisk" <sip:34000xxxxxx@voztele.com>;tag=as5805d795
To: <sip:9.....@voztele.com>;tag=ba3d51acad53eeb51d56ab2459dbff7b.038c
Call-ID: 69cd18e66b996cf8215b62577977bfb6@voztele.com Utilitzem el mateix Call-ID
CSeq: 102 INVITE Utilitzem el mateix número de seqüencia per que l'Asterisk relacioni la resposta.
Proxy-Authenticate: Digest realm="voztele.com", nonce="4ded00a73a639409eb63366edf7654bdb09c1306"
Server: OpenSER (1.2.1-notls (i386/linux))
Content-Length: 0
[3]ACK sip:9.....@voztele.com SIP/2.0 No te gaire importància, només confirma l'arribada del paquet [2]
[4]INVITE sip:9.....@voztele.com SIP/2.0 Em menjo texte perquè excepte el que poso es igual al anterior INVITE [...] CSeq: 103 INVITE [...]
Proxy-Authorization: Digest username="34000xxxxxx", realm="voztele.com", algorithm=MD5, uri="sip:9.....@voztele.com", nonce="4ded00a73a639409eb63366edf7654bdb09c1306", response="00c171a7e9f9582a6c5exxxxxxxxxxxxxxxxx" Aquí m'autentifico [...]
[5]SIP/2.0 100 Runing CPL script...
CSeq: 103 INVITE
[6]SIP/2.0 100 Giving a try
CSeq: 103 INVITE
```

Aquests dos paquets confirmen a l'Asterisk que s'està realitzant la trucada, missatge 100[16]

```
[7]i[9],INVITE sip:99xxxxx#00349.....@193.22.119.xx:5060 SIP/2.0
```

Formatejarem l'invite per a que el GW accepti la trucada identificant al Proveïdor SIP. Els Record i Via serveixen per a poder anar informant dels salts que hem anat fent a qui rep la petició. Això està heretat de l'HTTP

```
Record-Route: <sip:193.22.119.20;lr=on;ftag=as5805d795>
Via: SIP/2.0/UDP 193.22.119.20;branch=z9hG4bK4c0c.65ee542.0
Via: SIP/2.0/UDP 84.124.xxx.xxx:7123;received=84.127.xxx.xxx;branch=z9hG4bK311a66f7;rport=7123
Max-Forwards: 32
From: "asterisk" <sip:34000xxxxxx@voztele.com>;tag=as5805d795 Mantenim el From original i el To:
To: <sip:9.....@voztele.com>
Contact: <sip:34000xxxxxx@84.127.xxx.xxx:7123;nat=yes>
Call-ID: 69cd18e66b996cf8215b62577977bfb6@voztele.com
CSeq: 103 INVITE Mantenim el número de seqüència
Date: Mon, 06 Jun 2011 16:24:58 GMT
Content-Type: application/sdp
Content-Length: 278
P-RTP-Proxy: YES Marquem que utilitzarem un RTP Proxy per a encaminar l'àudio d'un extrem a l'altre.
Remote-Party-ID: <sip:9.....@voztele.com;user=phone>;privacy=off;party=calling
```

Tant el Remote-Party-ID com el P-Asserted-Identity serveixen per a passar el número de telèfon amb el que s'identificarà l'Asterisk que està trucant. Això ho configurem amb un número fix però ho podria enviar l'Asterisk si es volgués.

```
P-Asserted-Identity: <sip:9.....@voztele.com>
P-PPaid-ID: sip:34000xxxxxx@voztele.com Passem el ID per a comprovar el saldo al server ppaid
v=0
o=root 169170159 169170160 IN IP4 84.124.xxx.xxx
s=Asterisk PBX 1.6.2.18
c=IN IP4 193.22.119.13
```

Interessant aquest punt, canviem la IP per a rebre RTP per la del Proxy que quedarà enmig dels dos extrems.

```
t=0 0
m=audio 30194 RTP/AVP 18 101 Canviem el port 30194 per el que el RTP Proxy rebrà l'àudio del GW.
