

Universidad Politécnica de Madrid  
Escuela Técnica Superior de Ingenieros de Telecomunicación



**VÍDEO STREAMING DE ALTA CALIDAD  
SOBRE PLATAFORMAS DE  
DISTRIBUCIÓN DE CONTENIDOS**

**TRABAJO FIN DE MÁSTER**

**Felipe David Valdez Montero**

2016



Universidad Politécnica de Madrid  
Escuela Técnica Superior de Ingenieros de Telecomunicación

**Máster Universitario en  
Ingeniería de Redes y Servicios Telemáticos**

**TRABAJO FIN DE MÁSTER**

**VÍDEO STREAMING DE ALTA CALIDAD  
SOBRE PLATAFORMAS DE  
DISTRIBUCIÓN DE CONTENIDOS**

Autor

**Felipe David Valdez Montero**

Director

**Dña. Encarnación Pastor Martín**

Departamento de Ingeniería de Sistemas Telemáticos

2016



## Resumen

En los últimos años hemos sido testigos de la evolución exponencial que ha experimentado Internet. La aparición de nuevas tecnologías y técnicas de optimización de las redes han permitido el nacimiento de nuevos servicios y aplicaciones Web que antes eran impensables, sobretodo en las redes IP tradicionales. Servicios como el almacenamiento masivo de datos en la nube, servicios multimedia de radio y video streaming. Estos servicios han popularizado gracias a la demanda por parte los usuarios, y también, gracias a que contamos con dispositivos que cada vez son más avanzados y que cuentan con mejores funcionalidades multimedia. Dispositivos como los smartphones, Smart TV's, consolas de videojuegos, ordenadores, etc. Esto también ha supuesto una gran oportunidad de negocio para las tecnológicas como los proveedores de contenidos, empresas de desarrollo hardware, hasta las empresas tecnológicas de electrónica de consumo.

Por otro lado, la aparición de estos nuevos servicios han generado una gran demanda de recursos de Internet, debido principalmente a la gran cantidad de tráfico que generan, el mismo que puede llegar a congestionar y crea cuellos de botella en las redes IP. Se estima que en los próximos años, más del 50% del tráfico de Internet será tráfico de streaming, y gran parte de él será de video. Es por ello, que actualmente se utilizan e investigan nuevas técnicas que permitan mejorar el rendimiento de las redes. Técnicas como el uso de la Redes de Distribución de Contenido - Content Delivery Networks (CDNs), nuevos protocolos basados en HiperText Transfer Protocol (HTTP), en especial el Dynamic Adaptive Streaming over HTTP (DASH), nuevos codecs de video como H.264, H.265 o High Efficiency Video Coding (HEVC), etc.

El objetivo principal de todas estas técnicas y mecanismos, es poder ofrecer al usuario final una mejor calidad de servicio - Quality of Service (QoS) y calidad de experiencia - Quality of Experience (QoE) mediante el aprovechamiento de los recursos existentes en la red, sobretodo el ancho de banda del lado del cliente, ya que la última milla sigue generando problemas de cuello de botella.

Este trabajo pretende analizar las técnicas que se están implementando e investigando para lograr que el video de alta calidad pueda llegar al usuario final. Veremos los componentes principales que participan en la cadena de la creación, distribución y entrega del contenido de video al usuario, que protocolos son utilizados, codecs, como distribuyen los contenidos multimedia, y las líneas futuras que permitan mejorar la entrega del video.



## Abstract

In recent years we have all been witnessed about exponential evolution of Internet. The emergence of new technologies and network optimization techniques have allowed the birth of new services and Web applications that were unthinkable, especially in traditional IP networks. Services such as mass storage in the Cloud, radio streaming and video streaming. These services have become popular due to demand from users, and because of we have devices that are increasingly advanced and have better multi-media features. Devices such as smart phones, Smart TV's, game consoles, computers, etc. This has also supposed a great business opportunity for technology and content providers, hardware development and consumer electronics companies.

On the other hand, these new services have generated a strong demand of Internet resources, mainly due to the large amount of traffic they generate, it can get congested and creates bottlenecks over IP networks. It is estimated that in the coming years, more than 50% of Internet traffic will be streaming traffic, and specially video traffic. That is why, that nowadays new techniques are been used and investigated to improve network performance. Techniques such as the use of Content Delivery Networks (CDNs), new Hypertext Transfer Protocol (HTTP) - based protocols, especially Dynamic Adaptive Streaming over HTTP (DASH), new video codecs such as H.264, H.265 or High Efficiency Video Coding (HEVC), etc.

The main objective of all these techniques and mechanisms, is to offer a better Quality of Service (QoS) and Quality of Experience (QoE) to the end users, by taking advantage of network resources, especially bandwidth in the client side. Last mile continues to generate bottleneck problems.

This Master's Thesis analyzes the techniques which are being implemented and researched to make the high-quality video to reach the end user. We will see the main components involved in the chain of creation, distribution and delivery of video content to the user, which protocols are used, codecs, how multimedia contents are distributed, and future lines to improve the video delivery.





## Índice general

Resumen.....	i
Abstract .....	ii
Índice general .....	iv
Índice de figuras .....	vi
Índice de Tablas .....	viii
Siglas .....	x
1. Introducción.....	1
2. Estado del arte del video streaming.....	3
2.1. Video streaming en directo o Live Streaming.....	5
2.2. Video streaming bajo demanda - VoD.....	7
3. Formatos y protocolos de video streaming .....	11
3.1. Formatos de video streaming.....	11
3.1.1. Contenedores de video.....	11
3.1.2. Codecs de video .....	12
3.1.3. Codecs de audio.....	17
3.2. Protocolos de video streaming .....	19
3.2.1. Protocolos de video streaming no basados en HTTP .....	19
3.2.1.1. Real-Time Transport Protocol (RTP).....	20
3.2.1.2. Real-Time Streaming Protocol (RTSP).....	21
3.2.1.3. Real-Time Messaging Protocol (RTMP) .....	23
3.2.1.4. File Delivery over Unidirectional Transport Protocol (FLUTE) .....	25
3.2.2. Protocolos de video streaming basado en el protocolo HTTP.....	26
3.2.2.1. HTTP Live Streaming (HLS).....	29
3.2.2.2. Microsoft Smooth Streaming (MSS) .....	30
3.2.2.3. HTTP Dynamic Streaming (HDS).....	31
3.2.2.4. Dynamic Adaptive Streaming over HTTP (DASH) .....	32
4. Plataformas de Distribución de Contenidos .....	37
4.1. Redes de Distribución de Contenidos (CDNs).....	37
4.1.1. Tipos de CDNs .....	40
4.1.2. Atributos funcionales de una CDN.....	41
4.1.2.1. Composición de una CDN.....	41

4.1.2.2. Control y distribución del contenido .....	42
4.1.2.3. Encaminamiento de las solicitudes (Request - Routing) .....	45
4.1.2.4. Métricas de rendimiento de la CDN .....	47
4.1.3. Funcionamiento de una CDN .....	48
4.1.4. Desventajas de las CDNs .....	53
4.2. Peer-to-Peer (P2P).....	53
4.2.1. P2P Live Streaming.....	55
4.2.2. P2P Video on Demand (VoD) .....	57
4.2.3. Desventajas de las redes Peer-to-Peer (P2P).....	58
4.3. Modelos híbridos CDN-P2P.....	59
4.3.1. Clasificación según la entrega del contenido.....	60
4.3.2. Clasificación según el nivel de acoplamiento .....	61
4.4. Modelo Cloud - CDN (CCDN) .....	62
4.5. Comparación de las plataformas de distribución de contenidos .....	66
5. Última milla en la distribución de Video Streaming .....	67
5.1. Streaming sobre redes fijas (xDSL, HFC, FTTH).....	68
5.2. Streaming sobre redes Inalámbricas (Wireless - Redes móviles).....	69
5.3. Limitaciones actuales en la entrega del video streaming .....	74
6. Video Streaming a través de una Red Definida por Software SDN .....	80
6.1. Implementación de la red sobre Mininet .....	81
6.1.1. Escenario de un solo usuario (Servidor - Host 1).....	87
6.1.2. Escenario de más de un usuario (Servidor - Host 2 y Host 3).....	89
6.2. Resultados y conclusiones del experimento.....	91
7. Líneas futuras y Conclusiones .....	95
7.1. Líneas futuras.....	95
7.2. Conclusiones .....	98
Anexos.....	100
Bibliografía .....	107

## Índice de figuras

Figura 1. Crecimiento del tráfico IP (PB/Mes) (2016 -2019) [1].....	4
Figura 2. Ecosistema de la entrega de video extremo a extremo [3].....	5
Figura 3. Transmisión desde Ucrania mediante Facebook Live [4].....	6
Figura 4. Método de descarga y reproducción .....	7
Figura 5. Método de descarga progresiva .....	8
Figura 6. Modelos de VoD [5].....	9
Figura 7. Mensaje del DRM .....	10
Figura 8. Comparación de los diferentes tipos de resolución [11] .....	12
Figura 9. Evolución de los codecs de video .....	13
Figura 10. Comparación de los codecs H.264 vs H.265 .....	14
Figura 11. Video con diferentes resoluciones en Youtube .....	16
Figura 12. Contenido UHD ofrecido por Netflix.....	16
Figura 13. Comunicación con los protocolos RTP y RTCP .....	20
Figura 14. Sesión RTSP .....	22
Figura 15 Comunicación mediante RTMP [16].....	24
Figura 16. Comunicación FLUTE en un canal MBMS .....	26
Figura 17. Comparación del protocolo HTTP/1.1 y HTTP/2 .....	27
Figura 18. Buffering de un video .....	28
Figura 19. Transmisión de video utilizando la técnica ABR [8] .....	28
Figura 20. Estructura del archivo Manifests de HLS.....	29
Figura 21. Escenario con Microsoft Smooth Streaming (MSS) [16].....	31
Figura 22. Escenario con HTTP Dynamic Streaming de Adobe (HDS) .....	32
Figura 23. Estructura del Multimedia Presentation Description (MPD) [18].....	33
Figura 24. Arquitectura básica de una CDN .....	38
Figura 25. Ranking de las CDN's comerciales según Alexa 100 k .....	39
Figura 26. Relación entre los componentes de una CDN.....	41
Figura 27. Entrega de contenido estático.....	49
Figura 28. Arquitectura de un sistema Web .....	49
Figura 29. Arquitectura de Netflix.....	51
Figura 30. Sistemas P2P Live Streaming basados en Árbol .....	55
Figura 31. Sistemas P2P Live Streaming basados en Malla .....	57
Figura 32. Sistema P2P VoD basado en Árbol.....	58
Figura 33. Arquitectura híbrida P2P-CDN (PAC) de Akamai [48] .....	60
Figura 34. Arquitectura híbrida P2P-CDN (CAP)[50] .....	61
Figura 35. Arquitectura de un sistema Cloud-CDN (CCDN).....	64
Figura 36. Arquitectura para la entrega de video sobre Internet.....	67
Figura 37. Alcance de los estándares WiFi .....	71
Figura 38. Servicio de video en redes móviles.....	73
Figura 39. Ancho de banda necesario para video en SD, HD, UHD [65] .....	75
Figura 40. Ranking Netflix para los ISPs en España .....	77

Figura 41. Red tradicional vs SDN .....	81
Figura 42. Topología de la red implementada .....	83
Figura 43. Escenarios de análisis.....	87
Figura 44. Captura de la reproducción en el primer escenario .....	89
Figura 45. Captura de la reproducción en el segundo escenario .....	90
Figura 46. Variación del Bitrate en función del tiempo .....	91
Figura 47. Variación del Bitrate en función del ancho de banda disponible .....	92
Figura 48. Carga del buffer en función del ancho de banda disponible .....	93
Figura 49. Paquetes perdidos o descartados por los Hosts.....	93

## Índice de Tablas

Tabla 1. Global Consumer Internet Traffic, 2014–2019 [1].....	3
Tabla 2. Servicios de video con sus respectivos codecs .....	17
Tabla 3. Codecs soportados por los contenedores de video .....	18
Tabla 4. Protocolos de Streaming .....	19
Tabla 5. Características de los protocolos de Streaming.....	34
Tabla 6. Protocolos de Streaming según su estrategia PUSH/PULL.....	36
Tabla 7. Comparativa de las plataformas de distribución de contenidos .....	66
Tabla 8. Estándares WiFi .....	70
Tabla 9. Tecnologías de las redes de acceso a Internet.....	74



## Siglas

<b>ABR</b>	Adaptive Bit-Rate Streaming
<b>AL-FEC</b>	Application Layered - Forward Error Correction
<b>ALC</b>	Asynchronous Layered Coding
<b>AVoD</b>	Advertisement supported VoD
<b>CAGR</b>	Compound Annual Growth Rate
<b>CCDN</b>	Cloud-CDN
<b>CCN</b>	Content - Centric Networking
<b>CDN</b>	Content Delivey Network
<b>DASH</b>	Dynamic Adaptive Streaming over HTTP
<b>DHT</b>	Distributed Hash Table
<b>DNS</b>	Domain Name System
<b>DRM</b>	Digital Right Manager
<b>DSL</b>	Digital Subscriber Line
<b>FDT</b>	File Delivery Table
<b>FLUTE</b>	File Delivery over Unidirectional Transport
<b>FTTH</b>	Fiber to The Home
<b>HD</b>	High Definition
<b>HDS</b>	HTTP Dynamic Streaming
<b>HEVC</b>	High Efficiency Video Coding
<b>HFC</b>	Hybrid Fiber Coaxial
<b>HLS</b>	HTTP Live Streaming
<b>HSPA</b>	High Speed Packet Access
<b>HTTP</b>	HiperText Transfer Protocol
<b>ICN</b>	Information - Centric Networks
<b>IETF</b>	Internet Engineering Task Force
<b>IIS</b>	Internet Information Services
<b>IPTV</b>	Internet Protocol Television
<b>ISDN</b>	Integrated Services for Digital Network
<b>ISO</b>	International Organization for Standardization
<b>ISP</b>	Internet Service Provider

<b>ITU-T</b>	International Telecommunication Unit
<b>LTE</b>	Long Term Evolution
<b>MBMS</b>	Multimedia Broadcast Multicast Service
<b>MMT</b>	MPEG Media Transport
<b>MPD</b>	Media Presentation Description
<b>MPEG</b>	Moving Picture Experts Group
<b>MSS</b>	Microsoft Smooth Streaming
<b>NAT</b>	Network Address Translation
<b>NFV</b>	Network Functions Virtualization
<b>OTT</b>	Over-The-Top
<b>P2P</b>	Peer-to-Peer
<b>PaaS</b>	Platform as a Service
<b>PoP</b>	Points-of-Presence
<b>QoE</b>	Quality of Experience
<b>QoS</b>	Quality of Service
<b>ROUTE</b>	Real-time object delivery over unidirectional transport
<b>RTMP</b>	Real-Time Messaging Protocol
<b>RTP</b>	Real-Time Transport Protocol
<b>RTSP</b>	Real-Time Streaming Protocol
<b>SD</b>	Standard Definition
<b>SDN</b>	Software Defined Network
<b>SMPTE</b>	Society of Motion Picture & Television Engineers
<b>SPDY</b>	Speedy
<b>SVoD</b>	Subscription VoD
<b>TCP</b>	Transmisión Control Protocol
<b>TVoD</b>	Transactional VoD
<b>UDP</b>	User Datagram Protocol
<b>UHD</b>	Ultra High Definition
<b>VoD</b>	Video on Demand
<b>VoIP</b>	Voice over IP
<b>XML</b>	eXtensible Markup Language



## 1. Introducción

Internet ha experimentado un crecimiento exponencial, debido en gran parte a que muchas empresas, tecnológicas y no tecnológicas, han apostado por él para expandir su forma de hacer negocios y la forma de interactuar con sus clientes. Producto de ello surgieron nuevas oportunidades de negocios que algunos visionarios aprovecharon y que hoy en día constituyen las mayores empresas de tecnología y servicios Web. Los casos más claros son Youtube, Facebook, Amazon, Netflix, Google, entre muchos más.

Además del crecimiento de Internet, han habido grandes avances tecnológicos en el sector de la electrónica de consumo. Dispositivos como los smartphones, Smart TV's, PC's entre otros; cada vez cuentan con más recursos computacionales y funciones más avanzadas que permiten la posibilidad de que nuevos servicios puedan ser ofrecidos a través de ellos. Entre estos servicios se encuentran los servicios multimedia y en especial el servicio de video streaming.

Todo esto ha generado muchos desafíos y retos para la industria, ya que con estos nuevos servicios, se ha multiplicado enormemente el tráfico que circula a través de Internet. Hoy en día se estima que más del 50% del tráfico de datos que circula en Internet corresponde a los servicios multimedia, y se prevé que para el 2020, el 75% del tráfico en la Internet corresponderá a contenido multimedia. [2]

Es por ello que se han realizado un amplio abanico de investigaciones e implementaciones para mejorar la eficiencia de las redes. Hoy en día, los proveedores de contenido disponen de las CDNs para la distribución del mismo a los usuarios finales. Además, se han desarrollado nuevos protocolos y nuevos codecs de audio y video que permiten transportar el contenido multimedia y comprimir el audio y video respectivamente. Esto ha permitido mejorar la QoS y la QoE de los usuarios finales. La QoE es una medida subjetiva de la experiencia de un usuario ante un servicio prestado, en este caso el servicio multimedia. La QoE esta basada en la aceptabilidad de un servicio por parte del usuario y depende en gran medida de la QoS prestada.

Las CDNs permiten mejorar varias métricas que influyen en la QoS que se ofrece al usuario. Mejoran el aprovechamiento del ancho de banda, disminuyen la latencia, y gracias a ellas, es posible tener servicios de video streaming de una calidad aceptable e incluso de ultra alta definición - Ultra High Definition (UHD). Es por ello que las CDNs juegan un role muy importante dentro de la arquitectura de las proveedores de video streaming sobre Internet.

Por otro lado, la entrega de video streaming de alta calidad está atada a a aspectos tanto técnicos como económicos. Dentro de los aspectos técnicos, tenemos que aún, en muchos sitios, el acceso a la red ofrecido por los proveedores de servicios de Internet - Internet Service Providers (ISPs) aún constituyen en un cuello de botella, lo cual provoca que los usuarios finales no puedan recibir video streaming de UHD; en

algunas ocasiones, reciban solamente video streaming con una High Definition (HD) e incluso con una definición standard - Standard Definition (SD). En cuanto a los aspectos económicos, juegan un papel muy importante, ya que muchos proveedores de contenidos necesitan contratar los servicios de terceros para poder realizar la entrega del contenido a los usuarios finales, en este caso, la contratación de las CDNs en donde se acuerdan los costes por la utilización de su infraestructura. En [7] se demuestra como este aspecto limita que un usuario pueda disfrutar del servicio de video streaming de alta definición cuando hay congestión en la red; y en realidad si que podría disfrutar mediante la utilización de otras CDNs, pero esto implicaría aumentar los costos por parte de las CDNs a los proveedores de contenido. Además, muchos ISPs también ponen trabas para la entrega de video streaming de alta calidad ya que los proveedores de contenidos utilizan la infraestructura de los ISPs y estos limitan el ancho de banda de estos servicios dado que no llegan a acuerdos económicos que satisfagan a ambas partes.

Los objetivos del presente trabajo es, en primer lugar, realizar un análisis del estado del arte de los componentes que hacen posible el video streaming de alta calidad; desde los proveedores de contenido con los que contamos actualmente, los tipos de codecs que se utilizan, los protocolos de video streaming, tipos de plataformas de distribución de contenido, redes de acceso tanto fijas como móviles y finalmente analizar las líneas futuras para que nuevos servicios multimedia sean posibles. En el capítulo 6 se realizará una implementación de una red para el análisis del funcionamiento del protocolo DASH cuando hay variaciones del ancho de banda en una red.

## 2. Estado del arte del video streaming

La popularidad del video streaming ha incrementado considerablemente durante la última década, tanto así, que de acuerdo al informe anual VNI de Cisco, hoy en día, más del 50% del tráfico IP global corresponde a tráfico de video streaming y se estima que para el 2019 entre el 80% y el 90% de tráfico IP global corresponda a video (TV, VoD, Internet, P2P).[1]

Este impacto en el tráfico afectarán tanto a las redes fijas actuales como a las redes móviles de telefonía. Tan solo en redes móviles se estima que para el 2020, el tráfico IP de video que circulará a través de ellas será el 75% del tráfico total IP. [2]

En la Tabla 1 podemos observar que según el informe de CISCO VNI del año 2015, preveía que para el 2016 el tráfico de Internet en las redes fijas, por parte del consumidor, sería de 46.511 Petabytes (PB) por mes y con una proyección para el 2019 de 91.048 PB por mes. Por otro lado, en las redes móviles vemos que el tráfico para el 2016 sería de 5.599 PB por mes y que para el 2019 serán 20.544 PB por mes. Como podemos observar, las redes móviles serán las que experimenten mayor crecimiento con una tasa compuesta de crecimiento anual - Compound Annual Growth Rate (CAGR) del 59% entre el 2014 y el 2019. Actualmente, el tráfico del video sobre Internet es de 36.456 PB por mes y en el 2019 será aproximadamente de 89.319 PB por mes.

Consumer Internet Traffic, 2014–2019							
	2014	2015	2016	2017	2018	2019	CAGR 2014–2019
<b>By Network (PB per Month)</b>							
Fixed	31,545	37,908	46,511	58,115	72,933	91,048	24%
Mobile	2,050	3,430	5,599	8,906	13,587	20,544	59%
<b>By Subsegment (PB per Month)</b>							
Internet video	21,624	27,466	36,456	49,068	66,179	89,319	33%
Web, email, and data	5,853	7,694	9,476	11,707	14,002	16,092	22%
File sharing	6,090	6,146	6,130	6,168	6,231	6,038	0%
Online gaming	27	33	48	78	109	143	40%
<b>Total (PB per Month)</b>							
Consumer Internet traffic	33,595	41,338	52,110	67,021	86,520	111,592	27%

Source: Cisco VNI, 2015

Tabla 1. Global Consumer Internet Traffic, 2014–2019 [1]

Con tales proyecciones y con la continua popularidad de los servicios de video streaming que se entregan a través de Internet; las redes de distribución de contenido se han convertido en una componente clave y en un método predominante para la distribución y entrega de este contenido a los usuarios finales. A nivel mundial, se prevé que en el 2019, el 62% del tráfico global de Internet pasará por las CDNs, de los cuales el 72% del total corresponderá al video. Actualmente, se estima que unos 31.345 PB por mes del tráfico de Internet pasan a través de las CDNs y se espera que para el 2019 pasen cerca de 84.301 PB por mes.[1] (Figura 1).

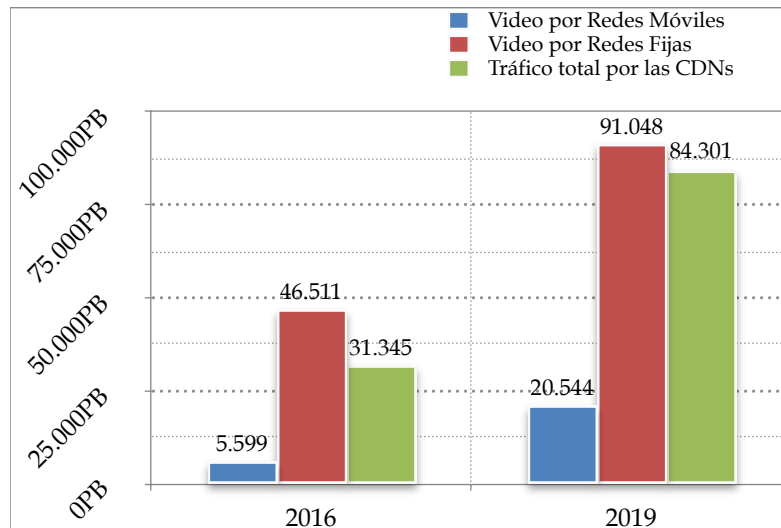


Figura 1. Crecimiento del tráfico IP (PB/Mes) (2016 -2019) [1]

Todo esto no parece tan descabellado, teniendo en cuenta que cada cierto tiempo nos encontramos con nuevos servicios y aplicaciones que permiten ver y compartir el video, ya sea video streaming producido de forma profesional como las series de Netflix, AMC, NBO, etc; o video producido de forma aficionada o no profesional como los videos que se cuelgan en Youtube, DailyMotion, videos de 360 grados, Facebook live, Periscope, entre otros. Estos últimos han tomado una gran relevancia en el último año ya que aparte de que permiten transmitir video en vivo, han sido utilizados para poder informar de una forma más inmediata los acontecimientos que suceden a nivel mundial.

Además de todo esto, tenemos que los canales clásicos de TV han empezado también a transmitir sus programas a través de Internet para no quedarse atrás en esta nueva forma de llegar a los usuarios. Es así que hoy en día tenemos dos tipos de sistemas que ofrecen servicios de video streaming a través de Internet. El primero es TV sobre IP también conocido como Internet Protocol Televisión (IPTV) y abarca a cualquier servicio de televisión, en vivo o Live Streaming y bajo demanda o Video On Demand (VoD). IPTV puede entregar datos multimedia de forma fiable; los mismos que se encuentran codificados con una tasa constante de bits; además, asegura cierto ancho de banda para poder ofrecer cierto nivel de QoS. Este sistema es utilizado por lo general por las cadenas de Televisión, suele ser de pago y hace énfasis proveer cierta QoS. En Live streaming la transmisión se realiza de modo multicast, mientras que para VoD la transmisión se realiza de modo unicast. El segundo tipo de sistema se denomina Over-The-Top (OTT) y al igual que IPTV, puede ofrecer servicios de VoD y Live streaming. En este servicio no reserva un ancho de banda fijo para los usuarios, por lo tanto la QoS no está garantizada; es por ello que se utilizan protocolos especiales que adaptan el streaming de los datos a las características de la red. Estos protocolos son basados en el protocolo HTTP, monitorizan periódicamente el ancho de banda y las capacidades del equipo del usuario (buffer, memoria, características del reproductor,

etc). Este sistema por lo general es usado para servicios VoD en donde el contenido se encuentra almacenado en servidores (ej. series, películas, etc). Por último, este servicio puede ser de pago mediante suscripciones y también de forma gratuita como veremos más adelante.

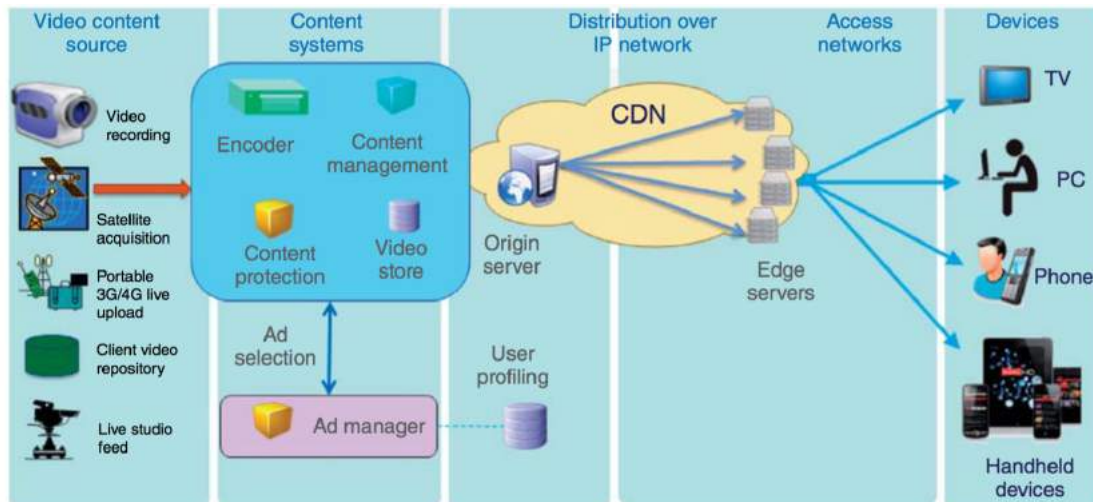


Figura 2. Ecosistema de la entrega de video extremo a extremo [3]

En la Figura 2 podemos observar el ecosistema de la entrega del video extremo a extremo, en donde se describe los componentes que conforman el mismo. Tenemos la fuente de donde genera el contenido, puede ser en vivo, o proceder de los servidores donde se encuentra almacenado. A continuación se realiza la tarea de la gestión del contenido en donde se modifica el video, se protege y se almacena. Luego pasa al almacenamiento y alojamiento del contenido en las CDNs, las cuales se encargarán de distribuir el contenido hacia sus servidores sustitutos que se encuentran estratégicamente distribuidos a nivel global con el objetivo de acercar la información y contenido a los usuarios finales. Por último, tenemos el acceso a la red en donde entran los ISPs, los cuales entregan el contenido a los dispositivos de los usuarios finales. Este contenido puede ser VoD o Live Streaming.

## 2.1. Video streaming en directo o Live Streaming

La transmisión de contenido en directo provocan un mayor uso de las CDNs debido a que el contenido debe estar disponible para todos los dispositivos que soliciten este servicio. Live streaming es utilizado para cubrir eventos deportivos, culturales, informativos, etc. Este servicio está orientado a la multidifusión; además, los eventos tienen lugar en un lapso de tiempo definido, es decir, sabemos cuando empieza y cuando terminará, por ejemplo, un partido de fútbol sabemos que dura 90 minutos. Como característica principal de este servicio es que no existe interactividad con los usuarios, únicamente se puede pausar la transmisión.

Entre los proveedores que generan contenido en vivo, tenemos a los canales de TV clásicos, desde los mundialmente conocidos como la CNN, BBC, HBO, etc; hasta los

menos conocidos que también han empezado a ofrecer este servicio para no quedarse atrás de la competencia. En España uno de los principales proveedores que ofrecen este tipo de servicio es A Tres Media. Dispone de una aplicación para dispositivos móviles llamada ATresPlayer<sup>1</sup> en donde podemos ver la transmisión en directo de canales de TV como Antena 3, Neox, Nova, y todos los canales pertenecientes al grupo de A Tres Media. Por lo general, el servicio de video streaming en directo es un servicio de pago, sobre todo, para los eventos deportivos como las carreras de formula 1, partidos de fútbol, partidos de tenis, etc.

Otras aplicaciones disponibles para dispositivos móviles y ordenadores son Facebook Live, Periscope de Twitter, etc; las cuales permiten transmitir video en directo, pero en este caso, son los usuarios de los dispositivos quienes generan el contenido a transmitir. (Figura 3)

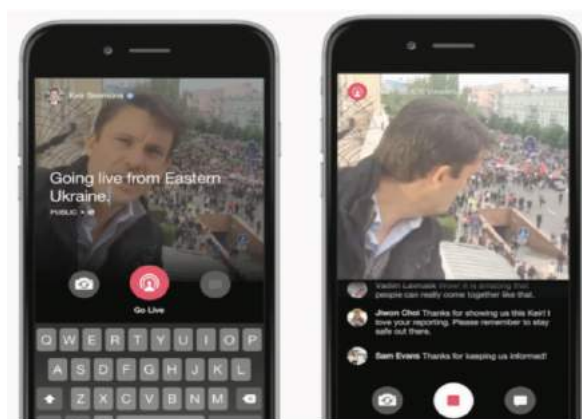


Figura 3. Transmisión desde Ucrania mediante Facebook Live [4]

Mediante la utilización de los protocolos basados en HTTP como DASH, Microsoft Smooth Streaming (MSS), HTTP Live Streaming (HLS), etc. El reproductor del usuario final será capaz de seleccionar una tasa de bits que mejor se ajuste a las condiciones de la red durante la transmisión y además puede ir cambiando dinámicamente la tasa de bit si las condiciones de la red mejoran o empeoran sin que ello afecte a la reproducción del contenido.

Es muy importante que la configuración en el origen sea la adecuada para evitar un impacto negativo en las CDNs y en la QoS durante la transmisión. Las CDNs obtienen el contenido de Live streaming desde el origen y lo distribuye hacia todos los servidores sustituto para la entrega a los usuarios finales.

Dado que el Live Streaming a través de internet es relativamente nuevo, en la práctica está generando un gran revuelo con respecto a los derechos de propiedad intelectual, debido a que muchos usuarios han empezado a utilizar las aplicaciones que permiten la transmisión en vivo para retransmitir de manera ilegal los evento

<sup>1</sup> <http://www.atresplayer.com>



espacio de disco en muy limitado. Además, el usuario tendrá que esperar hasta que la descarga del archivo se complete, todo dependerá de la velocidad de conexión a Internet que le ofrece su ISP. (Figura 4)

• **Descarga Progresiva.-** Ahora la reproducción puede realizarse simultáneamente mientras el contenido se está descargando desde los servidores donde está almacenado. Una vez que el contenido descargado es reproducido, se elimina del dispositivo del cliente, es decir, ya no se almacenará en el disco duro del dispositivo. Al inicio, el contenido se descarga con más frecuencia dado que tiene que llenar el buffer del reproductor, luego la descarga del contenido es progresiva mientras el usuario reproduce el contenido que ha solicitado<sup>3</sup>. La desventaja es que cuando un usuario quiere posicionar en un cierto punto la reproducción de un video, una película por ejemplo, al hacerlo tendrá que esperar a que se descargue el contenido para llenar el buffer lo suficiente para que inicie la reproducción. Youtube, Netflix, entre otros utilizan este método para entregar el contenido a los usuarios.(Figura 5)

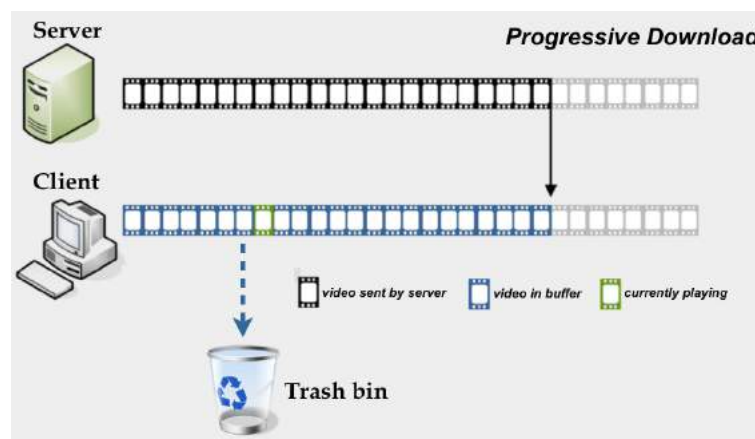


Figura 5. Método de descarga progresiva

Actualmente tenemos 3 modelos de VoD que dominan la industria. Estos son:

• **Modelo VoD bajo suscripción (SVoD).**- En este modelo, el usuario debe pagar una cuota mensual para acceder a los servicios de forma ilimitada. Entre las principales compañías que ofrecen este tipo de servicio tenemos a Netflix, HuluPlus, HBO GO, canales de las ligas deportivas americanas como la NBA, entre otras.

• **Modelo VoD con soporte de anuncios (AVoD).**- En este modelo, el contenido es gratuito para el usuario, a cambio de que el usuario vea los anuncios comerciales que se reproducen al inicio del contenido. Suelen ser anuncios cortos, de no más de 30 segundos; una vez que el anuncio ha sido reproducido el usuario podrá disfrutar del contenido que ha solicitado. Entre las principales compañías que ofrecen este tipo de servicio tenemos a Youtube, Crackle, entre otras.

<sup>3</sup> <https://www.linkedin.com/pulse/making-sense-video-streaming-protocols-dr-brijesh-kumar>



• **Modelo VoD con transacciones (TVoD).**- Este modelo es lo opuesto a SVoD, ahora el cliente compra o alquila de contenido individual, por ejemplo, una película que la puede ver durante un día o para tenerla almacenada para verla cuando quiera. Entre las principales compañías que ofrecen este servicio tenemos iTunes de Apple, Google Play, entre otras. Este modelo por lo general se realiza con el método de descarga y reproducción.

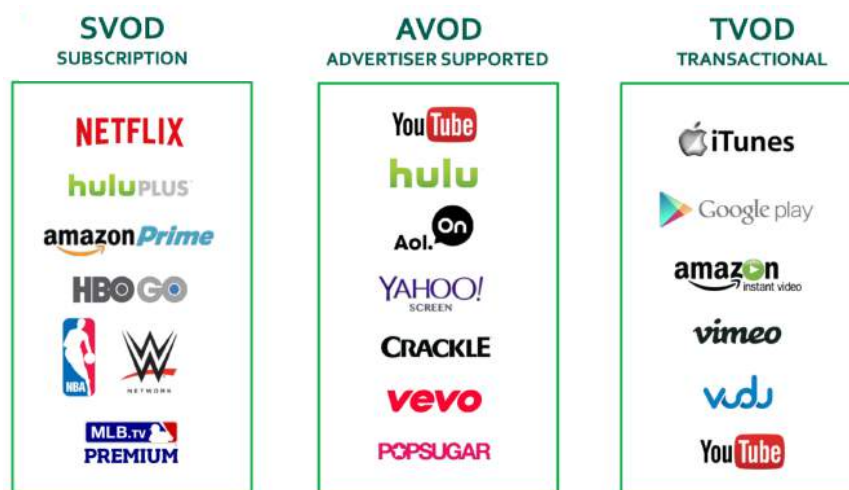


Figura 6. Modelos de VoD [5]

En España, además contar con los proveedores de contenidos como Netflix, Youtube entre otros; se cuenta también con compañías nacionales que ofrecen este mismo servicio y que por lo general son SVoD. Entre estas compañías tenemos a Yomvi, Wuaki, Movistar Series, entre otras<sup>4</sup>. En la Figura 6 podemos observar los diferentes tipos de modelos descritos y las principales compañías que los ofrecen.

Para la reproducción del contenido de VoD en los dispositivos finales, previamente, el cliente debe instalar los plugins necesarios en su dispositivo, que le permitirán reproducir el contenido que ha solicitado al proveedor de VoD. Entre los más conocidos tenemos a Silverlight de Microsoft, Adobe Flash Player, Move Networks Media Player, QuickTime, entre otros.[8] Por otro lado, para otros dispositivos como las Smart TV's, necesitarán decodificadores dedicados que les permitan visualizar el contenido. Entre estos tenemos a Roku, Apple TV, Android TV, y Amazon Fire, etc. Otros dispositivos que actúan como decodificadores y que no son dedicados, son las consolas de video juegos como la Xbox 360 de Microsoft, PlayStation de Sony, y Wii de Nintendo. [5]

Un elemento que juega un papel muy importante en la distribución de video streaming a través de Internet y que aún no le hemos mencionado es el Gestor de Derechos Digitales - Digital Right Manager (DRM). El DRM es el que garantiza que el contenido que solicita el cliente puede ser reproducido en una determinada localiza-

<sup>4</sup> <http://www.mediatrends.es/a/35318/netflix-espana/>

ción. Por ejemplo, si en EEUU se estrena una serie de televisión y el lanzamiento para Europa es el siguiente mes, cualquier usuario que quiera acceder a ese contenido desde Europa no lo podrá hacer de forma legal. Lo mismo sucede cuando queremos ver un video en Youtube y hay cierto contenido que no puede ser reproducido dependiendo del país donde se lo solicita.[9]

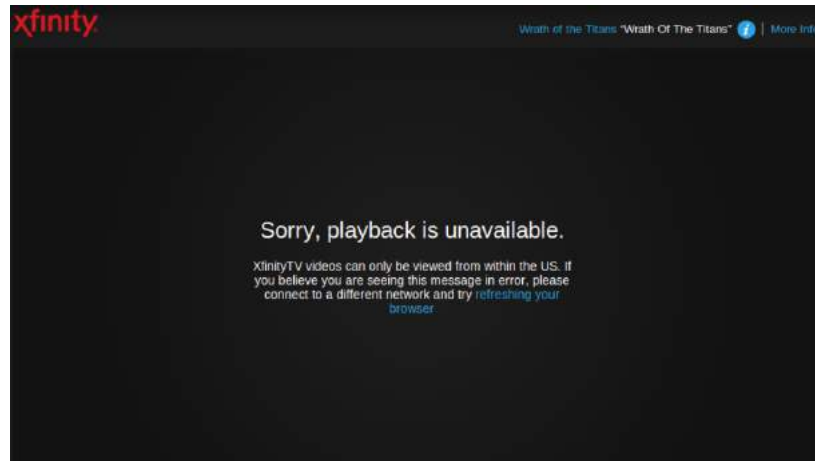


Figura 7. Mensaje del DRM<sup>5</sup>

En la Figura 7 podemos observar el típico mensaje que aparece cuando un contenido no se encuentra disponible debido a la localización del usuario. Entre los DRMs que podemos encontrar en el mercado tenemos a los DRMs de Microsoft PlayReady, Adobe Flash Access, Marlin<sup>6</sup> que es un estándar abierto, y NDS VGC de CISCO. [8]

En el siguiente capítulo analizaremos los formatos de video y los principales protocolos que se utilizan para la transmisión de video streaming, estudiaremos las características de cada uno y que ventajas tiene frente a los demás.

