



عنوان

زیرسیستم چنددرسانه ای مبتنی بر پروتکل اینترنت

نگارش
جابر دانش آموز - ۹۳۲۵۸۹۳

استاد راهنما
دکتر محمد حسین منشی

۹۷ شهریور

تشکر و قدردانی:

با تشکر از دکتر محمدحسین منشی و دکتر محمدرضا حیدرپور بابت راهنمایی ها و زحمات ایشان که موجب به ثمر رسیدن این پروژه شد.

چکیده

زیرسیستم چندرسانه‌ای مبتنی بر پروتکل اینترنت، یک معماری برای انتقال داده‌ی چندرسانه‌ای (مانند صدا و تصویر) بر روی شبکه اینترنت می‌باشد. این معماری که به آن IMS گفته می‌شود، قرار است جایگزین سوئیچ‌های مداری در شبکه سلولی شود. مزیت اصلی این سیستم نسبت به سیستم‌های کنونی صدا بر روی IP، فراهم کردن کیفیت سرویس (QoS) برای ارتباطات تعاملی نظیر تماس صوتی و ویدیوئی است. پس از انجام تحقیقات در مورد نحوه پیاده‌سازی، استانداردها، کاربردها و قابلیت‌های IMS، این سیستم به وسیله clearwater پیاده‌سازی شد. clearwater، یک پروژه متن‌باز است که هسته IMS را پیاده‌سازی کرده است. با استفاده از clearwater، سیستم IMS به صورت یکپارچه و در مقیاس کوچک پیاده‌سازی شد. سپس، با استفاده از IMS و در بستر اینترنت، تماس تلفنی با کیفیت بالا بر روی IP برقرار شد و کارکرد قسمت‌های مختلف این سیستم مورد آزمایش قرار گرفت. در مرحله‌ی بعد، کیفیت سرویس و سیگنالینگ‌های مورد استفاده در IMS و سیستم‌های متداول صدا بر روی IP، مورد مقایسه قرار گرفتند. همچنین، IMS به شبکه‌ی سلولی نسل چهارم (4G) متصل شد. با انجام این کار، تماس تلفنی در بستر شبکه‌ی سلولی و از طریق IMS برقرار شد. در مرحله‌ی آخر نیز سعی شد که ارتباط بین IMS و شبکه‌ی تلفن ثابت برقرار شود تا کاربران تلفن ثابت و IMS بتوانند با یکدیگر ارتباط تلفنی برقرار کنند. با توجه به عدم دسترسی به تجهیزات مورد نیاز در زمان مناسب، انجام این کار میسر نگردید.

کلمات کلیدی: زیرسیستم چندرسانه‌ای، شبکه سلولی، کیفیت سرویس، IMS

فهرست مطالب

۴۴	۷ مقایسه کیفیت سرویس در VOIP و IMS
۴۴	۱.۷ نیاز به کیفیت سرویس
۴۴	۲.۷ مقایسه کیفیت سرویس در شبکه اینترنت
۴۶	۱.۲.۷ مقایسه سیگنالینگ VOIP و IMS در شبکه اینترنت
۵۰	۳.۷ مقایسه کیفیت سرویس در شبکه سلولی
۵۱	۸ اتصال IMS به شبکه 4G
۵۱	۱.۸ پیاده‌سازی نسل چهارم ارتباطات
۵۲	۲.۸ اتصال IMS به هسته‌ی شبکه 4G
۵۴	۳.۸ برقراری تماس تلفنی بین کاربران 4G توسط IMS
۵۵	۹ نتیجه‌گیری
۵۷	۱۰ مسائل و مشکلات Virtualbox
۵۷	۱.۰ نصب
۵۸	۲.۰ خطای اجرا کردن ماشین مجازی
۶۱	۱۱ مسائل و مشکلات آزمایش clearwater
۶۴	پ اتصال به تلفن ثابت
۶۴	پ.۱ روش‌های استفاده از SIP trunk
۶۶	پ.۲ اتصال SIP trunk به IMS
۶۷	پ.۱.۲.۱ نصب و پیکربندی IBCF
۶۹	پ.۲.۲.۱ پیکربندی ENUM
۷۰	پ.۳.۲.۱ پیکربندی BGCF
۷۲	۱۲ مراجع

فهرست تصاویر

۱	۱.۱	معماری نسل اول شبکه‌های سلولی
۲	۲.۱	معماری نسل سوم شبکه‌های سلولی
۳	۳.۱	معماری نسل چهارم شبکه‌های سلولی
۴	۴.۱	معماری نسل پنجم شبکه‌های سلولی
۹	۱.۲	معماری لایه‌ای شبکه‌ی IMS
۱۲	۲.۲	ارتباط بین MGCF و IMS-MGW
۱۳	۳.۲	ارتباط بین MRFC و MRFP
۱۴	۴.۲	معماری لایه‌ی اپلیکیشن و ارتباط این لایه با سایر المان‌های IMS
۱۵	۵.۲	مثال از ترتیب فرآخوانی اپلیکیشن سرورها و کارکرد iFC
۱۷	۶.۲	معماری مرجع برای IMS
۱۸	۱.۳	مقایسه‌ی میزان سودآوری عوامل مختلف پس از پیاده‌سازی IMS
۱۹	۲.۳	نمایی از سوئیچ‌های مداری که در مراکز تلفن و مراکز سوئیچینگ تلفن همراه به کار می‌رود
۲۰	۳.۳	استفاده از روش ستونی و روش enabler مشترک
۲۲	۴.۳	پشتیبانی از شبکه‌های دسترسی مختلف توسط IMS
۲۴	۵.۳	ارتباط کاملاً دوطرفه‌ی بی‌سیم
۲۸	۱.۴	معماری clearwater
۳۶	۱.۶	Import Appliance
۳۷	۲.۶	Import OVF file
۳۷	۳.۶	صفحه‌ی ورود
۳۸	۴.۶	ثبتنام
۳۹	۵.۶	مدیریت کاربران
۴۰	۶.۶	صفحه‌ی پیکربندی حساب کاربری در نسخه‌ی دسکتاب اپلیکیشن zoiper
۴۱	۷.۶	توپولوژی مورد استفاده برای برقراری تماس تلفنی از طریق IMS
۴۳	۸.۶	خروجی اجرای آزمایش زنده‌ی clearwater
۴۷	۱.۷	ثبتنام SIP
۴۸	۲.۷	پیام‌های مبادله‌شده برای ثبتنام SIP در IMS
۴۹	۳.۷	پیام‌های مبادله‌شده برای ثبتنام SIP در VOIP
۴۹	۴.۷	اطلاعات ارسال شده به کاربر در آخرین مرحله‌ی ثبتنام SIP توسط سرور
۵۳	۱.۸	معماری مورد استفاده در پیاده‌سازی oai
۵۴	۲.۸	توپولوژی ارتباط IMS و 4G و برقراری تماس صوتی به‌وسیله‌ی این شبکه
۵۸	۱.۹	صفحه‌ای که پس از reboot نمایش داده می‌شود
۵۹	۲.۹	مدیریت MOK

ب. ۱. نحوه ارتباط SIP trunk با IMS و اتصال به PSTN

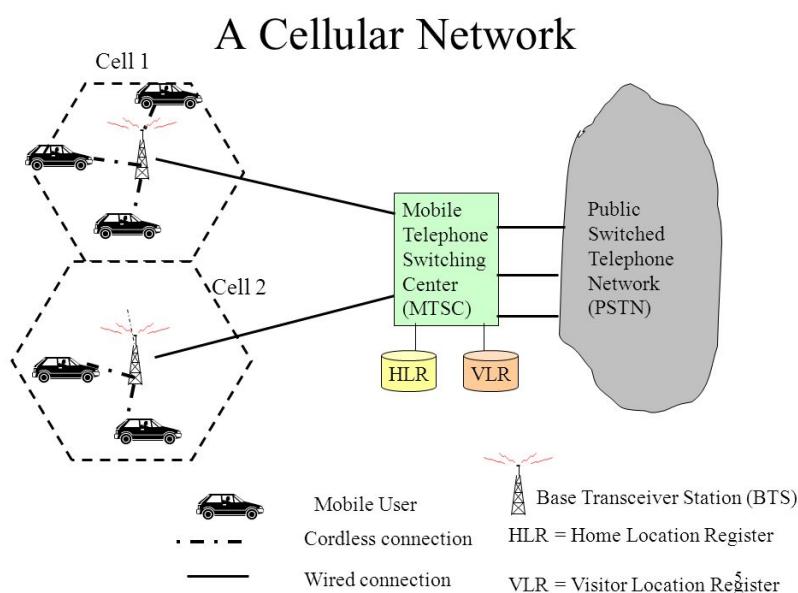
فصل ۱

مقدمه

۱.۱ تاریخچه شبکه‌های سلولی

نسل اول

در شبکه‌های سلولی نسل اول (1G)، از روش‌های مخابرات آنالوگ استفاده می‌شد که مشابه رادیوهای آنالوگ عمل می‌کردند. در مخابرات آنالوگ، از پهنای باند فرکانسی به صورت کارآمد استفاده نمی‌شود و نسبت به سیستم‌های دیجیتال، ظرفیت پایین‌تری برای ارائه خدمات دارند. این سیستم‌ها، صرفاً قادر به ارائه خدمات تماس صوتی به کاربران خود بودند. معماری 1G، در شکل ۱.۱ نشان داده شده است [۱].

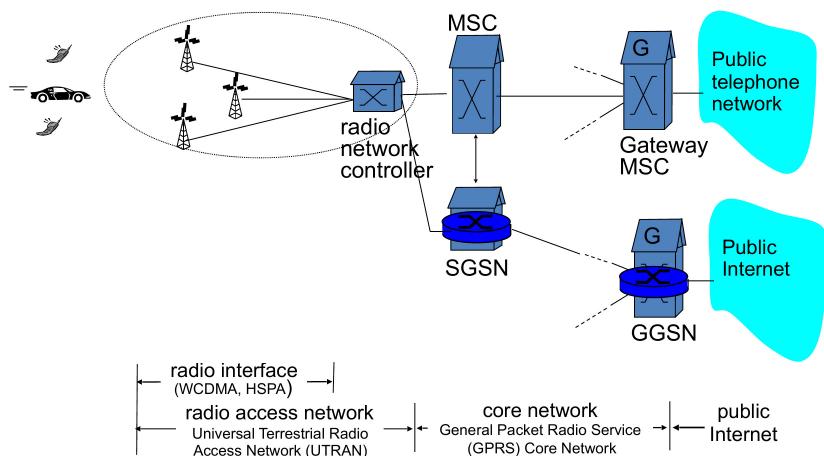


شکل ۱.۱: معماری نسل اول شبکه‌های سلولی

فصل دوّم

با ظهور نسل دوم شبکه‌های سلولی (G2)، برای اولین بار فناوری مخابرات دیجیتال مورد استفاده قرار گرفت. مخابرات دیجیتال، امکان استفاده بهینه‌تر از دامنه‌ی رادیویی را فراهم کرد. در این روش، انتقال داده‌ها به صورت دیجیتال صورت می‌گیرد و این امر باعث بالا رفتن کیفیت مکالمات صوتی نسبت به روش آنالوگ می‌شود. با وجود این‌که این سیستم‌ها در آغازِ کار، فقط برای پشتیبانی از سرویس صوتی طراحی شده بودند، اما با بهبودهایی که صورت گرفت، قادر به ارائه سرویس پیامک نیز شدند. سیستم نسل دوم GSM^۱ که به عنوان تکنولوژی مورد استفاده در اتحادیه‌ی اروپا طراحی شده بود، به محبوب‌ترین سیستم مخابراتی در جهان تبدیل شد [۱].

موفقیت G2 و رشد و گسترش اینترنت، باعث شد که اپراتورهای تلفن همراه، این دو سرویس را با یکدیگر ترکیب کنند. با این کار، کاربران تلفن همراه توانستند از طریق شبکه‌ی سلولی به اینترنت متصل شوند. این نسل که 2.5G نامیده می‌شود، در کنار سرویس صوتی، سرویس دیتا نیز به کاربران خود ارائه می‌دهد. برای ارائه سرویس دیتا، باید علاوه بر شبکه‌ی سوئیچ مداری^۲، از شبکه‌ی سوئیچ بسته‌ای^۳ نیز استفاده شود. همچنین، لازم است که رابط هوایی^۴ نیز به گونه‌ای تغییر کند که قادر به پشتیبانی از صوت و دیتا باشد. GSM با استفاده از سرویس GPRS^۵، موارد فوق را پیاده‌سازی کرد [۱].



شکل ۲.۱: معماری نسل سوم شبکه‌های سلولی

نسل سوم

گسترش روز افزون اینترنت و نیاز کاربران به اینترنت سریع‌تر و حجم دیتای بیشتر، تحولی دیگر در حوزه‌ی شبکه‌های سلولی رقم زد. در ابتدا، طراحان با استفاده از روش‌هایی مانند EDGE^۶، کارایی سیستم‌های G2 و

^۱ سرویزه‌ی عبارت Global System for Mobile communication و به معنای سیستم جهانی برای ارتباطات سیار

^۲ Circuit Switch

^۳ Packet Switch

^۴ Air Interface: لینک ارتباطی بین دو ایستگاه در مخابرات بی‌سیم که شامل لایه‌های فیزیکی و پیوند داده است.

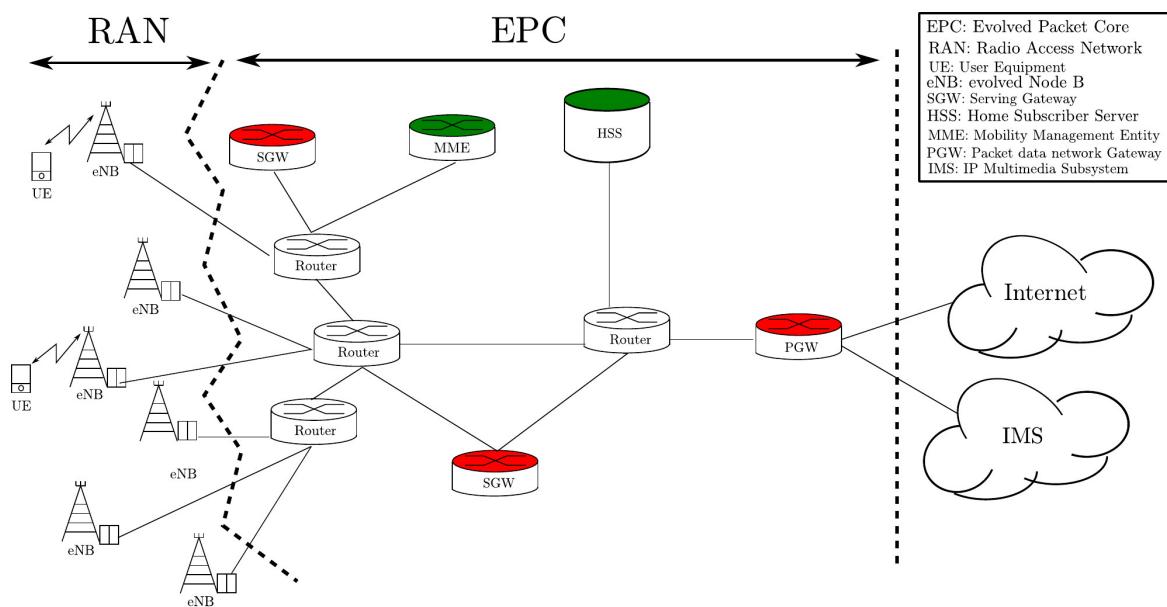
^۵ سرویزه‌ی عبارت General Packet Radio Service

^۶ Enhances Data rate for GSM Evolution

2.5G را افزایش دادند. پس از آن، نسل سوم شبکه‌های سلولی(3G) معرفی شد. در نسل سوم، هسته‌ی شبکه‌ی سلولی تغییری نکرد اماً تکنولوژی ناحیه‌ی دسترسی رادیویی، به طور کامل تغییر کرد. در ناحیه‌ی دسترسی رادیویی نسل سوم، از تکنولوژی WCDMA^۷ و یا TD-SCDMA^۸ استفاده می‌شود. این روش‌ها، باعث شد که 3G بتواند از حداکثر نرخ ارسال داده‌ی^۹ بیشتری پشتیبانی کند. در 3G نیز مانند 2.5G، شبکه‌ی دیتا کار خود را به صورت موازی با شبکه سرویس صوتی انجام می‌دهد. معماری 3G، در شکل ۲.۱ نشان داده شده است [۱].

نسل چهارم

تفاوت اصلی نسل چهارم(4G) با نسل‌های قبل از خود، عدم پشتیبانی از شبکه‌ی سوئیچ مداری است. در این نسل، تماس صوتی نیز از طریق شبکه سوئیچ بسته‌ای برقرار می‌شود. از آنجایی که شبکه‌های سوئیچ بسته‌ای، به طور ذاتی و اوّلیه، کیفیّت سرویس(QoS) را تضمین نمی‌کنند، نیاز به تمهیدات خاص و جداگانه‌ای برای فراهم کردن کیفیّت سرویس و ارائه‌ی سرویس صوتی با کیفیّت می‌باشد. همچنین در این نسل، استفاده از فناوری‌هایی مانند MIMO^{۱۰} و سلول‌های کوچک، باعث افزایش کارایی و نرخ دیتا شده است. معماری 4G، در شکل ۳.۱ نشان داده شده است [۱].



شکل ۳.۱: معماری نسل چهارم شبکه‌های سلولی

^۷سروازه‌ی عبارت Wideband Code Division Multiple Access

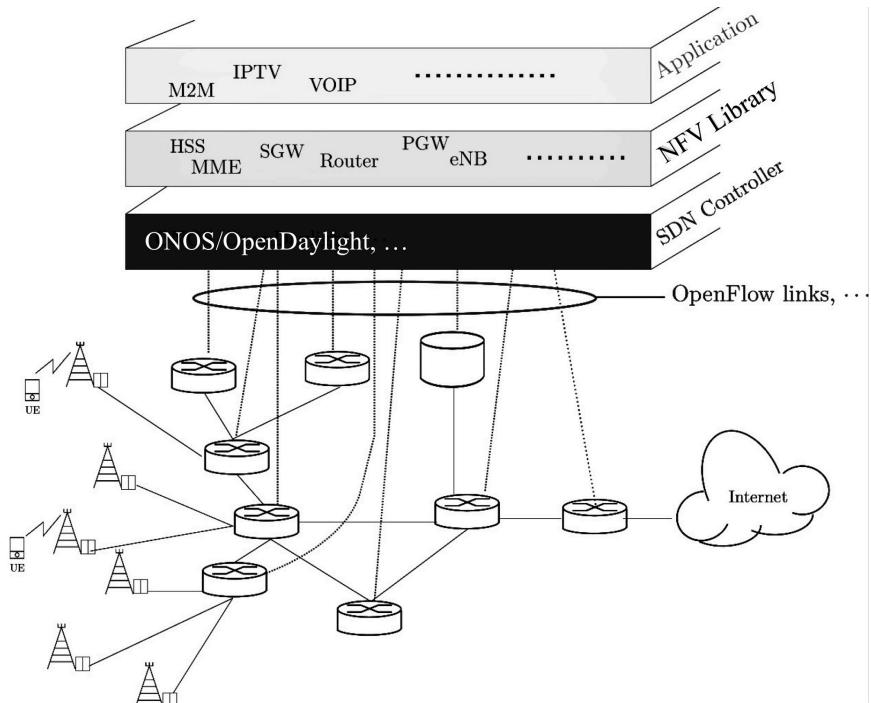
^۸سروازه‌ی عبارت Time Division Synchronous Code Division Multiple Access

^۹Maximum Data Rate

^{۱۰}سروازه‌ی عبارت Multiple Input and Multiple Output و به معنای چند ورودی و چند خروجی

فصل پنجم

نیازمندی‌ها و استانداردهای کمی و کیفی تعریف شده برای نسل پنجم شبکه‌های سلولی (5G)، باعث تفاوت قابل توجه این نسل، با نسل‌های پیشین خود می‌شود. ظرفیت^{۱۱} بالا، تأخیر کم، پشتیبانی از تعداد زیادی کاربر و مصرف توان کم، از جمله این موارد می‌باشد. حدّاًکثر نرخ دیتا در 5G، بیش از ۱۰۰۰ برابر بیشتر از 4G است و این موضوع، خود بیانگر تحوّلی بزرگ در شبکه‌های سلولی است. [از آنجایی‌که برای دستیابی به این



شکل ۴.۱: معماری نسل پنجم شبکه‌های سلولی

استانداردها و نیازمندی‌ها، منابع بسیار زیادی مورد نیاز است، 5G راه حل‌های جایگزینی برای این کار ارائه می‌دهد. بیشتر این راه حل‌ها، بر اساس روش‌های cloudification^{۱۲}، virtualization^{۱۳} و softwarization^{۱۴} می‌باشند.

۲۰.۱ دلایل پیدایش IMS

همانطور که در بخش ۱ توضیح داده شد، طبق استانداردهای نسل ۴ به بعد، تماس‌های صوتی باید از طریق سویچ‌های بسته‌ای منتقل شوند. بسیاری از اپراتورهای تلفن همراه، ادعا می‌کنند که نسل چهارم ارتباطات را پیاده‌سازی کرده‌اند، اماً این امر بیشتر در زمینه‌ی پهنه‌ی باند و افزایش سرعت اینترنت همراه محقق شده است.

Throughput^{۱۱}

^{۱۲} پیاده‌سازی نرم‌افزاری شبکه (software defined network)

^{۱۳} مجازی‌سازی

^{۱۴} انتقال دادن سرویس بر روی ابر

هنوز بسیاری از اپراتورهای تلفن همراه، تماس‌های صوتی را از طریق سوئیچ‌های مداری برقرار می‌کنند. هنگامی که شما اینترنت نسل ۴ را در اختیار دارید و در نوار وضعیت تلفن همراه شما علامت 4G نمایش داده می‌شود، اگر اقدام به برقراری تماس تلفنی کنید یا کسی با شما تماس بگیرد، متوجه خواهید شد که علامت 4G در نوار وضعیت، به علامت H^{۱۵} تغییر می‌آید. دلیل این تغییر وضعیت، عدم پشتیبانی اپراتور موردنظر (به خصوص اپراتورهای ایرانی همراه اول، ایرانسل و رایتل) از برقراری تماس تلفنی مبتنی بر تکنولوژی نسل ۴ است؛ لذا این اپراتورها، سرویس اینترنت را با استفاده از تکنولوژی نسل چهارم و سرویس تماس صوتی را با استفاده از تکنولوژی نسل سوم ارائه می‌دهند.

راحلی که برای مشکل مذکور به ذهن می‌رسد، استفاده از تکنولوژی VOIP^{۱۶} به جای سوئیچ‌های مداری است. از آنجایی که اپراتورهای تلفن همراه، شبکه‌ی داخلی مخصوص به خود را دارند و کاربر با اتصال به ایستگاه‌های مخابراتی^{۱۷}، می‌تواند آدرس IP به دست آورد، بدیهی است که می‌تواند از سرویس VOIP استفاده کند و با استفاده از این سرویس، تماس صوتی برقرار نماید. اپراتور موردنظر نیز می‌تواند با استفاده از درگاه‌های مختلف، برقراری تماس بین مشترک خود و مشترکین خارج از شبکه‌ی خود (چه مشترکین سایر اپراتورها و چه مشترکین تلفن ثابت) را فراهم کند.

مشکل اصلی را حل فوق، عدم تضمین کیفیت سرویس است؛ به این معنی که با وجود بالا بودن کیفیت ارتباط دستگاه کاربر با ایستگاه مخابراتی^{۱۸}، ممکن است کاربر نتواند تماس صوتی برقرار کند یا تماس صوتی با کیفیت پایین را تجربه کند. دلیل اصلی این مشکل، وجود تراکم^{۱۹} در لینک‌های داخلی و یا نودهای^{۲۰} شبکه‌ی اپراتور می‌باشد. همچنین، ممکن است تعداد زیادی کاربر به یک ایستگاه مخابراتی متصل شده باشند. از آنجایی که ایستگاه مخابراتی، باند فرکانسی^{۲۱} خود را بین تمام کاربران تقسیم می‌کند، به کاربران سهم کمتری از باند فرکانسی می‌رسد و در نتیجه، نرخ ارسال و دریافت دیتای کاربران پایین می‌آید. بنابراین مشکل عدم امکان برقراری تماس و یا تماس با کیفیت پایین به وجود می‌آید. از آنجایی که اپراتورهای تلفن همراه، موظف به ارائه‌ی سرویس تماس صوتی با کیفیت در بیشتر نقاط جغرافیایی هستند و همچنین وجود اهمیت برقراری تماس صوتی در شرایط مختلف و خاص، در صورت تحقق نشدن این امر، مشکلات زیادی به وجود خواهد آمد. جریمه شدن اپراتور موردنظر به دلیل عدم توانایی ارائه‌ی سرویس مناسب و همچنین نارضایتی مشترکین آن اپراتور، اصلی‌ترین مشکلات پیش رو هستند.

^{۱۵} حرف H، به عنوان نماد عبارت HSPA به کار می‌رود. عبارت HSPA، سرویسی High Speed Access Packet و به معنی دسترسی سریع به بسته‌ها است. این عبارت، نام تکنولوژی اینترنت نسل سوم ارتباطات است.

^{۱۶} سرویسی عبارت IP Voice Over و به معنای صدا بر روی IP Base Station^{۱۷}

^{۱۸} بالا بودن نسبت سیگنال به نویز که به آن Signal to Noise Ratio SNR می‌گویند. زمانی که SNR بالا باشد، کاربر قادر است اطلاعات زیادی را در زمان کم و به درستی، به ایستگاه مخابراتی بفرستد.

^{۱۹} Congestion

^{۲۰} مسیریاب‌ها و سوئیچ‌های درون شبکه

^{۲۱} البته روش‌های دیگر تقسیم مانند تقسیم زمان و ... نیز مورد استفاده قرار می‌گیرند.

از آنجایی که اهمیت امکان برقراری تماس صوتی و همچنین کیفیت سرویس در تماس صوتی (یا ویدیوئی) بسیار حیاتی تر از سایر سرویس‌ها مانند سرویس دریافت فایل از اینترنت است، می‌توان تدبیری اندیشید تا بسته‌های مربوط به تماس صوتی، اولویت بیشتری برای مسیریابی و گذر از شبکه داشته باشند. با انجام این کار، مشکل مذکور حل خواهد شد. با پیکریندی سرورهای VOIP و همچنین نودهای داخلی شبکه اپراتور، می‌توان این عمل را انجام داد؛ اما باز هم یک مشکل اساسی باقی خواهدماند. سرویس‌های متداول VOIP، قادر نیستند که با تعامل با المان‌های معماری نسل چهارم و پنجم، پنهانی باند تضمین شده‌ای را برای کاربر فراهم کنند؛ به این معنی که نمی‌توانند به وسیله‌ی ارتباط با یک ایستگاه مخابراتی که دستگاه کاربر به آن متصل است، از ایستگاه مخابراتی بخواهند که به کاربر مورد نظر، سهم بیشتری از باند فرکانسی و شبکه‌ی دسترسی رادیویی^{۲۲} اختصاص دهد تا کاربر بتواند تماس صوتی با کیفیت برقار کند. همچنین، در صورت استفاده از سرویس VOIP، ایجاد تماس صوتی بین مشترکین یک اپراتور با مشترکین تلفن ثابت و مشترکین سایر اپراتورها، دشواری‌های خاصی را به وجود می‌آورد.

وجود مسائل و مشکلات فوق، باعث شکل‌گیری سیستم جدیدی به نام IMS^{۲۳} شد. استانداردهای این سیستم، توسط سازمان 3gpp تدوین شده‌اند. سیستم IMS را می‌توان تکامل سیستم‌های VOIP دانست. سیستم IMS، به راحتی می‌تواند با المان‌های معماری نسل چهارم و پنجم ارتباطات تعامل کند و برخورداری از کیفیت سرویس در تماس‌های صوتی (یا ویدیوئی) را برای کاربران فراهم کند. المان‌های پیاده‌سازی شده در IMS، ایجاد تماس صوتی بین مشترکین یک اپراتور با مشترکین سایر اپراتورها، مشترکین تلفن ثابت و همچنین مشترکین سرویس‌های VOIP را بسیار آسان می‌کند.

استفاده از IMS، راهی برای کنار گذاشتن سوئیچ‌های مداری و ورود به عرصه‌ی سوئیچ‌های بسته‌ای در صنعت مخابرات است. با استفاده از IMS، هزینه‌های شبکه و سایر هزینه‌های اپراتورها کاهش می‌ابد و اپراتورها، قادر به ارائه‌ی سرویس‌های جدید و نوظهور، در کنار سرویس‌های قبلی خود می‌شوند. مشترکین اپراتورها، از سرویس‌های جدیدی بهره‌مند می‌شوند و سرویس‌های قبلی را با کیفیت بیشتری تجربه می‌کنند^[۲].