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20
a=sendrecv
a=nortpproxy:yes
```

Arribem al ppaid, un App-Server que comprova que hi tenen saldo al compte SIP per a realitzar la trucada.

[8],[10],[11],[13] i [15] SIP/2.0 100 Avisem al SIP Proxy que estem cursant la trucada.
[12],[14] i [16] INVITE sip:9992103#00349.....@193.22.119.xx:5060 SIP/2.0

Si hi ha saldo re-enviem l'INVITE tal com ens ha arribat.

[17] SIP/2.0 100 Trying Avisa el Gateway que està intentant la trucada.

Durant el 183 Session Progress canviem el RTP Proxy pel GW al Asterisk. Es poden veure altres senyalitzacions on aquest procés es porta al 200 OK, però l'arquitectura d'aquesta empresa va mantenir aquesta decisió de disseny per a trucades a GW primaris propis.

[18] i [20] SIP/2.0 183 Session Progress
Via: SIP/2.0/UDP 193.22.119.10;branch=z9hG4bK4c0c.fe8e5495.0,SIP/2.0/UDP 193.22.119.20;branch=z9hG4bK4c0c.65ee542.0,SIP/2.0/UDP 84.124.xxx.xxx:7123;received=84.127.xxx.xxx;branch=z9hG4bK311a66f7;rport=7123
From: "asterisk" <sip:34000xxxxxx@voztele.com>;tag=gj-2k5-4ce476fc-000079a6-00166845Rce9af0a5.a
To: <sip:9.....@voztele.com>;tag=3ECD45FC-983
Date: Mon, 06 Jun 2011 16:25:31 GMT
Call-ID: 69cd18e66b996cf8215b62577977bfb6@voztele.com
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 103 INVITE
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE, NOTIFY, INFO, UPDATE, REGISTER
Allow-Events: telephone-event
Contact: <sip:999210300349.....@193.22.119.41:5060>
Record-Route: <sip:193.22.119.10;lr=on>,<sip:193.22.119.20;lr=on;ftag=as5805d795>
Content-Disposition: session;handling=required
Content-Type: application/sdp
Content-Length: 273
v=0
o=CiscoSystemsSIP-GW-UserAgent 8666 6023 IN IP4 193.22.119.xx Es tracta d'un GW Cisco com UA
s=SIP Call
c=IN IP4 193.22.119.xx Aquest es la IP del GW, on espera l'àudio.
t=0 0
m=audio 18664 RTP/AVP 18 101

El port del GW és el 18664, també ofereix el G729 com a codec, així que s'establirà en G729 i el 101 per a DTMF

a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=ptime:20 utilitzem un ptime comú per a no tenir que processar res el RTP d'un extrem a l'altre.
[19] SIP/2.0 183 Session Progress Propaguem el mateix missatge cap al SIP Proxy
[21] SIP/2.0 183 Session Progress Propaguem cap el Asterisk però modificant la IP i port per al RTP
Via: SIP/2.0/UDP 84.124.xxx.xxx:7123;received=84.127.xxx.xxx;branch=z9hG4bK311a66f7;rport=7123
From: "asterisk" <sip:34000xxxxxx@voztele.com>;tag=as5805d795
To: <sip:9.....@voztele.com>;tag=gj-2k5-4ce476fc-000079a6-00166845Rce9af0a5.b
Date: Mon, 06 Jun 2011 16:25:31 GMT
Call-ID: 69cd18e66b996cf8215b62577977bfb6@voztele.com
Server: Cisco-SIPGateway/IOS-12.x
CSeq: 103 INVITE
Allow: INVITE, OPTIONS, BYE, CANCEL, ACK, PRACK, COMET, REFER, SUBSCRIBE, NOTIFY, INFO, UPDATE, REGISTER
Allow-Events: telephone-event
Contact: <sip:999210300349.....@193.22.119.xx:5060>
Record-Route: <sip:193.22.119.10;lr=on>,<sip:193.22.119.20;lr=on;ftag=as5805d795>
Content-Disposition: session;handling=required
Content-Type: application/sdp
Content-Length: 291
v=0
o=CiscoSystemsSIP-GW-UserAgent 8666 6023 IN IP4 193.22.119.xx
s=SIP Call
c=IN IP4 193.22.119.13 L'àudio li diem al Asterisk que l'entregui al RTP Proxy.