---

<sup>5</sup> <http://vpnfor.us/how-to/unblock-comcast-xfinity-tv/>

<sup>6</sup> <http://www.marlin-community.com>

### 3. Formatos y protocolos de video streaming

Para entender como funciona el video streaming, primero debemos conocer que hay detrás de ello. Los formatos de video streaming son muy importantes ya que de ellos depende que el video tenga cierta calidad cuando llega a los usuarios finales.

Un generador de contenidos necesitará que el contenido tenga un formato adecuado que le permita ser transmitido. En este caso, el contenido tendrá que pasar por un proceso de codificación o compresión que le permita reducir su tamaño en comparación con el video original. Para ello es necesario la utilización de codecs de audio y video. Mediante el uso de los codecs podemos mejorar la calidad de la imagen/audio reduciendo la tasa de bits necesarios para la reproducción del video. Mientras que, mediante los protocolos de video streaming conseguimos que el video codificado sea entregado a los usuarios finales a través de las CDNs. A continuación vamos a describir los diferentes tipos de codecs de video que se utilizan hoy en día y posteriormente analizaremos los principales protocolos que se utilizan para la distribución del video a través de Internet.

#### 3.1. Formatos de video streaming

Hoy en día disponemos de muchos formatos de video que nos permiten almacenar el contenido multimedia en un solo archivo. Según el formato del video tendremos una determinada calidad y diferentes tamaños del video; todo dependerá de los tipos de codecs utilizados para la compresión del video.

Los formatos de video están compuestos por los contenedores del video y los codecs de video y audio; por ejemplo, .MP4 es solo un contenedor, puede contener desde video de baja calidad SD hasta video de ultra alta calidad UHD y con varias calidades de audio del mismo contenido. A continuación vamos a ver los principales contenedores, codecs de video y codecs de audio que permiten la generación del video streaming.

##### 3.1.1. Contenedores de video

Un contenedor, como su nombre lo indica, contiene varios componentes del video como son: la imágenes del video, audio, metadatos, etc. Es por ello que en un archivo de video podemos encontrar el audio en varios idiomas, subtítulos, etc; siempre y cuando el formato del contenedor lo permita. Entre los formatos de los contenedores más conocidos tenemos a OGG, 3GP, AVI, MOV, MP4, Flash Video (FLV), WebM de Google, Matroska (MKV), entre otros. Entre los codecs de video, audio y metadatos tenemos:

✓ *Codec de video:* H.263, H.264, H.265, VP8, MPEG-4, etc.

✓ *Codec de audio:* Advanced Audio Coding (AAC), MP3, Vortis, Windows Media Audio (WMA), etc.

✓ *Metadatos:* XML, RDF, XMP, SRT, etc.

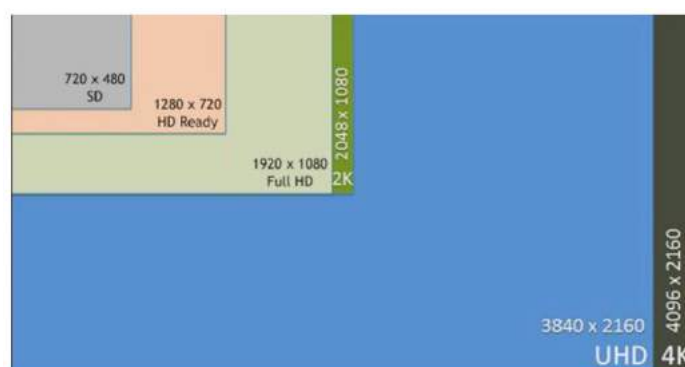
Cada archivo de video contiene ciertos atributos que describen a la señal de video que se transmitirá. La calidad del video y el ancho de banda que se utilizará dependen de estos atributos de la señal de video. Estos atributos son:

✓ **El tamaño del Marco (Frame size):** Es la resolución que tiene una imagen, cuanto mayor sea la resolución, mayor calidad tendrá el video, pero también requerirá más ancho de banda para la transmisión. En la Figura 8 podemos observar los diferentes tipos de resolución según la calidad del video.

✓ **La tasa del Marco (Frame rate):** Es la velocidad con que los marcos son capturados y posteriormente reproducidos. La tasa del marco más comunes son 15 frames per second (fps), 24 fps, 25 fps, 30 fps, y 60 fps.

✓ **Tasa de bit (Bitrate):** La tasa de bit es el número de bits que se transmiten en un tiempo determinado. El bitrate es una combinación entre el contenido de video y del audio en el archivo. Normalmente un Bitrate (Tasa de bit) alto implica una mejora calidad del video y un tamaño del mismo mayor.

✓ **Tasa de muestreo del audio:** El audio a su vez, también es comprimido usando los diferentes codecs. Dependiendo del codec que se utilice añadirá entre unos 128 - 256 kbps a la tasa de bit.



Source: Les Numériques, July 2014

Figura 8. Comparación de los diferentes tipos de resolución [11]

A continuación veremos algunos codecs de video usados para la transmisión de video streaming.

### 3.1.2. Codecs de video

El video sin compresión requiere una enorme cantidad de recursos, por ejemplo, el almacenamiento, ancho de banda en la red, etc. Es por ello que se requieren técnicas de compresión digital del video original y así reducir la redundancia del mismo. Hay dos tipos de codecs de video; los codecs con pérdidas y sin pérdidas. Aunque la mayoría de codecs son con pérdidas, el objetivo es minimizar las mismas mediante nuevas técnicas que permitan comprimir el video manteniendo una buena calidad en la imagen.

Muchos estándares de codecs de videos han sido propuestos desde los 80's, los cuales especifican las técnicas de codificación del video que utilizan las diferentes aplicaciones. (Figura 9)

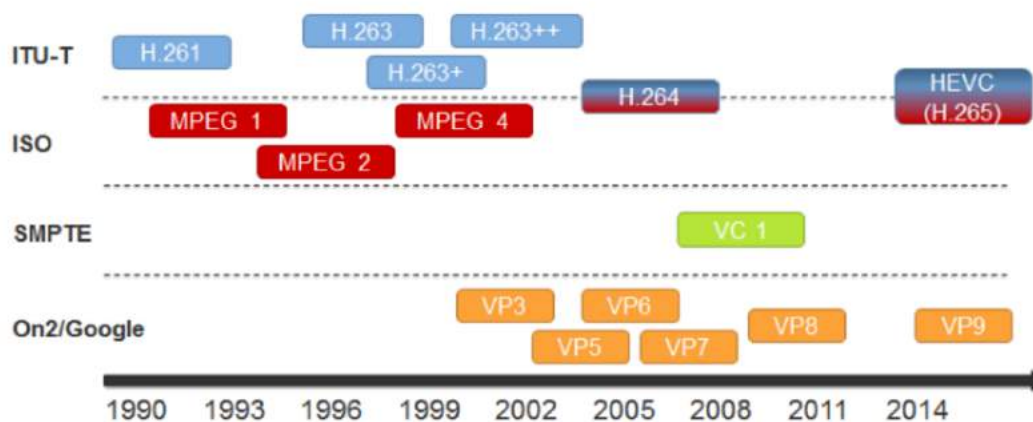


Figura 9. Evolución de los codecs de video<sup>7</sup>

Detrás del desarrollo de los codecs de video tenemos a varios organismos y organizaciones, como la International Telecommunication Unit (ITU-T), la International Organization for Standardization (ISO), la Society of Motion Picture & Television Engineers (SMPTE) y On2/Google.

La ITU-T ha desarrollado una gran variedad de codecs de video, el primero fue el **H.261**, a principios de los 90's, y que fue utilizado para aplicaciones que requerían baja latencia como las video conferencias y video llamadas a través de las Redes Digitales de Servicios Integrados (ISDNs), con una resolución del video pequeña (352x288 o 176x144). 352x288 quiere decir que tenemos 288 líneas de 352 pixeles cada una. La tasa de bits es múltiple de 64 Kbps y para la codificación, se divide la imagen en bloques de 16x16 pixeles.

Posteriormente se estandarizó el codecs **H.263**, y poco tiempo después a **H.263+** y **H.263++**. Este codec ofrecía una mejor calidad de video que sus antecesores con una resolución que van desde los 128x96 pixeles hasta los 1408x1152 pixeles; divide la imagen en bloques de 16x16 bloques y sub-bloques de 8x8 pixeles para las regiones de la imagen con más cambios. El codec H.263 fue utilizado para la compresión de los videos que se reproducían en las plataformas de Youtube, Google Play, etc. Actualmente se sigue utilizando este codec en los dispositivos que lo soportan.

Por su parte, la ISO estandarizó el codec **MPEG-1** que fue desarrollado por el comité Moving Picture Experts Group (MPEG), proveía una resolución de 352x288 pixeles para el sistema PAL y 352x240 para el sistema NTSC, con una tasa de bit cercana a 1.5 Mbps. Su aplicación principal fue para el almacenamiento de video en CDs.

<sup>7</sup> <https://www.linkedin.com/pulse/brief-history-video-codecs-yoav-nativ>

A mediados de los 90's se estandarizó el codec **H.262/MPEG-2**, que fue desarrollado conjuntamente por la ISO y la ITU-T. Este codec ofrecía mejor calidad y resolución de video que sus antecesores mantenido la misma tasa de bits, la aplicación de este codec fue principalmente en la compresión de películas en DVDs y transmisión de televisión digital, tiene varios tamaños de resolución que van desde los 176x144 pixeles con una tasa de bit de 96 Kbps hasta una resolución de 1920x1080 pixeles con una tasa de bits de 80 Mbps. Hoy en día hay codecs que ofrecen la misma resolución con una menor tasa de bits.

A finales de los 90's, la ISO estandarizó el codec **MPEG-4** que fue desarrollado por el comité MPEG. Este nuevo codec tiene varias partes; mejora la calidad del video que sus antecesores y además ofrece características nuevas que incluyen los subtítulos en el video y protección de derecho de autor. Se usa principalmente en la codificación de contenido web, video streaming, video sobre redes móviles 3G, 4G, etc.

A inicios del año 2001 se presentó el codec **H.264/AVC/MPEG-4 Part 10** que mejoraba las características de compresión del codec H.263. Se le conoce también como Codificación Avanzada de Video (AVC). Este estándar fue desarrollado conjuntamente entre la ITU-T y la ISO y es capaz de proveer una buena calidad de video con una tasa de bits más baja que los estándares anteriores, es por ello que fue usado en las primeras video conferencias en alta definición HD. Además, permitió que se pueda ofrecer video de mejor calidad en la plataformas de video sobre Internet. Esto es posible gracias a que permite la fragmentación de las imágenes en bloques cambiantes según los cambios en las imágenes. Las partes de las imágenes que tenían menos cambios se asignan bloques de 16x16 pixeles, mientras que en las partes que tienen más cambios se les asigna sub-bloques de 16x8, 8x16, o 8x8 pixeles y este último a su vez se puede descomponer en sub-bloques de 8x4, 4x8 o 4x4. De este modo se mejora la compresión de la imagen dado que se elimina más redundancia que con el codec H.263, el cual tenía el tamaño de bloque de 16x16 pixeles con sub-bloques de 8x8 pixeles. [12]

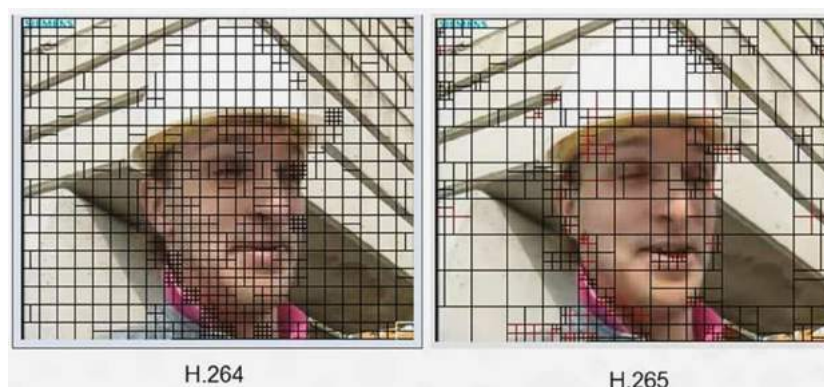


Figura 10. Comparación de los codecs H.264 vs H.265 <sup>8</sup>

<sup>8</sup><http://hipertextual.com/2015/04/h-265-y-vp9>

El último codec estandarizado por la ITU-T y la ISO es el High Efficiency Video Coding o **HEVC**, también conocido como H.265. Este nuevo estándar ofrece el mismo nivel de calidad de imagen que H.264 pero con una mejor compresión. Esta nueva característica es clave para la transmisión de video en ultra alta definición UHD, también conocida como video 4K. Este nuevo codec, al igual que H.264, fragmenta la imagen en bloques, pero en este caso de 64x64 píxeles que pueden dividirse en sub-bloques para las partes de las imágenes que tengan mayores cambios llegando a sub-bloques de 4x4 en las zonas de mayor cambio. Mejora el modelo de predicción espacio temporal de H.264 y puede soportar hasta 360 fps, H.264 soporta hasta 60 fps. En la Figura 10 podemos ver la diferencia entre el codec H.264 y el codec H.265 o HEVC, en cuanto a la división de la imagen en bloques. [37]

Por su parte, la SMPTE en el 2006, publicó el estándar del codec de video **VC-1** con el propósito de contar con una alternativa a los codecs desarrollados por la ITU-T y la ISO. Microsoft<sup>9</sup> empezó a implementar este codec en sus productos. La calidad de la imagen va desde la baja calidad hasta la alta calidad, puede contar con una resolución de 1920x1080 píxeles con una tasa de bits entre 6 a 30 Mbps; para una resolución de 2048x1536 píxeles necesita unos 135 Mbps. Por otro lado, para resoluciones bajas 160x120 píxeles tiene una tasa de bit de 10 Kbps. VC-1 divide la imagen en bloques de 16x16 píxeles y a su vez divide estos en sub-bloques de 8x8, 8x4, 4x8, y 4x4 píxeles.

Por su parte, On2/Google ha desarrollado sus propios codecs de compresión de video. Iniciaron a principios del año 2000 con el codec de video VP3 desarrollado por On2 Technologies. En el 2010 Google compró On2 Technologies y cuando lo hizo liberó las patentes del codec de video VP8 que estaba ese momento en el mercado. Hasta ese momento el codec que más se utilizaba para video conferencias y VoD era H.264. VP8 tiene características similares a H.264. Google utiliza el formato de video .WebM como contenedor de video, el cual es codificado con VP6 (usado en Flash), VP7 (usados en Skype), VP8 (usado en HTML5) y el audio codificado con el codec Vorbis.

A finales del 2011 se lanzó el codec VP9 que es de código abierto (open source), este codec reducía la tasa de bits en un 50% manteniendo la misma calidad de imagen que VP8. Es el equivalente al codec H.265 de las ITU-T/ISO, se lo conoce como Next Gen Open Video (NGOV) y Google lo implementado en su navegador Web "Chrome" y en Youtube. Este codec proporciona, al igual que H.265, video UHD o 4K y permite que los videos de una calidad más baja sean más fáciles de descargar para las conexiones de Internet lentas. En la Figura 11 podemos observar la imagen de un video con una resolución UHD o 4K de 4096x2160 píxeles. Actualmente, Google se encuentra desarrollando un nuevo codec denominado VP10 que prevé mejorar la calidad de la imagen para nuevos servicios como 8K y a la vez reducir la tasa de bits para mejorar el

---

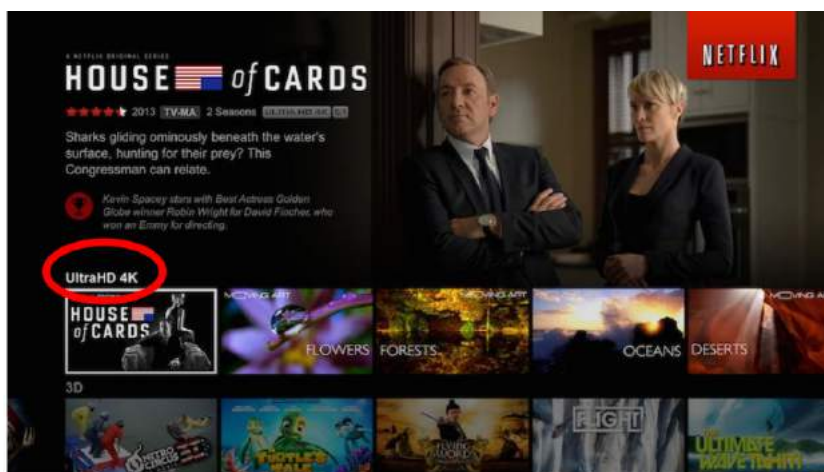
<sup>9</sup> <https://www.microsoft.com/windows/windowsmedia/howto/articles/vc1techoverview.aspx>

aprovechamiento del ancho de banda. Prevén que el nuevo codec esté listo para finales del 2016<sup>10</sup>.



Figura 11. Video con diferentes resoluciones en Youtube <sup>11</sup>

En Septiembre del 2015 se fundó el grupo Alliance for Open Media que lo integran grandes compañías de Internet como son Amazon, Cisco, Google, Intel Corporation, Microsoft, Mozilla y Netflix. El objetivo de esta alianza es desarrollar un nuevo codec de video que sea open source, que sea el sucesor a VP9 y una alternativa a HEVC de la ITU-T/ISO. Actualmente este codec se encuentra en fase de desarrollo pero han anunciado que tienen previsto que el nuevo codec vea la luz en Enero del 2017. [13]



Source: Netflix

Figura 12. Contenido UHD ofrecido por Netflix

Actualmente los proveedores de contenido VoD ya ofrecen servicios de video de alta definición en sus plataformas. En Mayo del 2014, Netflix estreno la segunda temporada de House of Cards que fue emitida con calidad UHD/4K; en el mismo año a mediados

<sup>10</sup> <http://www.techtimes.com/articles/80948/20150902/googles-vp10-codec-promises-to-cut-down-the-size-of-4k-videos-in-half.htm>

<sup>11</sup> <https://www.youtube.com/watch?v=wTcNtgA6gHs>



de Junio disponía de otros 62 episodios de otras series de renombre como Breaking Bad e incluyendo además algunas películas como los Caza fantasmas y los Pitufos 2. (Figura 12) [11]

Hasta ahora hemos visto la codificación en el lado del generador de contenido pero también se necesitan dispositivos que puedan soportar estos nuevos codecs. Las compañías que desarrollan los dispositivos finales (Smart TV's, smartphones, decodificadores, consolas de video juegos, tablets, etc), han empezado a adaptarlos para que los usuarios finales puedan disfrutar de estos nuevos servicios de video en UHD.

En la Tabla 2 podemos encontrar un resumen de los tipos de servicios que ofrecen video y que codecs utilizados para poder realizar la transmisión de los mismos según según la calidad.

SERVICIO	TASA DE BIT	CODECS UTILIZADOS
Transmisión de TV Digital	2 - 5 Mbps 5 - 15 Mbps para HD 18 - 34 Mbps UHD	MPEG-2 H.264/AVC/MPEG-4 HEVC/H.265
DVD, HD-DVD, Blu-Ray, HD Blu-Ray	2 - 20 Mbps	MPEG2 H.264/AVC, VC-1
Video streaming sobre Internet	0.5 - 2 Mbps 5 - 8 Mbps para HD 10 - 24 Mbps para UHD	H.263, H.264/AVC, MPEG-4, VC-1, VP8, VP9, HEVC/H.265
Video conferencia, video llamadas	20 - 400 Kbps 700 - 1000 Kbps para HD	H.261, H.263 H.264/AVC
Video sobre redes móviles 3G y 4 G	200 - 700 Kbps para SD 1.5 - 7 Mbps para HD	H.263, MPEG-4, H.264/ AVC, VC-1

Tabla 2. Servicios de video con sus respectivos codecs<sup>12</sup>

### 3.1.3. Codecs de audio

De igual forma, los codecs de audio tienen una gran importancia en el servicio de video streaming ya que no es lo mismo que el usuario reproduzca un video de una calidad alta y que el audio sea pobre. A continuación veremos los codecs de audio más utilizados.

**MPEG Audio Layer III o MP3**, es un codec con pérdida usado para la compresión de audio dentro de un archivo. Es comúnmente usado en dispositivos de reproducción de audio como los ordenadores, reproductores que llevan su mismo nombre, equipos de música, etc. No es el más eficiente pero es uno de los que más soporte tiene en los dispositivos, además puede comprimir con una tasa menor o mayor de bits por segundo según la calidad deseada. Tiene 1 canal de audio mono o 2 canales estéreo.

**AC3**, este codec presentaba una nueva característica que MP3 no lo hacía, es la del sonido envolvente, se lo conoce también como Dolby Digital. Ahora cuenta con un

<sup>12</sup> Datos tomados de diferente documentación en Internet y en las referencias.

canal de audio mono, dos estéreo, cuatro de audio cuadrafónico y hasta canales 5.1 (audio surround) que lo incluían muchas películas y videos cortos, es por ello que se empezó a popularizar.

**AAC** es similar al MP3 aunque es un poco más eficiente, es decir, podemos tener un archivo de igual calidad que MP3 pero que ocupa menos espacio de almacenamiento. Apple lo hizo muy popular con iTunes y iPod ya que soporta funciones DRM. Gracias a su alto rendimiento y calidad, es muy utilizado para la compresión de audio para transmisión por Internet, conexiones inalámbricas, y radio difusión digital<sup>13</sup>. MPEG-4 cuenta con AAC para la codificación del audio. Posteriormente salieron versiones mejoradas como AAC-LC (ACC de baja complejidad) y HE-ACC (ACC de alta eficiencia).

**WMA** fue desarrollado por Microsoft como alternativa a MP3, introdujo mejoras como el soporte opcional de DRM, además ofrece sonido envolvente como AAC. Como un punto en contra podemos decir que no tenía tanto soporte como MP3 o ACC dado que Microsoft tenía los derechos del codec.

**Ogg Vorbis**, utiliza el contenedor OGG, es de código abierto y es una alternativa a MP3, WMA, ACC. Es más eficiente y provee mejor calidad de sonido que MP3. Tiene muchos más canales que MP3, por lo tanto ofrece sonido envolvente. Es utilizado en video juegos y aplicaciones como Spotify que lo utiliza para su servicio de reproducción de música streaming por Internet. En la Tabla 3 podemos observar la relación de los contenedores con los codecs de audio y video que soportan.

CONTENEDOR	CODECS DE AUDIO	CODECS DE VIDEO
MKV	Vorbis, MP3, LC-AAC, HE-AAC, WMA, AC3	MPEG-1, MPEG-2, MPEG-4/H.264, H.265, VC-1, VP8, VP9
MP4	MP3, LC-AAC, HE-AAC, AC3	MPEG-1, MPEG-2, MPEG-4/H.264, VC-1, VP8, y posiblemente H.265.
FLV	MP3, LC-AAC, HE-ACC	VP6, MPEG-4/H.264/AVC
3GP	LC-AAC, HE-AAC	H.263, MPEG-4/H.264/AVC
MOV	MP3, WMA, HE-AAC, AAC, AC3	MPEG-1, MPEG-2, MPEG-4/H.264/AVC, VC-1
WebM	Vorbis	VP6, VP7, VP8, VP9
OGG	Vorbis	MPEG-1, MPEG-2, MPEG-4/H.264/AVC, VC-1

Tabla 3. Codecs soportados por los contenedores de video<sup>14</sup>

<sup>13</sup><http://www.extremetech.com/computing/56014-audio-codec-quality-shootout/2>

<sup>14</sup> Solamente se han tenido en cuenta los codecs que hemos explicado. Pueden haber más codecs que soporten estos contenedores y que no estén incluidos.  
[https://en.wikipedia.org/wiki/Comparison\\_of\\_video\\_container\\_formats](https://en.wikipedia.org/wiki/Comparison_of_video_container_formats)

Hasta ahora hemos visto la forma de transformar el contenido, en este caso el video, a los diferentes formatos que permitan realizar el streaming. A continuación vamos a ver los protocolos de streaming utilizados para la distribución del contenido a los usuarios finales.

### 3.2. Protocolos de video streaming

Actualmente contamos con una gran cantidad de técnicas y protocolos que permiten realizar la distribución del video streaming a través de Internet. Entre ellos podemos destacar al protocolo HTTP que utiliza el protocolo de transporte Transmisión Control Protocol (TCP) para asegurar que la transferencia de datos sea fiable. Por otro lado, tenemos al protocolo Real-Time Streaming Protocol (RTSP) que utiliza como protocolo de transporte al protocolo Real-Time Transport Protocol (RTP) y al protocolo User Datagram Protocol (UDP). UDP es un protocolo no orientado a conexión por lo que no se garantiza la entrega de los datos, en otras palabras es no fiable. Al inicio, UDP fue usado por los proveedores de IPTV para la entrega del contenido a redes cercanas en donde la QoS podía ser garantizada; también a menudo se utiliza en video conferencias sobre Internet.[16] Como punto a favor podemos decir que tiene una latencia menor que si se utiliza HTTP sobre TCP, debido a que no solicita retransmisión de datagramas perdidos. Como punto negativo, el protocolo RTSP puede ser bloqueado por los Firewalls; es por ello que su uso no ha crecido comparado con HTTP. HTTP es un protocolo ampliamente usado por los navegadores Web para acceder a la red, por lo que tiene menos problemas con Firewalls, y routers Networks Address Translation (NAT's). [3]

A los protocolos de utilizados para el video streaming los podemos clasificar en dos grupos: Los que utilizan protocolos de transporte que son fiables como HTTP/TCP y los protocolos que utilizan al protocolo de transporte no fiable UDP. (Tabla 4).

Protocolos basados en HTTP	Protocolos no basados en HTTP
HTTP Live Streaming de Apple (HLS)	Real-Time Protocol (RTP)
Microsoft Smooth Streaming (MSS)	Real-Time Streaming Protocol (RTSP)
HTTP Dynamic Streaming de Adobe (HDS)	Real-Time Messaging Protocol (RTMP)
Dynamic Adaptive Streaming over HTTP (DASH)	File Delivery over Unidirectional Transport Protocol (FLUTE)

Tabla 4. Protocolos de Streaming

#### 3.2.1. Protocolos de video streaming no basados en HTTP

El que un sistema no use al protocolo HTTP en el envío del video o contenido no significa que no se pueda brindar un servicio fiable. Protocolos de streaming como RSTP pueden utilizar como protocolo de transporte tanto a TCP como UDP. A conti-

nuación vamos a analizar algunos protocolos utilizados para el streaming de video que no se basan en HTTP.

### 3.2.1.1. Real-Time Transport Protocol (RTP)

RTP es un protocolo de transporte para aplicaciones en tiempo real (audio, video, datos); trabaja conjuntamente con UDP. Al ser UDP no proporciona QoS. Está definido por el RFC 3550 de la Internet Engineering Task Force (IETF) y fue diseñado principalmente para la transferencia de datos en tiempo real. RTP multiplexa los paquetes de varios flujos de datos y los envía a través de la red en un único flujo de paquetes UDP; cuando usuario los recibe, el reproductor irá re-ensamblando todos los paquetes UDP que le van llegando y los irá poniendo en el buffer hasta que esté lo suficientemente lleno para que permita la reproducción del contenido. Los paquetes perdidos en el camino son ignorados, manteniendo la sincronización entre el audio y el video durante la reproducción. Una sesión RTP puede contener varios streams (audio y video) que al llegar al receptor son combinados.[3]

RTP es usado conjuntamente con el protocolo Real-Time Transport Control Protocol (RTCP). Mientras RTP encapsula los streams de video y audio en los paquetes UDP que son transmitidos, RTCP realiza las tareas de monitorización de la comunicación. Con ello se puede mantener un flujo de datos multimedia adecuado entre el cliente y el servidor mediante el ajuste de la tasa de datos. Los mensajes RTCP son enviados de forma periódica entre el cliente y el servidor. Además, no transportan datos multimedia, solamente transmiten información del estado de la sesión entre los usuarios y el servidor con lo que permite conocer el estado de la conexión en cada instante.

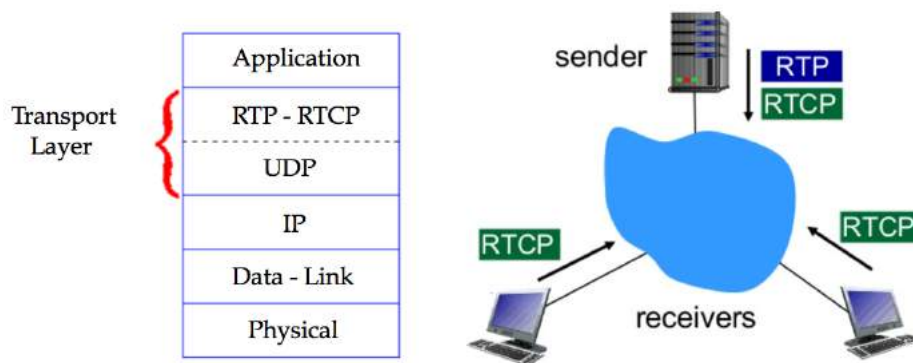


Figura 13. Comunicación con los protocolos RTP y RTCP <sup>15</sup>

En la cabecera de los paquetes RTP tiene campos en los que se podemos encontrar el tipo de codificación que llevan los datos multimedia (PCM, H.261, etc), el número de secuencia de los paquetes RTP enviados, el instante de muestreo del primer byte en ese paquete RTP y por último tenemos un campo que permite la identificación de la fuente del stream de datos RTP. Cada sesión RTP tendrá un número de identificación diferente.

<sup>15</sup> <http://www.slideshare.net/SavvasDimopoulos/chapter-7-v60>

RTP es usado en servicios como las video llamadas, aplicaciones de video conferencia, Voice over IP (VoIP), etc. Para la reproducción del contenido en los navegadores Web se necesita un Plug-in que permita el streaming RTP/UDP, aunque actualmente contamos con algunos reproductores como Flash que soporta RTP. Los reproductores como RealPlayer, Windows Media Player, y QuickTime Player también soportan RTP/UDP <sup>16</sup>.

### *3.2.1.2. Real-Time Streaming Protocol (RTSP)*

El protocolo RTSP está definido en el RFC 2326 de la IETF. Por defecto funciona en el puerto 554 y trabaja a nivel de aplicación; su función es establecer y controlar las sesiones multimedia entre servidor y el cliente con el objetivo de que los datos se entreguen correctamente ya que la transmisión de los flujos de contenido es muy sensible a la sincronización temporal.

RTSP es un protocolo que no depende del protocolo de transporte que se utilice para el envío de los flujos de datos; puede usar tanto UDP como TCP. En la mayoría de casos RTSP utiliza a TCP como protocolo de transporte para realizar el control de las sesiones y al protocolo RTP sobre UDP para la transmisión del contenido multimedia. RTP puede funcionar tanto con UDP como con otros protocolos de transporte como TCP, pero no es tan común. [21], [22]

Mediante el envío de los mensaje RTSP, el cliente puede controlar los flujos de datos mediante varios métodos, por ejemplo, el cliente puede pausar la reproducción, posicionar la reproducción en un punto determinado, adelantado rápido y retroceso de la reproducción; funciones similares a las de un reproductor de DVD.

Entre las característica más importantes de RTSP es que tiene la capacidad multi-servidor, es decir, el flujo perteneciente a una misma presentación pueden residir en diferentes servidores. Es extensible ya que se pueden añadir nuevos parámetros y métodos a RTSP. Entre los métodos usados por RTSP tenemos:

- ✓ **OPTIONS:** El cliente o el servidor informan al otro las opciones que el puede aceptar.
- ✓ **DESCRIBE:** El cliente obtiene la descripción de la presentación del contenido multimedia.
- ✓ **ANNOUNCE:** Se envía entre el cliente y servidor o viceversa. Se anuncia la publicación o actualización de la descripción de una presentación.
- ✓ **SETUP:** Cliente pregunta al servidor donde se encuentran los recursos multimedia para iniciar sesión RTSP.

---

<sup>16</sup> <http://www.garymcgath.com/streamingprotocols.html>

- ✓ **PLAY:** El cliente pide al servidor que inicie el envío de datos que fueron asignados por SETUP.
- ✓ **PAUSE:** El cliente temporalmente detiene el envío de datos sin liberar los recursos del servidor.
- ✓ **TEARDOWN:** El cliente solicita detener envío de datos y liberar los recursos del servidor.
- ✓ **GET\_PARAMETER:** Obtiene el valor de un parámetro, presentación o un flujo especificado en el Uniform Resource Identifier (URI).
- ✓ **SET\_PARAMETER** Coloca un valor de un parámetro para una presentación o flujo especificado por el URI.
- ✓ **REDIRECT** El servidor informa al cliente que debe conectarse a otro servidor.
- ✓ **RECORD:** El cliente empieza a grabar datos de acuerdo a la descripción de la presentación.

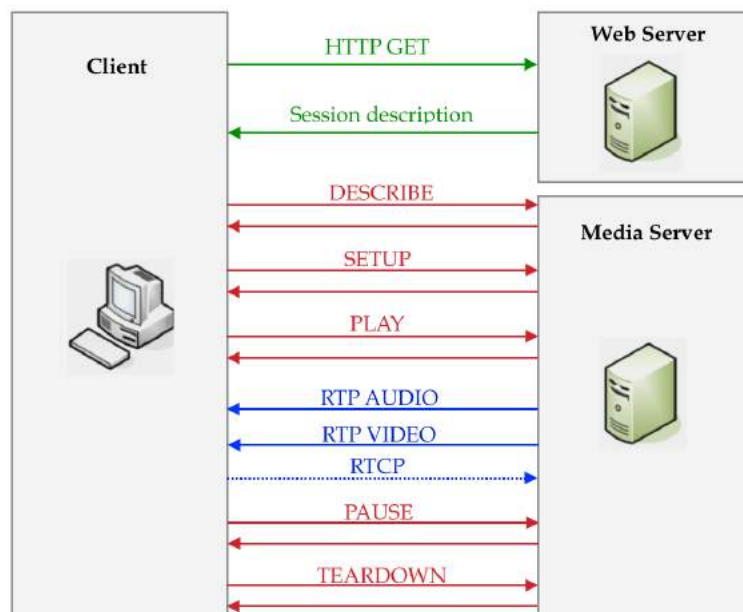


Figura 14. Sesión RTSP

RTSP ayuda a proveer el mismo servicio para el streaming de audio y video que HTTP los hace para el texto y gráficos. Fue diseñado con el propósito de que tenga el mismo sintaxis y operaciones. Cada presentación y streaming multimedia en RTSP es identificado por una Uniform Resource Locator (URL) RTSP. La presentación y del contenido multimedia están definidas en el archivo de descripción de la presentación, en el cual podemos encontrar el codec utilizado en la codificación del contenido multimedia, lenguaje, las URL's RTSP, dirección del destino, puerto, etc. El cliente puede

también obtener el archivo de descripción de la presentación usando HTTP, u otros mecanismos. [22] (Figura 14)

Sin embargo, RTSP se diferencia de HTTP en muchos aspectos:

- ✓ HTTP es un protocolo sin estado, un servidor RTSP tiene que mantener el estado de las sesiones para controlar que las solicitudes RTSP se corresponden con un determinado flujo de datos.
- ✓ HTTP es un protocolo asimétrico donde el cliente envía solicitudes y el servidor responde. En RTSP, ambos, servidor y cliente pueden enviar solicitudes. Por ejemplo, el servidor puede hacer una solicitud para establecer los parámetros de reproducción de un stream de datos. [22]

Como desventajas de RTSP tenemos que, tal y como ya habíamos dicho antes, RTSP al utilizar RTP sobre UDP es propenso a ser bloqueado por los Firewalls; además, al no ser un protocolo basado en HTTP, no utiliza la infraestructura ya existente como las caches HTTP que ayudan mucho a reducir la carga en los servidores. [3], [22]

### **3.2.1.3. Real-Time Messaging Protocol (RTMP)**

RTMP es un protocolo propietario, desarrollado por Adobe y publicado en el 2009. Está diseñado principalmente para la transmisión de flujos de datos (streams) de video, audio, y datos entre las plataformas Adobe Flash. Es un protocolo de baja latencia utilizado por Flash Player para reproducir el contenido bajo demanda y contenido en directo. Al disminuir la latencia en la comunicación, favorece la entrega fluida del contenido del audio y video. Tiene asignado por defecto el puerto 1935. Además, este protocolo trabaja sobre TCP, con lo que el control de congestión es más efectivo que RTP que trabaja con UDP.

RTMP utiliza la técnica de Dynamic Streaming o flujo dinámico, en donde la calidad de video se ajusta automáticamente ante los cambios que se produzcan en el ancho de banda. El ajuste de la calidad del video es posible gracias a que este protocolo divide el contenido a transmitir en pequeños bloques de información de igual tamaño llamados *chunks*. Cada *chunk* es almacenado en el servidor con diferentes características (bitrate, resolución, etc), con ello se consigue que RTMP transmita los chunks según las capacidades que tenga el cliente. La calidad del video se irá ajustando a las condiciones de la red, por ejemplo, a las variaciones del ancho de banda.

Los navegadores Web necesitarán tener instalado el Plug-in de Flash Media Player para la reproducción del contenido en los ordenadores y dispositivos multimedia como smartphones, tablets, etc. Cuando un cliente solicita el contenido multimedia, por ejemplo un video, al hacer clic en PLAY en el reproductor, este enviará un mensaje al servidor Flash, estableciendo así la conexión RTMP entre ambos, cliente y servidor. La conexión se realiza mediante la técnica "Three-way handshake" de TCP. Una vez

establecida la conexión, el reproductor ha obtenido un archivo Manifest que tiene una extensión (.smil) y el cual contiene las características del contenido, como el bitrate, nivel de calidad del video, idioma, URLs, etc. El reproductor solicitará los chunks de un contenido específico. Cuando el servidor recibe la solicitud, enviará directamente el contenido sobre la conexión RTMP. Este mismo contenido puede ser enviado a múltiples cliente que lo hayan solicitado, siempre y cuando el ancho de banda sea lo suficientemente bueno en ambos lados de la conexión para poder realizar una entrega fluida de los datos. (Figura 15)

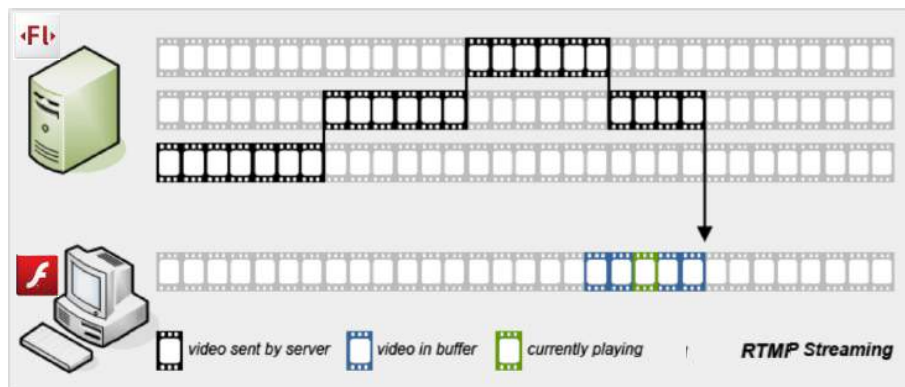


Figura 15 Comunicación mediante RTMP [16]

RTMP soporta los formatos de video MP4, FLV y audio codificado en AAC y MP3.

