Hacia una red de voz eficiente y virtualizada

Jesús Parejo Jimenez
Ingeniería de Telecomunicación
Telemática

Nombre Consultor/a
José López Vicario

Fecha Entrega 06/06/2021

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Hacia una red de voz eficiente y virtualizada*
Nombre del autor:	*Jesús Parejo Jimenez*
Nombre del consultor/a:	*José López Vicario*
Nombre del PRA:	*José López Vicario*
Fecha de entrega (mm/aaaa):	06/2021
Titulación:	*Ingeniería en Telecomunicación*
Área del Trabajo Final:	*Telemática*
Idioma del trabajo:	*Castellano*
Palabras clave	*VNF SDN Contenedores*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

En el documento se describe una primera red que posibilitó la comunicación entre dos dispositivos y cómo mediante la introducción de elementos de conmutación, a día de hoy, existe la posibilidad de comunicarse de manera prácticamente instantánea con cualquier persona del mundo.

El servicio de voz en la actualidad está en depreciación ya que prácticamente se "regala" en los paquetes de voz + datos + TV por las compañías de telecomunicaciones, pero para que esto sea posible la red que soporta dichos servicios ha tenido que evolucionar, cambiar la metodología empleada y hacer un uso eficiente de los recursos.

El análisis pretende explicar cómo ha sido posible asumir un crecimiento exponencial del servicio pasando de una red de voz donde para el establecimiento de la comunicación era necesario tener canales dedicados, metodología Circuit Switching (CS) con un alto consumo de recursos, lo que conlleva una gran inversión en infraestructura, a una red eficaz donde gracias a la implantación de las redes IP, y la metodología Packet Switching (PS), se ha podido absorber el gran volumen de tráfico generado. Además, se desarrolla su funcionamiento, arquitectura y protocolos.

La complejidad de la red obliga a dar un paso más y una vez mejorada la conmutación se analiza el uso de la virtualización en las redes de telecomunicaciones, se explica cómo se ha implementado en la red de voz, VNF y SDN, mediante servidores de altas prestaciones, reduciendo así costes, mejorando la seguridad y simplificando tareas, entre otros.

Finalmente se realiza una introducción y análisis sobre contenedores, Kubernetes y Dockers, un sistema que aún no se ha llevado a las redes de voz actuales en España,

estudiando su aplicabilidad y permitiendo así la automatización del despliegue, ajuste de escala y manejo de aplicaciones.

Abstract (in English, 250 words or less):

The document describes a first network that enabled communication between two devices and how by introducing switching elements, today, there is the possibility of communicating practically instantly with anyone in the world.

Voice service is currently in depreciation since it is practically "given away" in voice + data + TV packages by telecommunications companies, but for this to be possible the network that supports these services has had to evolve, change the methodology used and make efficient use of resources.

The analysis aims to explain how it has been possible to assume an exponential growth of the service from a voice network where, for the establishment of communication, it was necessary to have dedicated channels, Circuit Switching (CS) methodology with a high consumption of resources, which entails a great investment in infrastructure, to an efficient network where, thanks to the implementation of IP networks, and the Packet Switching (PS) methodology, it has been possible to absorb the large volume of traffic generated. In addition, its operation, architecture and protocols are developed.

The complexity of the network forces us to go one step further and once the switching has been improved, the use of virtualization in telecommunications networks is analyzed, it is explained how it has been implemented in the voice, VNF and SDN network, through high-end servers. benefits, thus reducing costs, improving safety and simplifying tasks, among others.

Finally, an introduction and analysis is made on containers, Kubernetes and Dockers, a system that has not yet been brought to the current voice networks in Spain, studying its applicability and thus allowing the automation of deployment, scaling and application management.

Contenido

1. Introducción 7
1.1 Contexto y justificación del Trabajo 7
1.2 Objetivos del Trabajo 7
1.3 Enfoque y método seguido 8
1.4 Planificación del Trabajo 8
1.5 Breve sumario de productos obtenidos 9
1.6 Breve descripción de los otros capítulos de la memoria 9
2. Estado del arte 11
3. Evolución de la red telefónica fija y móvil 15
3.1 Circuit Switching (CS) vs Packet Switching (PS) 15
3.2 PSTN 17
3.2.1 Arquitectura 17
3.2.2 Señalización SS7 18
3.3 Red fija 20
3.3.1 RTC - RTB 20
3.3.2 RDI/RDSI 20
3.3.3 xDSL 22
3.3.4 FTTx 25
3.4 Red móvil 29
3.4.1 PLMN 29
3.4.2 1G 29
3.4.3 2G 30
3.4.4 2,5G 31
3.4.5 3G 32
3.4.6 3,5G 33
3.4.7 4G 34
3.4.8 5G 35
4. NGN – IMS 38
4.1 Red de VOIP 38
4.2 NGN 38
4.3 IMS 38
4.3.1 Arquitectura IMS 39
4.3.2 Acceso 40
4.3.3 Componentes de señalización 40
4.3.4 Componentes de interconexión 41

4.3.5 Componentes relacionados con el aprovisionamiento de aplicaciones y servicios... 42
4.4 Señalización SIP ... 42
4.4.1 Mensajes ... 43
5. Virtualización ... 45
5.1 ¿Qué es? ... 45
5.2 ¿Cómo funciona? ... 46
5.3 Tipos de virtualización ... 46
5.4 VNF y SDN ... 47
5.4.1 VNF ... 47
5.4.2 SDN ... 48
5.4.3 NFV-VNF y su relación con SDN ... 52
5.5 Contenedores ... 54
5.5.1 Conceptos ... 55
5.5.2 Organización de contenedores ... 56
5.5.3 Beneficios de utilizar contenedores (pto. vista empresarial empresarial) ... 64
6. Laboratorio ... 66
6.1 LEARNING SWITCH – MININET y POX ... 66
6.2 ESCENARIO NFV – MANO CON OPENSTACK Y OSM (FALLIDO) ... 69
6.3 PRUEBA ESCENARIO LLAMADA SIP CON ASTERISK EN ENTORNO DOCKERIZADO Y CONECTIVIDAD ENTRE CONTENEDORES ... 73
6.3.1 Llamada SIP con Asterisk ... 73
6.3.2 Conectividad entre contenedores ... 78
7. Conclusiones ... 83
8. Glosario ... 84
9. Bibliografía ... 85

1. Introducción

1.1 Contexto y justificación del Trabajo

El proceso de digitalización y la necesidad de manejar una cantidad ingente de datos ha llevado a que las redes de telecomunicaciones se hayan sometido constantemente a cambios, de manera que sean más eficientes en el uso de los recursos siendo capaces de manejar más información y más rápido.

La red de voz en particular ofrece un servicio básico que cuenta ya con varios siglos de desarrollo o evolución sobre sus espaldas: comenzando con una red con canales dedicados para cada comunicación, con una arquitectura llena de centrales de conmutación, a un núcleo de red con unos pocos nodos capaces de proporcionar el servicio a millones de persones. Este cambio no es trivial y tiene una gran complejidad operativa donde se requiere de un gran esfuerzo para realizar dicha transición y hacer que convivan las nuevas redes con las antiguas, legacy.

Un gran salto que ha facilitado el despliegue de la red es la virtualización, aportando una capa de abstracción sobre el hardware donde se monta cada servidor o función de red, pero existe la posibilidad de dar un paso más virtualizando también el software, el sistema operativo, empleando para ello Dockers o Kubernetes y el uso de contenedores pudiendo implementar y ajustar la escala de aplicaciones rápidamente en cualquier entorno con la certeza de saber que el código se ejecutará.

Por lo que, debido a la envergadura de las redes de telecomunicaciones se hace necesario sintetizar y explicar cuál ha sido su evolución o transformación tanto en la parte de la telefonía fija como de la telefonía móvil permitiendo tener una visión global e histórica: desde la RTB y PSTN, explicando cada uno de los accesos (RDSI, xDSL, FTTx, etc.), pasando por la PLMN, y cada una de sus generaciones (2G, 3G, 4G, etc.), hasta llegar a la red convergente NGN y su proceso de virtualización, VFNs y SDN; viendo el funcionamiento y composición de cada una de ellas. Además, se explica una tecnología más reciente, los contenedores, que forma parte de los próximos pasos que se van a seguir para poder reducir costes y tener una red eficiente y moldeable que permita adaptarse a las necesidades de los servicios existentes y que están por venir con la entrada en escena del 5G.

1.2 Objetivos del Trabajo

El proyecto está compuesto por los siguientes objetivos específicos:

- Mostrar las metodologías en las que se basan las redes de telecomunicaciones: Circuit Switching y Packet Switching; resaltando sus diferencias y las ventajas e inconvenientes que ofrece cada una.

- Mostrar la evolución de la red de voz fija y cada uno de sus diferentes accesos: RTB, RDSI, xDSL y FTTH junto al despliegue de la malla fotónica. Descripción de la arquitectura, nodos de red, protocolos e interconexión con la red PSTN.
- Mostrar la evolución de la red móvil y el paso por cada una de las generaciones permitiendo comprender las ventajas que ofrece la implantación de la nueva tecnología 5G, sus ventajas y usos.
- Mostrar el paso o transición hacía la red IP que surge como evolución de la actual infraestructura de redes de telecomunicación y acceso telefónico con el objetivo de lograr la convergencia tecnológica de los servicios multimedia (voz, datos y todo tipo de medios como video), NGN-IMS.
- Describir la aplicabilidad de la virtualización en las redes de telecomunicaciones, VNF y SDN, las ventajas que aporta, su integración y las mejoras que ofrece.
- Analizar la integración de una nueva capa de virtualización, sobre el Sistema Operativo, mediante el uso de contenedores, Kubernetes y Dockers en el core de voz NGN para el despliegue de aplicaciones o funciones de red de manera rápida y eficiente. Realizar pruebas de escenarios de llamadas SIP mediante simulaciones en laboratorio virtual (SIPp, Kamailio, Asterisk, etc.).

1.3 Enfoque y método seguido

En el despliegue o formación de redes de telecomunicaciones, en particular la red de voz, se puede mejorar el funcionamiento y la eficiencia modificando configuraciones en los nodos que conforman la red, utilizando protocolos de comunicación más eficaces y menos complejos de modo que no sea necesario transmitir tantos mensajes de señalización o información de control en el establecimiento de una sesión y también tocando una parte más transversal a todos estos nodos como es el hardware y software. Todas estas medidas no son excluyentes entre sí, pero se ha decidido abordar el tema de eficiencia de la red desde este último punto de vista ya que de este modo la mejora afecta a cada máquina que compone la red provocando una mejoría significativa en el funcionamiento.

La virtualización ha cogido gran fuerza en el sector de la informática, pero es necesario estudiar su aplicabilidad a cada una de las redes de telecomunicación. La virtualización a nivel de hardware ya se ha llevado o se está llevando a cabo, pero no está tan clara su aplicación a nivel de SO mediante el uso de contenedores.

1.4 Planificación del Trabajo

Plan de atividades para elaboración del TFM	
Actividades	**Fecha estimada**
Actividad 1. Presentación de propuesta de elaboración del TFM.	3ª semana de Febrero
Actividad 2. Elaboración Plan de Proyecto TFM.	4ª semana de Febrero
Actividad 3. Búsqueda de bibliografía y fuentes, recopilación de información, normativa y RFCs sobre el modelo de Circuit Switching y Paquet Switching.	1ª semana de Marzo
Actividad 4. Estudio del contexto histórico. Evolución redes de acceso fija y móvil: arquitectura, funcionamiento, protocolos, etc.	2º y 3ª semana de Marzo
Actividad 5. Búsqueda de bibliografía y fuentes sobre virtualización, VNFs y SDN.	4ª semana de Marzo y primera de Abril
Actividad 6. Síntesis.	2ª semana de Abril
Actividad 7. Estudio sobre aplicación de contenedores sobre el core de red NGN	2ª y 3ª semana de Abril
Actividad 8. Crear laboratorio virtual para realizar pruebas con Kubernetes/Docker mediannte simualadores (SIPp, Kamailo, vSDNEmul, etc.)	4ª semana de Abril y 1ª semana de Mayo
Actividad 9. Realizar pruebas de escenarios SIP	2º semana de Mayo
Actividad 10. Depurar y alinear la memoria con los requisitos exigidos.	2ª quincena de Mayo
Actividad 11. Preparar presentación y defensa TFM.	1ª quincena de Junio
Actividad 12. Realizar defensa del TFM.	3ª semana de Junio

1.5 Breve sumario de productos obtenidos

- Síntesis de la evolución y el proceso de transformación de las redes de voz, tanto fija como móvil, hacía una red convergente, NGN-IMS.
- Comprender como la tecnología SDN puede mejorar o resolver las desventajas de los sistemas actuales.
- Analizar la relación entre SDN y VNF.
- Escenario de simulación de los conceptos explicados de SDN y NFV.
- Simulación en laboratorio virtual de un escenario de llamada VoIP a través de una PBx (emulando el comportamiento de una red NGN-IMS), con Asterisk, y aplicando la tecnología de virtualización mediante el uso de contenedores con Docker.

1.6 Breve descripción de los otros capítulos de la memoria

Se ha pensado en estructurar el TFM en los siguientes apartados:

1. Introducción

   Contiene una descripción de los objetivos y planificación del TFM.

2. Estado del arte

Donde se comenta la tecnología actual y se mencionan los proyectos o estudios relacionados y la contribución de este trabajo, que es el de poner foco en la evolución de las redes buscado siempre la eficiencia mediante el uso de tecnologías como la virtualización (VNF, SDN y contenedores).

3. Evolución de la red telefónica fija y móvil

En este apartado se realiza una descripción de la metodología de conmutación CS y PS describiendo su funcionamiento y realizando una comparativa entre ambas junto con una introducción sobre el funcionamiento de la PSTN, su arquitectura y el protocolo de señalización que emplean los equipos que conforman la red para comunicarse entre ellos.

Además, detalla la evolución de las redes fijas, desde la RTB hasta FTTx, y la móvil, desde el 1G al actual 5G.

4. NGN

Este apartado describe el concepto de red NGN y como aplicado sobre IMS se ha podido lograr la convergencia tecnológica de los servicios multimedia (voz, datos y todo tipo de medios como video) encapsulándolos en paquetes IP.

5. Virtualización

Descripción de concepto de virtualización, ventajas e implementación de la virtualización en las funciones de red. Definición de los servicios de red como un conjunto de funciones de red virtuales que pueden ser implementadas en software y ejecutadas dentro de máquinas virtuales sobre arquitecturas de propósito general con servidores de altas prestaciones. Se explica la aplicación de la virtualización “tradicional” mediante NFVI-VNF, la virtualización de redes mediante SDN y una tecnología más actual que va un paso más allá en la virtualización, los contenedores.

6. Laboratorio

Se lleva a cabo el desarrollo de varios escenarios donde se aplican las distintas tecnologías de virtualización: VNF, SDN y contenedores. Esta última trata de simular el comportamiento o funcionamiento de una red NGN donde realizar llamadas de prueba VoIP a través del protocolo SIP.

7. Conclusiones

Este último apartado recoge las conclusiones extraídas tras el desarrollo de la memoria.

2. Estado del arte

El mundo de las telecomunicaciones posiblemente sea unos de los sectores que más ha cambiado y evolucionado en las últimas décadas. Las redes en general y la red de voz en particular ha sufrido una gran transformación desde que se inventó el teléfono en 1854, por el italiano Antonio Meucci, y se patentó en 1876 por Graham Bell.

La idea inicial consiste en un aparato que transmite sonido a través de un cable mediante señales eléctricas. El mecanismo empleado, a priori, puede parecer sencillo; mediante una membrana, junto a una pieza metálica, se producen unas vibraciones que provocan que el electroimán que está en el interior del aparato genere variaciones en el campo eléctrico y esta corriente es la que se transmite por el cable o enlace. El dispositivo cuenta con una segunda pieza, el auricular, el cual se coloca en el oído para poder escuchar la señal que llega de la otra persona u otro extremo de la comunicación.

Al principio esta señal se transmitía por cables de hierro los cuales eran muy rudimentarios generando muchas pérdidas y sufriendo muchas interferencias por lo que la calidad de la comunicación era bastante mala. Para que la comunicación pueda establecerse correctamente era necesario que los dispositivos que intervienen en ella estuvieran conectados directamente, es decir, si se quería hablar con 5 personas diferentes era necesario disponer de 5 dispositivos cada uno conectados directamente a cada uno de los dispositivos con los que se quisiera contactar, teniendo conexiones dedicadas, punto a punto. La complejidad es el mecanismo principal que impide el escalado eficiente.

La primera transmisión de voz se estableció a través de un circuito ring-down, es decir, no hay marcación de número, ambas personas descolgaban el aparato y ya estaba establecida la comunicación.

Es fácil darse cuenta de que el diseño no es escalable ya que las necesidades van más allá de poder contactar con un número limitado de personas; el objetivo es poder comunicarse con cualquier persona del mundo en cualquier momento. Si se quisiera interconectar un numero N de dispositivos sería necesario utilizar N*(N-1) / 2 enlaces, es decir, **para interconectar 10 dispositivos** entre sí, similar a una red de malla, **serían necesarios un total de 45 enlaces**. Es imposible pensar en este diseño aplicado a la población de un país y menos aún aplicado a la población mundial.

Para solventar dicha casuística y hacer de esta una red escalable se crea la central local, que mediante un panel de conmutación permite conectar al llamante y llamado de modo que ya no sea necesario tener una línea o cable conectado directamente a cada teléfono con el que se quiera contactar. El empleo de paneles de conmutación da el pistoletazo de salida al diseño de la red de telefonía básica (RTB), donde para conectar un número N de dispositivos necesitamos N cables o líneas que conecten cada dispositivo a la central y esta ya se encarga de comunicar o contactar con el distante o llamado [Figura1]. Ahora el esquema quedaría del siguiente modo:

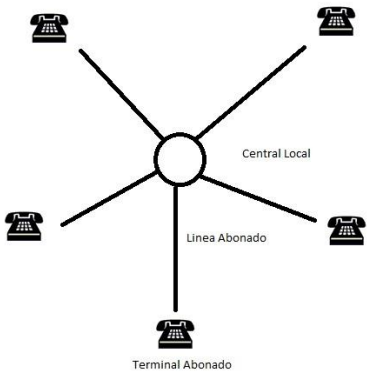


Figura1. Conmutación centralizada

No solo se han ido realizando mejoras en la topología de la red sino también en el cableado, solucionando así el problema de las interferencias, fue en 1881 cuando Graham Bell solicitó la patente del cable de par trenzado. Con la distribución de la energía eléctrica a los domicilios, el tendido eléctrico, se comprobó que intercambiando la posición de los conductores cada varios postes, transposición de conductores, se conseguía mitigar en gran medida las interferencias y de este modo se aplicó el trenzado para las comunicaciones de voz.

El uso de una central local permite la comunicación entre los diferentes abonados de una zona concreta, pero ¿qué ocurre si el dispositivo con el que se quiere contactar está en otra ciudad? ¿Y si está en otro país? Esta solución no era suficiente por lo que es necesario crear centrales de larga distancia que hagan las funciones de interconexión entre centrales locales. Aquí se va viendo cómo se crean dos capas o niveles diferentes donde existen unas centrales locales que conectan a los usuarios finales y un segundo nivel formado por las centrales troncales que permiten la comunicación entre centrales locales y comunicaciones de larga distancia [Figura2], así como la comunicación entre las distintas operadoras, a pesar de que la compañía de Graham Bell tenía prácticamente el monopolio del servicio de voz. Eran, en un principio, las zonas rurales donde no interesaba la inversión y donde otras operadoras veían su oportunidad.

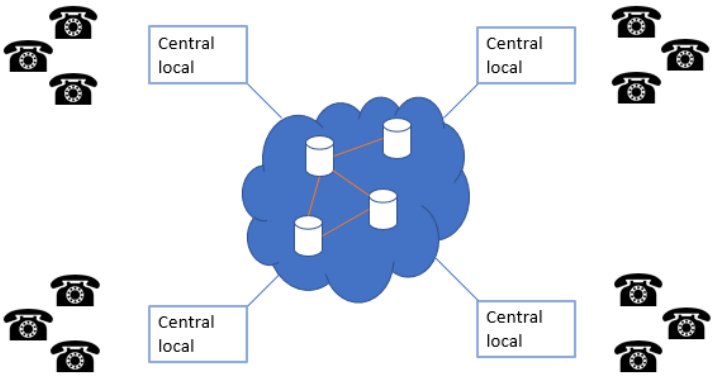


Figura2. Red telefónica jerárquica

Así se fue dando forma a lo que se conoce como Red Telefónica Pública Conmutada o lo que es lo mismo PSTN (Public Switched Telephone Network), una red de redes, definida

como un conjunto de elementos formado por los medios de transmisión y conmutación necesarios para enlazar dos equipos terminales a través de un circuito físico.

Para que todo esto sea posible es necesario que además de enviar por la red la señal analógica de voz también se envíe información de señalización que permita realizar tareas de control en el establecimiento de la comunicación.

A partir de este punto las redes de telecomunicaciones han ido evolucionando a pasos agigantados experimentando un continuo proceso de transformación.

El trabajo pretende ser una síntesis y estudio de las redes de telecomunicaciones, principalmente de la red de voz tanto fija como móvil, con el objetivo de poder ofrecer una visión general de estas, comprendiendo su arquitectura, funciones y funcionamiento; mostrando la complejidad que fueron adquiriendo conforme se añadían nuevos servicios y cómo se ha ido resolviendo el hándicap de contar con redes separadas o independientes lo cual lleva implícito unos elevados costes de operaciones y mantenimiento junto con tiempos muy elevados en el despliegue de los nodos.

Gran parte de la problemática expuesta, debido a la complejidad de la red, se solventa gracias a la tecnología de virtualización. Actualmente existen numerosos estudios sobre las ventajas y usos de la virtualización (añadiendo una capa entre el hardware y software de los equipos/servidores permitiendo aislarlos y hacer un uso eficiente de los recursos), VNF y SDN, y con carácter más actual o reciente ha llegado un gran avance tecnológico, los contenedores y el uso de Kubernetes y Docker.

Esta tecnología ofrece una multitud de posibilidades y ventajas ya que va un paso más allá en la virtualización introduciendo una nueva capa sobre el núcleo del sistema operativo y permitiendo que existan múltiples instancias aisladas de modo que en lugar de ofrecer, como se hace actualmente, un sistema operativo con el software o la aplicación, los suministradores entregan un paquete de código con todas sus dependencias (contenedor) que puede ser ejecutado en cualquier lugar.

En un principio puede parecer muy similar a la virtualización “tradicional” (del SW sobre el HW) pero los contenedores funcionan de forma muy diferente. Como el contenedor solo contiene la aplicación y los frameworks, librerías, etc. que utiliza, puede haber multitud de contenedores corriendo en un solo sistema operativo (con la virtualización “tradicional” cada aplicación o servidor corre bajo su propio sistema operativo y lo que se conseguía era ejecutar varios servidores lógicos en un mismo servidor físico) ya que ocupan generalmente poco espacio haciendo que sea muy eficiente y óptimo.

Actualmente existen herramientas que permiten el despliegue y gestión de contenedores como Kubernetes o Docker y se está trabajando sobre su aplicación en las redes de telecomunicaciones.

Existen algunos trabajos sobre SDN y NFV como:

- “Estudio de las tecnologías SDN y NFV” de Ibáñez García, Francisco José; Universitat Oberta de Catalunya; Mendoza Flores, Manuel Jesús; que comienza explicando:

*“**Durante un gran periodo de tiempo las redes prácticamente no han sufrido cambios** en sus arquitecturas. Pero, paralelamente, la tecnología ha cambiado y evolucionado con la aparición y consolidación de nuevas tendencias, como la aparición de una gran cantidad de dispositivos que se conectan a la red (Internet of Things -IoT), aplicaciones en tiempo real, el video streaming, la masificación de las redes sociales, cloud computing, nuevos servicios, …”*

Existe la idea de que en telecomunicaciones nos encontramos con una red muy estática y rígida que aparentemente no ha sufrido cambios. Con este trabajo se pretende dar una visión general de la evolución de las redes de voz fija y móvil y la convergencia hacía una red IP, desarrollando su funcionamiento y el proceso de transformación que se está realizando; **haciendo ver el constante proceso de transformación que ha sufrido** y el esfuerzo de las compañías por crear una red convergente, eficiente y virtualizada (NGN) donde gracias a la introducción de capas en los distintos niveles: hardware, red, S.O, etc. se ha conseguido crear tecnologías capaces de montar servicios completos en cuestión de horas, en lugar de meses e incluso años, de adaptarse de manera dinámica a las necesidades del momento y de utilizar los recursos HW de una manera eficiente reduciendo así costes y esfuerzos en la implantación y mantenimiento. Con la implantación de la red NGN-IMS y bajo el paradigma de la virtualización es posible tener una red flexible y eficiente capaz de adaptarse rápidamente y de ofrecer multitud de servicios multimedia con altas prestaciones.

- “Virtualització de xarxes” de Mañalich Dionisio, Ricard; Universitat Oberta de Catalunya; Mendoza Flores, Manuel Jesús; donde se explica la parte de virtualización relacionada con NFV y SDN.
- “Virtualització de servidors i escriptoris” de Riera Fargas, Gil; Universitat Oberta de Catalunya; Castillo Pedrosa, José Manuel
- “Avaluació de rendiment en plataformes virtualitzades” de Gómez Serrano, Alejandro; Universitat Oberta de Catalunya; Rodero Castro, Ivan
- “Despliegue de una plataforma de VDI en una universidad de Losada Dans”, Alén; Castillo Pedrosa, José Manuel

En los trabajos citados se explica una parte concreta de la virtualización como: la virtualización de escritorio, de servidores, SDN, su eficiencia, etc. Este trabajo pretende ir un paso más allá, mostrando la estrecha relación entre algunas técnicas de virtualización y cómo se apoyan unas sobre otras para sacarse el máximo partido, mostrando además su evolución, introduciendo el concepto de contenedores, el uso de Kubernetes y Docker y demostrar que es posible su aplicación en las redes de telecomunicaciones, detallando su funcionamiento y finalizando con un caso práctico en un laboratorio virtual sobre la que es y será la gran red de redes multimedia, NGN-IMS.

3. Evolución de la red telefónica fija y móvil

Merece la pena comenzar explicando las metodologías empleadas en las redes para conectar los dispositivos ya que es en gran parte la base del funcionamiento de estas, la conmutación de circuitos o CS (Circuit Switching) y la conmutación de paquetes o PS (Packet Switching) y sus diferencias.

3.1 Circuit Switching (CS) vs Packet Switching (PS)

La transmisión de información ya sea de señales analógicas, como hemos visto hasta ahora, o digitales, como se verá más adelante, se resume en la transmisión de la señal a través de una red de nodos conmutados que no están relacionados con el contenido de los datos y cuyo objetivo es conectar unos dispositivos finales (teléfonos, PCs, etc.), en nuestro caso abonados, unos con otros, estos nodos o equipos de conmutación forman lo que se denomina red de telecomunicaciones.

Para la PSTN interesa entender el funcionamiento de **Circuit Switching (CS)**. En este caso, cuando se va a establecer una comunicación, conversación, **se utiliza un circuito dedicado, exclusivo y temporal para dicha comunicación entre terminales o estaciones base, que se mantiene reservada y es de uso exclusivo mientras dure la comunicación (Figura3).** Existen tres fases durante este tipo de conmutación: establecimiento de circuito, transferencia de datos y desconexión del circuito.