همان‌طور که در شکل ۱.۳. دیده می‌شود، المان PGW، نقطه‌ی اتصال IMS به شبکه‌ی نسل چهارم ارتباطات است. در واقع، نقطه‌ی اتصال IMS به شبکه‌های سلولی، همان نقطه‌ای است که شبکه‌ی سلولی را به شبکه‌ی اینترنت وصل می‌کند. لذا، برای اتصال IMS به شبکه‌ی 5G نیز از المان PGW استفاده می‌شود.

۳.۱ کارهای انجام‌شده

در این پژوهه، ابتدا تحقیقات و مطالعاتی در مورد IMS و معماری آن و همچنین، مزایا و کاربردهای این سیستم انجام شد. پس از بررسی پیاده‌سازی‌های مختلف IMS، پیاده‌سازی متن‌باز clearwater (بخش ۴) انتخاب

^{۲۲} Radio Access Network که به آن RAN می‌گویند.
^{۲۳} IP Multimedia Subsystem سرویزه‌ی عبارت

شده clearwater را می‌توان به صورت توزیع شده و یا یکپارچه پیاده‌سازی کرد. پیاده‌سازی‌های توزیع شده، برای پشتیبانی از تعداد زیادی کاربر مورد استفاده قرار می‌گیرند؛ در حالی که پیاده‌سازی‌های یکپارچه برای امور تحقیقاتی و همچنین آشنایی با این سیستم مورد استفاده قرار می‌گیرند. لذا، پیاده‌سازی یکپارچه بر روی سکوی مجازی‌سازی انجام شد و از طریق آن، تماس تلفنی در بستر شبکه‌ی داخلی دانشگاه^{۲۴} صورت گرفت. این تماس تلفنی، بین کاربران تلفن همراه و همچنین کاربران رایانه‌ی شخصی صورت گرفت و کیفیت مطلوبی داشت (بخش ۶). پس از برقراری تماس تلفنی، کارکردهای سیستم مورد آزمایش قرار گرفت. این آزمایش، به وسیله‌ی کدهای آماده‌ای که clearwater در اختیار قرار داده است، انجام شد (بخش ۵.۶).

سپس، تحقیقاتی در مورد تأمین کیفیت سرویس توسط این سیستم انجام شد. مقایسه‌ی کیفیت سرویس در IMS و سیستم‌های VOIP متداول، در دو بخش صورت گرفت. در بخش اول که از IMS به عنوان یک ارائه‌دهنده‌ی سرویس VOIP در بستر شبکه‌ی محلی استفاده می‌شود، کیفیت سرویس در مدیا^{۲۵} مبادله شده مقایسه شد. سپس، سیگنالینگ‌های IMS با سیگنالینگ‌های یکی از سرویس‌های VOIP متداول مقایسه شدند. برای انجام این کار، سیستم VOIP متن‌باز Asterisk پیاده‌سازی شد و بسته‌های مبادله شده مورد ارزیابی و بررسی قرار گرفتند. در بخش دوم نیز، کیفیت سرویس IMS در صورت استفاده در شبکه‌های سلولی، با کیفیت سیستم‌های VOIP مورد مقایسه قرار گرفتند.

مستندات و تالار گفتمان clearwater نیز برای آشنایی بیشتر با این سیستم و یادگیری استفاده از قابلیت‌های آن، مورد مطالعه قرار گرفت. سپس، شبکه‌ی سلولی نسل چهارم (4G) با استفاده از پروژه‌ی کارشناسی یکی از دانشجویان دانشکده برق و کامپیوتر و به صورت نرم‌افزاری پیاده‌سازی شد. کاربران این شبکه، می‌توانستند از طریق سیم‌کارت، به این شبکه متصل شوند و از طریق آن، به اینترنت دسترسی داشته باشند. با اتصال IMS به این شبکه به عنوان سرور VOIP، کاربران تلفن همراه، از طریق سیم‌کارت و بدون استفاده از ارتباط وای‌فای، تماس تلفنی برقرار کردند.

پس از آشنایی کامل با پیکربندی و کارکرد المان‌های مختلف معماری clearwater، نحوه ارتباط این سیستم با شبکه‌ی تلفن ثابت مورد بررسی قرار گرفت. به دلیل عدم دسترسی به موقع به تجهیزات سخت‌افزاری و نرم‌افزاری موردنیاز برای ایجاد این ارتباط، پیاده‌سازی عملی صورت نگرفت و انجام این کار، به زمان آینده موكول شد. توضیحات لازم در مورد نحوه ارتباط این دو سیستم به یکدیگر، در پیوست (پ) آورده شده است.

^{۲۴} از هر شبکه‌ای که مبتنی بر پروتکل اینترنت است می‌توان استفاده کرد.
^{۲۵} Media

فصل ۲

معماری IMS

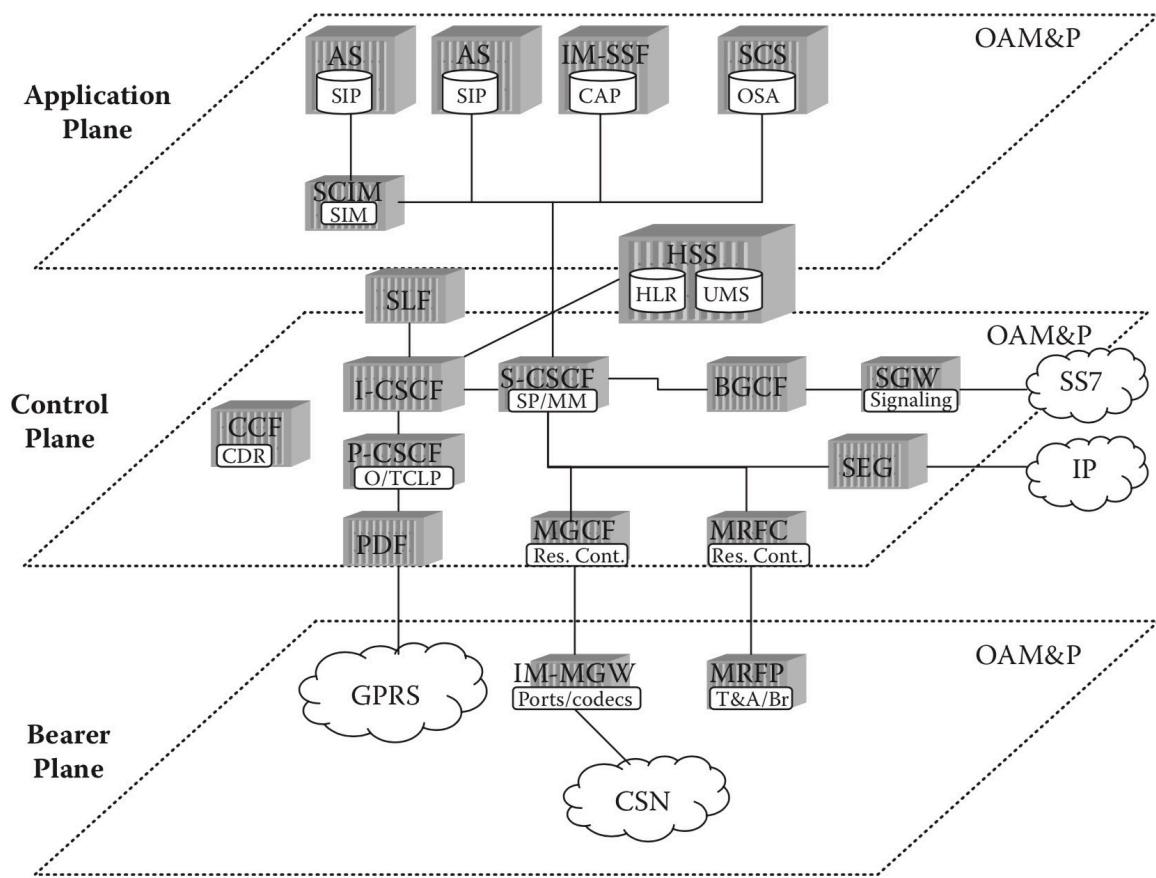
معماری IMS، از سه لایه مختلف تشکیل می‌شود که هر لایه، مستقل از سایر لایه‌ها، وظایف مخصوص به خود را انجام می‌دهد. پایین‌ترین لایه که به آن سطح حامل و گاهًا سطح انتقال گفته می‌شود، دارای منابع فیزیکی ضروری می‌باشد. این منابع، برای ایجاد ارتباط و همچنین حمل payload^۱ (صدا و یا دیتا) بین نقطه‌ی شروع و پایان ارتباط مورد استفاده قرار می‌گیرند. لایه‌ی میانی که به آن سطح کنترلی گفته می‌شود، از یک سری المان هوشمند تشکیل شده است. این المان‌ها، تعیین می‌کنند که آیا کاربر، اجازه‌ی استفاده از شبکه را دارد یا نه. همچنین این لایه، نحوه‌ی مسیریابی تماس یا مسیر انتقال دیتا را تعیین می‌کند که به این عمل، ایجاد جلسه^۲ می‌گویند. بالاترین لایه‌ی این معماری که به آن سطح اپلیکیشن گفته می‌شود، شامل تمام برنامه‌ها و دیتای لازم برای ارائه سرویس به کاربر می‌باشد. شکل ۱.۲، المان‌های معماری IMS را در لایه‌های مختلف و همچنین ارتباط آن‌ها با یکدیگر را نشان می‌دهد. در ادامه، این سه لایه به‌طور خلاصه شرح داده شده‌اند. مطالب این فصل، بر اساس [۲] و [۳] نوشته شده‌اند.

۱۰.۲ سطح کنترلی

در IMS، از آنجایی‌که تمامی سرویس‌ها شامل پروتکل IP هستند، کاربر به منظور فراخوانی هر یک از این سرویس‌ها، نیاز به ایجاد یک جلسه دارد. سطح کنترلی، وظیفه‌ی ایجاد این جلسات را دارد. به‌طور کلی، سطح کنترلی از چهار المان اصلی تشکیل می‌شود:

1. **CSCF**: Call Session Control Function(تابع کنترل جلسه‌ی تماس)
2. **HSS**: Home Subscriber Server(سرور مرکزی مشترکین)

^۱بسته‌های دیتا
^۲Session



شکل ۱.۲: معماری لایه‌ای شبکه‌ی IMS

۳. **BGCF:** Breakout Gateway Control Function(تابع کنترل درگاه خروجی)

۴. **SGW:** Signaling Gateway(درگاه سیگنالینگ)

CSCF

المان CSCF، پردازش تماس‌ها و توابع کنترلی را فراهم می‌کند و کاری مشابه سوئیچ‌های مداری را انجام می‌دهد. همچنین، از آنجایی‌که IMS یک شبکه‌ی مبتنی بر IP است، CSCF توابع کنترل جلسه را نیز فراهم می‌کند. استانداردها، CSCF را به عنوان یک سرور SIP^۳ تعریف می‌کنند که شامل سه جزء عملیاتی می‌باشد:

Proxy CSCF (P-CSCF) •

این المان، نقطه‌ی اولیه‌ی ورود به معماری IMS می‌باشد، گرچه ممکن است که نقطه‌ی اولیه‌ی ورود برای کاربر نباشد. دستگاه کاربر^۴، تمام پیام‌های سیگنالینگ IMS را به P-CSCF می‌فرستد. همچنین، عدم وابستگی

^۳ سروازه‌ی عارت Session Initiation Protocol و به معنای پروتکل راه‌اندزی جلسه User Equipment(UE)^۴

IMS به ناحیه‌ی دسترسی توسط این المان تأمین می‌شود. P-CSCF ممکن است که با PCRF^۵ ادغام شود. PCRF، وظیفه فراهم کردن کیفیت سرویس برای یک جلسه خاص را بر عهده دارد.

Interrogating CSCF (I-CSCF) •

این المان، پیام‌های فوروارد شده از P-CSCF را دریافت می‌کند و تعیین می‌کند که دستگاه کاربر، به کدام S-CSCF متصل خواهد شد. I-CSCF، پس از دریافت پیام ثبت‌نام^۶ از سمت دستگاه کاربر، از HSS پرس‌جو می‌کند که آیا این دستگاه، به یک S-CSCF متصل است یا خیر. در صورتی که این اتصال وجود نداشته باشد، I-CSCF، یک S-CSCF به دستگاه کاربر اختصاص می‌دهد. I-CSCF، تپولوژی، پیکربندی و ظرفیت شبکه‌ی اپراتور را از شبکه‌های همتا مخفی نگه می‌دارد. این کار بهوسیله‌ی تغییر و یا رمزگردن اطلاعات حساس مانند نام دائمی و اطلاعات مربوط به مسیریابی موجود در پیام‌ها انجام می‌شود.

Serving CSCF (S-CSCF) •

این المان، در مرکز IMS جای دارد و عمل پردازش تماس‌ها را انجام می‌دهد؛ درست مانند همان عملی که در سوئیچ‌های مداری انجام می‌شود. همچنین، پردازش سایر جلسات نیز بر عهده این المان می‌باشد. تمام پیام‌های SIP که از دستگاه کاربر و یا به دستگاه کاربر ارسال می‌شوند، از S-CSCF می‌گذرند. اطلاعات وضعیت تمام جلسات دستگاه کاربر را نگه می‌دارد و بنابر موقعیت، به عنوان SIP proxy server و یا SIP registrar عمل می‌کند.

HSS

المان HSS، محل ذخیره دیتای پروفایل تمام مشترکین و همچینین اطلاعات مربوط به احراز هویت کاربران است. دیتای موقت مربوط به مشترکین مانند وضعیت کنونی و مکان ثبت‌نام آن‌ها نیز در HSS ذخیره می‌شود. HSS، علاوه بر این که قادر به انجام تمام وظایف HLR^۷ است، باید قابلیت پشتیبانی از تمام وظایف مربوط به شبکه‌های مبتنی بر IP را نیز داشته باشد. HSS برای برقراری ارتباط با سایر المان‌های IMS نظیر I/S-CSCF، DIAMETER از پروتکل استفاده می‌کند. تشخیص هویت، تعیین اجازه‌ی دسترسی و تأمین امنیت اطلاعات کاربران از کارکردهای اصلی HSS هستند. به دلایل امنیتی، مقیاس‌پذیری و یا ایجاد افزونگی برای بالا بردن قابلیت اطمینان، ممکن است از چند HSS استفاده شود. در این صورت، المان SLF^۸ به معماری IMS اضافه می‌شود. این المان، اطلاعات لازم در مورد این که دیتای پروفایل کاربر در کدام HSS قرار دارد را فراهم می‌کند.

^۵سروژه‌ی عبارت Policy and Charging Rules Function و به معنای تابع قوانین شارژ و سیاست Register^۶

^۷سروژه‌ی عبارت Home Location Register و ^۸سروژه‌ی عبارت Subscription Locator Function

BGCF

المان BGCF مسئول تعیین نقطه‌ی خروج^۹ از شبکه‌ی IMS به دامنه‌ی سوئیچ مداری است. زمانی که S-CSCF دیگر نمی‌تواند مسیریابی جلسه را در شبکه‌ی IMS ادامه دهد (عموماً به این دلیل که نمی‌تواند آدرس را از طریق سرور نام دامنه به دست آورد)، پیام‌های SIP را به BGCF می‌فرستد. المان BGCF تشخیص می‌دهد که آیا این خروج در درون همان شبکه یا در شبکه‌ای دیگر رخ می‌دهد. اگر این عمل مربوط به همان شبکه باشد، یک MGCF^{۱۰} را برای برقراری ارتباط بین شبکه‌ای با دامنه سوئیچ مداری انتخاب می‌کند. در صورتی که این خروج در شبکه‌ی دیگری رخ دهد، BGCF پیام‌های سیگنالینگ جلسه را به BGCF شبکه‌ی مربوطه ارسال می‌کند. در صورتی که شبکه‌ی مقصد، فاقد المان BGCF باشد (یعنی شبکه‌ی مقصد IMS را پیاده‌سازی نکرده باشد)، BGCF این تماس را به شبکه‌ی سوئیچ مداری خودش انتقال می‌دهد^{۱۱}. شبکه سوئیچ مداری نیز برقراری تماس را از طریق مسیریابی استاندارد PSTN^{۱۲} انجام می‌دهد.

SGW

المان SGW معمولاً^{۱۳} به عنوان یکی از المان‌های IMS دیده نمی‌شود زیرا عملکرد آن در لایه‌های پایین‌تر از لایه SIP است. المان SGW می‌تواند به صورت مستقل و یا به صورت یکپارچه با المان‌های دیگر پیاده‌سازی شود. این المان، عمل تبدیل سیگنالینگ‌های لایه انتقال را برای پیام‌های سیگنالینگ کنترلی انجام می‌دهد تا شبکه‌های سیگنالینگ متفاوت را به یکدیگر متصل کند.

۲۰۲ سطح حامل

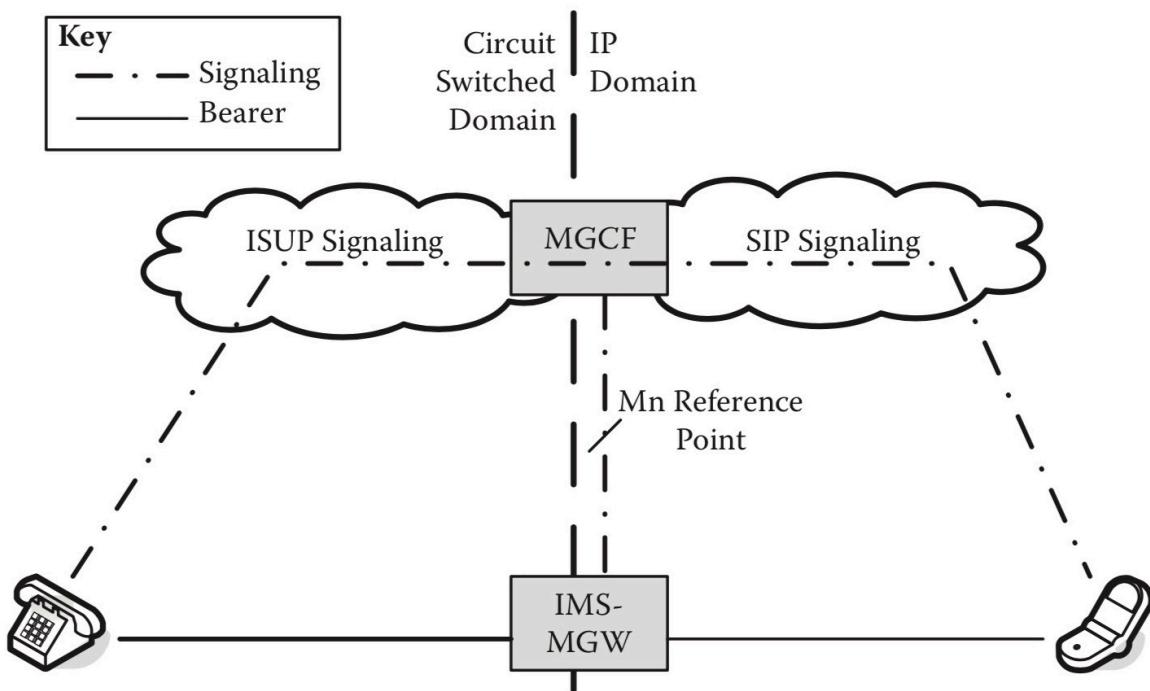
المان‌های MGCF^{۱۴} و MRFC^{۱۵} در سطح کنترلی قرار دارند، اما از آنجایی که این دو المان، ارتباط تنگاتنگی با یکی از المان‌های اصلی سطح حامل به نام IMS-MGW^{۱۶} دارند، این دو المان در این بخش مورد بحث قرار گرفته‌اند.

رابطه‌ی MGCF و IMS-MGW

هر زمان که یک کاربر در دامنه‌ی سوئیچ مداری بخواهد با یک کاربر در دامنه‌ی سوئیچ بسته‌ای ارتباط برقرار کند، باید عمل تبدیل بین حامل‌های مداری و بسته‌ای صورت گیرد. المان MGCF مسئول تبدیل سیگنال‌های

Breakout^۹^{۱۰} سروازه‌ی عبارت Media Gateway Control Function و به معنای تابع کنترل درگاه مديا Hand Off^{۱۱}^{۱۲} سروازه‌ی عبارت Public Switched Telephone Network و به معنای شبکه‌ی سوئیچ عمومی تلفن^{۱۳} سروازه‌ی عبارت Media Gateway Control Function و به معنای تابع کنترل درگاه مديا^{۱۴} سروازه‌ی عبارت Media Resource Function Controller و به معنای کنترل کننده‌ی تابع منابع مديا^{۱۵} کوتاه‌شده‌ی عبارت IMS Media Gateway و به معنای درگاه مديا IMS

کنترلی تلفن ثابت و SIP به یکدیگر است. در انجام این کار، MGCF عمل binding بین شناسه‌های سطح حامل IP و شناسه‌های سطح حامل سوئیچ مداری را انجام می‌دهد. المان MGCF باید این اطلاعات را با IMS-MGW مبادله کند تا تماس بتواند از طریق IMS-MGW برقرار شود(شکل ۲.۲). همچنین، به دلیل این‌که روش‌های کدگذاری زیادی برای جلسات صوتی و تصویری وجود دارد، IMS-MGW باید بداند که از کدام یک از این روش‌ها استفاده شده است.



شکل ۲.۲: ارتباط بین MGCF و IMS-MGW

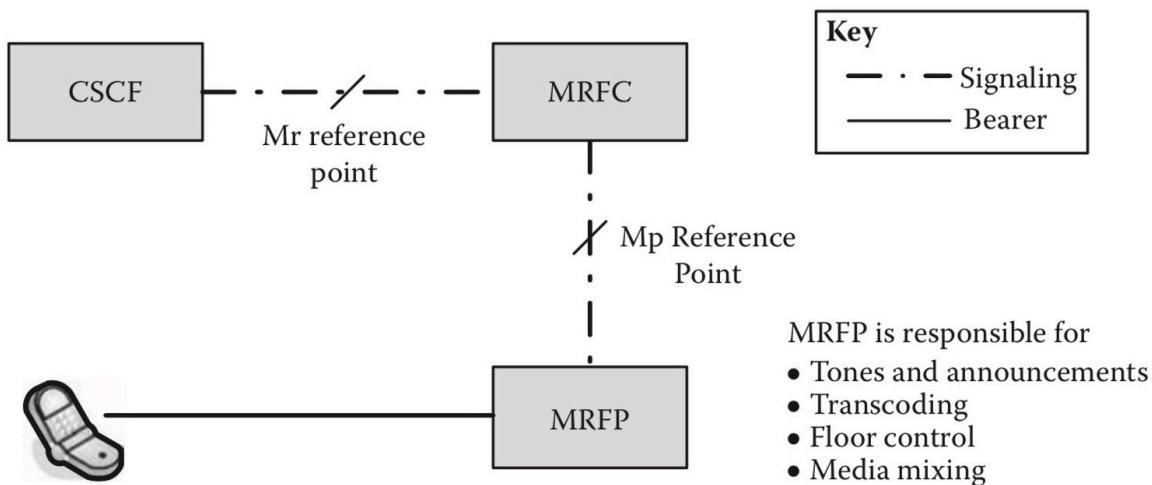
ارتباط بین MRFC و MRFN

المان MRFC در واقع کنترل‌کننده‌ی المان MRFP^{۱۶} و رابط بین MRFP و CSCF است(شکل ۳.۲). ارتباط بین المان MRFC و CSCF از طریق پروتکل SIP انجام می‌شود. المان MRFP مسئول انجام وظایف زیر است:

- آهنگ‌ها و اعلان‌ها

پخش صدای بوق آزاد و یا اشغال در هنگام تماس و یا پخش آهنگ پیشواز و همچنین پخش اعلان‌هایی مانند: مشترک مورد نظر در دسترس نمی‌باشد.

^{۱۶} سرویزه‌ی عبارت Multimeda Resource Function Controller Processor و به معنای پردازنده‌ی MRFC



شکل ۳.۲: ارتباط بین MRFC و MRFP

• تبدیل کدگذاری‌ها

IMS، روش کدگذاری را در هنگام ایجاد جلسه انتخاب می‌کند و هیچ تضمینی وجود ندارد که دو طرف مکالمه از روش کدگذاری یکسان استفاده کنند. لذا نیاز است که فرآیند تبدیل بین کدگذاری‌ها انجام شود.

• کنترل بستر

کنترل بستر، برای انجام مکالماتی که در هر لحظه، فقط یک نفر قادر به ارسال اطلاعات است، ضروری می‌باشد. شخصی که در هر لحظه، بستر را در اختیار دارد، تنها کسی است که می‌تواند اطلاعاتی مانند صوت را ارسال کند و مابقی شرکت‌کننده‌ها در آن جلسه، فقط صدای آن فرد را می‌شنوند. در سرویس‌هایی که اطلاعات ویدیوئی ارسال می‌کنند نیز، تنها تصویر شخصی که بستر را در اختیار دارد، برای سایر افراد نمایش داده می‌شود. سرویس PoC^{۱۷} و PTV^{۱۸} از جمله سرویس‌هایی هستند که به کنترل بستر نیازمند می‌باشند.

• درهم آمیختن مديا

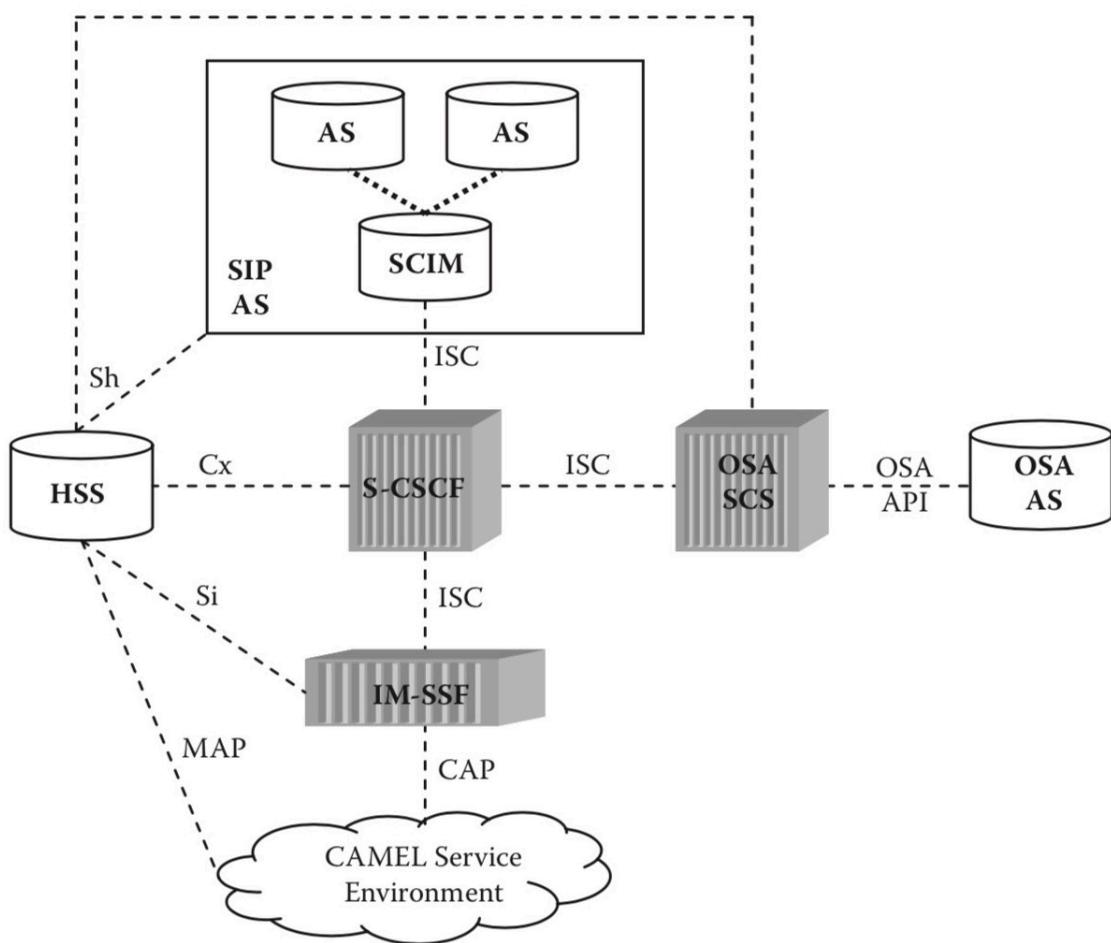
درهم آمیختن مديا، برای انجام کنفرانس‌های صوتی و یا ویدیوئی در بستر IMS مورد استفاده قرار می‌گیرد. شرکت‌کننده‌ها در کنفرانس‌های صوتی، ترکیبی از صدای سایر شرکت‌کننده‌ها را می‌شنوند. در کنفرانس ویدوئی نیز صفحه نمایش به بخش‌های مختلف تقسیم شده و هر بخش، تصویر یک شرکت‌کننده در کنفرانس را نمایش می‌دهد.

