



INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS
AVANZADAS
PROTOCOLO
INGENIERÍA TELEMÁTICA



DTA-PPT-01

TÍTULO DEL PROTOCOLO			
Modelado matemático, basado en cadenas de Markov, para servicios de video en vivo soportados por redes híbridas P2P-CDN			
DATOS DEL PROTOCOLO			
Número de Revisión <small>(Primera, segunda, tercera o Protocolo para Registro)</small>	Segunda	Semestre	2022/1
Número Proyecto Asignado <small>(Número asignado por el profesor de Especialidad)</small>		Fecha <small>(Fecha programada)</small>	16/ Noviembre/ 2021
Confidencialidad <small>(Público o confidencial, incluir documento que lo avale)</small>	Público	Número de Hojas <small>(Cantidad de Hojas del Protocolo)</small>	47
Patrocinador <small>(En caso de existir, incluir el nombre en caso contrario dejar en blanco)</small>			
Número Convenio o Registro <small>(Incluir número de convenio patrocinio o número de proyecto de investigación que patrocina)</small>			

ALUMNO 1	
DATOS ALUMNO 1	
Nombre del Alumno	Muñoz Ruiz Ulises
Número de boleta	2014030937
Teléfono	5587659705
Correo electrónico	umunozr1300@alumno.ipn.mx ulisesmo47@gmail.com

ALUMNO 2	
DATOS ALUMNO 2	
Nombre del Alumno	Ortiz Islas José Manuel
Número de boleta	2014111020
Teléfono	5569300023
Correo electrónico	jortiz1300@alumno.ipm.mx manolortiz3@gmail.com

ALUMNO 3	
DATOS ALUMNO 3	
Nombre del Alumno	
Número de boleta	
Teléfono	
Correo electrónico	

DATOS ASESOR 1		VISTO BUENO ASESOR 1	
Nombre Asesor (Grado Académico)	Dr. Noé Torres Cruz		
Academia	Telemática	Interno	<input checked="" type="checkbox"/> X
Cédula Profesional (Obligatorio)	3844740		
Correo electrónico	ntorresc@ipn.mx noe_torrescruz@yahoo.com.mx		

DATOS ASESOR 2		VISTO BUENO ASESOR 2	
Nombre Asesor (Grado Académico)			
Academia	Interno	<input type="checkbox"/>	<input type="checkbox"/>
Cédula Profesional (Obligatorio)			
Correo electrónico			

DATOS ASESOR 3		VISTO BUENO ASESOR 3	
Nombre Asesor (Grado Académico)			
Academia	Interno	<input type="checkbox"/>	<input type="checkbox"/>
Cédula Profesional (Obligatorio)			
Correo electrónico			

NOMBRE DEL PROFESOR TITULAR	Jefe del Departamento de Tecnologías Avanzadas
NOMBRE DEL PROFESOR DE ESPECIALIDAD	Jefe del Departamento de Tecnologías Avanzadas

¹ En caso de Asesores Externos, deberá incluirse copia de su Cédula Profesional y Curriculum Vitae resumido en un archivo anexo al Protocolo.

AVISO DE PRIVACIDAD: Los datos recabados serán protegidos, incorporados y tratados por el Departamento de Tecnologías Avanzadas de la UPIITA, cuya finalidad es el uso de los mismos exclusivamente para el proceso de registro de protocolos y proyectos de trabajo terminal y proyecto terminal, para la titulación por opción curricular de las carreras de Ingeniería Biónica, Ingeniería Mecatrónica, Ingeniería Telemática e Ingeniería en Sistemas Automotrices. El responsable de los datos personales es el Departamento de Tecnologías Avanzadas, perteneciente a la Subdirección Académica de la Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas del IPN, donde el interesado podrá ejercer los derechos de acceso y corrección en la dirección: Av. Instituto Politécnico Nacional No. Col. Barrio la Laguna Ticomán, Gustavo A. Madero, México DF, CP. 07340, en el Edificio 1 primer piso. Lo anterior se informa en cumplimiento del Decimoséptimo de los Lineamientos de Protección de Datos Personales, publicados en el Diario Oficial de la Federación el 30 de septiembre de 2005.