Como desventaja, tenemos que el puerto asignado por defecto (1935) puede estar bloqueado por los Firewalls, con lo que no se podrá establecer de la sesión RTMP. Es por ello que actualmente se cuenta con algunas variantes del protocolo RTMP que solucionan este problema y que añaden seguridad. Estas variantes son: [21],[16]

- **RTMP Tunneled (RTMPT):** Es el protocolo RTMP encapsulado en una solicitud HTTP, esto nos ayudara a atravesar Firewalls ya que el puerto 80 que utiliza HTTP no lo suelen bloquear los Firewalls. Al utilizar túneles provoca que aumente la latencia en la comunicación.
- **RTMP Secure (RTMPS):** Es el protocolo RTMP sobre una conexión TLS/SSL. Es otra forma de utilizar túneles en Redes privadas virtuales (VPNs). Esta vez el protocolo RTMP va encapsulado en HTTPS que utiliza el puerto 443.
- **RTMP Encrypted (RTMPE):** Es el protocolo RTMP cifrado con mecanismos de seguridad propios de Adobe.
- **RTMPT Encrypted (RTMPTE):** Es el protocolo RTMP utilizando túneles y aplicándole cifrado de Adobe.

Hulu, una plataforma que ofrece servicios SVoD y AVoD en EEUU y Canadá utiliza este protocolo (y su variante RTMPT) para ofrecer los servicio a los clientes que utilizan ordenadores de escritorio. [7]



#### *3.2.1.4. File Delivery over Unidirectional Transport Protocol (FLUTE)*

File Delivery over Unidirectional Transport (FLUTE) es un protocolo, que como su nombre lo indica, es utilizado para la entrega de archivos sobre enlaces unidireccionales usando UDP. Está definido por el RFC 3926 y RFC 6726. Como ya habíamos dicho, FLUTE fue diseñado para la transmisión sobre Internet usando UDP/IP ya que la transmisión es unidireccional (TCP necesita una conexión bidireccional por lo que no se utiliza con FLUTE). Además, FLUTE es independiente de la versión del protocolo IP que se utilice, puede ser IPv4 o IPv6 ya que no tiene ninguna especificación acerca de la versión del protocolo IP a utilizar.

Actualmente su uso es muy popular, especialmente en redes que prestan servicios de video streaming en entornos con muchos de usuarios. Puede ser usado para una comunicación multicast y unicast, pero su principal aplicación es para realizar la entrega de archivos mediante una comunicación multicast, es decir, desde un único transmisor hacia múltiples receptores simultáneamente. La entrega de este contenido se realiza mediante sesiones, en las cuales se pueden entregar los archivos, tanto grandes como pequeños, a muchos usuarios. Por ejemplo, el protocolo puede ser usado para entregar actualizaciones de software de gran tamaño a muchos usuarios. FLUTE no fue pensado originalmente para el video streaming, pero esta tecnología puede dar soporte a las aplicaciones de video streaming si fuera necesario.

FLUTE puede trabajar en muchos tipos de redes, incluyendo LANs, WANs, Intranets, redes inalámbricas, redes satelitales, y obviamente Internet<sup>17</sup>. Aunque principalmente se aplica en la transferencia fiable en servicios multimedia en directo sobre redes inalámbricas Long Term Evolution (LTE), ya que cada vez este servicio es más demandado por los usuarios.

La principal característica de este protocolo es que proporciona fiabilidad en la transmisión, muy necesaria en redes multicast que no garantizan la entrega fiable de los paquetes. Además, ofrece una gran escalabilidad, gestión y control de congestión. Los tres principales mecanismos usados por FLUTE para proporcionar fiabilidad son: Application Layered - Forward Error Correction (AL-FEC) el cual añade redundancia y corrección de errores; las retransmisiones se realizan por medio de un carrusel de datos que recibe los paquetes perdidos previamente; y un archivo de reparación de sesiones para solicitar ciertos paquetes que no han sido recibidos.

FLUTE trabaja sobre Asynchronous Layered Coding (ALC), el cual es un protocolo apropiado para los servicios de distribución de contenido multicast de forma unidireccional. A su vez, ALC esta construido sobre 3 bloques de construcción: Layered Coding Transport (LCT), un bloque de control de congestión (CC) y un bloque FEC. La transmisión en FLUTE está basada en sesiones de entrega, las cuales están identificadas

---

<sup>17</sup> <https://tools.ietf.org/html/rfc6726>

por la dirección IP destino y por un identificador llamado Transport Session Identifier (TSI). Además, cada sesión contiene uno o varios canales ALC/LTC, por medio de los cuales se realiza la entrega de los archivos. Cada canal envía el contenido multimedia con una determinada tasa de transmisión a un puerto dado.

Los archivos transmitidos son identificados por una localización del contenido y por un identificador de objeto llamado Transport Object Identifier (TOI). Las características principales, de los archivos que son enviados en una sesión, están descritas en un archivo Extensible Markup Lenguaje (XML) que se encuentra dentro de un mecanismo utilizado por FLUTE llamado File Delivery Table (FDT). La FDT es enviada junto con los archivos multimedia a través de los paquetes FLUTE llamados instancias FDT, que tienen el TOI igual a 0.

Cada archivo o objeto es dividido en bloques en el origen. A su vez, cada bloque origen es dividido en símbolos codificados. Un paquete FLUTE puede contener al menos un símbolo codificado. Hay dos tipos de codificación de símbolos: Símbolos fuente y Símbolos de paridad. Los primeros conforman los datos originales del contenido, mientras que los símbolos de paridad son creados de una combinación de símbolos fuente (y otros símbolos de paridad) a través de codificación FEC, para proporcionar técnicas de corrección de errores. Raptor y RaptorQ son de los códigos para la corrección de errores que utiliza FLUTE. En la Figura 16 podemos ver un ejemplo de la comunicación entre un servidor y un cliente a través de un canal Multimedia Broadcast Multicast Service (MBMS) que utilizan las redes LTE.

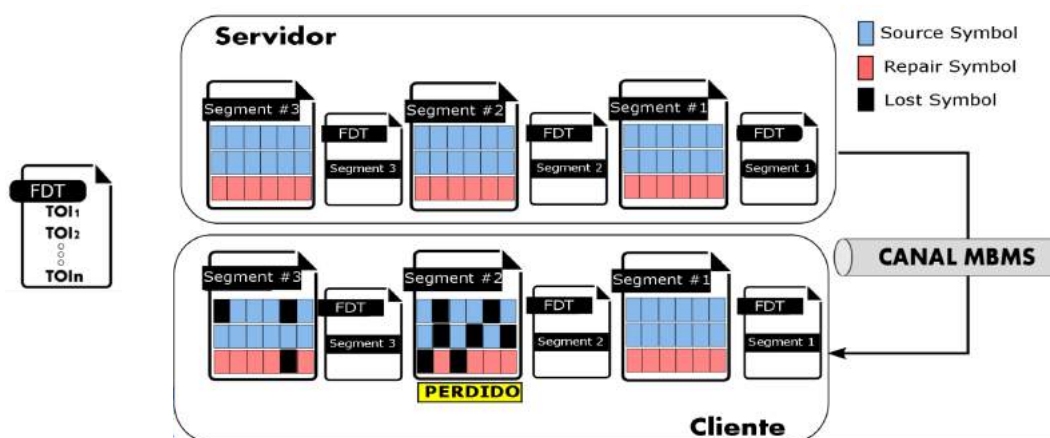


Figura 16. Comunicación FLUTE en un canal MBMS<sup>18</sup>

### 3.2.2. Protocolos de video streaming basado en el protocolo HTTP

HTTP es un protocolo basado en el método de petición/respuesta sobre el protocolo de transporte TCP; es un protocolo sin estado, es decir, el servidor no mantiene información acerca de las peticiones que un cliente ha realizado. Hoy en día conviven dos versiones, la HTTP/1.0 y la HTTP/1.1, pero a partir del Mayo del 2015 la IETF

<sup>18</sup> Fuente: Diapositivas del Master MUIRST, asignatura ASAI

propuso el nuevo estándar HTTP/2<sup>19</sup> el cual es compatible con las dos versiones anteriores. Esta nueva versión está basada en Speedy (SPDY), un protocolo desarrollado por Google. Entre las mejoras que presenta esta nueva versión de HTTP tenemos que la información viaja en formato binario, antes lo hacía de forma textual. Además, mediante una única conexión TCP, se puede multiplexar varias peticiones HTTP con lo que se mejora el mecanismo de descarga de contenido y se disminuye la latencia, por ejemplo, al cargar una página Web en el navegador. En la Figura 17 podemos ver la diferencia en latencia entre la versión HTTP/1.1 y HTTP/2.

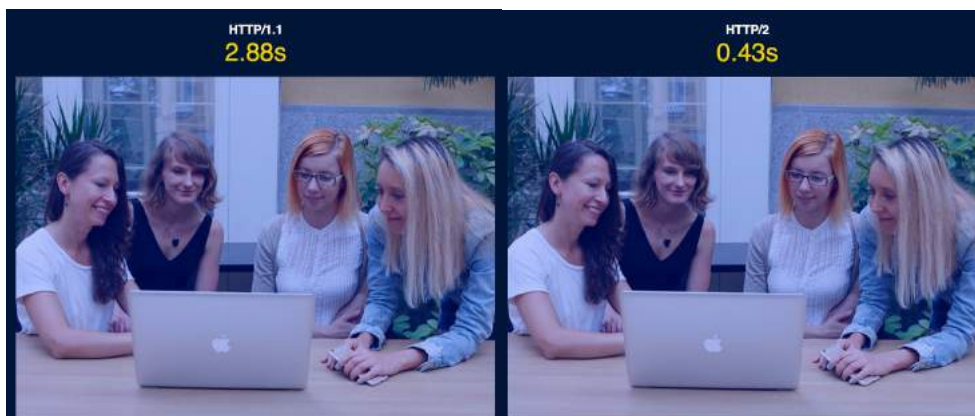


Figura 17. Comparación del protocolo HTTP/1.1 y HTTP/2<sup>20</sup>

HTTP tiene la ventaja de que no presenta problemas a la hora de atravesar routers NAT ni Firewalls, ya que HTTP al ser el protocolo usado por los navegadores Web es poco probable que un administrador de red bloquee el puerto que utiliza (80 en HTTP o 443 en HTTPS). También, cuenta con la ventaja que no necesita caches especiales, un archivo con contenido multimedia es tratado igual que cualquier otro archivo para una cache Web.

HTTP presenta otra ventaja sobre los otros protocolos a la hora de distribuir el contenido multimedia. Es utilizado por una técnica llamada Adaptive Bitrate Streaming (ABR), la cual es utilizada para optimizar la entrega del contenido multimedia a los usuarios finales. Todo esto ha colocado a HTTP como el protocolo preferido a la hora de transmitir de video streaming sobre Internet. Antes de la aparición de ABR, los proveedores de contenido transmitían video streaming con una tasa de bits constante, sin importar si la red estaba o no congestionada. En caso de que haya congestión en la red; si usuario solicitaba la reproducción de un video, el ancho de banda no permitía descargar suficientes chunks del video lo que ocasionaba que el buffer del cliente no esté lo suficientemente lleno para que la reproducción del video no se interrumpa; cuando la reproducción se interrumpía, el usuario tenía que esperar hasta que se descarguen suficientes chunks que permitan continuar con la reproducción. Figura 18.

<sup>19</sup> <https://tools.ietf.org/html/rfc7540>

<sup>20</sup> <http://www.http2demo.io>



Figura 18. Buffering de un video <sup>21</sup>

ABR fue una técnica desarrollada para mitigar este problema. Ahora los proveedores de contenidos codifican un mismo video con múltiples bitrates y calidades. A su vez, cada nuevo video (con su respectiva calidad y bitrate) es dividido en chunks de video que serán los que se enviarán a los clientes cuando lo soliciten. La longitud de los chunks varía entre 2 y 10 segundos de duración. Cuando un cliente solicita un video, el proveedor le envía un archivo llamado Manifest o Media Presentation Description (MPD) que contiene información acerca de las características disponibles del video (Bitrates, calidad de imagen, URL de la CDN donde se encuentra el contenido, etc), es entonces cuando el cliente selecciona las características que más se adecuan a capacidades de su equipo (ancho de banda disponible, codecs que soporta, memoria disponible, etc).

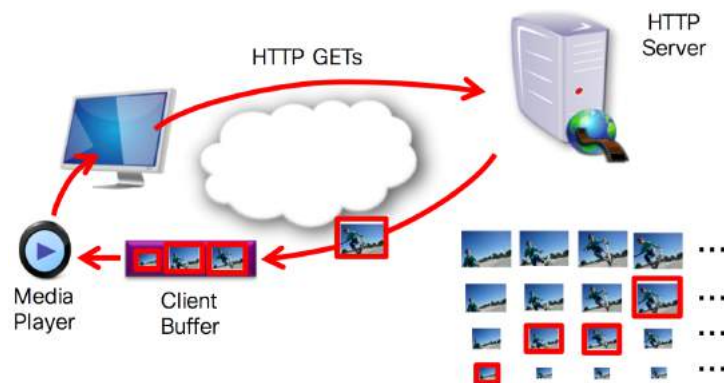


Figura 19. Transmisión de video utilizando la técnica ABR [8]

ABR al ser sin estado, cada solicitud enviada por un cliente para la descargar un chunk de video siempre será tratada como una nueva petición, por lo que, cuando el cliente solicite un nuevo chunk de video, podrá ir adaptando la descarga de los chunks de acuerdo a las características de su equipo que dispone en ese instante. Es

<sup>21</sup> <http://www.news18.com/news/tech/vp9-how-google-is-bringing-us-closer-to-high-quality-buffer-free-videos-980106.html>

decir, si un cliente dispone de poco ancho de banda, descargará contenido de menor calidad, menor bitrate; cuando disponga de más ancho de banda podrá descargar contenido de mejor calidad. Con ello se consigue que la reproducción no se interrumpa y que el cliente no tenga que esperar a que se llene el buffer de su equipo para continuar con la reproducción. Figura 19.

Actualmente, compañías como Adobe Systems, Apple, Microsoft están desarrollando y utilizando sus propias implementaciones de ABR. A continuación vamos a hablar de algunas de ellas.

### 3.2.2.1. HTTP Live Streaming (HLS)

HLS es un protocolo de streaming basado en HTTP desarrollado por Apple Inc. en el 2009. Su aplicación está enfocada a la comunicación de los dispositivos con sistemas operativos iOS y OS X, es decir, en sus propios dispositivos; principalmente en aplicaciones como QuickTime, FaceTime, Safari, Apple TV entre otras. Este protocolo está diseñado para permitir la transmisión de video sobre el puerto 80, dando mayor flexibilidad a los dispositivos móviles, los cuales pueden usar varias redes y que no estarían sujetos a puertos que podrían estar bloqueados por un Firewall.

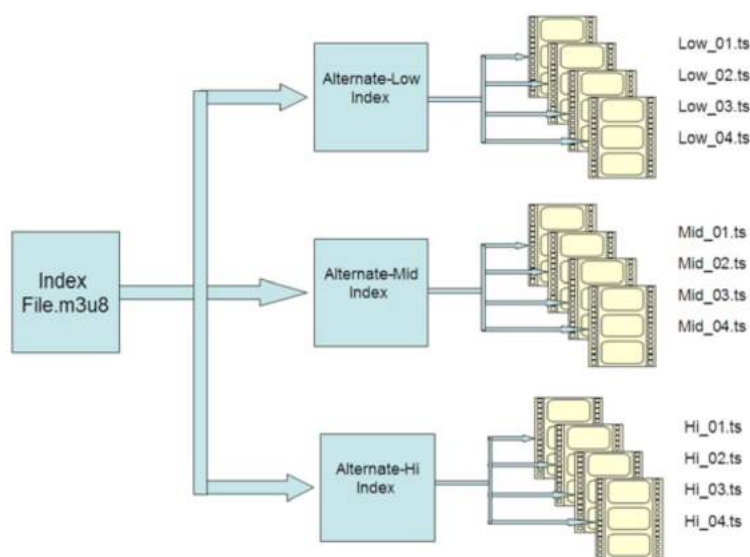


Figura 20. Estructura del archivo Manifest de HLS

HLS utiliza, de igual forma, chunks de video con diferentes bitrates y que llevan una extensión “.ts”. El servidor dispone de un archivo Manifest llamado MPEG versión 3.0 URL UTF-8 (M3U8), que contiene información acerca de los codecs, bitrate, y información de la URI de los chunks de video disponibles. Cuando un cliente quiere un contenido determinado, se conectará al servidor Web y descarga una copia del archivo M3U8 del video que ha solicitado. Luego, el reproductor irá solicitando (mediante HTTP) los chunks de video con el bitrate apropiado a las características de su equipo. Los chunks no necesitan ser descargados en ningún orden en particular, es el reproductor del cliente el responsable de la correcta reproducción del contenido. Durante la

reproducción del video o del contenido que se ha solicitado, el reproductor deberá ir actualizando periódicamente la copia del archivo M3U8 ya que el servidor Web puede realizar cambios en la información que contiene este archivo, incluso durante la reproducción. Dado que la conexión con el cliente es sin estado, el servidor no sabe que clientes requieren actualizaciones; entonces, son los clientes quienes deben ir actualizando su copia del archivo M3U8 para obtener cualquier actualización realizada en la misma. Esta característica de HLS permite ofrecer video streaming en directo o Live Streaming y además de contenido de VoD. En la Figura 20 podemos observar la estructura del archivo Manifest del HLS.

HLS es ampliamente soportado por servidores de streaming de varios fabricantes como Adobe, Microsoft, etc. Además, HLS utiliza el contenedor MPEG-TS con los codecs de video MPEG-4/H.264/AVC y los codecs de audio AAC, AC3 o MP3. Además, es soportado por varios tipos de DRMs, en especial los que tienen cifrado Advanced Encryption Standard (AES) de 128 bits.

#### *3.2.2.2. Microsoft Smooth Streaming (MSS)*

Smooth Streaming es un protocolo desarrollado por Microsoft en el año 2009. Está basado en el método de descarga progresiva sobre HTTP. Es una extensión de los servicios multimedia de Internet Information Services (IIS) de Microsoft. Del mismo modo, el contenido multimedia se codifica con distintos bitrates y niveles de calidad de video (baja, media y alta). Posteriormente, el contenido es dividido en pequeños fragmentos llamados chunks (generalmente de 2 segundos). Las descargas HTTP mediante los chunks, permite que el contenido multimedia sea más fácil y económico de almacenar en las caches Web de los servidores que proveen este contenido a los usuarios finales. MSS verifica que cada chunk de video ha llegado en el tiempo apropiado y que se ha reproducido con la calidad esperada. Si no se cumplen estos requerimientos, el próximo chunk entregado será de un nivel de calidad menor. Cuando las condiciones lo permitan, la calidad de los siguientes chunks podrán ser de mejor calidad.

Microsoft cuenta con un reproductor llamado Silverlight, que permite la reproducción del contenido que el cliente solicita. Silverlight se encarga de mantener y manejar la información indexada en el archivo Manifest XML, el cual contiene información acerca de los fragmentos "chunks" codificados. Además, Silverlight es el encargado de monitorizar los recursos de red y CPU disponibles en el equipo del cliente, lo hace cada 2 segundos, lo cual le permite hacer la solicitud de los próximos chunks de una calidad adecuada a las características del equipo y de la red.

Existen dos tipos de archivos Manifest en MSS, el primero que tiene extensión .ism y es el Manifest MSS del servidor Web, este archivo contiene una lista de los archivos con los múltiples bitrates disponibles y un identificador que apunta al Manifest del cliente; el segundo es el Manifest MSS del cliente, tiene una extensión .ismc y que contiene los

metadatos acerca del audio, video, subtítulos, etc. En la Figura 21 podemos observar un ejemplo de un escenario con MSS.

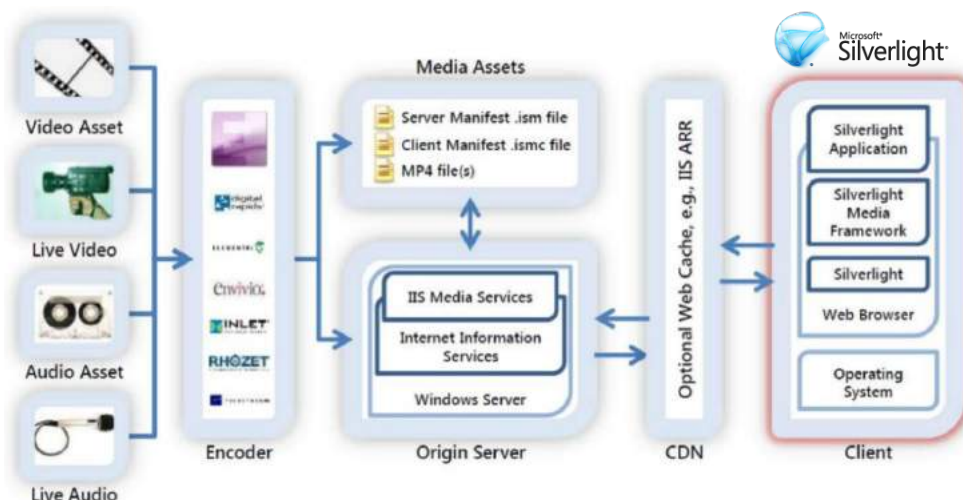


Figura 21. Escenario con Microsoft Smooth Streaming (MSS) [16]

SSM es utilizado tanto para Live Streaming como para servicios de VoD, utiliza el formato MP4, soporta los codecs de video VC-1 y H.264 (MPEG-4 AVC) y los codecs de audio WMA y AAC. Además, como DRM se usa a Microsoft PlayReady.

SSM es compatible con una serie de dispositivos: los navegadores Web con el Plug-in de Silverlight, Xbox, Windows Phone, decodificadores de TV como Roku, etc. También es compatible con dispositivos que tienen el sistema operativo de Apple como los iPhones y iPads.

### 3.2.2.3. HTTP Dynamic Streaming (HDS)

HTTP Dynamic Streaming es una tecnología desarrollada conjuntamente por Adobe y Akamai en el año 2010. El objetivo de HDS es la aplicación en los Servidores de Adobe Flash Media. Este protocolo esta basado en la técnica de streaming adaptativo, lo cual le permite manejar adecuadamente el streaming de video usando los protocolos HTTP y RTMP. Como vimos antes, RTMP es un protocolo de video streaming también desarrollado por Adobe pero que no utiliza HTTP para la entrega del contenido multimedia a los usuarios finales, por lo menos directamente. RTMP sobre HTTP lo puede hacer mediante túneles, tanto con HTTP como con HTTPS.

HDS combina las características de los método de descarga progresiva y las de streaming. Esto permite a los usuarios poder tener una reproducción instantánea y la capacidad de cambiar la reproducción en cualquier punto en el video.

Al igual que los protocolos anteriores (HLS, MSS), cuando se prepara un contenido multimedia para ser distribuido, este es fragmentado en chunks con diferentes bitrates y diferentes niveles de calidad, posteriormente son comprimidos en unos archivo que tiene un formato multimedia con extensión .F4F que utiliza Adobe. Cada archivo F4F

contiene un archivo Manifest (F4M) el cual describe las características del contenido que tiene disponible, el índice de los chunks de video (los chunks por lo general son de 1 a 4 segundo). El archivo Manifest F4M puede describir las características del contenido multimedia tanto para el protocolo HDS como para RTMP que también es desarrollado por Adobe <sup>22</sup>.

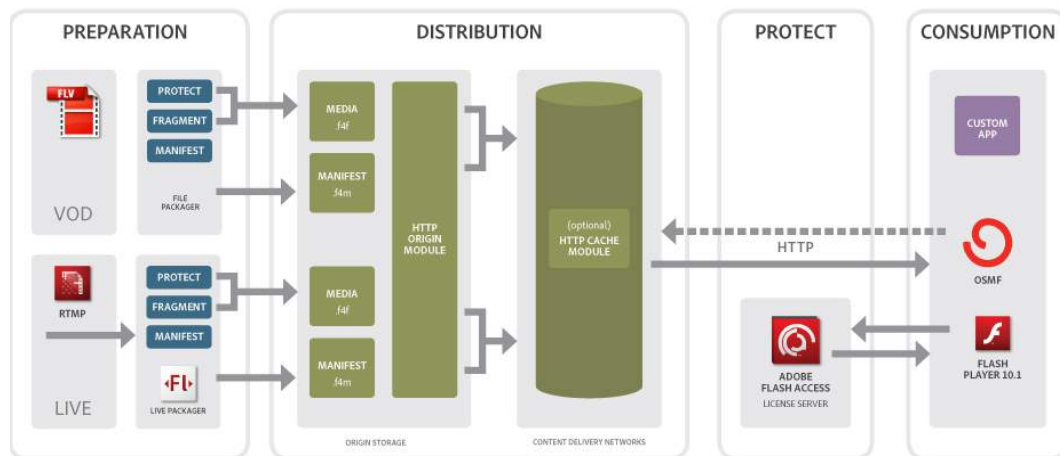


Figura 22. Escenario con HTTP Dynamic Streaming de Adobe (HDS) <sup>23</sup>

Para realizar la solicitud de un contenido, el cliente necesita tener instalado en su equipo el Plug-in de Adobe Flash Player u otra aplicación que soporte HDS de Adobe. Una vez que el cliente dispone del Plug-in ya puede solicitar el contenido al servidor que lo ofrece, un video por ejemplo. El reproductor solicitará el archivo Manifest (F4M) al Servidor, una vez que el servidor le responde, el cliente seleccionará las características y la localización de los chunks de video que más se ajusten a las características de la red y del equipo del cliente y que permitan la reproducción del mismo. En la Figura 22 podemos observar un escenario en el que se utiliza a HDS.

De igual forma que HLS, HDS utiliza como formato del contenedor de video MP4, soporta los codecs de video H.264 (MPEG-4/AVC), VP6 y como codecs de audio a AAC y MP3. Como DRM utiliza a Adobe Flash Access. En los servidores que almacenan el contenido o servidores origen requiere que sean un servidor Adobe Media o si son de otro fabricante, necesitan tener un Plug-in de Adobe instalado.

#### 3.2.2.4. Dynamic Adaptive Streaming over HTTP (DASH)

Dynamic Adaptive Streaming over HTTP (DASH) es una tecnología desarrollada por el comité MPEG, es conocida como MPEG-DASH y fue publicada en la ISO en el año 2012. El comité MPEG buscaba con DASH, el poder contar con un estándar común que resuelva la complejidad a la hora de realizar la entrega de contenido multimedia a

<sup>22</sup> <http://www.adobe.com/devnet/hds.html>

<sup>23</sup> <http://blog.denivip.ru/wp-content/uploads/2010/12/scheme.jpg>



diferentes dispositivos clientes que operaban bajo estándares propietarios como MSS, HLS, HDS.

Este nuevo estándar, puede ser utilizado para la transmisión bajo demanda (como VoD) o contenido de eventos en directo (Live Streaming). DASH también define un archivo Manifest llamado MPD, el cual describe las características del contenido multimedia almacenado en el servidor. El MPD es un documento XML que esta formado por periodos; cada periodo tiene un identificador, un tiempo de inicio y la duración del mismo. A su vez, cada periodo esta formado por uno o múltiples sets de adaptación (Adaptation Sets - AS); un AS proporciona la información acerca de uno o múltiples componentes multimedia y las varias alternativas de decodificación de las que dispone. Por ejemplo, un AS puede contener diferentes bitrates de un componente de video de un mismo contenido multimedia. Otro AS puede contener diferentes bitrates de la componente del audio de un mismo contenido multimedia. Cada AS incluyen múltiples representaciones del contenido. Una representación es la codificación de un mismo componentes multimedia, cada representación dentro de un AS varía en el bitrate, resolución, numero de canales, etc. Además, cada representación consta de uno o múltiples segmentos (chunks multimedia). Cada segmento o chunk tiene una URI para poder localizarlo dentro del servidor. Estos chunks son descargados por el cliente DASH mediante las peticiones HTTP GET. En la Figura 23 podemos observar la estructura del Manifest MPD de DASH.

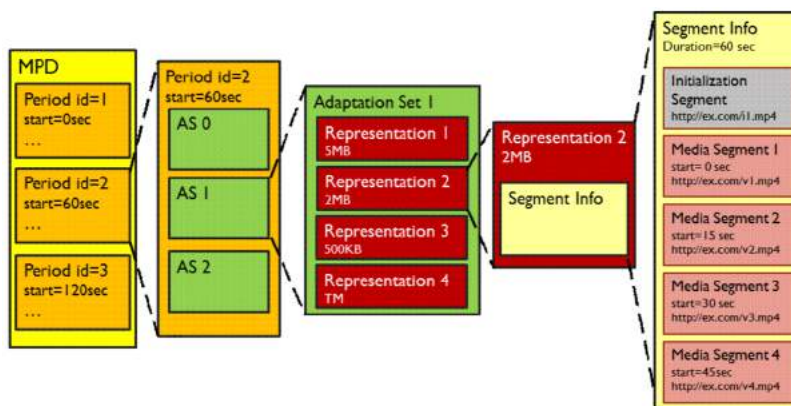


Figura 23. Estructura del Multimedia Presentation Description (MPD) [18]

Para la reproducción de contenido, el cliente DASH primero obtiene el archivo Manifest (MPD), el cual puede ser entregado mediante múltiples vías como HTTP, email, broadcast, etc. Una vez obtenido el MPD, cliente puede conocer las características del contenido multimedia. Utilizando esta información, el cliente selecciona las características que más se ajusten a su perfil, para luego comenzar la descarga y la reproducción del contenido; como ya lo hemos dicho, para descargar los chunks utilizará solicitudes HTTP GET. Además de la descarga y reproducción, el reproductor del cliente realiza la monitorización de la red y de las características del equipo del cliente. Dependiendo de estas medidas tomadas, el reproductor decidirá como adaptar

la descarga de los siguiente chunks. Esto permite que el reproductor pueda mantener la reproducción del contenido sin interrupciones y que el buffer tenga continuamente con datos.

Hay muchos beneficios al usar esta tecnología. Primero, el hecho de que la mayoría de compañías multimedia (Microsoft, Netflix, Google, Ericsson, Samsung, Adobe, entre otros) formen parte en su desarrollo directa o indirectamente[16], hace que este nuevo protocolo elimine los problemas técnicos en la entrega del contenido y la codificación del mismo, en definitiva, ayuda a combinar todas las tecnologías en una sola. DASH utiliza los contenedores de video con formatos MP4, MPEG-TS; además, soporta casi cualquier codec de audio y video. Respecto al DRM, DASH no especifica un método DRM en particular, pero da soporte a todas las técnicas DRM especificadas en ISO/IEC 23001-7 <sup>24</sup>. [16] En la Tabla 5, podemos observar un resumen de los protocolos que utilizan ABR y Dynamic Stream como RTMP.

Protocolo	Año	Contenedores	Manifest	Codecs Video	Codecs Audio
RTMP (Adobe)	2009	MP4, FLV	SMIL	H.263, H264	AAC, MP3
HLS (Apple)	2009	MPEG-TS	M3U8	H.263, H.264	AAC, AC3, MP3
MMS (Microsoft)	2009	MP4	ISM	VC-1, H.264	WMA, AAC
HDS (Adobe)	2010	MP4, FLV	F4M , SMIL	H.264, VP-6	ACC, MP3
MPEG-DASH (MPEG)	2012	MP4, MPEG-TS	MPD	Varios, HEVC	Varios

Tabla 5. Características de los protocolos de Streaming

A los protocolos de streaming se los puede clasificar también de acuerdo a la estrategia que utilizan al momento de distribuir el contenido entre los nodos de la red, incluyendo a los usuarios finales. Es así que los protocolos se pueden clasificar en dos categorías: Los que se basan en la estrategia PUSH y los que se basan en la estrategia PULL<sup>25</sup>

La **ESTRATEGIA PUSH** consiste en que una vez que el cliente y el servidor han establecido la conexión, el servidor transmite los paquetes hacia el cliente hasta que el mismo para o interrumpe la sesión.

<sup>24</sup> [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=68042](http://www.iso.org/iso/catalogue_detail.htm?csnumber=68042)

<sup>25</sup> En el presente trabajo se han clasificado los protocolos de video streaming en los que están basados en HTTP y los que no. Pero también podemos encontrarnos con esta clasificación de los protocolos, según la estrategia Push o Pull.

Con la estrategia Push, el servidor mantiene el estado de la sesión con el cliente/servidor y se mantiene escuchando el canal a la espera de cambios en el estado de la sesión. Si tiene nuevo contenido o ha habido cambios en la sesión, el servidor deberá enviar el nuevo contenido de inmediato a los clientes o servidores. El servidor envía el contenido con un bitrate que coincida con la media del bitrate del cliente. Esto permite que el buffer del cliente se mantenga estable todo el tiempo. En caso de cambios en las condiciones de la red y empeore el ancho de banda, el servidor pueden cambiar dinámicamente a bitrates más bajos que aseguren la reproducción del contenido en el cliente de la forma más eficiente. Este tipo de estrategia es ideal para servicios en tiempo real, como las video conferencias, mensajería instantánea, video en vivo, y para distribuir el contenido en las CDNs a sus servidores sustitutos, etc.

Uno de los protocolos que se basan en esta estrategia es el protocolo RTSP, el cual utiliza una serie de comandos para controlar las sesiones entre el servidor y el cliente, tal y como vimos en el punto 3.2.1.2. Los protocolos que se basan en esta estrategia, generalmente utiliza el protocolo RTP para la transmisión de la información; como ya vimos también, RTP por lo general usa el protocolo de transporte UDP y además utiliza al protocolo RTCP para el envío de reportes entre el servidor y el cliente. (Ver Figura 14)

El único inconveniente de esta estrategia, que como lo habíamos dicho, es el problema con los Firewalls y los routers NAT por lo que necesitan implementaciones especiales para hacer frente a estos problemas, por ejemplo, el uso del Session Transversal Utilities for NAT (STUN) o el protocolo Transversal Using Relay around NAT (TURN), etc.

La **ESTRATEGIA PULL** consiste en que ahora el cliente es quien solicita el contenido al servidor, es decir, el cliente es quien inicia la comunicación, unicast por lo general. El bitrate en el cual el cliente recibe el contenido dependerá del lado del cliente. El cliente es quien deberá ir ajustando dinámicamente el bitrate que puede soportar según las condiciones de la red. HTTP es un protocolo que utiliza esta estrategia en la entrega de contenido a través de Internet.

El método de descarga progresiva utiliza esta estrategia. En descarga progresiva, cuando un cliente realiza una solicitud HTTP al servidor e inicia la descarga lo más rápido posible. Cuando el cliente tiene el buffer lo suficientemente lleno, iniciará la reproducción mientras el contenido se sigue descargando y llenado el buffer.

Los protocolos que utilizan esta estrategia, también utilizan la técnica ABR y Dynamic Stream, en el caso de RTMP. Dividen el contenido en segmentos que están codificados con diferentes bitrates, resoluciones, etc. Además cuentan con un archivo manifest que lo envían al cliente, en el cual se le informa las características del contenido que ha solicitado. Esto permite al cliente ir solicitando los segmentos de

diferentes bitrates según las el ancho de banda disponible y las capacidades disponibles en su equipo. Aunque, las diferentes implementaciones que funcionan en base a HTTP difieren en las características de estos segmentos, el principio es el mismo.

La ventaja de esta estrategia, con respecto a la anterior, es que al ser usada por protocolos basados en HTTP, no tiene el problema cuando tiene que atravesar Firewalls y routes NAT. Además, también se benefician de la infraestructura que tiene HTTP en Internet, como por ejemplo las caches Web. Este tipo de estrategia es usada para proveer los servicios de streaming VoD, entre otros.

En la Tabla 6 podemos ver esta nueva clasificación de los protocolos según la estrategia que usan a la hora de la transmisión del contenido.

	Estrategia basada en PUSH	Estrategia basada en PULL
Comunicación	Unicast, Multicast	Unicast
Inicia la comunicación	Servidor	Cliente
Protocolos Streaming	RTSP	Basados en HTTP, RTMP
Protocolos de Transporte	UDP con RTP	TCP
Control y monitorización	RTCP	Propietario
Soporte de cache	No	Si para HTTP

Tabla 6. Protocolos de Streaming según su estrategia PUSH/PULL

## 4. Plataformas de Distribución de Contenidos

Recapitulando, hasta ahora hemos visto cómo una vez generado el video, ya sea en vivo o contenido ya almacenado, se ha aplicado la codificación al mismo mediante codecs de video, audio, y su respectivo contenedor de video. Con ello conseguimos que el tamaño del contenido codificado sea menor que del contenido original, esto permite la transmisión del mismo a través de la red de una forma más eficiente y que en este proceso se pierda lo más mínimo en cuanto a calidad de imagen y audio. Una vez el contenido se encuentra codificado o comprimido, hay que cargarlo en los servidores origen y otros servidores que permitan distribuirlo a través de la red para que llegue a los usuarios finales. En este proceso entran en juego los protocolos de streaming que hemos visto en el Capítulo 3.

En este capítulo analizaremos las diferentes Plataformas de Distribución de Contenido con las que contamos hoy en día, desde las CDNs hasta las soluciones basadas en la nube y soluciones híbridas entre P2P-CDN.

Las Plataformas de Distribución de Contenido son las responsables de que hoy en día podamos contar con servicios multimedia de alta calidad. Antes, cuando un usuario solicitaba el contenido a un servidor, este respondía sin importara la localización geográfica del mismo. Pero a medida de que el número de usuarios que solicitaban los contenidos al servidor incrementaban, el servidor no podía servirles porque agotaban sus recursos locales (memoria, CPU, etc). Además, la transmisión del contenido atravesaba toda la red hasta llegar desde el servidor al cliente, con lo que se congestionaba la red, aumentaba el retardo y la calidad del contenido se degradaba.

Ahora, gracias a las Plataformas de Distribución de Contenido, es posible disfrutar de contenido con calidad de hasta UHD en el caso de video; además ha disminuido la latencia en el servicio ya que ahora contamos con servidores distribuidos geográficamente que almacenan el mismo contenido que el servidor original. Un usuario puede acceder al servidor más conveniente, puede ser el más cercano físicamente o el que menos carga tenga. Ahora no hace falta atravesar toda la red para obtener un contenido.