^{۱۷} سروژه‌ی عبارت Push to Talk Over Cellular

^{۱۸} سروژه‌ی عبارت Push To Video

۳.۲ سطح اپلیکیشن

اپرаторها می‌توانند با استفاده از سطح اپلیکیشن، اپلیکیشن‌های نوظهور و یا اپلیکیشن‌های ترکیبی را در اختیار کاربران قرار دهند. به اپلیکیشن‌هایی که قابلیت ترکیب شدن با اپلیکیشن‌های فعلی، دستگاه‌ها و منابع محتوا را دارند و می‌توانند با این کار، سرویسی با ارزش‌تر را برای کاربران فراهم کنند، اپلیکیشن‌های ترکیبی گفته می‌شود. اپلیکیشن‌های فعلی می‌توانند اپلیکیشن‌هایی از دنیای اینترنت و یا اپلیکیشن‌هایی باشند که اپرатор آن‌ها را از قبل پیاده‌سازی کرده است. المان‌های شبکه‌ای که سطح اپلیکیشن را تشکیل می‌دهند، عموماً به S-CSCF و HSS متصل می‌شوند.



شکل ۴.۲: معماری لایه اپلیکیشن و ارتباط این لایه با سایر المان‌های IMS

اپلیکیشن سرورهای SIP

اپلیکیشن‌های IMS به طور عمده به وسیله اپلیکیشن سرورهای برمبنای SIP پیاده‌سازی شده‌اند. کاربر، با فرآخوانی این اپلیکیشن سرورها، از خدمات اپرатор بهره‌مند می‌شود. طبق استانداردها، IMS باید بتواند جلسه‌ای

را ایجاد کند که در آن، به طور همزمان از چند اپلیکیشن سرور استفاده می‌شود. این امکان وجود دارد که پاسخ و خرجی یک اپلیکیشن سرور، به عنوان ورودی اپلیکیشن سرور دیگر مورد استفاده قرار گیرد یا این که پس از فراغوایی یک اپلیکیشن سرور، نیاز به فراغوایی اپلیکیشن سرور بعدی نباشد. بنابراین باید مکانیزمی وجود داشته باشد که اپلیکیشن سرور(ها) مناسب را برای کاربر انتخاب کند و همچنین ترتیب فراغوایی اپلیکیشن سرورها را نیز کنترل کند.

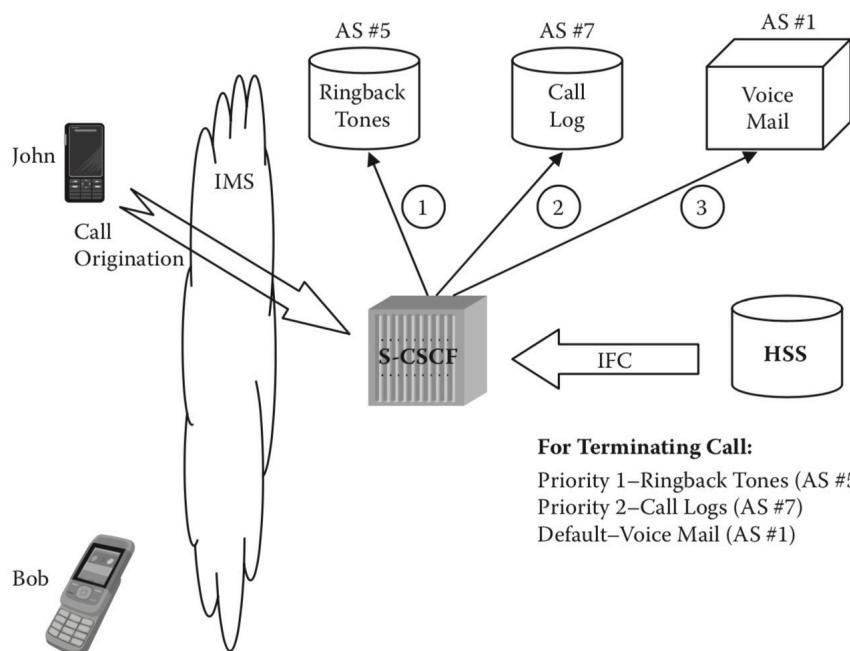
المان HSS در پروفایل دیتای مشترکین، لیستی از اپلیکیشن سرورهای مختص هر کاربر و ترتیب فراغوایی آنها را ذخیره می‌کند. این لیست که iFC^{۱۹} نام دارد، در طول فرایند ثبت‌نام، با S-CSCF به اشتراک گذاشته می‌شود. محتواهای کلی iFC که از HSS به S-CSCF فرستاده می‌شود، شامل موارد زیر است:

- آدرس اپلیکیشن سرور

- اولویت اپلیکیشن سرور(ترتیب فراغوایی)

- اطلاعات اختیاری مربوط به سرویس

- نقطه‌ی راهاندازی^{۲۰}



شکل ۲.۵: مثال از ترتیب فراغوایی اپلیکیشن سرورها و کارکرد iFC

در شکل ۲.۵، اهمیت ترتیب فراغوایی اپلیکیشن سرورها و همچنین کاربرد iFC، به وسیله‌ی یک مثال نشان داده شده است. در مثال شکل ۲.۵، تلفن همراه Bob یک تماس از تلفن همراه John دریافت می‌کند. المان

^{۱۹} سروازه‌ی عبارت initial Factor Criteria و به معنای معیارهای فاکتور اولیه

^{۲۰} نقطه‌ی راهاندازی تعیین می‌کند که چه زمانی iFC فراغوایی شود

S-CSCF به لیست دریافت شده از HSS نگاه می‌کند و اپلیکیشن سرور شماره ۵ را فراخوانی می‌کند و John صدای بوق تماس Bob یا آهنگ پیشواز او را می‌شنود. از آنجایی که Bob می‌خواهد سوابق تماس‌های خود را ثبت کند و نگهدارد، اپلیکیشن سرور شماره ۷ فراخوانی می‌شود. در آخر نیز، چون Bob مشغول انجام کاری است و نمی‌خواهد تلفن John را جواب بدهد، عملی که به صورت پیش‌فرض در نظر گرفته شده انجام می‌شود؛ یعنی اپلیکیشن سرور شماره ۱ فراخوانی شده و تماس به صندوق پست صوتی منتقل می‌شود. در شکل ۲، ترتیب فراخوانی اپلیکیشن سرورها بوسیله‌ی عددی بر روی پیکان‌ها مشخص شده است.

IM-SSF

المان IM-SSF^{۲۱}، طراحی شده است تا اپراتورهای تلفن همراه را قادر سازد که از تکنولوژی GSM کنونی خود یا زیرساخت اپلیکیشن G2، در کنار IMS استفاده کنند. با استفاده از IM-SSF، نیازی به صرف هزینه‌ی بالا برای توسعه و پیاده‌سازی دویاره‌ی این اپلیکیشن‌ها نیست. برای مثال، روش‌های متنوعی که برای خرید سرویس‌های متنوع از اپراتور وجود دارد (مثلاً خرید بسته اینترنت یا خرید بسته ترکیبی اینترنت و پیام کوتاه) قبل‌آن توسعه و پیاده‌سازی شده‌اند. در صورتی که این اپراتور بخواهد از IMS استفاده کند دیگر نیاز نیست اپلیکیشن مربوطه را دویاره و منطبق با معماری IMS طراحی و پیاده‌سازی کند و می‌تواند از همان اپلیکیشن قبلی خود استفاده کند.

مطابق شکل ۴.۲، IM-SSF پیام‌های SIP را از S-CSCF دریافت می‌کند و آن‌ها را به پیام‌های معادل در CAMEL^{۲۲} ترجمه می‌کند. لذا، IM-SSF از طرف المان S-CSCF به عنوان یک اپلیکیشن سرور دیده می‌شود. از آنجایی که استاندارد IMS توسط 3gpp توسعه یافته است، طبیعی است که آن‌ها مکانیزمی را در نظر گرفته‌اند که IMS با سیستم‌های پیشین (که بر اساس استاندارد 3gpp طراحی شده‌اند) تعامل داشته باشد.

درگاه OSA

درگاه OSA، ارتباط بین شبکه‌ی داخلی اپراتور و شرکت‌های شخص ثالث را برقرار می‌کند تا این شرکت‌ها بتوانند با استفاده از زیرساخت اپراتور مربوطه، سرویس خاصی را به مشترکین آن اپراتور ارائه دهند. می‌توان گفت که OSA، تسهیل‌کننده‌ی ورود سایر شرکت‌ها، خصوصاً شرکت‌های کوچک و استارت‌آپ‌های نویا به بازار مخابرات است. المان OSA، علاوه بر نقشی که به عنوان رابط دارد، وظیفه ایجاد امنیت و مدیریت شارژ حساب کاربران را نیز برعهده دارد^{۲۳}. رابطی که درگاه OSA را به اپلیکیشن شرکت شخص ثالث متصل می‌کند، Parlay نام دارد. از آنجایی که سرویس‌های شرکت‌های شخص ثالث، عمدهاً به صورت اپلیکیشن‌های مبتنی بر وب عرضه می‌شوند، یک نسخه‌ی مبتنی بر سرویس وب از این رابط به نام X Parlay طراحی شده است. رابط

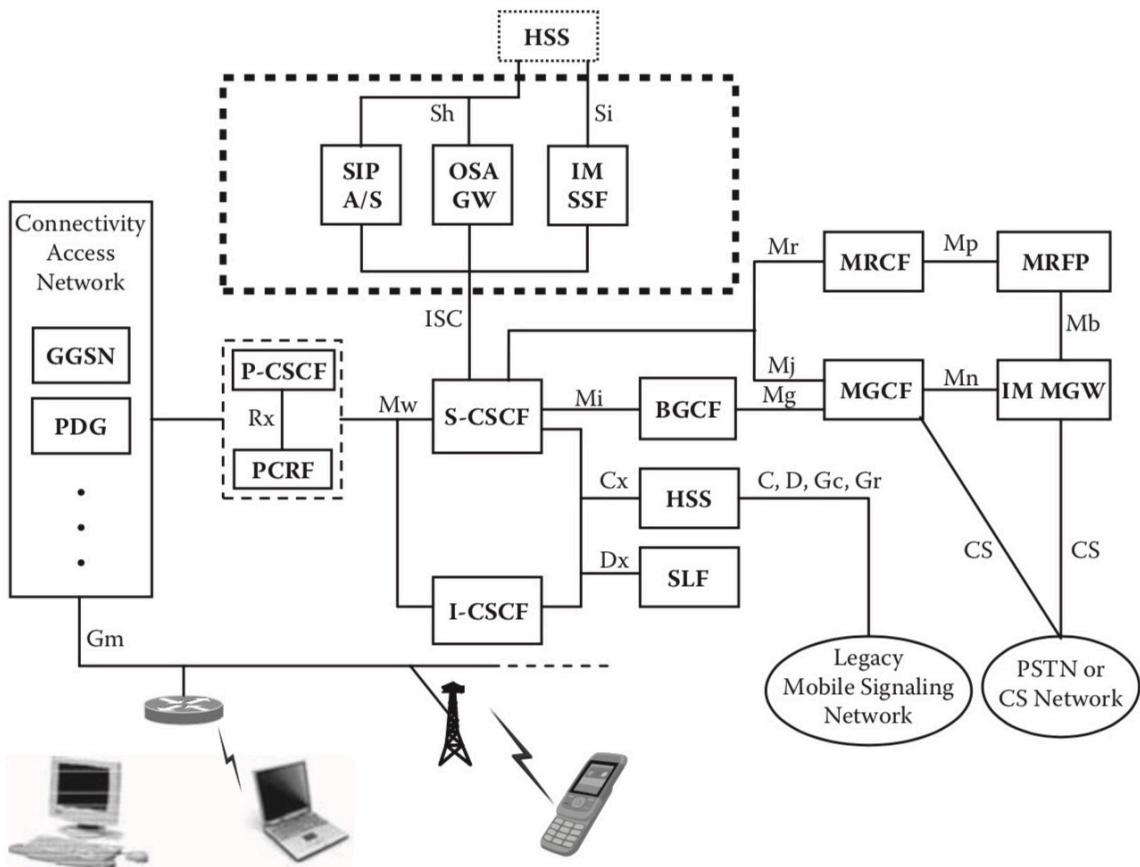
^{۲۱} سرویزه‌ی عارت IP Multimediа Service Switching Function و به معنای عملگر سرویس سوئیچینگ چندسانه‌ای IP

^{۲۲} سرویزه‌ی عارت Customized Application for Mobile Network Enhanced Logic

^{۲۳} این کار را برای شرکت‌های شخص ثالث انجام می‌دهد

Parlay X، توسعه‌دهنگان اینترنت و اپلیکیشن‌های تحت وب را قادر می‌سازد که بدون نیاز به یادگیری رابط Parlay و قواعد پیچیده‌ی آن، اپلیکیشن خود را مناسب با زیرساخت اپراتور طراحی کنند.

شکل ۶.۲، یک نمای کلی از معماری IMS را نشان می‌دهد. در این فصل، بیشتر المان‌ها و رابطه‌ای موجود بین المان‌های این معماری به‌طور خلاصه بیان شد. برخی از بخش‌های معماری IMS نیز در فصل‌های بعد توضیح داده شده‌اند.



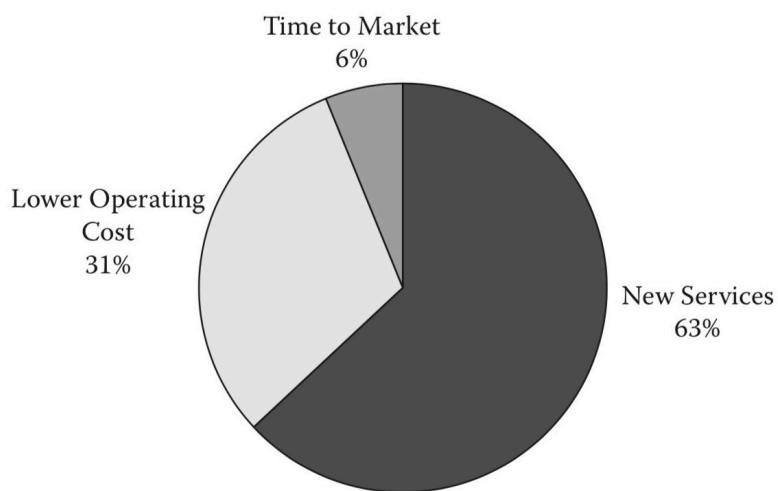
شکل ۶.۲: معماری مرجع برای IMS

فصل ۳

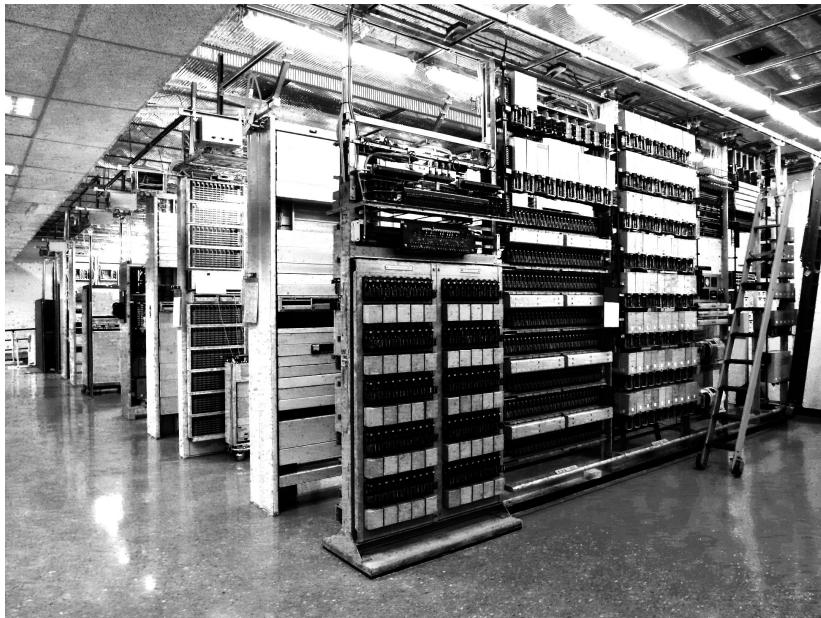
مزایا و کاربردهای IMS

۱۰.۳ کاهش هزینه‌ها، افزایش درآمد و رفع وابستگی

ارائه‌ی سرویس‌های جدید، سودآورترین عامل در استفاده از IMS است. پیاده‌سازی IMS، باعث کاهش هزینه‌های عملیاتی و هزینه‌های مربوط به شبکه می‌شود و برای اپراتورها، سودآور است.^۱ همچنین، کاهش زمان بین طراحی اپلیکشن‌های جدید و ارائه آن‌ها به مشترکین، از دیگر عوامل سودآوری است. شکل ۱.۳، سهم هر یک از عوامل فوق را در سودآوری برای اپراتورهای تلفن همراه نشان می‌دهد^[۲].



شکل ۱.۳: مقایسه‌ی میزان سودآوری عوامل مختلف پس از پیاده‌سازی IMS



شکل ۲.۳: نمایی از سوئیچ‌های مداری که در مراکز تلفن و مراکز سوئیچینگ تلفن همراه به کار می‌رود

۱۰.۳ جایگزین سوئیچ‌های مداری

از آنجایی که سوئیچ‌های مداری به صورت سخت‌افزاری پیاده‌سازی شده‌اند، هزینه‌ی بالایی دارند. همچنین، با زیاد شدن تعداد مشترکین، لازم است که تعداد سوئیچ‌های مداری افزایش یابد. از این رو جایگزین کردن این سوئیچ‌ها با سوئیچ‌های بسته‌ای بسیار مفروض به صرفه است؛ زیرا سوئیچ‌های بسته‌ای می‌توانند با هزینه‌ی کمتری نسبت به سوئیچ‌های مداری، به کاربران بیشتری سرویس ارائه کنند.

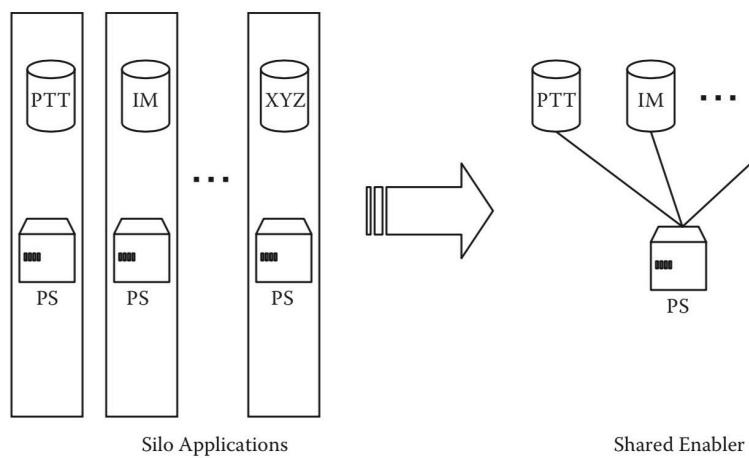
به دلیل تولید سوئیچ‌های مداری توسط شرکت‌های خارجی نظیر شرکت اریکسون و مشکلاتی نظری تحریم‌ها، حذف این سوئیچ‌ها قدمی بزرگ به سمت خودکفایی صنعت مخابرات به شمار می‌رود. از آنجایی که سوئیچ‌های مداری، به صورت تجاری عرضه می‌شوند و برخی شرکت‌های داخلی نیز قادر به تولید آن‌ها می‌باشند، وجود تحریم‌ها مانعی برای استفاده از آن‌ها به شمار نمی‌رود.

۲۰.۳ آسان کردن ارائه‌ی سرویس‌های نوظهور

در نسل دوم ارتباطات، ایجاد یک سرویس جدید مانند سرویس پیامک، دو تا سه سال زمان می‌برد. با ظهور نسل سوم ارتباطات و بهبود زیرساخت‌ها، این زمان به حدود ۶ ماه کاهش یافت. با ظهور IMS، می‌توان در مدت چند روز تا حداکثر چند هفته، اپلیکیشن طرّاحی شده را به شبکه اضافه کرد و سرویس موردنظر را در اختیار کاربران قرار داد.

^۱ پروژه‌ی clearwater ادعا می‌کند که در صورت استفاده از ابر آمازون به عنوان بستر پیاده‌سازی IMS توسط این پروژه، این هزینه به ازای هر کاربر در طول یک سال، برابر دو سنت (دو صدم دلار) است.

دلیل اول این موضوع، وجود رابطه‌های برنامه‌نویسی اپلیکشن^۲ در تمام نقاط معماری IMS است. لذا، نیاز نیست که برای هر اپلیکیشن جدیدی که قرار است به کار برد شود، روش‌های مستقرسازی^۳ خاص مورد استفاده قرار گیرد. مثلاً اپلیکیشن جدید می‌تواند به راحتی و با استفاده از ابطه‌های برنامه‌نویسی اپلیکشن، با المان CDF^۴ ارتباط برقرار کند و هزینه‌ی سرویسی را که به مشترک ارائه می‌کند، از حساب او کم کند[۲]. دلیل دوم، لایه‌ی enabler می‌باشد که به عنوان بخشی از لایه‌ی اپلیکیشن تعریف شده است و اطلاعات خاص شبکه را برای اپلیکیشن‌ها فراهم می‌کند. enabler، یکتابع خاص است که یک بار در شبکه اجرا می‌شود و اطلاعاتش بین چندین اپلیکیشن به اشتراک گذاشته می‌شود. این اشتراک گذاری، به دلیل افزایش کارایی، کاهش هزینه‌ها و همچنین ایجاد امکان به اشتراک گذاشتن اطلاعات اشتراکی بین اپلیکیشن‌های مختلف مورد استفاده قرار می‌گیرد. در شرایطی که سیستم‌های کنونی، از روش ستونی^۵ برای در اختیار قرار دادن اطلاعات شبکه به اپلیکیشن‌ها استفاده می‌کنند، نیاز به استفاده از سرورهای مجزا برای این کار را رفع می‌کند[۲].



شکل ۳.۳: استفاده از روش ستونی و روش enabler مشترک

به عنوان مثال، سرویس‌های PTT^۶ و پیامک^۷، هر دو نیاز دارند که از وضعیت حضور مشترکین در شبکه آگاه باشند. در سیستم‌های متداول امروزی، هر کدام از این اپلیکیشن سرورها، از سرور وضعیت^۸ مخصوص به خود استفاده می‌کنند و امکان به اشتراک گذاشتن وضعیت حضور مشترکین سرویس PTT با اپلیکیشن سرور پیامک وجود ندارد. اما هنگامی که از یک enabler مشترک استفاده شود، تنها با استفاده از یک سرور وضعیت

Application Programming Interface(API)^۹
deployment^{۱۰}

^۴ سروازه‌ی عارت Charging Date Function. این المان، برای ارائه خدمات شارژ و کنترل مالی حساب مشترکین به کار می‌رود.

^۵ به دلیل شباهت شکل ستون به سیلوی غلات، به روش ستونی، روش سیلو نیز می‌گویند.

^۶ Push To Talk: سرویسی است که امکان استفاده از تلفن همراه به عنوان بیسمی‌های مخابراتی را فراهم می‌کند. در واقع، ارتباط مشترکین این سرویس در هنگام تماس، به گونه‌ای است که در هر لحظه، فقط یک نفر می‌تواند صحبت کند.

Instant Messaging^{۱۱}
Presence Server^{۱۲}

می‌توان اطلاعات مربوط به وضعیت حضور مشترکین را هم‌زمان در اختیار اپلیکیشن سرور پیامک و اپلیکیشن سرور PTT و سایر اپلیکیشن سرورها قرار دارد(شکل ۳.۳). در صورتی که لازم باشد یک اپلیکیشن سرور جدید برای کاربردی نوظهور به شبکه اضافه شود و این اپلیکیشن سرور، نیاز به دانستن وضعیت حضور مشترکین داشته باشد، بدون نیاز به ساخت و راهاندازی یک سرور وضعیت جدید می‌توان از سرور وضعیت موجود در شبکه استفاده کرد. به این ترتیب، زمان و هزینه‌ی پیاده‌سازی اپلیکیشن سرورهای جدید و ارائه‌ی کاربردهای نوظهور، کاهش می‌ابد[۲].

۲.۳ ورود شرکت‌های شخص ثالث به بازار مخابرات

با توجه به مطالب بیان شده در بخش ۲.۱.۳، استفاده از IMS باعث کاهش زمان، پیچیدگی پیاده‌سازی و هزینه‌ی ایجاد اپلیکیشن سرور و ارائه‌ی سرویس‌های نوظهور می‌شود. این موضوع باعث هموار شدن راه سایر شرکت‌ها برای ورود به بازار مخابرات به عنوان شرکت‌های شخص ثالث می‌شود.

طرّاحی و فروش اپلیکیشن سرور

شرکت‌های شخص ثالث می‌توانند اپلیکیشن سرورهای جدید برای ارائه‌ی کاربردهای نوظهور طرّاحی کنند و این اپلیکیشن سرورها را به اپراتورهای تلفن همراه بفروشند. به دلیل آسان‌تر شدن پیاده‌سازی و راهاندازی اپلیکیشن سرورها در صورت استفاده از IMS، اپراتورهای تلفن همراه استقبال بیشتری از این موضوع خواهند کرد. همچنین، کاهش زمان بین طرّاحی اپلیکیشن سرور و ارائه‌ی سرویس جدید به کاربران، باعث می‌شود که طرح‌های جدید، زودتر به سودآوری برسند. این موضوع، هم برای شرکت‌های شخص ثالث و هم برای اپراتورهای تلفن همراه حائز اهمیّت است.

استفاده از زیرساخت اپراتورهای تلفن همراه

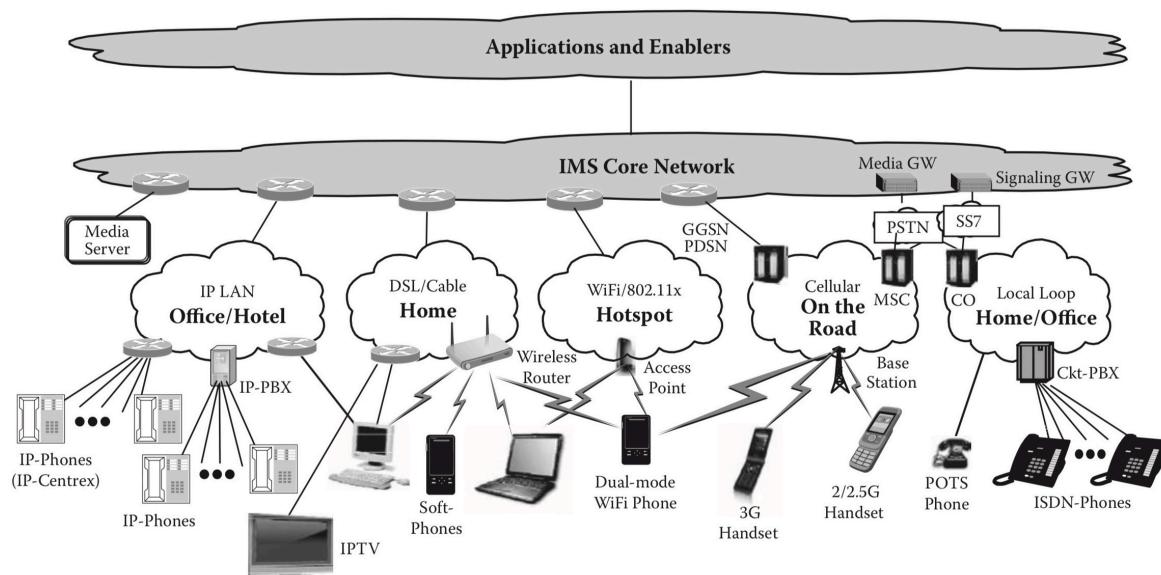
زیرساختی که اپراتورهای تلفن همراه برای ارائه‌ی سرویس خود ایجاد کرده‌اند، به‌طور کامل مورد استفاده قرار نمی‌گیرد و این قابلیت را دارد که سرویس‌های بیشتری را به کاربران بیشتری ارائه دهد. وجود درگاه OSA(بخش ۳.۲)، این امکان را فراهم می‌کند که شرکت‌های نوظهور بتوانند سرویس‌های خود را با استفاده از زیرساخت اپراتورها ارائه کنند.