**INSTITUTO POLITÉCNICO
NACIONAL**



UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS

Modelo matemático, basado en cadenas de Markov, para servicios de video en vivo soportados por redes híbridas P2P-CDN

Alumnos:

Muñoz Ruíz Ulises

Ortiz Islas José Manuel

Asesor:

Torrez Cruz Noé

Versión 2

Noviembre 2021

Resumen

Live streaming es la tecnología que permite difundir una señal de vídeo a través de Internet en tiempo real. Al utilizar esta tecnología, el usuario puede visualizar videos de acuerdo a sus preferencias. Sin embargo, en esta aplicación es difícil determinar la cantidad de recursos que la red debe asignar a cada usuario para mantener una Calidad de Servicio (QoS, por sus siglas en inglés) adecuada. Este problema es aún más complejo cuando se consideran implementaciones en las que los usuarios funcionan como *pares* (comparten entre ellos contenido previamente descargado) y a la vez tienen acceso al video mediante Redes de Distribución de Contenido (CDN, por sus siglas en inglés). Considerando lo anterior, en este proyecto se desarrollará un modelo matemático, basado en Cadenas de *Markov*, que describa el funcionamiento de la red y, por lo tanto, permita evaluar su desempeño en términos de parámetros de QoS, como el tiempo de descarga del video y la probabilidad de pausa en la reproducción. Además, este modelo se utilizará para evaluar parámetros de operación de la red, como el ancho de banda mínimo que debe proveer la CDN para garantizar la satisfacción de la QoS. Con base en los resultados obtenidos, también se propondrá un nuevo esquema de asignación de recursos, cuya eficiencia supere a esquemas actuales, en términos

de los parámetros antes mencionados. Este proyecto no contempla la implementación del esquema propuesto, pero sí su evaluación con el modelo desarrollado y considerando parámetros reales de operación de las redes actuales.

Palabras Clave: Cadenas de *Markov*, Red Híbrida *CDN-P2P*, Transmisión en Vivo, QoE.

Índice de Contenido

1.- Introducción.....	4
2.- Planteamiento del Problema.....	7
3.- Propuesta de solución.....	9
3.1 Alcances	15
4.- Objetivos.....	16
4.1.- Objetivo General.....	16
4.2 Objetivos Específicos	16
5.- Estado del Arte	17
5.1 Trabajos en donde los servicios de video se han modelado mediante cadenas de <i>Markov</i>	17
5.2 Trabajos en donde se menciona el comportamiento y arquitectura de los servicios de video en vivo.....	19
6.- Marco Teórico	24
6.1. Video en Vivo	24
6.1.1. Formatos de codificación	24
6.1.2. Formatos de empaquetamiento.....	25
6.1.3 Servicios de video en vivo.....	26
6.1.4. Protocolos de arquitectura y distribución de video en vivo por internet.....	28
6.2. Redes <i>CDN-P2P</i>	30
6.3. Servicios de video sobre redes híbridas <i>CDN-P2P</i>	33
7.- Escenario de Pruebas.....	34
8.- Cronograma de Actividades	36
9.- Referencias	44
Glosario de Términos.....	46

1.- Introducción

En la actualidad existen dos formas de transmitir video mediante los protocolos de Internet, *OTT* (*over the top*) e *IPTV* (*Internet Protocol Television*) [1]. Ambas tecnologías permiten distribuir contenido de video sobre la red, sin embargo, cada una de ellas lo realiza de manera distinta.

OTT tiene 2 enfoques: *VoD* (*Video on Demand*) y *live streaming*. En ambos casos se hace uso de servidores de Internet público para transmitir el contenido a todos los espectadores. Debido a que la calidad en la transmisión del video está sujeta a las condiciones normales de la red de internet del cliente, el archivo de video se divide en pequeños fragmentos con la finalidad de que el usuario los pueda recibir de manera cronológica, formar un *buffer* considerable y así poder reproducir el video sin complicaciones.

En *OTT* se habla de un servicio de 3 capas que garantizan una óptima experiencia de visualización de video sobre una red impredecible; la primera capa es la del video central, la segunda es la capa contenedora encargada de dividir el video en fragmentos y finalmente se presenta la capa de protocolo de transferencia basado en tiempo [1].