t=0 0
m=audio 38218 RTP/AVP 18 101

L'RTP Proxy rebrà l'àudio de l'Asterisk pel port 38218 i com hem vist abans el del GW pel 30194. Serà feina de l'RTP Proxy passar l'RTP entre els dos ports que signifiquen els dos extrems de la trucada. El RTP Proxy exercirà control del flux i remarcarà les dades del protocol de transport i de les capes inferiors.

a=rtpmap:18 G729/8000 [...]
[22],[23],[24],[25] i [26] SIP/2.0 200 OK [...]
CSeq: 103 INVITE [...]

Bàsicament reescriu el 183 Session Progress. L'OK significa que el GW ens avisa que han despenjat la trucada. L'OK el propaguem fins l'Asterisk, que comença a rebre RTP immediatament després de l'OK. Utilitzem el RTP Proxy com hem vist. DURANT AQUEST TEMPS COMENÇA LA TRUCADA I TRANFERÈNCIA RTP

[27],[28],[29],[30] i [31] ACK

Confirmem i propaguem la resposta a l'OK, per a confirmar al GW que la trucada està en marxa. El GW començarà a enviar RTP just després del primer OK però sense l'ACK tallaria la comunicació als pocs segons entenent que hi ha un problema. DURANT AQUEST TEMPS HI HA TRUCADA I TRANFERÈNCIA RTP

[32],[33],[34] i [35] BYE sip:999210300349.....@193.22.119.41:5060 SIP/2.0

Es penja la trucada per part de l'Asterisk (en la captura falta un paquet entre el 31 i 32) però és una falla del programari de la captura, ja que clarament es veu al cos del missatge que és Asterisk qui penja i es propaga fins al GW.

Via: SIP/2.0/UDP 193.22.119.20;branch=z9hG4bK1c0c.9f127ca6.0
Via: SIP/2.0/UDP 84.124.xxx.xxx:7123;received=84.127.xxx.xxx;branch=z9hG4bK3f5bedf2;rport=7123
Route: <sip:193.22.119.10;lr=on>
Max-Forwards: 32
From: "asterisk" <sip:34000xxxxxx@voztele.com>;tag=as5805d795
To: <sip:9.....@voztele.com>;tag=gj-2k5-4ce476fc-000079a6-00166845Rce9af0a5.b
Call-ID: 69cd18e66b996cf8215b62577977bfb6@voztele.com
CSeq: 104 BYE Es dona un nou número de seqüència per a rebre la confirmació l'Asterisk del BYE
X-Asterisk-HangupCause: Normal Clearing Es penja normalment la trucada.
X-Asterisk-HangupCauseCode: 16
Content-Length: 0
[36],[37],[38],[39] i [40] SIP/2.0 200 OK

Es confirma que es penja la trucada responent al BYE i es propaga fins l'Asterisk.

Via: SIP/2.0/UDP 193.22.119.10;branch=z9hG4bK1c0c.8e38dfe4.0,SIP/2.0/UDP
193.22.119.20;branch=z9hG4bK1c0c.9f127ca6.0,SIP/2.0/UDP
84.124.xxx.xxx:7123;received=84.127.xxx.xxx;branch=z9hG4bK3f5bedf2;rport=7123
From: "asterisk" <sip:34000xxxxxx@voztele.com>;tag=gj-2k5-4ce476fc-000079a6-00166845Rce9af0a5.a
To: <sip:9.....@voztele.com>;tag=3ECD45FC-983
Date: Mon, 06 Jun 2011 16:25:55 GMT
Call-ID: 69cd18e66b996cf8215b62577977bfb6@voztele.com
Server: Cisco-SIPGateway/IOS-12.x
Content-Length: 0
CSeq: 104 BYE

Respon al número de seqüència 104 per tal que l'Asterisk sàpiga a quina query correspon la resposta.