۳.۰.۳ عدم وابستگی به شبکه‌ی دسترسی

استفاده از شبکه‌های محلی و اینترنت

از ویژگی‌های اصلی IMS، عدم وابستگی به شبکه‌ی دسترسی^۹ می‌باشد. این ویژگی باعث می‌شود که کاربر بتواند از طریق اتصال به سایر شبکه‌ها، از سرویس IMS بهره‌مند شود. مشترکین اپراتور، می‌توانند با اتصال تلفن همراه خود به شبکه‌ی اینترنت (از طریق وای‌فای)، از خدمات اپراتور خود و IMS بهره‌مند شوند. استفاده از وای‌فای به جای ارتباط با ایستگاه مخابراتی، مزایای زیادی دارد. این کار باعث کاهش مصرف باتری تلفن همراه و صرفه‌جویی در مصرف برق می‌شود.^{۱۰} همچنین به دلیل کاهش سیگنال‌های مخابراتی که کاربر به ایستگاه‌های مخابراتی می‌فرستد، ضررها ناشی از این سیگنال‌ها کاهش خواهد یافت.^{۱۱}

به دلیل عدم وابستگی IMS به شبکه‌ی دسترسی، دیگر الزامی نیست که مشترکین تلفن همراه برای برقراری تماس تلفنی، به ایستگاه‌های مخابراتی متصل شود. لذا در نقاطی که پوشش آتندهی مناسبی وجود ندارد، مشترکین می‌توانند با استفاده از شبکه‌ی اینترنت^{۱۲} خود از خدمات اپراتور تلفن همراه خود بهره‌مند شوند. همچنین، این امکان وجود دارد که اپراتورهای تلفن همراه به جای ساخت ایستگاه مخابراتی در نقاط دور دست (که بسیار هزینه‌بر است)، یک شبکه‌ی محلی مبتنی بر پروتکل اینترنت در آن مناطق ایجاد کنند. با اتصال این شبکه‌ی محلی به شبکه‌ی داخلی اپراتور، کاربران قادر به استفاده از خدمات IMS و اپراتور موردنظر می‌شوند.



شکل ۴.۳: پشتیبانی از شبکه‌های دسترسی مختلف توسط IMS

Access Network Independent^۹.

^{۱۰} ارتباط وای‌فای، توان کمتری نسبت به ارتباطات مخابرات سلولی مصرف می‌کند.

^{۱۱} به دلیل پایین‌تر بودن دامنه‌ی فرکانسی وای‌فای نسبت به امواج مورد استفاده در ارتباط با ایستگاه مخابراتی، ضررها جسمانی کمتری به بدن انسان می‌رسد.

^{۱۲} اینترنتی که توسط فiber نوری و یا کابل تلفن ثابت در اختیار آن‌ها قرار می‌گیرد و عموماً به صورت سرویس ADSL است.

اتصال با دستگاه‌های مختلف

در IMS، این امکان وجود دارد که با دستگاه‌های مختلف به شبکه‌ی اینترنت متصل شد و سپس از خدمات اپراتور مربوطه و IMS مانند تماس صوتی، بهره‌مند شد. از آنجایی‌که IMS به شبکه‌ی دسترسی وابستگی ندارد، کاربر می‌تواند با دستگاه‌هایی نظیر کامپیوتر شخصی خود به اینترنت متصل شده و از طریق آن، تماس صوتی برقرار نماید. البته لازم است که تدبیری برای ایجاد این قابلیت اندیشیده شود. استفاده از یک رمزبور اختصاصی برای هر کاربر، لازمه‌ی این کار است. همچنین، این امکان وجود دارد که کاربر با یک شماره تلفن، هم‌زمان از طریق چند دستگاه به شبکه‌ی اپراتور موردنظر خود متصل شود^[۳].

جایه‌جایی

سیستم IMS، این قابلیت را دارد که سطح بالاتری از جایه‌جایی^[۱۳] را به کاربران خود ارائه دهد. در سیستم‌های گذشته، کاربران می‌توانستند به مکان‌های جغرافیایی مختلف بروند و به ایستگاه‌های مخابراتی مختلف اپراتور خود متصل شوند، بدون این که ارتباط آن‌ها قطع شود. سطح بالاتر جایه‌جایی که رومینگ^[۱۴] نام دارد، اتصال دستگاه کاربر به ایستگاه‌های مخابراتی سایر اپراتورها و دریافت سرویس اپراتور خود، از طریق این ایستگاه‌ها می‌باشد. سطح بالاتری از جایه‌جایی که توسط IMS ارائه می‌شود، اتصال کاربر به شبکه‌های دسترسی متفاوت و دریافت سرویس از اپراتور خود است. با وجود تغییر شبکه‌ی دسترسی کاربر، ارتباط کاربر قطع نخواهد شد و کاربر قادر است به راحتی و بدون قطع ارتباط، بین شبکه‌های دسترسی مختلف جایه‌جا شود^[۲].

۴.۳ کاربردهای نوظهور

کاربردهای نوظهور بسیاری توسط سیستم IMS فراهم می‌شوند که در اینجا، چند مورد از مهم‌ترین آن‌ها ذکر شده‌اند. برای مطالعه‌ی بیشتر، به [۲] و [۳] مراجعه شود.

تماس ویدیوئی و کنفرانس صوتی و ویدوئی

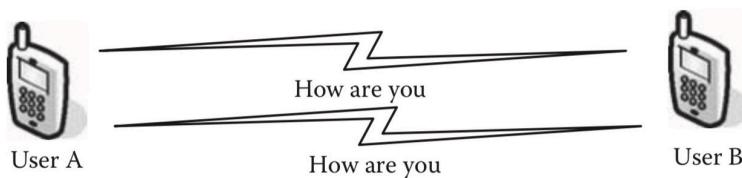
از آنجایی‌که IMS مبتنی بر پروتکل SIP است، می‌توان به وسیله‌ی آن تماس ویدیوئی برقرار کرد. همچنین، امکان ایجاد یک جلسه بین چند کاربر و برقراری کنفرانس صوتی و ویدوئی نیز وجود دارد. از آنجایی‌که در بسیاری از مواقع، کاربران IMS به شبکه‌ی داخلی اپراتور تلفن همراه خود متصل هستند، هزینه‌ی تماس ویدیوئی و یا برقراری کنفرانس (صوتی یا ویدیوئی) نسبت به سیستم‌های کنونی ارزان‌تر خواهد بود^[۲].

IP messaging

از دیگر قابلیت‌های IMS، فراهم کردن سرویس غنی ارتباطات است.^{۱۵} سیستم IMS این قابلیت را دارد که سرویسی مشابه اپلیکیشن‌های پیام‌رسان متدالو مانند تلگرام، واتس‌اپ، سروش و ... را به کاربران خود ارائه دهد. در این سرویس علاوه بر ارسال و دریافت پیام، امکان مشاهده و وضعیت حضور سایر کاربران، بهاشترانک گذاشتن فایل، چت گروهی و ... نیز وجود دارد. همچنین در سیستم پیامک، ذخیره‌سازی تاریخچه‌ی پیام‌ها بر روی دستگاه کاربر صورت می‌گیرد.^{۱۶} در سرویس IP messaging مورد استفاده در IMS، این تاریخچه بر روی پایگاه‌های داده‌ی جداگانه‌ای و برای مدتی طولانی ذخیره می‌شود. لذا، کاربر به راحتی می‌تواند به این تاریخچه دسترسی داشته باشد.^{۱۷} [۳].

PoC

سیستم‌های مخابراتی متدالو، یک ارتباط کاملاً دوطرفه^{۱۸} را ایجاد می‌کنند و به کاربران اجازه می‌دهند که به‌طور همزمان با یکدیگر مکالمه کنند(شکل ۵.۳). در سیستم‌های نیمه دوطرفه^{۱۹}، زمانی که یکی از کاربران در حال صحبت کردن است(دیتا ارسال می‌کند)، کاربر دیگر فقط می‌تواند گوش بدهد. این سیستم‌ها که با نام Push To Talk شناخته می‌شوند، توسط نیروهای پلیس، نگهبان‌ها و کارکنان پروژه‌های ساختمانی بزرگ و یا معادن مورد استفاده قرار می‌گیرند. مزیت اصلی این سیستم‌ها، قابلیت همه‌پخشی^{۲۰} است. همچنین در این سیستم‌ها، نیاز به شماره‌گیری نیست و با فشار دادن دکمه‌ی Push مخاطب صدای شما را می‌شنود.



شکل ۵.۳: ارتباط کاملاً دوطرفه‌ی بی‌سیم

سیستم IMS، قابلیت پشتیبانی از PTT را دارد. سرویس PTT در بستر شبکه‌ی سلوی، PoC^{۲۱} نام دارد. با استفاده از PoC، کاربران تلفن همراه می‌توانند از سرویس PTT بهره‌مند شوند. همچنین، سیستم PoC سرویس‌های جدیدی نیز ارائه می‌کند. از قابلیت‌های اصلی PoC می‌توان به موارد زیر اشاره کرد [۴]:

^{۱۵} Rich Communication Service(RCS)

^{۱۶} البته این تاریخچه، به مدت حدّاً کثیر چند ماه در پایگاه داده‌ی اپراتور مربوطه نگه‌داری می‌شود و برای دسترسی به آن، نیاز به اجازه‌ی قصاصی می‌باشد.

^{۱۷} مثلاً در صورت ذردیده شدن تلفن همراه، کاربر با احراز هویت در پایگاه داده‌ی موردنظر، می‌تواند تاریخچه‌ی پیام‌ها را بر روی تلفن همراه جدید بارگزاری کند.

^{۱۸} Full-duplex

^{۱۹} Half-duplex

^{۲۰} Broadcast

^{۲۱} PTT over Cellular

- ارسال محتوای چند رسانه‌ای (Push to Video) علاوه بر ارسال صوت
- ارتباط با سرویس‌های کنونی PTT
- ایجاد جلسه‌ی PoC بین ماشین و کاربر
- ایجاد ذخیره کردن مديای مبادله شده (PoC Box)

فصل ۴

پروژه clearwater

۱۰.۴ مقدمه

پروژه clearwater [۵]، یک پیاده‌سازی متن‌باز از معماری IMS است که توسط شرکت Metaswitch [۶] پشتیبانی می‌شود. پروژه clearwater از ابتدا، به منظور پیاده‌سازی در محیط‌های مجازی‌سازی و ابری طراحی شده است. این پروژه، بر اساس الگوهای طراحی خاصی پیاده‌سازی شده است که این الگوها معمولاً در تولید و پیاده‌سازی نرم‌افزارهای مقیاس‌پذیر استفاده می‌شوند؛ لذا با استفاده از clearwater می‌توان به میلیون‌ها کاربر خدمات تماس صوتی و تصویری و پیامک و ... ارائه کرد. معماری clearwater، تفاوت‌های اندکی با معماری متعارف IMS دارد. این تفاوت‌های جزئی، بیشتر مربوط به استفاده از روشی متفاوت در پیاده‌سازی رابط‌ها می‌باشد؛ اما می‌توان گفت که clearwater تمام المان‌ها و رابط‌های استاندارد مورد نیاز برای هسته‌ی IMS را پیاده‌سازی کرده است.

کنترل تماس‌های برنامه‌ی SIP برای ارتباطات صوتی و ویدیوئی توسط clearwater فراهم شده است. همچنین، clearwater کنترل‌های لازم برای ارسال پیامک برنامه‌ی SIP و اپلیکیشن‌های پیام‌رسان برنامه‌ی SIP را نیز فراهم می‌کند. پروژه clearwater می‌تواند به عنوان یک روش خوداتکا برای ارائه‌ی سرویس VOIP استفاده شود. به دلیل وجود پایگاه داده‌ی خوداتکا و همچنین المان‌های تعبیه‌شده در معماری clearwater که به راحتی می‌توان آنها را مقیاس کرد، این پیاده‌سازی می‌تواند پاسخگوی حجم عظیمی از کاربران VOIP باشد. همچنین می‌توان از clearwater به عنوان هسته‌ی IMS استفاده کرد و آن را با المان‌هایی نظیر اپلیکیشن سرورهای تلفن^۱ و یا HSS ترکیب کرد. زمانی که clearwater به عنوان هسته‌ی IMS مورد استفاده قرار می‌گیرد، تمام کارهایی را که انتظار می‌رود هسته‌ی IMS به عمل برساند را انجام می‌دهد.

نقشه‌ی ارتباط clearwater با شبکه سلوالی، مطابق همان چیزی است که در فصل‌های قبل در مورد ارتباط

^۱Telephone Application Server(TAS)

IMS و شبکه‌ی سلولی بیان شد. همچنین، clearwater شامل یک درگاه webRTC^۱ است که امکان ارتباط بین کاربر webRTC و کاربر IMS را فراهم می‌کند. پروژه‌ی clearwater، رابط لازم برای ارتباط با سایر شبکه‌های IMS و شبکه‌های VOIP و همچنین ارتباط با شبکه‌ی تلفن ثابت را نیز فراهم می‌کند. پیاده‌سازی‌های دیگری نظیر Open IMS Core [۷] برای IMS وجود دارد اماً با توجه به برتری‌های پروژه‌ی clearwater نسبت به آن‌ها، در این پروژه از پیاده‌سازی clearwater شده است. از مزایای اصلی clearwater نسبت به سایر پیاده‌سازی‌ها، می‌توان به موارد زیر اشاره کرد:

- متن‌باز است
- قابلیت‌های بیشتری دارد
- کامل‌تر بودن المان‌های پیاده‌سازی شده با توجه به استاندارد IMS
- carrier grade^۲ است
- منبع کلیه‌ی مطالب این فصل، مبتنی بر [۵] است.

۲۰.۴ معماری clearwater

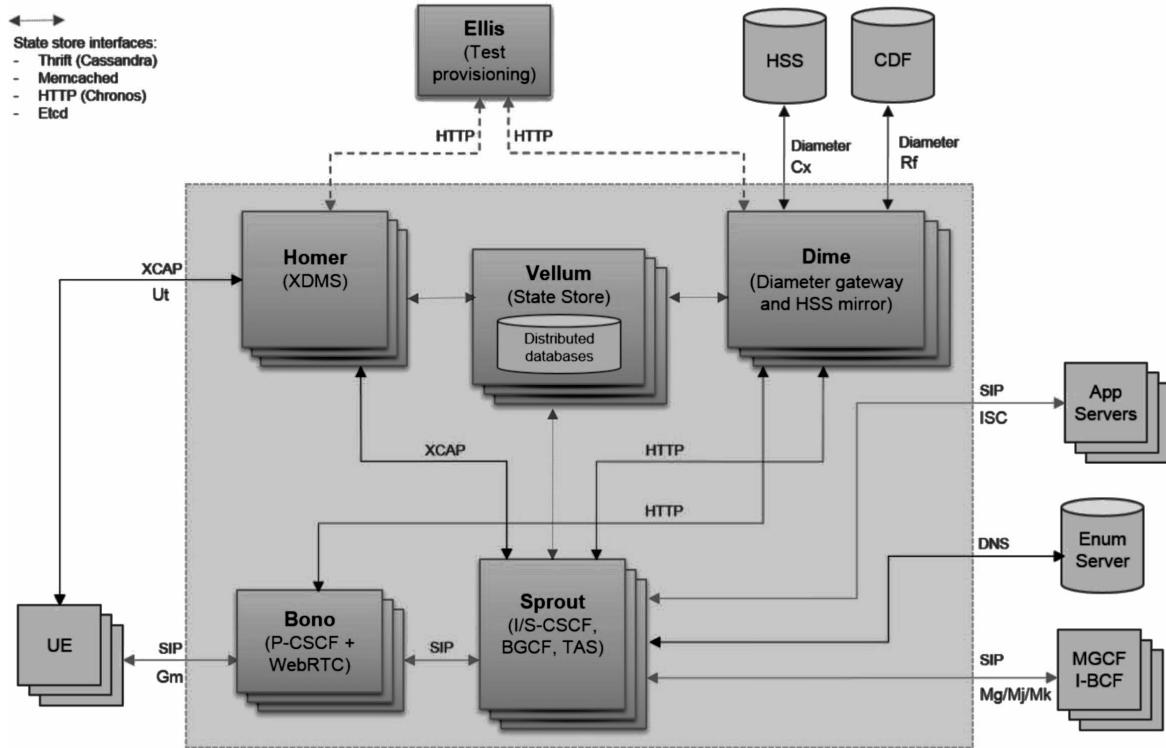
معماری clearwater، المان‌های مخصوص به خود را دارد که هر یک از این المان‌ها، در واقع پیاده‌سازی یک یا چند المان معماری IMS می‌باشد. در اینجا، بهجای این‌که توضیح کامل هر یک از المان‌های clearwater آورده شود، فقط گفته می‌شود که این المان، شامل چه المان‌هایی از IMS است و کار اصلی المان موردنظر چیست. شکل ۱۰.۴ المان‌های معماری clearwater، ارتباط بین آن‌ها و همچنین نوع رابط بین آن‌ها را نمایش می‌دهد.

Bono

این المان، نقش P-CSCF را دارد و نقطه‌ی ورود به IMS می‌باشد. فراهم کردن مکانیزم عبور از NAT^۳ و دیوار‌آتش^۴ توسعه این المان انجام می‌شود. کاربران می‌توانند با استفاده از پیام‌های SIP به Bono متصل شوند؛ این پیام‌های SIP می‌توانند از پروتکل‌های لایه‌ی انتقال UDP و TCP استفاده کنند. همچنین، المان Bono شامل رابط webRTC برای کاربران نیز می‌باشد و سیگنالینگ‌های ایجاد تماس را با استفاده از SIP بر روی webRTC^۵ یک پروژه‌ی متن‌باز است. این پروژه، از طریق یک رابط برنامه‌نویسی اپلیکیشن، قابلیت ارتباط بلادرنگ را برای مرورگرها و اپلیکیشن‌های تلفن همراه فراهم می‌کند.

^۱ در مخابرات، به سیستم، نرم‌افزار و یا سخت‌افزاری که قابل اعتماد است، به صورت کامل تست شده و قابلیت‌هایش اثبات شده است، carrier grade می‌گویند.

^۲ سروازه‌ی عارت Network Address Translation Firewall^۵



شکل ۱.۴ : معماری clearwater

سوکت وب انجام می‌دهد. clearwater می‌تواند با استفاده از P-CSCF‌های شخص ثالث و یا با استفاده از یک SBC^۶ که P-CSCF را پیاده‌سازی کرده است، راهاندازی شود؛ در این صورت دیگر نیازی به المان Bono نیست.

(SIP مسیریاب Sprout)

المان Sprout، به طور همزمان نقش I-CSCF، S-CSCF، BGCF و TAS را دارد. این المان، شامل اپلیکیشن سرور MMTEL^۷ نیز می‌باشد. تراکنش‌های SIP، به صورت متوازن در خوشی Sprout پخش شده‌اند. بنابراین هیچ وابستگی طولانی مدت بین کاربر و یک نود خاص Sprout وجود ندارد؛ در نتیجه، Sprout هیچ دینای طولانی مدت را ذخیره نمی‌کند، اما از طریق رابط سرویس وب، به Homer و Homestead متصل شده و اطلاعات لازم را از HSS به دست می‌آورد. نود Sprout، از طریق رابط برنامه‌نویسی اپلیکیشن، برای ذخیره دیتای ثبت‌نام مشترکین و همچنین راهاندازی زمان‌سنج، به Vellum متصل می‌شود. بسیاری از کارکردهای I-CSCF و S-CSCF در Sprout پیاده‌سازی شده‌اند و مابقی کارکردها توسط المان Dime تأمین می‌شوند.

^۶ سروازه‌ی عبارت Session Border Controller و به معنای کنترل کننده‌ی مرز جلسه

^۷ کوتاه‌شده‌ی عبارت Multimedia Telephony. اپلیکیشن سرور MMTEL، به صورت پیش‌فرض در معماری clearwater وجود دارد و با استفاده از آن می‌توان تماس صویی یا تصویری و همچنین سرویس پیامک را برقرار کرد.

Cluster^۸

(Diameter Dime

نود Dime، دو المان clearwater به نام‌های Homestead و Ralf را اجرا می‌کند.

(HSS (ذخیره‌گاه Homestead •

این المان، رابط سرویس وب را برای Sprout فراهم می‌کند تا Sprout بتواند اطلاعات مربوط به احراز هویت و اطلاعات پروفایل کاربران را به دست آورد.

Ralf •

این المان، یک رابط برنامه‌نویسی اپلیکشن HTTP را برای Bono و Sprout فراهم می‌کند. المان‌های Sprout و Bono، با استفاده از این رابط گزارش رخدادهای صورت‌حسابی^۹ کاربران را به المان CDF ارسال می‌کنند. المان Ralf از المان Vellum برای نگهداری اطلاعات طولانی مدت وضعیت جلسات استفاده می‌کند.

(محل ذخیره‌ی وضعیت) Vellum

المان Vellum، تمام وضعیت‌های طولانی مدت را نگه‌دارد. این کار با اجرای خوش‌های حافظه‌ی توزیع شده به نام Cassandra انجام می‌شود. Cassandra، یک پایگاه داده‌ی متن‌باز و توزیع شده است که برای مدیریت حجم زیادی از داده‌ها در سرورهای مختلف استفاده می‌شود. توزیع شده بودن این پایگاه داده، در دسترس بودن و قابلیت اطمینان را فراهم می‌کند.^[۸]

Homer

این المان، یک سرور مدیریت دیتای XML است که مستندات مربوط به تنظیمات سرویس MMTEL را برای هر کاربر ذخیره می‌کند. همانند سایر المان‌ها، Homer از نود Vellum برای ذخیره‌ی تمام دیتاهای طولانی مدت استفاده می‌کند.

Ellis

نود Ellis، کارهایی مانند نام‌نویسی^{۱۰}، مدیریت رمزعبور، کنترل خط و همچنین کنترل تنظیمات سرویس MMTEL را آسان کرده و به صورت خودکار انجام می‌دهد. این نود، به عنوان یک المان اصلی پروژه clearwater شناخته نمی‌شود؛ بلکه برای راحت کردن استفاده از این پروژه طراحی شده است. با توجه به پیاده‌سازی Ellis با استفاده از MySQL، مقایس کردن آن کاری دشوار و پیچیده است.

Billable^۹
Sign-up^{۱۰}

۳.۴ قابلیت‌های clearwater

پروژه‌ی clearwater، بسیاری از کاربردهای استاندارد IMS را پشتیبانی می‌کند. از پیاده‌سازی clearwater می‌توان به عنوان یک سیستم خوداتکا برای ارائه‌ی سرویس VOIP در بستر شبکه‌ی محلی و یا اینترنت استفاده کرد. این سیستم، قابلیت اتصال به سایر سیستم‌های ارائه‌دهنده‌ی سرویس تماس صوتی نظیر PSTN، سایر شبکه‌های VOIP و webRTC را دارد. همچنین، می‌توان با اتصال clearwater به شبکه‌ی سلولی، سرویس تماس صوتی را از طریق IP و با کیفیت تضمین شده را به مشترکین اپراتورها ارائه کرد.

از دیگر قابلیت‌های این سیستم، امکان اضافه کردن اپلیکشن سرور است. توسعه‌دهنگان نرم‌افزار، می‌توانند اپلیکیشن سرورهای جدید برای ارائه‌ی کاربردهای نوظهور طراحی کنند و به راحتی این اپلیکیشن سرورها را به clearwater اضافه کنند. لذا، ارائه‌ی سرویس‌های نوظهور بر مبنای IMS، از قابلیت‌های clearwater بهشمار می‌رود. از دیگر قابلیت‌های این پروژه، عدم وابستگی به ناحیه‌ی دسترسی است (بخش ۳).

مقیاس‌پذیری این سیستم، چه در بستر سخت‌افزاری و چه در بستر ابر، از قابلیت‌های مهم این سیستم بهشمار می‌رود. clearwater به‌گونه‌ای طراحی شده است که مقیاس کردن آن آسان است. این سیستم، قابلیت ایجاد خوش‌های از نودهای یکسان را دارد. با ایجاد خوش‌های از یک نود، قابلیت اطمینان سیستم و همچنین سرعت و کارایی سیستم افزایش می‌یابد و سیستم قادر به ارائه‌ی سرویس به کاربران بیشتری می‌شود. می‌توان با ایجاد یک افزونگی جغرافیایی از کل معماری سیستم، پس از به وجود آمدن نقص در سیستم اصلی، همچنان سرویس خود را ارائه کنیم. با ایجاد افزونگی جغرافیایی، در صورت از کار افتادن سیستم اصلی می‌توان با استفاده از روش‌های ارائه‌شده، پیکربندی و داده‌های موجود در آن را بازیابی کرد.

فصل ۵

روش‌های پیاده‌سازی

۱۰.۵ معرفی روش‌های پیاده‌سازی

در این فصل، انواع روش‌های ممکن برای پیاده‌سازی IMS توسط پروژه clearwater، به‌طور خلاصه بیان شده‌اند. مطالب این فصل بر اساس [۵] نوشته شده‌اند. انواع روش‌های پیاده‌سازی عبارتند از:

- پیاده‌سازی یکپارچه و در مقیاس کوچک

۱. نصب بر روی ابر محاسباتی منعطف آمازون (Amazon EC2)

۲. نصب خودکار بر روی سکوهای مجازی‌سازی

۳. نصب دستی بر روی سکوهای مجازی‌سازی

- پیاده‌سازی توزیع شده و در مقیاس بزرگ

۱. نصب خودکار بر روی Amazon EC2

۲. نصب دستی سرورها بر روی ماشین‌های مجرزا

- نصب با استفاده از کد منبع

۲۰.۵ پیاده‌سازی یکپارچه در مقیاس کوچک

۱۰۲.۵ معرفی

با وجود این که claerwater طوری طراحی شده است که قابلیت مقیاس‌پذیری در ابعاد وسیع را دارد، اما این امکان وجود دارد که بتوان تمام اجزای آن را روی یک رایانه‌ی شخصی نصب کرد. این روش نصب، نسبت

به سایر روش‌ها ساده‌تر است اما عملکرد محدودی نسبت به آن‌ها دارد. با وجود عملکرد محدود، پیاده‌سازی clearwater در مقیاس کوچک باعث آشنایی با معماری، المان‌ها و پیکربندی clearwater می‌شود. این کار باعث می‌شود که بتوانید عملکرد بهتر و آگاهانه‌تری در پیاده‌سازی مقیاس بزرگ با استفاده از روش‌های توزیع شده داشته باشید.

۲۰۲۰۵ انواع روش‌های یکپارچه

در تمام روش‌های یکپارچه، از All-In-One Image استفاده می‌شود. All-In-One Image شامل موارد زیر است:

- سیستم عامل اویونتو ۱۴.۰۴، پیکربندی شده برای استفاده از DHCP
- المان‌های Bono, Sprout, Homestead, Homer, Ellis
- اسکریپت‌های پیکربندی خودکار clearwater

در هنگام راهاندازی، image از طریق DHCP پیکربندی IP خود را به دست می‌آورد. سپس، اسکریپت پیکربندی خودکار، المان‌های BONO, Sprout, Homestead, Homer و Ellis را مطابق با IP به دست آمده، پیکربندی می‌کند. این image طوری طراحی شده است که روی یک ماشین مجازی با پردازنده‌ی تک‌هسته‌ای، ۲ گیگابایت RAM، و ۸ گیگابایت حافظه‌ی دیسک سخت قابل اجرا است. در صورت استفاده از ابر محاسباتی منعطف آمازون، مدل t2.small، این حدّاقل موارد مورد نیاز را فراهم می‌کند.

نصب بر روی ابر محاسباتی منعطف آمازون

می‌توان All-In-One Image را بر روی ابر محاسباتی منعطف آمازون نصب کرد. برای نصب ابتدا باید سرویس مورد نظر خود را که از لحاظ امکانات سخت‌افزاری، حدّاقل کارایی موردنیاز برای نصب All-In-One Image دارد (مانند t2.small) را خریداری کنید. نحوه‌ی نصب بر روی ابر محاسباتی منعطف آمازون در [۹] بیان شده است.

نصب خودکار بر روی سکوی مجازی‌سازی

سکوی مجازی‌سازی مورد استفاده در این روش، باید قابلیت پشتیبانی از OVF^۱ را داشته باشد. تمامی سکوی‌های مجازی‌سازی که از این قابلیت پشتیبانی می‌کنند، می‌توانند برای نصب مورد استفاده قرار گیرند؛ تاکنون، سکوی‌های زیر تست شده‌اند و این تست، موفقیت‌آمیز بوده است.

^۱ سرویزه‌ی عبارت Open Virtualization Format و به معنای فرمت مجازی‌سازی باز. OVF، یک استاندارد متن‌باز برای بسته‌بندی و توزیع نرم‌افزارهای ماشین مجازی است.