A continuación, se muestra gráficamente cómo sería el proceso:

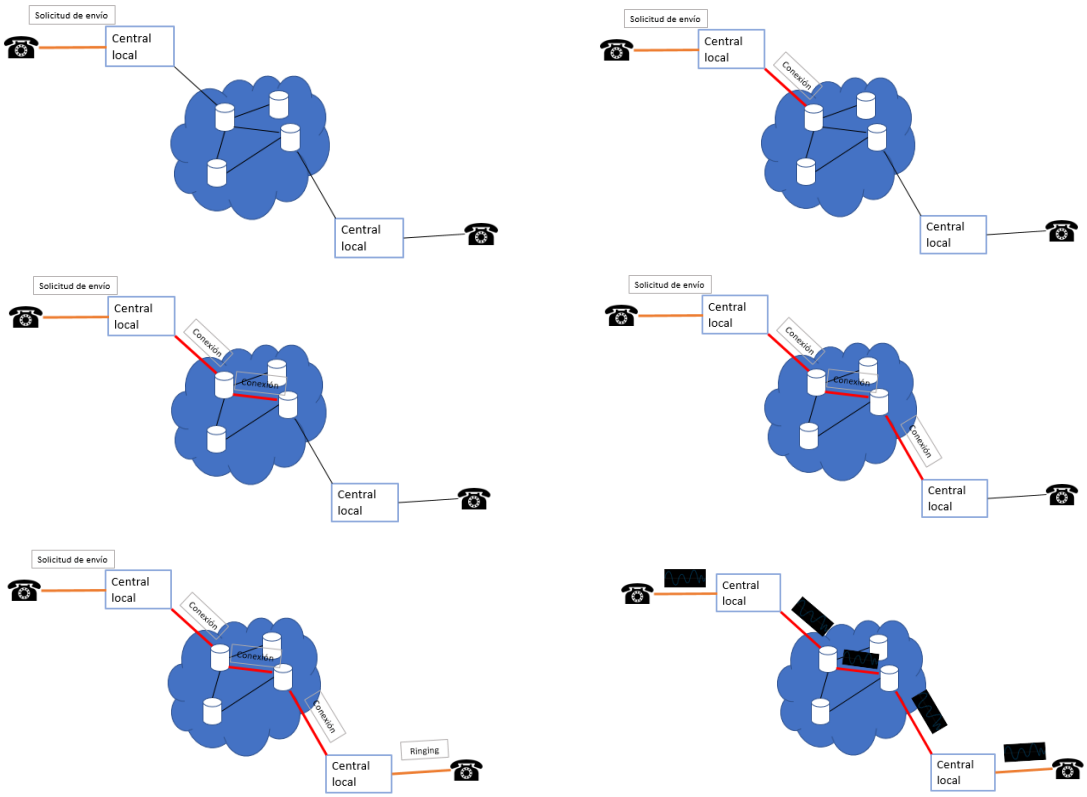


Figura3.Ejemplo establecimiento llamada en CS

Packet Switching (PS) se define como un método que permite agrupar datos y transmitirlos en paquetes a través de una red digital. Dichos paquetes están compuestos por unos encabezados que contienen información de control, que indica la ruta a seguir a lo largo de la red hasta el destino del paquete, y el payload o carga útil donde se encuentra la información que se desea transmitir. La conmutación de paquetes permite la entrega de flujos de datos de velocidad de bits variable, realizados como secuencias de paquetes, a través de una red informática que asigna los recursos de transmisión según sea necesario utilizando técnicas de multiplexación estadística o asignación dinámica de ancho de banda. A medida que atraviesan hardware de red, como conmutadores y enrutadores, los paquetes se reciben, se almacenan en buffer, se ponen en cola y se retransmiten (almacenan y reenvían), generando una latencia y un rendimiento variables según la capacidad del enlace y la carga de tráfico en la red, es lo que en Internet se conoce como Best Effort (Figura4); esto a priori puede parecer negativo, pero hay mecanismos que permiten subsanar estos inconvenientes. Lo importante es que los paquetes se transmiten por la red y es solo durante ese instante cuando se están consumiendo recursos o ancho de banda en cada enlace por donde pasa el paquete.

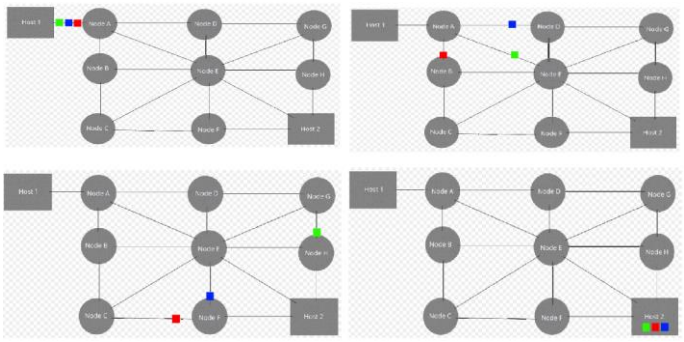


Figura4.Ejemplo envío de datos en PS [1]

	Ventajas	**Inconvenientes**
Circuit Switching (CS)	- Disminuye el retardo que experimenta el usuario antes y durante una llamada, minimiza la demora durante la llamada. - Proporciona un ancho de banda, canales y una velocidad de datos constantes. - Garantiza que los paquetes de datos se entregan en el orden que se enviaron.	- Los canales se reservan y son de uso exclusivo, consumiendo muchos recursos. - Como consecuencia del pto. anterior, es más caro e ineficiente. - Si no hay recursos disponibles la llamada no se realiza
Packet Switching (PS)	- EFICIENCIA. No necesita canales dedicados para comunicación lo que hace que la metodología sea mucho más eficiente. - CONFIABLE. A diferencia de CS permite el reenvío de paquetes de datos.	- No es conveniente para servicios que requieran de una latencia muy muy baja ya que el funcionamiento es “best effort” y no se puede garantizar un retardo máximo ni fijo. - La implementación es compleja y costosa.

	- COSTES. Debido al primer punto una red basada en PS reduce muchos los costes en comparación con una red que utiliza CS.	

Tabla1. Ventajas e inconvenientes de CS y PS

3.2 PSTN

La red telefónica pública conmutada (PSTN) es la red de las redes telefónicas públicas conmutadas de circuitos del mundo. Originalmente era una red de sistemas telefónicos analógicos de línea fija, pero con el paso del tiempo ha ido sufriendo un proceso de digitalización y ahora es casi completamente digital e incluye interconexión con teléfonos móviles y fijos, dando además acceso a servicios de voz y datos (la voz a través de la red de CS y los datos a través de una red de PS). Fue el primer ejemplo de ingeniería de tráfico para ofrecer garantías de calidad de servicio (QoS) y solo en las partes más antiguas de la red telefónica todavía usan tecnología analógica para cualquier otra cosa que no sea el bucle de la última milla (conexión entre abonado y central telefónica) [29].

Al principio los procesos de conmutación se realizaban manualmente con operadores humanos en las centrales, pero con el paso del tiempo se pasó a conmutadores automatizados introducidos por AT&T en la década de 1950 capaces de manejar miles de llamadas de manera simultánea. La red era conmutada en el sentido de que los nodos era conmutadores que establecían comunicaciones punto a punto entre A y B. Los teléfonos eran analógicos, solo soportaban voz y mandaban la información a las centrales a través del par de cobre a través de las líneas de acceso y la llamada se enruta gracias al plan de numeración E.164.

3.2.1 Arquitectura

La arquitectura está diseñada de la siguiente manera (Figura5):

- **Red local o red de abonado**. El abonado, hogar o empresa, dispone de un teléfono terminal de línea fija o POTS (Plain Old Telephone Service) a través el cuál realiza las llamadas. La red local es el conjunto de elementos utilizados para unir una central local con los abonados.
- **Central primaria**. Estas centrales permiten a los abonados que se encuentran en distintas centrales locales comunicarse entre sí. Cubren zonas primarias y abarca las áreas locales a las que interconecta. La conexión entre las distintas centrales locales y la primaria es la sección primaria.
- **Central secundaria**. Se encarga de conectar centrales primarias mediante la sección secundaria. Una central primaria solo puede depender de una central secundaria. El área que cubre suele ser generalmente el de una provincia.
- **Central terciaria o nodal**. Se encarga de conectar centrales secundarias mediante la sección terciaria. Suelen cubrir el área de una región y la unión o secciones que unen centrales terciarias son las secciones cuaternarias o grandes rutas nacionales.

Resumiendo, el funcionamiento de la red sería: los abonados se encuentran conectados a las centrales locales (rara vez a una central primaria) que les da acceso a los distintos

servicios. Dichas centrales pueden abarcar pueblos o barrios en función de la densidad de población y los clientes que haya. En las centrales están los equipos, de distintas tecnologías, que dan acceso a los servicios, aplican las lógicas necesarias (análisis numeración, facturación, etc.) y se encargan de buscar el camino hacía el destino. Las centrales de conmutación realizan un análisis del canal por el que entra la llamada, número origen y número destino y con la ayuda de unas tablas de enrutamiento se elige la ruta por la que se va a establecer la comunicación. En el caso de que el número destino se encontrase alojado en la misma central local sería esta misma la que dirige la llamada hacía el destino y, en caso contrario, se envía la llamada a las centrales del nivel superior, conocidas como centrales de tránsito, y estas se encargarían de buscar la central local donde se encuentra el destinatario de la llamada basándose en el plan de numeración.

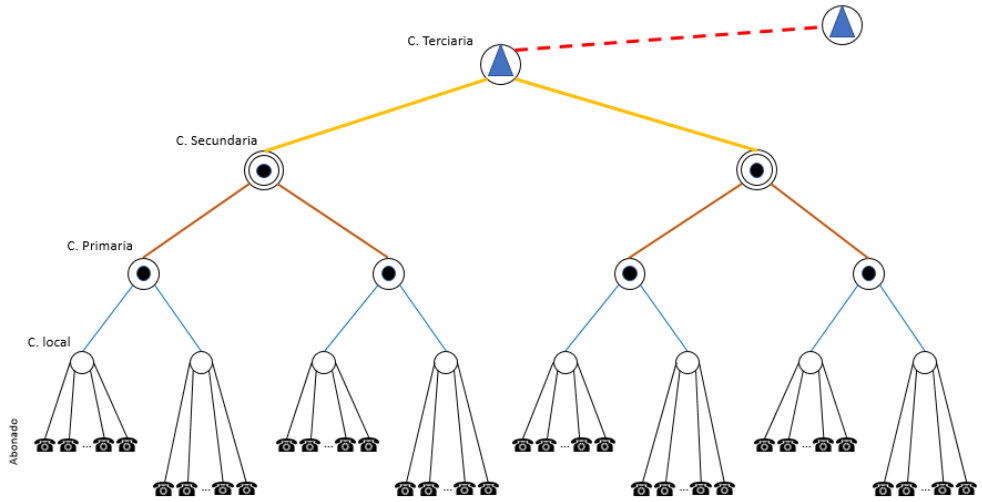


Figura5. Arquitectura PSTN

Cuando se establece una llamada entre dos abonados se van utilizando cada una de las uniones entre el bucle de abonado-central y central-central estableciendo lo que se conoce como ruta final. Para que la comunicación se establezca correctamente es necesario que además de la señal de voz exista un intercambio de mensajes se señalización entre usuario<->central y central<->central. Estos mensajes contienen la información necesaria para el establecimiento y mantenimiento de la llamada, reserva de circuito y recursos, enrutamiento, etc.

3.2.2 Señalización SS7

De manera genérica la señalización podría definirse como el conjunto de datos que intercambian dos equipos, nodos o puntos de red para que puedan realizar funciones de: **direccionamiento**, permite el establecimiento de la llamada, **supervisión**, realiza las tareas relacionadas con el cambio de estado de los elementos de la red (liberación o bloqueo del circuito, descolgado, etc.) y **explotación**, encargado de las tareas de gestión y mantenimiento de la red [30].

La señalización PSTN, incluidos el establecimiento, la gestión y la desconexión de llamadas, está controlada por el sistema de señalización de canal común denominado Sistema de señalización 7 (SS7). Es una pila de protocolos que consta de cuatro capas,

las capas inferiores proporcionan conectividad de red y funciones de enrutamiento, mientras que la capa superior proporciona soporte específico para la aplicación. Para la configuración y desactivación de llamadas de telefonía, la capa superior define un formato llamado ISDN User Part (ISUP), ISUP es el protocolo utilizado para configurar y cancelar llamadas telefónicas.

A continuación, se muestra un diagrama (Figura6) con los mensajes que típicamente se intercambian en el establecimiento de una llamada "normal" (sin fallos, desvíos, etc.) indicando el nombre de cada mensaje y el nodo que lo envía y recibe:

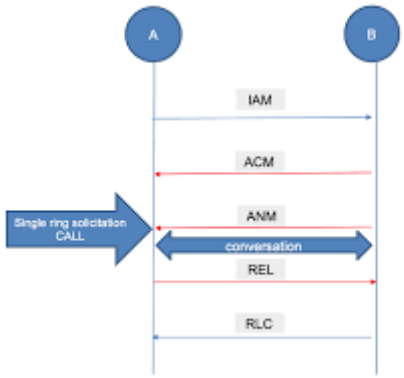


Figura6. Ejemplo llamada - ISUP [2]

Algunos de los principales mensajes serían:

- **Mensaje de Dirección Inicial** (Initial Address Message, **IAM**): contiene la información para el inicio de una llamada, es el primer mensaje que se envía a la central destino durante el establecimiento de la comunicación. Se utiliza para ocupar un canal útil y contiene toda la información necesaria para el direccionamiento hasta la central de destino.
- **Mensaje Completo de Dirección** (Address Complete Message, **ACM**): indica a la central emisora que se alcanzó la central de destino. La parte receptora escucha el timbre del teléfono sobre el tráfico de voz.
- **Mensaje de Respuesta** (Answer Message, **ANM**): Cuando el receptor contesta, la central destino concluye la emisión de tonos de timbrado y envía un ANM a la central origen. Posteriormente se verifica que el abonado que llama se conecta al tráfico reservado y comienza la facturación.
- **Mensaje de Liberación** (Release Message, **REL**): Indica el final de la comunicación y la causa. Las comunicaciones establecidas sin éxito también se liberan con un mensaje REL y, en este caso, contiene también la información de por qué no tuvo éxito el establecimiento de la comunicación.
- **Mensaje de Liberación Completa** (Release Complete Message, **RLC**): el RLC indica el fin de la interconexión de un canal útil y confirma la recepción del REL de modo que el canal útil se encuentra liberado y disponible para establecer una nueva comunicación. Es el fin de la llamada y la tarifación.

Por cómo estaba diseñada la red, su funcionamiento, cada vez que las compañías querían ofrecer un nuevo servicio era necesario montar prácticamente una nueva red. Una vez explicada la matriz que nos permite acceder a los servicios, la PSTN, se va a explicar cómo ha evolucionado el acceso de los usuarios a la red, las tecnologías empleadas, tanto la parte de la red fija como móvil.

3.3 Red fija

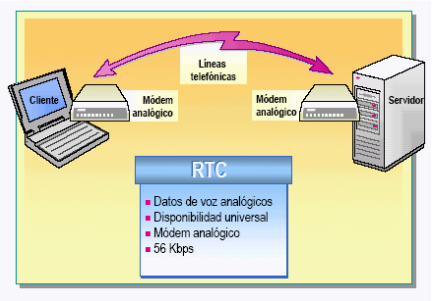
A continuación, se muestra la evolución de la red de acceso en la telefonía fija, posteriormente se hará lo mismo con la red móvil de modo que se pueda entender cuál ha sido la evolución y cómo se va transformando a redes cada vez más eficientes y que buscan la convergencia.

3.3.1 RTC - RTB

La Red Telefónica Conmutada (RTC) —también llamada Red Telefónica Básica (RTB)— es la red original (analógica). La señal del ordenador (la del teléfono ya se explicó anteriormente), que es digital, se convierte en analógica a través del módem y se transmite por la línea telefónica (Figura 7), alcanzando como máximo una tasa de 56 kbits. Al ser una conexión muy lenta dificulta enormemente la descarga de archivos de gran tamaño y el acceso a páginas con contenidos multimedia (imagen, sonido, flash, etc.). Es la red de menor velocidad y calidad.

Los módems analógicos permiten la transmisión de datos entre el equipo de cliente y el ISP a través de la red telefónica conmutada (RTC). Características:

- Tasa de transferencia limitada 56Kbps.
- Módems en banda vocal: → Impide la simultaneidad de la voz con el servicio de Internet.



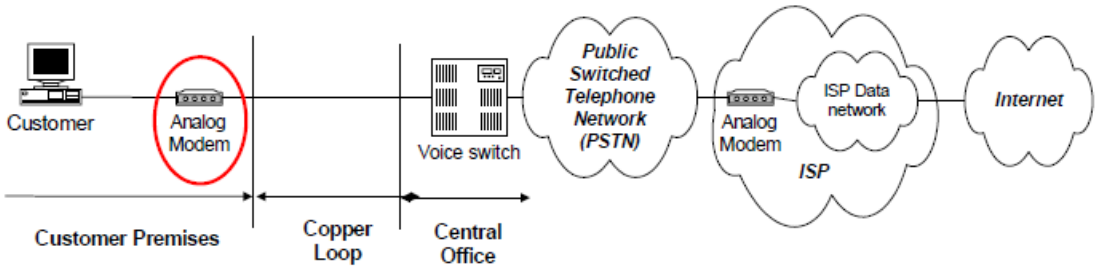


Figura7.Esquema del envío de datos a través de la RTB [3]

3.3.2 RDI/RDSI

Surge como evolución de la RTB o RTC y facilita las conexiones **digitales** extremo a extremo entre los terminales conectados a ella (PC, teléfono, fax, etc.) proporcionando así una amplia gama de servicios a través de interfaces normalizados por la ITU-T.

La naturaleza de la señal analógica es más inestable que la señal digital, sufre más interferencias y mete más ruido por lo que la siguiente apuesta o evolución fue la generación de una señal digital más fácil de tratar, procesar y que abre las puertas a ofrecer una gran cantidad de servicios a los clientes, un aspecto clave fue la mejora de la tolerancia al ruido de la señal digital junto a su versatilidad. Además, la red tradicional comienza un proceso de digitalización (Figura 8), empezando por los enlaces troncales (enlaces entre centrales), en los cuales hay tramos donde la señal deja de ser analógica para ser digital y aprovecharse de las ventajas que esta ofrece. Se empezaron a ofrecer servicios como el 900 a través de lo que se llama la red Inteligente, capaz de realizar un tratamiento complejo de la información, control y procesamiento.

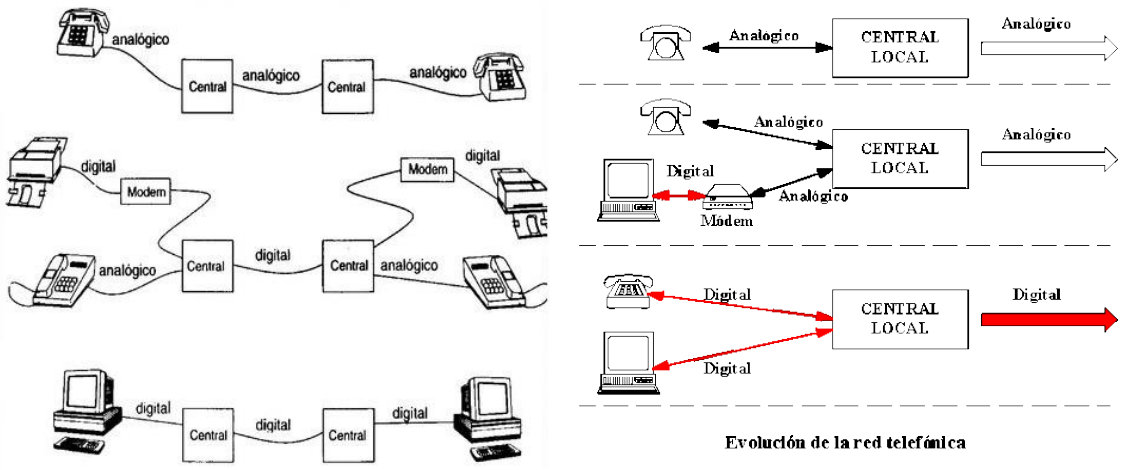


Figura8.Proceso de digitalización de la red [4]

RDSI – ISND se trata de un conjunto de estándares de comunicación para la transmisión digital simultánea de voz, video, datos y otros servicios de red a través de los circuitos tradicionales de la red telefónica pública conmutada [31]. ISDN es un sistema de red telefónica con conmutación de circuitos, pero también proporciona acceso a redes con conmutación de paquetes (PS) permitiendo la transmisión datos (Figura 8).

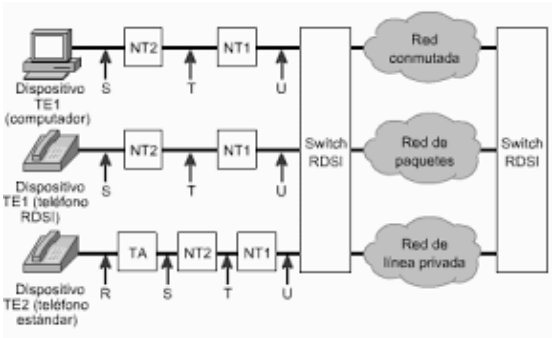


Figura8.Acceso a la red de paquetes en RDSI [5]

La Red Digital de Servicios Integrados es una RDI donde el bucle de abonado ya es digital. Tecnológicamente hablando supone un gran salto y aparece con el objetivo de proporcionar una capacidad de interoperatividad en red que permita a los usuarios acceder fácilmente a los servicios además de integrar y compartir información de todo

tipo: datos, voz, texto, imagen y vídeo; con independencia de las fronteras geográficas, organizativas y tecnológicas. Alguna de las diferencias que aportan son:

- Un único medio de acceso para la transferencia de voz, datos o imágenes.
- Mejora en el tiempo de establecimiento de la llamada (< 800ms).
- Acceso a través de interfaces normalizados.
- Audio de 7KHz, frente a los 3,1 KHz de la RTB, mejorando así la calidad.
- La señalización entre centrales RDSI se apoya en el Sistema de Señalización por canal común número 7, SS7.

A pesar de su capacidad para ofrecer una mejora de los servicios ya existentes y ofrecer otros muchos, la RDSI supuso un fracaso debido principalmente por la irrupción de la tecnología XDSL que tuvo una mejor acogida por lo clientes.

Servicios

RDSI no se limita a ofrecer comunicaciones de voz, ofrece otros muchos servicios como transmisión de datos informáticos, videoconferencia, cobro revertido, la llamada en espera, telefax, conexión a Internet, ... (Figura 9), y opciones como llamada en espera, identidad del origen, ... Los servicios portadores permiten enviar datos mediante conmutación de circuitos o mediante conmutación de paquetes [32].

Figura9.Servicios en RDSI [6]

3.3.3 xDSL

La línea de abonado digital o línea de suscriptor digital, (DSL), es una familia de tecnologías que proporciona acceso a Internet mediante la transmisión de datos digitales a través del par trenzado de hilos de cobre convencionales de la red telefónica básica o conmutada, constituida por las líneas de abonado. La tecnología DSL supone la eliminación de la transformación de la señal digital a señal analógica, por lo que la información es transferida y recibida de forma digital utilizando así todo el ancho de banda que el cableado permite.

DSL aprovecha el ancho de banda no utilizado del bucle local creando canales de 4312,50 Hz. El conjunto de canales utilizables se divide en dos bandas de frecuencia diferentes para el tráfico ascendente y descendente, en función de una relación preconfigurada (esta segregación reduce la interferencia). Una vez que se establecen los circuitos ascendentes y descendentes, el suscriptor puede conectarse al servicio de acceso a Internet u otros servicios de red, como por ejemplo una red MPLS corporativa.

Esta tecnología ofrece servicios de banda ancha sobre conexiones que no superen los 6km de distancia entre la central telefónica y el lugar de conexión del abonado; la velocidad máxima alcanzada depende de la calidad de las líneas, distancia, calibre del cable y esquema de modulación utilizado.

Funcionamiento

Casi todas las líneas telefónicas existentes pueden transportar frecuencias de hasta 1 MHz, pero el servicio telefónico analógico solo requiere una frecuencia máxima de 3.3 KHz, dejando una gran parte del ancho de banda sin usar. xDSL hace uso de este espacio (Figura 10) que de otro modo sería desperdiciado. Otro de los beneficios que ofrece esta tecnología es que el tipo de conexión es permanente y no necesita activación como conexiones analógicas, es decir, antes para realizar una conexión a Internet había que esperar a que se estableciera la conexión con el ISP (proveedor de servicios) antes de poder navegar, con la tecnología xDSL esta conexión siempre esta activa por lo que no es necesario esperas de ningún tipo.

Las primeras implementaciones utilizaban el protocolo DHCP para proporcionar al equipo del abonado los parámetros de configuración de red, como la dirección IP, a través de la autenticación de dicho equipo mediante su dirección física o un nombre de host asignado. Las implementaciones posteriores usan el Protocolo Punto a Punto (PPP), en sus variantes Protocolo Punto a Punto sobre Ethernet (PPPoE) o Protocolo Punto a Punto sobre ATM (PPPoA), para autenticar al equipo del lado del abonado mediante un identificador de usuario y contraseña y proporcionar los parámetros de configuración de red.

Figura10.Distribución frecuencias en ADSL/ADSL2+

A continuación, se muestra un esquema de cómo se dividen los servicios de telefonía básica, POTS, y los servicios de datos en la red (Figura 11):

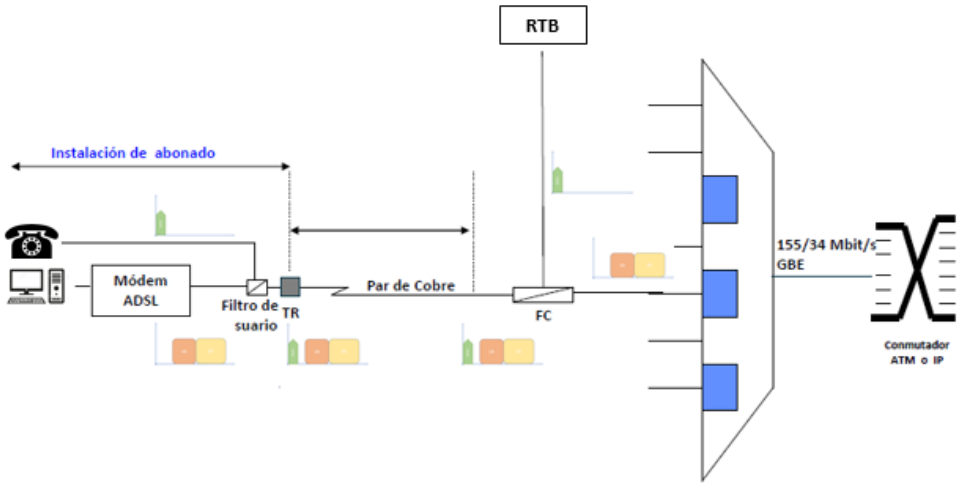


Figura11.Esquema división de los servicios en xDSL

Mediante un filtro y la utilización del DSLAM se separan los servicios de telefonía (POTS) de los digitales. DSLAM (Digital Subscriber Line Access Multiplexer) es un equipo de central que realiza las funciones de:

- Concentrar en un mismo chasis los módems de central (por cada usuario conectado por ADSL se necesita un módem en la central local)
- Multiplexar / demultiplexar sobre una conexión WAN el tráfico de los usuarios.

Evolución xDSL

Existen multitud de variantes o implementaciones de xDSL en función de su tasa de bps y la distancia máxima permitida entre el abonado y la central: ADSL, ADSL2, ADSL2+, SDSL, IDSL, HDSL, SHDSL, VDSL y VDSL2 (Figura 12). A continuación se explican algunas de ellas:

- **ADSL -** Es la más popular forma de tecnología xDSL. El ancho de banda upstream y downstream es asimétrico y la velocidad que ofrece esta comprendida entre 256kbps y 9Mbps de bajada y entre los 64kbps y 1.5Mbps en subida.
- **HDSL -** Este tipo de tecnología DSL simétrica es capaz de llegar a los 2Mbps en una distancia de 3.5Km y es la más avanzada de las tecnologías DSL (que no usan repetidores).
- **SHDSL -** es una variación de HDSL que permite ser utilizada sobre un solo par de cobre y transmisión en distancias mayores, unos 4 Km, y utiliza modulación de amplitud de pulso (PAM), es una técnica para un alto ancho de banda.
- **RADSL -** Utiliza modulación CAP (Carrier-less Amplitude Modulation), versión eficiente de QAM para implementaciones digitales. Es una tecnología de transmisión que tiene ambas aplicaciones simétrica y asimétrica sobre una línea de par de cobre trenzado simple y con velocidades de datos de 7Mbps.
- **SDSL -** Es una versión reducida del HDSL que permite operar bajo una sola línea de cobre trenzado ofreciendo el acceso telefónico tradicional y el intercambio de datos, aunque es más lento.

- **VDSL -** Es propuesto para enlaces locales más cortos, hasta 1 Km. Alcanza velocidades de entre 13 y 52 Mbps en la bajada y entre 1.5 y 6 Mbps en la subida.

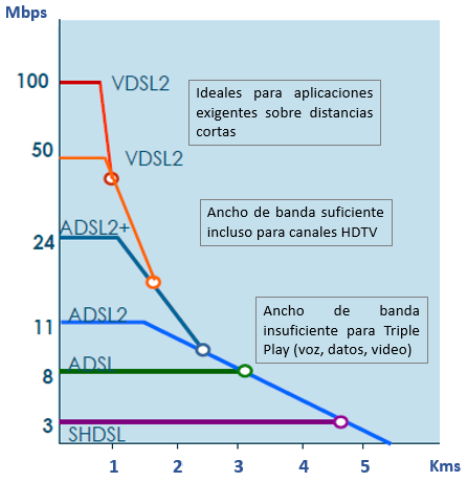


Figura12.Gráfica de troughput/dmáx en xDSL

La tecnología permite a las compañías telefónicas ofrecer servicios de banda ancha sin necesidad de recablear la red y puede implementarse de manera rápida y rentable, especialmente porque beneficia tanto al consumidor (con velocidades de datos más rápidas) como a los proveedores de servicios (con nuevos ingresos a través de cables viejos).