### 4.1. Redes de Distribución de Contenidos (CDNs)

Una Red de Distribución de Contenidos o Content Delivery Content (CDN) es una red Overlay, es decir, una red superpuesta en la actual infraestructura de Internet y que se encuentra desplegada en múltiples centros de datos. Está formada por miles de servidores repartidos geográficamente alrededor del mundo, a estos servidores se los conoce como servidores sustitutos y se encuentran estratégicamente ubicados con el fin de mejorar la QoS y QoE, disminuir la latencia del servicio, aprovechar mejor el ancho de banda disponible, mejorar el rendimiento de la red, proveer seguridad a los sitios Web, etc.

La arquitectura básica de una CDN consta de uno o varios servidores principales, llamados también origen, los cuales almacenan todo el contenido y/o los metadatos de todo el contenido que ofrece a los clientes. Los servidores principales copian o replican este contenido en los servidores sustitutos, los cuales se encuentran distribuidos alrededor del mundo con el fin de acercar el contenido al usuario final. Por lo general mediante la estrategia PUSH. Los servidores sustitutos tienen la función de proporcionar el contenido a los usuarios que lo solicitan. Cuando el contenido no está localmente disponible, el servidor sustituto debe solicitar el mismo al servidor principal, lo hace mediante la estrategia PULL, una vez que el servidor principal responde, el servidor sustituto lo almacena, para alguna próxima solicitud del mismo por parte del mismo usuario u otro de la misma región. Otra característica de los servidores sustitutos es que pueden descargar contenido desde el servidor principal que puede ser de interés para los usuarios, esto lo realiza utilizando técnicas que requieren algoritmos estadísticos y de minería de datos para determinar que tipo de contenido será atractivo para un cliente dado. [14]

Una CDN sigue el principio del modelo cliente-servidor, pero en este caso, el cliente en lugar de acceder al contenido de servidor principal, accederá al contenido del servidor sustituto que se encuentre más cercano al mismo o que tenga menos carga y que proporcione mejor QoE al usuario final. No siempre el servidor más cercano geográficamente es el más óptimo ya que puede estar cargado con solicitudes de otros clientes. El servidor principal, actúa como “servidor” para los servidores sustitutos y estos a su vez realizan la misma función frente a los clientes o usuarios finales. Es por ello que el modelo en que se basa una red CDN corresponde a un modelo cliente - servidor de dos capas. En la Figura 24 podemos observar la arquitectura básica de una CDN, podemos ver el servidor principal que copia el contenido en los servidores sustitutos.

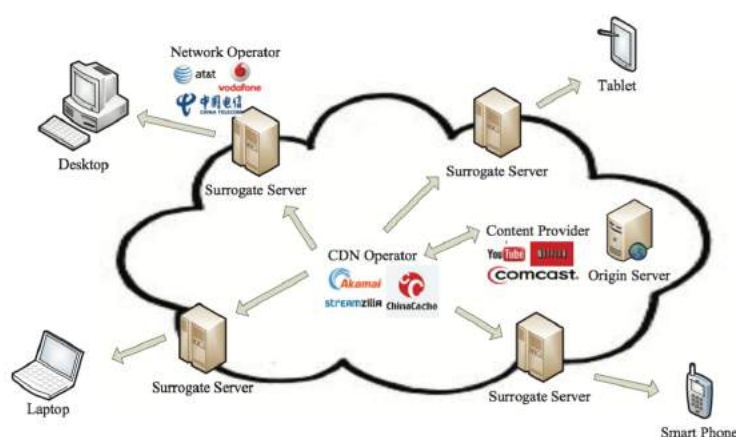


Figura 24. Arquitectura básica de una CDN

Actualmente tenemos dos modelos de CDNs. El primero se le denomina como *Enter Deep Into ISPs*. En este modelo, los proveedores de servicios CDN tienen su infraestructura dentro de los Points-of-Presence (PoPs) de los ISPs. La idea es estar cerca de

los usuarios finales, con ello se mejora el desempeño percibido por el usuario de la red en términos de latencia y rendimiento. Este modelo implica el tener desplegados una cantidad enorme de servidores en clústers alrededor del mundo. Akamai es un ejemplo de proveedor de servicio CDN a nivel mundial; Akamai tiene desplegados clústers de servidores (más de 170.000) en 102 países a través de más de 1.300 redes<sup>26</sup>. El segundo modelo se le denomina *Bring ISPs to Home*. En este modelo, los proveedores de servicios CDN cuentan con una red privada que se encuentra distribuida en unos cuantos centros de datos. Estos centros de datos se encuentran localizados en ubicaciones claves y están conectados entre ellos, mediante conexiones privadas de alta velocidad. Ahora en lugar de estar dentro de los PoPs de los ISPs, estas CDNs tienen cada centro de datos ubicado cerca de los PoPs de los ISPs más importantes, por ejemplo, a unos cuantos kilómetros de los PoPs de operadores como AT&T, Orange, Verizon, etc. Este modelo vende su presencia global a los clientes de este tipo de servicio de CDNs. Limelight es un ejemplo de proveedor de servicios CDN que tiene este modelo, cuenta con más de 18.000 servidores en más de 80 ubicaciones cerca de los PoPs de las redes de acceso. Además, está conectado a más de 900 ISPs a nivel mundial <sup>27</sup>.

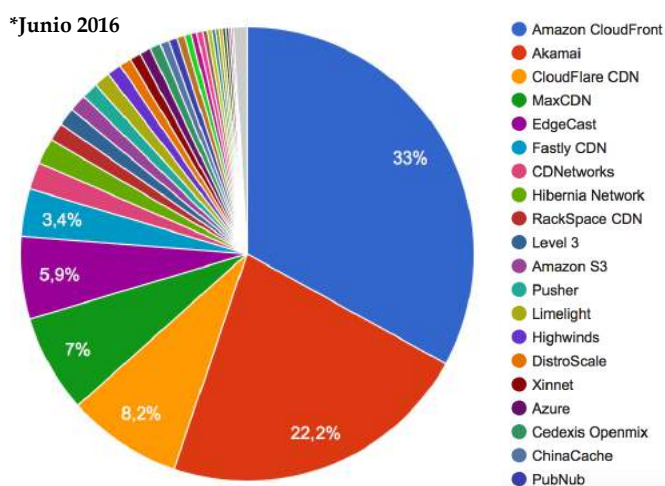


Figura 25. Ranking de las CDNs comerciales según Alexa 100 k<sup>28</sup>

El uso de las CDNs varía según el servicio que un cliente está buscando; es así que las CDNs pueden ser usadas para: *Mejorar el desempeño de un sitio Web*; con las CDNs un sitio Web puede mejorar el rendimiento, mejorar la entrega tanto del contenido estático como del dinámico, con ello las páginas Web se cargarán más rápidamente. *Descarga y actualización de software*, mediante las CDNs podemos mejorar la descarga automática o bajo demanda de software o archivos, incluyendo las actualizaciones software. *Streaming de contenido multimedia*, las CDNs ayudan a la entrega del contenido

<sup>26</sup> <https://www.akamai.com/>

<sup>27</sup> <http://www.cdnplanet.com/cdns/limelight/>

<sup>28</sup> <https://www.datanyze.com/market-share/cdn/Alexa%20top%20100K>

do multimedia, mediante la utilización de los protocolos basados en HTTP como DASH, HLS, HDS, MSS. El streaming del contenido puede ser en directo o bajo demanda. IPTV, las CDNs son utilizadas por los servicios de IPTV, este tipo de servicios puede ser en directo o entrega bajo demanda, etc.

Actualmente tenemos una gran variedad de empresas proveedoras de servicios CDNs comerciales y no comerciales. Entre las CDNs comerciales tenemos a Akamai, MaxCDN, Amazon CloudFront, Limelight, CloudFlare CDN, EdgeCast, Level 3, etc. Entre las CDNs no comerciales tenemos a las CDNs académicas, entre las cuales tenemos a CoDeeN, COMODIN, Coral, Globule, etc. En la Figura 25 podemos ver el porcentaje de las CDNs comerciales más utilizadas por los sitios Web según el ranking Top 100K de Alexa.

#### 4.1.1. Tipos de CDNs

El mercado de las CDNs es muy dinámico, con muchos segmentos y ofertas comerciales diferentes. Es así que a las CDNs las podemos clasificar en los siguientes tipos:

✓ **CDN Dedicada (Pure-play CDN):** Son las CDNs que prestan el servicio de entrega del contenido de audio y video sin la participación de los ISPs en el control y distribución del contenido, es decir, la CDN dedicada entrega el contenido sobre la red de los ISPs (Akamai) o con su propia infraestructura (Limelight, Level 3).

✓ **Telco CDN:** Este tipo de CDNs son desplegadas y utilizadas por los proveedores de servicios de telecomunicaciones y de banda ancha como AT&T, Orange, Telefónica, Verizon, etc<sup>29</sup>. Estas CDNs generalmente no son destinadas al uso comercial; son utilizadas principalmente para la distribución de contenido interno en las redes de los operadores con el objetivo de reducir la demanda interna en la red de backbone. Algunas Telco CDNs ofrecen servicios CDN de forma comercial, por ejemplo, AT&T ofrece sus servicios CDN a Akamai, para la entrega de contenidos mediante su infraestructura en ubicaciones donde Akamai no tiene presencia.

✓ **CDN Controlada (Managed CDN):** Las CDNs dedicadas pueden ayudar a controlar los componentes de una CDN en una red de transporte de contenido, mediante los servicios que ofrecen para este fin, por ejemplo, Limelight ofrece su plataforma Limelight Deploy<sup>30</sup> para el control de los componentes de las CDN. Mediante este enfoque se puede hacer uso de la experiencia, infraestructura y software de una CDN Dedicada.

✓ **CDN Acreditada (Licensed CDN):** Las CDNs dedicadas pueden proporcionar software CDN para la integración, prueba, e implementación de una infraestructura

---

<sup>29</sup> [http://www.idate.org/en/News/Telco-CDN\\_776.html](http://www.idate.org/en/News/Telco-CDN_776.html)

<sup>30</sup> <http://investors.limelightnetworks.com/press-release/limelight-networks-launches-limelight-deploy-managed-cdn-platform-network-operators-ac>



de transporte de contenido. Aunque la asistencia en la integración es inicialmente proporcionado por las CDNs dedicadas, la CDN Acreditada es controlada por el operador de red. Por ejemplo, EdgeCast, Highwinds y Akamai proporcionan productos CDN acreditados.

✓**CDN Federada (Federated CDN):** Múltiples CDNs pueden interconectar sus redes y competir de forma más directa con las CDNs Dedicadas. Esta es una alternativa para un proveedor de contenido que quiere entregar su contenido a un amplio número de clientes que usan la CDN federada. Por ejemplo, Cisco<sup>31</sup> y Highwinds están trabajando a través de CDNs federadas.

#### 4.1.2. Atributos funcionales de una CDN

Los atributos funcionales de una CDN pueden ser descritos bajo 4 aspectos: La composición de una CDN, Control y distribución de contenido, El envío de las solicitudes, y las métricas del desempeño de la CDN.

##### 4.1.2.1. Composición de una CDN

La composición de una CDN depende de varios aspectos de la infraestructura, como son los servidores origen, servidores sustitutos, relación entre los clientes con los servidores sustitutos y cachés proxy, los protocolos que se utilizan para la interacción entre los elementos de la red y por último el tipo de contenido que se distribuye a través de la CDN. [29]

#### • Relación entre componentes de una CDN

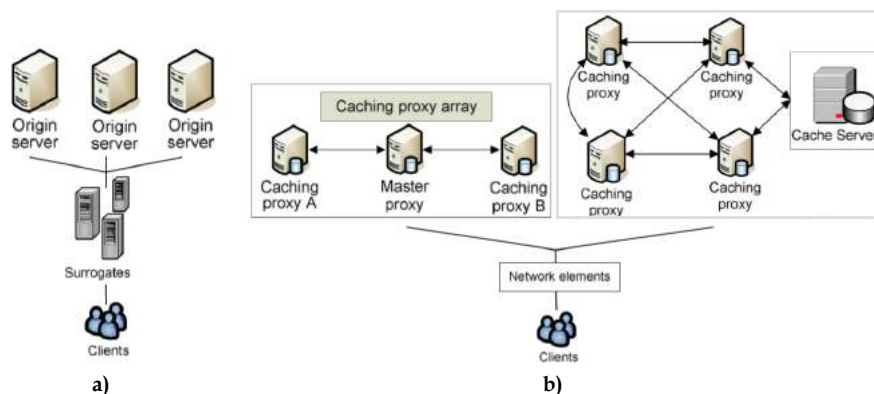


Figura 26. Relación entre los componentes de una CDN

Como hemos dicho la relación entre los elementos tales como los clientes, servidores sustitutos, servidores principal, proxy cachés de las CDNs y proxy cachés Web. Un cliente cuando solicita el contenido un sitio Web, el proceso puede ser de dos formas. El cliente recibe el contenido a través del servidor sustituto de forma transparente, el cliente pensará que lo recibe del sitio Web al que ha accedido.(Figura 26(a)). En caso de que el contenido se encuentre en un servidor proxy caché ya sea en el ISP del cliente o de la CDN, la solicitud del cliente será servida por este servidor proxy caché. Así la

<sup>31</sup> [http://www.cisco.com/c/dam/en\\_us/about/ac79/docs/sp/CDN-PoV\\_IBSG.pdf](http://www.cisco.com/c/dam/en_us/about/ac79/docs/sp/CDN-PoV_IBSG.pdf)

solicitud no tiene que atravesar la red hasta alcanzar al servidor sustituto. (Figura 26(b)) Pueden haber varios proxy cachés y pueden estar configurados en array y en malla para una mejor disponibilidad del contenido en ellos. De todas formas, el contenido entregado por el servidor sustituto al cliente puede ser cacheado por los proxy cachés.

- **Protocolos de interacción**

Los protocolos de interacción entre los elementos de una CDN tenemos dos tipos: Los primeros son los protocolos de interacción o comunicación entre los elementos de la red; entre ellos tenemos al protocolo Network Element Control Protocol (NECP), el protocolo Web Caché Control Protocol (WCCP) y el protocolo SOCKS para los Firewalls. Los segundos son los protocolos de interacción entre las caches; entre ellos tenemos al protocolo Cache Array Routing Protocol (CARP), el protocolo Internet Cache Protocol (ICP), el protocolo HyperText Caching Protocol (HTCP), y el protocolo Cache Digest.

Estos protocolos de interacción están fuera del alcance de este Trabajo Fin de Master, simplemente los hemos mencionado para saber que existen. Para más información acerca de estos protocolos lo podemos encontrar en [29].

#### *4.1.2.2. Control y distribución del contenido*

Las estrategias en cuanto al control y la distribución del contenido es fundamental en una CDN, de ello depende la entrega del contenido de una manera eficiente al usuario final. Para la **distribución** del contenido entra en juego la localización de los servidores sustitutos, la selección del contenido a entregar y la estrategia de cargar los servidores sustitutos. En cuanto al **control del contenido**, depende de las técnicas de organización de las cachés.

- **Distribución del contenido**

Los servidores sustitutos deben estar **localizados** en posiciones estratégicas, lo más cerca posible de los usuarios finales. Aquí entran en juego los dos modelos que hemos explicado antes, *Enter Deep Into ISPs* y *Bring ISPs to Home*. El primer modelo trabaja mejor en sitios con volúmenes de tráfico bajo y medio, mientras que el segundo modelo trabaja mejor en sitios con alto volumen de tráfico. [29]

La eficiencia en la entrega del contenido recae en la correcta **selección del contenido** que va a ser entregado a los usuarios finales. Los servidores principales deben aplicar las técnicas necesarias para cargar los servidores sustitutos con el contenido apropiado. Una correcta selección del contenido influye en un mejor tiempo de descarga por parte del cliente y también reduce la carga en los servidores. Hay dos técnicas utilizadas por las CDNs: Selección y entregar del contenido de forma **completa**; y la Selección y entrega del contenido de forma **parcial**.

En la primera técnica, todo el contenido de los servidores origen o principales son cargados o copiados a los servidores sustitutos, con ello, el cliente podrá encontrar cualquier contenido en el servidor sustituto designado por la CDN. Esta técnica tiene la ventaja en cuanto a su simplicidad, pero en la práctica es poco factible dado la gran cantidad de información que deberían almacenar los servidores sustitutos lo cual implica invertir en muchos más recursos para almacenamiento. Por otro lado, con la segunda técnica, la carga de los contenidos sustitutos se realiza de forma parcial, ahora, por ejemplo, un cliente que accede a una página web obtendrá la página base HTML del servidor origen, mientras que los objetos que lleva la página HTML son cargados desde los servidores sustitutos. Esta técnica es mejor que la anterior debido a que reduce la carga en el servidor origen. La selección de los objetos que son cargados en los servidores sustitutos se puede realizar bajo 3 estrategias: 1) Realizada por el administrador Web, 2) Según la popularidad, 3) Basada en cluster.

En la primera estrategia, el **administrador del sitio Web** selecciona el contenido que va a ser replicado en los servidores sustitutos. Esta estrategia tiene la desventaja en que una mala decisión del contenido a replicar puede volver ineficiente la entrega y distribución del contenido. En la segunda estrategia, los contenidos son replicados según la **popularidad** de los mismo. Esta estrategia tiene como desventaja que no se garantiza la entrega del contenido estático dado que la popularidad del mismo varía considerablemente, además, si el contenido es muy nuevo no podría estar disponible debido a la falta de popularidad. Por último, en la tercera estrategia "**clustering**", el contenido Web es agrupado según la correlación que existe entre él y la frecuencia con que se accedido al mismo. Esta técnica puede estar basada en las sesiones de los usuarios o en las URLs. Si es por las sesiones de los usuarios, un archivo de logs es usado para agrupar las sesiones de navegación de los usuarios, las cuales tienen características similares. Esto permite determinar a los grupos de usuarios que tienen patrones de navegación parecidos, además permite agrupar a las páginas que tienen contenido relacionado. Por otro lado, si se basa a partir de las URLs, el agrupamiento del contenido del sitio Web es realizado según la topología del sitio Web. Los objetos más populares de un sitio Web son replicados en los servidores sustitutos según ciertas métricas de correlación entre las URLs. Esta estrategia reduce el tiempo de descarga de los clientes y la carga en los servidores.

Dada una apropiada localización de los servidores sustitutos en la infraestructura CDN y elegir correctamente el contenido que se va a entregar, necesitamos la forma de **replicar el contenido** a los servidores sustitutos. Para esta tarea, las CDNs pueden utilizar 3 estrategias: 1) Estrategia PUSH cooperativa, 2) Estrategia PULL no cooperativa, y 3) Estrategia PULL cooperativa.

En la **estrategia PUSH cooperativa**, el contenido del servidor origen es replicado en los servidores sustitutos y los servidores sustitutos cooperan para reducir la replicación

y los costos de actualización. La CDN mantiene un mapeo entre el contenido y los servidores delegados, con lo que cada solicitud es re-direccionada hacia el servidor sustituto más cercano que tenga ese contenido y sino lo tiene ninguno, la solicitud es re-direccionada hacia el servidor principal. En la **estrategia PULL no cooperativa**, la solicitud de un cliente (usando re-dirección DNS o re-escritura de la URL) a los servidores sustitutos más cercanos. En caso de no tener el contenido el servidor sustituto, este lo solicita al servidor principal, y una vez que lo ha descargado lo almacena y lo entrega al cliente. Por último, la **estrategia PULL cooperativa** se diferencia de la anterior en el sentido que ahora los servidores sustitutos cooperan entre ellos cuando un servidor sustituto no tiene el contenido almacenado. Las solicitudes de los clientes son dirigidas hacia el servidor sustituto más cercano (mediante re-direccionamiento DNS). En el caso de no tener el contenido, el servidor sustituto usará un índice distribuido para encontrar las copias más cercanas del contenido solicitado y almacenarlo en su caché.

- **Control del contenido**

El control del contenido es esencial para el desempeño de una CDN, el cual depende de la técnica de organización de caché utilizada por la CDN. La organización de la caché está compuesta de las técnicas de caching usadas y la frecuencia de la actualización de la caché que asegure la frescura, disponibilidad y veracidad del contenido. Los contenidos en los servidores sustitutos tienen asociado un tiempo de expiración después del cual caducan. Para garantizar la frescura del contenido es necesario enviar la información de la fecha del mismo. Si hay retardo en la propagación del contenido, un proveedor CDN debe ser consciente que el contenido puede ser inconsistente y/o caducado. Para controlar la consistencia y frescura del contenido de los servidores sustitutos, las CDNs tienen implementadas diferentes técnicas de actualización de la caché.

El técnica más común es la **actualización periódica**; con esta técnica el proveedor de contenidos configura el servidor origen para que proporcione instrucciones acerca de que contenido se puede almacenar, cada cuanto el contenido es considerado como caduco, cuando comprobar el servidor origen para actualizar el contenido, etc. Como desventaja esta técnica puede generar tráfico innecesario generado por cada intervalo de actualización. Una segunda técnica es la **inundación de actualizaciones**, las cuales se propagan ante algún cambio en el contenido. Una versión actualizada de un archivo es entregada a todas las cachés cuando un cambio se ha realizado en el archivo que se encuentra en el servidor origen. Esta técnica tiene la desventaja de que cuando un contenido tiene muchos cambios genera mucho tráfico al realizar las actualizaciones de las cachés. Otra técnica utilizada se denomina **actualización bajo demanda**, en la cual la última versión de un contenido es propagado a la cache de un servidor sustituto cuando previamente ha recibido una solicitud por ese contenido. Mediante esta técnica

el contenido no es actualizado a menos que sea solicitado. Por último, existe otra técnica denominada **invalidación**, en la cual, se envía un mensaje de invalidación a todos los cachés de los servidores sustitutos cuando un contenido está siendo modificado en el servidor origen. Las caches de los servidores sustitutos se mantienen bloqueadas cuando el contenido está siendo modificado. Luego cada cache necesita traer una versión actualizada del contenido. Generalmente las CDNs dan a los proveedores de contenido el control sobre la frescura del contenido y asegurar que el contenido en la CDN sea consistente.

#### *4.1.2.3. Encaminamiento de las solicitudes (Request - Routing)*

Un sistema de encaminamiento de solicitudes es el responsable de enviar las solicitudes de los clientes hacia un servidor sustituto adecuado. Este sistema dirige la solicitud del cliente al servidor más cercano al mismo, aunque, como ya dijimos antes, no siempre el servidor más cercano físicamente al cliente es el mejor para servir al cliente. Un sistema de encaminamiento de solicitudes usa un conjunto de métricas de QoS (proximidad del servidor, latencia percibida por el cliente, carga en el servidor sustituto, y congestión en la red) que permitan dirigir al usuario al servidor sustituto que le pueda servir de una forma eficiente su solicitud. El sistema de encaminamiento de solicitudes en una CDN tiene dos partes: despliegue de un algoritmo de encaminamiento de solicitudes y el uso de un mecanismo de encaminamiento de solicitudes.

- **Algoritmos de encaminamiento de solicitudes**

Un **algoritmo de encaminamiento de solicitudes** es invocado cuando recibe la solicitud de un cliente. Este especifica como seleccionar un servidor sustituto en respuesta a una solicitud de un cliente. Este algoritmo puede ser adaptativo o no adaptativo. Un **Algoritmo Adaptativo** considera el estado actual del sistema para seleccionar a la cache de un servidor para que entregue el contenido. Las condiciones actuales del sistema se obtienen mediante varias métricas como la carga en un servidor sustituto o la congestión de los enlaces de la red seleccionada. El **Algoritmo No-Adaptativo de encaminamiento de solicitudes** utiliza un método de ensayo y error, es decir, no tiene en cuenta el estado actual del sistema. Este tipo de algoritmo es fácil de implementar y trabaja efectivamente en sistemas estables donde se pueda conocer de antemano el posible comportamiento del sistema. Por otro lado, el algoritmo adaptativo proporciona una gran robustez al sistema.

- **Mecanismos de encaminamiento de solicitudes**

Un mecanismo de encaminamiento de solicitudes es una forma de informar al cliente acerca de la selección. Este mecanismo primero invoca al algoritmo de encaminamiento de solicitudes y después informa al cliente acerca de los resultados de selección obtenidos. Estos mecanismos se pueden clasificar de acuerdo a varios criterios, entre los principales mecanismos tenemos: [14]

✓ *Global Server Load Balancing (GSLB)*: Ayuda a optimizar el uso de los recursos, maximiza el rendimiento, minimiza el tiempo de respuesta, y evita la sobrecarga de los recursos. Los servidores sustitutos cuentan con un switch Web, el cual interconecta a los servidores sustitutos que se encuentran distribuidos globalmente. Ahora contamos con dos nuevas características, la primera es que todos los servidores sustitutos informan acerca de su estado de carga, es decir, contamos con información global de la red; la segunda característica es que contamos con una entidad DNS inteligente la cual redireccionará la solicitud hacia los servidores sustitutos que presten mejor servicio. Teniendo la información global de la red, es posible realizar el balanceo de la carga y asignar al usuario a los mejores servidores sustitutos.

✓ *Encaminamiento de la solicitud basada en DNS*: Es ampliamente utilizado en Internet y por muchas CDNs. Los servidores DNS manejan los nombres de dominio de un sitio Web o contenido deseado. El cliente inicia una búsqueda en el servidor DNS local, el cual se encarga de responder con la dirección del servidor sustituto cercano al cliente. Si el DNS local no tiene la dirección, reenviará la búsqueda al servidor DNS raíz. El servidor DNS raíz responderá con la dirección del el DNS autorizado por el sitio Web. El servidor DNS autorizado entonces responderá con la dirección del un servidor sustituto cercano al cliente basándose en un encaminamiento especializado, monitorización de la carga, y mecanismos de mapeo de Internet. Por último, el cliente obtendrá el contenido desde el servidor sustituto designado.

✓ *Redirección HTTP*: Aprovecha las características de redirección del protocolo HTTP. Este mecanismo esta incorporado en un servidor web especial que puede inspeccionar una solicitud de un cliente, elige al servidor sustituto más adecuado y redirige al cliente a esos servidores. Este mecanismo provee flexibilidad para controlar la replicación pero como punto negativo, supone añadir mas información en las cabeceras debido al uso de mensajes extras en la comunicación.

✓ *Reescritura de URL*: El servidor principal redirige a los clientes a diferentes servidores sustitutos mediante la reescritura de las URLs de los enlaces de la página, por ejemplo la imágenes, las URLs de las imágenes pueden ser descargadas de los mejores servidores sustitutos designados por el servidor principal. La principal ventaja de este mecanismo es que la descarga de contenido no depende de un solo servidor sustituto, las URLs de una página pueden señalar a diferentes servidores sustitutos; como desventaja es que el contenido en algún servidor sustituto puede ser obsoleta o esté modificada en origen y no esté actualizada en el servidor sustituto.

✓ *Anycasting*: Es una tecnología de enrutamiento nueva, basada en IPv6. Este es un mecanismo en el cual los datagramas de un único transmisor son encaminados hacia el nodo más cercano de un grupo de potenciales receptores, aunque este puede enviar a muchos nodos, todos están identificados bajo una misma dirección de des-

tino. Las CDNs pueden usar Anycast para encaminar las solicitudes de los clientes a sus centros de distribución o DNS.

✓ *CDN Peering*: Una red de contenido P2P está formada por conexiones simétricas entre ordenadores. Una CDN por pares o CDN Peering, entrega el contenido en el nombre de los demás. Una CDN puede expandir su alcance a un gran número de clientes mediante el uso de parejas de servidores CDN cercanas. Un proveedor de contenidos tiene un contrato con una única CDN y cada CDN contacta con otros pares de CDNs en el nombre del proveedor de contenidos. Este sistema es más tolerante a fallos ya que la información obtenida de la red puede ser almacenada en los miembros “peers”, en lugar de depender de una infraestructura dedicada como las tradicionales CDNs. El contenido en una CDN Peering puede ser colocado de varias formas, por ejemplo, mediante el uso de directorio centralizado, una tabla de distribución Hash o Distributed Hash Table (DHT), un modelo de inundación de solicitudes, o un modelo de encaminamiento de documentos. Cuando veamos las redes P2P, se explicará más en detalle estos modelos.

Los mecanismos más utilizados en la actualidad por las CDNs comerciales son el encaminamiento de las solicitudes basada en DNS, Redirección HTTP, y Reescritura de la URL. Aunque puede haber CDNs que utilicen otros mecanismos. [29]

#### *4.1.2.4. Métricas de rendimiento de la CDN*

Las mediciones de las métricas permiten determinar la capacidad que tiene una CDN para servir a un cliente con un contenido o servicio que requiera. La medición del rendimiento ofrece la capacidad de predecir, monitorizar y asegurar el funcionamiento extremo a extremo de una CDN. La medición es lograda mediante la combinación de hardware y sondas basadas en software que se encuentran distribuidas alrededor de la CDN. Por lo general, 5 métricas claves son usadas por los proveedores de contenido para evaluar el rendimiento de una CDN. Estas son: La tasa de aciertos en la caché, El ancho de banda reservado, la latencia, la utilización del servidor sustituto, la fiabilidad de la CDN mediante medidas de los paquetes perdidos. Las mediciones del rendimiento pueden ser realizadas por los propios proveedores de contenidos o también mediante terceros que certifiquen y garanticen el rendimiento de una determinada CDN.

Para las medidas del rendimiento de una CDN es importante tener en cuenta los datos estadísticos de la red, los cuales pueden ser obtenidos mediante varias técnicas de adquisición de información. Estas técnicas pueden ser:

✓ *Sondeo de la red*: Un ejemplo de esta técnica es el envío de mensajes ICMP ECHO periódicamente a un potencial cliente desde un servidor sustituto o un conjunto de ellos para determinar una o varias métricas.

✓ *Monitorización del tráfico*: En esta técnica, el tráfico entre el cliente y el servidor sustituto o conjunto de servidores sustitutos es monitorizado para conocer las métri-

cas del rendimiento actuales. Estos datos son pasados al sistema de encaminamiento de solicitudes.

✓ *Retroalimentación de los servidores sustitutos:* Esta técnica puede obtener la información mediante sondeos periódicos a un servidor sustituto o conjunto de ellos mediante el empleo de solicitudes específicas HTTP y tomando las medidas relacionadas a las solicitudes.

Los métricas típicas obtenidas mediante el empleo de las técnicas descritas pueden ser: la proximidad geográfica del servidor sustituto, la proximidad de la red, latencia, carga en el servidor, ancho de banda promedio, pérdidas de paquetes, el rendimiento en el servidor, etc.

#### 4.1.3. Funcionamiento de una CDN

Normalmente, cuando un navegador Web quiere acceder a un recurso, primero realiza una solicitud al servidor DNS local, en la cual se le pasa el nombre del dominio y espera recibir una dirección IP del servidor donde se encuentra el recurso que él busca. Con la dirección IP, el navegador podrá entonces contactar directamente con el servidor mediante una subsecuencia de solicitudes.

En el caso de haber una CDN de por medio, cuando el navegador Web hace una solicitud al servidor DNS, hay un proceso diferente que con un sitio con una sola IP. Ahora pueden haber muchas direcciones IP asignadas a un mismo nombre de dominio; esto se debe a que hay muchos servidores sustitutos que se encuentran replicados, lo cual mejora la disponibilidad de la información o de los recursos. El servidor DNS manejará estas solicitudes analizándolas para determinar el mejor conjunto de servidores sustitutos para ese cliente. Al tener los servidores distribuidos geográficamente, los proveedores de servicios CDN pueden optimizar otros aspectos de las CDNs, por ejemplo, redireccionar a un servidor mucho más barato para ejecutar o realizar un balanceo de carga entre varios servidores. Las CDNs responderán inteligentemente con la mejor dirección IP posible para satisfacer las solicitudes de los usuarios.

A continuación veremos el funcionamiento de las CDNs según el tipo de contenido que entreguen, estos pueden ser: contenido estático, contenido dinámico, contenido entregado de forma segura, contenido de streaming y contenido bajo demanda.

##### ✓ **Entrega de contenido estático**

Este es el uso de las CDNs más básico, ayuda a los sitios Web a mejorar el rendimiento, disminuir la latencia; en general, proporcionar mejor QoS y QoE al usuario final. Por ejemplo, cuando un usuario accede a un sitio Web, lo realiza mediante su navegador Web introduciendo la URL del sitio Web. La solicitud viaja por la red hasta alcanzar el servidor origen. Cuando el servidor origen recibe la solicitud, toma la decisión de proporcionar solamente el contenido básico de la página Web, por ejemplo,



index.html). La solicitud del usuario es redirigida por el servidor origen hacia la CDN. La CDN mediante algún algoritmo de encaminamiento de solicitudes (generalmente propietario) seleccionará al servidor sustituto que proporcione al cliente los objetos que se encuentran embebidos en la página Web (Imágenes, logos, anuncios publicitarios, etc), por lo general será el más cercano al cliente. En la Figura 27 podemos observar este proceso. Podemos observar que en este caso el mecanismo de encaminamiento de la solicitud puede estar basado en redireccionamiento HTTP o reescritura de la URL.

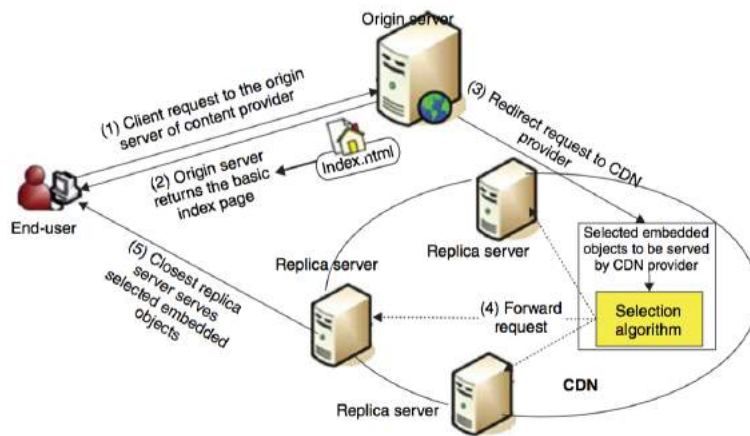


Figura 27. Entrega de contenido estático

### ✓Entrega de contenido dinámico

Este tipo de contenido, a diferencia del anterior, está continuamente cambiando en base a una situación contextual, por ejemplo el perfil de un usuario o características de interacción del usuario. Este tipo de contenido es generado mediante aplicaciones Web e incluyen animaciones, contenido generado mediante scripts como Javascript, PHP, Ruby, etc. [3]

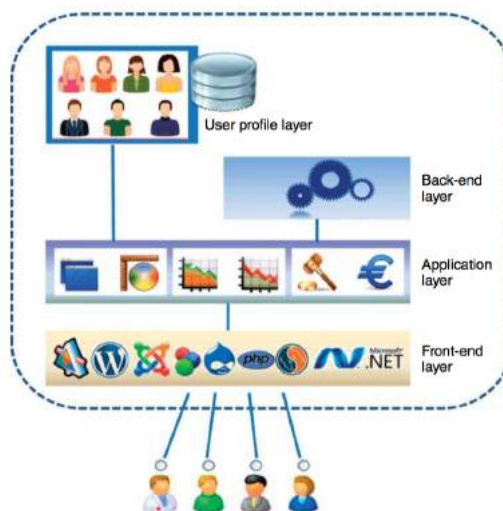


Figura 28. Arquitectura de un sistema Web

La entrega de contenido dinámico sigue una arquitectura lógica de un sistema Web, en donde se separa la interfaz HTTP, la lógica de aplicación, el repositorio de datos, y en caso de que exista, autenticación y personalización del contenido relacionado con el usuario. Cuando se usa una CDN de por medio, todas las capas de este sistema Web son replicadas. En la Figura 28 podemos ver al arquitectura de un sistema Web.

La capa de Front-End recibe las solicitudes HTTP de los usuarios finales, sirve el contenido estático y realiza la función de interface entre la capa de aplicación y el usuario. La capa de aplicación maneja la lógica, es la encargada de proporcionar el contenido dinámico. La generación del contenido dinámico requiere la interacción entre la capa de Back-End y el usuario final. El Back-End maneja la información principal del servicio Web. Generalmente es un servidor de base de datos y que almacena información crucial para la generación del contenido dinámico. La capa del Perfil de usuario es la que almacena la información de las preferencias del usuario. Esta información es utilizada para la generación de contenido personalizado que será entregado al usuario.

#### ✓Entrega de contenido sobre HTTPS

HTTP puede usar dos protocolos que proporcionan seguridad a las comunicaciones, estos protocolos son Secure Sockets layer/Transport Layer Security (SSL/TLS). HTTPS es cómo se le denomina a HTTP cuando utiliza estos protocolos. HTTPS asegura la comunicación extremos a extremos mediante el cifrado de la información y la autenticación de las entidades que se comunican.

Las CDNs son usadas para la entrega de contenido sobre HTTPS, y es de vital importancia en la industria bancaria, e-commerce, instituciones financieras, etc. que necesitan esta característica para la entrega de contenido dinámico, información de transacciones, etc. Para que una CDN pueda proporcionar la entrega de contenido sobre HTTPS, se necesitan tener los siguientes elementos en la plataforma CDN:

✓*Certificado*: Un certificado es documento digital de identificación que permiten a los servidores de la CDN y a los usuarios autenticarse entre ellos.

✓*Clave de sesión*: El usuario final y el servidor sustituto usan una clave de sesión para cifrar la información. Esta es creada por el usuario final a través de la clave pública del servidor sustituto.

✓*Clave pública*: El usuario final cifra la clave de sesión con la clave pública del servidor sustituto.

✓*Clave privada*: La clave privada del servidor descifrá la sesión del cliente.

La entrega de contenido sobre HTTPS asegura la entrega de contenido estático, streaming de música, y la descarga progresiva. Los proveedores de servicios CDN son

empleados para entregar contenido de VoD o Live streaming envían mediante HTTPS los archivos Manifest entre los servidores sustitutos y los reproductores de los clientes. [7]

### ✓Entrega Live Streaming y On-Demand (Ejemplo NETFLIX)

El streaming de eventos en directo es muy similar al streaming de contenido VoD. Para VoD, todo el contenido a distribuir se encuentra disponible en el servidor origen y el archivo manifest es estático. Para Live Streaming o transmisión en directo, los nuevos segmentos son codificados, luego son almacenados en el servidor y el manifest se irá actualizando periódicamente para incluir los segmentos más recientes. Los segmentos antiguos pueden ser eliminados o se los puede mantener en el servidor para convertirlos en contenido de VoD cuando el evento en directo haya finalizado. Manteniendo los segmentos antiguos también permitiría a un usuario empezar la reproducción del evento desde el inicio.[16]

A continuación vamos a analizar como es el funcionamiento de Netflix<sup>32</sup> cuando un usuario solicita un contenido. Netflix es una plataforma de streaming de contenido que ofrece un servicio de SVoD, es decir, VoD basado en suscriptores. En el 2014, Netflix alcanzó los 35 millones de suscriptores en EEUU y alrededor de 48 millones en todo el mundo.[7] En el primer trimestre del 2016, Netflix reportó que alcanzó los 81.5 millones de suscriptores, de los cuales el 42% de los mismos se encuentran fuera de los EEUU <sup>33</sup>.