- VMware Player
- VirtualBox
- VMware ESXi

از آنجایی که All-In-One Image برای به دست آوردن پیکریندی IP خود از DHCP استفاده می‌کند، سکوی مجازی‌سازی موردنظر باید یا در نهاد خود DHCP را به کار برد و یا به یک شبکه‌ی حاوی سرور DHCP متصل باشد. مراحل نصب در این روش، در بخش ۶ توضیح داده شده‌اند.

نصب بر روی سکوی مجازی‌سازی با استفاده از کد منبع

اگر سکوی مجازی‌سازی مورد استفاده، OVF را پشتیبانی نمی‌کند، این امکان وجود دارد که نود All-In-One را به صورت دستی build کرد. برای این کار، ابتدا باید سیستم عامل اووبونتو ۱۴.۰۴ نسخه‌ی سرور را بر روی ماشین مجازی نصب کرد. سپس باید فایل اسکریپت مورد نیاز برای انجام تغییرات لازم، از GitHub دریافت شود. برای دریافت این فایل به [۱۰] مراجعه شود.

با اجرای این اسکریپت، المان‌های clearwater و رابطه‌ای آن‌ها بر روی ماشین مجازی نصب خواهند شد و سپس، All-In-One Image ساخته خواهد شد. با راهاندازی مجدد ماشین مجازی، All-In-One Image در پیاده‌سازی یکپارچه است. شده و مابقی کارها مانند برقراری تماس تلفنی، مشابه روش استفاده از فایل OVF در پیاده‌سازی یکپارچه است. همچنین، برای اطلاعات بیشتر در مورد نحوه اجرای اسکریپت و انجام تغییرات لازم به [۱۱] مراجعه شود.

۳.۲.۵ قابلیت‌ها و محدودیت‌ها

از آنجایی که All-In-One Image شامل تمام اجزای معما ری clearwater می‌باشد، می‌توان گفت که تقریباً تمام قابلیت‌های اصلی یک نصب توزیع شده و در مقیاس بزرگ (مانند برقراری تماس تلفنی و تنظیمات مربوط به شبکه و ...) را دارد. محدودیت‌های اصلی این روش، به شرح زیر می‌باشند.

عملکرد و کارایی ضعیف

از آنجایی که تمام نرم‌افزارها و اجزاء در روش پیاده‌سازی یکپارچه، روی یک ماشین مجازی نصب و اجرا می‌شوند، عملکرد و کارایی سیستم در این روش نسبت به روش توزیع شده ضعیف‌تر است. از این رو، با استفاده از این روش پیاده‌سازی تنها قادر به ارائه سرویس به تعداد کاربران محدودی خواهیم بود.

عدم وجود قابلیت مقیاس‌پذیری

اصلی‌ترین محدودیت این روش، عدم امکان مقیاس‌کردن است. در واقع، هیچ راه و گزینه‌ای وجود ندارد که بتوانیم بهوسیله‌ی آن، این نوع پیاده‌سازی را مقیاس کنیم و ماشین‌های مجازی بیشتری اضافه کرده تا ظرفیت سیستم را ارتقاء دهیم.

عدم امکان ایجاد افزونگی

سروریس سنتی ارتباطات، برای دستیابی به قابلیت اطمینان از روش تکرار و انعکاس داده‌ها استفاده می‌کند که غالباً هم بهوسیله‌ی طراحی یک‌به‌یک پیاده‌سازی شده‌اند. این روش پرهزینه و پیچیده را نمی‌توان بهخوبی در محیط ابری یا مجازی‌سازی شده به‌کار برد. در clearwater، این کار از طریق ایجاد یک خوشه از یک جزء و افزونگی‌هایش، و همچنین از طریق ایجاد افزونگی جغرافیایی صورت می‌گیرد. امکان ایجاد افزونگی در روش پیاده‌سازی در مقیاس کوچک وجود ندارد.

از آنجایی‌که در این روش، کل اجزاء روی یک ماشین مجازی نصب و اجرا می‌شوند، اگر این ماشین مجازی از کار بیافتد و یا بهدرستی کار نکند، کل سیستم از کار می‌افتد. در صورتی که در روش‌های توزیع‌شده، خرابی یک بخش، تنها باعث از کار افتادن بخشی از سرویس می‌شود؛ وجود افزونگی‌ها نیز باعث می‌شود که با از کار افتادن بخشی از سیستم بتوانیم بدون بهوجود آمدن اشکال و اختلال، سرویس خود را ارائه کنیم.

ناحیه سخت-رمزی

سخت-رمزی^۲ یک روش توسعه است که در آن ورود و پیکربندی داده‌ها بهصورت مستقیم در درون کد منبع جاسازی شده است. لذا، توسعه‌دهندگان دیگر نمی‌توانند بهراحتی آن را تغییر دهند. در روش پیاده‌سازی یکپارچه، از یک ناحیه سخت-رمزی به نام example.com استفاده می‌شود. بنابراین SIP URI کاربران مشابه example.com@6505551234@example.com خواهدبود(بخش ۶). بهصورت پیش‌فرض، SIP از این ناحیه برای مسیریابی استفاده می‌کند اما example.com میزبان شما را بهدست نمی‌آورد؛ بنابراین نیاز داریم که برای هر کاربر SIP، یک وکیل خارج باند^۳ پیکربندی کنیم. این پیکربندی، در اپلیکیشن SIP سمت کاربر انجام می‌شود.

Hard-coded^۲
Outbound proxy^۳

۳.۵ پیاده‌سازی در مقیاس بزرگ و پیاده‌سازی با کد منبع

پیاده‌سازی در مقیاس بزرگ، برای ارائه سرویس به تعداد زیادی کاربر استفاده می‌شود. این کار که بیشتر جنبه‌ی تجاری دارد، به سه روش قابل انجام است. در روش اول از ابرآمازون به عنوان بستر پیاده‌سازی استفاده می‌شود. در این روش، فقط با استفاده از یک دستور می‌توان پیاده‌سازی کامل clearwater در ابعاد(مقیاس) دلخواه را ایجاد کرد. لذا، راحت‌ترین روش برای پیاده‌سازی در مقیاس بزرگ است.

در روش دوم که به آن روش نصب دستی گفته می‌شود، باید بسته‌های دبیان^۴ را نصب کرده و هر ماشین را به طور جداگانه پیکربندی کرد. پیاده‌سازی این روش، نیازمند حداقل ۶ ماشین است. بر روی هر یک از این ماشین‌ها، باید نسخه‌ی سرور سیستم عامل اوبونتو ۱۴.۰۴ نصب شود. سپس، هر یک از نودهای معماری clearwater بر روی یک ماشین نصب می‌شود. پس از نصب نودها، تنظیمات مربوط به دیوار آتش و سرویس نام دامنه باید به صورت دستی انجام شوند.

در صورتی که سیستم عامل استفاده شده، مبتنی بر اوبونتو نیست و یا نیاز است که تغییراتی در کدهای منبع اعمال شود، می‌توان از روش سوم استفاده کرد. در این روش، باید با استفاده از کدهای منبع clearwater، هر یک از اجزای معماری آن را جداگانه build کرد. دستورالعمل استفاده از هر یک از اجزای معماری توسط clearwater منتشر شده است و با کمک این دستورالعمل‌ها، می‌توان این سیستم را از طریق کد منبع پیاده‌سازی کرد.

فصل ۶

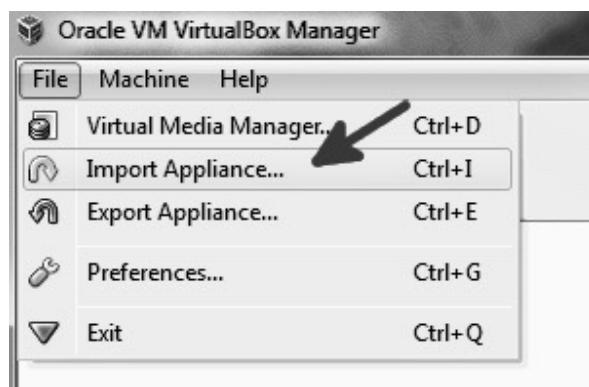
پیاده‌سازی در مقیاس کوچک بر روی سکوی مجازی‌سازی

۱.۶ نصب

برای پیاده‌سازی به این روش، نیاز داریم فایل OVF را که برای این نوع نصب بر روی سکوی مجازی‌سازی ارائه شده است، در اختیار داشته باشیم. این فایل را می‌توان از قسمت مربوط به نصب بر روی سکوی مجازی‌سازی وب‌سایت پروژه clearwater و یا از طریق لینک زیر دانلود کرد.

<http://vm-images.cw-ngv.com/cw-aio.ova>

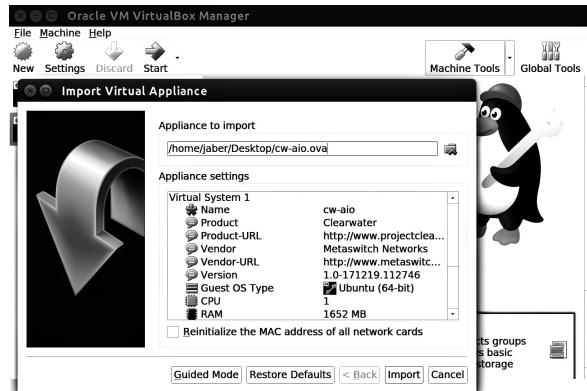
برای نصب بر روی Virtualbox، ابتدا این نرم‌افزار را بازکرده و سپس از منوی **File**، گزینه‌ی **Import**^۱ را انتخاب کنید(شکل ۱.۶). راه حل مشکلات احتمالی در نصب و پیکربندی VirtualBox در پیوست (آ) آورده شده است.



شکل ۱.۶ : Import Appliance

^۱Appliance در واقع یک image ماشین مجازی است که از قبل پیکربندی شده است.

در پنجره بازشده، نواری وجود دارد که با کلیک بر روی آن، می‌توان فایل موردنظر را انتخاب کرد. سپس بر روی دکمه **Next**، کلیک کنید. در صفحه جدید، می‌توانید تنظیمات مربوطه را مشاهده کنید. با کلیک بر روی گزینه **Import**، عملیات نصب به طور خودکار انجام می‌شود(شکل ۲.۶). در صورت وجود مشکل در ایجاد و اجرای ماشین مجازی، به پیوست (آ) مراجعه شود.

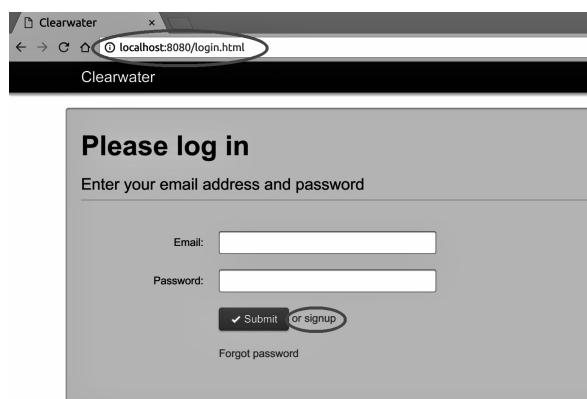


شکل ۲.۶ Import OVF file :

۲.۶ راهاندازی ماشین مجازی و ایجاد شماره تلفن

پس از اتمام نصب، یک ماشین مجازی به نام **cw-aio** به لیست ماشین‌های مجازی نصب شده بر روی **Virtualbox** شما اضافه می‌شود. با کلیک بر روی آن، ماشین مجازی **clearwater** شروع به کار می‌کند. پس از بالا آمدن صفحه ورود، از شما خواسته می‌شود که نام کاربری و رمز عبور را وارد کنید. نام کاربری 'ubuntu' و رمز عبور 'cw-aio' است.

پس از ورود، از طریق ماشین میزبان یک مرورگر را به دلخواه باز کرده و در قسمت URL آن، عبارت <http://localhost:8080> را وارد کنید. با انجام این کار، وارد صفحه‌ای مشابه شکل ۳.۶ می‌شوید.



شکل ۳.۶: صفحه‌ی ورود

در صورت عدم دسترسی به صفحه‌ی وب، باید در قسمت URL، عبارت <IP address>:8080 را وارد کنید و به جای عبارت <IP address>، آدرس IP ماشین میزبان را وارد کنید. لازم به ذکر است که پس از اجرای ماشین مجازی، باید اندکی صبر کنید تا سرور مربوطه شروع به کار کند و صفحه‌ی مذکور، در دسترس شما قرار گیرد.

با ورود به این صفحه می‌توان از طریق رابط کاربری وب، مشترکان پیاده‌سازی clearwater را مدیریت کرد. در صفحه‌ای که وارد آن شده‌اید، بر روی گزینه **signup** کلیک کنید (اگر قبلًا ثبت‌نام نکرده‌اید). در صفحه‌ی ثبت‌نام (شکل ۴.۶)، اطلاعات خواسته‌شده (نام، آدرس ایمیل و رمزعبور) را وارد کنید. اگر تیک گزینه **Demo Account** را بزنید، انقضای حساب کاربری ساخته‌شده پس از یک هفته تمام می‌شود. در قسمت **secret**، عبارت 'secret' را وارد کنید. در آخر، با کلیک بر روی دکمه **signup code**، حساب کاربری شما ساخته شده و وارد صفحه مربوط به مدیریت مشترکان می‌شود (شکل ۵.۶).

Sign up

Please enter your details

Your name: This will be your name in the address book

Email: This will be the email address you log in with

Password: Great, that password looks hard to guess.

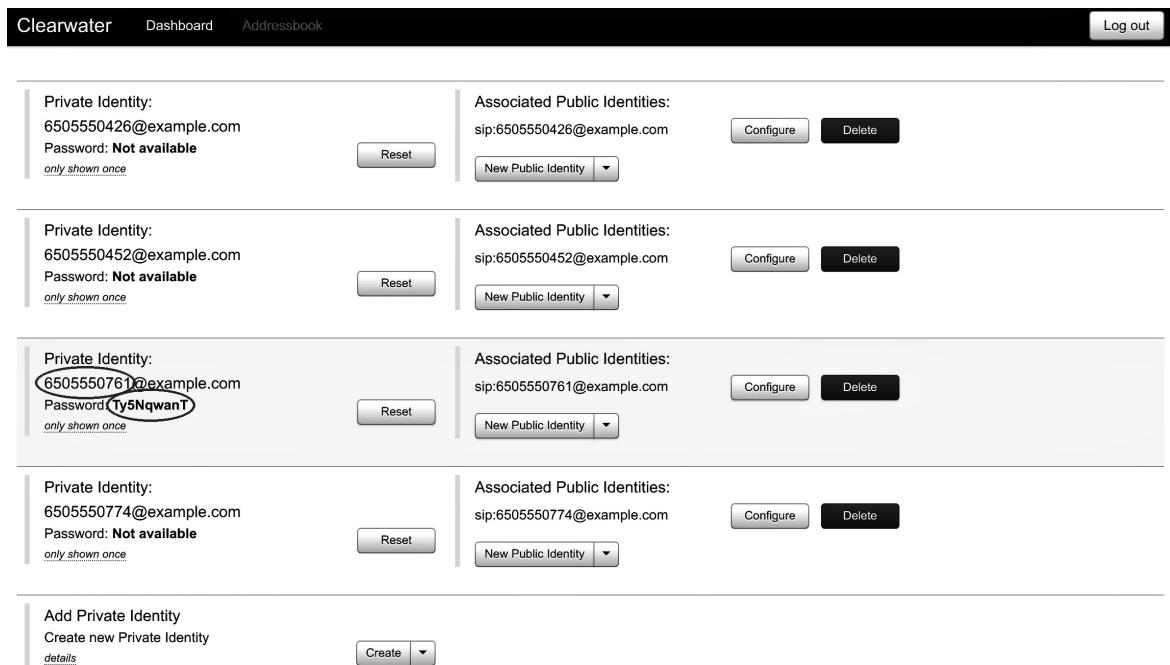
Confirm Password: Confirm the password you entered above.

Demo account? Demo accounts automatically expire after one week.

Signup code: You should have been sent a signup-code when invited to the beta

Sign up

شکل ۴.۶: ثبت‌نام



شکل ۵.۶: مدیریت کاربران

در صفحه‌ی مورد نظر، در قسمت **Create**، با کلیک بر روی گزینه **Create**، یک شماره تلفن جدید به فرمی مشابه آنچه در زیر می‌بینید، ایجاد می‌شود. قسمت 6505551234 هم به عنوان نام کاربری و هم به عنوان شماره تلفن کاربر استفاده می‌شود. رمزعبور نیز در هنگام ثبت‌نام SIP استفاده می‌شود. وقت شود که رمزعبور فقط یک بار نمایش داده می‌شود و در دفعات بعدی که وارد این صفحه می‌شوید، این رمز نشان داده نخواهد شد. با کلیک بر روی دکمه **Reset**، رمزعبور جدیدی برای شماره تلفن موردنظر ایجاد می‌شود.

6505551234@example.com

Password: 3At7dH5Md

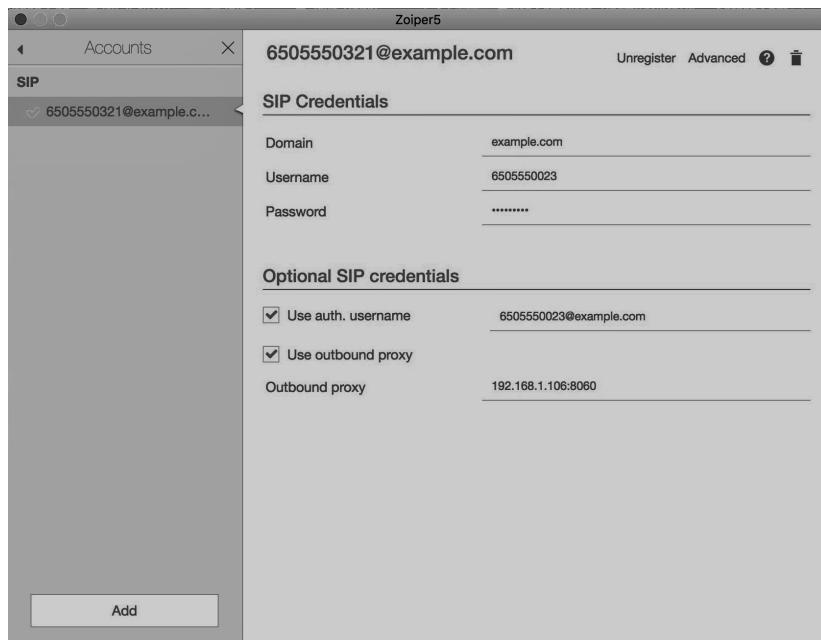
۳.۶ پیکربندی دستگاه کاربران

پس از ایجاد شماره تلفن برای کاربر، باید یک اپلیکیشن SIP روی دستگاه کاربر (تلفن همراه، رایانه شخصی و ...) نصب و سپس به طور مناسب، پیکربندی شود. به دلایل خاصی، اگر این نوع پیاده‌سازی روی سکوی مجازی سازی Virtualbox انجام شده باشد، حتماً باید یکی از طرفین مکالمه از اپلیکیشن zoiper استفاده کند. در این قسمت، برای سهولت کار فقط پیکربندی اپلیکیشن zoiper به همراه جزئیات توضیح داده شده است؛ بدیهی است که پیکربندی سایر اپلیکیشن‌ها نیز مشابه همین روش خواهد بود. همچنین، سیستم میزبان و دستگاه کاربران همگی باید به آدرس IP یکدیگر دسترسی داشته باشند؛ یعنی بتوانند از طریق پروتکل IP، با

یکدیگر ارتباط برقرار کنند.

پس از نصب برنامه zoiper ، وارد آن شده و از نوار بالای صفحه، گزینه **Config** را انتخاب کنید. سپس وارد قسمت Accounts شده و بر روی عبارت **Add account** کلیک کنید. در پاسخ به پرسش **Do you already have an account?** گزینه **Yes** را انتخاب کنید. در پنجره جدید، گزینه **configuration Manual** را انتخاب کنید. در صفحه جدیدی که باز می‌شود، از شما خواسته می‌شود که نوع حساب کاربری را انتخاب کنید. از آنجایی که clearwater از پروتکل SIP استفاده می‌کند، باید گزینه **SIP** را انتخاب کنید. با این کار، وارد بخش پیکربندی حساب کاربری می‌شوید.

در صفحه پیکربندی حساب کاربری، روی گزینه **Account name** کلیک کرده و یک نام دلخواه برای خود انتخاب کنید. سپس روی گزینه **Host** کلیک کرده و عبارت 'example.com' را وارد کنید. در قسمت **Username** و **Password**، نام کاربری و رمز عبوری را که در قسمت ایجاد کاربر دریافت کرده‌اید، وارد کنید. عبارت example.com را به نام کاربری خود اضافه کرده و در قسمت **Authenticatoin user** وارد کنید. (به عنوان مثال: 6505551234@example.com)



شکل ۶.۶: صفحه پیکربندی حساب کاربری در نسخه دسکتاپ اپلیکیشن zoiper

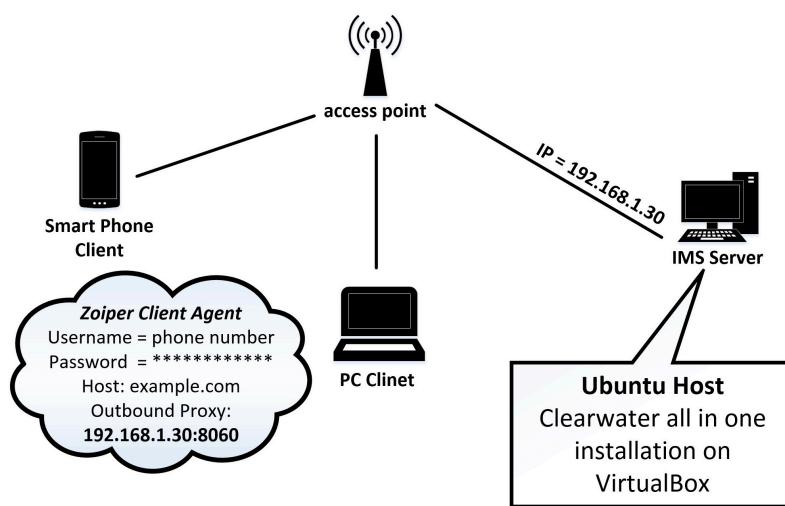
پیکربندی انجام شده تا این قسمت، مربوط به ورود اطلاعات کاربر است. اکنون باید تنظیمات مربوط به شبکه انجام شود. آدرس IP ماشین میزبان خود را به دست آورید. این آدرس را به همراه شماره پورت 8060، در قسمت **Outbound proxy** وارد کنید. به عنوان مثال، اگر آدرس IP ماشین میزبان، 192.168.1.30 است، شما باید عبارت 192.168.1.30:8060 را در قسمت مذکور وارد کنید. سپس گزینه **Network Setting** را انتخاب کنید. مطمئن شوید که **Transport type** از نوع TCP باشد. همچنین، **for signalling**

۶.۶ Use STUN باشد. قسمت Use RPORT را نیز غیر فعال کنید.

پس از انجام مراحل فوق، از قسمت مربوط به تنظیمات شبکه خارج شوید و سپس گزینه‌ی Save را انتخاب کنید. اکنون، مراحل فوق را برای یک کاربر دیگر (با شماره تلفن متفاوت) انجام دهید. از آنجایی‌که اپلیکیشن zoiper، قابل نصب بر روی سیستم عامل‌های مختلف است، می‌توانید یک نسخه‌ی مربوط به رایانه‌ی شخصی(شکل ۶.۶) خود را دریافت کرده و از طریق رایانه شخصی نیز تماس تلفنی برقرار کنید.

۴.۶ برقراری تماس

پس از ذخیره تنظیمات و پیکربندی‌ها در مرحله قبل، اپلیکیشن به‌طور خودکار عمل ثبت‌نام را انجام می‌دهد. پس از انجام مراحل فوق، هرگاه که کاربر به شبکه‌ای متصل شود که از طریق آن به IP ماشین میزبان دسترسی داشته باشد، اپلیکیشن zoiper به‌طور خودکار عملیات ثبت‌نام را انجام می‌دهد. پس از انجام ثبت‌نام، کاربر می‌تواند با شماره‌گیری سایر مشترکان این سیستم، با آن‌ها تماس تلفنی برقرار کند(شماره‌گیری باید در قسمت dialer اپلیکیشن zoiper انجام شود). این تماس تلفنی، از طریق هسته IMS و بر روی IP شکل می‌گیرد و کیفیت بالایی دارد. شکل ۷.۶، توپولوژی پیاده‌سازی شده برای راهاندازی IMS و برقراری تماس تلفنی بین دو کاربر تلفن همراه و کاربر رایانه‌ی شخصی را نشان می‌دهد. در این پروژه تماس تلفنی از طریق IMS بین دو کاربر تلفن همراه، بین دو کاربر رایانه‌ی شخصی و همچنین بین کاربر تلفن همراه و کاربر رایانه‌ی شخصی برقرار شد.



شکل ۷.۶: توپولوژی مورد استفاده برای برقراری تماس تلفنی از طریق IMS

۵.۶ آزمایش clearwater

روش‌های گوناگونی برای آزمایش پیاده‌سازی IMS، توسط clearwater معرفی شده‌اند. یکی از این روش‌ها، آزمایش زنده^۲ است. آزمایش زنده را می‌توان به راحتی و به صورت منظم روی پیاده‌سازی خود به اجرا درآورد تا بتوان از عملکرد صحیح کارکردهای سطح بالای سیستم^۳، اطمینان حاصل کرد.

نصب وابستگی‌ها

برای اجرای این آزمایش، نیاز است که نسخه ۱.۹.۳ زبان Ruby بر روی ماشین مجازی clearwater نصب شده باشد. برای انجام این فرایند، دستورات زیر را به ترتیب اجرا کنید. در صورت عدم اجرای موفقیت‌آمیز هر یک از این دستورات، به پیوست (ب) مراجعه شود.

1. sudo apt-get install build-essential git --yes
2. curl -L http://get.rvm.io | bash -s stable
3. source ~/.rvm/scripts/rvm
4. rvm autolibs enable
5. rvm install 1.9.3
6. rvm use 1.9.3

دانلود و نصب مجموعه برنامه‌های آزمایش clearwater

ابتدا باید کلید ssh برای برقراری ارتباط امن با github.com در ماشین مجازی clearwater وجود داشته باشد. برای بررسی وجود کلید ssh و ساخت این کلید در صورت موجود نبودن آن، به [۱۲] مراجعه شود. پس از ایجاد کلید ssh، دستورات زیر باید به ترتیب اجرا شوند. این دستورات، مجموعه برنامه‌های آزمایش clearwater را دانلود و بر روی سیستم نصب می‌کند. در صورت عدم اجرای موفقیت‌آمیز هر یک از دستورات زیر، به پیوست (ب) مراجعه شود.

7. git clone -b stable --recursive git@github.com:Metaswitch/clearwater-live-test.git
8. cd clearwater-live-test
9. bundle install

اجرای آزمایش روی پیاده‌سازی یکپارچه

برای اجرای آزمایش زندهی clearwater، دستور شماره 10 را در خط فرمان ماشین مجازی clearwater اجرا کنید. در این دستور، به جای عبارت `<code>` باید **signup code** سیستم قرار گیرد. این کد به صورت پیشفرض، عبارت 'secret' است. در صورتی که از پیادهسازی بر روی سکوی مجازی‌سازی به روش خودکار استفاده شده است، به جای عبارت `<aio-identity>`، باید آدرس IP ماشین مجازی قرار گیرد. این آدرس IP که توسط DHCP به ماشین مجازی اختصاص داده شده است، آدرس آدپتور شبکه‌ی^۴ ماشین مجازی است که روی حالت NAT تنظیم شده است. این آدرس را می‌توان با اجرای دستور **ifconfig** و یا دستور **-I hostname** روی ماشین مجازی، مشاهده نمود. خروجی اجرای تست، مانند شکل ۸.۶ خواهد بود.