3.3.4 FTTx

Este ha sido uno de los saltos que ha marcado un antes y un después en el mundo de las telecomunicaciones permitiendo ofrecer una gran variedad de servicios, con excelentes prestaciones y a un coste reducido gracias a la optimización de la red y al empleo de una infraestructura formada por componentes pasivos.

Diseño

FTTx (del inglés Fiber to the x) es un término genérico empleado para designar cualquier acceso de banda ancha sobre fibra óptica, que sustituya total o parcialmente el cobre del bucle de acceso. En función de hasta donde se lleva la instalación de FO cambia la última letra, existiendo distintas configuraciones desplegadas (Figura 13):

- **FTTN**: fibra hasta el nodo, la fibra óptica termina en una central del operador de telecomunicaciones que presta el servicio.
- **FTTC**: fibra hasta la acera, Curve, Cabinet, por lo que está más cerca del usuario que FTTN.
- **FTTB**: fibra hasta el edificio - Red de distribución interior. La fibra óptica normalmente termina en un punto de distribución intermedio en el interior o inmediaciones del edificio de los abonados. Desde este punto de distribución intermedio, se accede a los abonados finales del edificio o de la casa mediante la tecnología VDSL2 (Very high bit-rate Digital Subscriber Line 2) sobre par de cobre o Gigabit Ethernet sobre par trenzado CAT6.
- **FTTD**: fibra hasta la puerta. En este caso la red de dispersión se realiza con par de cobre.
- **FTTH**: fibra hasta el hogar, llega hasta el interior de la misma casa del abonado. Es la configuración por excelencia y conecta la centralita del operador con el hogar u oficina y se considera la forma más directa, rápida y fiable de conectarse a Internet.

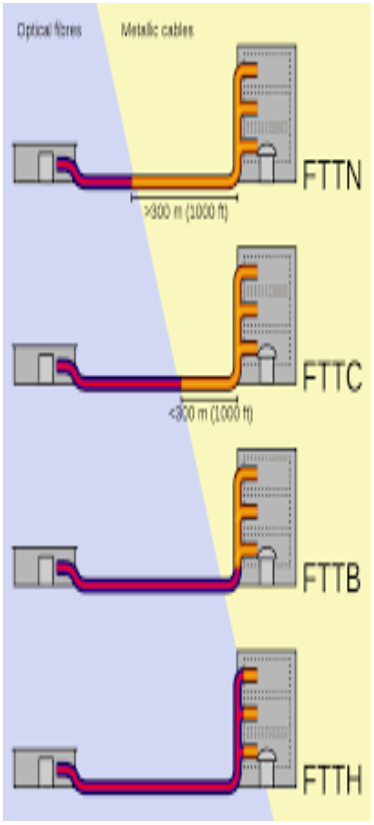


Figura13. Configuraciones FTTx [7]

Tramos que conforma la red y sus elementos

- Central cabecera: donde se encuentran los equipos de terminación de Red Óptica (OLTs) que atienden una determinada zona, equivalente a la central local en las redes RTB, PSTN (se utilizan las centrales que ya hay construidas para la PSTN pero se accede a salas diferentes, en este caso hay una sala de equipos de acceso específicos del servicio FTTx).
- **Red de alimentación**: comprende el tramo de red que va en canalización principal, desde la central cabecera FTTH hasta las cámaras de registro donde se reparte hacía la red de distribución.
- **Red de distribución**: comprende los tramos de red entre la cámara de registro de la canalización principal, donde se encuentra la red de alimentación y desde donde se agrega la red de distribución, y las cajas terminales ópticas o CTOs.
- **Red de dispersión**: discurre entre las cajas terminales ópticas (CTO) y los domicilios del cliente (acometidas).
- Red de cliente: comprende la red en el interior de cada domicilio de cliente.

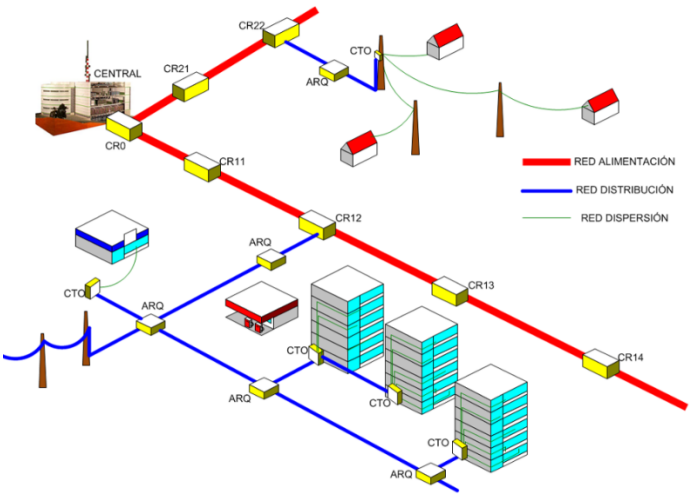


Figura14.Esquema general red FTTH [8]

La **red de alimentación** (Figura 15) está compuesta por:

1. Central del operador donde se encuentran instaladas las OLT (Optical Line Terminal) con tecnología GPON (redes de fibra óptica pasiva con capacidad de Gigabit) y es el enlace que conecta la OLT con el dispositivo que se instala en los hogares, la ONT (Optical Node Terminal), accediendo así a los servicios de banda ancha contratados.
2. Repartidor óptico
3. Equipos de supervisión de fibra
4. Cables de alimentación
5. Cajas de empalme en CR

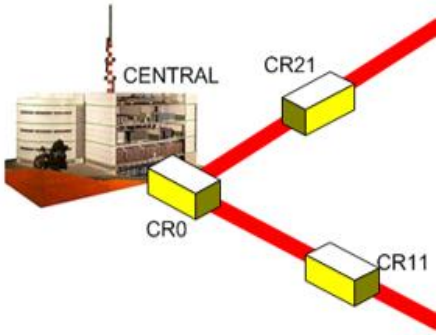


Figura15.Esquema red de alimentación [9]

La **red de distribución** (Figura 16) está compuesta principalmente por:

1. Arquetas y postes
2. Fachada – CTO exterior. Armarios de distribución.
3. Red de distribución en el interior del edificio

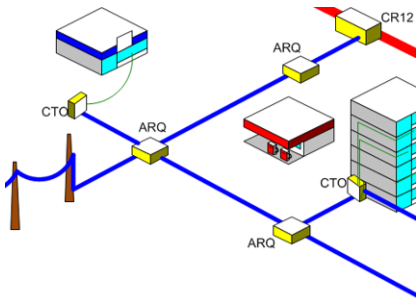


Figura16.Esquema red de distribución [10]

La **red de dispersión** (Figura 17) está formada por:

1. Retenedores y acometidas de exterior
2. Riser
3. Caja de derivación
4. Acometida de interior
5. Roseta óptica

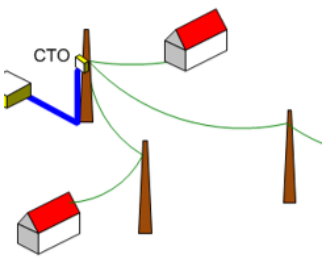


Figura17. Esquema red de distribución [11]

Componentes básicos

OLT (Terminal de línea óptica)

Es el motor que impulsa el sistema FTTH. Normalmente se instala en central, encargándose de la programación del tráfico, control del búfer y asignación de ancho de banda entre otras funciones [33].

Splitter óptico

El splitter óptico divide la potencia de la señal, es decir, cada enlace de fibra que entra al splitter puede dividirse en un número dado de fibras a su salida, permitiendo así que muchos usuarios compartan cada fibra.

ONT (Terminal de red óptica)

La ONT es el dispositivo que se le instala al cliente, está conectado a la OLT por medio de fibra óptica y no tiene elementos activos presentes en el enlace. En GPON, el transceptor en la ONT es la conexión física entre las instalaciones del cliente y la oficina central OLT.

Estándares y evolución

Por último, se muestra la evolución de los últimos estándares y cómo han ido permitiendo aumentar en el número de usuarios y la capacidad de la red.

XGPON	XGSPON	NGPON2
Es la norma de la ITU T G 987 y G 988 que regula las redes ópticas pasivas con capacidad de 10 Gigabit/s • XG PON1 se basa en un upgrade del BW con 10 Gbps DL y 2.5 Gbps UL. Con un rango óptico distinto de GPON para conseguir convivencia. • Se necesita un elemento óptico para multiplexar las distintas lambdas (CEX)	Es un nuevo estándar del 2015 de tecnología PON con velocidades de 10 Gbps simétricos: G.9807.1 • Longitud de onda de 1577 nm down y 1270 nm up (misma que XG PON1. • Tasas de 10 Gbps simétricos • Necesario elemento CEx (multiplexor de lambdas) para convivir con GPON. • No afecta la convivencia de tecnologías (GPON o NG PON2 futuro)	La serie G.989 es la norma que describe la tecnología NG PON2 • Se trata de una tecnología que permite trabajar con 40 Gbps simétricos con cuatro lambdas (futuro abierto a 8). • combina la multiplexación en tiempo y en longitud de onda. Cada ONT pide un slot de tiempo para transmitir como GPON

Tabla2. Estándares de evolución xPON

Entre los principales **beneficios** de la FO destacan:

- Mayor velocidad de transmisión y ancho de banda.
- Inmune a interferencias eléctricas.
- Distancias de transmisión más largas.
- Tamaño reducido.
- Seguridad.
- Eficiencia energética.

De manera análoga existe una red para los servicios móviles que han ido evolucionando con el paso del tiempo y que guarda gran similitud con la lógica y el diseño que han ido siguiendo los servicios de telefonía fija. A continuación, se explicará de manera más sucinta la evolución de tecnologías desde la aparición de la PLMN pasando por el 2G, 3G, 4G y una breve introducción al 5G; de modo que se puedan ver las ventajas o beneficios que ofrece montar una red convergente sobre IP, la NGN.

3.4 Red móvil

El sistema celular fue creado para satisfacer la demanda de comunicación móvil dentro de un espectro de radiofrecuencia limitado. Esta limitación es el factor original que motivó y sigue motivando el desarrollo del teléfono móvil, frente a otros de los varios factores como la duración de la batería.

El sistema celular consiste en la subdivisión de un territorio en pequeñas áreas (llamadas celdas), cada una con una antena de transmisión, de forma que la misma frecuencia puede utilizarse en distintas zonas a la vez dentro de una ciudad. Esto permite un uso mucho más eficiente del espectro. Cuanto más pequeñas sean las celdas, más frecuencias pueden reusarse y más usuarios pueden utilizar el servicio.

En primer lugar, los teléfonos celulares deben funcionar con frecuencias altas, donde las transmisiones pueden limitarse a celdas pequeñas. Además, para poder poner en práctica el sistema celular, un usuario atravesando la ciudad debería poder pasar de una celda a otra sin que la llamada se corte. Para ello es necesario que el sistema pueda saber dónde abandonó el usuario la primera celda, localizar la siguiente, e hilar automáticamente la conversación entre celda y celda. Es decir, el sistema necesita una base de datos con información sobre dónde estaba el teléfono, hacia dónde iba, y quién lo estaba usando; y esta base de datos debía ser rápida. Para realizar esta función el dispositivo necesitaba utilizar un sintetizador de frecuencia y por aquel entonces su precio hacía inviable su uso. Los laboratorios Bell fueron los que consiguieron presentar un sistema que cumplía con los complejos requisitos y la patente fue aprobada en 1972.

3.4.1 PLMN

Una Public Land Mobile Network (PLMN) es una red que proporciona servicios de telecomunicaciones móviles terrestres al público, cada operador que proporciona servicios móviles tiene su propia PLMN. Esta red pública se especifica en el Mobil Country Code (MCC) y cada operador dispone de su propio código de red. Estas redes tienen interconexión unas con otras y, además, con las PSTN proporcionando servicios de red (voz y datos) [34].

3.4.2 1G

En 973 Martín Cooper directivo de Motorola realizó la primera llamada desde un teléfono móvil del proyecto DynaTAC 8000X. El teléfono pesaba cerca de 1 kg, tenía un tamaño de 33 x 4,4 x 8,9 centímetros y su batería duraba una hora de comunicación o una jornada laboral (ocho horas) en espera.

En 1981 el fabricante Ericsson lanza el sistema NMT 450 (Nordic Mobile Telephony 450 MHz). Este sistema seguía utilizando canales de radio analógicos (frecuencias en torno a 450 MHz) con modulación en frecuencia (FM). Cinco años después modernizó el sistema, llevándolo hasta el nivel NMT 900. Esta nueva versión funcionaba prácticamente igual que la anterior, pero a frecuencias superiores (del orden de 900 MHz). Esto posibilitó dar servicio a un mayor número de usuarios y avanzar en la portabilidad de los terminales.

CARACTERÍSTICAS

- Analógica
- Varios estándares (NMT, AMPS, TACs, …)
- Solo proporciona servicios de voz
- No es interoperable con diferentes redes (no roaming entre PLMNs)
- Utiliza la técnica FDMA (Frequency Division Multiple Access) para poder transportar la señal por el interfaz radio. El acceso al medio se realiza dividiendo el espectro en canales (rangos de frecuencia), la frecuencia es un recurso escaso y caro.

3.4.3 2G

En la década de 1990 aparece la segunda generación que utiliza sistemas como GSM y que, en Europa, utiliza las frecuencias de 900 y 1800 MHz. La piedra angular de esta generación es la digitalización de las comunicaciones ofreciendo una mejor calidad de voz que las analógicas, además se aumenta el nivel de seguridad y se simplifica la fabricación del terminal.

Muchas operadoras telefónicas móviles implementaron TDMA, acceso múltiple por división de tiempo, y CDMA, acceso múltiple por división de código, sobre las redes AMPS existentes convirtiéndolas en redes D-AMPS. Esto trajo como ventaja poder lograr una migración de señal analógica a señal digital sin tener que cambiar elementos como antenas, torres, cableado, etc. Inclusive, esta información digital se transmitía sobre los mismos canales ya existentes y en uso por la red analógica. La gran diferencia es que con la tecnología digital es posible hacer multiplexación de modo que en un canal donde antes se transmitía una sola conversación ahora es posible transmitir varias conversaciones de manera simultánea, incrementando así la capacidad operativa y el número de usuarios que pueden hacer uso de la red en una misma celda en un momento dado.

El estándar que ha universalizado la telefonía móvil ha sido el GSM (Global System for Mobile communications). Se trata de un estándar europeo nacido de los siguientes principios:

- Buena calidad de voz (gracias al procesado digital).
- Itinerancia (Roaming).
- Terminales más pequeños y a un precio accesible.
- Instauración de un mercado competitivo con multitud de operadores y fabricantes.

CARACTERÍSTICAS

- Utiliza sistemas estandarizados
- En Europa se utiliza GSM
- Proporciona servicios de voz (datos a muy baja velocidad, FAX – 9 kbps)
- Killer Application: SMS (Short Message Service)
- Núcleo de red basado en circuitos
- Mejora el aprovechamiento del espectro mediante TDMA. El acceso al medio se realiza dividiendo el espectro en canales y a su vez en instantes de tiempo, time-slot.

GSM es una red de circuitos conmutados, ideal para la entrega de voz, pero con limitaciones para el envío de datos.

ARQUITECTURA

La arquitectura de red consta de (Figura 18):

1. **Subsistema terminal**: consta del dispositivo físico y la SIM
2. **Subsistema radio**: es un subsistema que consta de dos niveles jerárquicos y permite dar cobertura geográfica (BTS/BSC)
3. **Núcleo de red**: subsistema que proporciona la posibilidad de ofrecer servicios de voz a los usuarios que se conecten a la red móvil. Está basado en conmutación de circuitos y las principales entidades son: Radio base o BTS, controlador de estaciones base o BSC (intermediario entre el "core" de la red y las antenas), HLR (albergan la ficha de abonado de datos de red y permite localizar al abonado) y MSC (elemento de conmutación y gestión de llamadas)

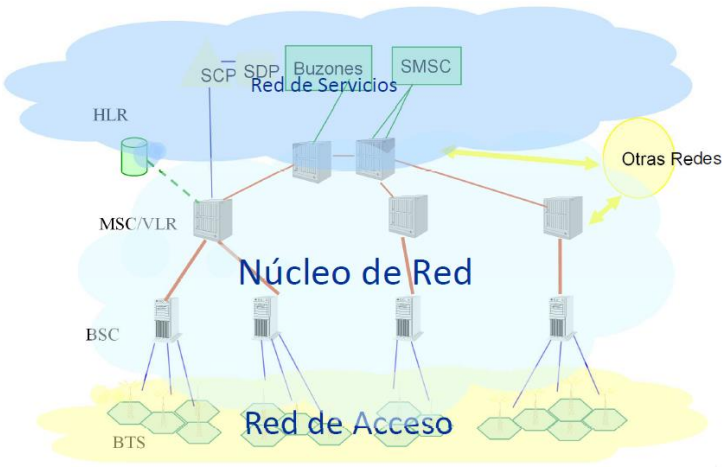


Figura18. Arquitectura red 2G [12]

3.4.4 2,5G

La tecnología 2G pronto empezó a resultar insuficiente debido a la baja velocidad de datos que ofrece por lo que se empezó a desarrollar la siguiente generación, el 3G, pero como aún no podía implementarse porque se encontraba en una fase prematura se optó por una tecnología intermedia, el 2,5G. En la versión incrementada del 2G, el 2.5G, se ofrecen nuevos servicios como EMS (servicio de mensajería mejorado permitiendo la inclusión de melodías e iconos dentro de los mensajes) y MMS (Sistema de Mensajería

Multimedia). Fue posible proporcionar este tipo de servicios al disponer de una mayor velocidad de transferencia de datos, que se hizo realidad con las tecnologías GPRS y EDGE.

- GPRS: agrega la funcionalidad de conmutación de paquetes a las redes GSM. Alcanzaba velocidades de 40 Kbps en el enlace descendente y 14 Kbps en el enlace ascendente al agregar ranuras de tiempo GSM en un portador.
- EDGE: casi 3G o GRPS mejorado. Con una nueva técnica de modulación que produce un aumento de tres veces en la tasa de bits (8PSK reemplazando a GMSK) y una nueva codificación de canal para la eficiencia espectral,

CARACTERISTICAS

- Sistemas estandarizados, en Europa GPRS.
- Permite la introducción de servicios de datos (núcleo de red basado en la conmutación de paquetes)
- EDGE y mejora la capacidad de transferencia de datos en torno a 200 kbps.

ARQUITECTURA

1. Terminal: aparece un nuevo terminal (terminal GPRS) que es compatible con la red GSM y permite implementar las conexiones a las redes de datos.
2. Acceso radio (GERAN): el subsistema radio es el mismo que para redes GSM con una evolución de la BSC (se introduce la PCU para poder conectar el subsistema radio al nuevo dominio de paquetes).
3. Núcleo móvil: se añade un nuevo dominio dentro del núcleo de red basado en la conmutación de paquetes. A través de este nuevo dominio se permite la conexión con las redes de datos (Packet Data Network): Internet y otros operadores móviles (PLMNs). De esta forma se introduce la posibilidad de cursar servicios de datos y de voz a través del mismo interfaz radio.

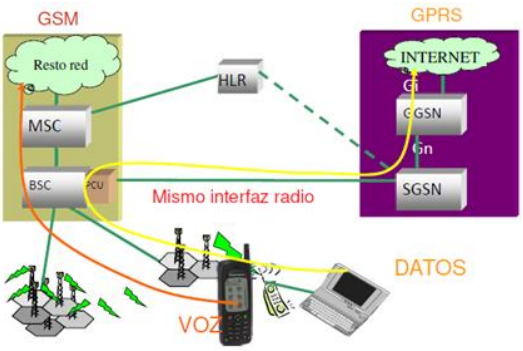


Figura19. Arquitectura red GPRS [13]

3.4.5 3G

El objetivo de los sistemas de tercera generación es la universalidad del servicio en términos de usuario y prestación del mismo, UMTS (Universal Mobile Telecomunication System. En 1999, este sistema ya es capaz de utilizar WCDMA lo cual hace posible

alcanzar velocidades elevadas, entre 144kpbs hasta 7,2 Mbps, permitiendo ofrecer servicios como videoconferencia, televisión y descarga de archivos.

CARACTERISTICAS

- Se estandariza por el 3GPP
- Mejora las prestaciones de servicios de datos (alcanza 384kbps)
- El núcleo de red conta de:
    - Conmutación de circuitos: proporciona los servicios de voz (y videollamada)
    - Conmutación de paquetes: proporciona conexión a redes de datos (hasta 2Mbps en DL y 384 Kbps en UL)
- Mejora de la eficiencia espectral utilizando como técnica de acceso WCDMA mediante la aplicación de códigos ortogonales.

ARQUITECTURA

1. Terminal: aparece un nuevo terminal 3G capaz de soportar las tecnologías anteriores.
2. Acceso Radio (UTRAN): aparece un nuevo subsistema radio, diferente del subsistema radio de las redes 2G/GPRS/EDGE. Este nuevo subsistema radio consta de dos niveles jerárquicos (nodB/RNC)
3. Núcleo Móvil: la arquitectura UMTS es una evolución de la arquitectura de 2G/3G en el núcleo móvil, por lo que la mayoría de las entidades funcionales son duales para ambas tecnologías. De esta manera, el núcleo móvil consta de dos dominios, circuitos y paquetes, que permite manejar los servicios de voz y datos, respectivamente, de manera diferenciada.

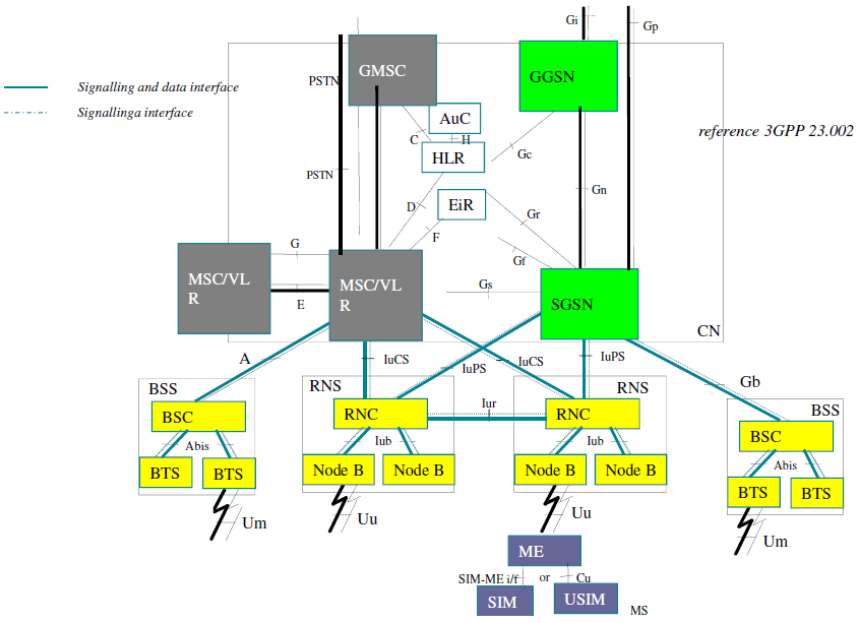


Figura20. Arquitectura red 3G [14]

3.4.6 3,5G

Surge como la optimización de la tecnología espectral UMTS/WCDMA a través de la tecnología HSDPA. Consiste en el uso de un canal compartido en el enlace descendente,

DL, que mejora significativamente la capacidad máxima de transferencia de información pudiéndose alcanzar tasas de bajada de hasta 14 Mbps y soporta tasas de throughput promedio cercanas a 1 Mbps. Actualmente, también está disponible la tecnología HSUPA, con velocidades de subida de hasta 5,8 Mbps, y HSPA+ con velocidades de hasta 84 Mbps de bajada y 22 Mbps en la subida.

HSDPA: 14,4 Mbps (DL) » HSUPA: 7,2 Mbps (UL) » HSPA+: 28 Mbps (DL) » HSPA+ A: 42 Mbps (DL)

3.4.7 4G

El objetivo es la evolución o migración de los sistemas 3GPP hacia un sistema basado en conmutación de paquetes con mayores tasas de bit, menores latencias y capaz de soportar múltiples accesos radio.

La arquitectura LTE consta de dos partes la EPC (Evolved Packet Core) y EUTRAN. EUTRAN es la encargada de todas las funciones relacionadas a la interfaz de radio y el control de los móviles y, por otro lado, la EPC brinda acceso a otras redes de paquetes IP y donde se gestiona los aspectos relacionados a la seguridad, calidad de servicio, gestión de recursos y movilidad.

4G es la evolución tecnológica que ofrece al usuario de telefonía móvil internet con mayor rapidez y mayor ancho de banda permitiendo, entre muchas otras cosas, la recepción de televisión en alta definición.

ARQUITECTURA

1. Terminal: el nuevo terminal (terminal LTE) es compatible con las tecnologías anteriores (GPRS/UMTS)
2. Acceso Radio (eUTRAN): el subsistema radio es totalmente diferente, solo tiene un nivel jerárquico compuesto por enodoB para poder simplificar y abaratar el acceso, que se encarga de: funciones de gestión de recursos de radio, enrutamiento en el plano de usuario y reportes de configuración para movilidad entre otros.
3. Núcleo móvil (EPC): Solo existe un dominio basado en la conmutación de paquetes que proporciona la posibilidad de ofrecer servicios de voz (VoLTE) y servicios de datos de manera simultánea y tener un tratamiento de la información adecuado (incorpora calidad de servicio en el tráfico de datos para poder ofrecer servicios en tiempo real).
    - **MME** proporciona conectividad entre el eNodoB y la red GSM o UMTS a través del SGSN y provee señalización, seguridad y control.
    - **SGW** es controlado por el MME y es un punto donde se monitorizan las políticas de conexión y servicio establecidos.

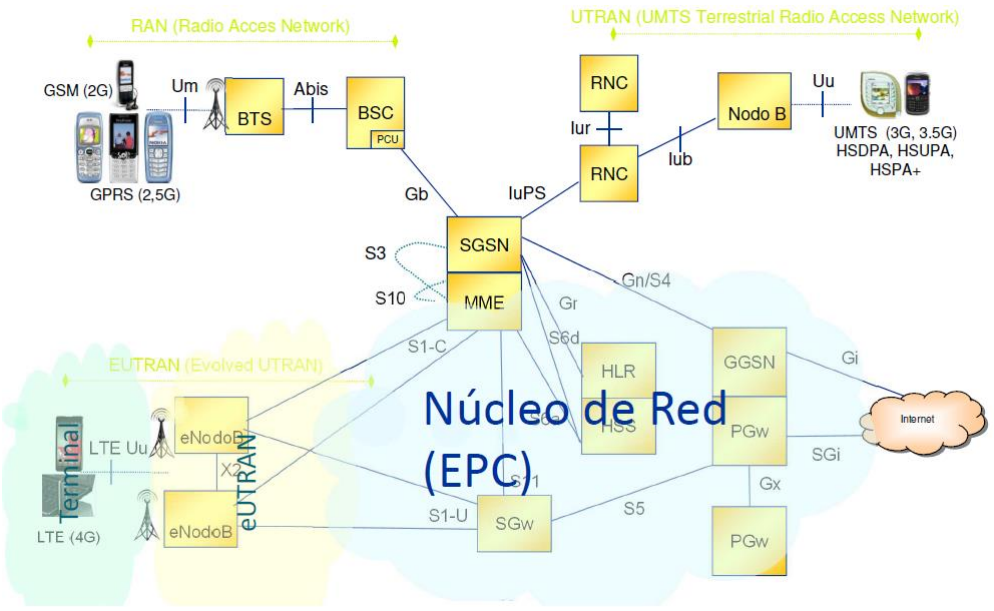


Figura21. Arquitectura red 4G [15]

3.4.8 5G

5G pertenece a la nueva generación de redes de comunicaciones móviles que ha sido estandarizada hace relativamente poco tiempo por el 3GPP (3rd Generation Partnership). Es conocida actualmente como New Radio dentro del 3GPP, y comercialmente como 5G.

- El estándar 5G incluye una nueva red de acceso radio, un nuevo núcleo de red, y se han estandarizado nuevas bandas para su operación (soporta también las ya existentes).
- Está optimizada para los servicios de banda ancha móvil (MBB), con soporte para servicios de voz/vídeo sobre paquetes.
- La red 5G evoluciona el soporte del MBB hacia eMBB (mayor velocidad y prestaciones).