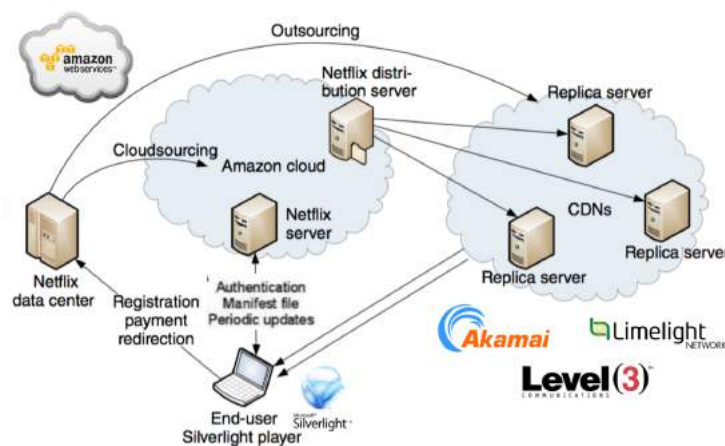


Figura 29. Arquitectura de Netflix

En la Figura 29 podemos observar la arquitectura de Netflix, vemos que esta está compuesta por: Data Center de Netflix, Los servicios Cloud de Amazon, las CDNs, y los reproductores. [7],[10]

<sup>32</sup> Netflix utiliza un modelo de plataforma de distribución de contenido Cloud-CDN que lo veremos más adelante, simplemente este ejemplo es para ver el funcionamiento de las CDN en cuanto a la entrega del contenido VoD

<sup>33</sup> <http://money.cnn.com/2016/04/18/media/netflix-first-quarter-earnings/index.html>

✓ *Data Center de Netflix*: Netflix maneja el dominio [www.netflix.com](http://www.netflix.com). Este servidor se encarga de dos tareas claves. La primera es el registro de las cuentas de los usuarios y la captura de la información de la forma de pago (tarjetas de crédito, cuentas de PayPal, etc). La segunda es la redirección de los usuarios al servidor que almacena el contenido o al servidor de control en el caso de que el usuario no haya iniciado sesión en su cuenta. Este servidor ya no interactuará con el usuario una vez que la película o serie haya sido reproducida.

✓ *La Nube de Amazon*: Los demás servidores que tiene Netflix se encuentran alojados en Amazon Cloud como sus servidores de películas y series, los servicios del DRM, selección de las CDNs, soporte a los dispositivos móviles, análisis y registros de logs son realizados en Amazon Cloud.

✓ *CDNs*: Netflix emplea múltiples CDNs. Los videos una vez codificados y protegidos, son almacenados en Amazon Cloud y luego copiados a las CDNs. Netflix utiliza los servicios CDN de Akamai, Limelight, y Level 3; en las cuales se replica el mismo contenido, en el mismo nivel de calidad y con la misma codificación.

✓ *Reproductores*: Netflix utiliza a Silverlight de Microsoft para descargar, decodificar y reproducir el contenido de Netflix en los ordenadores personales de escritorio. Existen otros dispositivos para la reproducción del contenido como Wii, Roku, etc. Netflix utiliza el protocolo DASH para el streaming del video.

Cuando un cliente de Netflix quiere reproducir un contenido, por ejemplo, una película, una serie, etc. Seguirá el siguiente proceso:

1. Primero el cliente necesitará tener instalado en su navegador Web el Plug-in de Silverlight. A continuación el cliente tendrá que iniciar sesión en su cuenta de usuario. Cuando un cliente quiere reproducir un contenido de Netflix, lo buscará en el sitio Web de Netflix "[www.netflix.com](http://www.netflix.com)"; una vez encontrado el contenido, cuando el usuario hace un clic en "Reproducir Ahora" se iniciará la descarga de la aplicación de Silverlight. La aplicación empezará a descargar los chunks del contenido para iniciar la reproducción. La aplicación será descargada cada vez que se reproduce un video.
2. Como ya vimos, el Manifest informa acerca de las características del contenido a descargar. Es enviado por el servidor de control al reproductor; el reproductor podrá encontrar en él información acerca del contenido que ha solicitado como el bitrate, codecs, etc. El Manifest es generado acorde a las capacidades de cada cliente y es enviado al cliente mediante una conexión SSL para evitar que el contenido pueda ser visto por terceros.
3. Silverlight soporta trickplays simples como pausa, adelantar, retroceder y búsqueda aleatoria.

4. Por último, se realiza la descarga de los chunks de video y audio; al inicio es más frecuente debido a que se debe llenar el buffer lo suficiente para que pueda comenzar la reproducción. Luego las descargas de los chunks se vuelven periódicas, aproximadamente cada 4 segundos. El Manifest contiene múltiples URLs para un mismo video, cada URL corresponde a una calidad diferente del video. También contiene información acerca de las CDNs donde se encuentran los chunks.

5. Después de que la reproducción del contenido haya comenzado, el reproductor reportará periódicamente al servidor de control, los mensajes de control, como los keep-alive y mensajes de errores en el caso de que existan.

#### 4.1.4. Desventajas de las CDNs

Aunque las CDNs son de las principales plataformas para la distribución y entrega de contenido a los usuarios finales, hay dos desventajas grandes desventajas. La primera, es que es una infraestructura muy costosa y compleja de administrar. Al ser una infraestructura costosa, la escalabilidad de la misma es otro punto en contra ya que se necesitaría desplegar nuevos clústers de servidores cercanos a los PoPs de los ISPs. Por último, una desventaja que sufren las CDNs es cuando gran cantidad de usuarios se conectan a un evento de streaming en vivo, es posible que una gran cantidad de usuarios saturen a los servidores sustitutos reduciendo la calidad de la imagen o incluso que deje de proporcionar contenido los servidores, a este fenómeno se le conoce como “flash crowd”. Un ejemplo de este problema surgió cuando el presidente de los EEUU Barack Obama tenía previsto ofrecer una entrevista en directo a través de Facebook Live, pero ante la gran cantidad de usuarios, los servidores de Facebook no soportaron y la entrevista finalmente fue realizada por Youtube.<sup>34</sup>

## 4.2. Peer-to-Peer (P2P)

Peer-to-Peer (P2P) es otra red de tipo overlay que es usada para la transferencia de flujos o streams de chunks de video. Es una plataforma muy potente para una gran variedad de aplicaciones multimedia de streaming sobre Internet como VoD, video conferencia, transmisión en vivo, etc. Un sistema P2P es muy barato ya que utiliza los recursos (CPU, almacenamiento, ancho de banda de subida) de las equipos miembros de esta red, también conocidos como Peers. Cada nodo puede acceder directamente a los recursos de otro nodo. Este modelo de red constituye un modelo computacional distribuido.

En la redes P2P todos los nodos tienen el mismo estatus, permitiendo a los usuarios intercambiar directamente entre ellos información, compartir recursos, sin depender de un sistema centralizado como un servidor. Los nodos pueden realizar las funciones del cliente y las del servidor al mismo tiempo. Si el número de usuarios o nodos incrementa, los recursos globales de la red también lo harán. Esto permite que se logre

---

<sup>34</sup> <http://www.omicro.com/2016/05/bittorrent-live-streaming-por-p2p/>

un balanceo de carga entre todos los nodos. Además un nodo puede unirse a la red o salir en cualquier momento, a este concepto se lo llama “churn”.

Para conocer donde se encuentra un determinado recurso, un archivo o video por ejemplo, o para publicar que un Peer está dispuesto a compartir cierto contenido, las redes P2P cuentan con varios modelos que utilizan para informar a los demás nodos acerca del contenido que dispone la red P2P en el momento que otro nodo lo solicita. Estos modelos son: [29]

✓ *Modelo de directorio centralizado*: Los Peers informan a un servidor, que contiene un directorio centralizado, acerca del contenido que ellos están dispuestos a compartir. Cuando un par elimina o añade nuevo contenido, informará inmediatamente al servidor que contiene el directorio centralizado. En este modelo se basó Napster. Como desventaja de este modelo es que tenía un punto central de fallo y que tenía problemas de escalabilidad.

✓ *Modelo de inundación de solicitudes*: Este modelo es simple pero no escala bien. Cuando un nodo quiere encontrar un recurso en la red, el cual puede estar en un nodo que el no conoce, el nodo solicitante simplemente transmite su consulta a sus vecinos inmediatos. Si los vecinos no tienen el recurso, el nodo inicial pide a sus vecinos que retransmitan su solicitud a los vecinos de sus vecinos. Este proceso es repetido hasta encontrar el recurso o hasta que todos los nodos hayan sido contactados, o tal vez cuando las solicitudes hayan alcanzado un límite de saltos (TTL) impuestos por la red para evitar bucles.

✓ *Modelo de DHT*: En las Tablas Distribuida Hash, los peers se encuentran indexados a través de claves hash y son encontrados a mediante consultas complejas en un sistema distribuido. Cada peer solamente conoce un pequeño número de peers. Cuando se realiza una consulta, la solicitud irá saltando entre los peers hasta alcanzar al peer objetivo. El número de saltos entre peers dependerá del tipo de algoritmo usado, el algoritmo puede ser Chord o Pastry. Este modelo realiza un adecuado balanceo de carga, es decir, la descarga se realizará de los pares que se encuentran menos cargados. Este modelo es el más utilizado en las redes P2P debido a su eficacia y que permite escalar la red.

Las redes P2P juegan un papel importante en una amplia gama de programas y negocios como Skype, Spotify, Adobe Flash Player 10.0, Word of Warcraft, etc. [19] Además, actualmente contamos con muchas aplicaciones de streaming P2P que las utilizan millones de usuarios, como PPLive, PPStream, SopCast, entre otras. [27] En Mayo del 2016, BitTorrent anuncio el lanzamiento de su nueva plataforma de streaming de video BitTorrent Live <sup>35</sup> la cual ofrecerá el servicio P2P Live Streaming de

---

<sup>35</sup> <http://blog.bittorrent.com/2016/05/17/bittorrent-live-multichannel-app-for-live-and-linear-programming-unveiled-at-intx/>

canales de noticias, deportes, música, tecnología y cultura. Este servicio tendrá una latencia baja y sin la necesidad de una infraestructura costosa como las CDNs.

Como hemos visto, las redes P2P permiten que servicios de streaming sean posibles, es así que las plataforma P2P que ofrecían solamente el intercambio de archivos y contenido inmutable ahora también ofrezcan servicios P2P VoD y P2P Live Streaming. En P2P Live Streaming uno o más usuarios tienen sus reproducciones sincronizadas para proporcionar su contenido almacenado a otros peers. Por otro lado, los usuarios de P2P VoD tienen la flexibilidad de mirar cualquier tipo de video en cualquier momento, ellos no necesitan tener sus reproducciones sincronizadas.

#### 4.2.1. P2P Live Streaming

Los servicios distribuidos de P2P Live streaming han proliferado gracias al desarrollo de terminales, aplicaciones y a las mejoras en las prestaciones de las redes. La difusión de video en directo realizado de forma no profesional y la transmisión en directo de programas educativos son ejemplos de los servicios distribuidos de Live Streaming. A continuación veremos las dos topología que este sistema utiliza.

##### ✓ Sistemas P2P Live Streaming basados en Árbol

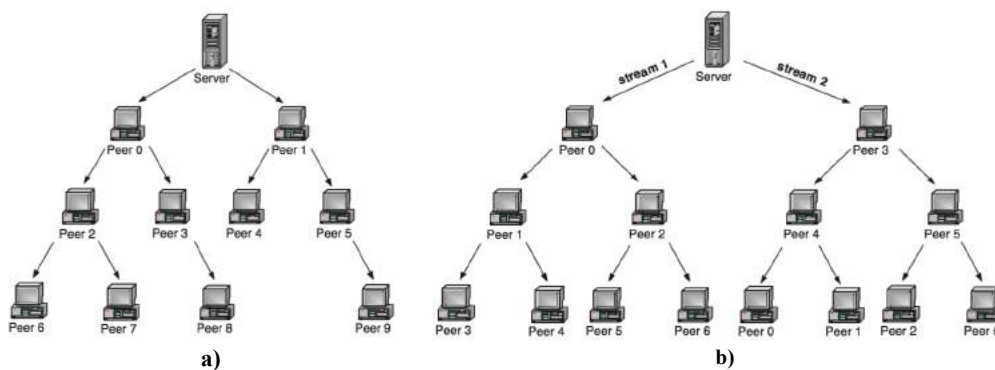


Figura 30. Sistemas P2P Live Streaming basados en Árbol

Este sistema sigue una estructura jerárquica, cuenta con un nodo raíz que en este caso será el servidor que tiene el contenido. Esta basado en sesiones multicast, en la cuales el servidor está conectado a los usuarios que participan en la sección. Los servidores y los usuarios forman una red overlay a nivel de aplicación para distribuir el contenido. Este sistema puede estar basado en un único árbol o en multi-árboles.

✓ *Streaming sobre un único árbol:* En este modelo, cada usuario se une al árbol en cierto nivel. Este recibe el video desde su peer padre en el nivel superior y transmite el contenido recibido a sus peers hijos en el nivel inferior. En la Figura 30 a) podemos observar la estructura de este modelo, en donde el primer nivel está formado solamente por dos peers que reciben el contenido directamente del servidor, estos a su vez transmiten el contenido a los peers hijos que se encuentran en el nivel inferior y el proceso se repite hasta llegar al último nivel. Como desventaja de este modelo

tenemos que los peers de los últimos niveles no contribuyen con su ancho de banda de subida, además si un peer abandona la red, el servidor debe tener un mecanismo de control que permita asignar a los nodos que se quedan huérfanos a nodos de un nivel superior que se encuentren disponibles. ZIGZAG es un algoritmo que permite mantener y construir los árboles de streaming de una manera distribuida. [27]

✓ *Streaming sobre Multi-árbol*: Este modelo mejora el comportamiento de los peers del último nivel los cuales no contribuían al sistema con el ancho de banda de subida. Ahora el servidor fuente divide el stream en múltiples sub-streams. Un simple árbol es dividido en varios sub-árboles, uno para cada sub-stream. Cada peer se unirá a todos los sub-árboles disponibles para obtener los sub-streams y tener diferente posición en cada uno de ellos. Un único peer es posicionado en un nivel intermedio en un solo sub-árbol que esté subiendo un sub-stream, mientras que en los demás sub-arboles puede estar en otros niveles, incluso al final del sub-árbol. Esto permite que se realice un completo balanceo del streaming en todo el sistema. En la figura 30 b) podemos observar esta estructura.

#### ✓ **Sistemas P2P Live Streaming basados en Malla**

Para evitar el problema del modelo anterior, en cuanto a la salida de un peer y dejar huérfanos a los peers en niveles inferiores, muchos sistemas P2P de streaming adoptaron el modelo basado en malla. La principal característica de este sistema es que ya no es un modelo estático; ahora un peer puede colaborar con múltiples peers vecinos descargando o cargando video concurrentemente. Esto lo hace muy robusto cuando un peer abandona la red. Un analizador "Tracker", el cual realiza la monitorización de las actividades durante la sesión de video de un peer, será el responsable de coordinar los peers. En la Figura 31.a podemos ver el proceso cuando un peer quiere unirse a la red. Peer 1 contacta primero con el analizador, dándole su propia información (dirección IP, número de puerto). Luego el analizador le proporcionará una lista que contiene información de un grupos de peers disponibles. Peer 1 intentará iniciar sesiones con los peers de la lista, si son exitosas empezaran a intercambiarse el contenido de video, sino lo seguirá intentando. Para manejar adecuadamente las salidas inesperadas de los peers, los peers se intercambian mensajes de keep-alive frecuentemente. Además, dependiendo de la estrategia de la red, también se pueden intercambiar mensajes cuando un peer tiene mejor desempeño de streaming que otro.

El concepto de video stream en este modelo en sí no es válido, debido a que el stream se realiza mediante chunks de video los cuales tiene un número de secuencia único con el cual se identifican. Cada chunk está distribuido en todos los peers de la malla. Dado que los chunks pueden tomar diferentes caminos para alcanzar un peer, los chunks llegaran en un orden de secuencia diferente. El peer tendrá que ordenar los chunks en el buffer antes de entregarlos al reproductor, asegurando así la reproducción continua.



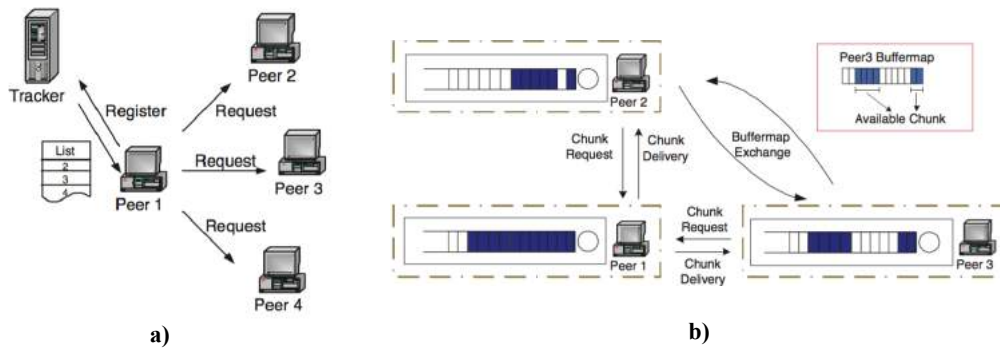


Figura 31. Sistemas P2P Live Streaming basados en Malla

Hay dos formas de intercambiar información en sistema P2P Live Streaming basado en malla: *PUSH* y *PULL*. En un sistema que utiliza *PUSH*, un peer recibe un chunk y lo envía a los peers vecinos que lo necesiten. Sin embargo, debido a la topología de la malla, se producen envíos de chunks redundantes que provocan que los peers gasten ancho de banda de subida innecesario. Por otro lado, un sistema que utiliza *PULL* es usado para evitar este problema. Esta técnica permite a los peers intercambiar periódicamente la disponibilidad de los chunks de acuerdo al estado del buffer “buffermap”. Un estado de un buffer mantiene el número de secuencia actual de los chunks que se encuentran en el buffer del peer; de esta forma un peer puede decidir de que peers descargará los chunks, evitando así la redundancia de los chunks en la transmisión. En la Figura 31.b podemos ver que Peer 1 envía los chunks que necesitan el Peer 2 y Peer 3, además entre Peer2 y Peer 3 se intercambian el buffermap. [27]

#### 4.2.2. P2P Video on Demand (VoD)

Este sistema ofrece más flexibilidad a los usuarios permitiéndoles a los mismos ver cualquier tipo de video cuando quiera. En P2P VoD, el usuarios tiene el control completo del streaming del contenido, mediante la utilización de funcionalidades como pausar, adelantar, retroceder la reproducción de video. En el servicio VoD, muchos usuarios pueden esta mirando el mismo video aunque puede ser en diferente porción del mismo y todo esto asincrónamente. P2P VoD utiliza, al igual que P2P Live streaming, una topología en árbol y otra en malla.

##### ✓Sistemas P2P VoD basados en Árbol

En está topología, los usuarios son agrupados en sesión según sus tiempos de llegada. Cada peer tiene un cache de almacenamiento de un cierto periodo de tiempo. Cuando un peer se une a la red, es asignado a otro peer si esta dentro de su ventana temporal de caché, sino lo está, formará otro grupo de sesión y el servidor origen le proporcionará el contenido. En la Figura 32 podemos ver este proceso en donde la ventana temporal de la cache es de 10 segundo, el peer de la sesión 3 ha llegado en el instante 20 y los peers hijos están dentro de su ventana temporal. El primer peer de la sesión 4, ha sido asignado a una nueva sesión debido a que no esta dentro del la

ventana temporal del peer de la sesión 3, los hijos según van llegando les proporcionara el contenido actual “Base stream” el peer padre mientras que mientras que otro peer le puede proporcionar el contenido anterior a su llegada “Patch Stream”.

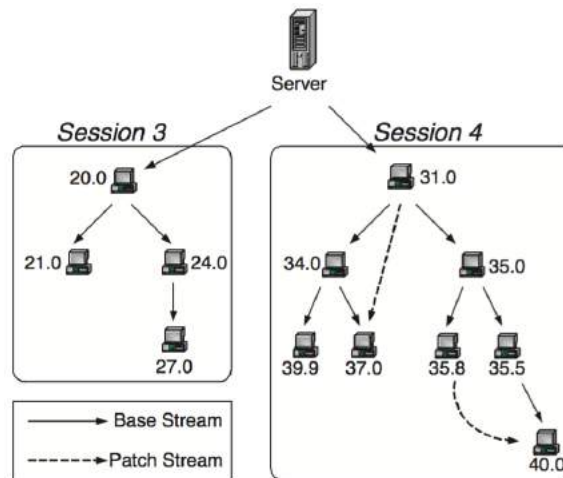


Figura 32. Sistema P2P VoD basado en Árbol

#### ✓Sistema P2P VoD basados en Malla

En este sistema, un archivo es dividido en pequeños bloques de datos, el servidor envía los bloques a diferentes usuarios. Los usuarios descargan desde sus peers vecinos los bloques que ellos no tienen. Para una mejor utilización del ancho de banda de envío (subida) y tener un rendimiento en la descarga lo más alto posible, los usuarios almacenan diferentes bloques de datos con el fin de que siempre haya algo que intercambiar. Esto mejora el rendimiento del sistema. Sin embargo, debido a que en un peer los bloques de datos le llegan en sin un orden específico, para reproducir el contenido debe ordenar el contenido previamente, lo cual provoca cierto retardo en la reproducción.

En [48] se propone un modelo híbrido llamado mTreebone, en el cual, la idea principal es identificar un conjunto de nodos estables para construir un árbol base llamado Treebone. Estos nodos estables, junto con otros nodos no estables están organizados en una topología en malla, esto permite que el Treebone organice a los nodos dinámicos y poder sacar el máximo rendimiento al ancho de banda disponible entre los nodos de la red.

#### 4.2.3. Desventajas de las redes Peer-to-Peer (P2P)

Las redes P2P no son sencillas de administrar, además no proveen QoS debido principalmente al problema de la salida inesperada de los peers “churn”. Los peers por lo general son egoístas e ignoran el beneficio global de la red, suelen eliminar contenido o no lo comparten lo cual implica incrementar el tráfico en la red troncal “backbone” ya que los peers que buscan el contenido lo harán de otros más lejanos físicamente. Es por ello que la popularidad de las aplicaciones de streaming P2P ha

provocado el incremento del tráfico en las redes de los ISPs. Por último, otra desventaja que no es técnica, sino legal, es que la relación entre el propietario de contenido, ISPs y usuarios se ha visto afectada debido a derechos de propiedad intelectual de los contenidos que pueden intercambiarse los usuarios mediante las aplicaciones P2P.

### 4.3. Modelos híbridos CDN-P2P

Como hemos visto antes, las CDNs pueden proporcionar una excelente QoS si la CDN está distribuida adecuadamente, pero, como también dijimos, esto implicaría altos costes en la infraestructura y mantenimiento. Por otro lado, las redes P2P tienen bajo coste pero presentan algunas desventajas como, por ejemplo, la falta de una QoS alta. Es por ello que algunos proveedores CDN y plataformas P2P han combinado ambas tecnologías, proporcionando así una buena solución para proporcionar el servicio de video streaming, ya sea en directo o VoD. A esta nueva arquitectura se le conoce como *redes híbridas CDN-P2P*. Las redes híbridas CDN-P2P presentan una relación QoS - escalabilidad aceptable, pero debido al problema “churn” que presentan las redes P2P, se necesita alquilar algunos servidores de un proveedor CDN y ello puede ser costoso. [42]

Como hemos dicho, esta nueva arquitectura ha generado gran interés tanto para los proveedores CDN como para las plataformas P2P. Akamai, entró en el negocio de las redes híbridas CDN-P2P en el 2007, cuando adquirió la empresa de tecnología P2P Red Swoosh por \$ 18.5 millones<sup>36</sup>. El objetivo principal del uso de P2P en las redes CDN de Akamai fue el de combinar el software de distribución de archivos con su sistema de control de la red global de los servidores sustitutos.

Esta nueva arquitectura permite unir las ventajas de ambas tecnologías para la entrega de servicios de streaming a los usuarios. Como hemos dicho, ahora tendremos mejor QoS, baja latencia y servidores CDN dedicados ofrecidos por las CDNs, mientras que, las redes P2P ayudan a mejorar la QoE del usuario gracias a que ayudan a reducir la carga en los servidores CDN, esto evita el problema de “flash crowd”. Como única desventaja podemos decir que la integración entre ambas tecnologías no es tan simple.

En cuanto a la clasificación de esta arquitectura, en [48] y [46] encontramos que este modelo híbrido CDN-P2P puede ser clasificado según la tecnología de enrutamiento empleada, es decir, si la entrega del contenido lo realiza un servidor sustituto o si lo hace un peer de la red P2P. También pueden ser clasificados según el nivel de acoplamiento entre ambas tecnologías. Por lo tanto, tenemos que un modelo híbrido CDN-P2P pueden ser un sistema híbrido fuertemente acoplado o un sistema híbrido débilmente acoplado.

---

<sup>36</sup> <http://techcrunch.com/2007/04/12/payday-for-red-swoosh-15-million-from-akamai/>

### 4.3.1. Clasificación según la entrega del contenido

Según esta clasificación, los sistemas híbridos pueden ser clasificados en P2P-aided CDN (PAC) y CDN-aided P2P (CAP). PAC implica que una CDN puede necesitar los servicios de una infraestructura P2P para realizar la entrega del contenido a los usuarios finales, mientras que, en CAP sucede lo contrario, una red P2P puede utilizar los recursos de una infraestructura CDN para entregar el contenido.

#### ✓P2P-aided CDN (PAC)

Muchos de los proveedores CDN como Akamai, ChinaCache han integrado sus infraestructuras CDN con redes P2P. En PAC los usuarios que soliciten contenido multimedia será servidos generalmente por la CDN. El sistema P2P ayuda a mejorar el rendimiento del usuario, mejorar la QoE y disminuir el estrés sobre la red CDN ya que otros peers pueden realizar la función de los servidores sustitutos. Con este método, una parte del contenido multimedia es distribuido a los usuarios finales a través de la red P2P. Por ejemplo, la Figura 33, en los sistemas comerciales de Akamai tiene integrado este sistema. Cuando un usuario “peer” solicita cierto contenido, la petición es enviada a un servidor DNS; Akamai utiliza el mecanismo de enrutamiento de solicitudes basado en DNS para el envío de las solicitudes a los nodos CDN más convenientes. El servidor DNS reenviará la solicitud hacia el servidor sustituto CDN más cercano o a un peer de la red P2P que disponga del contenido solicitado. El servidor DNS y los servidores sustitutos realizan también la tarea de Tracker. Por lo tanto, un servidor sustituto CDN puede proporcionar el contenido a un peer o delegar esta tarea a los peers de la red P2P que dispongan de tal contenido. Cuando delega su tarea a los peers, el servidor sustituto CDN actuará solamente como Tracker y ayudará a los peers de la red P2P a encontrar a sus peers vecinos, además, con ello permite no sobrecargar a lo servidores sustitutos de la CDN.

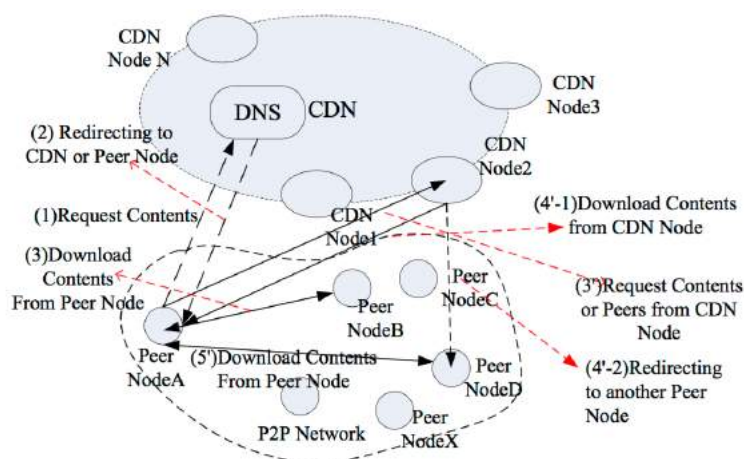


Figura 33. Arquitectura híbrida P2P-CDN (PAC) de Akamai [48]

### ✓CDN-aided P2P (CAP)

En este modelo, mucho del contenido streaming es distribuido a los usuarios mediante las redes P2P. Los servidores CDN ayudan a los peers que no pueden ser servidos por otros peers. Muchas de las aplicaciones P2P como PPLive, UUSee, PPStream han integrado sus redes P2P con las CDNs. [48],[52] Esto les permite reforzar la entrega de contenido a los usuarios. En este sistema la CDN actual como un backup para cuando la red P2P lo necesite. En la Figura 34 podemos observar la arquitectura de este modelo. Cuando un peer se une a la red, este enviará la solicitud al Tracker y este será el encargado de asignar al peer o peers que contienen el contenido que ha solicitado. Además, cuando un servidor CDN quiere necesite contenido de otro servidor CDN también tendrá que informar al Tracker para que le asigne uno. El Tracker contiene información de todo el contenido en el sistema, información de los clientes y de los servidores CDN, además balancea la carga en el sistema. Es por ello que cuenta con otro Tracker de backup en caso de que este caiga. Si un peer designado por el Tracker no puede ofrecer suficiente contenido al peer solicitante, este informará al Tracker, el cual redireccionará al peer usuario hacia un servidor de una CDN o hacia otro peer en la red P2P para que le proporcione el contenido.

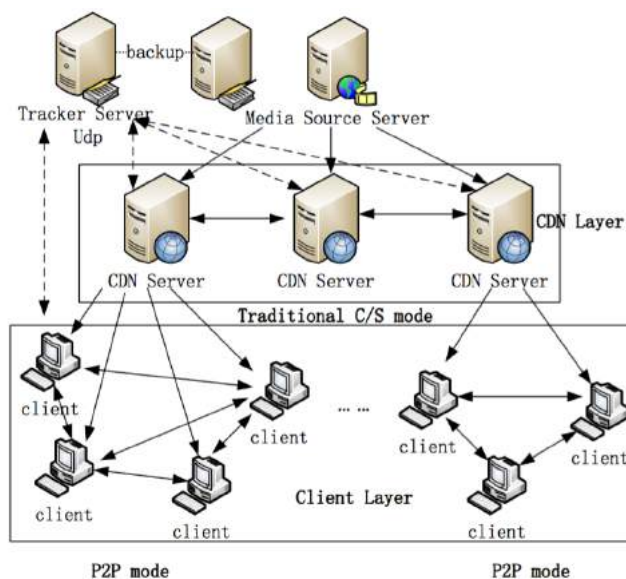


Figura 34. Arquitectura híbrida P2P-CDN (CAP)[50]

#### 4.3.2. Clasificación según el nivel de acoplamiento

Como hemos dicho antes, las redes híbridas CDN-P2P pueden clasificarse según su nivel de acoplamiento. Estos modelos pueden ser: Modelo híbrido fuertemente acoplado y Modelo híbrido débilmente acoplado.

#### ✓Modelo híbrido fuertemente acoplado

También conocido como 1+1; consiste en que un sistema CDN solamente puede integrarse con un sistema P2P. En este modelo, los nodos CDN y los peers P2P colabo-

ran para realizar la función de entrega de contenido a los usuarios finales, los nodos CDN actúan como Trackers para participar en la construcción de la red P2P. Este modelo es eficiente para la integración de una red CDN con una red P2P, pero es ineficiente para la integración con múltiples redes P2P. Por ejemplo, PPLive y PPStream son dos plataformas P2P de streaming multimedia, si PPLive o PPStream quisieran integrarse con una CDN tendrían que romper su actual estructura de red overlay y los mecanismos de transmisión de datos para adaptarse a la CDN.[48]

Peer CDN es una arquitectura híbrida CDN-P2P fuertemente acoplada. Peer CDN realiza el control regional o geográfico cuando construye la red. La red es construida geográficamente y en cada región es dirigida por un nodo CDN fuerte "Strong", el nodo fuerte está colocado cerca de un grupo de usuarios y realiza la tarea de construir la red P2P, controlar y distribuir el contenido entre los peers. En [48] se describe el funcionamiento de esta arquitectura.

#### ✓Modelo híbrido débilmente acoplado

También conocido como 1+N debido a que una CDN puede proporcionar servicios a múltiples sistemas P2P. Ahora la CDN no está involucrada en la construcción de la red P2P, además, la CDN no dirige a la red P2P. Ahora la red P2P será la que solicite los servicios de la CDN. Con lo que, la red P2P puede usar algunos nodos CDN para el servicio de entrega del contenido a los usuarios finales cuando hay problemas de flash crowd. En [48] podemos encontrar una arquitectura que es débilmente acoplada, esta arquitectura es Web Service-based Content Delivery Service Peering (WS-CDSP). La función principal es la de permitir la cooperación e integración de redes P2P, CDN y VoD.

Este modelo puede ser fácilmente extendido en modelos híbridos N+N. En Cloud computing requiere que los servicios de contenidos bajo demanda estén disponibles de una manera abierta y débilmente acoplados.

Una de las mayores plataformas de Streaming VoD en China llamada Xunlei Kanka, utiliza una arquitectura híbrida débilmente acoplada en donde las solicitudes de los usuarios son manejadas por su red CDN y la red P2P independientemente. Cuando tiene un rendimiento ineficiente de su red P2P, utiliza su red CDN para mejorar la QoS y QoE de los usuarios finales.[46]

#### 4.4. Modelo Cloud - CDN (CCDN)

Otra opción con la que cuentan los proveedores de contenidos para la entrega de los mismos a los usuarios finales es la combinación de la Nube o Cloud con las CDNs. Las CDNs cuentan con el problema que ante el incremento de usuarios que solicitan recursos, se necesita invertir en infraestructura que permita hacer frente a estos requerimientos, ello implica costos prohibitivos para los pequeños proveedores CDN

lo cual hace imposible poder competir con los grandes proveedores de servicios CDN como Akamai. Cloud Computing se ha convertido cada vez más y más popular ya que permite almacenar gran cantidad de información, procesar datos masivamente, tiene una alta disponibilidad, escalabilidad y fiabilidad a un bajo coste comparado con las CDNs, lo cual es una solución efectiva para las limitaciones que presentan las CDN.

Con la expansión de la tecnología de Cloud computing, muchas compañías Web han construido sus propias plataformas Cloud; entre estas compañías tenemos a la plataforma de Google Cloud, Amazon EC2, Azure de Microsoft, etc.

La arquitectura de Cloud Computing permite que las CDNs puedan disponer de hardware y software a un coste bajo. Este servicio que ofrece Cloud Computing se lo conoce como Plataforma como servicio - Platform as a Service (PaaS). De igual forma, una plataforma Cloud puede utilizar la estrategia global de balanceo de carga de las CDNs; esto permitiría reducir la congestión en la red cuando se transfiere el contenido y además disminuye el retardo de acceso a los recursos por parte del usuario.

A este sistema que integra las dos tecnologías se lo denomina Cloud-CDN (CCDN). Es una solución flexible, ya que permite a los proveedores de contenido colocar su contenido de una forma inteligente en uno o más servidores de almacenamiento de la plataforma Cloud. Utilizando la plataforma Cloud, los proveedores de contenido pueden hacer frente más efectivamente a los inconvenientes, como el flash crowd, sin la necesidad de invertir en el despliegue de infraestructura nueva. Debemos tener en cuenta que las CDNs siguen siendo la infraestructura principal en distribución y entrega de contenido; además, a este modelo de plataforma se la puede considerar como una CDN asistida por una infraestructura Cloud.

A continuación veremos las principales ventajas que proporciona el modelo CCDN:

✓ *Modelo CCDN Pay-as-you-go*: La CCDN permite consumir el contenido según la cantidad de dinero que están dispuestos a pagar.

✓ *Incremento de los PoP*: El contenido es copiado cerca de los usuarios con relativa facilidad en los sistemas CCDN que en las tradicionales CDNs debido a la omnipresencia de la Nube. Las CCDNs pueden reducir la latencia ya que ante la necesidad, se puede alquilar más recursos del proveedor Cloud para incrementar la visibilidad y el alcance de la CDN bajo demanda.

✓ *Interoperabilidad*: la interoperabilidad de las CDNs a través de la Nube permite a los proveedores de contenido alcanzar nuevos mercados y regiones, dado que puede aprovechar la presencia de los servidores Cloud en una región para asignar dinámicamente servidores sustitutos en la misma.

En la Figura 35 tenemos la arquitectura de un sistema CCDN, El proveedor de contenidos puede colocar el contenido en la Nube, el Master será el responsable de la distribución del contenido. Este asigna en una localización adecuada y guarda un numero fijo de copias en otra área. El proveedor de contenidos no tiene que mantener el servidor origen, porque el Master será el responsable de la integridad de los datos después de que los mismos hayan sido almacenados en la Nube. El servidor origen del proveedor de contenido puede ser visto como un cliente más en la plataforma de Cloud. Cuando el proveedor de contenidos quiere colocar contenido en los servidores cache, primero tendrá que comunicarse con el Master, obtendrá la ubicación de los nodos caché y por último se comunicará con ellos para almacenar el contenido. Una CCDN está dividida en diferentes áreas según la localización geográfica, cada área tiene al menos un servidor caché y un servidor sustituto. A diferencia de la plataformas tradicionales de Cloud, CCDN incorpora un Servidor DNS para disponer con el mecanismo de enrutamiento DNS. [54] Esto permite a los usuarios obtener el contenido que ellos quieren desde el servidor sustituto más cercano reduciendo así el tiempo de respuesta; además, permite mejorar en el balanceo de la carga entre los servidores sustitutos y los cachés.

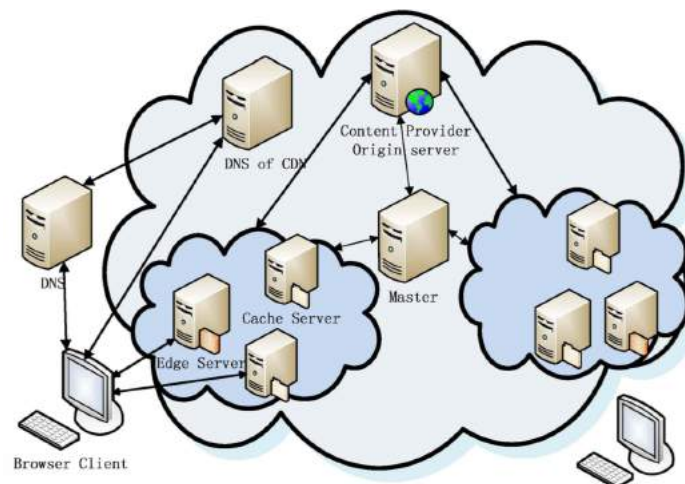


Figura 35. Arquitectura de un sistema Cloud-CDN (CCDN)

Dentro de las soluciones comerciales de las plataformas de distribución de contenidos que utilizan este modelo tenemos a [3][14]:

- *Netflix*: Como vimos en el capítulo 3, Netflix tiene su propio centro de datos; además, utiliza los servicios de la Nube de Amazon (EC2 y S3) para alojar servidores de contenidos y control; y por último utiliza los servicios de tres CDN's (Akamai, Limelight y Level 3) para la distribución y entrega del contenido.
- *RackSpace Cloud Files*: RackSpace ofrece Cloud Files como un servicio CCDN; este servicio es pay-as-you-go y utiliza la red de distribución de contenidos de Akamai.



- *Amazon CloudFront*: Amazon ofrece a CloudFront como un servicio de entrega de contenidos que puede ser integrado con su servicio Cloud Amazon EC2. Este sistema también tiene el modelo pay-as-you-go.

- *MetaCDN*: Es un proveedor de entrega de contenido que ofrece dos tipos de servicios CDN, uno para contenido estático y otro para streaming multimedia en directo. Puede usar los servicios de Cloud de varios proveedores como Amazon S3, Limelight, etc.

- *Limelight Orchestrate*: Limelight es uno de los mayores proveedores de servicios CDN en el mundo, ofrece además servicios de almacenamiento en la Nube, control de contenidos, seguridad y entrega de contenidos a dispositivos móviles.