10. rake test[example.com] SIGNUP_CODE=<code> PROXY=<aio-identity> ELLIS=<aio-identity>

```
[root@cw-alio ~] Your bundle is complete!
[Your bundle is complete!]
[Your bundle is complete!] To see where a bundled gem is installed.
[cw-alio]ubuntu@cw-alio:~/clearwater-live-test$ hostname -I
10.0.2.15
[cw-alio]ubuntu@cw-alio:~/clearwater-live-test$ rake test[example.com] SIGNUP_CODE=secret PROXY=10.0.2.15 ELLIS=10.0.2.15
Basic Call - Mainline (TCP) - (6505550451, 6505550922) Passed
Basic Call - SDP (TCP) - (6505550213, 6505550391) Passed
Basic Call - Tel URIs (TCP) - (6505550329, 6505550444) Passed
Basic Call - Unknown number (TCP) - (6505550599, 6505550365) Passed
Basic Call - Rejected by remote endpoint (TCP) - (6505550916, 6505550078) Passed
Basic Call - Router - Paper model (ID) - (6505550453, 6505550704) Passed
Basic Call - Pracks (TCP) - (6505550124, 6505550134) Passed
Basic Registration (TCP) - (6505550583) Passed
Emergency Registration (TCP) - Skipped (Not running with emergency registrations)
  - Call with EMERGENCY REG= to run test
Multiple Identities (TCP) - (6505550349, 6505550586) Passed
Call Barling - Outbound Routing (TCP) - (6505550867, 6505550214) Passed
Call Barling - Allow non-international call (TCP) - Skipped (No PSTN support)
  - Call with PSTN=true to run test
Call Barling - Reject international call (TCP) - Skipped (No PSTN support)
  - Call with PSTN=true to run test
Call Barling - Unbound Redirect (TCP) - (6505550445, 6505550805) Passed
Call Diversion AS - Unconditional (TCP) - Skipped (No Call Diversion AS hostname provided)
  - Call with CDIV_AS=<hostname>
Call Diversion AS - Busy (TCP) - Skipped (No Call Diversion AS hostname provided)
  - Call with CDIV_AS=<hostname>
Call Diversion AS - No answer (TCP) - Skipped (No Call Diversion AS hostname provided)
  - Call with CDIV_AS=<hostname>
Call Diversion - Not registered (TCP) - (6505550372, 6505550131, 6505550760) Passed
Call Diversion - Not reachable (not registered) (TCP) - (6505550928, 6505550446, 6505550034) Passed
Call Diversion - Not reachable (408) (TCP) - (6505550781, 6505550227, 6505550469) Passed
Call Diversion - Wait redirect (TCP) - (6505550345, 6505550301, 6505550800) Passed
Call Diversion - Not reachable (500) (TCP) - (6505550547, 6505550508, 6505550906) Passed
Call Diversion - Busy (TCP) - (6505550840, 6505550548, 6505550696) Passed
Call Diversion - Unconditional (TCP) - (6505550870, 6505550668, 6505550069) Passed
Call Diversion - No answer (TCP) - (6505550985, 6505550796, 6505550616) Passed
Call Diversion - Bad target URI (TCP) - (6505550037, 6505550597) Passed
Call Diversion - Audio-only call (TCP) - (6505550526, 65055505094, 6505550018, 6505550976) Passed
Call Diversion - Audio-video call (TCP) - (6505550562, 6505550001, 6505550311, 6505550649) Passed
Call Forking - Mainline (TCP) - (6505550806, 6505550079) Passed
Call Forking - Endpoint offline (TCP) - (6505550694, 6505550771, 6505550718) Passed
Call Forking - Endpoint online (TCP) - (6505550580, 6505550581, 6505550817) Passed
Call Waiting - Cancelled (TCP) - (6505550802, 6505550807, 6505550835) Passed
CANCEL - Mainline (TCP) - (6505550461, 6505550761) Passed
Filtering - Accept-Contact (TCP) - (65055505971, 6505550984) Passed
Accept-Contact with response (SIP_51195) (TCP) - (6505550628, 6505550805) Passed
Filtering - Accept-Contact no match (TCP) - (6505550188, 6505550819) Passed
Filtering - Accept-Contact negated match (TCP) - (6505550811, 6505550561) Passed
Filtering - RFC3841 example (TCP) - (6505550852, 6505550709) Passed
Filtering - Reject-Contact no match (TCP) - (6505550990, 6505550852) Passed
```

شکل ۸.۶: خروجی اجرای آزمایش زندگی clearwater

فصل ۷

مقایسه کیفیت سرویس در VOIP و IMS

۱۰.۷ نیاز به کیفیت سرویس

همانطور که در فصل ۱ توضیح داده شد، مزیت اصلی IMS نسبت به سیستم های متداول VOIP، تضمین کیفیت سرویس در مکالمات است. شبکه ای اینترنت و پروتکل های استفاده شده در این شبکه، هیچگونه تضمینی برای برقراری سطح خاصی از کیفیت سرویس را ارائه نمی کنند^۱. از طرفی، اپلیکیشن های مختلف، نیاز دارند که پروتکل های لایه ای انتقال، حداقل های مورد نیازشان را تأمین کند.

این نیازها که به سه بخش تقسیم می شوند، عبارتند از: نیاز به قابل اعتماد بودن و عدم از دست رفتن داده ها، نیاز به حداقل نرخ ارسال (دریافت) داده ها^۲ و حساسیت نسبت به زمان. اپلیکیشن هایی که برای تماس صوتی و یا ویدیوئی مورد استفاده قرار می گیرند، نسبت به از دست رفتن داده ها، حساسیت کمی دارند و می توانند با وجود از دست رفتن بخشی از داده ها، سرویس موردنظر را به درستی ارائه کنند. اما این اپلیکیشن ها، شدیداً وابسته به نرخ ارسال (دریافت) هستند و حساسیات بالایی نیز نسبت به زمان دارند؛ به طوری که در صورت کم بودن نرخ ارسال (دریافت) داده ها و زیاد بودن زمان رفت و برگشت^۳، قادر به ارائه هیچ سرویس نیستند. همچنین، با توجه به شرایط فوق، امکان دارد کیفیت سرویس ارائه شده پایین بیاید و باعث نارضایتی کاربر شود [۱۳].

۲۰.۷ مقایسه کیفیت سرویس در شبکه اینترنت

سیستم های مبتنی بر پروتکل SIP، به صورت مستقیم در تبادل مديا شرکت نمی کنند و فقط سیگنالینگ های لازم برای برقراری، کنترل و خاتمه ای ارتباط را انجام می دهند. در واقع می توان گفت که اپلیکیشن های مبتنی

^۱ شبکه های ATM می توانند با اختصاص لینک اختصاصی به هر ارتباط، کیفیت سرویس مورد نیاز را ارائه دهند

^۲ Throughput

^۳ Round Trip Time(RTT)

بر پروتکل SIP، وظیفه‌ی ایجاد و کنترل یک ارتباط نظری به نظری^۴ بین تماس‌گیرنده و مخاطب وی را بر عهده دارند[۲]. کیفیت سرویس ارائه شده نیز بیشتر مربوط به میدیا می‌باشد و سیگنال‌های مبادله شده، تأثیر چندانی در میزان کیفیت سرویسی که کاربر تجربه می‌کند، ندارند. لذا، می‌توان گفت که کیفیت سرویس، در صورت استفاده از IMS به عنوان یک سرویس VOIP در شبکه‌ی اینترنت و یا شبکه‌های محلی، تفاوت چندانی با استفاده از سایر سیستم‌های VOIP ندارد.

برای ایجاد کیفیت سرویس در میدیا مبادله شده، می‌توان از روش‌هایی مانند MPLS^۵ و یا Diffserv^۶ استفاده کرد. روش MPLS، مشابه روش شبکه‌های مدار مجازی عمل می‌کند.^۷ البته، مسیریاب‌های موجود در مسیر طرفین مکالمه، باید همگی از MPLS پشتیبانی کنند. روش Diffserve نیز از طریق تفاوت قائل شدن بین سرویس‌های مختلف و در نظرگرفتن اولویت بالاتر برای عبور بسته‌های سرویس‌های مکالمه‌ای، باعث بالا رفتن نرخ ارسال و کاهش زمان ارسال این نوع بسته‌ها می‌شود[۱۳].

استفاده از روش MPLS و Diffserv، مربوط به پیکربندی شبکه‌ی محلی و یا شبکه‌ی اینترنت می‌باشد. لذا ممکن است که شبکه‌ی مورد نظر، استفاده از این دو روش را پشتیبانی نکند. همچنین، سیستم‌های VOIP متداول نیز می‌توانند مانند IMS، از این روش‌ها بهره‌مند شوند. IMS نیز مانند برخی سیستم‌های VOIP نظری Asterisk، پروتکل SIP را با استفاده از pjsip پیاده‌سازی کرده است. pjsip، یک پشتی پروتکل^۸ SIP است که با زبان برنامه‌نویسی C نوشته شده و متن‌باز است. قابلیت استفاده از Diffserv و قرار دادن مقدار مربوط به اولویت در سرآمد بسته‌های IP و همچنین برخی روش‌های دیگر برای ایجاد کیفیت سرویس را دارد[۱۴].

با توجه به مطالب گفته شده و از آنجایی که pjsip، توسط سایر سیستم‌های VOIP نیز استفاده می‌شود، می‌توان نتیجه گرفت که در بخش میدیا مبادله شده (در صورت استفاده از IMS به عنوان سرور VOIP در بستر اینترنت)، تفاوتی بین IMS و سایر سیستم‌های VOIP وجود ندارد. بنابراین، استفاده از شبیه‌سازی برای مقایسه کیفیت سرویس در میدیا مبادله شده توسط کاربران این سیستم‌ها، کاری بیهوده است. لذا، سیگنالینگ‌های مورد استفاده در این دو سیستم مورد بررسی قرار می‌گیرند تا مشخص شود که آیا سیستم IMS به عنوان یک ارائه‌دهنده‌ی VOIP، نسبت به سایر سیستم‌های متداول VOIP برتری دارد یا خیر.

^۴peer to peer

^۵سرویزه‌ی عبارت Multi Protocol Label Switching و به معنای سوئیچینگ برچسب‌های چندپروتکل

^۶کوتاه‌شده‌ی عبارت Differentiated Service و به معنای سرویس تفاوت قائل شده می‌باشد.

^۷virtual circuit network

^۸Protocol Stack

۱۰.۷ مقایسه سیگنالینگ VOIP و IMS در شبکه اینترنت

بستر ایجاد شده برای انجام مقایسه

در این بخش، سیگنالینگ های مربوط به ثبت نام و برقراری تماس VOIP و IMS (به عنوان ارائه دهنده سرویس VOIP) مورد بررسی قرار می گیرند. سیستم های IMS و VOIP، هر دو به صورت ماشین مجازی بر روی سکوی مجازی سازی Virtualbox نصب شدند. پیاده سازی IMS، به صورت سیستم یکپارچه بر روی سکوی مجازی سازی انجام شد (بخش ۶). نسخه ۱۳ نرم افزار Asterisk [۱۵] نیز به عنوان سرور VOIP بر روی ماشین مجازی لینوکس اوپونتو ۱۶.۰۴ نصب شد. اپلیکیشن zoiper نیز به عنوان اپلیکیشن سمت کاربر مورد استفاده قرار گرفت.

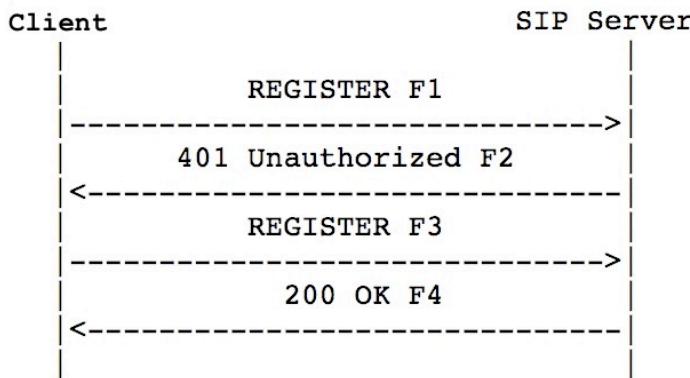
اپلیکیشن zoiper، از طریق ارسال پیام های سیگنالینگ به سیستم میزبان (سیستمی که سکوی مجازی سازی بر روی آن نصب شده است)، عمل ثبت نام کاربر در سیستم را انجام می دهد. سیستم میزبان و دستگاه کاربر، با اتصال به مودم وای فای، آدرس IP خود را به دست می آورند. برای ارتباط با clearwater، لازم است که اپلیکیشن کاربر، به آدرس IP سیستم میزبان دسترسی داشته باشد و پیام ها، به طور مستقیم به ماشین مجازی موردنظر فرستاده نمی شوند؛ بلکه پیام ها از طریق ماشین میزبان، به ماشین مجازی clearwater (از طریق آداتپور شبکه NAT) فرستاده می شوند. برای ارتباط با سرور Asterisk، اپلیکیشن کاربر باید پیام های SIP را مستقیماً به ماشین مجازی که Asterisk بر روی آن نصب شده است، بفرستد. لذا آداتپور شبکه ای این ماشین مجازی، در حالت Bridge قرار داده شد.

مودم وای فای، به ماشین میزبان، ماشین مجازی لینوکس (سرور Asterisk) و دستگاه کاربر، به ترتیب آدرس IP های ۱۹۲.۱.۱۰۶، ۱۹۲.۱.۱۰۷ و ۱۹۲.۱.۱۰۴ را اختصاص داد. zoiper، پیام های سیگنالینگ برای ارتباط با IMS را به آدرس IP سیستم میزبان می فرستد. همچنین، پیام های سیگنالینگ برای ارتباط با Asterisk نیز به آدرس IP ماشین مجازی لینوکس فرستاده می شوند. این دو آدرس IP، هر دو مربوط به اینترفیس وای فای سیستم میزبان می باشند. لذا با capture کردن بسته های عبوری از اینترفیس وای فای سیستم میزبان توسط نرم افزار وایرشارک، می توان پیام های SIP مبادله شده بین دستگاه کاربر و سرور IMS و سرور Asterisk را مشاهده کرد.

ثبت نام SIP

پس از پیکربندی zoiper برای اتصال به سرور IMS و سرور VOIP، عمل ثبت نام به طور خودکار توسط این اپلیکیشن انجام می شود. عمل ثبت نام SIP، در چهار مرحله انجام می شود (شکل ۱.۷). در مرحله ای اول (F1)، کاربر درخواست ثبت نام SIP را به سرور SIP ارسال می کند. این درخواست، شامل نام کاربری (همان شماره هی تلفن)، آدرس IP و یا دامنه کاربر، اپلیکیشن مورد استفاده توسط کاربر و برخی اطلاعات دیگر می باشد. از

آنچایی که سرور نمی‌تواند از طریق این اطلاعات که به صورت متن عادی ارسال شده‌اند، احراز هویت را انجام دهد^۹، این درخواست ثبت‌نام مورد قبول واقع نمی‌شود[۱۶].



شکل ۱.۷: ثبت‌نام SIP

در مرحله‌ی دوم(F2)، سرور SIP، یک چالش امنیتی برای کاربر ایجاد می‌کند تا بتواند از طریق آن، احراز هویت را انجام بدهد. این احراز هویت، برای ایجاد امنیت و جلوگیری از حملاتی مانند حمله‌ی مرد میانی^{۱۰} و حمله‌ی بازپخش^{۱۱} لازم است. کاربر با حل چالش مطرح شده، می‌تواند هویت خود را برای سرور اثبات کند. در این چالش، سرور SIP اطلاعاتی نظیر روش احراز هویت، الگوریتم موردادستفاده و عدد یک‌بار مصرف^{۱۲} را برای کاربر مشخص می‌کند[۱۶].

در مرحله‌ی سوم(F3)، کاربر باید با استفاده از رمزعبور خود، اطلاعات دریافت‌شده از سمت سرور را به‌وسیله‌ی الگوریتم موردنظر رمز کند و خروجی به‌دست‌آمده را به سرور ارسال کند. سرور نیز الگوریتم موردنظر را با همین ورودی‌ها و رمزعبور^{۱۳} اجرا می‌کند. در صورت برابر بودن خروجی این الگوریتم با مقداری که کاربر ارسال کرده است، احراز هویت انجام می‌شود و سرور، ثبت‌نام را تأیید می‌کند. در مرحله‌ی آخر(F4)، سرور به کاربر اطلاع می‌دهد که ثبت‌نام، با موفقیت انجام شد[۱۶].

مقایسه بسته‌های ارسالی و دریافتی

در هر دو سرور(IMS و Asterisk)، پیام‌های مبادله‌شده برای ثبت‌نام SIP، منطبق بر استانداردهای پروتکل SIP است که در بخش قبل توضیح داده شد. پیکربندی این سیستم IMS به‌گونه‌ای است که از پروتکل لایه‌ی انتقال TCP^{۱۴} استفاده می‌کند(شکل ۲.۷) اما سرور Asterisk مورد استفاده، از پروتکل UDP(شکل ۳.۷)

^۹به‌دلیل مسائلی که در زمینه‌ی امنیت شبکه وجود دارد، ارسال رمز عبور نمی‌تواند احراز هویت را به درستی انجام دهد.

^{۱۰}Man In The Middle Attack

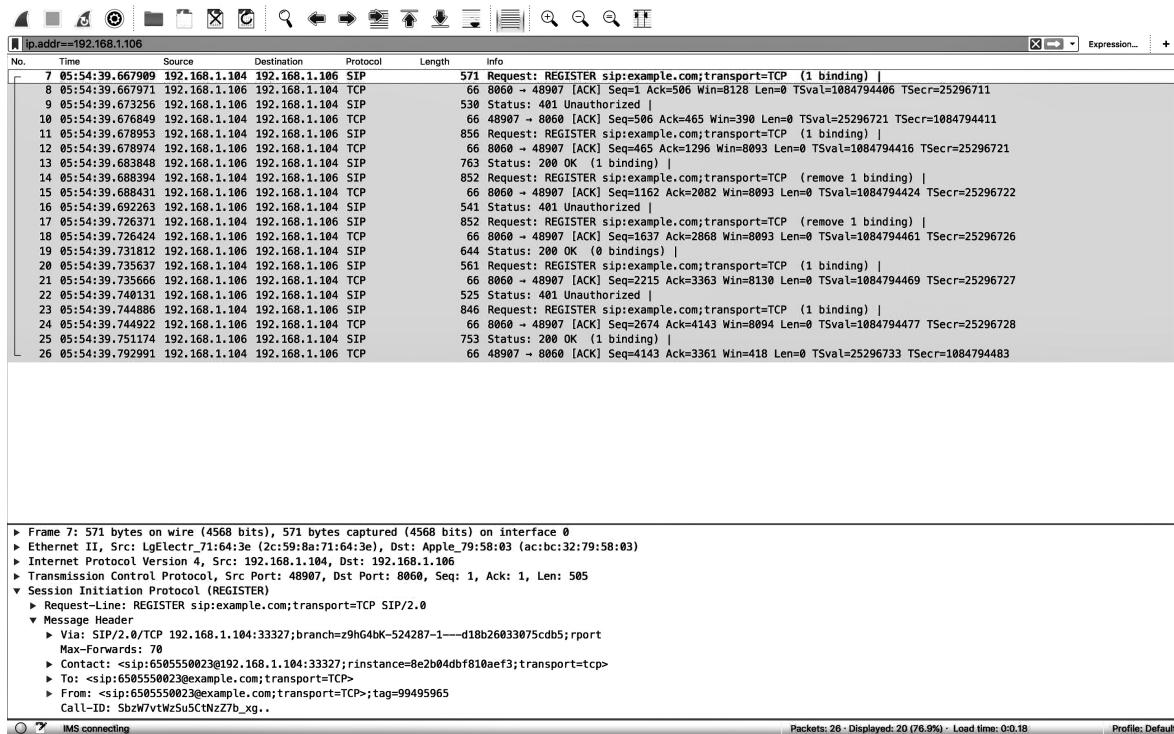
^{۱۱}Replay attack

^{۱۲}Nonce. یک عدد تصادفی که توسط سرور ایجاد می‌شود تا کاربر، در محاسبات الگوریتم احراز هویت، این عدد را نیز دخالت بدهد. با این کار، از حمله‌ی بازپخش جلوگیری می‌شود.

^{۱۳}رمزعبور کاربر در سرور موجود است.

^{۱۴}به‌راحتی می‌توان این پیکربندی را طوری تغییر داد که از پروتکل UDP استفاده کند.

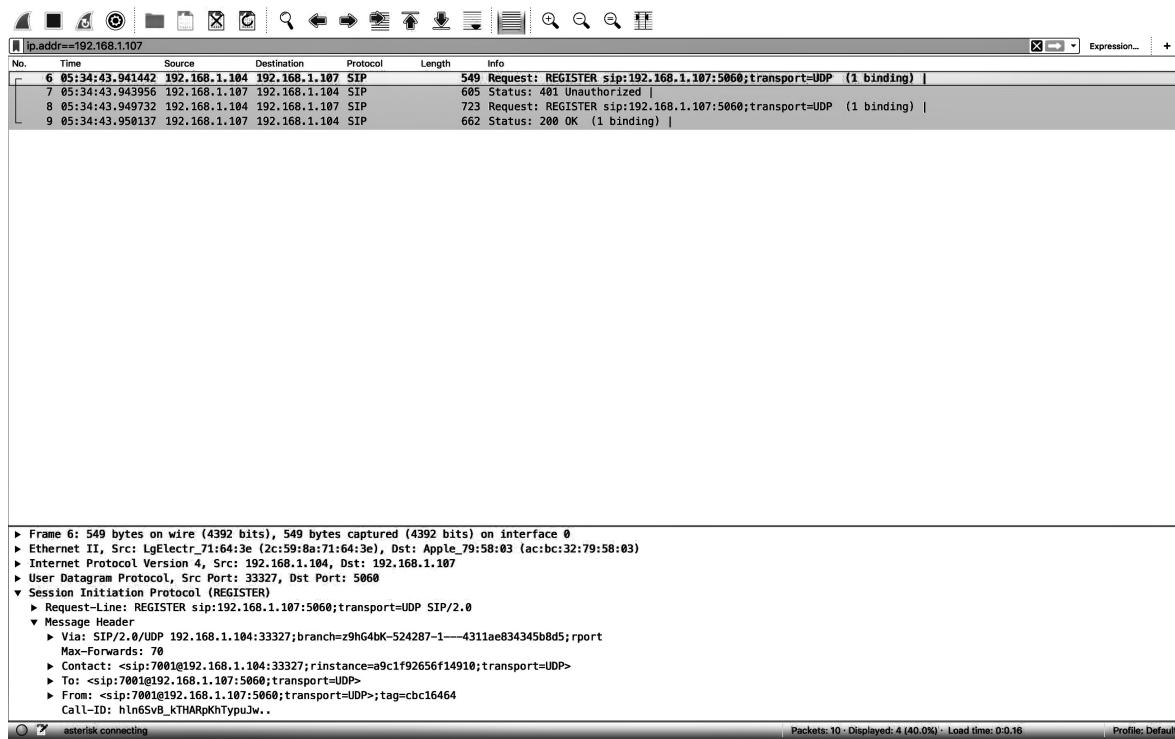
استفاده می‌کند. لذا، با توجه به قابل اعتماد بودن پروتکل TCP، علاوه بر پیام‌های SIP مبادله شده، بسته‌های ACK نیز ارسال می‌شوند [۱۳]. زمان بین اولین درخواست ثبت‌نام توسط کاربر و پایان ثبت‌نام، هم در IMS و هم در VOIP، حدود چندهزارم ثانیه است. لذا، با توجه به نوسانات موجود در شبکه و تفاوت بسیار اندک در مدت زمان ثبت‌نام، نمی‌توان در مورد سرعت سیستم‌های موردنظر در انجام این کار اظهار نظر کرد.



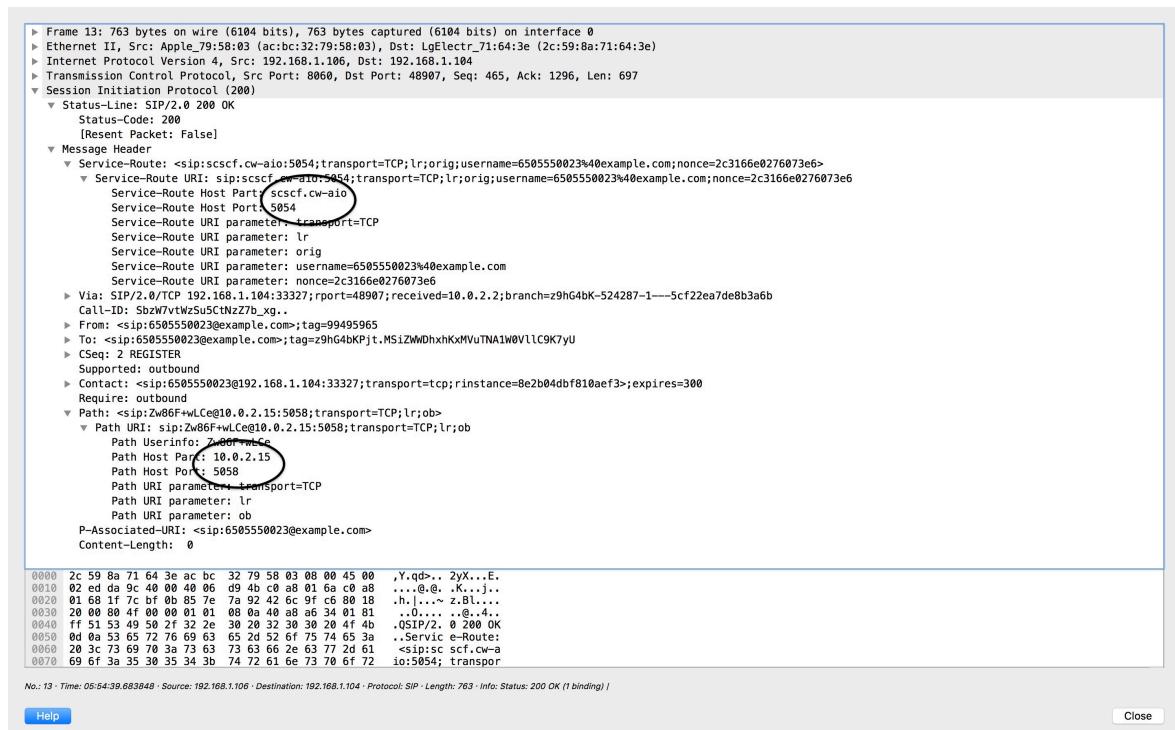
شکل ۷: پیام‌های مبادله شده برای ثبت‌نام در SIP

مطابق مطالب گفته شده در بخش ۲، المان I-CSCF باید در هنگام ثبت‌نام، نود S-CSCF مناسب را به دستگاه کاربر اختصاص دهد. در آخرین مرحله از ثبت‌نام SIP در IMS، نود S-CSCF اختصاص یافته به کاربر به همراه شماره پورت مورد استفاده و همچنین آدرس IP ماشین مجازی clearwater که در پشت NAT قرار دارد، برای کاربر ارسال می‌شود. این اطلاعات، در شکل ۴.۷ مشخص شده‌اند.

همچنین، سیگنالینگ‌های استفاده شده در IMS و VOIP برای ایجاد جلسه و برقراری تماس تلفنی نیز به همین روش مورد بررسی قرار گرفت. با وجود تفاوت‌های جزئی در پیام‌های مبادله شده، هر دو سیستم از استانداردهای پروتکل SIP پیروی می‌کنند. مدت زمان ایجاد جلسه در هر دو سیستم، تقریباً یکسان است و نتایج مشابه نتایج مقایسه‌ی سیگنالینگ‌های ثبت‌نام SIP به دست آمد. لذا از ارائه‌ی جزئیات مربوط به این مقایسه، اجتناب شده است.



شکل ۳.۷: پیام‌های مبادله شده برای ثبت‌نام SIP در VOIP



شکل ۴.۷: اطلاعات ارسال شده به کاربر در آخرین مرحله‌ی ثبت‌نام SIP توسط سرور IMS

۳.۷ مقایسه کیفیت سرویس در شبکه سلولی

برتری اصلی IMS نسبت به سیستم‌های VOIP کنونی، فراهم کردن کیفیت سرویس است. اما همانطور که در بخش‌های قبل گفته شد، در صورت استفاده از IMS به عنوان سرور VOIP، چنین مزیتی وجود ندارد. برقراری کیفیت سرویس توسط IMS، مربوط به کنترل بستر مدیا شبکه سلولی و کنترل بخش شبکه دسترسی رادیویی است. IMS می‌تواند از طریق ارتباط با المان‌های شبکه سلولی در ناحیه دسترسی رادیویی و همچنین المان‌هایی که بستر عبور مدیا در شبکه سلولی را فراهم می‌کنند، کیفیت سرویس موردنیاز کاربران را فراهم کند. سیستم‌های VOIP متدالو، چنین قابلیتی را ندارند.