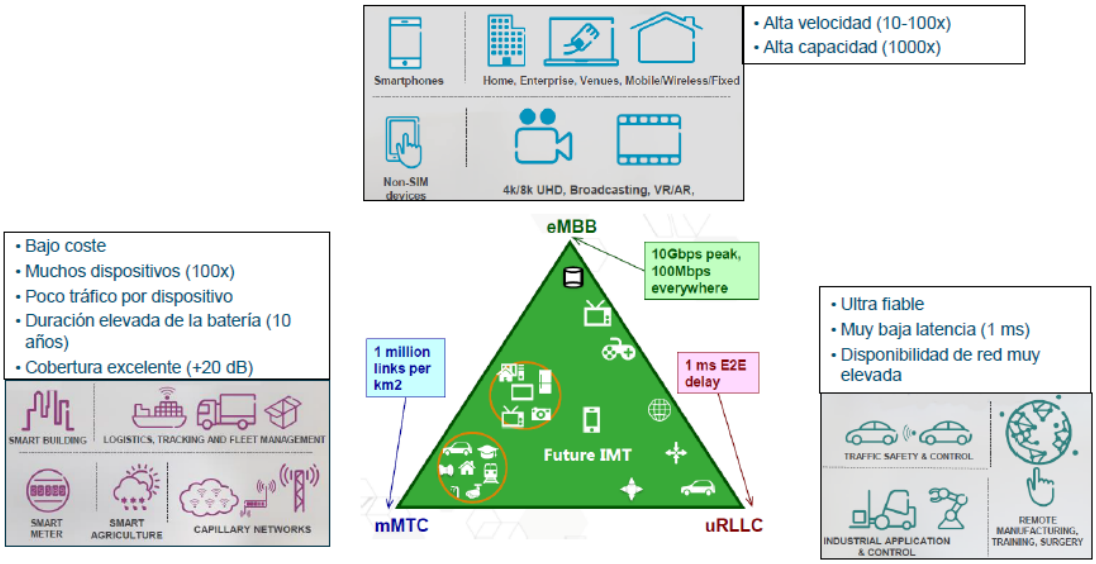
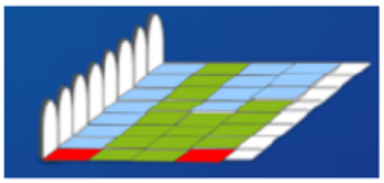


Figura22. Requisitos servicios 5G [16]

Avances tecnológicos

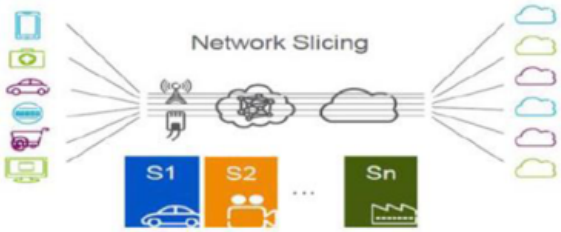
Para que 5G pueda dar todo lo que promete y sea posible alcanzar ciertas velocidades para ofrecer determinados servicios, la generación cuenta con una serie de avances tecnológicos tanto en la capa física, como el uso de la técnica de transmisión OFDM, como en las técnicas de acceso a la red, con mejoras como MIMO o Network Slicing.

Uso de bandas de frecuencia altas con mayor ancho de banda



Multiplexación OFDM con numerología flexible
Multiplexación de servicios en la misma portadora
Filtrado de subportadora para facilitar coexistencia de servicios
Flexibilidad en la asignación del espectro

Tecnologías multiantena avanzadas
- **Haces direccionales**
- **Multi User MIMO**
- **Massive MIMO**



Una sola red divida en redes lógicas, con recursos repartidos dinámicamente entre distintos servicios

Merece la pena explicar brevemente las dos últimas tecnologías comentadas:

MIMO

Permite usar antenas con un número muy elevado de elementos radiantes (típicamente 192) ofreciendo 2 potenciales ventajas: SU MIMO (Single User MIMO), enfoca la energía del sistema radiante en un solo usuario y envía un número elevado de flujos de información (layers) en paralelo (hasta 8/4 DL/UL), MU MIMO (Multi User MIMO), crea múltiples haces simultáneos, de modo que se puedan servir simultáneamente (en el mismo recurso tiempo/frecuencia) a varios usuarios con 1 o varios flujos para cada uno.

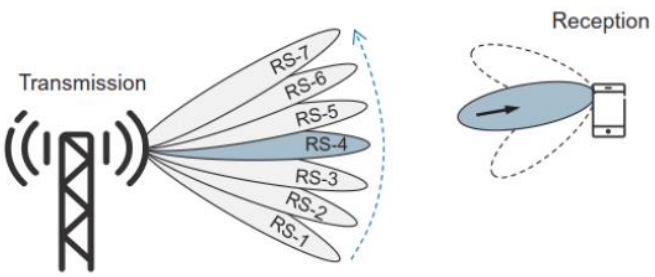


Figura23. Funcionamiento MIMO

NETWORK SLICING

El Network Slicing es un cambio de filosofía respecto a la idea tradicional de dedicar redes físicas y recursos diferenciados para distintas aplicaciones. Con el Network Slicing una sola red física es dividida en varias redes lógicas, especializadas en distintos tipos de servicios, es una arquitectura que permite el fraccionamiento de redes lógicas virtualizadas e independientes utilizando la misma infraestructura de red física. La materialización de esta tecnología se sostiene en los conceptos de redes definidas por software (SDN, Software Defined Networking) y la virtualización de funciones de red (NFV, Network Function Virtualization) las cuales permiten la implementación de segmentos de redes flexibles y escalables sobre una infraestructura de red física común y que se explicaran posteriormente.

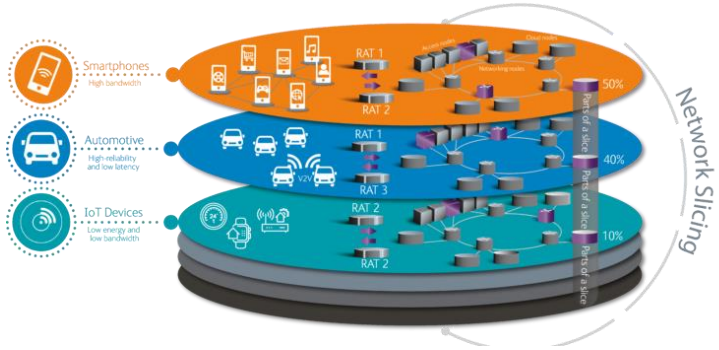


Figura24. Ejemplo de separación de servicios mediante Network Slicing. [17]

ARQUITECTURA

- Las estaciones base se llaman gNB, a diferencia de los eNB del 4G, y las estaciones base 4G evolucionan a ng eNB para poder tener interfaces tanto con las estaciones 5G como con el núcleo de red 5G.
- Los nodos del núcleo de red también evolucionan (AMF/UPF), y tienen el nuevo interfaz NG (N2 plano de control, N3 plano de usuario) tanto con estaciones base 5G como 4G evolucionadas.
- El interfaz Xn es la evolución del X2, y sirve para comunicar estaciones base 4G evolucionadas y/o 5G.

La gNB CU se podrá desplegar con NFV (Network Function Virtualization), es decir, podrá ser implementada mediante SW sobre HW genérico.

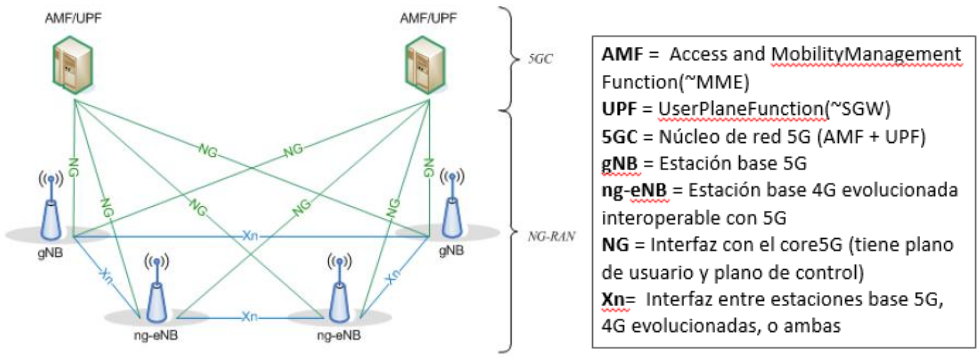


Figura25. Arquitectura red 5G [18]

4. NGN – IMS

4.1 Red de VOIP

Voz sobre IP (VoIP) se refiere a la transmisión del tráfico de voz sobre redes basadas en Internet en lugar de las redes telefónicas tradicionales PSTN (red telefónica pública conmutada). Para implantar el servicio de voz sobre IP es necesario disponer de una red compatible con esta tecnología y es aquí cuando se hace la introducción de la red de telecomunicaciones actual, que ha sido diseñada para manejar de manera eficiente todo tipo de información tanto de voz como datos o servicios multimedia, entre otros, permitiendo la convergencia de las redes y un uso eficiente de la misma.

4.2 NGN

Red de Siguiente Generación o Red de Próxima Generación, NGN por sus siglas en inglés, es un término que se refiere a la evolución de la actual infraestructura de redes de telecomunicación y acceso telefónico con el objetivo de lograr la convergencia tecnológica de los servicios multimedia (voz, datos y todo tipo de medios como video) encapsulándolos en paquetes IP, similares a los que se usan en Internet. Es una red basada en la transmisión de paquetes capaz de proveer servicios integrados, incluyendo los tradicionales telefónicos, y capaz de explotar al máximo el ancho de banda del canal haciendo uso de las Tecnologías de Calidad del Servicio (QoS) de modo que el transporte sea totalmente independiente de la infraestructura de red utilizada.

Existe una separación bien definida entre la red de transporte (conectividad) y los servicios que corren por encima de esta, lo que quiere decir que siempre que un proveedor telefónico desee habilitar un nuevo servicio, puede hacerlo fácilmente definiéndolo desde la capa de servicio directamente sin tener en cuenta la capa de transporte. Es una red que se caracteriza por los siguientes aspectos fundamentales: transferencia basada en paquetes, soporte para una amplia gama de servicios, aplicaciones y mecanismos basado en módulos de servicios (principalmente multimedia), capacidades de banda ancha con calidad de servicio y la transparencia de extremo a extremo, interfuncionamiento con redes heredadas o legacy a través de interfaces abiertas y convergencia entre red fija/móvil.

4.3 IMS

Subsistema Multimedia IP (IMS, IP Multimedia Subsystem), es un conjunto de especificaciones que describen la arquitectura de las redes de siguiente generación (Next Generation Network, NGN), para soportar telefonía y servicios multimedia a través de IP. Más concretamente, IMS define un marco de trabajo y arquitectura base para tráfico de voz, datos, video, servicios e imágenes conjuntamente a través de infraestructura basada en el enrutamiento de paquetes a través de direcciones IP. Esto permite incorporar en una red todo tipo de servicios de voz, multimedia y datos en una plataforma accesible a través de cualquier medio con conexión a Internet, ya sea fija o móvil. Sólo requiere que los equipos utilicen el protocolo de sesión SIP (Session Initiation Protocol) que permite la señalización de sesiones, se explicará posteriormente.

IMS no solamente sirve para proporcionar nuevos servicios, lo que realmente pretende es servir para todo tipo de servicios, tanto actuales como futuros, que se puedan prestar por Internet y permitirá que los operadores puedan controlar y facturar cada uno de ellos. El objetivo es el de proveer una plataforma de red común convergente, donde el servicio es agnóstico del acceso. Se trata de una arquitectura de red global que permite ofrecer servicios multimedia a distintos tipos de accesos y terminales a través de un núcleo de control (que enruta los mensajes) y los vincula con los servidores de aplicación donde residen los distintos servicios.

4.3.1 Arquitectura IMS

La arquitectura IMS (IP Multimedia Subsystem) está dividida en varios niveles o capas (Figura 25), atendiendo cada capa a sus elementos de red y funcionalidades.

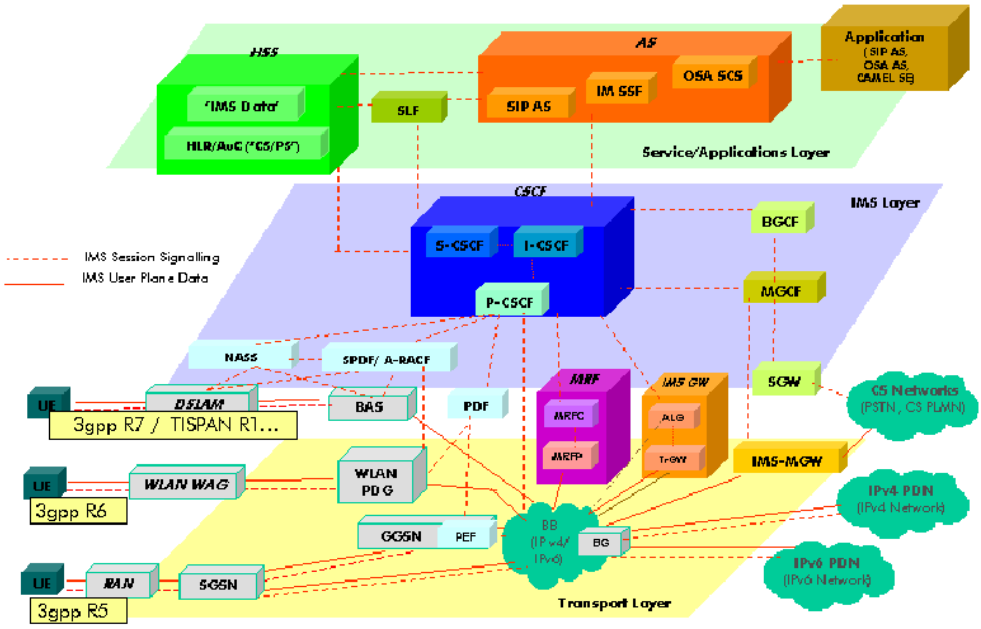


Figura25. Arquitectura IMS [19]

Capa de Acceso

La capa de acceso puede representar todo tipo de acceso de alta velocidad como, por ejemplo: xDSL, FTTHx, Wi-Fi (Wireless Fidelity), Wi-Max, LTE (Long Term Evolution), 5G, etc.

Capa de Transporte

La capa de transporte representa una red IP. Esta red podrá integrar mecanismos de calidad de servicio con MPLS, DiffServ o RSVP principalmente. Está compuesta por routers (“edge routers” para el acceso y “core routers” para el tránsito), conectados a través de una red de transmisión.

Capa de Control

Está formada por diversos controladores de sesión (Session Controllers) responsables del encaminamiento de la señalización entre usuarios y de la invocación de los servicios. Estos nodos se llaman “Call Session Control Function” o CSCF. También aquí se encuentra la Base de Datos (HSS) y el “Multimedia Resource Function” (MRF) que los proveedores llaman Servidores de Media IP (“IP Media Server” o “IP MS”). El MRF se divide en dos partes, el MRFC (MRF Control), ubicado en la Capa de Control, y el MRFP (MRF Processor), ubicado en la Capa de Transporte. La interconexión entre ambos es a través del protocolo H.248.

Capa de Aplicación

La capa de aplicación introduce las aplicaciones (servicios de valor agregado) propuestas a los usuarios o empresas. Esta capa está formada por servidores de aplicación (Application Server, AS). En esta capa residen TODOS los servicios. Aquellos que alguna vez estuvieron integrados junto con el control en las redes monolíticas (independientes o aisladas), hoy están a disposición de cualquier usuario sin importar cuál sea su acceso.

4.3.2 Acceso

El usuario puede conectarse a una red IMS de varias formas el único requisito es que pueda utilizar IP y ejecutar agentes de usuario del Protocolo SIP (Figura 26). Por parte del acceso fijo estaría el xDSL o FTTH, por ejemplo, en la parte de acceso móvil está la red 4G y GPRS, entre otras, y por parte de acceso inalámbrico Wifi o Wi-MAX serían compatibles. ¿Qué ocurre con el resto de los accesos? Otros sistemas telefónicos, como el antiguo servicio telefónico simple (POTS, los teléfonos analógicos antiguos), H.323 y sistemas VoIP no compatibles con IMS tiene acceso a través de las puertas de enlace.

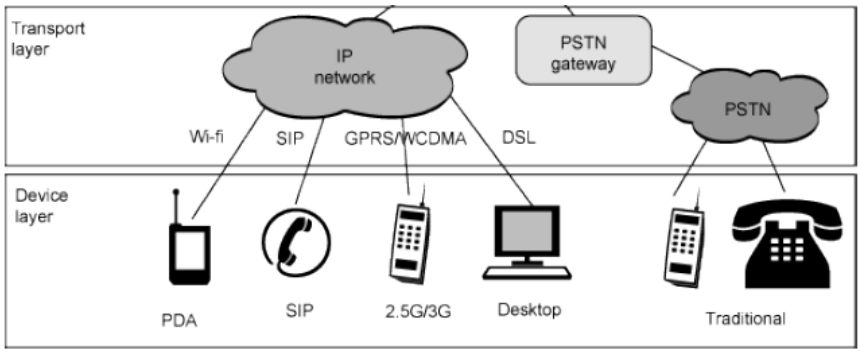


Figura26. Ejemplos de acceso a red NGN-IMS

4.3.3 Componentes de señalización

Los componentes del IMS responsables del procesamiento y reenvío de mensajes SIP se denominan "Call Session Control Function" (CSCF) y, dependiendo de su funcionalidad, podemos distinguir tres tipos: Proxy (P-CSCF), Interrogating (I-CSCF) y Serving (S-CSCF).

- Proxy-Call Session Control Function (P-CSCF): actúa como proxy de salida y es el primer punto de contacto para el equipo de usuario (UE) del abonado. Por tanto, es la puerta que da acceso a la red IMS.
- Interrogating-Call Session Control Function (I-CSCF): Es el punto de contacto en la red del operador para todas las llamadas/sesiones hacia un usuario (designa

el S-CSCF del usuario en la fase de registro, encamina mensajes SIP a otros nodos de la red del operador, obtiene la IP de S-CSCF mediante consultas DIAMETER a HSS e interacciona con los I-CSCF de otras redes).

- Serving-Call Session Control Function (S-CSCF): actúa como componente central en una red IMS. La tarea principal del S-CSCF es recibir solicitudes, verificar la ubicación del usuario y reenviar las solicitudes al destinatario (y a los servidores de aplicaciones correspondientes). Cuando un usuario registra una identidad SIP en IMS, se le asigna un S-CSCF (se acampa en un S-CSCF y se mantiene en este durante todo el tiempo que el usuario mantenga el registro). Lleva a cabo el control de las sesiones SIP hacia los servicios de los clientes manteniendo el estado de cada sesión (procesa solicitudes de registro, establece, modifica y termina sesiones, interacciona con los AS, monitoriza llamadas, etc..).
- Función de resolución de nombres DNS-ENUM: permite identificar cada equipo/servidor de una red IP, mediante nombres descriptivos (nombres de dominio o FQDN), y asocia cada nombre a otros parámetros de red, los parámetros más habituales son las direcciones IP y puertos.

  El protocolo ENUM (tElephone NUmber Mapping), permite obtener información de encaminamiento de las sesiones SIP al almacenar para cada cliente su identidad en formato numérico (E.164) y su traslación a formato URI, típicamente SIP URI.

4.3.4 Componentes de interconexión

Dado que las redes IMS se van desplegando gradualmente tendrán que interactuar, no solo con otras redes IMS, sino también con otras redes legacy como las redes de conmutación de circuitos, por ejemplo, PSTN basadas en SS7; por lo que es necesario disponer de elementos de interconexión.

- Función de control de fronteras de interconexión (IBCF): actúa como nodo de interconexión entre las redes IMS y otras redes IMS o SIP.
- Función de control de puerta de enlace fronteriza (BGCF): es un proxy SIP que enruta las solicitudes SIP destinadas a un usuario en el dominio de conmutación de circuitos mediante un plan de marcado. Cuando un número de teléfono es servido por otra red, la BGCF reenvía la solicitud al BGCF o MGCF de esa red destino.
- Función de control de pasarela de medios (MGCF): es responsable del mapeo de los protocolos de establecimiento de sesión entre las redes IMS y CS (Tabla 3). El MGCF convierte la señalización SIP en PUSI sobre IP o BICC sobre IP. En el lado de IMS, el MGCF actúa como un UE de IMS que inicia y termina sesiones SIP.

  Además de convertir los protocolos, el MGCF también activa el establecimiento de una ruta de medios en Media Gateway (MGW). A continuación, se muestra una tabla de equivalencia entre los mensajes del protocolo SIP y PUSI, que es la traducción que realiza el MGCF.

ISUP Message	SIP Message
Initial address message (IAM)	INVITE
Address complete message (ACM)	180 RINGING / 183 PROGRESS
Answer message (ANM)	200 OK
Connect message (CON)	200 OK
Call progress message(CPG)	183 CALL PROGRESS
Release message (REL)	BYE/CANCEL
Release complete message (RLC)	NA
Reset circuit (RSC)	BYE/CANCEL (only for a call) *
Circuit group reset message(GRS)	BYE/CANCEL (only for a call) *
Circuit group blocking message(CGB)	BYE/CANCEL (only for a call) *
Suspend message (SUS)	Re-Invite or INFO **
Resume message (RES)	Re-Invite or INFO **

Tabla3. Equivalencia entre menajes de señalización ISUP-SIP. [20]

- Pasarela de señalización (SGW): mientras que el MGCF es responsable de convertir los mensajes SIP en ISUP / BICC y viceversa, la SGW realiza la conversión de protocolo de capa inferior. El SGW traduce el transporte SCTP/IP utilizado en IMS en MTP utilizado en redes SS7.
- Pasarela de medios (MGW): mientras que el MGCF convierte la parte de señalización de una llamada entre un usuario de IMS y un usuario de CS, el MGW convierte los medios, por ejemplo, audio o vídeo, intercambiados entre origen-destino.

4.3.5 Componentes relacionados con el aprovisionamiento de aplicaciones y servicios

- Application Servers (AS): son servidores SIP que alojan y ejecutan los servicios. Se comunican con el S-CSCF a través de la Interfaz de control de servicios (ISC) basada en SIP. En general, se puede distinguir entre servidores de aplicaciones SIP nativos que alojan y ejecutan servicios basados en SIP y servidores de aplicaciones que proporcionan una interfaz entre el IMS y las infraestructuras de servicios heredadas.
- MRF: un tipo especial de servidor de aplicaciones es la función de recursos multimedia (MRF) que proporciona servicios de medios, como sesiones de conferencias.

En IMS hay una base de datos principal: **Home Subscriber Server (HSS)**. El HSS es la base de datos principal para almacenar toda la información de los suscriptores y los datos relacionados con el servicio, esto incluye la identidad del abonado, la información de acceso necesaria para autenticar al abonado y el perfil de servicio del abonado que describe a qué servicios está abonado el usuario y a qué servidores de aplicaciones deben utilizarse cuando se llama al abonado o está estableciendo una llamada.

4.4 Señalización SIP

SIP se define en la RFC 3261, es un protocolo cliente/servidor de nivel de aplicación basado en el paradigma de petición/respuesta, en SIP las peticiones son generadas por una entidad (el cliente) y enviadas a otra entidad encargada de recibirlas (el servidor),

el par petición/respuesta recibe el nombre de transacción. Es un protocolo definido para proveer servicios avanzados de telefonía sobre una red IP para establecer y terminar llamadas de voz sobre redes IP permitiendo controlar el establecimiento de comunicaciones o sesiones de voz entre dos o más peers.

En una transacción el cliente establece una conexión con el servidor y le envía un mensaje que encapsula el servicio deseado, por su parte el servidor contestará con un mensaje similar pero que adjunta información de estado e información adicional. El propio protocolo dispone de mecanismos que garantizan la fiabilidad de las comunicaciones. Además, SIP depende del protocolo SDP (Session Description Protocol) para llevar a cabo la negociación del codec empleado en la transmisión del medio.

Los mensajes emplean los campos de cabecera para especificar información como el llamante, llamado, el camino que seguirá el mensaje, el tipo y la longitud del cuerpo del mensaje. Algunas de las características del protocolo son:

- Permite establecer, modificar y terminar sesiones multimedia.
- No se preocupa de qué tipo de medio se intercambia durante una sesión, de esto se encarga SDP.
- Soporta movilidad del usuario mediante redireccionamiento de las sesiones hacia la ubicación actual del cliente.
- Puede montarse sobre TCP o UDP.

4.4.1 Mensajes

Existen tipos de mensajes SIP:

- *INVITE:* Inicio de sesión
- *OPTIONS:* Preguntar por opciones y capacidades
- *ACK:* Reconocimiento
- *BYE:* Terminación
- *CANCEL:* Cancelación de INVITE
- *REGISTER:* Registro de URL SIP
- *PRACK:* Reconocimiento provisional
- *INFO:* Transporte de información en llamada
- *SUBSCRIBE:* Requerir notificación de evento
- *UNSUBSCRIBE:* Cancelación notificación de evento
- *NOTIFY:* Notificación de evento
- *MESSAGE:* Mensaje instantáneo
- *REFER:* Transferencia a otra URL SIP

El mensaje de respuesta empieza con una línea denominada Status-Line, seguida de las cabeceras y del cuerpo del mensaje. La Status-Line tiene el siguiente formato:

SIP-versión space Status-Code space Reason-Phrase CRLF

Donde Status-Code es un entero de tres dígitos que indica el resultado del intento de servir la petición (Tabla 4). El primer dígito indica el tipo de estado de la respuesta. En SIP existen 6 tipos de estado, tal como se muestra a continuación:

Código de estado	Categoría
1xx	Información de estado de llamada en progreso
2xx	Éxito
3xx	Desvío
4xx	Error en el cliente
5xx	Error en el servidor
6xx	Fallo general

Tabla4. Códigos de estado protocolo SIP.

Este sería el intercambio de mensajes de señalización SIP en un escenario de llamada "normal" (sin desvíos ni errores):

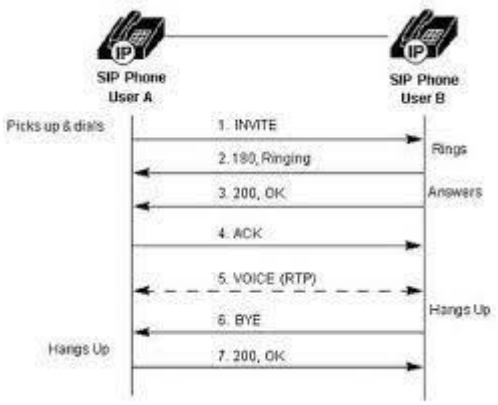


Figura27. Flujo llamada SIP completa (con RTP) [21]

5. Virtualización

Llegado este punto hay que introducir un nuevo componente o avance tecnológico, la virtualización, una tecnología que ha cambiado completamente el mundo de la informática y las telecomunicaciones; mejorando los costes, tiempos y eficiencia en el despliegue de los nodos de la red, facilitando además las tareas de integración, aprovechando todos los recursos disponibles y aportando a su vez flexibilidad y seguridad al entorno.

5.1 ¿Qué es?

La virtualización introduce una nueva capa entre el hardware y el software, lo que posibilita que se puedan compartir un conjunto común de recursos. En un hardware virtualizado, se puede gestionar una gran variedad de cargas de trabajo sin que implique que cada una de ellas no esté aislada de la otra, migrar libremente entre infraestructuras y escalar, según sea necesario.

Esto supone para las empresas un aumento no solo del capital sino también de la eficiencia operativa mediante la virtualización, ya que implica una mejor utilización y consolidación de los servidores, la asignación y gestión dinámica de recursos, el aislamiento de las cargas de trabajo, la seguridad y la automatización. La virtualización permite el auto-aprovisionamiento de servicios a demanda y hace posible la coordinación y ampliación de recursos en función de las necesidades específicas de la empresa.

Esta tecnología utiliza software que simula la funcionalidad del hardware para crear un sistema virtual permitiendo operar múltiples sistemas operativos y varias aplicaciones en un solo servidor, es decir, utilizar la capacidad total de una máquina física distribuyendo sus capacidades entre servidores lógicos ofreciendo como beneficios una mayor eficiencia y economías de escala.

Explicado de manera más práctica o visual, supongamos que una empresa cuenta con 3 servidores físicos con propósitos individuales dedicados: un servidor de correo, un servidor web (intranet) y el último ejecuta aplicaciones internas. Cada servidor utiliza aproximadamente el 30% de su capacidad, en los momentos de alta carga, solo una fracción de su potencial de funcionamiento, por lo que se está desaprovechando el 70% de sus recursos o capacidad. Con la virtualización, se puede dividir el servidor de correo en 3 servidores virtuales donde corran los tres servicios de manera independiente aprovechando así toda su capacidad (Figura 27). Es el mismo hardware, solo que se está usando de manera más eficiente, de modo que los otros dos servidores, ahora vacíos, podrían reutilizarse para otras tareas o retirarse por completo reduciendo costes de mantenimiento.

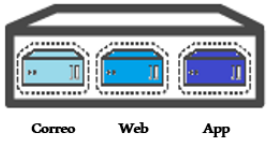


Figura27. Ejemplo de 3 VMs en un servidor físico

5.2 ¿Cómo funciona?