Tenemos muchos otros ejemplos de soluciones comerciales, además tenemos también CCDNs académicas como COMODIN, Codeen, CodaaS.[14]

Aunque CCDN ha hecho muchos progresos en los últimos años, aún tienen algunos aspectos limitantes, por ejemplo, mover contenido entre la Nube podría acarrear algunos retos en seguridad y rendimiento que pueden impactar en la eficiencia y productividad de la CDN, y por lo tanto, afectar al negocio de los clientes. Otro reto es que las CCDN funcionan mejor en la distribución de contenido estático como video, audio, texto. Pero no funciona muy bien en situaciones donde el contenido es producido de forma dinámica. Por ejemplo, en donde muchos actores pueden modificar el contenido.

Las investigaciones y estudios han permitido que todas las plataformas de distribución de contenidos hayan experimentando grandes avances; mediante la integración de diferentes tecnologías hemos visto que se consigue que la distribución y la entrega de contenido sea cada vez más eficiente, permitiendo así que el usuario pueda disfrutar de servicios y contenidos cada vez de mejor calidad, mejor QoS y QoE del usuario final. Hemos visto que Cloud computing puede proporcionar mejoras en el rendimiento y QoS tanto en las redes CDN; pero también lo pueden hacer con las redes P2P. En [55] se presenta a Cloud-Assisted Live Media Streaming (CALMS) el cual es un framework que facilita la migración del contenido de Live streaming a las plataformas Cloud, ya sea de redes CDN como de redes P2P. Por otro lado, la virtualización también es otro campo de investigación, por ejemplo, en [53] se presenta una solución basada en Redes Definidas por Software - Software Defined Networks (SDNs) que permitan simplificar la implementación de funciones avanzadas de la red para el control del servicio de video streaming en CCDN. Las SDNs permiten una gran flexibilidad en el control de las redes, ya que separa el plano de control, del plano de datos.

#### 4.5. Comparación de las plataformas de distribución de contenidos

Como hemos visto a lo largo del capítulo 4, tenemos varias plataformas de distribución de contenidos disponibles en el mercado, tanto comerciales como académicas, cada una con sus ventajas e inconvenientes. A continuación vamos a hacer una comparación con las principales características de cada una. [3][48]

Características	CDN	P2P	CDN-P2P	Cloud-CDN
<b>Capacidad del servicio</b>	Capacidad del servicio es limitado	Capacidad puede crecer si incrementan los peers	Toma las características de las redes P2P	Capacidad de servicio no es tan limitado gracias a Cloud
<b>Escalabilidad</b>	Escalabilidad costosa	Escalabilidad barata	Escalabilidad no tan costosa	Escalabilidad no tan costosa
<b>QoS</b>	Puede ser garantizada	Best Effort, no es controlable	Puede ser garantizada	Puede ser garantizada
<b>Fiabilidad y estabilidad</b>	Alta fiabilidad y buena estabilidad	Baja fiabilidad, pobre estabilidad	Alta fiabilidad y buena estabilidad	alta fiabilidad, buena estabilidad
<b>Almacenamiento y distribución</b>	Servidores sustitutos	Nodos Peer	Servidores sustitutos y nodos peers	servidores sustitutos y servidores caché de Cloud
<b>Latencia</b>	Baja	Depende del número de peers ofreciendo su ancho de banda y contenido	Normalmente baja, depende de los servidores sustitutos y de los peers	Normalmente baja, depende de los servidores sustitutos y de los servidores caché
<b>Control</b>	Depende del mecanismo de enrutamiento, DNS por lo general	Un servidor Tracker realiza el control de la red.	Tracker en modelo CAP y nodos CDN en modelo PAC	Mecanismo de enrutamiento basado en DNS

Tabla 7. Comparativa de las plataformas de distribución de contenidos

## 5. Última milla en la distribución de Video Streaming

Hasta aquí hemos visto la importancia de la distribución del contenido a los usuarios finales mediante el uso de las plataformas de distribución de contenidos, sin embargo, el papel de los ISPs es uno de los más importantes ya que de ellos depende que el usuario pueda disfrutar del contenido de alta calidad, en este caso el video streaming, ya sea en HD, UHD o 4K, 8K.

El servicio de video streaming VoD de alta definición UHD requiere una capacidad de Internet entre 10 Mbps y 24 Mbps por programa y por usuario, lo cual implica que si hay el fenómeno de flash crowd o un número grande de usuarios en una zona, la capacidad disponible por parte del ISP para los usuarios finales puede degradar la calidad del servicio de video streaming.[11]

En [7] se realizó un estudio de la entrega de video streaming por parte de dos plataformas de VoD, Netflix y Hulu, y se concluyó que la última milla aún sigue generando un cuello de botella, es decir, limita la entrega de video de alta calidad a los usuarios finales. Llegando los mismos a recibir, en algunos casos, video de una calidad estándar SD.

Como vimos en el Capítulo 3, las CDN tienen sus centros de datos y clústers de servidores sustituido cerca de los PoP de los ISPs. Hoy en día, los ISPs cuentan con varias infraestructuras a la hora de proveer los servicios de Internet de banda ancha a los usuarios, por ejemplo, mediante línea de abonado digital o Digital Subscriber Line (xDSL), fibra óptica o Fiber to The Home (FTTH), cable coaxial o Hybrid Fiber Coaxial (HFC), Wifi y redes de telefonía 3G y LTE.

En este capítulo realizaremos un análisis de las dos formas como un ISP presta el servicio de Internet de banda ancha a los usuarios, mediante redes fijas como xDSL, fibra óptica; y mediante redes inalámbricas como WiFi, 3G y LTE.

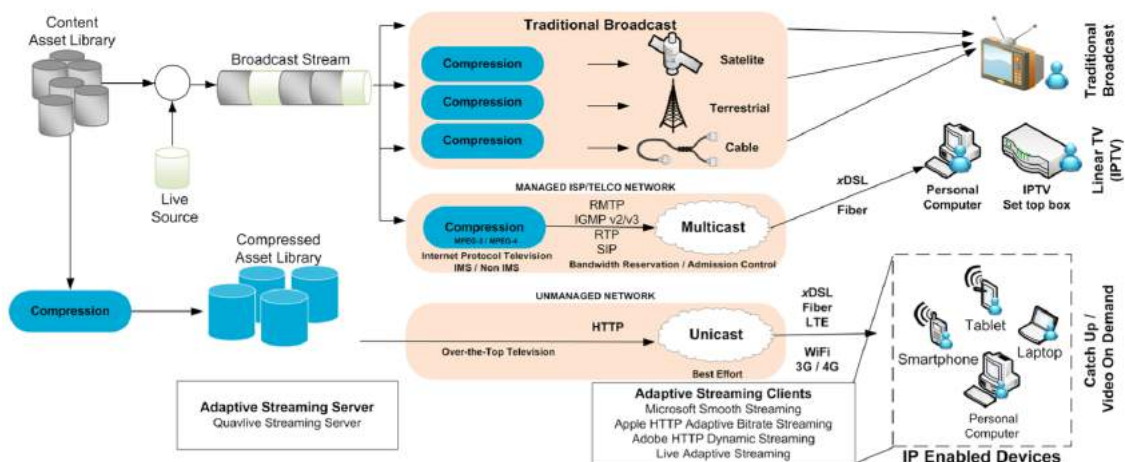


Figura 36. Arquitectura para la entrega de video sobre Internet

## 5.1. Streaming sobre redes fijas (xDSL, HFC, FTTH)

La mayoría de usuarios que están conectados a Internet lo hace mediante cable coaxial y xDSL, de hecho, el 50% de los usuarios usan cable, el 42% usa DSL, y el 8% usa fibra óptica, satélite o sistemas inalámbricos<sup>37</sup>. Sin embargo, xDSL sigue dominando en Europa y el resto del mundo; junto con el cable coaxial.

Los ISPs por lo general conectan sus centrales y PoPs mediante fibra óptica, y solamente la última milla, es decir, desde el bucle de abonado hasta el usuario final lo realiza mediante cable telefónico, coaxial o fibra óptica. A continuación vamos a ver las características de cada una de estas tecnologías.

### ✓xDSL

Como hemos dicho antes, DSL es la línea de abonado digital, esta tecnología de acceso a Internet de nada ancha se realiza mediante el cable (de cobre) telefónico, y tiene la ventaja que se encuentra ampliamente desplegada ya que se aprovechó la infraestructura de la telefonía fija. Actualmente cuenta con diferentes variantes; entre los más conocidas tenemos a el ADSL que es el DSL asíncrono, ADSL2, ADSL2+, y Very High Speed DSL (vDSL).

Con ADSL podemos tener velocidades entre 2 a 10 Mbps de enlace de bajada y hasta 1 Mbps de enlace de subida, en una distancia de hasta 2.5 Km. Por su parte ADSL2 y ADSL2+, como su nombre lo indican, son la evolución de ADSL; estos ofrecen velocidades de 12 Mbps y 24 Mbps de enlace de bajada respectivamente y hasta 1.2 Mbps de enlace de subida. Por último, vDSL ofrece una velocidad entre cinco y diez veces superior al ADSL, lo cual permite que se ofrezcan servicios de HDTV, video bajo demanda, live streaming, entre otros servicios que requieren velocidades altas. vDSL puede ofrecer velocidades de 150 Mbps de enlace de bajada y 12 Mbps de enlace de subida en forma asimétrica y de 26 Mbps en forma simétrica, en una distancia máxima de 0.5 Km.[62]

Como podemos ver, la tecnología xDSL tiene la ventaja que aprovecha una infraestructura ya desplegada. Como desventaja podemos decir que el rendimiento de la red es muy sensible a la distancia, vDSL ofrece mejor velocidad pero a distancias superiores de 0.5 Km se degrada la señal.

Esta tecnología tiene fecha de caducidad en España, en el 2020, Telefónica se dejará de utilizarla, dando paso a otras tecnologías como HFC y FTTH <sup>38</sup>.

---

<sup>37</sup> <http://electronicdesign.com/communications/what-s-difference-between-cable-and-dsl-broadband-access>

<sup>38</sup> <http://www.expansion.com/blogs/uriondo/2014/11/19/telefonica-pedira-en-2015-el-cierre-de.html>

## ✓HFC

Esta tecnología utiliza la fibra óptica y cable coaxial para ofrecer servicios de banda ancha. Esta tecnología conecta al usuario con el PoP o bucle de abonado mediante cable coaxial y interconecta los PoP y centrales de zona de los ISPs mediante fibra óptica. Utiliza la infraestructura de la televisión por cable o Community Antenna Televisión (CATV) lo cual permite tener una capacidad de ancho de banda muy grande que puede superar los 100 Mbps con unas distancias entre los 2 Km y hasta 50 Km con el uso de amplificadores, lo cual es ideal para servicios como el video streaming de alta calidad; además, soporta otros servicios como VoIP, transferencia de datos, etc.

En España un operador que ofrecía estos servicios era ONO, que fue comprada por Vodafone España en el 2014<sup>39</sup>. Con el servicio de TV por cable, ahora ofrecen también servicios de telefonía fija e Internet de banda ancha sobre la misma infraestructura.

## ✓FTTH

Dado a que cada vez hay más demanda de banda ancha para el acceso a Internet, actualmente muchos operadores están desplegando sus redes de fibra óptica para conectar a los usuarios finales con sus PoPs. Esto permite asignar a los usuarios más capacidad en la red; una de las características principales de la fibra óptica es que cuenta con un gran ancho de banda y además no es sensible a la distancia como en el caso del xDSL. Con FTTH se puede alcanzar distancias entre 10 y 60 Km.

Esta tecnología tiene una gran proyección de futuro, debido principalmente que es una solución más barata que el cable de cobre el cual tiene un precio alto en el mercado, el ciclo de vida útil de la fibra es largo y los costes de mantenimiento de la infraestructura es menor con respecto a las otras tecnologías. FTTH ofrece una gran capacidad de ancho de banda, llegando a sobrepasar el 1 Gbps; además, tiene gran escalabilidad, y es inmune a radiación electromagnética la cual puede provocar atenuación de la señal.

En España los principales operadores ofrecen este servicio de Internet de banda ancha, Telefónica, Orange, Jazztel, Vodafone. Además, también ofrecen servicios de telefonía fija y televisión sobre la misma infraestructura FTTH.

## 5.2. Streaming sobre redes Inalámbricas (Wireless - Redes móviles)

Como todos sabemos, los servicios ofrecidos mediante Internet también pueden ser entregados a los usuarios finales por medio de infraestructuras inalámbricas, a las cuales los usuarios pueden conectarse mediante sus dispositivos finales, ya sean ordenadores, tablets, smartphones, etc.

---

<sup>39</sup> <http://www.abc.es/tecnologia/20140724/abci-vodafone-formaliza-compra-201407232328.html>

Entre las tecnologías inalámbricas tenemos a las redes Wifi, WIMAX y las redes de telefonía móvil 3G, y LTE. A continuación vamos a analizar las características de cada una.

### ✓WiFi y WIMAX

La tecnología WiFi permite comunicar a los dispositivos de una forma inalámbrica a un punto de acceso de Internet, un router generalmente. Un operador puede ofrecer los servicios de banda ancha mediante una infraestructura fija y colocar el punto de acceso en el usuario final, con lo que el usuario tiene la opción de conectarse de forma inalámbrica mediante WiFi o de forma fija mediante el cable de Ethernet que se conecta al router.

WiFi tiene el estándar de la IEEE 802.11, y dentro de este tiene diferentes variantes. En la Tabla 8 podemos observar los diferentes estándares WiFi y sus respectivas características.

Estándar	Banda (GHz)	Cobertura (m) Interior/Exterior	Capacidad máxima
802.11	2.4	20 / 100	2 Mbps
802.11a	5	85 / 185	54 Mbps
802.11b	2.4	50 / 140	11 Mbps
802.11g	2.4	65 / 150	54 Mbps
802.11n	2.4/5	120 / 300	300 Mbps
802.11ac	5	60 / 100	1.3 Gbps
802.11ad	60	1 - 10	7 Gbps

Tabla 8. Estándares WiFi<sup>40</sup>

El último estándar certificado 802.11ad puede alcanzar los 7 Gbps que es más que suficiente para disfrutar de video streaming de UHD o 4K e incluso superior, pero para alcanzar esta velocidad se necesita tener una visión directa ya que a la frecuencia que trabaja, 60 GHz, no puede atravesar paredes. Actualmente se encuentran en proceso de estandarización otros dos estándares WiFi que ofrecen un mayor alcance. Estos futuros estándares son los 802.11ah y 802.11af pensados principalmente para los nuevos retos que trae el Internet de las cosas o Internet of Things (IoT). El primero podría alcanzar hasta 1 Km de distancia a costa de limitar la velocidad, desde 150 kbps hasta 18 Mbps como máximo. El segundo, utiliza el espectro que dejó la televisión analógica y se prevé que tendrá un alcance de unos cuantos kilómetros, con velocidades entre 25 y 425 Mbps. En la Figura 37 podemos ver el alcance de los diferentes estándares WiFi.

<sup>40</sup> <http://www.wi-fi.org>

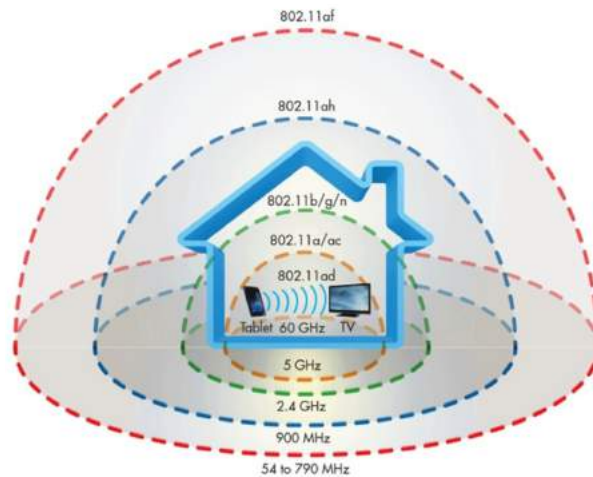


Figura 37. Alcance de los estándares WiFi <sup>41</sup>

Entre las ventajas que encontramos con WiFi es que tiene no tiene conste por el uso de espectro radioeléctrico, pero por otro lado, como desventaja podemos decir que esta limitado por las interferencias radioeléctricas de otros dispositivos que usen el mismo espectro, atenuación debida a obstáculos como paredes, puertas; y además tiene unos rangos de cobertura muy limitados (~ 300 m).[62]

Por otro lado, la tecnología WIMAX ofrece el servicio de internet de banda ancha en zonas rurales y suburbanas, en donde el Internet de banda ancha tradicional no puede llegar debido a la falta de infraestructura. Esta tecnología está estandarizada por IEEE 802.16, como características tenemos que puede tener una cobertura de hasta 80 Km en las frecuencias entre 2.5 a 5.8 GHz, las velocidades que puede alcanzar son 1 Gbps para usuarios que se encuentran en una localización fija y de 100 Mbps para usuarios en movimiento. Tiene dos estándares, 802.16d para WIMAX fijo y 802.16e para WIMAX móvil. Las principales aplicaciones de WIMAX tenemos el acceso a Internet de banda ancha, video conferencia, IPTV y VoIP, entre otros. ya que incluye mecanismos de seguridad y QoS. [63]

Como desventaja de WIMAX es que contratar su servicio tiene un coste superior a los servicios que proporcionan los operadores que cuentan con infraestructuras en zonas urbanas, otra desventaja es que también es susceptible a interferencias radioeléctricas al igual que WiFi.

### ✓ Redes móviles

El streaming de video sobre redes móviles presenta un gran desafío en cuanto a la QoS y QoE, debido principalmente al incremento de los usuarios que utilizan esta infraestructura. Los smartphones y dispositivos móviles cada vez cuentan con más

<sup>41</sup> <http://www.androidauthority.com/wifi-standards-explained-802-11b-g-n-ac-ad-ah-af-666245/>

funcionalidades, soporte para nuevas aplicaciones y servicios multimedia; lo cual implica un aumento exponencial del volumen de tráfico sobre las redes móviles.

La primera tecnología que soportaban la transmisión de video streaming en las redes móviles fue el Sistema Universal de Telecomunicaciones Móviles - Universal Mobile Telecommunications Systems (UMTS) que fue la sucesora del Sistema Global para las Comunicaciones Móviles - Global System for Mobile Communications (GSM), GSM soportaba velocidades máximas de 170 Kbps, las cuales eran insuficientes para soportar servicios de video streaming. UMTS es la primera tecnología 3G y soportaba una velocidad de transmisión de datos de 380 Kbps. ya con esta tecnología empezaron a aparecer los primeros dispositivos que soportaban contenido de audio y video en la telefonía celular.

Posteriormente, fuimos testigos de la evolución del 3G hacia el 3.5G y el 3.75G con los estándares llamados High Speed Packet Access (HSPA) y HSPA+ respectivamente; las cuales soportaban mayores velocidades de transición que UMTS, es así que se alcanzaban velocidades de hasta 7.2 Mbps en condiciones normales con HSPA y hasta 22 Mbps con HSPA+.

Debido a las necesidades del mercado, la evolución tecnología en las redes móviles no ha cesado. Actualmente contamos con la tecnología 4G, dentro de ella se encuentra los estándares LTE y LTE-A que es la versión avanzada de LTE; soportan velocidades de transmisión de datos de 50 Mbps y 150 Mbps (teórico) respectivamente. Con LTE la telefonía móvil ya cuenta con una red de banda ancha que permite disfrutar de muchos servicios disponibles que antes solamente eran posibles por medio de las redes fijas, por ejemplo, el video streaming de alta calidad en aplicaciones como Youtube, Facebook, Skype, Periscope, Netflix, entre otras.

Las redes móviles usan los Servicios Multimedia Broadband Multicast Service (MBMS) y evolved MBMS (eMBMS), tecnología desarrollada y estandarizada por el grupo 3GPP. Este servicio nace como necesidad del envío de contenido multimedia sobre redes móviles ya que permite la transmisión punto-multipunto, es decir, permite la transmisión de datos de un mismo transmisor o fuente hacia múltiples receptores. Como principal característica de este servicio es que mejora la escalabilidad broadcast y multicast en las redes móviles, mediante el uso de un canal común para enviar los mismos datos a múltiples receptores. Con el objetivo de mejorar el rendimiento de MBMS se propuso el uso de una frecuencia única (SFN) en todas las estaciones base pertenecientes a una misma área, en donde se transmite la misma señal al mismo tiempo y en una misma frecuencia; además, permite la gestión eficiente de los recursos radio cuando un grupo de usuarios esta mirando el mismo contenido simultáneamente, con lo que eMBMS reduce la demanda de tráfico mediante la transmisión del un mismo contenido a múltiples usuarios al mismo tiempo. LTE sobre MSMB es conocido también como eMSMB y al igual que MBMS permite mejorar el rendimiento



de las transmisiones multicast; también permite reducir la tasa de error en el canal eMBMS gracias al uso de SFN y a técnicas de corrección de errores como FEC, códigos Raptor, interleaving, etc.

Como vimos en el Capítulo 3, el protocolo FLUTE es utilizado para la entrega fiable de contenido multimedia, video streaming por ejemplo, sobre un canal eMBMS. FLUTE mediante el uso de técnicas de corrección de errores permite la transmisión fiable de servicios multimedia sobre redes LTE, en especial servicio de video streaming en directo o Live streaming.[28] Actualmente podemos disfrutar de varios servicios de video streaming sobre las redes móviles, entre ellos tenemos al video VoD, Live streaming o video multicast sobre eMBMS, video chat y subida de video a plataformas como Youtube. [60] (Figura 38)

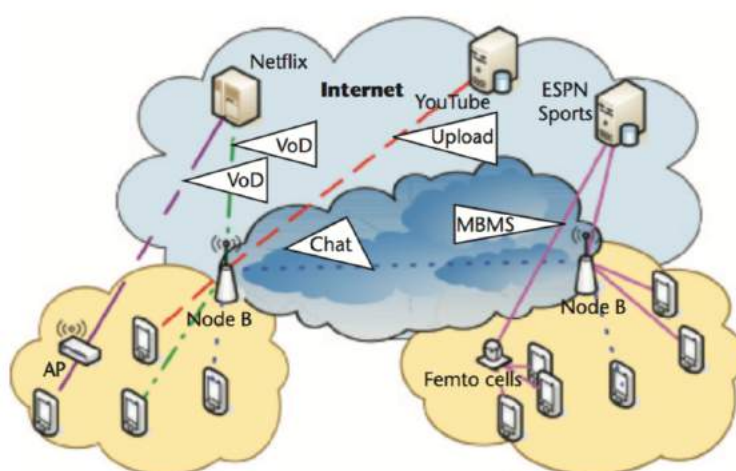


Figura 38. Servicio de video en redes móviles

En España, la mayoría de operadores de telefonía móvil ya ofrecen el servicio 4G a sus clientes, pero la principal desventaja de esta tecnología no es en sí técnica, sino económica. Por lo general, los clientes de las operadoras de telefonía móvil tienen asignados una cantidad fija de datos al mes en sus contratos, dado que los servicios multimedia de alta calidad consumen los datos más que otros servicios, los clientes al final del mes pueden ver incrementadas sus facturas telefónicas en el caso que las empresas facturen cuando un usuario exceda el consumo de los datos contratados. En el mes de Mayo del 2016, Movistar <sup>42</sup> empezó a cobrar por el consumo de datos extras a sus cliente. Esto provoca que los clientes usen otras tecnologías como el WiFi para acceder a contenido multimedia de alta calidad desde sus dispositivos móviles, como los smartphones, tablets, etc. dejando de lado la infraestructura de la red móvil para este tipo de servicios.

<sup>42</sup> [http://www.elconfidencial.com/tecnologia/2016-05-18/telefonica-movistar-internet-datos\\_1176634/](http://www.elconfidencial.com/tecnologia/2016-05-18/telefonica-movistar-internet-datos_1176634/)

Como hemos visto, todas las tecnología empleadas para la entrega de contenido en la ultima milla pueden dar soporte a servicios multimedia de alta calidad como el video streaming. Pero debido a que en al ciudades y zonas urbanas puede haber saturación de usuarios, estos servicios se ven limitados. En la Tabla 9 tenemos un resumen de las tecnologías de las redes de acceso a Internet.

Tecnología	Velocidad típica (Mbps)	Cobertura (Km)	Fortalezas	Debilidades
ADSL, ADSL2, ADSL2+, vDSL	2 - 150	0.4 - 2	Infraestructura de cable telefónico ya desplegada	Sensible a la distancia y mantenimiento costoso
HFC	100	2 - 50	Infraestructura CATV ya desplegada	Costoso donde CATV no tenga presencia
FTTH	1000	10 - 60	Proyección de futuro	Coste en desplegar infraestructura
WiFi	2 - 7000	0.001 - 0.3	Velocidades altas	Afecta interferencias radioeléctricas
WIMAX	100 - 1000	60	Acceso en zonas rurales y suburbanas	Afecta interferencias radioeléctricas
Móviles 3G, LTE	4 - 150	Depende de las estaciones base	Rápida cobertura	Escalabilidad limitada, torres difíciles de ubicar

Tabla 9. Tecnologías de las redes de acceso a Internet

### 5.3. Limitaciones actuales en la entrega del video streaming

Las limitaciones que tenemos hoy en día a la hora de disfrutar de servicios de video streaming de alta calidad no solo tiene que ver con los aspectos técnicos; además de los aspectos técnicos, influye también en gran parte los aspectos económicos, convenios y alianzas entre operadores ISPs, proveedores de servicios CDN y proveedores de contenido, lo que limita la entrega de contenido multimedia de alta calidad en muchos casos.

A continuación vamos a analizar algunas de las limitaciones que en todas las fases que conforman el ecosistemas de la entrega de video sobre Internet.

#### ✓Limitaciones en la generación de contenidos

Debido a la incompatibilidad de muchos dispositivos (Smart TV's, decodificadores, consolas, etc) con el nuevo estándar de compresión de video H.265 o HEVC, el video con resolución 4K se está comprimiendo también con el estándar H.264. Actualmente disponemos de Smart TV's y dispositivos que son compatibles tanto con HEVC como con VP9 y estándares anteriores como VP8 <sup>43</sup>.

<sup>43</sup> <http://4k.com/tv/vizio-p-series-2016-4k-hdr-ultra-hd-tv-review-p50-c1-p55-c1-p65-c1-p75-c1/>

El codificar un video en 4K con H.264 supone un uso mayor de ancho de banda para su transmisión y recepción del mismo que si se utilizara el codec HEVC. En la Figura 39 podemos ver el ancho de banda necesario que un usuario debería para disfrutar las diferentes calidades de video, codificados con H.264/AVC y con HEVC o H.265. Debemos tener en cuenta que en los próximos años se ofrecerá video en calidad 8K, lo cual consumirá aun más el ancho de banda de la red de acceso a Internet.

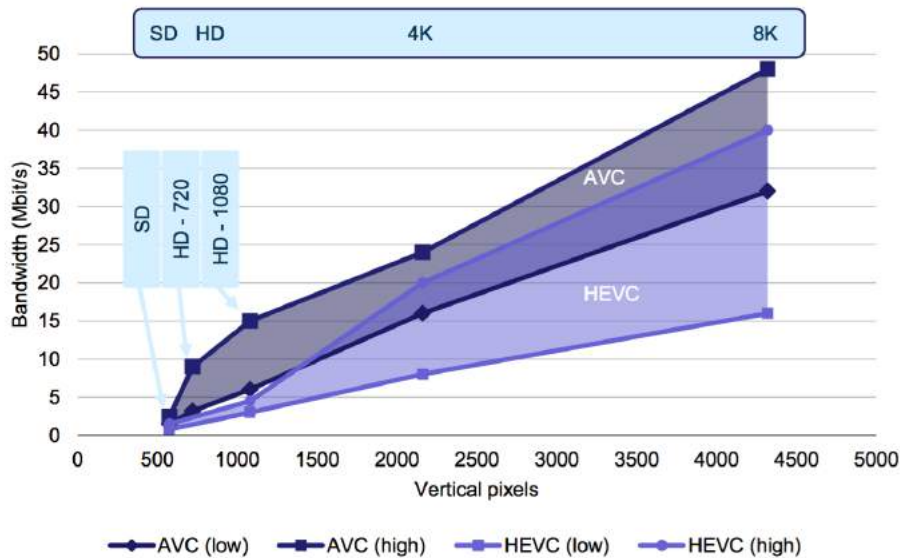


Figura 39. Ancho de banda necesario para video en SD, HD, UHD [65]

Podemos ver que para video en 4K se necesitaría una capacidad aproximada de 25 Mbps para H.264, mientras que para HEVC se necesitaría 20Mbps. [65]

### ✓Almacenamiento

En cuanto al almacenamiento, vemos que mientras más calidad tiene un contenido, el espacio en el almacenamiento es mayor. Sabemos que con H.264 podemos disponer de videos de diferentes calidades pero con un gran volumen de almacenamiento. Por ejemplo, una película de 90 minutos de duración, codificada en 4K con H.264, y con una tasa de transmisión de 25Mbps ocuparía un espacio en almacenamiento de 17 GB, mientras que si se utiliza HEVC con una tasa de transmisión de 20 Mbps ocuparía 14 GB.

Para el contenido de alta calidad, el almacenamiento es un gran reto a tratar, ya que por ejemplo, Netflix tiene alrededor de 13.000 títulos entre series, películas, programas; incluyendo los títulos en 4K. Solo en el 2016 tienen previsto tener 600 horas de contenidos en 4K.<sup>44</sup>

<sup>44</sup><http://www.xatakahome.com/servicios-de-smart-tv/estos-son-los-planes-de-netflix-para-2016-mucho-4k-con-hdr-pero-nada-de-realidad-virtual>

Como podemos ver, esto requiere muchísimo espacio de almacenamiento tanto en los servidores principales como en los servidores sustitutos de las CDNs, o servidores caches de Cloud Computing. Este será un gran reto en los próximos años cuando se popularice aún más el streaming de video en UHD 4K y 8K.

Actualmente este problema se soluciona, en el caso de las CDNs mediante el uso de más infraestructura, es decir más servidores de almacenamiento, pero como hemos visto, es una solución costosa. También se utiliza los servicios de almacenamiento en la Nube de los proveedores como Amazon o Google. Se estima que actualmente hay más de 75 millones de servidores de Internet; Microsoft tiene más de 1 millón de servidores, Google tiene más de 10 millones, Amazon Web Services tiene 1.3 millones, Akamai más de 170.000 servidores.[41]

Pero en un futuro muchas infraestructuras tendrán problemas de escalabilidad sobretodo por el punto de vista económico ya que las mismas requieren una gran inversión. Se estima que para el 2020 se necesitarán 400 millones de servidores Web para hacer frente al los servicios de almacenamiento, procesamiento y computo de grandes volúmenes de datos provenientes de IoT, Big Data, contenidos multimedia, etc.<sup>45</sup> El incremento del numero de servidores y centros de datos conllevan problemas en cuanto a la eficiencia energética.

### ✓Distribución

En cuanto a la distribución, se necesitan de mecanismos y algoritmos que mejoren la eficiencia en la asignación o carga del contenido en los servidores sustitutos y cachés. La mayoría de veces se usa el modo de carga PULL del contenido, pero sería más eficiente si contenido con cierto nivel de popularidad se cargara de modo PUSH para que los servidores sustitutos no tengan que solicitar el contenido a los servidores principales. Además, muchas plataformas de servicios de video streaming, como Netflix y Hulu, utilizan diferentes técnicas a la hora de asignar a un usuario a la CDN para que obtenga el contenido. En [7] se demuestra que Netflix asigna a un usuario a una CDN (de las tres que utiliza) de forma aleatoria y lo mantiene fijo durante días, sin tener en cuenta si otra CDN le puede ofrecer un mejor servicio. Por su parte Hulu, asigna al usuario a una CDN, también aleatoriamente, pero ahora cambiara, de entre las tres CDNs que utiliza, con cada reproducción de un video; ya no permanecerá días asignado a una sola CDN como con Netflix.

### ✓ISPs

Según comenta Reed Hastings, CEO de Netflix, que para ver bien un contenido multimedia en HD se debería tener conexiones de Internet superiores a 15 Mbps. El

---

<sup>45</sup> <https://www.linkedin.com/pulse/400-million-new-servers-needed-2020-think-different-mark-thiele>

stream de video HD requiere un ancho de banda de 5.8 Mbps, mientras que para videos en UHD se necesitan 15.6 Mbps. Por lo que, un hogar necesitara una conexión a Internet de 25 Mbps como mínimo para poder disfrutar de estos servicios. La mayoría de hogares estadounidenses se tienen conexiones en un promedio de 10 Mbps en horas pico, lo cual no es suficiente para video UHD o 4K y mucho peor para 8K.

En España la situación es aún peor; según un ranking publicado por Netflix, en el cual se indica la tasa promedio de bits que ofrecen los ISPs cuando se ofrece el servicio de Netflix en horarios de máxima audiencia. Para saber cual es esta tasa promedio de bits, Netflix ha medido la velocidad a través de los dispositivos finales de los usuarios durante un mes cuando reciben servicios de Netflix. Estas medidas se han hecho solamente en redes fijas, no en redes móviles. En la Figura 40 podemos apreciar las velocidades que ofrecen los ISPs cuando un usuario accede a los servicios de Netflix en España. Esto hace imposible el poder disfrutar de servicios de video streaming UHD de eventos que sean de interés para muchos usuarios.



Figura 40. Ranking Netflix para los ISPs en España <sup>46</sup>

Las causas de este problema no tiene nada que ver con los aspectos técnicos de la Infraestructura, son debido principalmente a la falta de acuerdos comerciales con los ISPs. Muchos ISPs han invertido mucho dinero en sus infraestructuras y las plataformas de OTT ofrecen sus sus servicios sobre estas infraestructuras, sin haber invertido dinero en ellas, es por esto que en la mayoría de casos los ISPs limitan la velocidades de ciertos servicios. En EEUU Netflix tuvo que llegar a acuerdos económicos con los principales ISPs de ese país para que los clientes puedan tener mejor QoS y QoE. Suponemos que los ISPs del resto del mundo esperan el mismo trato. Limitando la

<sup>46</sup> <https://ispspeedindex.netflix.com/country/spain/>

velocidad obligan a las plataformas OTT como Netflix a llegar a mejores acuerdos económicos. Además, muchos ISPs ofrecen su propio servicio de VoD como es el caso de Telefónica tiene el servicio de Movistar Plus <sup>47</sup>.

### ✓Dispositivos

Actualmente la gran cantidad de contenido de video que se ofrece a lo usuarios es contenido HD (720p) y Full HD (1080p) tanto a través de Internet como en otras plataformas como la Televisión Digital Terrestre (TDT), Televisión por Cable (CATV), o satélite. Esto ha provocado que muchos de los usuarios no vean la necesidad de cambiar sus dispositivos (Smart TV's, decodificadores, consolas de videojuegos, ordenadores, etc) para la recepción de video en calidad UHD. La única tecnología mediante la cual las plataformas de video streaming han empezado a ofrecer video UHD a sido Internet. Es así que a partir del 2010 Youtube empezó a ofrecer video en 4K (2160p), actualmente podemos encontrar videos en calidad 8K (4320p)<sup>48</sup>. Otras plataformas como Netflix, Amazon Instant Video, Vimeo, etc.

Los dispositivos finales de los clientes también limitan la recepción de contenido de calidad UHD ya sea mediante Internet como por otras plataformas. Por ejemplo, los Smart TV's necesitan tener una resolución mayor a la FullHD, soportar los nuevos codecs (VP9 y HEVC), soportar el estándar WiFi 802.11ac (en caso de conectarse a la Internet directamente) y contar con puertos HDMI 2.0 que permitan 4K a 60 fps, HDMI 1.4 solamente soporta 30 fps (En el caso de que se conecte a Internet a través de un decodificador o consola de video juegos). Los Smart TV's a partir del 2014 empezaron a implementar el soporte para H.265, pero los anteriores de esta fecha no lo tienen. Actualmente en España se pueden encontrar Smart TV's que soportan 4K a partir de 500 €.

Con respecto a otros dispositivos, los decodificadores, consolas de video juegos como Xbox o Play Station de Sony también deben ser cambiadas por otros que soporten HEVC y VP9. Microsoft lanzará esta año la versión de Xbox One S que soporta UHD tanto para TV como para Blu-Ray, lo mismo sucede con Sony que lanzará la versión de PS4 Neo que soporta contenido 4K tanto para video juegos como para TV y Blu-Ray. Como podemos ver, las consolas aún necesitan adaptarse a esta nueva tecnología de video UHD. En cuanto a los decodificadores, actualmente ya encontramos varias marcas que soportan UHD.

Por último, Smartphones y Tablets por lo general están limitados para la reproducción de contenido en UHD ofrecido en Internet ya sea por las dimensiones de la pantalla como por la batería del dispositivo. La gama media-alta de dispositivos cuenta

---

<sup>47</sup> <https://www.movistar.es/particulares/television/movistar-imagenio/>

<sup>48</sup> <https://www.youtube.com/watch?v=ChOhcHD8fBA>

ya con pantallas con una resolución de 1440p, lo cual permite ver el contenido de video en calidad Full HD, pero el trabajo de computo del dispositivo en cuanto a la decodificación y reproducción del video consumen rápidamente el sistema de energía interno.

A pesar de las limitaciones que presentan los dispositivos móviles, la industria ha apostado por este sector y es así que hoy en día contamos con smartphones capaces de grabar y reproducir video en 4K, entre ellos tenemos a los iPhone 6S/6S Plus, Sony Xperia Z5, Samsung Galaxy S6, entre otros.

En la Display Week 2016 celebrado el mes de Mayo en San Francisco se mostró una pantalla de 5.2 pulgadas con una resolución 4K, también una pantalla para Tablets con resolución de 4K y 8K<sup>49</sup>. Esto demuestra que muchas marcas están apostando por la UHD en dispositivos móviles como los smartphones y Tablets.

Dada esta gran apuesta por parte de la industria en cuanto a dispositivos móviles, han surgido nuevos servicios en cuanto al video, entre las nuevas aplicaciones tenemos a la grabación del video en 360 grados y videos en realidad virtual (VR), que en los últimos meses se han puesto de moda en plataformas como Youtube, Facebook, Twitter, etc.

---

<sup>49</sup> <http://www.displayweek.org>

## 6. Video Streaming a través de una Red Definida por Software SDN

Una de las tendencias en cuanto al despliegue de las redes de comunicación sobre IP es la Virtualización de Funciones de Red - Network Functions Virtualization (NFV) y las Redes Definidas por Software - Software Defined Networks (SDN). Mediante estas nuevas técnicas se busca hacer frente a los nuevos retos que conllevan la escalabilidad, gestión y mejorar el rendimiento de las infraestructuras de redes.