کیفیت سرویسی که IMS برای هر جلسه فراهم می‌کند، به عوامل مختلفی بستگی دارد. عامل اول، پروفایل کاربر موردنظر می‌باشد. کاربرانی که سرویس طلایی را خریداری می‌کنند، نسبت به کاربران عادی کیفیت سرویس بهتری را دریافت می‌کنند. عامل دیگر، نوع سرویس مورد استفاده است. استاندارد 3GPP، چهار کلاس کیفیت سرویس را تعریف کرده است که به آن، کلاس ترافیک نیز می‌گویند. کلاس‌های ترافیک، به ترتیب اولویت برای دریافت کیفیت سرویس بالاتر، به شرح زیر هستند^[۲]

۱. کلاس مکالمه: مانند مکالمات صوتی و ویدیوئی

۲. کلاس streaming: مانند تماشای ویدیوهای زنده و یا از قبل ضبط شده به صورت آنلاین

۳. کلاس تعاملی^{۱۵}: مانند وب‌گردی

۴. کلاس پس‌زمینه^{۱۶}: مانند پست الکترونیک

المان PCRF در معماری IMS، مسئول برقراری ارتباط با المان‌های معماری شبکه سلولی و فراهم کردن کیفیت سرویس است. این المان، معمولاً به عنوان بخشی از المان P-CSCF پیاده‌سازی می‌شود^[۲]. در پیاده‌سازی clearwater، المان Bono نقش P-CSCF را دارد. المان Bono که به صورت متن‌باز پیاده‌سازی شده است، فاقد المان PCRF بوده و کیفیت سرویس را فراهم نمی‌کند. شرکت Metaswitch که پروژه clearwater را راهاندازی کرده و رهبری می‌کند، المانی به نام Perimeta را به عنوان محصولی تجاری، ارائه می‌کند. این المان که یک کنترل‌کننده مرز جلسه است، به جای المان Bono در معماری clearwater قرار می‌گیرد. با استفاده از Perimeta، ارتباط clearwater با شبکه سلولی برقرار می‌شود. Perimeta، علاوه بر فراهم کردن کیفیت سرویس، خدمات شارژینگ را نیز انجام می‌دهد^[۱۷]. به دلیل در اختیار نداشتن این محصول تجاری (و گران‌قیمت بودن آن)، امکان بررسی کیفیت سرویس فراهم شده توسط IMS در شبکه سلولی، وجود ندارد؛ اما با توجه به ادعای شرکت Metaswitch و نظرات مشتریانی که از Perimeta استفاده کرده‌اند، می‌توان قبول کرد که پیاده‌سازی clearwater، توانایی فراهم کردن کیفیت سرویس در شبکه سلولی را دارد.

فصل ۸

اتصال IMS به شبکه‌ی 4G

همانطور که در بخش‌های قبل توضیح داده شد، IMS می‌تواند از طریق المان PGW به معماری نسل چهارم ارتباطات متصل شود. برای ارتباط پیاده‌سازی clearwater به VoLTE (به طوری که IMS قادر به تعامل با المان‌های شبکه‌ی سلولی باشد)، نیاز به محصول تجاری Perimeta می‌باشد که توسط شرکت Metaswitch ارائه می‌شود. همچنین، می‌توان از PCRF مورد استفاده توسط اپراتورهای تلفن همراه نیز برای این کار استفاده کرد؛ اما ممکن است که PCRF مورد استفاده توسط این اپراتورها، برخی قابلیت‌ها مانند تضمین کیفیت سرویس را نداشته باشد. با توجه به مطالب فوق، تصمیم بر آن شد که clearwater، به عنوان یک ارائه‌دهنده‌ی سرویس VOIP، به هسته‌ی معماری نسل چهارم ارتباطات متصل شود و قابلیت برقراری تماس برای کاربران نسل چهارم از طریق سوئیچ بسته‌ای فراهم شود.

۱۰.۸ پیاده‌سازی نسل چهارم ارتباطات

پیاده‌سازی‌ها مختلف

پیاده‌سازی‌های مختلفی برای معماری نسل چهارم ارائه شده‌اند که یکی از مهم‌ترین و کامل‌ترین آن‌ها، پیاده‌سازی OpenAirInterface است. یک پیاده‌سازی متن‌باز از نسل چهارم ارتباطات است که این معماری را در دو بخش eNB و EPC پیاده‌سازی کرده است. srsLTE نیز یک پیاده‌سازی متن‌باز است و در ابتدا فقط eNB را پیاده‌سازی کرده بود ولی اخیراً، EPC مخصوص به خود را نیز ارائه می‌کند. پیاده‌سازی‌های دیگری نظیر Amarisoft نیز وجود دارند. طبق بررسی‌های انجام‌شده،^۱ oai^۲ و srs^۳، بهترین پیاده‌سازی‌های موجود متن‌باز از معماری نسل چهارم ارتباطات هستند. لذا در این پژوهه، ابتدا عملکرد این دو پیاده‌سازی مورد بررسی قرار گرفت و سپس گزینه‌ی بهتر، برای اتصال IMS به 4G مورد استفاده قرار گرفت.

^۱ سرواژه‌ی عبارت OpenAirInterface. در این گزارش، منظور از oai، پیاده‌سازی نسل چهارم ارتباطات توسط OpenAirInterface است.

^۲ سرواژه‌ی عبارت Software Radio System. در این گزارش، منظور از srs، پیاده‌سازی نسل چهارم ارتباطات توسط شرکت Software Radio System است.

مقایسه‌ی پیاده‌سازی‌های متن‌باز oai و srs

این امکان وجود دارد که eNB پیاده‌سازی شده توسط سایر شرکت‌ها نظیر OpenAirInterface و Amarisoft Gringoli همکاران [۱۸]، مقایسه‌ای برای بررسی عملکرد سیستم 4G در دو حالت مختلف انجام شده است. در هر دو حالت، از oai EPC به عنوان هسته‌ی شبکه استفاده شده است. در حالت اول، از oai eNB و در حالت دوم، از srs eNB استفاده شده است. در این مقاله، با استفاده از سه دستگاه کاربر مختلف، نرخ ارسال (دريافت) داده‌ها و همچنین میزان استفاده از CPU توسط سرورهای مورد استفاده، بررسی شده است. در مورد نرخ ارسال داده‌ها، هر دو حالت تقریباً برابر می‌کنند؛ اما در استفاده از CPU، حالت اول که از oai eNB استفاده می‌کند، بسیار عملکرد بهتری دارد و از توان CPU کمتر استفاده می‌کند.

پیاده‌سازی oai و srs

در انجام این پروژه، oai و srs پیاده‌سازی شدند. در پیاده‌سازی oai [۱۹]، از گزارش پروژه‌ی کارشناسی یک از دانشجویان^۳ دانشکده برق و کامپیوتر دانشگاه صنعتی اصفهان تحت عنوان "پیاده‌سازی نرمافزاری شبکه‌ی LTE" استفاده شد. پس از پیاده‌سازی oai، تلفن همراه از طریق سیم‌کارت پروگرام شده، به هسته‌ی شبکه متصل شد و از طریق آن، به شبکه‌ی محلی دانشگاه و همچنین شبکه‌ی اینترنت دسترسی پیدا کرد.

srs با استفاده از مستندات موجود در [۲۰] پیاده‌سازی شد^۴. پس از پروگرام کردن سیم‌کارت، متناسب با پیکربندی و اطلاعات درون پایگاه داده‌ی srs، دستگاه کاربر با استفاده از سیم‌کارت مذکور به srs eNB متصل شد. با مشاهده بسته‌های مبادله‌شده بین srs EPC و srs eNB، مشخص گردید که کاربر نمی‌تواند به متصل شود. با جستجو در تالار گفتمان srs مشخص شد که افراد دیگری نیز با همین مشکل رویه‌رو شده‌اند و راه حلی یافت نشد. به احتمال زیاد، به دلیل این‌که srs به تازگی پیاده‌سازی شده است، نیاز به رفع اشکال و عیوب‌یابی دارد. با توجه به این مشکل و همچنین به دلیل عملکرد بهتره oai، تصمیم گرفته شد که این سیستم برای اتصال به IMS مورد استفاده قرار گیرد.

۲.۸ اتصال IMS به هسته‌ی شبکه‌ی 4G

در پیاده‌سازی oai، نود eNB از طریق کابل USRP B210 به بورد می‌شود و شبکه‌ی دسترسی رادیویی را تشکیل می‌دهند. نود EPC نیز حاوی المان‌های هسته‌ی شبکه‌ی سلوی است. این المان‌ها عبارتند از:

^۳ امیر رضا بلوجی

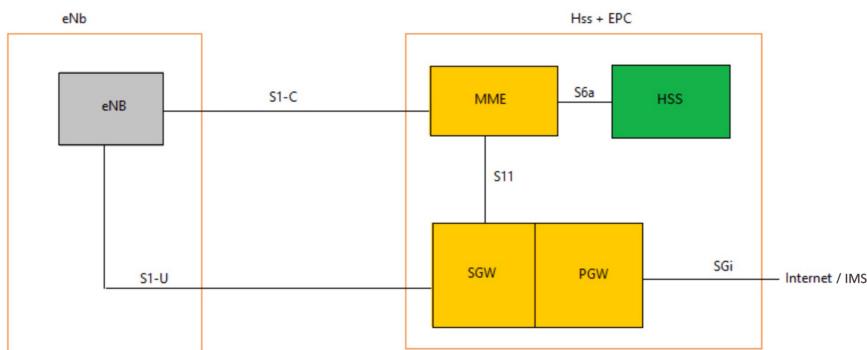
^۴ کل پروژه‌ی srsLTE پیاده‌سازی شد. در این پیاده‌سازی، هم در بخش eNB و هم در بخش EPC، از srs استفاده شد.

- HSS: مسئول نگهداری پروفایل دیتای مشترکین

- MME: مسئول مدیریت جابه‌جایی

- SPGW: مسئول برقراری ارتباط با اینترنت و IMS

معماری oai، مطابق شکل ۱.۸ می‌باشد. ابتدا بخش eNB و EPC بر روی دو رایانه‌ی مجزاً نصب و پیکربندی شدند. پس از برقراری ارتباط بین eNB و EPC از طریق کابل اترنت، سیستم به‌طور کامل راهاندازی شد. سپس سیم‌کارت‌ها، متناسب با پیکربندی و اطلاعات پایگاه داده‌ی oai پروگرام شدند. پس از قرار دادن سیم‌کارت پروگرام شده در دستگاه تلفن همراه، سیم‌کارت با فرستادن سیگنال‌های ارتباطی به بورد USRP B210، ابتدا به eNB متصل شد. eNB نیز اطلاعات دریافت شده از کاربر را به EPC ارسال کرد و عمل ثبت‌نام پایان یافت. پس از پایان ثبت‌نام، کاربر با فعال کردن دیتای تلفن همراه خود، می‌تواند آدرس IP به‌دست آورد و از طریق SPGW، به اینترنت متصل شود.



شکل ۱.۸: معماری مورد استفاده در پیاده‌سازی oai

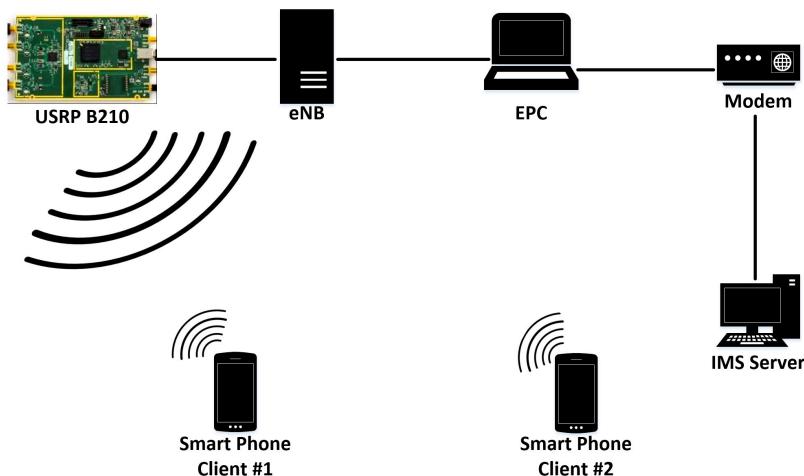
در این پروژه، سیستمی که EPC و المان SPGW روی آن نصب شده است، از طریق شبکه‌ی داخلی دانشگاه به اینترنت جهانی متصل می‌شود. لذا، در پیکربندی VPN، اینترفیس SPGW به عنوان رابط SPGW با شبکه‌ی خارجی معروفی شد. در صورت قرار دادن اینترفیس مربوط به شبکه‌ی داخلی دانشگاه^۵ در پیکربندی مربوطه، می‌توان ارتباط oai با اینترنت جهانی را قطع کرده و صرفاً شبکه‌ی داخلی دانشگاه را در اختیار مشترکین قرار داد. با این کار، مشترکین قادر به استفاده از شبکه‌ی دانشگاه و IMS خواهند بود ولی دیگر از طریق سیم‌کارت مربوط به oai، نمی‌توانند به اینترنت جهانی دسترسی داشته باشند. از آنجایی که استفاده از شبکه‌ی داخلی دانشگاه برای کاربران رایگان است، دیگر لازم نیست که میزان استفاده این کاربران از اینترنت جهانی (از طریق اتصال دیتای سیم‌کارت و بدون استفاده از VPN) را کنترل کرد. از آنجایی که خدمات شارژینگ در المان PCRF انجام می‌شود و این المان، با عنوان تجاری Perimeta به فروش می‌رسد و ما به آن دسترسی نداریم، این راه حل، کمک به ارائه‌ی سرویس IMS و LTE در سطح دانشگاه می‌کند.

⁵ مثلاً می‌توان اینترفیس وای‌فایی که به یکی از مودم‌های شبکه‌ی دانشگاه متصل است را برای این کار در نظر گرفت.

برای اتصال IMS به oai، کافی است که دسترسی این دو سیستم به یکدیگر از طریق پروتکل IP را فراهم شود. در صورت دسترسی SPGW به IMS از طریق IP، کاربران oai نیز قادر به دسترسی به IMS هستند و می‌توانند عملیات ثبت‌نام در IMS را انجام دهند و تماس تلفنی برقرار کنند. ساده‌ترین راه برای انجام این کار، اتصال سیستم IMS و EPC به یک شبکه‌ی محلی مشترک است.

۳.۸ برقراری تماس تلفنی بین کاربران 4G توسط IMS

در این پروژه، به منظور برقراری این ارتباط، هر دو فای مرکز اوپیونیک دانشگاه متصل شدند. از طریق آدرس IP خصوصی که مودم به این دو سیستم اختصاص داد، ارتباط آن‌ها با یکدیگر برقرار شد و دسترسی دو سیستم به یکدیگر، از طریق ping بررسی شد. از پیاده‌سازی یکپارچه clearwater بر روی Virtulabox به عنوان سیستم IMS استفاده شد. پس از اتصال تلفن همراه کاربر از طریق سیم‌کارت به oai و روشن کردن دیتای اینترنت تلفن همراه، دسترسی تلفن همراه کاربر به IMS فراهم شد. با نصب اپلیکیشن zoiper بر روی تلفن همراه و انجام پیکربندی لازم مطابق بخش ۳.۶، ثبت‌نام کاربر در IMS با موفقیت انجام شد. با انجام روشی مشابه، کاربر دوم نیز به IMS متصل شد. سپس، دو کاربر توانستند با یکدیگر تماس تلفنی برقرار کنند.



شکل ۲.۸: توپولوژی ارتباط IMS و 4G و برقراری تماس صوتی به‌وسیله‌ی این شبکه

فصل ۹

نتیجه‌گیری

تحقیقات انجام‌گرفته در این پژوهه در مورد معماری، دلایل پیدایش، مزایا و کاربردهای IMS، باعث ترقیب اپراتورهای تلفن همراه برای استفاده از این سیستم و همچنین هموار شدن مسیر برای راهاندازی و استفاده آن می‌شود. پیاده‌سازی IMS توسط پروژه‌ی متن‌باز clearwater در این پژوهه، می‌تواند اثباتی بر این موضوع باشد که می‌توان این سیستم را به صورت بومی و بدون نیاز به کشورهای خارجی راهاندازی کرد. وجود روش‌های پیاده‌سازی در مقیاس بزرگ و همچنین تجربیات به دست آمده از پیاده‌سازی این سیستم در مقیاس کوچک، این امکان را به وجود می‌آورد که بتوان سرویس IMS را در آینده‌ای نزدیک و در مقیاس بزرگ راهاندازی کرد. با این کار، علاوه بر کاهش هزینه‌های اپراتورها، سودآوری آن‌ها از طریق ارائه‌ی سرویس‌های نوظهور افزایش خواهد یافت و کاربران تلفن همراه از این سرویس‌ها بهره‌مند خواهند شد. راهاندازی IMS، امکان ورود شرکت‌های شخص ثالث به بازار مخابرات را نیز فراهم می‌کند.

در این پژوهه، پس از راهاندازی شبکه‌ی 4G و IMS به وسیله‌ی پیاده‌سازی‌های متن‌باز، ارتباط این دو سیستم با یکدیگر برقرار گردید. سیستم IMS به عنوان یک ارائه‌دهنده‌ی سرویس صدا بر روی IP به شبکه‌ی 4G متصل شد و برقراری تماس صوتی نیز از طریق سیم‌کارت و به وسیله‌ی IMS صورت گرفت. نرم‌افزاری شدن هسته‌ی شبکه‌های سلولی و پیاده‌سازی نرم‌افزاری IMS، باعث عدم وابستگی اپراتورهای تلفن همراه به شرکت‌های خارجی می‌شود و تحریم در حوزه‌ی مخابرات نمی‌تواند باعث ایجاد مشکل در ارائه‌ی سرویس اپراتورهای تلفن همراه شود.

اقدام به برقراری ارتباط IMS با شبکه‌ی تلفن ثابت، از کارهای دیگری است که در این پژوهه صورت گرفت. به دلیل عدم دسترسی به موقع به تجهیزات موردنیاز، این ارتباط برقرار نشد. کارهای انجام‌شده در این زمینه، در پیوست (پ) آورده شده است. اتصال IMS به شبکه‌ی تلفن ثابت و برقراری تماس تلفنی بین کاربران تلفن ثابت و کاربران IMS، در آینده‌ای نزدیک انجام خواهد گرفت.

در این پژوهه، کیفیت سرویس ارائه‌شده توسط IMS و VOIP مورد مقایسه قرار گرفت. با انجام این

مقایسه، مشخص گردید که در صورت استفاده از IMS به عنوان ارائه‌دهنده سرویس VOIP در بستر اینترنت و یا شبکه‌ی محلی، هر دو سیستم از قابلیت یکسانی در فراهم کردن کیفیت سرویس دارند. اما در صورت اتصال IMS به شبکه‌ی سلولی، این سیستم می‌تواند کیفیت سرویس را برای کاربران شبکه‌ی سلولی فراهم کند. با این وجود، IMS این قابلیت را دارد که به عنوان یک ارائه‌دهنده سرویس VOIP به حجم عظیمی از کاربران، مورد استفاده قرار گیرد.

پیاده‌سازی IMS به صورت توزیع شده و در مقیاس بزرگ به عنوان سرور VOIP دانشگاه صنعتی اصفهان از کارهای پیش رو است. دانشجویان و سایر دانشگاهیان که کاربران اصلی این سیستم هستند، می‌توانند با استفاده از بستر شبکه‌ی دانشگاه، از تماس تلفنی رایگان داخلی (در سطح دانشگاه) بهره‌مند شوند. همچنین، با پیشرفت پروژه‌ی IUT-LTE^۱، دانشگاهیان می‌توانند از طریق سیم‌کارت، به هسته‌ی این شبکه‌ی سلولی متصل شوند و توسط IMS تماس تلفنی برقرار کنند.

اتصال IMS از طریق PCRF به شبکه‌ی سلولی به صورت جدی پیگیری خواهد شد. با این کار، IMS می‌تواند با المان‌های شبکه‌ی سلولی تعامل کند و کیفیت سرویس را فراهم کند. همچنین، خدمات مالی حساب کاربران نیز از طریق PCRF صورت می‌گیرد. لذا با استفاده از آن، می‌توان این سیستم را به صورت تجاری عرضه کرد. عدم وابستگی IMS به شبکه‌ی دسترسی، از دیگر مزایای این سیستم است که به کاربران امکان جابه‌جایی بین شبکه‌های دسترسی مختلف و همچنین برقراری تماس تلفنی از طریق دستگاه‌های مختلف (نظیر تلفن همراه، رایانه‌ی شخصی و ...) را می‌دهد. این قابلیت، در کنار سرویس‌های نوظهوری که توسط IMS ارائه می‌شوند، باعث افزایش تمایل کاربران به استفاده از این سیستم می‌شود و برای اپراتورهای تلفن همراه، مزیت رقابتی ایجاد می‌کند.

ایجاد اپلیکیشن سرورهای جدید برای ارائه سرویس‌های نوظهور توسط IMS، از دیگر برنامه‌هایی است که قرار است در دست کار قرار گیرد. ارائه سرویس‌هایی نظیر تماس ویدیوئی، کنفرانس صوتی و ویدیوئی، PoC و IP messaging ارزش افزوده‌ی بالایی دارد. این سرویس‌ها، می‌توانند جایگزینی برای اپلیکیشن‌های خارجی متداول نظیر Skype، واتس‌اپ و تلگرام باشند. با توجه به استفاده‌ی این سیستم از ترافیک داخلی کشور و عدم استفاده از سرویس‌های خارج از کشور، هزینه‌ی استفاده از این سرویس نسبت به سرویس‌های مشابه خارجی کمتر بوده و از امنیت بالاتری برخوردار است.

^۱ پروژه‌ی پیاده‌سازی نرم‌افزاری شبکه‌ی LTE در دانشگاه صنعتی اصفهان

پیوست آ

مسائل و مشکلات Virtualbox

آ. نصب

فایل نصبی برنامه Virtualbox را می‌توان از سایت virtualbox.org، متناسب با نوع سیستم عامل خود دانلود کرد. توصیه می‌شود که بر روی سیستم عامل اوبونتو نصب شود. برای نصب بر روی سیستم عامل اوبونتو، پس از دریافت فایل نصبی، از طریق خط فرمان، به پوشه‌ای که فایل در آن ذخیره شده است^۱ رفته و دستور زیر را اجرا کنید:

1. sudo dpkg -i virtualbox*

روش دیگر، نصب بسته از طریق مخازن نرم‌افزار است. ابتدا مخازن خود را با دستور شماره 2 بهروز کرده و سپس دستور شماره 3 را اجرا کنید. در صورت موجود نبودن نرم‌افزار Virtualbox در مخازن و عدم نصب آن، با دستورات 4 تا 6، کلید مخازن Virtualbox و مخزن آن را به سیستم خود اضافه کرده و سپس، دستورات 2 و 3 را اجرا کنید. البته ممکن است به دلیل فیلترینگ و یا عدم اجازه‌ی دسترسی کاربران ایران به مخازن Virtualbox، استفاده از این روش امکان‌پذیر نباشد.

2. sudo apt-get update

3. sudo apt-get install virtualbox

4. wget -q https://www.virtualbox.org/download/oracle_vbox_2016.asc -O- | sudo apt-key add -

5. wget -q https://www.virtualbox.org/download/oracle_vbox.asc -O- | sudo apt-key add -

6. sudo sh -c 'echo "deb http://download.virtualbox.org/virtualbox/debian \$(lsb_release -sc) contrib" » /etc/apt/ sources.list.d/virtualbox.list'

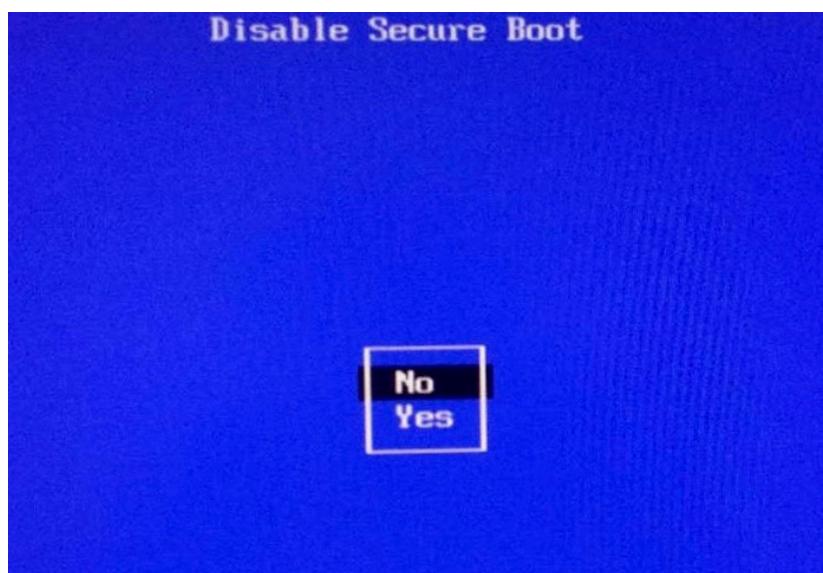
¹ معمولاً پوشه‌ی Download

۲۰ آ خطا اجرا کردن ماشین مجازی

پس از نصب Virtualbox و اضافه کردن ماشین مجازی موردنظر، ممکن است در هنگام اجرای ماشین مجازی با خطای Required key not available و یا خطاهایی با این مضامون مواجه شوید. دلیل وجود این خطا، عدم اجازه اجرای ماژول‌های کرنل امضانشده^۲ توسط کرنل‌های اوپنونتوی نسخه ۴.۴.۰ به بعد است. این عدم اجازه، مربوط به زمانی است که boot امن فعال باشد. بنابراین، برای رفع این خطا دو راه وجود دارد. راه اول، غیرفعال کردن boot امن سیستم عامل است. راه دوم، امضا کردن این ماژول‌های کرنل به صورت دستی است.

غیرفعال کردن boot امن

برای غیرفعال کردن boot امن سیستم عامل، می‌توان وارد BIOS یا UEFI شد و از طریق تنظیمات آن، امن را غیرفعال کرد. این کار، ساده‌ترین روش است. اما با توجه به این‌که انجام این تنظیمات، در BIOS یا UEFI های مختلف، متفاوت است و ممکن است به راحتی قادر به انجام این کار نباشد، یک راه نرم‌افزاری برای این کار ارائه شده است که برای تمام سیستم‌ها، یکسان است. برای این کار، ابتدا با دستور شماره ۷، نرم‌افزار mokutil را نصب کنید. سپس دستور شماره ۸ را اجرا کنید. بعد از اجرای دستور شماره ۸، از شما خواسته می‌شود که یک رمزعبور به طول حداقل ۸ حرف را ایجاد کنید. پس از انجام این کار، سیستم عامل را reboot کنید. بعد از reboot، صفحه‌ای مطابق شکل آ.۱ نمایش داده می‌شود.



شکل آ.۱: صفحه‌ای که پس از reboot نمایش داده می‌شود

در این صفحه از شما پرسیده می‌شود که آیا می‌خواهید تنظیمات امنیتی را تغییر دهید؟ گزینه Yes را انتخاب کنید. پس از انتخاب این گزینه از شما خواسته می‌شود که رمزعبوری را که ایجاد کردید، وارد کنید. پس از

^۲ Unsigned Kernel Module

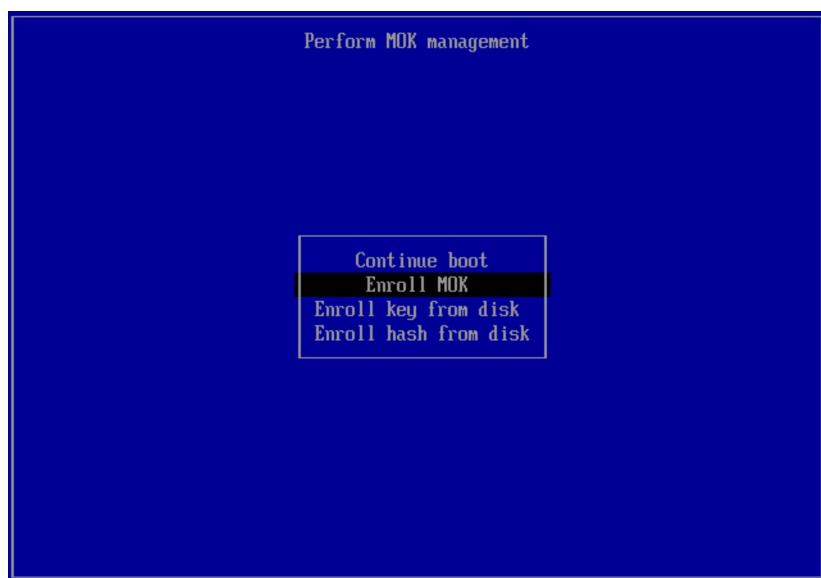
وارد کردن رمز عبور، مراحل غیرفعال کردن boot امن پایان میابد. توجه شود که در برخی از نسخه های UEFI از شما نمی خواهند که رمز عبور را به طور کامل وارد کنید؛ بلکه می خواهند بعضی از حروف رمز عبور را وارد کنید. مثلاً از شما می خواهند که حرف اوّل، سوم و هشتم رمز عبور خود را وارد کنید.