Un uso clave de la tecnología de virtualización es la virtualización de servidores, que utiliza una capa de software, llamada hipervisor, para emular el hardware subyacente. Esto a menudo incluye la memoria de la CPU, la entrada/salida (E/S) y el tráfico de red. Los hipervisores toman los recursos físicos y los separan para que puedan ser utilizados por el entorno virtual. Pueden colocarse encima de un sistema operativo o pueden instalarse directamente en el hardware, esta es la forma en que la mayoría de las empresas virtualizan sus sistemas.

El **hipervisor** es un software que se encarga de administrar las interacciones de bajo nivel que se producen entre las máquinas virtuales (VM) y el hardware físico. En otras palabras, el hipervisor permite la creación, ejecución y gestión simultáneas de varias máquinas virtuales en un entorno físico. Con la ayuda del hipervisor, el sistema operativo invitado, que normalmente interactúa con el hardware, ahora lo hace con una emulación de software de ese hardware. Si bien el rendimiento de este sistema virtual no es igual al rendimiento del sistema operativo que se ejecuta en hardware real, la tecnología funciona porque la mayoría de los sistemas operativos y aplicaciones invitados no necesitan el uso completo del hardware subyacente. Esto permite una mayor flexibilidad, control y aislamiento al eliminar la dependencia de una plataforma del hardware. El proceso de virtualización sigue los pasos que se enumeran a continuación:

1. Los hipervisores separan los recursos (CPU, memoria, etc.) de su entorno físicos.
2. Los recursos se toman y se dividen, según sea necesario, desde el entorno físico hasta los diversos entornos virtuales.
3. Los usuarios del sistema trabajan y realizan cálculos dentro del entorno virtual.
4. Una vez que se está ejecutando el entorno virtual, un usuario o programa puede enviar una instrucción que requiera recursos adicionales del entorno físico. En respuesta, el hipervisor transmite el mensaje al sistema físico y almacena los cambios, es un sistema que puede ser totalmente dinámico ofreciendo así los recursos que necesita una máquina en el momento que los necesita.

La máquina virtual funciona como un solo archivo de datos y, como cualquier archivo digital, se puede mover de una maquina a otra, abrir en cualquiera de ellas, funcionando de la misma manera.

5.3 Tipos de virtualización

1. **Virtualización de datos**: Los datos que se distribuyen por todas partes se pueden consolidar en una sola fuente. La virtualización de datos permite a las empresas tratar los datos manera dinámica, proporcionando capacidades de procesamiento que pueden reunir datos de múltiples fuentes, añadir fácilmente nuevas fuentes y transformarlos de acuerdo con las necesidades del usuario, permitiendo la entrega de los datos necesarios en el momento adecuado para cualquier aplicación o usuario.
2. **Virtualización de escritorio**: permite a un administrador central (o herramienta de administración automatizada) implementar entornos de escritorio simulados

en cientos de máquinas físicas a la vez. A diferencia de los entornos de escritorio tradicionales que se instalan, configuran y actualizan físicamente en cada máquina, la virtualización de escritorios permite a los administradores realizar configuraciones masivas, actualizaciones y controles de seguridad en todos los escritorios.

3. **Virtualización de servidores**: Los servidores son maquinas diseñadas para procesar un gran volumen de tareas específicas con un alto rendimiento. La virtualización de un servidor permite realizar más de estas funciones específicas e implica particionarlo para que los componentes se puedan utilizar para realizar múltiples funciones.
4. **Virtualización de Sistema Operativo**: La virtualización del sistema operativo ocurre en el kernel, el administrador de tareas central de los sistemas operativos. Es una forma útil de ejecutar entornos Linux y Windows en paralelo y ofrece las siguientes ventajas:
    a. Reducción de costes (al igual que en los otros tipos).
    b. Aumenta la seguridad, ya que todas las instancias virtuales se pueden monitorear y aislar.
    c. Limita el tiempo dedicado a servicios de TI como actualizaciones de software.
5. **Virtualización de funciones de red**: La virtualización de funciones de red (NFV) separa las funciones clave de una red (como servicios de directorio, uso compartido de archivos y configuración de IP) para que puedan distribuirse entre entornos. Una vez que las funciones de software son independientes de las máquinas físicas, las funciones específicas se pueden empaquetar en una nueva red y asignar a un entorno. La virtualización de redes reduce la cantidad de componentes físicos, como conmutadores, enrutadores, servidores, cables y concentradores, que se necesitan para crear redes múltiples e independientes, y es particularmente popular en la industria de las telecomunicaciones. Este es el tipo de virtualización que afecta principalmente a los nodos que componen la NGN-IMS y que se desarrollará a continuación.

5.4 VNF y SDN

5.4.1 VNF

El principal objetivo de NFV o Network Function Virtualization es el de reemplazar el hardware que tiene un propósito específico por el uso de hardware de propósito general sobre el que se montan o crean las VM con las distintas funciones de red; SDN y VNF van de la mano en este sentido. Mediante la virtualización de funciones de red se reducen los costes y acelera la implementación de servicios para los operadores de red, al desacoplar funciones como un firewall o cifrado del hardware dedicado y trasladarlas a servidores virtuales.

Antes de la virtualización cuando se quería montar un nodo de la red, por ejemplo, un CSCF, era necesario montar un equipo físico, especifico y exclusivo para ese nodo de manera que el software o la aplicación estaba vinculada completamente al hardware. La VNF, apoyándose sobre NFV, hace posible tener varias funciones o aplicaciones de red virtualizadas corriendo en un mismo equipo físico de propósito general capaz de ofrecer los recursos necesarios (memoria, CPU, almacenamiento, etc.). Si un cliente desea agregar una nueva función de red, el proveedor de servicios puede simplemente poner en marcha una nueva máquina virtual para realizar esa función sin necesidad de implementar un nuevo dispositivo de hardware.

Este nuevo enfoque o modelo reduce la dependencia de dispositivos de hardware dedicados para los operadores otorgando una gran escalabilidad, agilidad y eficiencia.

BENEFICIOS

- Espacio reducido para hardware.
- Reducción del consumo de energía.
- Reducción de los costes de mantenimiento.
- Actualizaciones de red más fáciles.
- Ciclos de vida más largos para el hardware.
- Reducción de los costes de hardware.

5.4.2 SDN

Las redes definidas por software (SDN) son un enfoque arquitectónico de la red que permite ser controlada de manera inteligente y central, o "programada", mediante aplicaciones de software que utilizan API abiertas. Esto ayuda a que los operadores gestionen toda la red de manera constante e integral, independientemente de la tecnología de red subyacente. SDN consta de tecnologías que estarán definidas por un controlador, el cual centralizará mediante software la inteligencia de todas las funciones de las capas de la red dentro de un mismo dispositivo, logrando que sean programables de una manera más ágil y fácil.

Hay cuatro áreas críticas en las que la tecnología SDN puede marcar la diferencia para una organización:

1. **Programabilidad de la red**: al desacoplar el hardware del software, los operadores pueden introducir nuevos servicios diferenciados e innovadores rápidamente, libre de las limitaciones de las plataformas cerradas y patentadas.
2. **Centralizar inteligencia y control lógicamente**: Con el tipo de control centralizado que ofrece una red basada en SDN, la administración, la restauración, la seguridad y las políticas de ancho de banda pueden ser muy inteligentes y optimizadas permitiendo a una organización obtener una visión integral de la red.

3. **Abstracción de la red**: Los servicios y las aplicaciones que se ejecutan en la tecnología SDN se abstraen de las tecnologías subyacentes y del hardware que proporcionan conectividad física de control de la red. Las aplicaciones van a interactuar con la red a través de las API, en lugar de las interfaces de administración estrechamente acopladas al hardware.
4. **Apertura**: las arquitecturas de SDN dan paso o permiten la interoperabilidad de múltiples proveedores.

Las SDN conocidas también como redes programables y automatizadas se presentan como la novedad que brinda una mayor velocidad, una infraestructura más ágil y mejores costes en plataformas cloud IT, dando respuesta al dinamismo y permitiendo adaptarse de manera natural a un entorno en continuo cambio. Es importante saber que, en una red definida por software, el administrador de red puede darle forma al tráfico desde una consola de control centralizada sin tener que tocar conmutadores individuales y este puede cambiar cualquier regla de los conmutadores de red cuando estos lo requieran.

La principal premisa es la de llevar a cabo un desacoplamiento total entre el plano de control y el plano de datos, donde para el plano de control se emplearán tecnologías más propias de las telecomunicaciones y redes y en el plano de datos habrá tecnologías que permitan realizar una gestión dinámica de la información con la que se está trabajando, cambiando la manera tradicional de configuración de dispositivos, usando instrucciones de bajo nivel, por una manera novedosa, mediante herramientas software un alto nivel.

Este paradigma surge principalmente por las limitaciones que tienen las tecnologías de redes actualmente. Las dos principales son:

- **Dificultad de escalabilidad**: A medida que el tráfico de una red aumenta, la red debe hacer lo mismo. Esto implica nuevos dispositivos que deben ser configurados y gestionados. La cantidad de dispositivos de una red aumenta de forma progresiva, y debido a ello, la configuración y gestión tradicionales se convierten en un mecanismo tedioso, propenso a errores y poco sostenible a largo plazo.
- **Dependencia del fabricante**: Cada fabricante diseña sus dispositivos de red de una manera particular, con su propio sistema operativo y su propia sintaxis de configuración.

En una red de telecomunicaciones es habitual que su infraestructura esté compuesta por dispositivos de diferentes vendedores y fabricantes, y es necesario conocer a fondo cada uno, haciendo de las tareas de configuración y mantenimiento algo muy complejo. A continuación, se muestran los componentes principales y la arquitectura de SDN:

- **Controlador SDN**: es el cerebro, forma el núcleo de la arquitectura SDN comunicándose con los switches a través de la SBI (SouthBound Interface) y con las distintas aplicaciones a través de la NBI (NorthBound Interface).

    - SBI: conecta el controlador SDN con el plano de datos facilitando la configuración de la red, transfiriendo dichas configuraciones a los dispositivos mediante, generalmente, el protocolo OpenFlow.
    - NBI: se encarga del proceso de automatización de la red mediante la comunicación del controlador con las diferentes aplicaciones a través de una interfaz REST.
- **Aplicaciones SDN**: Son programas o aplicaciones que se conectan al controlador a través de la NBI, las cuales tienen la lógica que desarrollan un propósito concreto y que transfiere los datos u órdenes al controlador SDN. Algunas de sus aplicaciones serían:
    - **Seguridad**: son capaces de detectar determinados patrones de tráfico y mitigar ataques.
    - **Simplicidad**: permitiendo la configuración de la red a alto nivel.
    - **Enrutamiento adaptativo**.

A modo de resumen, la capa de aplicación es la encargada de transmitir a la capa controladora las necesidades y el comportamiento esperado o deseado de la red y esta segunda toma las decisiones pertinentes para gestionar y adaptar la red de modo que cumpla con los requisitos que ha solicitado la aplicación. Esto se consigue mediante la configuración de unas tablas de flujos que contienen las directrices que sigue la capa o plano de datos para el reenvío de la información.

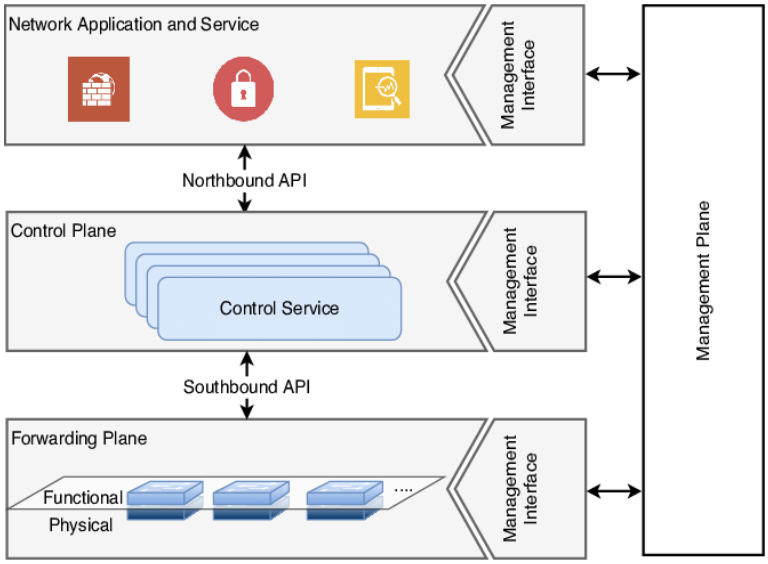


Figura28. Capas arquitectura SND [22]

PLANO DE DATOS

Es la primera capa o capa más baja, encargada del envío y transmisión de paquetes, es decir, su tarea es la transmisión de paquetes que, mediante unas tablas llamadas Flow Tables (se pueden agrupar en Group Tables), toma una serie de acciones para el enrutamiento y saber cómo dirigirlos. Se puede entender como el esqueleto de la tecnología SDN encargado de la transmisión de los paquetes entre los distintos elementos de la red (host, switch, routers, etc.). Para entender mejor su

comportamiento, se muestra el diagrama de flujo que se sigue en el envío de los paquetes:

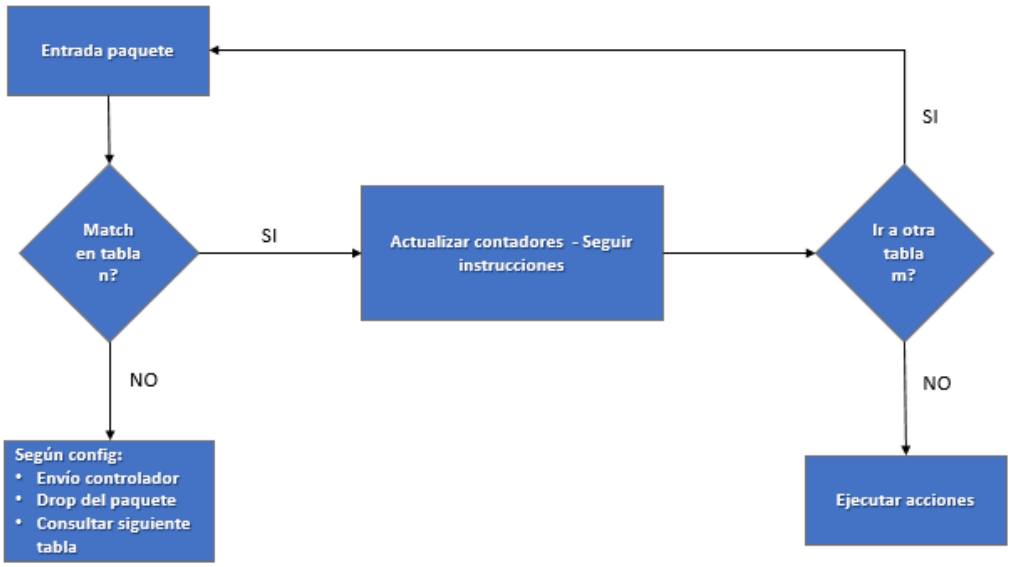


Figura29. Diagrama de flujo SDN

Según puede apreciarse hay distintos tipos de configuración permitiendo así adaptarse a las circunstancias y necesidades del momento. Por ejemplo, si hubiese congestión en la red se podría hacer que todos los paquetes que lleguen al plano de datos y que no encuentren ninguna coincidencia en las Flow Tables se descarte directamente para evitar saturarla.

PLANO DE CONTROL

Es donde reside la lógica de la red en SDN y, por lo tanto, es el lugar donde se define el comportamiento de la misma. Existen varios controles SDN y para el laboratorio se va a utilizar POX ya que cuenta con bastantes ejemplos de uso y documentación. Generalmente, el controlador está compuesto por diferentes módulos que sirven para un propósito concreto y donde se establece un orden por el cual unos módulos llaman a otros para conseguir que el comportamiento de la red sea el esperado. Ya que quizá sea el plano más crítico de la solución, es conveniente contar con varias máquinas que desempeñen la función de control, no solo como contingencia en caso de que falle uno de los controladores sino también para mejorar la eficiencia y evitar congestiones en las distintas zonas en las que puede estar dividida la red.

PLANO DE GESTIÓN

Es el plano más transversal y tiene una visión más general y amplia del estado en el que se encuentra la red y puede intercambiar información con otras redes colindantes.

Hay dos razones que le han dado un mayor reconocimiento a SDN.

- **Uso de la nube**: SDN logra que los servicios basados en la nube se tomen como un dispositivo más de red.

- **Capacidad de programación del usuario**: OpenFlow trabaja con Python y otros lenguajes de programación, por lo que el administrador puede definir sus propias herramientas y protocolos sobre cómo se comporta la red y cómo responden a los cambios.

5.4.3 NFV-VNF y su relación con SDN

NFV es el concepto general de ejecutar las funciones de red mientras que las VNFs es el resultado lógico donde se separa el software del hardware subyacente. Las VNFs están formadas por máquinas virtuales, VMs, que se ejecutan software sobre la infraestructura NFV (NFVi), dicha infraestructura está compuesta por hardware genérico donde se mete la capa de virtualización. Esta implementación permite reducir los costes mejorando la eficiencia.

Como se ha ido explicando partimos de sistemas o redes con problemas de escalabilidad, flexibilidad y, como consecuencia, innovación.

¿Solución? VIRTUALIZACIÓN. La virtualización de las redes se basa principalmente en las dos tecnologías vistas y, a continuación, se explica cómo trabajan conjuntamente o se apoya la una en la otra.

NFV

Generalmente se cuenta con un pool de servidores sobre los que se monta un hipervisor para poder gestionarlo y desplegar las máquinas virtuales que tienen o desempeñan un propósito concreto. Con la interconexión de las VMs es cuando se consigue formar una función de red y cuando se conectan distintas funciones de red (CSCF, DNS, HSS, etc.) se consigue ofrecer un servicio completo.

¿Quién organiza todo esto para tener un escenario de red real y formar los servicios? El **orquestador MANO (MAnagemet and Network Orquestation)**, arbitra el uso de recursos y de las redes comunes en un entorno de virtualización, despliega y monitoriza las apps virtuales de red y realiza las operaciones de escalado de las VNFs (Figura 30).

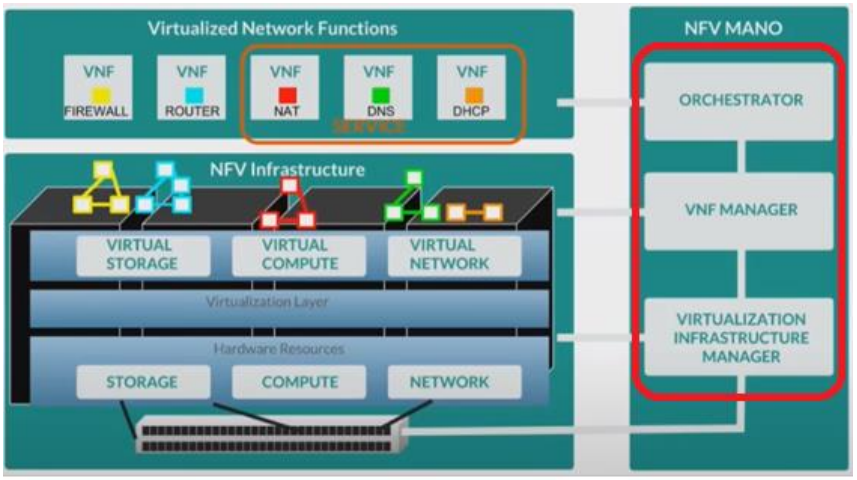


Figura30. Esquema NFV+SDN (MANO) [23]

- **Virtualization Infraestructure Manager (VIM)**: gestiona los recursos de una NFVI en “un dominio” - Gestiona el ciclo de vida de los recursos en un dominio NFVi (crear, mantener y eliminar VMs). Este componente solo entiende a nivel de máquina virtual, no sabe qué va por encima a nivel de aplicación. Además, se encarga de mantener el inventario de VMs asociadas a los recursos físicos junto a la gestión de rendimiento y fallos en el hardware, software y recursos virtuales.

- **VNF Manager**: gestiona el ciclo de vida de una VNF (crea, mantiene y elimina instancias VNF). Este componente ya no solo ve o entiende de VMs aisladas, sino que gestiona el conjunto de VMs que interconectadas entre si forman una determinada función de red, es decir, sabe qué tiene que crear, qué VMs interconectadas entre sí, para poder crear una función de red concreta.
  Es responsable del FCAPS de las VNFs (Fault, configuration, accounting, performance and Security Management) y permite escalar las VNFs aumentando la asignación de recursos de CPU, memoria, etc. en función de las necesidades. Si, por ejemplo, hay una VNF que está saturada, el VNF Manager se encarga de ampliarle los recursos y si, por el contrario, una función de red ya no se usa es capaz de eliminarla de modo que se puedan utilizar esos recursos para otro elemento de red.

- **OSCHESTRATOR**: Orquesta servicios y recursos. El orquestador de recursos: coordina, autoriza, comunica y emplea recursos NFVI entre diferentes infraestructuras a través del VIM correspondiente. Puede hablar con distintos VIMs, dominios, para comunicar uno con otro o desplegar lo que hay en uno en el otro.

Las interacciones a la hora de crear una instancia de una nueva función de red virtualizada sería la siguiente (Figura 31):

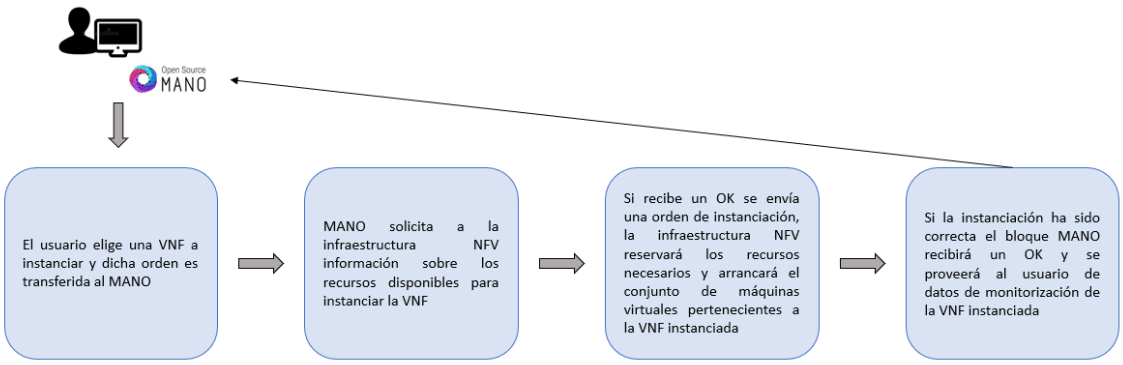


Figura31. Flujo al crear VNF en NFVI

SDN en NFVi

En la arquitectura que conforma la solución NFVi el pool de servidores donde se crean las máquinas virtuales que conforman las VNFs están conectados a través de unos switch o router y es aquí donde se aplica la solución de SDN que permite la configuración de forma dinámica del esos switch permitiendo la creación de redes y las conexiones entre máquinas virtuales necesarias.

A través del controlador se puede configurar el router o el switch para que funcione como se quiere. El módulo VIM de NFVI se encarga de hablar o comunicarse con el controlador de SDN y es capaz de especificarle cómo es la configuración necesaria para que las VMs y VNFs puedan hablar entre ellas, generalmente mediante configuración de VLANs.

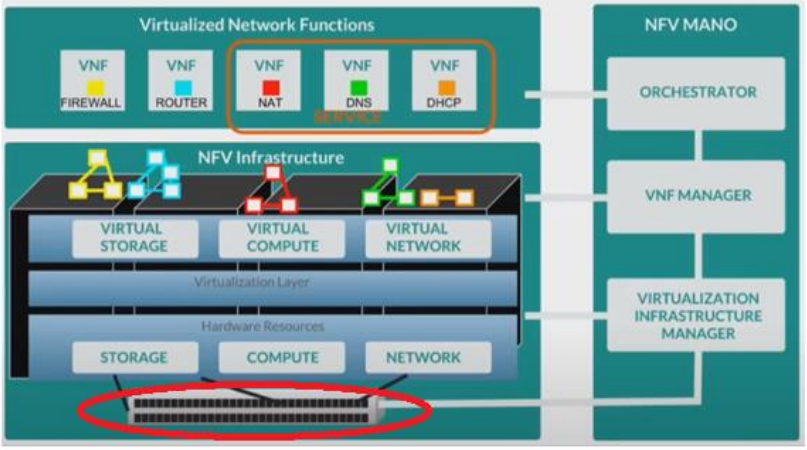


Figura32. Esquema NFV+SDN (SDN) [24]

SDN es un conjunto de técnicas para agilizar la gestión y configuración de redes de telecomunicación gracias a herramientas software. En cambio, NFV es un paradigma que persigue un cambio total en la infraestructura de una red de telecomunicación mediante el uso de funciones de red virtualizadas que sustituyan a los dispositivos físicos. Gracias a su uso conjunto de ambas tecnologías, ambos objetivos han convergido a uno solo, el de crear el concepto de Network Service Chaining (NSC).

5.5 Contenedores

Comenzar explicando algunos de los motivos por los que es conveniente usar contendores frente a otros métodos de virtualización y que ha empujado al desarrollo y evolución de esa tecnología.

Por ejemplo, Google ejecuta todo su servicio o motor de búsqueda en contenedores lo cual permite implementar software de manera eficaz, operando a una escala sin precedentes. En lugar de virtualizar la pila de hardware como en el enfoque de las máquinas virtuales, los contenedores realizan la virtualización en el nivel del sistema operativo, con varios contenedores que se ejecutan directamente en el kernel del SO. Esto significa que los contenedores son mucho más livianos, se inician con bastante más rapidez y usan una fracción de la memoria en comparación con el arranque de todo un SO. Algunas de sus características son:

- **Entorno uniforme**: una de las virtudes de los contenedores es ofrecer la capacidad de crear entornos controlados aislados de otras aplicaciones. Los contenedores incluyen, entre otras cosas, dependencias de software que la aplicación necesita junto con librerías de software lo que permite al desarrollador crear un entorno uniforme, independiente y que garantiza el mismo comportamiento o funcionamiento en cualquier entorno donde se

despliegue. Los contenedores se pueden ejecutar virtualmente en cualquier lugar, lo cual facilita en gran medida el desarrollo y la implementación en los sistemas operativos Linux, Windows y Mac; en máquinas virtuales o equipos físicos; en la máquina del desarrollador o en centros de datos locales, y, por supuesto, en la nube pública.
Esto se traduce en productividad ya que los equipos de operaciones de TI no necesitan dedicar tanto tiempo y recursos a depurar y diagnosticar fallos y diferencias entre entornos (dependencias, librerías, versiones, etc.).

- **Aislamiento**: Los contenedores virtualizan la CPU, la memoria, el almacenamiento y los recursos de red al nivel del SO, lo que brinda a los desarrolladores una vista en zonas de prueba del SO aislada lógicamente de otras aplicaciones.

	Beneficios de los contenedores	Beneficios de las máquinas virtuales
Entorno de tiempo de ejecución uniforme	✓	✓
Zonas de pruebas para aplicaciones	✓	✓
Tamaño pequeño en el disco	✓	
Baja sobrecarga	✓	

Tabla5. Comparativa contenedores vs VMs

5.5.1 Conceptos

Los contenedores ofrecen un mecanismo de empaquetado lógico en el que las aplicaciones se pueden abstraer del entorno en el que se ejecutan. Esta separación permite que las aplicaciones basadas en contenedores se implementen de manera fácil y uniforme, independientemente de si el entorno objetivo es un centro de datos privado, la nube pública o incluso un PC personal. La creación de contenedores ofrece una separación clara, haciendo que los desarrolladores se puedan enfocar en la lógica y las dependencias de las aplicaciones, mientras que los equipos de Operaciones de TI se pueden concentrar en la implementación y la administración, sin preocuparse por los detalles de las aplicaciones, como las versiones específicas del software y las configuraciones específicas de la aplicación.

Aunque contenedores y máquinas virtuales pertenecen ambas al mundo o tecnología de la virtualización son conceptos completamente diferentes ya que una máquina virtual es un sistema operativo completo funcionando de manera aislada sobre otro sistema operativo completo, la tecnología de VMs permite compartir el hardware de modo que lo puedan utilizar varios sistemas operativos al mismo tiempo tal y como ya se ha explicado en el apartado anterior. Sin embargo, la filosofía de los contenedores es totalmente diferente a la de las VMs, tratan también de aislar a las aplicaciones y generar un entorno replicable y estable para que funcione corectamente, pero, en lugar de albergar un sistema operativo completo, lo que hacen es compartir los recursos del propio sistema operativo "host" sobre el que se ejecutan (Figura 33).