Con el uso de NFVs se consigue que funciones de los dispositivos de la red puedan ser virtualizadas y desarrolladas por máquinas virtuales en lugar de ser desarrolladas por hardware dedicado o propietario. Entre las funciones que se pueden virtualizar tenemos a los routers NAT, Firewalls, conmutadores, balanceadores de carga, almacenamiento. Con ello se mejora la gestión de la red asignando o quitando funciones a los dispositivos virtualizados, por ejemplo, si tengo virtualizado un servidor de almacenamiento puedo configurarlo para asignar más espacio, más capacidad de cómputo, etc. [74]

Por otro lado, las SDNs permiten mejorar la programabilidad de la red, además de facilitar la gestión de la misma gracias a la aplicación de técnicas de ingeniería de tráfico. En una red IP tradicional, la programabilidad de un dispositivo, como los routers o switches, se encuentra dentro de cada dispositivo, con lo que, al realizar cambios en la topología de una red significa realizar cambios en la configuración de los dispositivos intermedios, tablas de flujos, listas de accesos, VLANs, etc. Con las SDNs, el plano de control se separa del plano de datos; esto permite que la gestión de la red sea más sencilla, mediante el uso de un controlador centralizado podemos configurar los dispositivos intermedios de la red que soporten el protocolo Openflow. Openflow<sup>50</sup> es uno de los protocolos estándar más utilizados para la comunicación entre el controlador SDN y los dispositivos intermedios mediante una API de Openflow. La API de Openflow es una API Southbound, es decir, las que permiten la comunicación entre el controlador y los dispositivos de la capa de datos; las APIs Northbound son aquellas que permiten la comunicación entre el controlador o la capa de control con la capa de aplicación o de orquestación. Al inicio, con la versión OF1.0, se definían 12 campos en la cabecera de un paquete IP, actualmente con la versión OF1.5 se definen hasta 44 campos en las cabeceras de los paquetes. En el controlador SDN es el que define las tablas de enrutamiento en los dispositivos intermedios; cuando un paquete llega a un dispositivo intermedio y no tiene ninguna regla definida para el mismo, el dispositivo consulta al controlador acerca de las reglas que debe aplicar al flujo del que proviene dicho paquete, entonces el controlador instala las reglas necesarias para que el paquete pueda ser encaminado hacia el destino. En la Figura 41 podemos ver la diferencia entre las redes tradicionales y las redes definidas por software.

---

<sup>50</sup><https://www.opennetworking.org/>



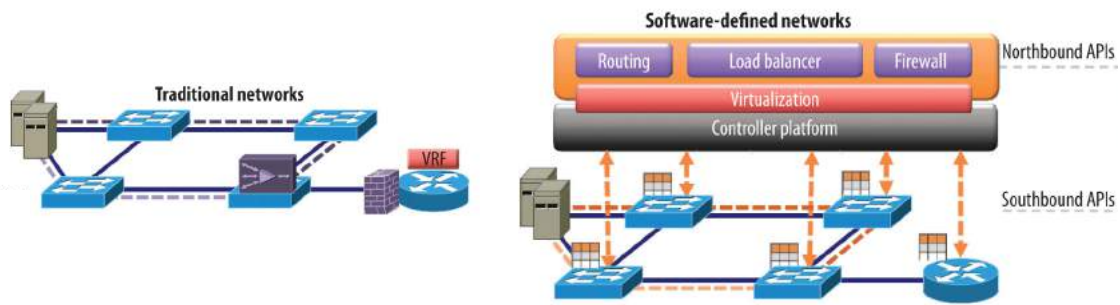


Figura 41. Red tradicional vs SDN

Entre los controladores más conocidos tenemos a NOX, Trema y MUL los cuales están basados en C/C++; Floodlight, Beacon y Maestro basados en Java; POX, RYU y Pyretic basados en Python.

Actualmente las SDNs son aplicadas en un sinnúmero de campos, entre los más importantes tenemos a al campo de la seguridad, que mediante la utilización del controlador se puede definir los tráficos y conexiones permitidos hacia ciertos destinos dependiendo de cual sea su fuente; otro campo son la redes móviles, mediante las redes móviles definidas por software (SDMN) la cual es una arquitectura que permite, del mismo modo que las redes IP, mejorar la programabilidad, apertura e innovación de la infraestructura sin depender de fabricantes de hardware propietario. En [69] podemos encontrar una propuesta para mejorar la calidad del video en redes 5G mediante el uso de SDN. Las SDNs tendrán un role importante en las redes móviles 5G, las cuales se espera que sean desplegadas para el 2020. Otro campo donde se aplican las SDNs es en los servicios multimedia; las SDNs permiten mejorar la QoE de los usuarios finales mediante la optimización de las tareas de administración multimedia, optimización de rutas. Al tener un controlador centralizado, este tiene información acerca de la carga de cada dispositivo intermedio y con esta información puede cambiar la ruta de los flujos de datos por otros dispositivos que permitan optimizar la entrega.

En el siguiente caso práctico evaluaremos el comportamiento de una red virtual cuando se transmite video sobre ella. Para ello hemos utilizado una herramienta que permite emular redes en un entorno Linux. Hemos utilizado una máquina virtual Ubuntu de 32 bits ejecutada sobre VirtualBox<sup>51</sup>, el cual se encuentra instalado en un ordenador MacBook Pro con un procesador de 2,6 GHz i5 y 8GB de Memoria RAM.

### 6.1. Implementación de la red sobre Mininet

Como hemos dicho, Mininet<sup>52</sup> nos permite emular redes definidas por software, es decir, podemos crear host, switches, enlaces virtuales; los cuales se comportan del mismo modo que lo haría un dispositivo hardware que realicen las mismas funciones.

<sup>51</sup> <https://www.virtualbox.org>

<sup>52</sup> <http://mininet.org>

Esto permite el desarrollo y pruebas de nuevas implementaciones antes de llevarlas a la práctica, optimizando así los recursos que se empleen. Una de las principales características de Mininet es que los switches virtuales soportan el protocolo Openflow lo que permite la implementación de redes definidas por software. Todos los elementos creados en Mininet se pueden personalizar, por ejemplo, definir el ancho de banda disponible en un enlace, retardo, el porcentaje de paquetes perdidos, entre otras características.

Los motivos que me han llevado al uso de esta herramienta para la implementación de esta red, fue que me llamo la atención las ventajas que presenta en cuanto a su sencillez. Utilicé Mininet por primera vez en una práctica de la asignatura del master MUIRST llamada Temas Avanzados de Redes de Ordenadores (TARO). La principal característica que vi en Mininet es la rapidez en cuando al despliegue de un escenario; permite describir los escenarios mediante línea de comandos o mediante un script en Python.

Mininet trabaja con un controlador Openflow que se encuentra preinstalado por defecto. Aunque también se puede hacer uso de Sistemas Operativos de Red - Network Operating System (NOS), como los controladores descritos anteriormente. En nuestro caso utilizaremos el controlador POX, el cual es de código abierto y que se encuentra implementado en Python, además soporta Openflow. El controlador POX es un NOS que permite controlar los switches virtuales creados en Mininet; realiza las funciones de un conmutador de capa 2.

Para la realización de este caso práctico se ha implementado una red para la transmisión de video (*videostreaming\_net.py*), la cual consta con 3 hosts (Host 1, Host 2 y Host 3), un servidor Web Apache en el cual hemos alojado el contenido multimedia (Dataset de video codificado en H.264), 3 switches Openflow, 3 routers y un controlador. [72] El experimento consta de dos partes: la primera consiste en servir video streaming desde el servidor hacia una red en donde se encuentra un único usuario y así comprobar las características ABR que ofrecen los protocolos basados en HTTP, precisamente el protocolo DASH, cuando el ancho de banda disponible cambia. La segunda parte consiste en servir a dos clientes que comparten una misma red, con ello comprobaremos, que ante las mismas condiciones de la red que en el primera parte, como afecta a la calidad del video recibido por ambos usuarios. Para la variación del ancho de banda, hemos creado dos script (*bw\_1h.sh* y *bw\_2h.sh*) los cuales permiten ir cambiando el ancho de banda disponible en los enlaces de los usuarios en cada uno de los casos. Estas variaciones van desde los 30 Mbits hasta los 500 Kbits y viceversa. Además, mediante otros dos scripts (*estadisticas\_1h.sh* y *estadisticas\_2h.sh*) comprobaremos algunas métricas en los enlaces de los usuarios como el ancho de banda disponible, el número de paquetes descartados, etc. Cada caso ha sido repetido 10 veces para

la toma de medidas y parámetros presentes ante tales variaciones del ancho de banda. (Los scripts se encuentran en la sección de Anexos)

El contenido multimedia con el que trabajaremos es un corto de animación llamado Big Buck Bunny<sup>53</sup> creado por el Instituto Blender, el cual se encuentra disponible en formato DASH, segmentado en chunks de 4 segundos cada uno, codificado con el codec H.264 con diferentes resoluciones que van desde SD hasta Full HD. Según va cambiando el ancho de banda en el enlace del usuario, la calidad del video entregado por el servidor irá adaptándose al ancho de banda disponible.

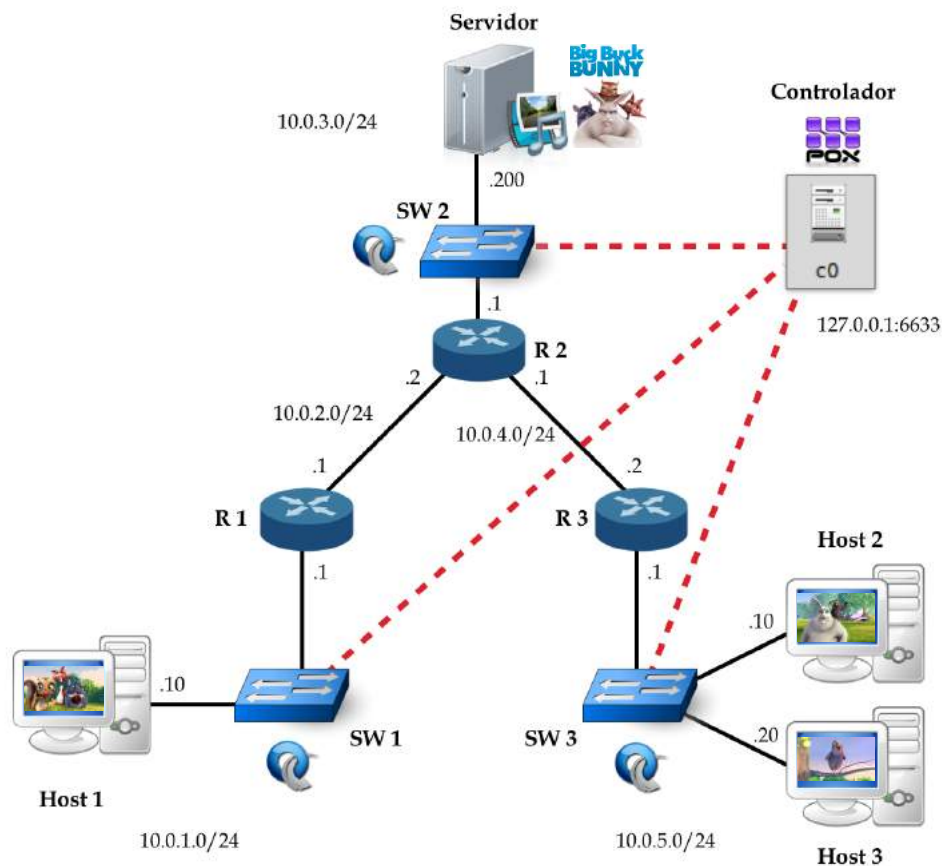


Figura 42. Topología de la red implementada

En la Figura 42 podemos observar la topología de la red implementada. Tenemos 5 subredes, que van desde la 10.0.1.0/24 hasta la 10.0.5.0/24; en la primera subred (10.0.1.0/24) se encuentra el Host 1 (10.0.1.10); en la subred 10.0.3.0/24 se encuentra el servidor (10.0.3.200) y en la subred 10.0.5.0/24 se encuentran los Host 2 (10.0.5.10) y Host 3 (10.0.5.20).

Como hemos dicho, para la variación del ancho de banda se ha implementado dos scripts, uno (*bw\_1h.sh*) para el escenario en donde solamente tenemos al Host 1 y el otro script (*bw\_2h.sh*) para el escenario en donde se sirve video streaming a los Host 2 y

<sup>53</sup> [http://www-itec.uni-klu.ac.at/dash/?page\\_id=207](http://www-itec.uni-klu.ac.at/dash/?page_id=207)

Host 3. Hoy en día es muy común tener contratado una conexión a Internet de unos 30 Mbits, es así que para el experimento hemos ido variando el ancho de banda cada 12 segundos, dado que los chunks de video son del 4 segundos, hemos tomado como referencia un valor múltiplo de 4. Pasados 12 segundos el ancho de banda cambiará a 20 Mbits, luego a 10 Mbits, 5 Mbits, 4 Mbits, 3 Mbits, 2 Mbits, 1 Mbit, 500 Kbits y por último 250 Kbits. Cuando llega a los 250 Kbits irás incrementándose el ancho de banda hasta llegar otra vez a los 30 Mbits.

La variación del ancho de banda se ha realizado mediante la herramienta *iproute2* que dispone Linux a partir de la versión del Kernel 2.2.X. En nuestro caso utilizaremos la herramienta que ofrece *iproute2* llamada *Traffic Control (tc)* que permite realizar la gestión del tráfico en la red. Es así que mediante el comando *tc* podemos ir limitando el ancho de banda en las interfaces de los dispositivos mediante el uso de disciplinas de colas. La disciplina de cola puede seguir varios mecanismos de encolado de paquetes como FIFO (First In, First Out), Token Bucket, y el modelo estocástico o aleatorio. En [68] tenemos disponible toda la información acerca de la herramienta *tc* y las opciones disponibles para realizar la gestión del tráfico en nuestra red. En el presente trabajo nos limitaremos a explicar los comandos utilizados para limitar el ancho de banda en cada caso.

Cabe destacar que *tc* también ofrece la opción de mostrar las estadísticas de una interfaz, en ellas podemos obtener el ancho de banda del enlace, los paquetes recibidos, lo paquetes descartados, el retardo, entre otras métricas.

En nuestro experimento hemos utilizado *Wireshark*, la herramienta *tc*, y las herramientas de desarrollo que incorporan los navegadores Web. Hemos utilizado el navegador Firefox y Chromium. *Wireshark*<sup>54</sup> es una herramienta de software libre que permite capturar y analizar los paquetes intercambiados por los dispositivos en una red en tiempo real, es muy útil ya que presenta la opción de obtener las estadísticas de los datos analizados.

A continuación explicaremos los pasos seguidos para desplegar la topología de la red, como hemos dicho antes, hemos creado un script en Python con la topología de la red (*videostreaming\_net.py*). Entonces los pasos a seguir son los siguientes:

- Abrimos nuestra máquina virtual Linux, en nuestro caso Ubuntu 15.10.
- En el script de la topología de la red utilizamos un controlado remoto "POX" por lo que debemos iniciar el controlador. Hemos utilizado la función que dispone el controlador para el aprendizaje en la capa 2 "*forwarding.l2\_learning*", es así que ejecutamos el siguiente comando; nos aparecerá la información del controlador y los logs de aprendizaje.

---

<sup>54</sup><https://sourceforge.net/projects/wireshark/>

```

vnx@vnxvm:~/pox$ ./pox.py log.level -DEBUG forwarding.l2_forwarding
POX 0.5.0 (eel) / Copyright 2011-2014 James McCauley, et al.
DEBUG:core:POX 0.5.0 (eel) going up...
DEBUG:core:Running on CPython (2.7.10/Oct 14 2015 16:09:02)
DEBUG:core:Platform is Linux-4.2.0-41-generic-i686-with-Ubuntu-15.10-wily
INFO:core:POX 0.5.0 (eel) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[00-00-00-00-00-02 3] connected
DEBUG:forwarding.l2_learning:Connection [00-00-00-00-00-02 3]
INFO:openflow.of_01:[00-00-00-00-00-01 1] connected
DEBUG:forwarding.l2_learning:Connection [00-00-00-00-00-01 1]
INFO:openflow.of_01:[00-00-00-00-00-03 2] connected
DEBUG:forwarding.l2_learning:Connection [00-00-00-00-00-03 2]

```

Con `./pox.py` iniciamos el controlador y con `log.level -DEBUG` obtenemos información acerca del controlador POX, además nos permite observar los flujos instalados en las tablas de flujos y que permitirán tomar las acciones apropiadas para cada uno de ellos.

- Ejecutamos el script que contiene la topología de la red (*videostreaming\_net.py*) el cual se cargará en Mininet. En nuestro caso lo tenemos almacenado en una carpeta llamada “vse” localizada en el escritorio. En la sección de Anexos se encuentra el script `videostreaming_net.py`.

```

vnx@vnxvm:~/Desktop/vse$ sudo python videostreaming_net.py
*** Definimos los Routers
*** Definimos los Switches
*** Definimos los hosts
*** Definimos los servidores
*** Definimos los enlaces
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 r1 r2 r3 server
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (h2, s3) (h3, s3) (r1, r2) (r2, r3) (s1, r1) (s2, r2) (s3, r3)
(server, s2)
*** Configuring hosts
h1 h2 h3 r1 r2 r3 server

```

```
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet>
```

• Una vez desplegada la topología de la red, comprobamos la conectividad entre todos los dispositivos de la red, mediante el comando *pingall*:

```
mininet > pingall
*** Ping: testing ping reachability
h1 -> h2 h3 r1 r2 r3 server
h2 -> h1 h3 r1 r2 r3 server
h3 -> h1 h2 r1 r2 r3 server
r1 -> h1 h2 h3 r2 r3 server
r2 -> h1 h2 h3 r1 r3 server
r3 -> h1 h2 h3 r1 r2 server
server -> h1 h2 h3 r1 r2 r3
*** Results: 0% dropped (42/42 received)
mininet>
```

Si todos los paquetes fueron recibidos, entonces nuestra red está funcionando perfectamente.

• Por último nos falta configurar el servidor para que pueda ofrecer el contenido multimedia que se encuentra almacenado en el directorio `/var/www/html`. Para ello en la consola de mininet ejecutamos el comando *xterm* para desplegar las consolas o terminales de los dispositivos server, host 1, host 2 y host 3.

```
mininet > xterm h1 h2 h3 server
```

En el terminar o consola del server cambiamos de directorio del root hacia el directorio donde tenemos almacenado el contenido multimedia, en nuestro caso se encuentra en `/var/www/html`. Previamente hemos configurado el archivo *index.html*, incluyendo en el mismo el código html del reproductor Dash-js que se encuentra disponible en la página Web de Dash-Industry-Forum<sup>55</sup>. El archivo *index.html* se encuentra en la sección de los Anexos.

Una vez que nos encontremos dentro del directorio donde se encuentra el contenido multimedia, en la consola del servidor, arrancamos el servidor apache para que atienda las solicitudes HTTP de los usuarios.

<sup>55</sup> <https://github.com/Dash-Industry-Forum/dash.js>

```

root@vnxvm:~/var/www/html# sudo service apache2 start
root@vnxvm:~/var/www/html# sudo python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...

```

Hasta aquí, tenemos lista nuestra topología de la red para el envío de video a los usuarios (Host 1, Host2 y Host 3) desde el servidor (Server). A continuación veremos los dos casos de estudio; el primero es de un solo usuario y el segundo caso es cuando el enlace es compartido por más de un usuario. En la Figura 43 podemos observar ambos escenarios.

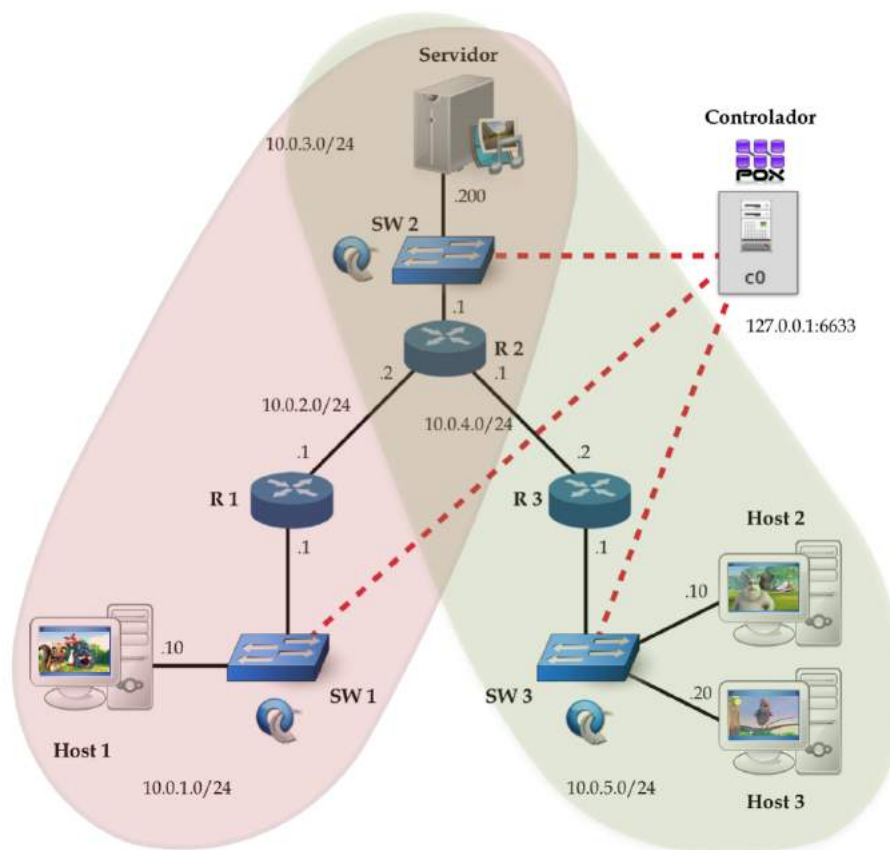


Figura 43. Escenarios de análisis

### 6.1.1. Escenario de un solo usuario (Servidor - Host 1)

En este primer caso, el usuario Host 1 solicitará el video Big Buck Bunny de 9:56 de duración. Desde el terminal o consola del Host 1, desplegado mediante mininet con el comando xterm, abrimos el navegador Web Firefox o Chromium de la siguiente forma.

```

root@vnxvm:~/Desktop/vse# sudo firefox &
0
root@vnxvm:~/Desktop/vse# sudo chromium-browser &

```

Una vez abierto el navegador Web, introducimos la dirección IP del servidor (<http://10.0.3.200>) dando lugar a la reproducción del video. Al contar el reproductor con todo el ancho de banda disponible en el enlace, el video se reproducirá con la máxima calidad disponible, es decir, Full HD 1080p con un bitrate de 3.9 Mbps.

Para poder variar el ancho de banda del enlace del usuario, vamos a aplicar el script *bw\_1h.sh* desde la consola principal de Ubuntu. En este script hemos utilizado la herramienta *tc* para limitar el ancho de banda en las interfaces del switch 1. A continuación describiremos los comandos utilizados en este script:

```
sudo tc qdisc add dev <interface> root handle 1: tbf rate <RATE> burst 2000  
latency 60ms  
sudo tc qdisc add dev <interface> parent 1:1 handle 10: netem delay 17ms 3ms  
15%
```

Con `tc qdisc add dev <interface>` añadiremos una disciplina de cola a la interfaz del switch 1; `handle` es un descriptor que permite referenciar la disciplina de cola; `tbf` es el tipo de la disciplina de cola, en este caso Token Bucket Filter en donde tenemos que definir el `rate` el cual es la frecuencia de los tokens en la disciplina de cola, este será el ancho de banda de salida de la interfaz; el tiempo máximo de permanencia del paquete en la cola se define mediante la `latency` y el tamaño del buffer se define mediante `burst` que en nuestro caso será de 2000 bytes.

Con la segunda línea especificamos el retardo del enlace, en nuestro caso hemos puesto 17 ms con una variación de  $\pm 3$  ms el 15 % de veces. Hemos tomado 17 ms dado que según un informe de la Comisión Europea<sup>56</sup> señala que el retardo medio en una red xDSL a nivel europeo se encuentra en 35ms, al aplicar a las dos interfaces del switch 1 las disciplinas de colas, tendremos 34 ms de retardo en total  $\pm 3$  ms de variación. El script irá variando el ancho de banda cada 12 segundos, como ya lo habíamos dicho antes; empezará desde los 30Mbps hasta los 250 Kbits. En la sección de anexos podemos encontrar el script *bw\_1h.sh*.

Por otro lado, para obtener las estadísticas de la reproducción, como ya lo dijimos antes, hemos utilizado el comando *tc*, los logs de los servidores, las herramientas de los navegadores y Wireshark. Hemos creado otro script (*estadisticas\_1h.sh*) con el comando *tc* que nos permite obtener las estadísticas acerca del ancho de banda en la interfaz cada 8 segundos, los paquetes recibidos, los paquetes descartados, etc. Los comandos utilizados en este script son los siguientes:

Con `show`, `-p`, `-d`, `-s` obtendremos los detalles de las estadísticas tomadas en cada interfaz; del igual forma, hemos tomado las estadísticas en ambas interfaces del switch 1.

---

<sup>56</sup> <https://blog.cnmcs.es/2015/04/16/no-es-la-velocidad-es-la-latencia/>



```
sudo tc -p -s -d qdisc show dev <interface>
```

Desde Ubuntu abrimos otros dos terminales o consolas para la ejecución de cada uno de los scripts, en la primera ejecutamos el script *bw\_1h.sh* y en la otra el script *estadisticas\_1h.sh*. Una vez arrancados los scripts, debemos reiniciar otra vez la reproducción en el navegador Web.

```
vnx@vnxvm:~/Desktop/vse$ sh ./bw_1h.sh
```

```
vnx@vnxvm:~/Desktop/vse$ sh ./estadisticas_1h.sh
```

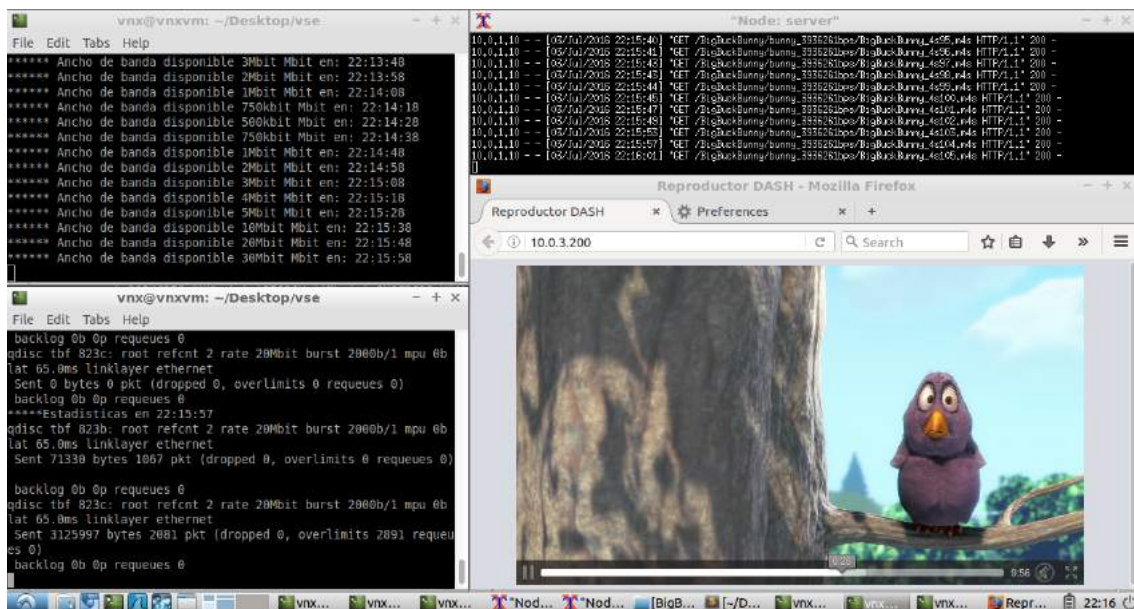


Figura 44. Captura de la reproducción en el primer escenario

En la Figura 44 podemos observar la reproducción del video en el primer caso, tenemos los dos terminales ejecutando los dos scripts y el terminal del servidor “Server” en donde podemos observar la calidad del video que el servidor envía al Host 1 en cada instante.

### 6.1.2. Escenario de más de un usuario (Servidor - Host 2 y Host 3)

Para este segundo escenario, utilizaremos los terminales desplegados desde la consola de mininet con el comando *xterm*; además, utilizaremos el script *bw\_2h.sh* para realizar la variación del ancho de banda en las interfaces del switch 3, al igual que el primer caso; y el script *estadisticas\_2h.sh*, que junto con Wireshark y las herramientas de los navegadores, nos servirá para la obtención de las métricas mientras el video es reproducido en ambos hosts. Ambos scripts son similares a los utilizados en el primer escenario, la variación del ancho de banda se realiza cada 12 segundos yendo desde los

30 Mbits hasta los 250 Kbits y la medición de las métricas se realiza cada 8 segundos en las interfaces del switch 3.

Para iniciar la reproducción del video en los usuarios Host 2 y Host 3, desde las consolas de los Hosts 2 y 3, lanzamos los navegadores Web, en este caso usaremos Chromium para Host 2 y Firefox para el Host 3. En las 10 veces que se ha reproducido en video para este escenario se han ido intercambiado los navegadores Web entre ambos hosts para que no afecte en los resultados, y hemos comprobado que no hay diferencia alguna.

A continuación ejecutamos, desde dos consolas o terminales de Ubuntu, los scripts *bw\_2h.sh* y *estadisticas\_2h.sh* e iniciamos la reproducción del video mediante las solicitudes al servidor "Server" introduciendo la URL en los navegadores Web.

```
vnx@vnxvm:~/Desktop/vse$ sh ./bw_2h.sh
```

```
vnx@vnxvm:~/Desktop/vse$ sh ./estadisticas_2h.sh
```

El servidor empezará a servir a ambos host pero al empezar a decrecer el ancho de banda disponible en los enlaces, la calidad del video se vera reducida en ambos hosts, en algunos casos en uno más que en el otro. En la Figura 45 podemos observar la captura de pantalla para este nuevo escenario, en donde podemos observar que la calidad del video en el Host 2 (navegador Chromium) es mejor que la del Host 3 (navegador Firefox).

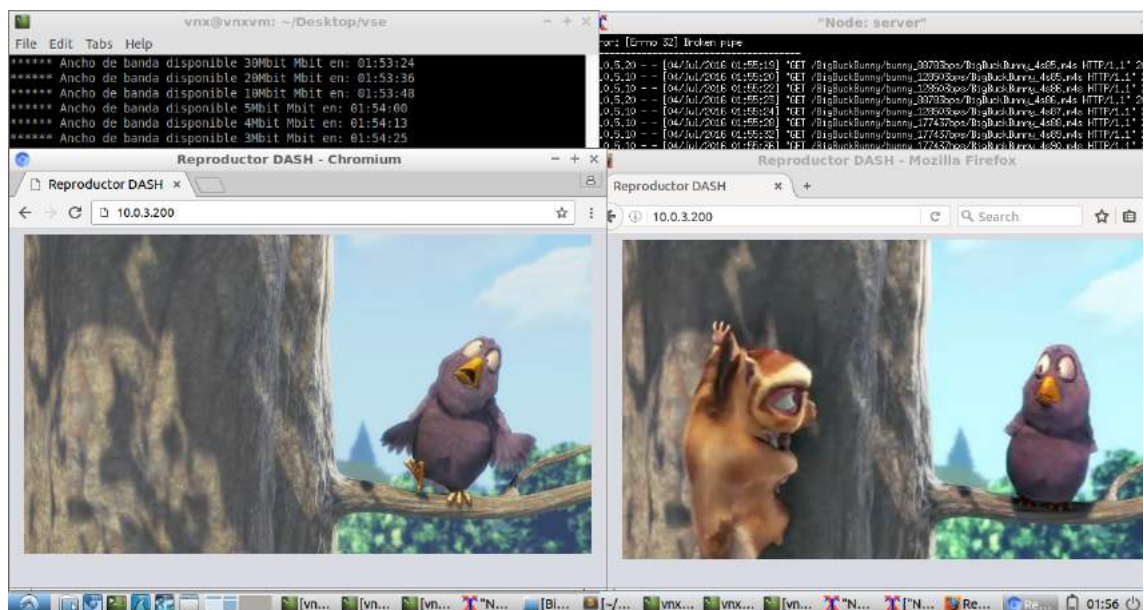


Figura 45. Captura de la reproducción en el segundo escenario

A continuación vamos a analizar los resultados obtenidos para cada caso y así poder sacar las conclusiones.

## 6.2. Resultados y conclusiones del experimento

Como hemos dicho antes, para la obtención de los resultados hemos utilizado los scripts, Wireshark y las herramientas de desarrollador disponibles en los navegadores. Con los scripts hemos obtenido el ancho de banda en el enlace cada 8 segundos y también los paquetes descartados o perdidos. Mediante Wireshark hemos podido obtener el total de paquetes enviados desde el servidor hacia el Host 1, además de comprobar el intercambio de paquetes entre el servidor y el Host 1; mediante las herramientas de desarrollador de los navegadores hemos obtenido el periodo de carga de los buffer en los navegadores. Es así que hemos podido evaluar la calidad de video que es entregado a cada usuario, el comportamiento de los buffer de los navegadores, el comportamiento del ancho de banda en los enlaces de los usuarios y el número de paquetes perdidos en cada caso. Para ello primero hemos realizado las mediciones en un escenario en donde no existe ninguna perturbación en cuanto al ancho de banda disponible en el enlace de los usuarios para poder compararlo cuando si existen perturbaciones.

En la Figura 46 podemos observar la calidad de video que el usuario recibe en cada instante. Tenemos la medición de un Host ante ninguna variación en el ancho de banda, en donde todo el tiempo recibe el video en calidad Full HD.

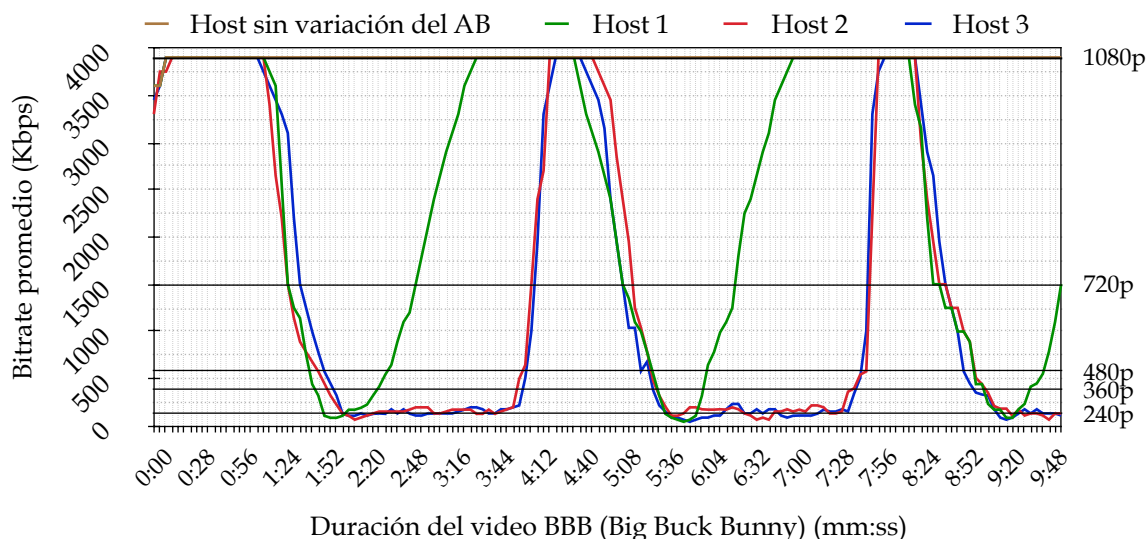


Figura 46. Variación del Bitrate en función del tiempo

Podemos observar que para el escenario del Host 1, ante las variaciones del ancho de banda, la mayoría de tiempo ofrece calidad en HD y Full HD, decayendo a calidad SD cuando el ancho de banda baja de 1 Mbps. Por otro lado, podemos observar el comportamiento de la calidad del video los Host 2 y Host 3, en este caso, debido a que comparten el enlace de la red, la mayoría del tiempo van a reproducir video en SD y

HD, solamente reproducirán video Full HD cuando la capacidad del canal llega a 20 Mbits una vez superado el tope de los 30 Mbits. En la Figura 47 podemos ver la variación de la calidad del video en función del ancho de banda disponible en el enlace.

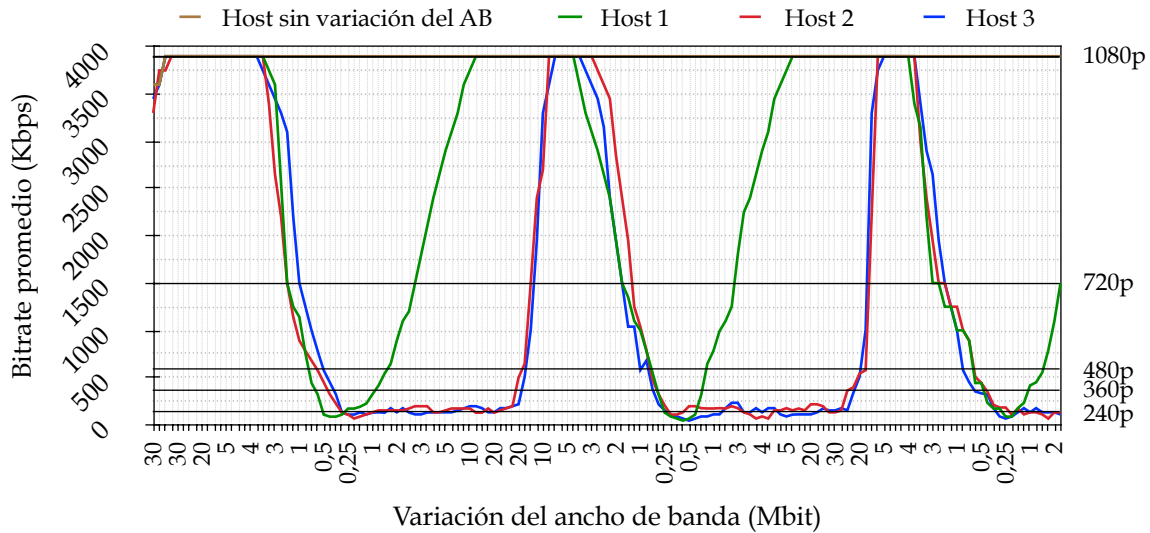


Figura 47. Variación del Bitrate en función del ancho de banda disponible

Podemos observar en el segundo escenario, que a pesar de disponer más ancho de banda, luego de haber llegado hasta los 250 Kbits y luego ir incrementando hasta los 30 Mbits, al momento de tener más de 20 Mbits, el servidor aún entrega calidad en SD. Esto es debido a que los buffers no puede recibir los paquetes de mayor calidad hasta que se hayan despachado los que se encuentran encolados, y que tienen una calidad SD. Al pasar los 10 Mbits, la calidad del video mejora a HD y Full HD debido a que ya ha encolado suficientes paquetes de mejor calidad. En el caso de un solo Host esto no sucede debido a que al ser el enlace para un único usuario la mayor parte de tiempo el buffer recibirá chunks de calidad HD y Full HD, cuando el ancho de banda decrece a menos de 1 Mbit el buffer empezará a llenarse de chunks SD pero dado que tiene aún contenido en HD y Full HD en el buffer, cuando vuelve a aumentar el ancho de banda volverá a ofrecer video de alta calidad, y solamente reproducirá video SD unos pocos segundos, menos de un minuto en el peor de los casos; mientras que en el segundo escenario se reproducirá contenido SD, aproximadamente dos minutos consecutivos cuando decae el ancho de banda.

En la Figura 48 podemos observar como se llenan y vacían los buffers en función del ancho de banda disponible en el enlace, en este caso hemos tomado de referencia una única reproducción para apreciar las diferencias entre el host 2 y el host 3. Si tomamos el promedio de las 10 medidas las dos gráficas son muy similares.