7. sudo apt-get install mokutil
8. sudo mokutil --disable-validation

امضا کردن مازول ها

برای امضا کردن مازول ها، ابتدا باید با دستور شماره ۹، کلیدهای لازم برای امضا کردن را ایجاد کرد. سپس، با اجرای دستور شماره ۱۰، مازول ها را امضا کرده و در آخر نیز با دستور شماره ۱۱، کلیدها را در boot امن، ثبت کرد. در صورتی که دستور شماره ۱۱ به درستی اجرا نشد، ابتدا با دستور شماره ۷، برنامه mokutil را نصب کرده و سپس دستور شماره ۱۱ را اجرا کنید. پس از اجرای این دستور، از شما خواسته می شود که یک رمز عبور به طول حداقل ۸ حرف را ایجاد کنید. پس از انجام این کار، سیستم عامل را reboot کنید. پس از انجام این مراحل، وارد صفحه ای مشابه آ. ۲ خواهید شد.

9. openssl req -new -x509 -newkey rsa:2048 -keyout MOK.priv -outform DER -out MOK.der -nodes -days 36500 -subj "/CN=Descriptive name/"
10. sudo /usr/src/linux-headers-\$(uname -r)/scripts/sign-file sha256 ./MOK.priv ./MOK.der /path/to/module
11. sudo mokutil --import MOK.der

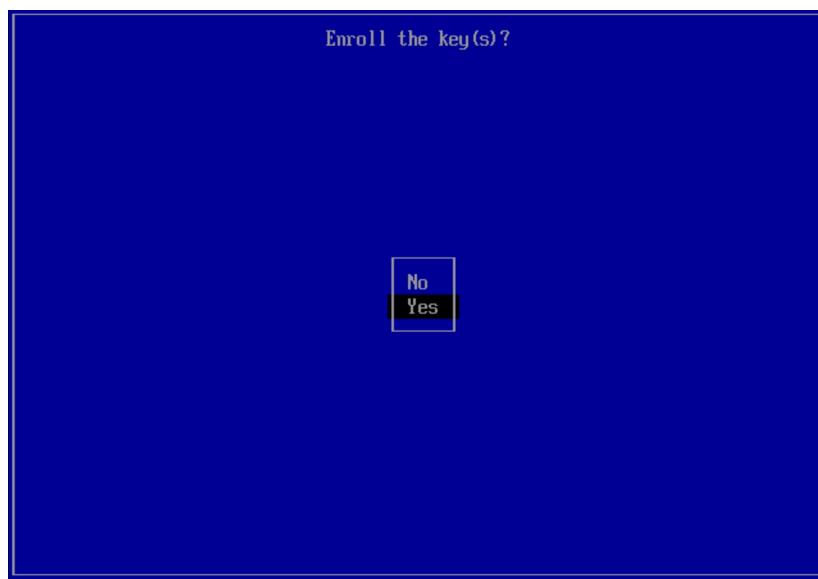


شکل آ. ۲: مدیریت MOK

در این صفحه، گزینه‌ی Enroll MOK را انتخاب کنید. سپس وارد صفحه‌ای مشابه آ.۳ می‌شوید. در این صفحه، گزینه Continue را انتخاب کنید. سپس وارد صفحه‌ی تأیید ثبت کلیدها(شکل آ.۴) خواهیدشد. گزینه‌ی Yes را انتخاب کنید. باقی مراحل، مانند روش قبل می‌باشد؛ یعنی وارد صفحه‌ای می‌شوید که از شما خواسته می‌شود رمزعبوری را که ساخته‌اید وارد کنید.



شکل آ.۳: ثبت کلیدها



شکل آ.۴: تأیید ثبت

پیوست ب

مسائل و مشکلات آزمایش clearwater

دستور شماره ۱

1. sudo apt-get install build-essential git --yes

دستور فوق که اوّلین دستور از مجموعه دستورات این بخش می‌باشد، build-essential را دانلود و بر روی سیستم نصب می‌کند. build-essential، یک meta package است که شامل reference^۱ به تعداد زیادی بسته^۲ می‌باشد و این بسته‌ها را بر روی سیستم نصب می‌کند. این بسته‌های نصب شده، پیش‌نیاز استفاده از سایر زبان‌ها برنامه‌نویسی در سیستم مورد نظر می‌باشند. همچنین، هرگاه لازم باشد برنامه‌ای از طریق کد منبع اجرا شود، باید این بسته‌ها از قبل روی سیستم نصب شده باشند.

در واقع، یک build essential reference به تمام بسته‌های مورد نیاز برای کامپایل کردن یک نرم‌افزار یا یک بسته دیبیان است. این بسته‌ها، به‌طور معمول، شامل کامپایلرهای gcc/g++، کتابخانه‌های C و C++ و یک سری ابزار دیگر می‌باشند. با توجه به مطالب گفته شده، دستور فوق، بسته‌های مورد نیاز برای استفاده از زبان Ruby را بر روی سیستم ما نصب می‌کند.

دستور شماره ۲

2. curl -L http://get.rvm.io | bash -s stable

دستور فوق، RVM^۳ را بر روی سیستم نصب می‌کند. RVM یک ابزار خط فرمان است که امکان نصب آسان، مدیریت و کار با Ruby را فراهم می‌کند. به احتمال زیاد، به دلیل عدم وجود کلید gpg^۴ بر روی سیستم، دستور فوق به درستی انجام نشود. برای ایجاد یا به دست آوردن این کلید، یکی از دو دستور زیر باید اجرا شوند. پس از

^۱ ارجاع

^۲ Package

^۳ Ruby Version Manager

^۴ GNU Privacy Guard: برای رمزگاری داده‌های ارسالی بین سرور و کلاینت استفاده می‌شود

اجرای یکی از دستورات زیر، مشکل رفع خواهد شد.

I) gpg --keyserver hkp://keys.gnupg.net --recv-keys 409B6B1796C275462A1703

113804BB82D39DC0E37D2BAF1CF37B13E2069D6956105BD0E739499BDB

II) curl -sSL https://rvm.io/mpapis.asc | gpg --import -

دستور شماره ۳

3. source ~/.rvm/scripts/rvm

برخی از اسکریپت‌های bash، بهجای این‌که توسط یک دستور اجرایی^۵، اجرا شوند، باید توسط syntax زیر اجرا شوند. در دستور فوق، به جای عبارت <the script name>، باید مکان قرارگیری اسکریپت و نام آن را قرار داد.

source <the script name>

مشکلی که امکان دارد در اجرای دستور شماره ۳ به وجود آید، تفاوت در پوششی نصب rvm است. برای رفع این مشکل، می‌توانید آدرس پوششی موردنظر را بهجای علامت ~ قرار دهید. راه دیگر این است که خط فرمان در هنگام اجرای دستورات ۱ و ۲، در پوششی خانه قرار داشته باشد. آدرس این پوشش، به صورت ~ و یا /home/ubuntu است.

دستور شماره ۴

4. rvm autolibs enable

دستور فوق، autolibs را فعال می‌کند. یک ویژگی پیاده‌سازی شده در RVM است که امکان نصب خودکار یک سری وابستگی‌ها را بر روی سیستم فراهم می‌کند. این وابستگی‌ها، معمولاً برنامه‌هایی از قبیل OpenSSL و YAML می‌باشند.

دستور شماره ۵ و ۶

5. rvm install 1.9.3

6. rvm use 1.9.3

برای اجرای تست زنده، نیاز است که نسخه 1.9.3 زبان Ruby بر روی سیستم نصب شده باشد. دستور شماره ۵، با استفاده از RVM این کار را انجام می‌دهد. دستور شماره ۶ نیز به خط فرمان دستور می‌دهد که به صورت پیش‌فرض، از این نسخه Ruby استفاده شود.

دستور شماره ۷

دستور زیر، کدهای تست زنده‌ی clearwater را از git دریافت می‌کند.

7. `git clone -b stable --recursive git@github.com:Metaswitch/clearwater-live-test.git`

دستور شماره ۸ و ۹

8. `cd /clearwater-live-test`

9. `bundle install`

احتمالاً پس از اجرای دستور شماره ۹، پیغام خطای Could not find quaff-0.7.4 in any of the sources نمایش داده شود. برای رفع این مشکل باید دستور زیر را اجرا کرد.

`git install quaff -v 0.7.4`

پیوست پ

اتصال به تلفن ثابت

یکی از مسائل مهم در پیاده‌سازی clearwater، ارتباط این معماری با سایر شبکه‌ها نظیر PSTN است. با اتصال IMS به PSTN یا ISDN^۱، کابران تلفن ثابت و کاربران IMS می‌توانند با یکدیگر تماس تلفنی برقرار کنند. شبکه‌های PSTN، ساختاری قدیمی و آنالوگ دارند اما شبکه‌های ISDN، ساختاری جدید و دیجیتال دارند. برای اتصال با PSTN یا ISDN، باید از SIP trunk استفاده کرد. SIP، پیام‌های کنترلی مربوط به را به پیام‌های کنترلی مورد استفاده در خطوط تلفن ثابت تبدیل می‌کند. از آنجایی‌که سرویس VOIP و سرویس IMS، هر دو از پروتکل SIP استفاده می‌کنند، می‌توان از SIP trunk هایی که برای اتصال سرویس VOIP به تلفن ثابت استفاده می‌شود، برای IMS نیز استفاده کرد.

پ.۱ روش‌های استفاده از SIP trunk

استفاده از SIP trunk تأمین‌کنندگان VOIP

روش‌های متفاوتی برای استفاده از SIP trunk و اتصال IMS به شبکه‌ی تلفن ثابت وجود دارد. روش اول، استفاده از SIP trunk ایجاد شده توسط سایر تأمین‌کنندگان سرویس VOIP است. از آنجایی‌که بسیاری از ارائه‌کنندگان سرویس VOIP، از طریق SIP trunk به شبکه‌ی تلفن ثابت دسترسی دارند، می‌توان از شبکه‌ی آن‌ها به عنوان یک درگاه برای ارتباط با تلفن ثابت، بهره‌مند شد.

پیشنهاد clearwater، استفاده از تأمین‌کنندہ‌ای به نام voxbone است. در مستندات clearwater، نحوه اتصال به voxbone با استفاده از رابطه‌ای برنامه‌نویسی اپلیکیشن آورده شده است. با اتصال به voxbone می‌توان از SIP trunk پیاده‌سازی شده‌ی آن برای ارتباط با تلفن ثابت استفاده کرد. voxbone، در بیش از ۹۰ کشور دنیا سرویس ارائه می‌کند؛ اما از آنجایی‌که ایران، جزو کشورهای تحت پوشش این سرویس نیست، استفاده

^۱ سرویزه‌ی عبارت Integrated Service Digital Network و به معنای شبکه‌ی سرویس یکپارچه‌ی دیجیتال

از این روش مقدور نمی‌باشد [۲۱][۵].

با وجود اینکه voxbone، تنها ارائه‌کننده سرویس VOIP است که توسط توسعه‌دهندگان تیم clearwater،^۲ مورد آزمایش و ارزیابی قرار گرفته است، اما بدیهی است که می‌توان از سایر ارائه‌کنندگان نیز برای این کار استفاده کرد. به عنوان مثال، این امکان وجود دارد که با اتصال IMS به سرویس VOIP دانشگاه صنعتی اصفهان، از SIP trunk پیاده‌سازی شده توسط این سرویس استفاده کرد.

پیاده‌سازی SIP trunk

SIP trunk های تجاری به نام درگاه وویپ (VOIP) وجود دارند و می‌توان از آن‌ها استفاده کرد. شرکت Sangoma، یکی از تولیدکنندگان بزرگ این درگاه‌ها است و سرویس VOIP دانشگاه صنعتی اصفهان نیز، از محصولات همین شرکت برای ایجاد SIP trunk استفاده می‌کند. متأسفانه، این درگاه‌ها، قیمت بالایی (چند میلیون تومان) دارند. لذا، راه حل جایگزین، طراحی و ساخت SIP trunk است.

طراحی SIP trunk در دو بخش نرم‌افزاری و سخت‌افزاری صورت می‌گیرد. در پروژه‌ی کارشناسی دانشجویان^۳ دانشکده‌ی برق و کامپیوتر تحت عنوان "طراحی و پیاده‌سازی رابط شبکه‌ی VOIP و PSTN"، بخش نرم‌افزاری و سخت‌افزاری SIP trunk طراحی و ساخته شده است. بخش نرم‌افزاری شامل سه بخش است:

- پیاده‌سازی یک نرم‌افزار برای ایجاد تماس تلفنی

- توسعه‌ی نرم‌افزار برای دریافت و پاسخ به تماس تلفنی

- نرم‌افزار ارسال و دریافت صدا

سخت‌افزار SIP trunk نیز به طور کلی شامل پنج قسمت است:

- مدار Off Hook^۳ تلفن

- مدار تشخیص زنگ تلفن

- مدار دریافت و ارسال صوت

- مدار راهانداز USB

- مدار مربوط به میکروکنترلر

با استفاده از سخت‌افزار و نرم‌افزار موردنظر، می‌توان شبکه‌ی VOIP را به شبکه‌ی تلفن ثابت متصل کرد. برای اتصال SIP trunk، باید آدرس IP اینترفیسی که آن را به سرور مربوطه متصل می‌کند، در پیکربندی سرور وارد

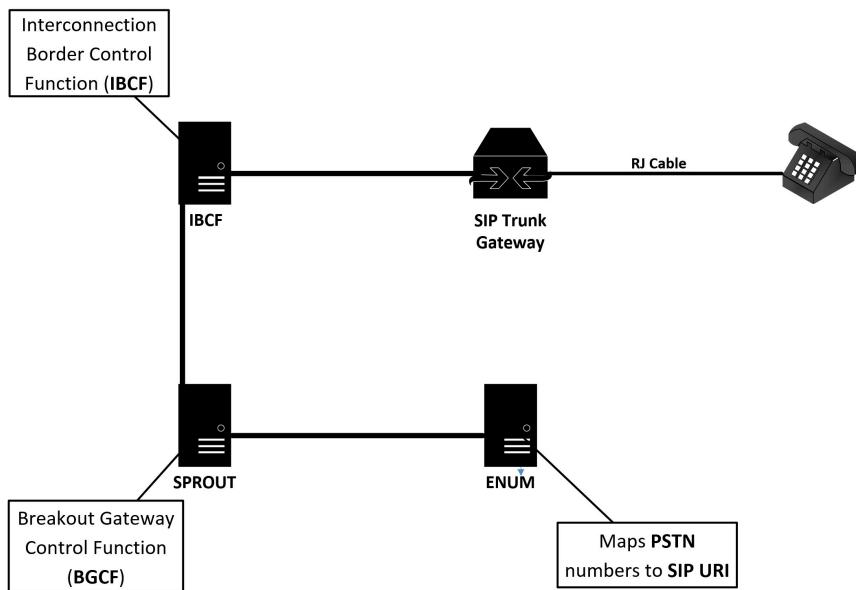
² بخش نرم‌افزاری توسعه آقای حسین خلیلیان و بخش سخت‌افزاری توسعه آقای احمد رضا بدیهی انجام شده است.

³ زمانی که خط تلفن اشتغال است.

شود. در بعضی از سیستم‌ها، نیاز است که شماره‌ی پورت مورد استفاده نیز مشخص شود. برای اطلاعات بیشتر، به گزارش پروژه‌ی مذکور مراجعه شود.

پ. ۲. اتصال به SIP trunk

برای ارتباط SIP trunk با clearwater، باید المان موردنیاز برای این کار، به معماری clearwater اضافه شود و در پیکربندی سایر المان‌های مربوطه نیز، تغییراتی ایجاد شود. المان IBCF، رابط بین پیاده‌سازی clearwater و SIP trunk است. المان IBCF توسط clearwater پیاده‌سازی شده است، بسیاری از کارکردهای موردنیاز را تأمین می‌کند؛ اماً بعضی کارکردها مانند پنهان‌سازی توپولوژی شبکه‌ی داخلی IMS، هنوز پیاده‌سازی نشده‌اند. برای استفاده از IBCF، حداقل باید یک نود IBCF را نصب و پیکربندی کرد. سپس باید المان ENUM را برای نگاشت شماره‌ی تلفن، پیکربندی کرد. سپس باید پیکربندی مربوط به مسیریابی BGCF را که در نود Sprout قرار دارد، انجام داد. شکل پ. ۱، نحوه اتصال تلفن ثابت به شبکه‌ی IMS را نشان می‌دهد. مطالب این بخش، بر اساس مستندات پروژه‌ی clearwater [۵] نوشته شده‌اند.



شکل پ. ۱: نحوه ارتباط SIP trunk با IMS و اتصال به PSTN

۱۰.۲ نصب و پیکربندی IBCF

نصب IBCF، مانند نصب دستی نود Bono می‌باشد. بنابراین، ابتدا باید نود Bono را نصب کرد. روش نصب نودهای معماری clearwater، در مستندات این پروژه و در قسمت نصب به صورت دستی بر روی سیستم‌های آورده شده است. در این قسمت، نصب نود Bono به صورت جداگانه بیان شده است.

پیش‌نیازها

ابتدا باید نسخه‌ی سرور ۶۴ بیت سیستم عامل اوبونتو ۱۴.۰۴ را بر روی یک ماشین نصب کرد. حداقل سخت‌افزار مورد نیاز برای راهاندازی این نود، مانند سخت‌افزار لازم برای پیاده‌سازی‌های یکپارچه و در مقیاس کوچک (بخش ۲.۲.۵) است. همچنین، نیاز است که یک آدرس IP عمومی^۴ و یک آدرس IP خصوصی به ماشین اختصاص داده شود.

اضافه کردن مخازن clearwater

برای ایجاد فایل موردنظر، دستور زیر را وارد کنید و عبارت /binary/stable را به فایل موردنظر اضافه کنید.

```
sudo nano /etc/apt/source.list.d/clearwater.list
```

سپس با استفاده از دستور زیر، کلیدهای امضای مورد استفاده توسعه سرور clearwater را نصب کنید.

```
curl -L http://repo.cw-ngv.com/repo_key | sudo apt-key add -
```

با استفاده از دستور زیر، اثر انگشت^۵ را چک کنید.

```
sudo apt-key finger
```

خروجی دستور فوق، باید شامل عبارت زیر باشد:

```
pub      4096R/22B97904 2013-04-30
```

```
Key fingerprint = 9213 4604 DE32 7DF7 FEB7 2026 111D BE47 22B9 7904
```

```
uid      Project Clearwater Maintainers <maintainers@projectclearwater.org>
```

```
sub      4096R/46EC5B7F 2013-04-30
```

پس از انجام موفقیت‌آمیز مراحل فوق، دستور زیر را اجرا کنید.

```
sudo apt-get update
```

^۴ به جای IP آدرس عمومی، می‌توان یک آدرس IP در شبکه‌ی محلی خود به ماشین اختصاص داد، به طوری که از تمام نقاط شبکه‌ی داخلی می‌توان به آن آدرس IP دسترسی داشت. در این صورت، آدرس IP خصوصی که برای ارتباط نودهای clearwater مورد استفاده قرار می‌گیرد، می‌تواند آدرس پشت NAT در همان شبکه محو باشد.

^۵ Fingerprint

تنظیمات دیوار آتش

نودهای clearwater، از طریق پروتکل IP، با یکدیگر ارتباط برقرار می‌کنند. برای این ارتباط، از پورت‌های مختلفی استفاده می‌شود. لازم است که در تنظیمات دیوار آتش، اجازی استفاده از این پورت‌ها داده شود. برای برقراری ارتباط SSH، لازم است که پورت 22 TCP/22 باز باشد. برای برقراری سیگنالینگ‌های STUN، لازم است که پورت 3478 هم برای TCP و هم برای UDP باز باشد. برای سیگنالینگ‌های SIP، پورت‌های TCP/5060 و UDP/5060 مورد استفاده قرار می‌گیرد. همچنین، برای فورواردینگ RTP، نیاز است که پورت‌های 32768-65535 UDP باز باشند. برای بازکردن پورت‌های مذکور، دستورات زیر را اجرا کنید.

```
sudo ufw allow 22,3478,5060,5062/tcp
```

```
sudo ufw allow 3478,5060,32768:65535/udp
```

پیکربندی محلی

قبل از نصب نود، نیاز است که پیکربندی اولیه برای ارتباط با سایر نودها انجام شود. برای این کار، با استفاده از دستور شماره 1، فایل پیکربندی محلی (local_config) را ایجاد کنید. در صورت عدم وجود پوششی موردنظر، ابتدا دستورات 2 و 3 را اجرا کرده و سپس دستور 1 را اجرا کنید.

1. sudo nano /etc/clearwater/local_config

2. cd /etc

3. mkdir clearwater

پس از ایجاد فایل موردنظر، اطلاعات زیر را وارد آن کنید. در عبارت زیر، به جای <privateIP>، آدرس خصوصی ماشین و به جای publicIP، آدرس IP عمومی ماشین را قرار دهید. به جای عبارت hostname نیز آدرس IP عمومی را قرار دهید. به جای عبارت <comma separated list of private IPs>، باید آدرس IP خصوصی تمام نودهای معماری clearwater را قرار دهید و بین این آدرس‌ها، به وسیله‌ی علامت ','، فاصله ایجاد شود. به عنوان مثال، عبارت 10.0.0.1,10.0.0.2,10.0.0.3,10.0.0.4,10.0.0.5 می‌تواند یک نمونه از مقداردهی برای etcd_cluster باشد.

local_ip=<privateIP>

public_ip=<publicIP>

public_hostname=<hostname>

etcd_cluster="<comma separated list of private IPs>"

نصب Bono

برای نصب نود Bono، دستورات زیر را اجرا کنید:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install Bono restund -yes
```

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install clearwater-management -yes
```

پیکربندی IBCF

پس از نصب نود Bono، باید IBCF را پیکربندی کرد تا SIP trunk یا SIP های^۵ مورد استفاده را به رسمیت بشناسد. برای این کار، باید فایل user_settings را ویرایش کنید. این فایل، در پوشه‌ی /etc/clearwater قرار دارد. در صورت موجود نبودن این فایل، آن را ایجاد کنید. با اجرای دستور زیر، فایل موردنظر باز می‌شود(در صورت موجود نبودن، ایجاد می‌شود).

```
sudo nano /etc/clearwater/user_settings
```

سپس، آدرس IP مربوط به SIP trunk یا SIP trunk را به عنوان trusted_peers معرفی کنید. برای این کار، به جای <trunk IP address>، آدرس IP مربوطه را قرار دهید.

```
trusted_peers="<trunk 1 IP address>,<trunk 2 IP address>, ..."
```

پس از انجام این پیکربندی، نود Bono را با دستور زیر، دوباره شروع^۶ کنید.

```
service stop bono
```

۲۰.۲ پیکربندی ENUM

سیستم ENUM، برای نگاشت شماره تلفن‌های PSTN به SIP URI^۷ مورد استفاده قرار می‌گیرد. این کار، با استفاده از بایگانی^۸ NAPTR^۹ سیستم نام دامنه صورت می‌گیرد. علت استفاده از NAPTR، پشتیبانی Sprout از این نوع بایگانی است. بازه‌ی شماره تلفن‌هایی که قرار است از طریق SIP trunk مسیریابی شوند، باید در پیکربندی ENUM وارد شوند. از آنجایی که ENUM، پیکربندی‌ها و نگاشت‌های مخصوص به خود را دارد، نیاز است که ابتدا بخش ENUM Guid در [۲۲] مطالعه شود. پس از کسب اطلاعات لازم در مورد ENUM، می‌توان نحوه‌ی نگاشت شماره تلفن‌ها را وارد پیکربندی این سیستم کرد.

^۶می‌توان هم‌زمان از چند SIP trunk استفاده کرد
Restart^۷

^۸سروازه‌ی عبارت Uniform Resource Identifier و روشنی برای نمایش آدرس‌های SIP است. این آدرس‌ها به شکل user@domain.tld می‌باشند. به عنوان مثال، SIP URI یک 6505551234@example.com است.

^۹Record^۹
^{۱۰}کوتاه‌شده‌ی عبارت Name Authority Pointer است. NAPTR گونه‌ای از بایگانی منابع در سیستم نام دامنه اینترنت است. بایگانی‌های NAPTR بیشتر در اپلیکیشن‌های VOIP مورد استفاده قرار می‌گیرند و عموماً کار نگاشت آدرس سرورها و کاربران را در پروتکل SIP انجام می‌دهند.

۳۰.۲ پیکربندی BGCF

فایل پیکربندی BGCF به نام bgcf.json در پوشه‌ی /etc/clearwater قرار دارد. ابتدا باید با دستور شماره ۴، یک کپی از این فایل را ایجاد کرده و سپس فایل کپی‌شده را ویرایش کنید. فایل کپی‌شده در پوشه‌ی clearwater-config-manager قرار می‌گیرد. برای ویرایش فایل مورد نظر، از دستورات ۵ و ۶ استفاده کنید. در دستور شماره ۶، بهجای عبارت <user>، نام کاربری خود در سیستم عامل را قرار دهید.

4. clearwater-config downlaod bgcf_json
5. cd /home/<user>/clearwater-config-manager/<user>
6. sudo nano bgcf.json

فایل bgcf.json دو نوع نگاشت را ذخیره می‌کند. در نوع اول، آدرس IP مربوط به SIP trunk و نام‌های IBCF URI های IBCF نگاشت می‌کند. در نوع دوّم، شماره‌های تلفن را به SIP URI های SIP تبدیل می‌کند. برای هر نام دامنه، IP آدرس SIP trunk و شماره تلفن، تنها یک مسیر می‌تواند قرار داده شود. اگر برای هر یک از موارد گفته شده، بیش از یک مسیر در درنظر گرفته شود، فقط اولین مسیر مورد استفاده قرار می‌گیرد. فایل bgcf.json، به فرمت JSON است. عبارات زیر، نمونه‌ای از جدول مسیرها می‌باشد که به فرمت JSON نوشته شده است.

{

```

"routes" : [
    {
        "name" : "<route 1 descriptive name>",
        "domain" : "<SIP trunk IP address or domain name>",
        "route" : ["<IBCF SIP URI>"]
    },
    {
        "name" : "<route 2 descriptive name>",
        "domain" : "<SIP trunk IP address or domain name>",
        "route" : ["<IBCF SIP URI>", "<IBCF SIP URI>"]
    },
    {
        "name" : "<route 3 descriptive name>",
        "number" : "<Telephone number>",
        "route" : ["<IBCF SIP URI>", "<IBCF SIP URI>"]
    }
]
```

```
    ]  
}
```

بعد از اعمال تغییرات در این فایل، دستور زیر را اجرا کنید تا تغییرات لازم در نود Sprout که المان BGCF در آن قرار دارد، اعمال شود.

```
cw-config upload bgcf_json
```

مراجع

- [1] C. Cox, *An Introduction To LTE*, 2nd edition. Chichester: John Wiley & Sons, 2014.
- [2] M. Wuthnow, M. Stafford and J. Shih, *IMS A New Model For Blending Applications*. CRC Press, 2010.
- [3] G. Camarillo and M.A Garcia, *The 3G IP Multimedia Subsystem(IMS)*, 3rd editoin. Chichester: John Wiley & Sons, 2008.
- [4] <http://www.openmobilealliance.org/release/PoC>
- [5] <http://www.projectclearwater.org>
- [6] <https://www.metaswitch.com>
- [7] <https://www.openimscore.com>
- [8] <http://cassandra.apache.org>
- [9] https://clearwater.readthedocs.io/en/stable/All_in_one_AMI_Installation.html
- [10] <https://github.com/Metaswitch/clearwater-vm-images/blob/master/ubuntu.ovf/ubuntu-server.seed>
- [11] https://clearwater.readthedocs.io/en/stable/All_in_one_Images.html
- [12] <https://help.github.com/articles/connecting-to-github-with-ssh/>
- [13] J.F Kurose and K.W Ross, *Computer Networkig A Top-Down Approach*, 6th edition. Pearson, 2012.
- [14] <http://www.pjsip.org>
- [15] <https://www.asterisk.org>
- [16] [RFC 3665] A. Johnston, S. Donovan, C. Cunningham and K. Summers, "Session Initiation Protocol (SIP) Basic Call Flow Example," RFC 3665, December, 2003.
- [17] <https://www.metaswitch.com/products/core-network/perimeta-sbc>
- [18] F. Gringoli, P. Patras, C. Donato, P. Serrano and Y. Grunenberger, "Performance Assessment of Open Software Platforms for 5G Prototyping," *IEEE Wireless Communications Magazine*, 2018.
- [19] <http://www.openairinterface.org>
- [20] <https://github.com/srsLTE/srsLTE>
- [21] <https://www.voxbone.com>
- [22] <https://clearwater.readthedocs.io/en/stable/ENUM.html>