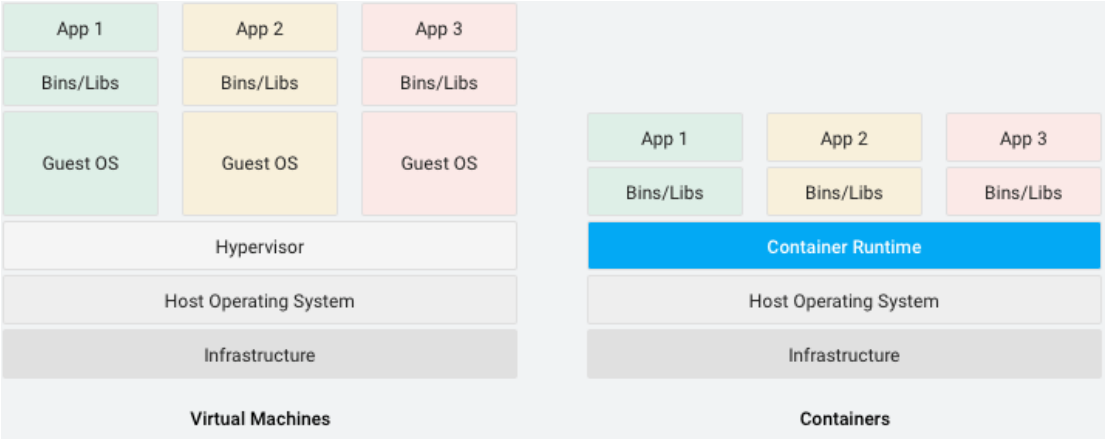


Figura33. Capas en VM y contenedores [25]

El objetivo principal que hay detrás de todo es el de **pasar de una arquitectura monolítica a una arquitectura basada en servicios**.

Beneficios

Los contenedores se han vuelto populares porque brindan beneficios adicionales, como:

- Creación e implementación de aplicaciones ágiles: mayor facilidad y eficiencia de la creación de imágenes de contenedores en comparación con el uso de imágenes de VM.
- Desarrollo, integración e implementación continuos: proporciona una construcción e implementación de imágenes de contenedores frecuentes y confiables con reversiones rápidas y eficientes (debido a la inmutabilidad de la imagen).
- Consistencia: funciona de la misma manera en una computadora portátil que en la nube.
- Portabilidad de distribución de SO y nube: se ejecuta en Ubuntu, RHEL, CoreOS, en las instalaciones, en las principales nubes públicas y en cualquier otro lugar.
- Gestión centrada en aplicaciones: aumenta el nivel de abstracción desde la ejecución de un sistema operativo en hardware virtual hasta la ejecución de una aplicación en un sistema operativo utilizando recursos lógicos.
- Aislamiento de recursos: rendimiento predecible de la aplicación.
- Utilización de recursos: alta eficiencia y densidad.

5.5.2 Organización de contenedores

Para la puesta en marcha de un entorno basado en contenedores es necesario el empleo de herramientas que permitan su organización y gestión. Casi tan importante es conocer el contendedor como el saber gestionarlos por lo que a continuación se muestran dos de las principales plataformas para este fin y que son las más extendidas actualmente.

Docker

Comenzar explicando que Docker ha sido desarrollado en el lenguaje de programación Go y aprovecha varias características del Kernel de Linux para ofrecer su servicio o funcionalidad. Utiliza la tecnología de namespaces (es una característica del kernel de Linux que divide los recursos del kernel de modo que un conjunto de procesos ve un conjunto de recursos mientras que otro conjunto de procesos ve otro conjunto diferente de recursos) para ofrecer un espacio de trabajo aislado y seguro denominado contenedor. Es decir, cuando se crea un contenedor se crea un conjunto de namespaces para dicho contenedor lo que proporciona una capa de aislamiento, permitiendo que un contenedor que se ejecuta en un namespace solo pueda acceder a ese namespace [35].

¿En qué cosiste?

Docker es una plataforma abierta capaz de desarrollar, enviar y ejecutar aplicaciones. Permite separar las aplicaciones de su infraestructura de modo que el despliegue del software sea mucho más rápido y eficiente tanto en entornos de prueba como de producción.

Esto es posible ya que las aplicaciones se empaquetan y ejecutan dentro de un contenedor aislado que contiene todo lo necesario para ejecutar la aplicación (librerías, frameworks, etc.) sin depender de lo que esté instalado en el host donde se encuentre. Estos contenedores son muy livianos en cuanto a carga o volumen, ocupando generalmente unas decenas o centenas de MB.

ARQUITECTURA

La arquitectura en la que se basa es cliente – servidor de manera que el cliente Docker habla con el servidor o demonio Docker (dockerd), encargado de realizar las tareas o funciones de construir, ejercitar y distribuir los contenedores. Ambos roles se pueden encontrar en el mismo sistema o se pueden comunicar de manera remota, el cliente se encarga de mandar las peticiones mediante una API REST (utiliza comandos REST) a través de los sockets de UNIX o de una interfaz de red al servidor.

Docker Daemon

El demonio recibe las peticiones o solicitudes a través de la API y se encarga de administrar los objetos de Docker como son: imágenes, contenedores, redes y volúmenes. Un demonio Docker tiene la posibilidad de comunicarse con otros demonios para administrar servicios de cierta envergadura o complejidad.

Docker Client

Es la manera habitual o principal mediante la cual los usuarios hacen uso de la plataforma e interactúan con Docker. El cliente ofrece una serie de comandos mediante los cuales se pueden realizar acciones para el uso de la plataforma y la gestión o administración de los contenedores. Por ejemplo, mediante el comando *Docker run*, el cliente envía una serie de comandos a dockerd y este se encarga de ejecutarlos; el cliente Docker es capaz de comunicarse con uno o varios demonios.

Registro de Docker

El registro de Docker contiene o almacena imágenes de Docker preparadas para ser desplegadas en Docker. Docker Hub, es un registro público al cual cualquier persona puede acceder y que contiene imágenes de Docker (la plataforma está configurada para que, por defecto, busque imágenes en este "repositorio").

Al utilizar, mediante el cliente, el comando *Docker pull* o *Docker push*, la plataforma de encarga de buscar las imágenes que necesita o que quiere exportar al registro que tiene configurado.

Objetos Docker

Con la ejecución de los comandos que ofrece la plataforma y al construir un entorno dockerizado se están creando y usando imágenes, contenedores, redes, volúmenes, complementos y otros objetos. A continuación, se describen brevemente algunos de los objetos más comunes:

- **IMÁGENES:** es una plantilla con las instrucciones necesarias para crear un contenedor Docker. Es habitual que sobre una imagen se realice alguna modificación o personalización, por ejemplo, metiendo alguna aplicación web sobre una imagen y de ahí crear otra imagen nueva. Es decir, sobre una plantilla o imagen se pueden añadir nuevos componentes, como java, y crear una nueva imagen customizada.

En este punto conviene hacer un paréntesis y explicar un tipo de fichero imprescindible en Docker para el correcto funcionamiento de la plataforma, **Dockerfiles**.

Dockerfiles

Un DockerFile es un documento de texto que contiene todos los comandos necesarios para armar una imagen. Esta imagen se creará mediante el comando docker build que irá siguiendo las instrucciones. El comando arma una imágen siguiendo las instrucciones del DockerFile, la creación de la imagen es ejecutada por el daemon de Docker. El demonio corre las instrucciones del Dockerfile línea por línea y va lanzando los resultados en pantalla. Un punto importante es que cada instrucción es ejecutada en nuevas imágenes, hasta que muestra el ID de la imágen resultante una vez finalizada las instrucciones, el daemon irá haciendo una limpieza automáticamente de las imágenes intermedias. Durante el despliegue de la imagen en el contender se van creando imágenes intermedias hasta que finalmente devuelve el ID de la imagen definitiva (este punto es importante a la hora de ejecutar las instrucciones).

FORMATO

FROM

FROM indica la imagen base que va a utilizar para seguir futuras instrucciones. Buscará si la imagen se encuentra localmente, en el caso de que no, la descargará de internet.

MAINTAINER

Esta instrucción nos permite configurar datos del autor que genera la imágen.

RUN

Esta instrucción ejecuta cualquier comando en una capa nueva encima de una imágen y hace un commit de los resultados. Esa nueva imágen intermedia es usada para el siguiente paso en el Dockerfile.

ENV

Esta instrucción configura las variables de ambiente, estos valores estarán en los ambientes de todos los comandos que sigan en el DockerFile. Pueden por igual ser sustituidos en una linea.

ADD

Esta instrucción copia los archivos, directorios de una ubicación especificada y los agrega al sistema de archivos del contenedor en la ruta especificada.

EXPOSE

Esta instrucción le específica a Docker que el contenedor escucha en los puertos especificados en su ejecución. EXPOSE no hace que los puertos sean accesibles desde el host, para eso debemos mapear los puertos usando la opción -p en docker run

CMD

Solo puede existir una instrucción CMD en un DockerFile, si se coloca más de una, solo la última tendrá efecto. El objetivo de esta instrucción es proveer valores por defecto a un contenedor. Estos valores pueden incluir un ejecutable u omitir un ejecutable, en cuyo caso se debe especificar un punto de entrada o ENTRYPOINT en las instrucciones.

ENTRYPOINT

Cualquier argumento que pasemos en la línea de comandos mediante docker run será anexado y anulará cualquier elemento especificado con CMD, esto permite pasar cualquier argumento al punto de entrada.

En este ejemplo se correrá un httpd, en modo verbose, en el ENTRYPOINT, y los argumentos que pueden cambiar en CMD, puerto 80. Si se quisiera correr el contenedor con httpd en el puerto 8080, solo se tendría que ejecutar docker run centos:centos7 -p 8080.

VOLUME

Esta instrucción crea un punto de montaje con un nombre específico y lo marca con un volumen montado externamente desde el host u otro contenedor. El valor puede ser pasado en formato JSON o argumento plano.

WORKDIR

Está instrucción configura el directorio de trabajo para cualquier instrucción **RUN, CMD, ENTRYPOINT, COPY o ADD** en un DockerFile. Puede ser usada varias veces dentro de un DockerFile. Si se da una ruta relativa, esta será la ruta relativa de la instrucción WORKDIR anterior.

Para construir una imagen es necesario crear un fichero DockerFile con la sintaxis comentada anteriormente que contenga los pasos necesarios para poder crear la imagen y ejecutarla. Con cada instrucción del DockerFile se va construyendo una capa en la imagen de modo que cuando se modifica el fichero y se reconstruye la imagen, solo se reconstruye las capas que se han cambiado, esto hace que sea un proceso mucho más ligero y rápido que otras tecnologías de virtualización.

- **CONTENEDORES**: los contenedores son instancias ejecutables de una imagen. Es posible crear, iniciar o arrancar, detener o incluso mover un contenedor mediante una API o el Command Line Interface de Docker. En principio un contenedor está securizado y aislado con respecto a los demás contenedores tanto a nivel de red como de almacenamiento; lo que no quiere decir que varios contenedores se puedan comunicar entre si sino más bien que cada contenedor crea como una instancia única e independiente del resto de contenedores.

Posteriormente en las pruebas de laboratorio, se creará un contenedor con Asterisk permitiendo simular un entorno donde realizar pruebas de llamadas SIP, se verán en detalle los comandos y sus resultados. En este caso se hace uso de Docker-compose que facilita la creación de contenedores.

Kubernetes

Kubernetes es una plataforma portable y extensible de código abierto que permite administrar cargas de trabajo y servicios en contenedores, facilitando la configuración, despliegue y automatización [36].

Introducción

Kubernetes ofrece un entorno de administración centrado en contenedores que orquesta la infraestructura de cómputo o CPU, redes y almacenamiento. De este modo se ofrece la simplicidad de las Plataformas como Servicio (PaaS) junto con la flexibilidad de la Infraestructura como Servicio (IaaS), permitiendo la portabilidad entre diferentes proveedores de infraestructura. Kubernetes ha sido diseñado para poder construir un ecosistema de componentes y herramientas que hacen más fácil el desplegar, escalar y administrar aplicaciones.

En Kubernetes se utilizan los objetos de la API de Kubernetes para describir el estado deseado del clúster: qué aplicaciones u otras cargas de trabajo se quieren ejecutar, qué imágenes de contenedores usar, el número de replicas, configuración de red, qué recursos de almacenamiento asignar o reservar, etc. Se especifica el estado deseado del clúster mediante la creación de objetos usando la API de Kubernetes, típicamente mediante la interfaz de línea de comandos, *kubectl*. Una vez se ha especificado el estado

deseado del clúster coge el testigo el Plano de Control de Kubernetes, que realizará las acciones necesarias (parar o arrancar contenedores, escalar el número de réplicas de una aplicación dada, etc.) para que el clúster tenga o alcance el estado deseado. El **Plano de Control** de Kubernetes consiste en un grupo de daemons que corren en el clúster:

- El Máster es un conjunto de tres demonios que se ejecutan en un único nodo del clúster, nodo máster: kube-apiserver, kube-controller-manager y kube-scheduler.
- Los restantes nodos no máster, contenidos en el clúster, ejecutan los siguientes dos daemons: kubelet, el cual se comunica con el máster de Kubernetes, y kube-proxy, un proxy de red que implementa los servicios de red de Kubernetes en cada nodo, posteriormente se verá más en detalle cada componente.

Además, es relevante otro termino que se emplea en Kubernetes, y es básico para su entendimiento, los **objetos**. Son abstracciones que representan el estado del sistema: aplicaciones contenerizadas desplegadas, cargas de trabajo, recursos de red y almacenamiento e información adicional acerca de lo que el clúster está haciendo en cada momento. También, existen unas abstracciones de nivel superior llamadas **controladores**, se basan en los objetos básicos, y proporcionan funcionalidades adicionales sobre ellos.

Obj. Básico	Controlador
Pod Service Volume Namespace	ReplicaSet Deployment StatefulSet DaemonSet Job

Componentes principales

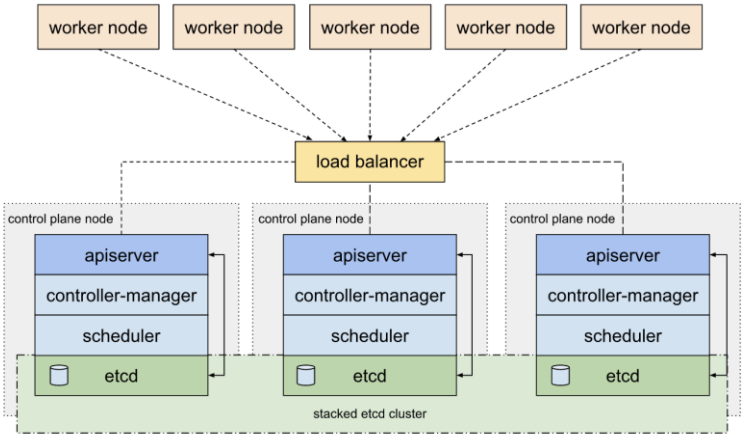


Figura34. Componentes principales de Kubernetes [26]

Plano de Control (toman decisiones globales sobre el clúster y responden a eventos)	**Componentes de nodo** (corren en cada nodo, manteniendo a los pods en funcionamiento y proporcionando el entorno de ejecución de Kubernetes)
kube-apiserver: expone la API de Kubernetes. Se trata del frontend de Kubernetes, recibe las peticiones y actualiza acordemente el estado en etcd (Almacén de datos persistente).	**Runtime de contenedores**: es el software responsable de ejecutar los contenedores. Kubernetes soporta varios de ellos: Docker, containerd,
kube-scheduler: está pendiente de los Pods que no tienen ningún nodo asignado y selecciona uno donde ejecutarlo (en función de diversos factores: requisitos de recursos, restricciones de hardware/software, etc.	**kube-proxy**: permite abstraer un servicio en Kubernetes manteniendo las reglas de red en el anfitrión y haciendo reenvío de conexiones.
kube-controller-manager: • Es el responsable de detectar y responder cuándo un nodo deja de funcionar • Es el responsable de mantener el número correcto de pods para cada controlador de replicación del sistema • hace una unión entre los Services y los Pods • Controladores de tokens y cuentas de servicio	**Kubelet**: Se asegura de que los contenedores estén corriendo en un Pod. Toma un conjunto de especificaciones de Pod, llamados PodSpecs, que han sido creados por Kubernetes y garantiza que los contenedores descritos en ellos estén funcionando y en buen estado.

Tabla6. Funciones de los principales componentes de Kubernetes. [27]

El núcleo del plano de control de Kubernetes es el servidor API que expone una interfaz HTTP y permite a los usuarios finales acceder a las diferentes partes del clúster y que los componentes externos se comuniquen entre sí (Figura 35). Permite consultar y manipular el estado de los objetos (por ejemplo: pods, espacios de nombres, ConfigMaps y eventos).

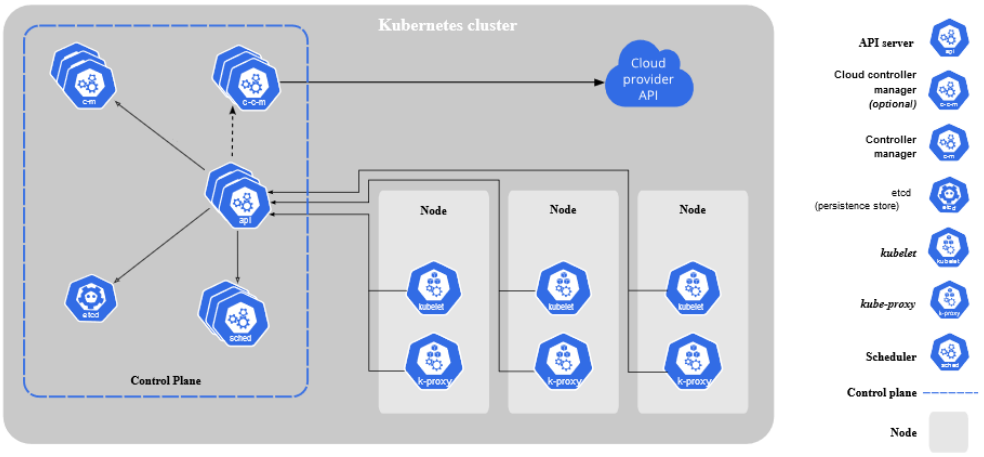


Figura35. Esquema de la API de Kubernetes [28]

Objetos de Kubernetes

Los objetos de kubernetes son entidades dentro de la plataforma y se utilizan para poder presentar el estado del clúster. Estas entidades describen:

- Qué aplicaciones contenerizadas se están ejecutando y en qué nodo.

- Los recursos que se están consumiendo.
- Las políticas de las aplicaciones, cómo restart/upgrade, tolerancia a fallos etc.

Se podría decir que cada objeto es un «registro de intención», es decir, con cada objeto que se crea se le está diciendo a Kubernetes como debería ser la carga de trabajo del clúster, es decir, el estado deseado.

Con cada objeto, que se define, existen dos campos anidados, el spec y el status. El campo spec describe como es el estado deseado y el campo status es el gestionado por el sistema de Kubernetes, es el estado actual del objeto el cual será guardado. En cualquier momento el pano de control verifica si el estado actual coincide con el estado deseado, en el caso en el que esto no sea así se toman medidas, decisiones.

Por ejemplo, si en un objeto Deployment, el cual sirve para desplegar una aplicación en el clúster, en el campo Spec se pone que se quieren tres réplicas, si en cualquier momento hubiese un fallo en alguna de las instancias el sistema de Kubernetes lo solucionaría instanciando una nueva.

Para poder definir cualquiera de estos objetos se puede realizar en JSON o en YAML. Una vez que se ha definido el fichero, será enviado al API server a través del Kubectl.

Tipos de Objeto

Pods

Un Pod es el objeto más simple y pequeño en Kubernetes, representa una sola instancia de la aplicación.

ReplicaSet

Utilizando un ReplicaSet, podemos escalar el número de Pods que ejecutan una imagen. Un ReplicaSet asegura que el número establecido de réplicas se están ejecutando en cualquier momento, es decir, mantener el estado deseado.

No es necesario administrar ReplicaSets y Pods por separado, la mejor aproximación es usar Deployments y configurar la replicación.

Deployments

El objetivo de un deployment es proporcionar una manera declarativa de actualizar Pods y ReplicaSets.

Docker vs Kubernetes

Tras haber visto las características de ambos conviene tener claras las diferencias entre ellos y, además, saber que no son plataformas excluyentes sino más bien todo lo contrario. Una manera eficaz de trabajar con contenedores y explotar todo su potencial es mediante el uso de Docker junto con Kubernetes.

Mientras que Docker proporciona un estándar abierto para empaquetar y distribuir aplicaciones en contenedores, crear/borrar/correr contenedores, Kubernetes se encarga de coordinar y orquestar este entorno. A continuación, se muestran las principales diferencias entre ambos:

- Docker se ejecuta en un solo nodo, mientras que Kubernetes está diseñado para ejecutarse en un clúster (en el laboratorio solo ha sido necesario el uso de Docker).
- Docker se puede usar sin Kubernetes, mientras que Kubernetes necesita un runtime de contenedor para poder utilizarse.
- Docker se ha convertido en el estándar para el desarrollo y la implementación de contenedores.
- Kubernetes se ha adoptado ampliamente y, en este punto, se ha convertido en el estándar de facto para la gestión y orquestación de contenedores.

Es necesario el uso de un orquestador como Kubernetes en entornos de producción complejos donde existe una gran cantidad de contenedores y una tarea fundamental es una correcta gestión y orquestación de los mismos. En los entornos de producción, a menudo, los beneficios de utilizar un orquestador de contenedores no superan el costo de la complejidad adicional.

5.5.3 Beneficios de utilizar contenedores (pto. vista empresarial empresarial)

Anteriormente se hablaba de las máquinas virtuales y los diferentes tipos de virtualización. A día de hoy la manera más eficiente de crear un entorno de producción y pruebas donde se monten los diferentes servicios es haciendo un uso combinado de máquinas virtuales y contenedores (Figura 36). Es decir, sobre una infraestructura de propósito general formada por servidores de alto rendimiento de van creando máquinas virtuales.

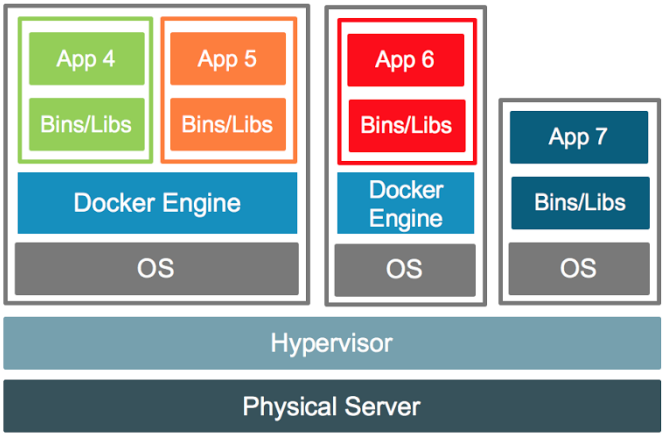


Figura36. Capas de una solución de contenedores sobre VM [29]

Se puede apreciar cómo, debido a la envergadura de los datacenter y los servicios que se ofrecen actualmente, cada vez es más relevante el empleo o uso de plataformas que ayuden a la orquestación del entorno y hacer un uso eficiente de los recursos. Según las

características que se han visto del 5G se requiere de un entorno que ofrezca un gran dinamismo por lo que, junto con demás beneficios que ofrece, la mejor opción es el uso de contenedores. Desde un punto de vista empresarial existen numerosos beneficios

1. La primera ventaja empresarial de usar contenedores es el **ROI** (el retorno de la inversión), el uso de contenedores ayuda a facilitar este tipo de ahorro reduciendo costes en recursos de infraestructura.
2. Los contenedores garantizan consistencia a lo largo del proceso de desarrollo y despliegue de una aplicación, estandarizando el entorno en el que se ejecuta. Una de las principales ventajas de esta arquitectura es precisamente la **estandarización**. Docker permite desarrollos, compilaciones, pruebas y entornos de producciones repetibles y replicables y sin necesidad de compartir enormes archivos. Esto reduce la cantidad de tiempo que se pierde encontrando bugs y aumenta el tiempo disponible para programar mejores y más funcionalidades.
3. **Erradica el problema de "en mi máquina funciona"** de una vez por todas. La paridad con las imágenes usadas en los contenedores significa que estas se ejecutan igual independientemente del servidor de la máquina física en la se ejecutan permitiendo dedicar menos tiempo configurando servidores, depurando errores específicos de problemas del entorno, etc.
4. Se consigue **reducir el despliegue** a de segundos. Esto se debe al hecho de que crea un contenedor para cada proceso y no arranca un sistema operativo.
5. En los últimos años, todos los proveedores principales de computación en la nube incluyendo Amazon Web Services (AWS), Microsoft Azure o Google Compute Platform (GCP), han adoptado la disponibilidad de Docker y Kubernetes, por lo que está **disponible en plataformas multi-cloud**.
6. Se garantiza que las **aplicaciones** estén **aisladas y segregadas**. Se asegura de que cada contenedor tenga sus propios recursos y que estén aislados de otros contenedores.

6. Laboratorio

6.1 LEARNING SWITCH – MININET y POX

En esta primera prueba se va a poner en práctica, a pequeña escala y sin utilizar todos los componente debido a las limitaciones de las herramientas gratuitas y OpenSource, lo explicado en el apartado de virtualización correspondiente a SDN donde de tratará de simular el comportamiento de aprendizaje arp de un switch virtual y como, mediante mecanismos a través de software, se crea una red que permite comunicar varios host para observar cómo de manera ágil, dinámica y eficiente se puede configurar una red ad hoc abstrayéndose de las limitaciones del hardware subyacente.

Para poder llevar a cabo la simulación ha sido necesario el uso de:

- MININET: es un programa capaz de crea una red virtual con el comportamiento de una red real, ejecutando kernel real, en una máquina (en este caso una VM Linux), en segundos [37].

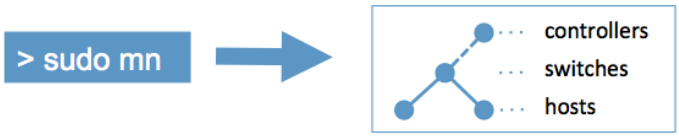


Figura37. Esquema de emulación MININET

  Es un emulador de red o más bien un sistema de orquestación de emulación de red. Ejecuta una colección de hosts finales, conmutadores, enrutadores y enlaces en un solo kernel de Linux utiliza virtualización ligera para hacer que un solo sistema parezca una red completa. Un host Mininet se comporta como una máquina real donde se puede usar ssh y ejecutar multitud de programas (incluido todo lo que esté instalado en el sistema Linux subyacente). Estos programas que ejecuta pueden enviar paquetes a través de lo que parece una interfaz ethernet real, con una velocidad de enlace y un retraso determinado.
  Es decir, los hosts virtuales, conmutadores, enlaces y controladores de Mininet son reales la diferencia es que se crean utilizando software en lugar de hardware y su comportamiento es similar al de los elementos hardware.
- Controlador POX: observando la figura anterior sobre MININET existe un componente 'controllers' y es aquí donde interviene POX. Es un controlador OpenFlow/Software Defined Networking (SDN) de código abierto basado en Python que se utiliza para un desarrollo más rápido permitiendo la creación de prototipos de nuevas aplicaciones de red. En este caso al hacer una instalación completa de MININET ya viene preinstalado el controlador POX. Este permite de una forma sencilla ejecutar experimentos crear dispositivos concentradores, conmutadores, balanceadores de carga y cortafuegos.

El esquema de escenario de red es el siguiente (dentro de una VM Linux Groovy):

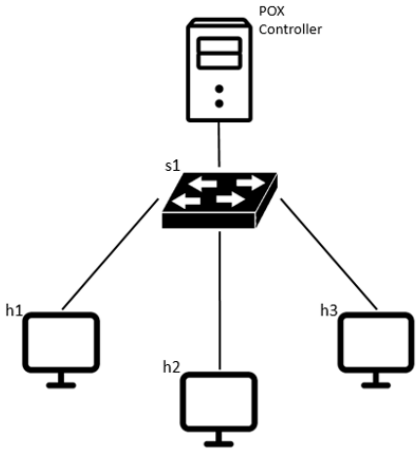


Figura38. Esquema red Controlador POX

En un primer lugar se crea el escenario de la red propuesta mediante MININET pasándole los parámetros necesarios para la simulación (ARP, MAC, Controller, etc) donde con 'Controller=remote' se indica que el controlador va a ser remoto (POX).

```
ubuntu@ubuntu2010:~/pox$ sudo mn --topo single,3 --mac --arp --switch ovsk --controller=remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Unable to contact the remote controller at 127.0.0.1:6633
Setting remote controller to 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X
h2 -> X X
h3 -> X X
*** Results: 100% dropped (0/6 received)
```

Figura39. Creación de escenario MININET y prueba "pingall"

Como se puede observar, al crear la red y hacer un 'pingall' entre los hosts no se alcanzan porque aún el switch no está configurado, no está arrancado el controlador POX con los datos.