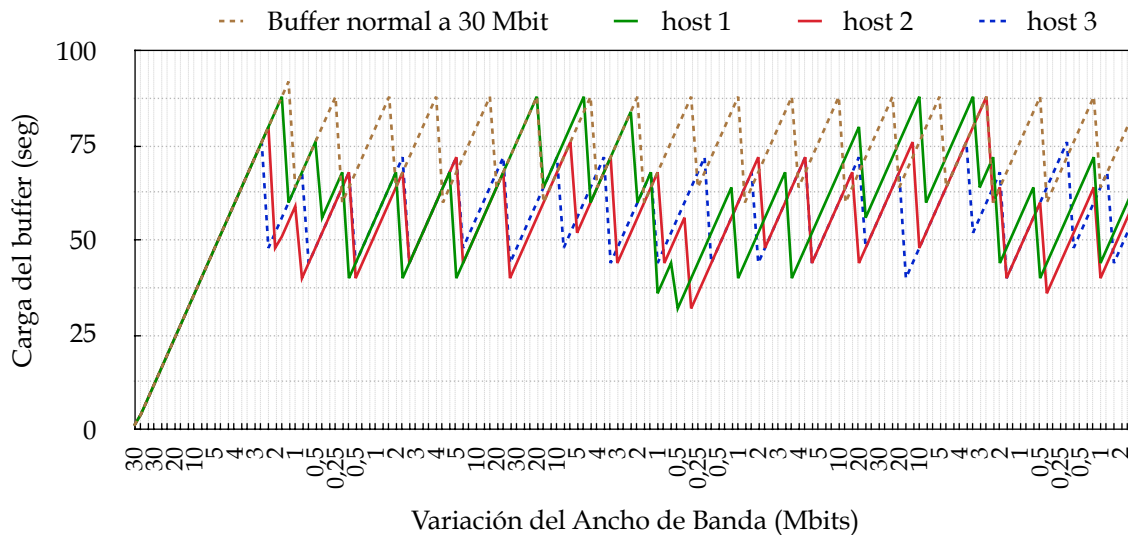


Figura 48. Carga del buffer en función del ancho de banda disponible

En condiciones normales, el buffer se carga con valores máximos de 88 segundos de contenido, y llegando a valores mínimos de 60 segundos. Pero cuando hay variaciones en el ancho de banda la cosa cambia. En el primer escenario vemos que empieza a cargarse igual que en el caso ideal hasta que empieza a haber problemas en el ancho de banda, llegando a tener un mínimo de 40 segundos en su buffer en el peor de los casos y un máximo de 88 segundos cuando cuenta con todo el ancho de banda. Para el Host 2 llega a tener un buffer mínimo cuando el ancho de banda es de 250 Kbits luego irá almacenando paquetes pero, como hemos dicho antes, serán de calidad SD dado que el ancho de banda no permite reproducir video de un bitrate superior. Una situación similar sucede con el Host 3, aunque en este caso, al aumentar el ancho de banda se carga, hay ocasiones que en un Host se ha cargado mejor calidad de video que en otro. En el segundo escenario, el buffer como máximo llega a tener 76 segundos como máximo en el mejor de los casos.

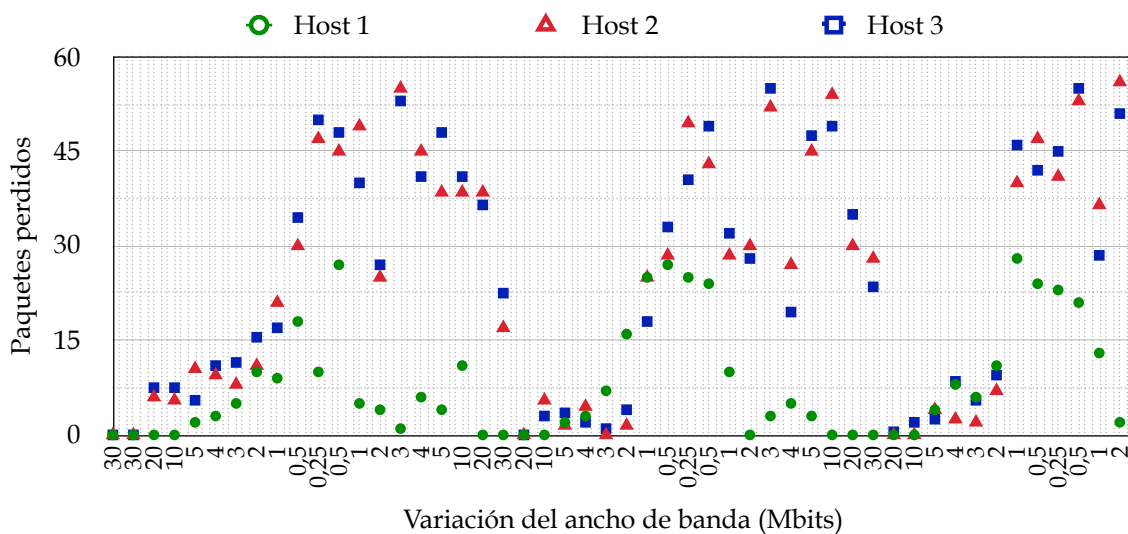


Figura 49. Paquetes perdidos o descartados por los Hosts

Finalmente, en la Figura 49 podemos observar los paquetes descartados o perdidos debido a la variación del ancho de banda. Es así que el servidor envía los paquetes de cierta calidad y cuando el buffer no puede almacenarlos son descartados y el servidor enviará paquetes de menor calidad para mantener el buffer con datos, permitiendo que la reproducción no se detenga, aunque para ello tenga que disminuir la calidad de la imagen.

Con este caso práctico hemos verificado como la limitación del ancho de banda para un servicio de streaming de contenido multimedia afecta la QoS y QoE. En el caso de Netflix ante la queja de muchos usuarios que no podían disfrutar de contenido UHD, publicó el ancho de banda que ofrecen los ISPs en las horas de más audiencia. En el caso de España no llegan a los 4 Mbps, visto en el Capítulo 5, lo cual hace imposible la reproducción de contenido en UHD.

## 7. Líneas futuras y Conclusiones

### 7.1. Líneas futuras

A lo largo del trabajo hemos analizado el estado actual de todos los componentes que conforman el ecosistema del video streaming, desde la generación del mismo hasta la entrega al usuario final y sus dispositivos.

Dentro de cada componente del ecosistema del video streaming se están realizando investigaciones y estudios con el objetivo de mejorar el rendimiento de la infraestructura y poder optimizar los recursos de la misma. Permitiendo así que las plataformas y proveedores de contenidos puedan ofrecer nuevos servicios a los usuarios y mejorar los ya existentes; mejorando a la vez, la QoS y QoE. Estos dos parámetros han tomado mucha importancia en los últimos años, sobretodo la QoE. Tanto es así que en [59] se realiza un estudio de la forma de estimar la QoE a partir de la QoS de la infraestructura; QoE es un parámetro complejo de estimar ya que es un valor subjetivo del usuario.

- **Codificación y compresión del contenido**

En el campo de la codificación y compresión del contenido, tal como lo dijimos en el Capítulo 3, Google se encuentra desarrollando el nuevo codec VP10 para la compresión de contenido multimedia de UHD para la reproducción en sus plataformas y que se supone que estará listo para finales del 2016. Por otro lado, el grupo Alliance for Open Media, fundado en el 2015 y formado por varias empresas tecnológicas de Internet, se encuentran desarrollando un codec open source que sea el sucesor de VP9 y una alternativa a HEVC. Actualmente se encuentra en fase de desarrollo pero han anunciado que tienen previsto que el nuevo codec vea la luz en Enero del 2017. [13]

- **Distribución del contenido**

Con respecto a la distribución de contenido, también se están realizando avances y se han ofrecido soluciones para mejorar la QoS y QoE del usuario final. En esta sección se presentarán unas cuantas investigaciones y estudios que considero importantes. Por ejemplo, en [7] se realiza un estudio de las plataformas de Netflix y Hulu; en donde se propone un algoritmo mediante el cual se utilizarían las tres CDNs simultáneamente (Akamai, Limelight y Level 3) con el objetivo de que el usuario final siempre disfrute del contenido multimedia en alta calidad.

Por otro lado, también se proponen el uso de redes Information - Centric Networks (ICN) para la entrega y replicación de contenido en las CDNs; el objetivo es mejorar la QoS [3], [45]. Se demostró que mediante la ICN se disminuyó el tráfico un 22% y la carga en los servidores un 34%. La diferencia entre las CDNs y las ICNs es que las CDNs realizan la entrega de contenido extremo a extremo a nivel de aplicación, mientras que las ICNs se centran en la información que necesita el usuario, sin depender de

la localización física del mismo; el contenido se encuentra referenciado por un identificador. Una ICN provee un servicio similar a un modelo híbrido P2P-CDN. De igual forma, han surgido otras propuestas de redes, como las Content - Centric Networking (CCN), las cuales presentan la característica principal de que el contenido puede ser cacheado en cualquier lugar a lo largo de camino, lo que permitiría que otros usuarios puedan obtener el contenido directamente de las caches presentes en routers intermedios. En [44] se realiza un estudio de la transmisión de streaming adaptativo DASH sobre una red CCN. Las CCNs identifican el contenido por su nombre con lo cual el contenido puede estar almacenado donde sea.

La aplicación de dispositivos inteligentes para la entrega de contenido también es analizado, es así que en [43] se estudia a la plataforma de Youku, un proveedor de contenido multimedia, que junto con ChinaCache, una CDN, utilizan smart routers para asistir la entrega del contenido de video streaming mediante un modelo Peer CDN. Youku ha desplegado más de 300.000 smart routers en los hogares de sus clientes. Esto permite disponer de un monitor efectivo de QoS, el cual es usado para mejorar la redirección de peticiones de los usuarios hacia los mejores servidores de la CDN, mejorando así la QoS y QoE.

Otras investigaciones realizadas proponen el uso de algoritmos que permiten mejorar la calidad del video UHD y reducir la pérdida de paquetes mediante la implementación de mecanismos que controlen la congestión en la red [38]. También la comparación del ancho de banda entre los usuarios para disfrutar de video streaming en UHD [40]. La optimización de los mecanismos de encaminamiento de solicitudes para la entrega de contenido en las CDNs mediante el uso de tablas de enrutamiento que se encuentran en los nodos de la CDN y en las cuales, las solicitudes del cliente son tratadas bajo dos criterios; el primero es la localización del usuario y el segundo criterio es el tipo de contenido [52]. También se propone la sincronización de los servidores sustitutos de una red CDN mediante el uso de redes P2P; en [51] se propone este modelo con el objetivo de reducir el retardo y no sincronización de las transmisiones en directo o Live streaming.

Habíamos dicho que dentro de las mejoras de la eficiencia, escalabilidad y gestión de la infraestructura se está haciendo hincapié en la utilización de las redes definidas por software (SDN) y la virtualización de funciones de red (NFV). Por ejemplo, en [53] se utiliza las SDNs para el balanceo de la carga de las solicitudes de video streaming entre diferentes servidores multimedia. En [39] se utilizan las SDNs para optimizar la entrega del contenido mediante un algoritmo que mejora la fiabilidad de la reproducción del video mientras se reduce la utilización de los recursos de los servidores. Por último, como hablamos dicho en el Capítulo 6, las SDNs cumplen un role muy importante en la implementación de las redes 5G que se prevé que se empiecen a desplegar para el año 2020. Estas redes se estima que ofrecerán 1 Gbps para distancias



de hasta 2 kilómetros. En [69] se realiza un estudio de como se podría realizar el control de QoE desde el plano de control de la SDN y su aplicación en las futuras redes 5G.

- **Protocolos**

En cuanto a los protocolos de comunicación ha habido muchos avances, desde los protocolos basados en HTTP como DASH, hasta protocolos como FLUTE que permiten mejorar la transmisión en redes móviles. Habíamos hablado de las ventajas que presenta el protocolo HTTP/2 el cual mejora el tiempo de respuesta de un sitio Web. En [58] se realiza un estudio de la transmisión de VoD con DASH sobre HTTP/2 en donde se demuestra la ventajas de utilizar HTTP/2; esto permitiría mejorar la latencia del sistema y mejorar la QoS.

Un grupo liderado por Organizaciones japonesas, coreanas y chinas han propuesto un nuevo protocolo llamado MPEG Media Transport (MMT) [41] como el sucesor de los protocolos de streaming actuales. Los protocolos actuales no estaban pensados para dar soporte contenido de UHD, por ejemplo, HTML esta diseñado para la entrega de texto e imágenes, no video; RTP no provee QoS, además el uso de RTCP añade complejidad, MPEG Transport Stream no tiene control de errores, además no ofrece QoS y los paquetes son muy pequeños 188 bytes. MTT soluciona las deficiencias de los protocolos actuales en la entrega de contenidos UHD para plataformas como redes móviles, televisión digital y transmisiones vía satélite. Provee QoS, es compatible con TCP/UDP sobre IPv4/IPv6 y además da soporte completo a contenido UHD. [34] [35]

En [36] se presenta al protocolo sucesor de FLUTE, llamado Real-time object delivery over unidirectional transport (ROUTE), que mediante el cual se puede realizar la entrega de contenido mediante DASH al igual que FLUTE.

- **Cachés**

Las caches son muy importantes a la hora de disminuir la carga en los servidores, ya sea de los principales o delegados, sobretodo en horas donde hay gran cantidad de usuarios que solicitan contenido al mismo tiempo, ya sea debido a la transmisión de un evento en vivo o por el estreno de alguna serie y evitar el fenómeno de flash crowd . En este sentido, en [66] se propone la utilización de una caché en el equipo del usuario final "CPE Caching", en donde el contenido puede ser cacheado mediante dos modelos predictivos y participativos. Por otro lado, en [47] se propone el concepto de Cache as a Service. Este modelo se denomina OpenCache el cual es un servicio de caching para el streaming de VoD en la última milla. Es decir, es un servicio dirigido a los ISPs en los cuales el contenido de interés puede ser cacheado. OpenCache hace uso de las SDNs para mejorar la utilización de la res e incrementar la QoE del usuario final.

Como podemos ver, en todos los campos se están haciendo grandes esfuerzos para optimizar lo recursos existentes en el ecosistema del video streaming sobre Internet.

Como habíamos visto en el capítulo 5, las redes de acceso también cuentan con mejoras, desde la implementación cada vez más de HFC y FTTH reemplazando al xDSL, los nuevos estándares WiFi, el uso de LTE en las redes móviles y el futuro 5G.

En cuanto a los dispositivos finales, vemos que cada poco tiempo salen al mercado nuevos dispositivos con más funciones multimedia, y que se ha apostado en gran medida por los dispositivos móviles como smartphones y Tablets. El mundo multimedia sobre Internet ha llegado para quedarse y todos los datos apuntan a que seguirá evolucionando en Internet, ofreciendo nuevos servicios y mejorando los ya existentes.

## 7.2. Conclusiones

A lo largo del trabajo hemos visto como se realiza la generación, distribución y entrega de contenido multimedia sobre Internet. Hoy en día disponemos con servicios de contenido multimedia en UHD, VR, 360 grados, 3D, etc. Pero todo ello está suponiendo muchos retos tanto para la generación del contenido como para la visualización del mismo en los dispositivos finales. El uso de nuevos codecs como HEVC permiten mejorar la calidad y a la vez reducir el ancho de banda utilizado, pero aún no se encuentra consolidado dentro del mercado; se estima que será más eficiente en los próximos 5 años.

Por otro lado, en cuanto a la distribución y almacenamiento del contenido, hemos visto que también supondrá un reto en los próximos años, debido principalmente a la gran cantidad de información que se necesita almacenar, esto obligará al despliegue de más centros de datos que permitan almacenar la información y procesar los datos. Es por ello que las mejoras en cuanto a la compresión del contenido ayudará a paliar este problema en cierta medida, pero habrán otros retos como la eficiencia energética debido al consumo eléctrico de los centros de datos.

El uso de NFV y SDNs permitirán mejorar la eficiencia, rendimiento y utilización de las redes e infraestructuras, sobretodos en los grandes centros de datos y permitiendo mejorar la QoS y QoE de los usuarios.

El problema principal para el video streaming UHD sobre las plataformas de distribución de contenidos en Internet sigue siendo por motivos económicos. Las redes de accesos a Internet aún siguen siendo un cuello de botella. Como vimos en el Capítulo 5, muchos ISPs limitan el ancho de banda a ciertos servicios multimedia lo que imposibilita disfrutar de contenido en UHD en óptimas condiciones. Además, debido a los acuerdos comerciales, muchos proveedores de contenidos hacen uso de CDNs que les resulta más económico su uso o que simplemente tengan exclusividad y que no opten por otras CDNs que puedan prestar mejor servicio.

Por último, en cuanto a los dispositivos finales, los usuarios parece que han perdido el interés en actualizar sus dispositivos, ya sea por aspectos económicos y por la falta

de oferta de los contenidos en UHD. La gran mayoría de contenido UHD se oferta a través de Internet, como por ejemplo Youtube, Netflix, Vimeo, etc. Pero las demás plataformas como la TDT, en el caso de España, con el estándar DVB-T no ofrecen contenido en UHD aún.

Podemos concluir que el campo del video streaming aún no ha tocado techo y que en los próximos años seremos testigos de la optimización de los servicios, disminución de costes de suscripción y aumento de la oferta de contenido en UHD, sobretodo en Internet. Aún queda mucho trabajo por realizar.

## Anexos

### ✓Index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Reproductor DASH</title>
    <style>
      video {
        width: 640px;
        height: 360px;
      }
    </style>
  </head>
  <body>
    <div>
      <video id="videoPlayer" controls></video>
    </div>
    <script src="test/dash.all.min.js"></script>
    <script>
      (function(){
        var url = "http://10.0.3.200/BigBuckBunny/BigBuckBunny_4s_simple.mpd";
        var player = dashjs.MediaPlayer().create();
        player.initialize(document.querySelector("#videoPlayer"), url, true);
      })();
    </script>
  </body>
</html>
```

### ✓videostreaming\_net.py

```
#!/usr/bin/python
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSController
from mininet.node import CPULimitedHost, Host, Node
from mininet.node import OVSKernelSwitch, UserSwitch
from mininet.node import Node
from mininet.log import setLogLevel, info
from mininet.cli import CLI
from mininet.link import TCLink
from mininet.util import dumpNodeConnections
```

```

class LinuxRouter( Node ):
    "A Node with IP forwarding enabled."
    def config( self, **params ):
        super( LinuxRouter, self ).config( **params )
        # Enable forwarding on the router
        self.cmd( 'sysctl net.ipv4.ip_forward=1' )
    def terminate( self ):
        self.cmd( 'sysctl net.ipv4.ip_forward=0' )
        super( LinuxRouter, self ).terminate()

class NetworkTopo( Topo ):
    "A LinuxRouter connecting three IP subnets"
    def build( net, **_opts ):

        ## DEFINIMOS LOS ROUTERS
        info( '*** Definimos los Routers\n' )
        r1 = net.addHost( 'r1', cls=LinuxRouter, ip='10.0.1.1/24' )
        r2 = net.addHost( 'r2', cls=LinuxRouter, ip='10.0.3.1/24' )
        r3 = net.addHost( 'r3', cls=LinuxRouter, ip='10.0.5.1/24' )

        ## DEFINIMOS LOS SWITCHES
        info( '*** Definimos los Switches\n' )
        s1 = net.addSwitch( 's1', listenPort=6673 )
        s2 = net.addSwitch( 's2', listenPort=6674 )
        s3 = net.addSwitch( 's3', listenPort=6675 )

        ## DEFINIMOS LOS HOSTS
        info( '*** Definimos los hosts\n' )
        h1 = net.addHost( 'h1', ip='10.0.1.10/24', mac='00:00:00:00:00:aa',
                          defaultRoute='via 10.0.1.1' )
        h2 = net.addHost( 'h2', ip='10.0.5.10/24', mac='00:00:00:00:00:bb',
                          defaultRoute='via 10.0.5.1' )
        h3 = net.addHost( 'h3', ip='10.0.5.20/24', mac='00:00:00:00:00:cc',
                          defaultRoute='via 10.0.5.1' )

        ## DEFINIMOS LOS SERVIDORES
        info( '*** Definimos los servidores\n' )
        server = net.addHost( 'server', ip='10.0.3.200/24',
                              mac='00:00:00:00:00:ee', defaultRoute='via 10.0.3.1' )

        ## DEFINIMOS LOS ENLACES
        info( '*** Definimos los enlaces\n' )
        net.addLink( s1, r1, intfName2='r1-eth1', params2={'ip':'10.0.1.1/24'} )
        net.addLink( s2, r2, intfName2='r2-eth1', params2={'ip':'10.0.3.1/24'} )

```

```

net.addLink(s3, r3, intfName2='r3-eth1', params2={'ip':'10.0.5.1/24'})
net.addLink(r1, r2, intfName1='r1-eth2', params1={'ip':'10.0.2.2/24'},
            intfName2='r2-eth2', params2={'ip':'10.0.2.1/24'})
net.addLink(r2, r3, intfName1='r2-eth3', params1={'ip':'10.0.4.1/24'},
            intfName2='r3-eth2', params2={'ip':'10.0.4.2/24'})

for h, s in [(h1,s1),(server,s2),(h2,s3),(h3,s3)]:
    net.addLink (h, s)

## INICIO DE LA CONFIGURACION
def run():
    "Test linux router"
    topo = NetworkTopo()
    net = Mininet( topo=topo, controller=RemoteController,
switch=OVSKernelSwitch)
    net.start()

## DEFINIMOS LAS TABLAS DE ENRUTAMIENTO DE CADA ROUTER
net[ 'r1' ].cmd( 'route add -net 10.0.3.0/24 gw 10.0.2.1 r1-eth2' )
net[ 'r1' ].cmd( 'route add -net 10.0.4.0/24 gw 10.0.2.1 r1-eth2' )
net[ 'r1' ].cmd( 'route add -net 10.0.5.0/24 gw 10.0.2.1 r1-eth2' )

net[ 'r2' ].cmd( 'route add -net 10.0.1.0/24 gw 10.0.2.2 r2-eth2' )
net[ 'r2' ].cmd( 'route add -net 10.0.5.0/24 gw 10.0.4.2 r2-eth3' )

net[ 'r3' ].cmd( 'route add -net 10.0.1.0/24 gw 10.0.4.1 r3-eth2' )
net[ 'r3' ].cmd( 'route add -net 10.0.3.0/24 gw 10.0.4.1 r3-eth2' )
net[ 'r3' ].cmd( 'route add -net 10.0.2.0/24 gw 10.0.4.1 r3-eth2' )

CLI( net )
net.stop()
if __name__ == '__main__':
    setLogLevel( 'info' )
    run()

```

## ✓bw\_1h.sh

```
#!/bin/bash
clear
RATEDOWN=0
RATEUP=0
echo "***** Ancho de banda disponible 30Mbit en:" $(date +"%H:%M:%S")
sudo tc qdisc add dev s1-eth1 root handle 1: tbf rate 30Mbit burst 2000
latency 60ms
sudo tc qdisc add dev s1-eth1 parent 1:1 handle 10: netem delay 17ms 3ms 15%
sudo tc qdisc add dev s1-eth2 root handle 1: tbf rate 30Mbit burst 2000
latency 60ms
sudo tc qdisc add dev s1-eth2 parent 1:1 handle 10: netem delay 17ms 3ms 15%
sleep 12
sudo tc qdisc del dev s1-eth1 root
sudo tc qdisc del dev s1-eth2 root
for i in $(seq 0 2)
do
for RATEDOWN in 20Mbit 10Mbit 5Mbit 4Mbit 3Mbit 2Mbit 1Mbit 500kbit 250kbit
do
echo "***** Ancho de banda disponible $RATEDOWN Mbit en:" $(date +"%H:
%M:%S")
for INTERFACE in s1-eth1 s1-eth2
do
sudo tc qdisc add dev $INTERFACE root handle 1: tbf rate $RATEDOWN
burst 2000 latency 60ms
sudo tc qdisc add dev $INTERFACE parent 1:1 handle 10: netem delay 17ms
3ms 15%
done
sleep 12
for INTERFACE in s1-eth1 s1-eth2
do
sudo tc qdisc del dev $INTERFACE root
done
done
for RATEUP in 500kbit 1Mbit 2Mbit 3Mbit 4Mbit 5Mbit 10Mbit 20Mbit
30Mbit
do
echo "***** Ancho de banda disponible $RATEUP Mbit en:" $(date +"%H:
%M:%S")
for INTERFACE in s1-eth1 s1-eth2
do
sudo tc qdisc add dev $INTERFACE root handle 1: tbf rate $RATEUP burst
2000 latency 60ms
```

```

    sudo tc qdisc add dev $INTERFACE parent 1:1 handle 10: netem delay 17ms
3ms 15%
    done
    sleep 12
    for INTERFACE in s1-eth1 s1-eth2
    do
    sudo tc qdisc del dev $INTERFACE root
    done
done
echo "***** Final del test"

```

### ✓bw\_2h.sh

```

#!/bin/bash
clear
RATEDOWN=0
RATEUP=0
echo "***** Ancho de banda disponible 30Mbit en:" $(date +"%H:%M:%S")
sudo tc qdisc add dev s3-eth1 root handle 1: tbf rate 30Mbit burst 2000
latency 60ms
sudo tc qdisc add dev s3-eth1 parent 1:1 handle 10: netem delay 17ms 3ms 15%
sudo tc qdisc add dev s3-eth2 root handle 1: tbf rate 30Mbit burst 2000
latency 60ms
sudo tc qdisc add dev s3-eth2 parent 1:1 handle 10: netem delay 17ms 3ms 15%
sudo tc qdisc add dev s3-eth3 root handle 1: tbf rate 30Mbit burst 2000
latency 60ms
sudo tc qdisc add dev s3-eth3 parent 1:1 handle 10: netem delay 17ms 3ms 15%
sleep 12
sudo tc qdisc del dev s3-eth1 root
sudo tc qdisc del dev s3-eth2 root
sudo tc qdisc del dev s3-eth3 root
for i in $(seq 0 2)
do
    for RATEDOWN in 20Mbit 10Mbit 5Mbit 4Mbit 3Mbit 2Mbit 1Mbit 500kbit
250kbit
    do
    echo "***** Ancho de banda disponible $RATEDOWN Mbit en:" $(date +"%H:
%M:%S")
        for INTERFACE in s3-eth1 s3-eth2 s3-eth3
        do
            sudo tc qdisc add dev $INTERFACE root handle 1: tbf rate
$RATEDOWN burst 2000 latency 60ms
            sudo tc qdisc add dev $INTERFACE parent 1:1 handle 10:
netem delay 17ms 3ms 15%

```



```

done
sleep 12
for INTERFACE in s3-eth1 s3-eth2 s3-eth3
do
    sudo tc qdisc del dev $INTERFACE root
done
done
for RATEUP in 500kbit 1Mbit 2Mbit 3Mbit 4Mbit 5Mbit 10Mbit 20Mbit
30Mbit
do
echo "***** Ancho de banda disponible $RATEUP Mbit en:" $(date +"%H:
%M:%S")
for INTERFACE in s3-eth1 s3-eth2 s3-eth3
do
    sudo tc qdisc add dev $INTERFACE root handle 1: tbf rate
$RATEUP burst 2000 latency 60ms
    sudo tc qdisc add dev $INTERFACE parent 1:1 handle 10:
inetem delay 17ms 3ms 15%
done
sleep 12
for INTERFACE in s3-eth1 s3-eth2 s3-eth3
do
    sudo tc qdisc del dev $INTERFACE root
done
done
done
echo "***** Final del test"

```

### ✓estadisticas\_1h.sh

```

#!/bin/bash
clear
echo "*****Inicio con 30Mbit"
sleep 8
for i in $(seq 0 149)
do
    echo "*****Estadisticas en" $(date +"%H:%M:%S")
    sudo tc -p -s -d qdisc show dev s1-eth2
    sudo tc -p -s -d qdisc show dev s1-eth2
    sleep 8
done

```

## ✓estadisticas\_2h.sh

```
#!/bin/bash
clear
echo "*****Inicio con 30Mbit"
sleep 8
for i in $(seq 0 149)
do
    echo "*****Estadisticas en" $(date +"%H:%M:%S")
    sudo tc -p -s -d qdisc show dev s3-eth2
    sudo tc -p -s -d qdisc show dev s3-eth3
    sleep 8
done
```

## Bibliografia

- [1] Cisco, "Cisco Visual Networking Index : Forecast and Methodology , 2014 - 2019", 2016.
- [2] Cisco, "Cisco Visual Networking Index : Global Mobile Data Traffic Forecast Update, 2015 - 2020", 2016.
- [3] M. Pathan, R. K. Sitaraman, D. Robinson: "Advanced Content Delivery, Streaming, and Cloud Services", IEEE computer society, WILEY, 2014.
- [4] N. Newman, "Journalism, media and technology predictions 2016", Reuters Institute, 2016.
- [5] MEC Global, "Fast Take On: State of Streaming Video", March 2015
- [6] Media Excel, "Keeping Live Streaming LIVE," no. June, pp. 76-77, 2015.
- [7] V. K. Adhikari, Y. Guo, F. Hao, and V. Hilt, "Measurement Study of Netflix , Hulu , and a Tale of Three CDNs," vol. 23, no. 6, pp. 1984- 1997, 2015.
- [8] C. Timmerer, A. Begen, "Over the Top Content Delivery: State of the Art and Challenges Ahead", IEEE International Conference on Multimedia and Expo, 2015.
- [9] R. S. Veerubhotla, A. Saxena, and S. Member, "A DRM Framework Towards Preventing Digital Piracy," pp. 1-6, 2011.
- [10] Pomelo, LLC Tech Memo, "Analysis of Netflix's security framework for 'Watch Instantly' service", March-April, 2009.
- [11] S. Ropert, "Ultra High Definition. Which network to deliver UHD content?", IDATE Research, December, 2014.
- [12] K. R Rao et al, "Video Coding Standards", ISBN: 978-94-007-6741-6, pp. 37-51, 2014.
- [13] Streaming Media Website, "The State of Video Codecs 2016", Available: [On-line](#) (Último acceso: Julio 2016).
- [14] M. Wang, P. P. Jayaraman, et al, "An overview of Cloud based Content Delivery Networks: Research Dimensions and state-of-the-art," pp. 1-26, 2015.
- [15] T. D. Monchamp, "ADAPTIVE BIT-RATE STREAMING. Minimizing End-User Buffer Times in Real-Time Video Delivery", vol. 9, no. 2, pp. 1-12, 2013.
- [16] S. Vermulm, "OTT / ABR Demystified and Monitored", Sencore, pp. 1-34, 2015.

- [17] S. Lederer, C. Müller, and C. Timmerer, "Dynamic Adaptive Streaming over HTTP Dataset," pp. 89–94, 2012.
- [18] I. Sodagar, "MPEG-DASH : The Standard for Multimedia Streaming Over Internet", Microsoft Corporation, 2011.
- [19] IHS Technology, "Current Market and Technology Trends in the Broadcasting Sector", May 2015.
- [20] A. Rao, D. Towsley, Y. Lim, C. Baraka et al, "Network Characteristics of Video Streaming Traffic", 2011.
- [21] S. Laine, I. Hakala, "H.264 QoS and Application Performance with Different Streaming Protocols", 2015.
- [22] A. Durresi, R Jain, "RTP, RTCP, and RTSP - Internet Protocols for Real-Time Multimedia Communication", 2005.
- [23] R. Belda, I. De Fez, F. Fraile, P. Arce, and J. C. Guerri, "Hybrid FLUTE/DASH Video Delivery over Mobile Wireless Networks", 2014.
- [24] A. Yamada, H. Matsuoka, and R. Kitahara, "Storage-based Broadcasting in Multimedia Broadcasting for Mobile Terminals", Vol. 9, No. 8, Aug. 2011.
- [25] A. K. Pathan and R. Buyya, "A Taxonomy and Survey of Content Delivery Networks", pp. 1–44, 2008.
- [26] Z. Lian-ying and Z. Xiao, "The research of VoD system performance based on CDN and P2P technologies", vol. 4, pp. 385–388, 2010.
- [27] K. Hareesh and D. H. Manjiah, "Peer-to-Peer Live Streaming and Video on Demand Design Issues and Its Challenges" vol. 2, no. 4, 2011.
- [28] C. M. Lentisco, "Técnicas de corrección de errores Streaming Multimedia en redes móviles", Transparencias del master MUIRST - UPM, asignatura ASAI, 2015- 2016.
- [29] M. Gupta and D. Kumar, "State-of-the-art of Content Delivery Network", ISSN: 0975-9646, vol. 5, pp. 5441–5446, 2014.
- [30] A. Begen, T. Akgul and M Baugher, "Watching Video over the Web. Part 1: Streaming Protocols", Cisco, pp. 54–63, 2011.
- [31] A. Begen, T. Akgul and M Baugher, "Watching Video over the Web. Part 2: Applications, Standardization, and Open Issues", Cisco, pp. 59–63, 2011.
- [32] F. Bronzino, R. Gaeta, M. Grangetto, and G. Pau, "A Adaptive Hybrid CDN/P2P Solution for Content Delivery Networks", 2012.

- [33] R. Kadlic, J. Blichar, and P. Podhradsky, "New Trends in Content Delivery Networks", IWSSIP 2014, pp. 155-158, May 2014.
- [34] A. Vetro, Y. Lim, K. Park, et al, "MMT: An Emerging MPEG Standard for Multimedia Delivery over the Internet", pp. 80-85, 2013.
- [35] T. Nakachi, T. Yamaguchi et al, "Next-generation Media Transport MMT for 4K/8K Video Transmission.", Vol. 12, No. 5, May 2014.
- [36] G. K. Walker, T. Stockhammer, et al, "ROUTE/DASH IP Streaming - Based System for Delivery of Broadcast, Broadband, and Hybrid Services," Vol. 62, No. 1, 2016.
- [37] J. J. Quinlan, A. H. Zahran and C. Sreenan, "Datasets for AVC (H.264) and HEVC (H.265) Evaluation of Dynamic Adaptive Streaming over HTTP ( DASH )", 2016.
- [38] S. Lee, H. Lee, and W. Ryu, "Congestion Control Scheme for UHD Video Streaming over the Internet", pp. 736-738, 2015.
- [39] A. Advani, C. Joseph, D. Parwani and S. Bhol, "Optimizing Video Delivery using OpenFlow," 2015.
- [40] I. Montes, J. Chua, M. Cruz, A. R. Ii, and J. D. Young, "Streaming 4K/UHD Video to the Rest of Us: Can Bandwidth Sharing Help ?", 2015.
- [41] B. Hankinson, "Streaming 4K/8K Video over IP Networks: Daunting Problems, Proposed Solutions" Streamonix Limited, 2013.
- [42] M. S. Seyfabad and B. Akbari, "Virtual Machine Allocation in P2P-Cloud Live Video Streaming", IEEE, 2015.
- [43] M. Ma, Z. Wang, K. Su, and L. Sun, "Understanding the Smartrouter-based Peer CDN for Video Streaming", May 2016.
- [44] Y. Liu, J. Geurts, S. Lederer et al, "Dynamic Adaptive Streaming over CCN : A Caching and Overhead Analysis," Next-Generation Networking Symposium, 2013.
- [45] D. Trossen, M. J. Reed, M. Georgiades et al, "IP Over ICN - The Better IP ? An Unusual Take on Information-Centric Networking," EuCNC, 2015.
- [46] C. V. Streaming, G. Zhang, W. Liu, X. Hei, and W. Cheng, "Unreeling Xunlei Kankan : Understanding Hybrid," vol. 17, no. 2, pp. 229-242, 2015.
- [47] P. Georgopoulos, et al, "Cache as a Service : Leveraging SDN to Efficiently and Transparently Support Video-on-Demand on the Last Mile", ICCCN 2014.
- [48] Z. Lu, Y. Wang, and Y. R. Yang, "An Analysis and Comparison of CDN-P2P-hybrid Content Delivery System and Model," vol. 7, no. 3, pp. 232-245, 2012.

- [49] M. Meskovic, et al, "Content Delivery Architectures for Live Video Streaming: Hybrid CDN-P2P as the best option", pp. 26-32, 2012.
- [50] S. Kang and H. Yin, "A hybrid CDN-P2P system for Video-on-Demand", pp. 309-313, 2010.
- [51] Y. Nam, C. Lee, S. Kang, and J. Park, "Synchronization among CDN Edge Servers using P2P Networking", pp. 466-468, 2015.
- [52] W. Benchaita, S. Ghamri-Doudane, and S. Tixeuil, "On the Optimization of Request Routing for Content Delivery", pp. 347-348, 2015.
- [53] P. A. L. Rego, M. S. Bonfim, et al, "An OpenFlow-based Elastic Solution for Cloud-CDN Video Streaming Service", 2015.
- [54] L. Ling, M. Xiaozhen and H. Yulan, "CDN Cloud: A Novel Scheme for Combining CDN and Cloud Computing", 2013.
- [55] F. Wang, J. Liu and M. Chen, "CALMS : Cloud-Assisted Live Media Streaming for Globalized Demands with Time/Region Diversities", 2012.
- [56] H. Nam, D. Calin, and H. Schulzrinne, "Intelligent Content Delivery over Wireless via SDN," 2014.
- [57] K. A. Noghani and M. O. Sunay, "Streaming Multicast Video over Software-Defined Networks.", 2014.
- [58] J. M. Batalla, P. Krawiec, et al, "On Providing Cloud-awareness to Client's DASH Application by Using DASH over HTTP/2", 2015.
- [59] Y. Chen, S. Member, K. Wu, and Q. Zhang, "From QoS to QoE: A Tutorial on Video Quality Assessment," pp. 1-41, 2014.
- [60] A. Vetro, A. Pande, V. Ahuja et al, "Video Delivery Challenges and Opportunities in 4G Networks," pp. 88-94, 2013.
- [61] A. Blair, G. Parr, P. Morrow, et al, "A Unified Architecture for Video Delivery Over the Internet.", 2011.
- [62] S. Levine, "How to Select your Broadband Technology", USDA, Available: [On-line](#) (Último acceso: Julio 2016)
- [63] M. D. Oskouei and S. N. Razavi, "A Study on WiMAX: IEEE 802.16 Standard", vol. 131, no. 8, pp. 22-27, 2015.
- [64] Akamai White Paper, "Streaming toward television's future: A detailed look at 4K video and how Akamai is making it a reality", 2015.

- [65] M. Yardley, C. Jones, and S. Montakhab, "New service developments in the broadcast sector and their implications for network infrastructure", 2014.
- [66] S. Loach, "CPE Caching - Using Network Inefficiency to Delivery Better Internet Streaming", 2015.
- [67] S. Taksande, K. Joshi, V. Chikaraddi, and S. Raksha, "Video Streaming Techniques and Issues," vol. 3, no. 1, pp. 1-4, 2015.
- [68] Linux-IP, "Traffic Control Manual" Available: [On-line](#), (Último acceso: Julio 2016)
- [69] O. Awobuluyi, J. Nightingale et al, "Video Quality in 5G Networks. Context-Aware QoE Management in the SDN Control Plane", 2015.
- [70] Unisphere Research, "Over-The-Top Video Delivey: Challenges and Opportunities for Global OTT Service Providers", May 2015.
- [71] Unisphere Research, "The OTT - Video Services Market: Today's Trends and What is Next for 4K, HDR, HFR & VR", May 2016.
- [72] T. Yuan, "Desarrollo de servicios de distribución de contenidos multimedia en redes corporativas con tecnología SDN", UPM, TFM 2015.
- [73] J. J. Quinlan, A. H. Zahran et al, "Delivery of Adaptive Bit Rate Video: Balancing Fairness, Efficiency and Quality," pp. 1-6, 2015.
- [74] White Paper, "Network Functions Virtualisation (NFV)", ETSI, Available: [On-line](#), (Último acceso: Julio 2016).