Una vez arrancado se comprueba en un primer momento que entradas ARP tiene el Switch, es decir, ¿ha aprendido dónde se encuentra cada host? La respuesta es NO, debido a que no ha habido ningún intercambio de mensajes entre ellos, será cuando se realice un 'pingall' de nuevo cuando el switch inunde la red preguntando por todos los puertos que tiene configurados cuando sepa dónde está cada host, lo anote en la tabla y sepa llegar a cada uno:

```
ubuntu@ubuntu2010:~/pox$ sudo /home/ubuntu/pox/pox.py forwarding.l2_learning
POX 0.7.0 (gar) / Copyright 2011-2020 James McCauley, et al.
WARNING:version:Support for Python 3 is experimental.
INFO:core:POX 0.7.0 (gar) is up.
```

```
mininet> xterm h1 h2 h3
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=6.529s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth1",vlan_t
ci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:"s
1-eth2"
 cookie=0x0, duration=6.525s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth2",vlan_t
ci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:"s
1-eth1"
 cookie=0x0, duration=6.507s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth1",vlan_t
ci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:03,nw_src=10.0.0.1,nw_dst=10.0.0.3,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:"s
1-eth3"
 cookie=0x0, duration=6.500s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth3",vlan_t
ci=0x0000,dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.3,nw_dst=10.0.0.1,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:"s
1-eth1"
 cookie=0x0, duration=6.492s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth2",vlan_t
ci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:"s
1-eth1"
 cookie=0x0, duration=6.489s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth1",vlan_t
ci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:"s
1-eth2"
 cookie=0x0, duration=6.481s, table=0, n_packets=1, n_bytes=98, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth2",vlan_t
ci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:03,nw_src=10.0.0.2,nw_dst=10.0.0.3,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:"s
1-eth3"
```

Figura40-41. Carga Controlador POX – Tabla de flujos Switch (pingall)

Además, para comprobar el correcto funcionamiento se ha abierto con xterm un terminal en cada host y se ha realizado un ping del h1 → h2 (test2) ejecutando además un tcpdump en h2 y h3 para ver cómo llega la pregunta del switch para saber dónde está cada uno:

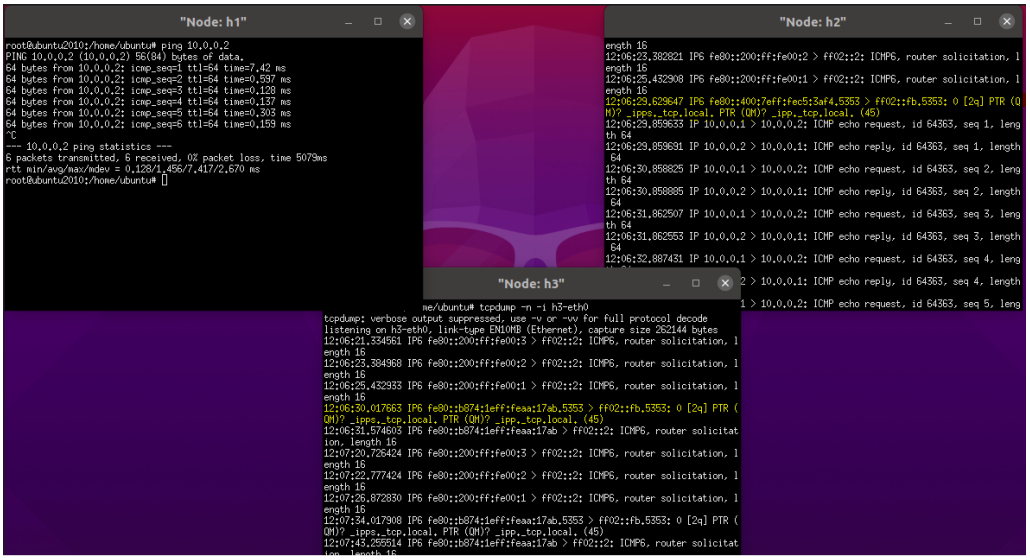


Figura42. TCPDUMP de cada host

Y una vez realizado el "descubrimiento" de por donde se llega a cada host se anota la ruta que interesa, en este caso h1 <-> h2:

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
 cookie=0x0, duration=2.448s, table=0, n_packets=3, n_bytes=294, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth1",vlan_
tci=0x0000,dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:02,nw_src=10.0.0.1,nw_dst=10.0.0.2,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:"
s1-eth2"
 cookie=0x0, duration=2.443s, table=0, n_packets=3, n_bytes=294, idle_timeout=10, hard_timeout=30, priority=65535,icmp,in_port="s1-eth2",vlan_
tci=0x0000,dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:01,nw_src=10.0.0.2,nw_dst=10.0.0.1,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:"
s1-eth1"
```

Figura43. Tabla de flujos Switch (ping h1-h2)

Por ver alguna posibilidad más se ha tirado el enlace s1-h3 para comprobar que desde ese momento el host 3 queda inalcanzable realizando un pingall.

Figura44. Prueba pingall con enlace h3 caído

En lo que respeta al módulo *l2_learning*, la prueba confirma el comportamiento que se esperaba donde el primer paquete que se envía acaba desencadenando que se instale una regla en el switch siguiendo los siguientes pasos:

1. En un primer momento el paquete no encuentra ninguna regla en el switch que encaje con él y se manda al controlador.
2. El controlador guarda en una tabla la dirección MAC ORIGEN del paquete junto a la interfaz de red por la que ha venido. Posteriormente, manda al switch hacer inundación del paquete.
3. Una vez el host destino recibe el paquete, responde con un paquete de contestación.
4. El nuevo paquete de contestación vuelve a no encontrar ninguna regla en el switch y es mandado al controlador.
5. El controlador se guarda la información del paquete y envía un mensaje al switch para que éste instale una regla en la flow table del switch con los parámetros pertinentes en ambos sentidos.
6. A partir de este momento, los paquetes posteriores al primer paquete van a encontrarse con una regla en el switch que lo trasmitirá a su destino directamente.

De este modo **comprobamos el comportamiento o funcionamiento descrito en el apartado de SDN, la agilidad y ventajas que ofrece**, así como la manera de crear una función de red sin la necesidad de disponer de un harware físico que lo soporte ni con las limitaciones que este, en algunos casos, viene. Esto hace posible que se pueda configurar, mediante un módulo de control, el manejo y creación de redes dinámicas adaptadas a las circunstancias o exigencias del momento y lo que va a ser una de las piezas fundamentales en el despliegue del 5G.

6.2 ESCENARIO NFV – MANO CON OPENSTACK Y OSM (FALLIDO)

A pesar de que el resultado no ha sido el esperado se va a explicar un entorno de testeo que se ha creado para poder comprobar el funcionamiento de NFVi y la creación de VNF y SDN de manera ágil. Para este escenario se han instalado las siguientes plataformas:

- Openstack: OpenStack es un proyecto open source enfocado al Cloud Computing, diseñado para desplegar nubes públicas o privadas orientadas a ofrecer infraestructuras como servicio a los usuarios. Está compuesto por siete módulos principales diferentes., cada uno proporciona herramientas para que

los administradores puedan gestionar los recursos (a través de la GUI o CLI). Como se explicó anteriormente, el VIM es responsable de controlar y administrar la NFVI y Openstack es el responsable de hacer este papel [39].

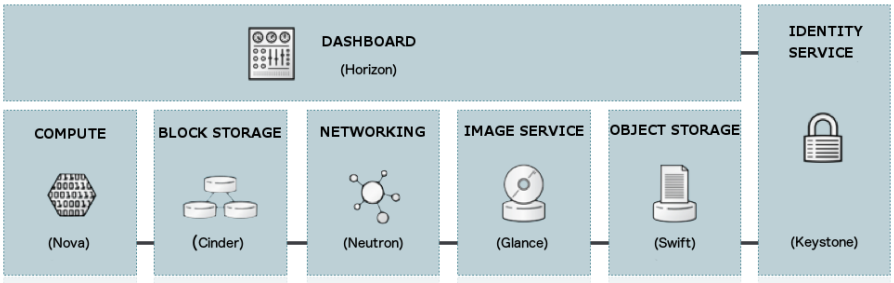
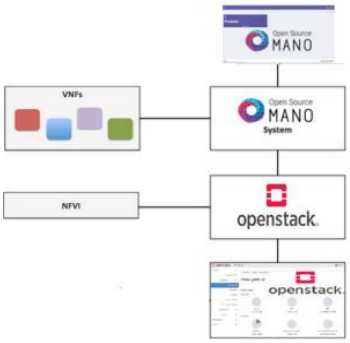


Figura45. Componentes OpenStack

- MANO-OSM: es una iniciativa alojada en ETSI, European Telecommunications Standards Institute, para desarrollar un stack de software open source NFV Management and Orchestration (MANO). OSM se centra en entregar un stack de gestión y orquestación capaz de satisfacer los requisitos de producción de redes NFV, implementado en software de código abierto [39].

La implementación de OSM en la infraestructura NFV se ha vuelto muy ágil y sencilla y, por otro lado, OpenStack es conocido por permitir la simplicidad para administrar infraestructuras virtualizadas y basadas en contenedores.



En un primer lugar se han creado dos VM Ubuntu 20.10, instalando en una Openstack:

```
ubuntu@ubunu2004:~$ microstack.openstack catalog list
+-----------+-----------+-----------------------------------------------------------------------+
| Name      | Type      | Endpoints                                                             |
+-----------+-----------+-----------------------------------------------------------------------+
| cinderv3  | volumev3  | microstack                                                            |
|           |           |   public: http://10.0.2.15:8776/v3/515c7847a89b41f690b2a5b6f2a7b04c   |
|           |           | microstack                                                            |
|           |           |   admin: http://10.0.2.15:8776/v3/515c7847a89b41f690b2a5b6f2a7b04c    |
|           |           | microstack                                                            |
|           |           |   internal: http://10.0.2.15:8776/v3/515c7847a89b41f690b2a5b6f2a7b04c |
|           |           |                                                                       |
| nova      | compute   | microstack                                                            |
|           |           |   internal: http://10.0.2.15:8774/v2.1                                |
|           |           | microstack                                                            |
|           |           |   public: http://10.0.2.15:8774/v2.1                                  |
|           |           | microstack                                                            |
|           |           |   admin: http://10.0.2.15:8774/v2.1                                   |
|           |           |                                                                       |
| cinderv2  | volumev2  | microstack                                                            |
|           |           |   public: http://10.0.2.15:8776/v2/515c7847a89b41f690b2a5b6f2a7b04c   |
|           |           | microstack                                                            |
|           |           |   internal: http://10.0.2.15:8776/v2/515c7847a89b41f690b2a5b6f2a7b04c |
|           |           | microstack                                                            |
|           |           |   admin: http://10.0.2.15:8776/v2/515c7847a89b41f690b2a5b6f2a7b04c    |
|           |           |                                                                       |
| placement | placement | microstack                                                            |
|           |           |   public: http://10.0.2.15:8778                                       |
|           |           | microstack                                                            |
|           |           |   admin: http://10.0.2.15:8778                                        |
|           |           | microstack                                                            |
|           |           |   internal: http://10.0.2.15:8778                                     |
|           |           |                                                                       |
| neutron   | network   | microstack                                                            |
|           |           |   internal: http://10.0.2.15:9696                                     |
|           |           | microstack                                                            |
|           |           |   public: http://10.0.2.15:9696                                       |
|           |           | microstack                                                            |
|           |           |   admin: http://10.0.2.15:9696                                        |
|           |           |                                                                       |
| keystone  | identity  | microstack                                                            |
|           |           |   public: http://10.0.2.15:5000/v3/                                   |
|           |           | microstack                                                            |
|           |           |   admin: http://10.0.2.15:5000/v3/                                    |
|           |           | microstack                                                            |
|           |           |   internal: http://10.0.2.15:5000/v3/                                 |
|           |           |                                                                       |
| glance    | image     | microstack                                                            |
|           |           |   admin: http://10.0.2.15:9292                                        |
|           |           | microstack                                                            |
```

Figura46. Instalación OpenStack

Que cuenta con una interfaz web desde donde poder gestionar el entorno:

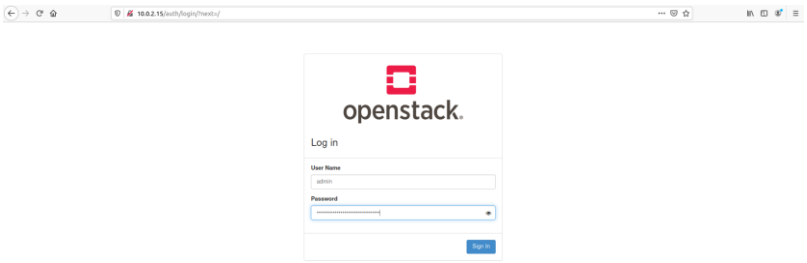


Figura47. GUI OpenStack

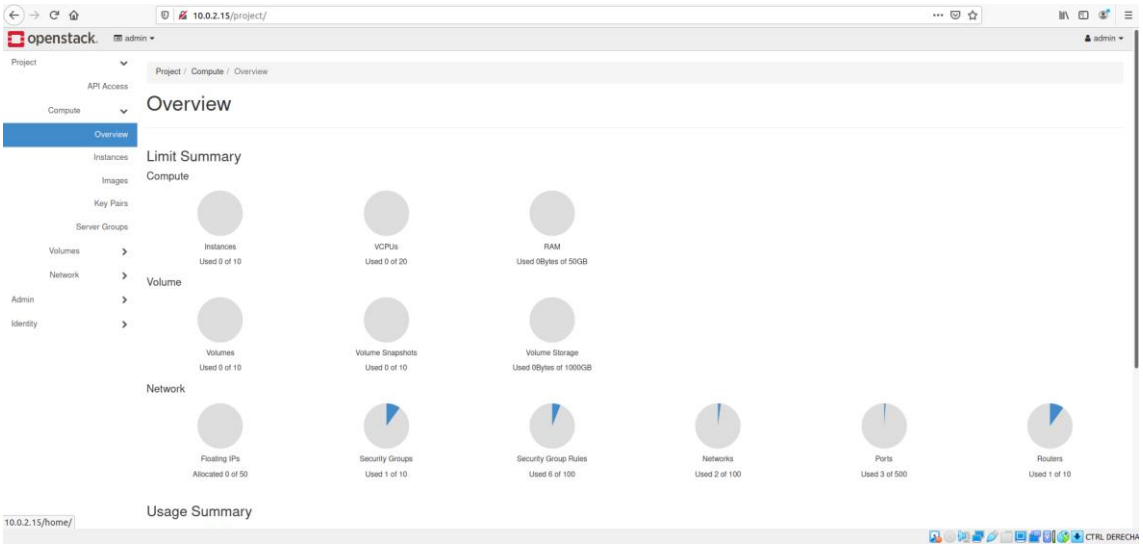


Figura48. OverView entorno desplegado de OpenStack

En la otra máquina se ha instalado OSM junto a Docker para si instanciación a través de contenedores:

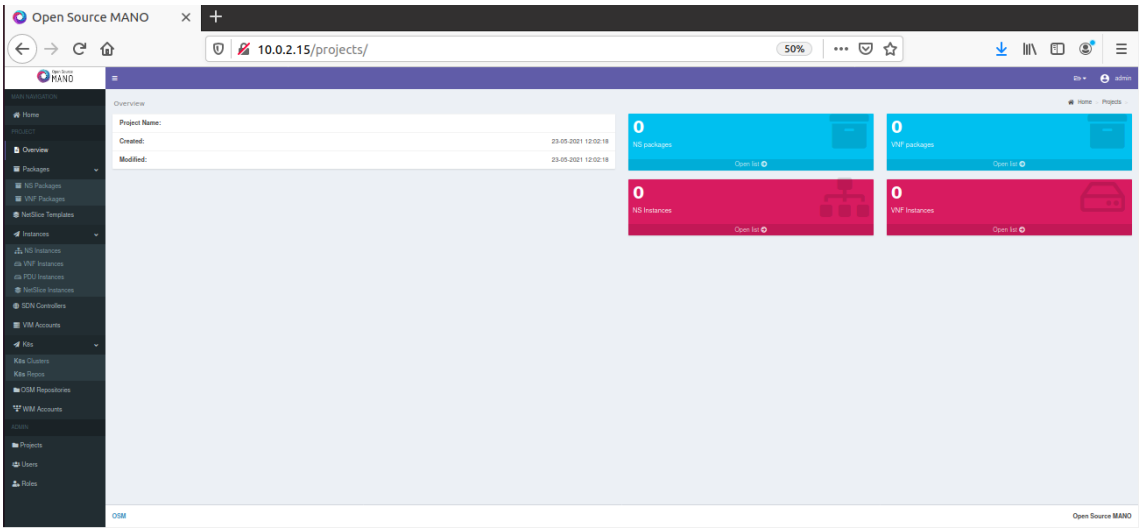


Figura49. GUI OSM MANO

Tras la instalación es necesario unir MANO-OSM a Openstack para que este lo controle y le dé las ordenes necesarias para crear las VMs. En este paso han surgido numerosos

problemas y no se ha podido conectar OSM a OpenStack pero resulta útil mostrar su funcionamiento.

Una vez relacionados ambos componentes el siguiente paso es subir lo paquetes VNF y SN para el despliegue de las funciones de red virtualizadas:

- VNF Package: es un archivo ZIP que incluye VNFD, imágenes de software para las máquinas virtuales y otros recursos como scripts, archivos de configuración y un archivo de metadatos estructurado (especifica el número de CPUs de la VM, memoria, almacenamiento). Contiene toda la información necesaria para crear una VNF para el NFVO.
- NS o NSD Package: es el descriptor de servicio de red y contiene valores como las NFV que contiene el NSD, la conexión entre las VNFs y descriptores.

Ejemplo de VNFD:

```
vnfd:
-   id: cirros_vnfd
    name: cirros_vnf
    short-name: cirros_vnf
    description: Simple VNF example with a cirros
    vendor: OSM
    version: '1.0'

    # Place the logo as png in icons directory and provide the name here
    logo: cirros-64.png

    # Management interface
    mgmt-interface:
        cp: eth0

    # Atleast one VDU need to be specified
    vdu:
    -   id: cirros_vnfd-VM
        name: cirros_vnfd-VM
        description: cirros_vnfd-VM
        count: 1

        # Flavour of the VM to be instantiated for the VDU
        # flavor below can fit into m1.micro
        vm-flavor:
            vcpu-count: 1
            memory-mb: 256
            storage-gb: 2

        # Image/checksum or image including the full path
        image: 'cirros-0.3.5-x86_64-disk.img'
        #checksum:

        interface:
        # Specify the external interfaces
        # There can be multiple interfaces defined
        -   name: eth0
            type: EXTERNAL
            virtual-interface:
                type: PARAVIRT
                bandwidth: '0'
                vpci: 0000:00:0a.0
            external-connection-point-ref: eth0
```

Figura50. Descriptor para despliegue VNF

Cargando ambos paquetes en el OSM, este manda las ordenes necesarias a Openstack que se encargará de realizar el despliegue de las VNF (VM que la componen, CPU que

necesita cada una, almacenamiento, memoria, conexiones entre VM para montar la VNF). De este modo, como se explicaba anteriormente se consigue mediante la arquitectura de NFVI crear un entorno donde las funciones de red se desvinculan completamente del HW y gracias a los componentes con los que cuenta el usuario puede crear/gestionar redes muy complejas desde una interfaz gráfica abstrayéndose completamente de lo que hay por debajo.

6.3 PRUEBA ESCENARIO LLAMADA SIP CON ASTERISK EN ENTORNO DOCKERIZADO Y CONECTIVIDAD ENTRE CONTENEDORES

6.3.1 Llamada SIP con Asterisk

Debido a las limitaciones que ofrecen algunas herramientas o programas gratuitos y OpenSource, el escenario que se plantea para el laboratorio es el siguiente:

Figura51. Representación escenario de prueba SIP

A continuación, se describen los programas utilizados para posteriormente explicar el funcionamiento:

- **VirtualBox**: es un es un software de virtualización desarrollado por Oracle Corporation para arquitecturas x86/amd64. Ha sido necesario el uso de software de virtualización para crear una VM compatible con Docker ya que actualmente solo se dispone de un portátil Windows 8 y Docker es compatible con Windows 10.
- **Docker**: se ha instalado la plataforma de Docker para poder demostrar las ventajas que se han comentado en la memoria y una herramienta muy interesante que ofrece llamada docker-compose.
- **Asterisk**: Asterisk es una solución de VoIP (Voz sobre IP), telefonía VoIP de código abierto (bajo licencia GPL) que proporciona funciones de una centralita telefónica PBX o call center. La centralita deja de ser una caja negra inaccesible y permite gestionar las funciones telefónicas de manera sencilla [40].

Una central telefónica IP o PBX IP consiste básicamente en uno o más teléfonos SIP (clientes), y un servidor. Con ellos, se puede realizar una llamada interna mediante un directorio de teléfonos/usuarios presentes en la red interna, o enrutar una llamada externa a través de VoIP.

- **Zoiper**: es un software multiplataforma (funciona con Windows, Linux, MAC, iPod Touch, iPad, iPhone, tablets y Android), diseñado para trabajar con los sistemas de comunicación IP basado en el protocolo SIP. La solución softphone permite ahorrar la necesidad de tener equipos dedicados para la telefonía, y un ordenador, tablet o teléfono móvil serán suficientes para poder usar la telefonía IP.

Para poder crear el entono ha sido necesario descargar una imagen de Linux y a través de VirtualBox se ha creado una VM Linux Ubuntu 5.8. En la VM se ha descargado la plataforma de Docker a través de la consola de comandos:

```
ubuntu@ubuntu2010:~/prueba1/docker-asterisk/demo$ sudo docker version
Client: Docker Engine - Community
 Version:           20.10.6
 API version:       1.41
 Go version:        go1.13.15
 Git commit:        370c289
 Built:             Fri Apr  9 22:46:38 2021
 OS/Arch:           linux/amd64
 Context:           default
 Experimental:      true
```

Figura52. Versión Docker instalada

Como ya se comentó existe un repositorio, configurado por defecto, donde se pueden descargar distintas imágenes para desplegar en los contenedores de Docker. En este caso se ha descargado una imagen de Asterisk que permite simular un entorno de red SIP, a través de una centralita PBX, y realizar llamadas de prueba.

```
ubuntu@ubuntu2010:~/prueba1/docker-asterisk/demo$ sudo docker images
REPOSITORY      TAG       IMAGE ID       CREATED         SIZE
ubuntu          latest    7e0aa2d69a15   3 weeks ago     72.7MB
hello-world     latest    d1165f221234   2 months ago    13.3kB
mlan/asterisk   latest    29cd9b44b790   2 months ago    73.2MB
```

Figura53. Imágenes cargadas en Docker

Aquí se puede apreciar uno de los puntos fuertes de los contenedores que es el **peso de las imágenes**, lo que hace que sean muy ágiles en su despliegue y manejo. Vamos a crear una aplicación totalmente funcional e independiente que ocupa únicamente 73.2 MB lo que hace que su instanciación y manejo sea muy liviano y ágil. Para crear un contenedor Asterisk y utilizar las facilidades que ofrece Docker se ha utilizado Docker-compose que a través de un fichero docker-compose.yml, que contiene las ordenes necesarias, se crea el contenedor.

```
ubuntu@ubuntu2010:~/prueba1/docker-asterisk/demo$ sudo docker-compose up
Starting demo_tele_1 ... done
ubuntu@ubuntu2010:~/prueba1/docker-asterisk/demo$ sudo docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED      STATUS                          PORTS

                                   NAMES
0e63e30e4f5c   mlan/asterisk   "docker-entrypoint.s…"   4 days ago   Up 8 seconds (health: starting)   0.0.0.0:5060->5060/udp, :::5060->5060/u
dp, 0.0.0.0:5060-5061->5060-5061/tcp, 0.0.0.0:10000-10099->10000-10099/udp, :::5060-5061->5060-5061/tcp, :::10000-10099->10000-10099/udp, 0.0.
0.0:8080->80/tcp, :::8080->80/tcp   demo_tele_1
```

Figura54. Creación de contenedor desde Docker-compose

Es necesario realizar el mapeo de puertos entre el contenedor y el host, que serán a los que apunten los softphones (Zoiper), para el registro y realización de llamadas (SIP y RTP Ports).

Una vez levantado es posible conectarse al contenedor mediante “Docker exec” para así poder gestionar el contenedor de Asterisk. Para la realización de las pruebas están definidos dos endpoints o usuarios que van a poder conectarse (Jane y John) que aparecen en el estado no disponible:

```
0e63e30e4f5c*CLI> pjsip show endpoints

 Endpoint:  <Endpoint/CID.....................................>  <State.....>  <Channels.>
    I/OAuth:  <AuthId/UserName...................................................................>
        Aor:  <Aor...........................................>  <MaxContact>
      Contact:  <Aor/ContactUri..........................> <Hash....> <Status> <RTT(ms)..>
  Transport:  <TransportId........>  <Type>  <cos>  <tos>  <BindAddress...................>
   Identify:  <Identify/Endpoint...............................................................>
        Match:  <criteria........................>
    Channel:  <ChannelId.....................................>  <State.....>  <Time.....>
        Exten: <DialedExten...........>  CLCID: <ConnectedLineCID.......>
==========================================================================================

 Endpoint:  jane.doe/+12025550183                                Unavailable   0 of inf
     InAuth:  jane.doe-iauth/jane.doe
        Aor:  jane.doe                                          10

 Endpoint:  john.doe/+12025550160                                Unavailable   0 of inf
     InAuth:  john.doe-iauth/john.doe
        Aor:  john.doe                                          10
```

Figura55. Estado de Endpoints definidos en Asterisk

Para que los usuarios se registren en la PBx y puedan realizar llamadas es necesario configurar el cliente Zoiper especificando nombre de usuario, dominio con la IP de la VM y contraseña. Una vez configurados, los usuarios mandan una petición SIP REGISTER y aparecen en la PBx virtual como disponible, “Available”:

```
0e63e30e4f5c*CLI> pjsip show endpoints

 Endpoint:  <Endpoint/CID.....................................>  <State.....>  <Channels.>
    I/OAuth:  <AuthId/UserName...................................................................>
        Aor:  <Aor...........................................>  <MaxContact>
      Contact:  <Aor/ContactUri..........................> <Hash....> <Status> <RTT(ms)..>
  Transport:  <TransportId........>  <Type>  <cos>  <tos>  <BindAddress...................>
   Identify:  <Identify/Endpoint...............................................................>
        Match:  <criteria........................>
    Channel:  <ChannelId.....................................>  <State.....>  <Time.....>
        Exten: <DialedExten...........>  CLCID: <ConnectedLineCID.......>
==========================================================================================

 Endpoint:  jane.doe/+12025550183                                Not in use    0 of inf
     InAuth:  jane.doe-iauth/jane.doe
        Aor:  jane.doe                                          10
      Contact:  jane.doe/sip:jane.doe@192.168.1.140:47689; 2a72ae25a5 Avail        11.788

 Endpoint:  john.doe/+12025550160                                Not in use    0 of inf
     InAuth:  john.doe-iauth/john.doe
        Aor:  john.doe                                          10
      Contact:  john.doe/sip:john.doe@192.168.1.126:48381; fc0d834582 Avail         3.298
```

Figura56. Estado de Endpoints definidos en Asterisk (2)

En este punto los terminales ya están registrados y pueden realizar llamadas que se capturarán en wireshark. En la VM está registrado John y en el Smarthphone Android está registrada Jane. El escenario de prueba consiste en realizar una llamada VoIP desde John a Jane, descolar y comprobar que hay audio (RTP):

John:

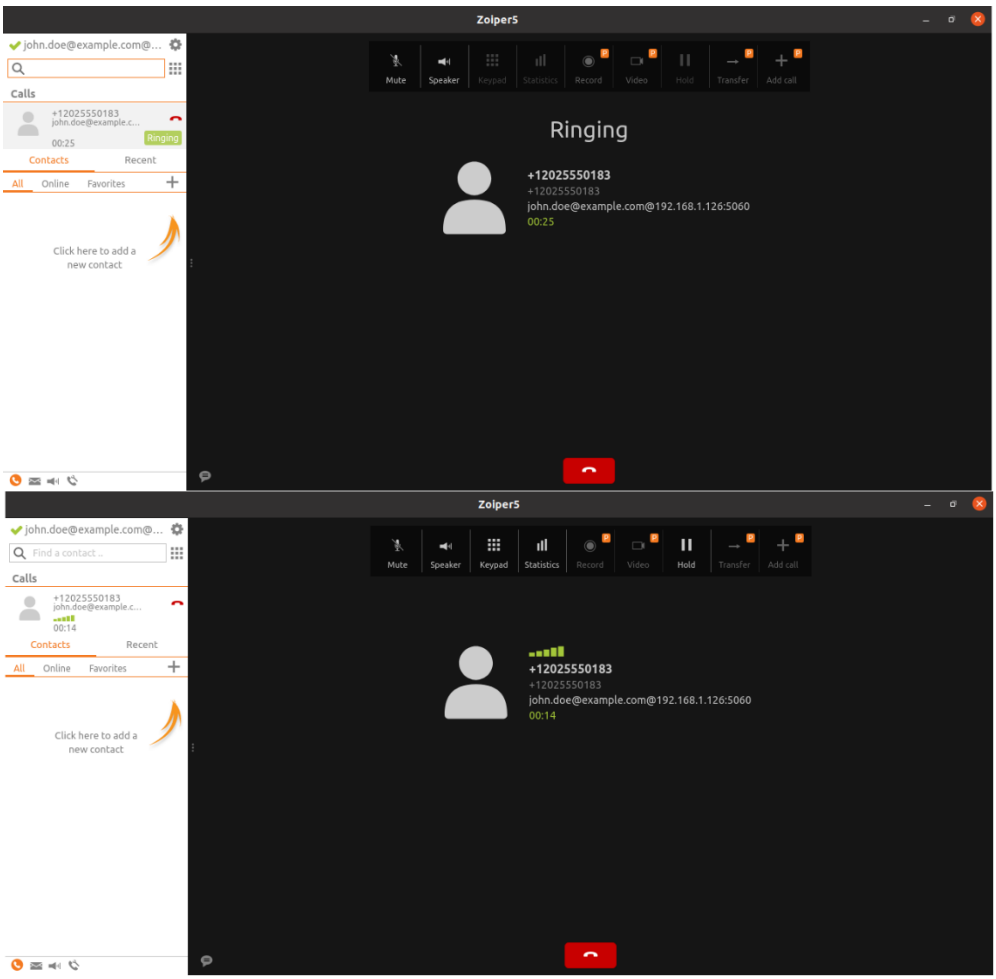


Figura57.Llamada de prueba John

Jane:

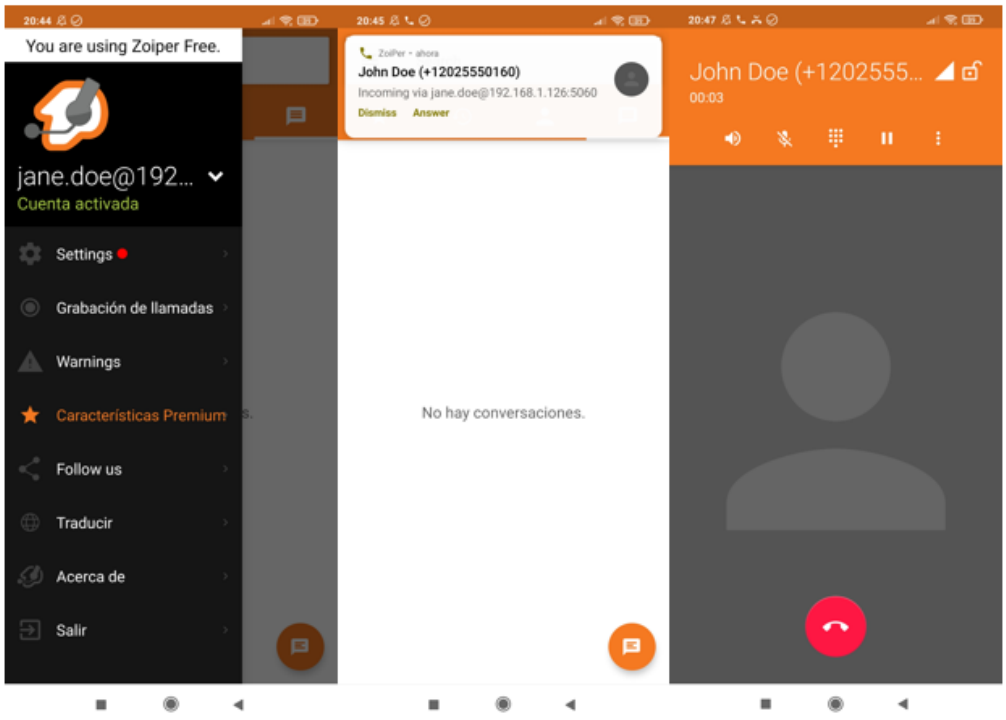


Figura58.Llamada de prueba Jane

De este modo, contando únicamente con un portátil como HW y a través de una máquina virtual ha sido posible crear una centralita privada capaz de procesar y dirigir llamadas, en este caso entre dos endpoints o terminales pero que podría ser extensible al ámbito de una empresa con la ventaja, además, de poder configurar el servicio ad-hoc adaptado a las condiciones particulares de cada usuario. Así **se ha podido probar**, a pequeña escala, **la posibilidad de implantar un servicio de VoIP mediante el uso o empleo de contenedores**.

Para asegurar o verificar que el tratamiento de la llamada es el adecuado se ha querido comprobar el intercambio de mensajes de señalización que ha tenido lugar entre ambos endpoints y el envío de audio a través de RTP.

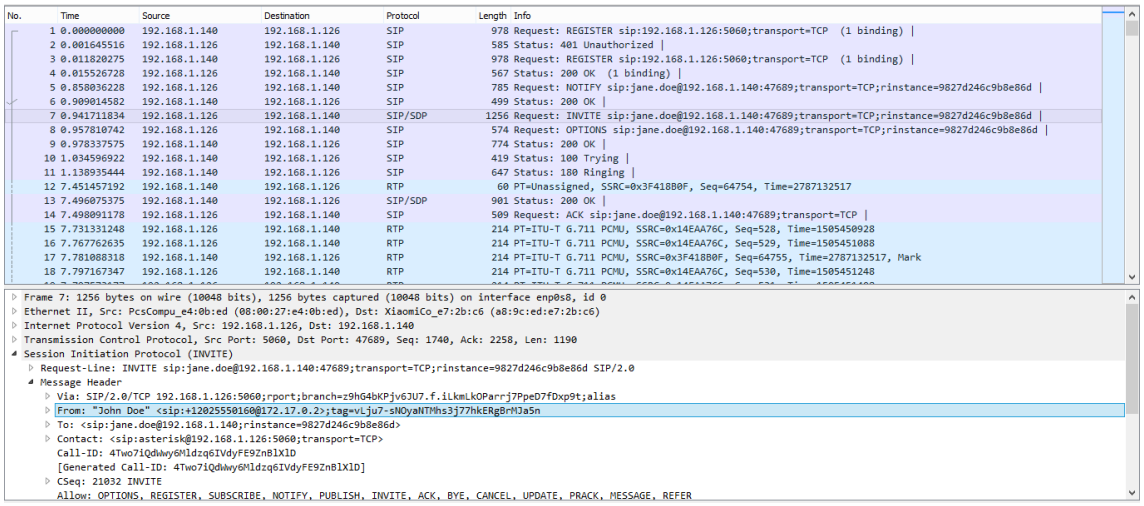


Figura59.Traza establecimiento de llamada de prueba

En la traza, mensaje INVITE (cabeceras ‘FROM’ - ‘TO’) se puede observar el comienzo de la llamada de John a Jane y como, en el registro nº 15 mensaje OK, Jane descuelga la llamada y comienza el flujo RTP.

Nota: Obviar mensajes ‘OPTIONS’, es un mecanismo de KeepAlive de SIP (similar a un ping), y ‘NOTIFY’, únicamente comunica la finalización de un evento.

Finalmente, Jane corta la comunicación lo que se traduce en un mensaje BYE hacía John:

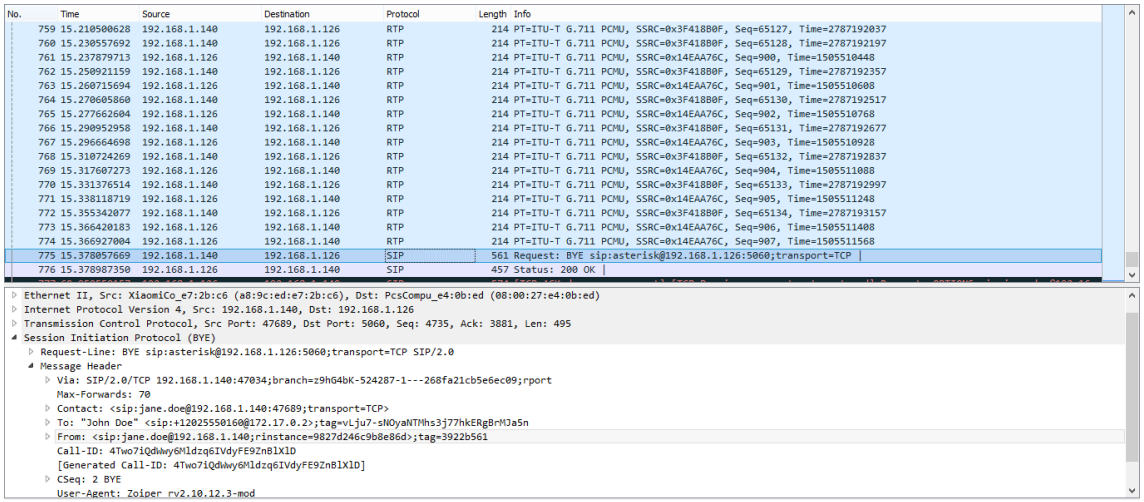


Figura60.Traza finalización de llamada de prueba

De manera gráfica, se muestra el CallFlow de la llamada:

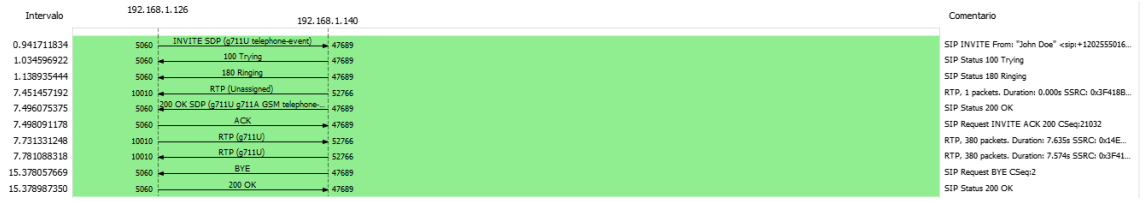


Figura60.CallFlow llamada de prueba

También ha sido posible comprobar algunas de las propiedades de los contenedores:

- **Entorno uniforme:** sin necesidades de instalar ningún paquete adicional el contendor arranca y se comporta perfectamente independientemente de las versiones que tiene el Linux anfitrión.
- **Agilidad:** el contenedor con la imagen de Asterisk queda instanciada y corriendo en 1,5-2 segundos gracias a que apenas ocupa 73.2 MB, mientras que una VM que necesita arrancar por completo con sistema operativo puede tardar en torno a 2-5 minutos ocupando 2GB.

6.3.2 Conectividad entre contenedores

Para comprender la configuración de red en los contenedores es necesario entender SDN de una manera más abstracta o como una parte integrada dentro de Docker, no hay que pensar en la mecánica de una capa SDN separada. No es necesario preocuparse por el enrutamiento, los puertos, las listas de control de acceso (ACL) o las direcciones IP; la infraestructura se preocupa de todo eso y, simplemente, funciona. Para ello se crea un escenario de prueba donde se pretende comprender la conectividad de

contenedores y comprobar que la plataforma nos permite llevar a cabo la configuración de red mediante software sin necesidad de conocer las capas subyacentes.

En este segundo caso se ha querido aplicar la configuración de una red virtual en el entorno de los contenedores con la ayuda de docker-network. Para ello se han creado 3 contenedores con una imagen OpenSwitch llamadas ops1, ops2 y ops3.

```
ubuntu@ubuntu2010:~$ sudo docker ps
CONTAINER ID      IMAGE         COMMAND         CREATED              STATUS              PORTS     NAMES
9f93e30c75ed      ops:latest    "/sbin/init"    7 seconds ago        Up 4 seconds                  ops3
622b65b5a9f0      ops:latest    "/sbin/init"    36 seconds ago       Up 34 seconds                 ops2
6f596a6b9ed5      ops:latest    "/sbin/init"    About a minute ago   Up About a minute             ops1
ubuntu@ubuntu2010:~$
ubuntu@ubuntu2010:~$
ubuntu@ubuntu2010:~$
ubuntu@ubuntu2010:~$ sudo docker network inspect bridge | egrep 'Name|IPv4Address'
        "Name": "bridge",
                "Name": "ops2",
                "IPv4Address": "172.17.0.3/16",
                "Name": "ops1",
                "IPv4Address": "172.17.0.2/16",
                "Name": "ops3",
                "IPv4Address": "172.17.0.4/16",
```

Figura61.Conectividad entre contenedores

Como se puede observar, por defecto, Docker crea la siguiente arquitectura o esquema de conexión:

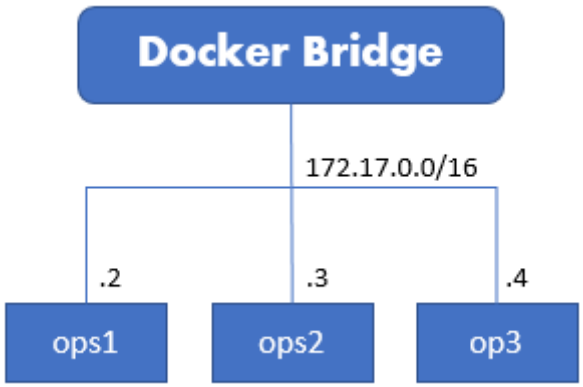


Figura62.Esquema de conexión

Ahora, se quiere crear una red donde únicamente estén ops1-op2 y otra donde esté ops2-ops3 por lo que se crean dos subredes locales 192.168.101.0/24 (ops1-ops2) y 192.168.102.0/24 (ops2-ops3) y se conectan los contenedores a la subred que corresponda a través de un bridge que hace las funciones de switch en este caso quedando el escenario del siguiente modo:

```
ubuntu@ubuntu2010:~$ sudo docker network create --subnet=192.168.101.0/24 --gateway=192.168.101.254 net101
cf3c9fc05777c07fabb828c8eba1c9735ab45855d7b5d045af204387d36d2696
ubuntu@ubuntu2010:~$ sudo docker network create --subnet=192.168.102.0/24 --gateway=192.168.102.254 net102
703e208f807a12bb5e9ca9758c70be0a5a382a6e012ed041a4326e43a0e840d9
ubuntu@ubuntu2010:~$ sudo docker network connect net101 ops1
ubuntu@ubuntu2010:~$ sudo docker network connect net101 ops2
ubuntu@ubuntu2010:~$ sudo docker network connect net102 ops2
ubuntu@ubuntu2010:~$ sudo docker network connect net102 ops3
```

Figura63.Crear subredes

Config. escenario de prueba

```
ubuntu@ubuntu2010:~$ sudo docker network inspect net101 | egrep 'Name|Gateway|IPv4Add'
        "Name": "net101",
                    "Gateway": "192.168.101.254"
                "Name": "ops2",
                "IPv4Address": "192.168.101.2/24",
                "Name": "ops1",
                "IPv4Address": "192.168.101.1/24",
ubuntu@ubuntu2010:~$ sudo docker network inspect net102 | egrep 'Name|Gateway|IPv4Add'
        "Name": "net102",
                    "Gateway": "192.168.102.254"
                "Name": "ops2",
                "IPv4Address": "192.168.102.1/24",
                "Name": "ops3",
                "IPv4Address": "192.168.102.2/24",
```

Figura64.Muestra de subredes

Configuración de las interfaces del Host (VM)

```
11: br-cf3c9fc05777: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:53:54:dd:de brd ff:ff:ff:ff:ff:ff
    inet 192.168.101.254/24 brd 192.168.101.255 scope global br-cf3c9fc05777
       valid_lft forever preferred_lft forever
    inet6 fe80::42:53ff:fe54:ddde/64 scope link
       valid_lft forever preferred_lft forever
12: br-703e208f807a: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c6:8b:c0:11 brd ff:ff:ff:ff:ff:ff
    inet 192.168.102.254/24 brd 192.168.102.255 scope global br-703e208f807a
       valid_lft forever preferred_lft forever
    inet6 fe80::42:c6ff:fe8b:c011/64 scope link
       valid_lft forever preferred_lft forever
14: vethac98c8e@if13: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-cf3c9fc05777 state UP group default
    link/ether 26:39:36:ae:0d:2f brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet6 fe80::2439:36ff:feae:d2f/64 scope link
       valid_lft forever preferred_lft forever
16: veth145cf83@if15: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-cf3c9fc05777 state UP group default
    link/ether 7a:34:b8:ef:1c:0e brd ff:ff:ff:ff:ff:ff link-netnsid 1
    inet6 fe80::7834:b8ff:feef:1c0e/64 scope link
       valid_lft forever preferred_lft forever
18: veth37096ed@if17: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-703e208f807a state UP group default
    link/ether 16:4f:e5:43:42:b6 brd ff:ff:ff:ff:ff:ff link-netnsid 1
    inet6 fe80::144f:e5ff:fe43:42b6/64 scope link
       valid_lft forever preferred_lft forever
20: veth85f5a6c@if19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-703e208f807a state UP group default
    link/ether 76:29:72:0b:db:11 brd ff:ff:ff:ff:ff:ff link-netnsid 2
    inet6 fe80::7429:72ff:fe0b:db11/64 scope link
       valid_lft forever preferred_lft forever
```

Figura65.Configuración de las interfaces del host

Configuración de los interfaces de contenedor Ops2 (es el mejor ejemplo ya que está en dos redes):

```
root@switch:~# ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
7: eth0@if8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:11:00:03 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.3/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
15: eth1@if16: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:a8:65:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.101.2/24 brd 192.168.101.255 scope global eth1
       valid_lft forever preferred_lft forever
17: eth2@if18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:c0:a8:66:01 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 192.168.102.1/24 brd 192.168.102.255 scope global eth2
       valid_lft forever preferred_lft forever
```

Figura66.Configuración de interfaces del contenedor ops2

De manera esquemática y más visual sería:

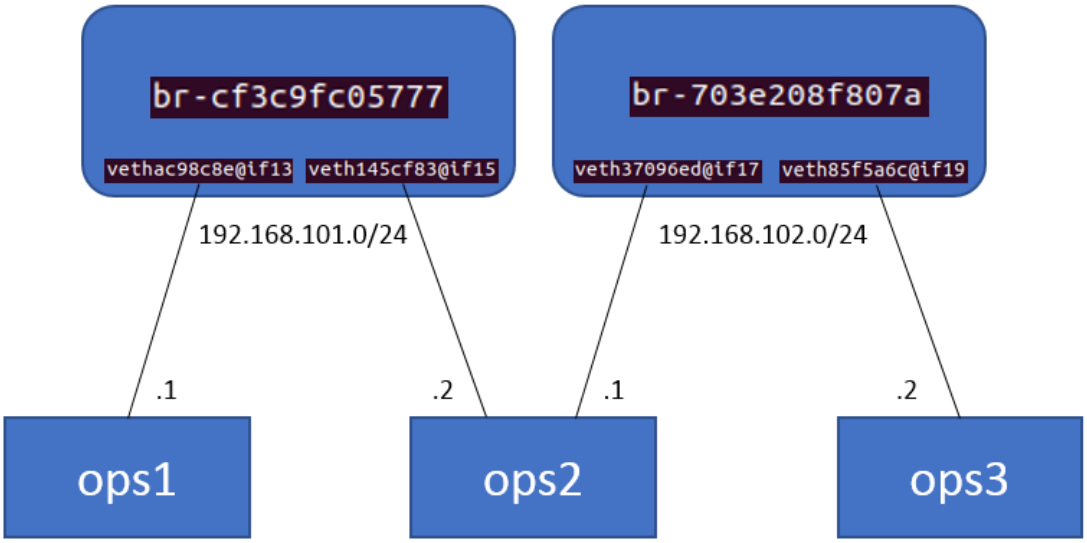


Figura67. Esquema de conectividad con dos subredes

Se han creado dos bridges virtuales independientes con interfaces eth. que van conectados a cada uno de los contenedores. Para comprobar que el funcionamiento es el esperado se han realizado ping entre contenedores para probar la conectividad entre ellos:

Ops1 (solo se alcanza el contenedor ops2)

```
root@switch:~# ping 192.168.102.2
PING 192.168.102.2 (192.168.102.2) 56(84) bytes of data.
^C
--- 192.168.102.2 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6133ms
```

Figura68. Ping ops1 – ops3

Ops2 (se alcanzan ambos contenedores al tener configuradas las dos subredes)

```
root@switch:~# ping 192.168.102.2
PING 192.168.102.2 (192.168.102.2) 56(84) bytes of data.
64 bytes from 192.168.102.2: icmp_seq=1 ttl=64 time=0.572 ms
64 bytes from 192.168.102.2: icmp_seq=2 ttl=64 time=0.169 ms
^C
--- 192.168.102.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1010ms
rtt min/avg/max/mdev = 0.169/0.370/0.572/0.202 ms
root@switch:~# ping 192.168.101.1
PING 192.168.101.1 (192.168.101.1) 56(84) bytes of data.
64 bytes from 192.168.101.1: icmp_seq=1 ttl=64 time=0.383 ms
64 bytes from 192.168.101.1: icmp_seq=2 ttl=64 time=0.140 ms
```

Figura69. Ping ops2 a ops1 y ops3.

Ops3 (solo se alcanza el contenedor ops2)

```
root@switch:~# ping 192.168.101.1
PING 192.168.101.1 (192.168.101.1) 56(84) bytes of data.
^C
--- 192.168.101.1 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1022ms

root@switch:~# ping 192.168.102.1
PING 192.168.102.1 (192.168.102.1) 56(84) bytes of data.
64 bytes from 192.168.102.1: icmp_seq=1 ttl=64 time=0.188 ms
64 bytes from 192.168.102.1: icmp_seq=2 ttl=64 time=0.166 ms
```

Figura70. Ping ops3 a ops1 y ops2.

De este modo **probamos la configuración de contenedores de una manera sencilla** y sin bajar en ningún caso a nivel de bridge por lo que es posible crear redes mediante software y sin la necesidad de saber qué hay por debajo.

7. Conclusiones

En primer lugar, se ha querido realizar un análisis y/o síntesis de la evolución de las redes de telecomunicaciones poniendo de manifiesto el esfuerzo de las compañías por reducir costes para ofrecer multitud de servicios a un precio reducido a pesar de la complejidad que tienen y la infraestructura que hay montada detrás de ellos.

Para alcanzar el punto en el que hoy nos encontramos ha sido necesario sufrir un proceso de transformación constante que pone de manifiesto la tendencia actual, y futura, de la automatización de tareas y la apuesta por un entorno completamente virtual.

Tras evaluar el funcionamiento y aplicabilidad de la virtualización en las redes de telecomunicación se ha llegado a la conclusión o se ha comprobado la eficacia y agilidad que ofrece esta tecnología.

Las telecomunicaciones es un sector muy dinámico y en constate cambio que requiere, cada vez de manera más marcada, de soluciones ágiles y eficientes. Con este trabajo se comprueba que la virtualización se ajusta a las necesidades actuales permitiendo a las compañías ahorrar costes, ya no solo en el hardware, sino también en tareas de ingeniería, despliegues, desarrollos, pruebas, mantenimiento, etc. Es por ello que algunas operadoras de telecomunicaciones como Telefónica ya ha empezado a desplegar tecnologías NFV para poder proporcionar servicios 5G.

Con este proyecto se ha podido experimentar con estas tecnologías que trabajan conceptos como NFV, SDN y contenedores; mostrando como en un ordenador portátil sin demasiadas prestaciones se han podido simular escenarios relativamente complejos.

8. Glosario

CS - Circuit Switching

PS - Packet Switching

VNF - Virtual Network Function

NFV – Network Function Virtualization

NFVI - Network Function Virtualization Infrastructure

SND - Software Network Defined

RTB - Red Telefónica Básica

RTC - Red Telefónica Conmutada

PSTN - Public Switched Telephone Network

NGN - Next Generation Network

RDSI – Red digital de Servicios Integrados

xDSL - x Digital Subscriber Line

IMS – IP Multimedia Subsystem

SIP – Session Initiation Protocol

POTS - Plain Old Telephone Service

ISUP - ISDN User Part

MPLS – MultiProtocol Laber Switching

ISP – Internet Service Provider

PLMN - Public Land Mobile Network

TDMA – Time Division Multiple Access

CDMA – Code Division Multiple Access

GSM – Global System Mobile

EMS – Enhanced Message Service

MMS – Multimedia Message Service

GPRS – General Packet Radio Service

EDGE – Enhanced Data Rates for GSM Evolution

QoS – Quality of Service

UMTS – Universal Mobile Telecomunications System

NSC - Network Service Chaining

9. Bibliografía

Gráficos y Tablas

[1] GIF funcionamiento PS - https://es.m.wikipedia.org/wiki/Archivo:Packet_Switching.gif

[2] Call Flow SS7 Protocol - https://ichi.pro/es/deteccion-de-llamadas-de-solicitud-de-anillo-unico-wangiri-en-trafico-isup-71604017008234

[3] RTB-RTC Scema - Comunicación en redes, CIDECAM. Disponible en: http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro27/index.html

[4] Digitalización de la red RDSI - http://manque.cl.tripod.com/webarq/c_6.4_ISDN_RDSI.htm

[5] Acceso a la red de paquetes en RDSI - http://informatica.uv.es/iiguia/AER/Tema8.pdf

[6] Servicios en RDSI - https://www.web20.co/tecnologia/rdsi-red-digital-de-servicios-integrados/

[7] Configuración FTTx - https://es.wikipedia.org/wiki/FTTx

[8][9][10][11] – Esquema general red FTTH https://www.euskadi.eus/contenidos/informacion/peba_dit/es_def/adjuntos/Plan%20PEBA%20Euskadi.%202018-2019%20Despliegue%20FTTH_cas.pdf

[12][13][14][15][16] - Documentación autoformación

[17] Network Slicing - https://www.viavisolutions.com/es-es/node/71717

[18] Arquitectura 5G - http://axvr-tech.com/explicando-la-red-5g-que-es-realmente/

[19] Arquitectura IMS - https://es.wikipedia.org/wiki/Subsistema_Multimedia_IP

[20] ISUP to SIP - https://www.dialogic.com/webhelp/IMG1010/10.5.3/WebHelp/sipss7interworking.htm

[21] SIP Call Flow - https://www.onsip.com/voip-resources/voip-solutions/sip-call-flow-explained

[22] SDN Planes - https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/VMDC/SDN/SDN.html

[23][24] NFV & SDN - https://t3chfest.es/2018/programa/nfv-sdn-las-redes-del-futuro-del-presente/?lang=es

[25] Capas en VM y contenedores - https://blog.netapp.com/blogs/containers-vs-vms/

[26][27][28] - Documentación página oficial Kubernetes. Disponible en: https://kubernetes.io/docs/home/

[29] Documentación página oficial Docker. Disponible en: https://docs.docker.com/

Documentación

[29] Documento: “Unidad 2: Red Telefónica Pública Conmutada (PSTN).” Disponible en: https://mestreacasa.gva.es/c/document_library/get_file?folderId=500021199667&name=DLFE-1781907.pdf

[30] Developer Information – Introduction ISUP. Disponible en: https://www.dialogic.com/webhelp/MSP1010/10.4.0/WebHelp/msp_dg/isup/

[31] PSTN, DSLAM y ISDN ó RDSI, 19 de enero de 2011. Disponible en: https://startcom1.wordpress.com/2011/01/19/pstn-dslam-y-isdn-o-rdsi/

[32] RDSI (Red Digital de Servicios Integrados), por Ramón Jesús Millán Tejedor, 2008. Disponible en: https://www.ramonmillan.com/tutoriales/rdsi.php

[33] FTTH y FTTx, ¿Qué es?, Conectrónica. Disponible en: https://www.conectronica.com/fibra-optica/ftth-fftx-fibra-optica/ftth-redes-fttx-fibra-optica

[34] El teléfono celular. Historia y evolución de los celulares, Tecnología + Informática. Disponible en: https://www.tecnologia-informatica.com/telefono-celular-historia-evolucion-celulares/

RFC 1306, por A. Nicholson y J. Young (Cray Research, Inc.), Marzo 1992. Disponible en: https://datatracker.ietf.org/doc/html/rfc1306

RFC 3439, por R.Bush y D.Meyer, Diciembre 2001. Disponible en: https://www.ietf.org/rfc/rfc3439.txt

[35] Documentación página oficial Docker. Disponible en: https://docs.docker.com/

[36] Documentación página oficial Kubernetes. Disponible en: https://kubernetes.io/docs/home/

[37] Mininet Walkthrough. Disponible en: http://mininet.org/walkthrough/

[38] OpenStack -Documentation for Wallaby (April 2021). Disponible en: https://docs.openstack.org/wallaby/

[39] Open Source MANO's documentation. Disponible en: https://osm.etsi.org/docs/user-guide/

[40] Repositorio GitHub - mlan/Docker-asterisk. Disponible en: https://github.com/mlan/docker-asterisk