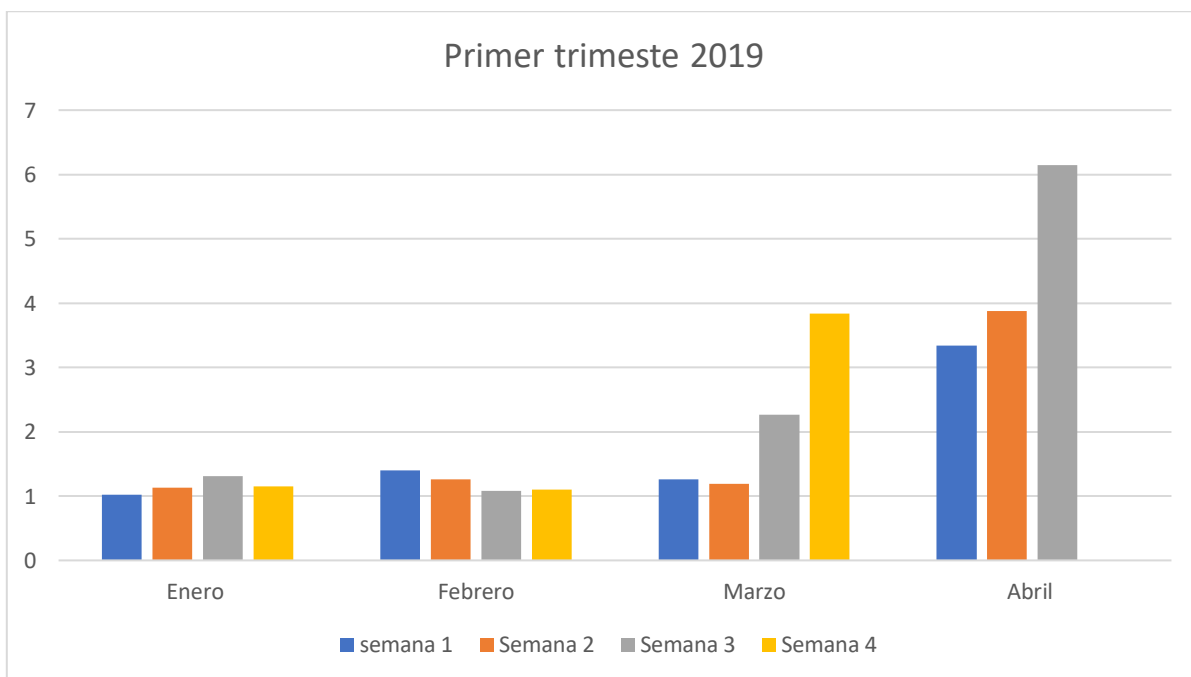
Figura 1. Capas de OTT

Por otro lado, *IPTV* está enfocado en hacer llegar el contenido disponible en Internet al sistema clásico de televisión, en cierto sentido se puede definir como un sistema híbrido de *OTT* y televisión por cable. Este tipo de transmisión emplea una red *CDN* dedicada para garantizar una buena calidad de servicio, donde el video no viaja directamente a un navegador, sino que va desde un ruteador a un decodificador que

permite su visualización en una TV. En este tipo de transmisión el proveedor de servicios es el responsable de proporcionar una *CDN* estable para no recurrir al servicio de proveedores terceros.

Una vez conocidas estas 2 formas de transmisión de video, el presente proyecto se enfoca en el servicio *OTT* dirigido a *live streaming*, con el propósito de desarrollar un modelo matemático que describa a este tipo de sistemas y por lo tanto, ayude a entender los fenómenos que afectan la calidad en el servicio. Además de lo anterior, también se plantea, como parte de este proyecto, definir un esquema de asignación de recursos que mejore el desempeño de propuestas publicadas con anterioridad.

La difusión de video en vivo, mejor conocido como *live streaming*, tiene como función principal permitir el acceso remoto a los usuarios que deseen conectarse a la red para acceder al contenido. Esto permite a los usuarios visualizar diversos eventos masivos a distancia (por ejemplo, conciertos, premiaciones, etc.), cuando su ubicación geográfica no le permite estar presente físicamente en dicho evento. Con el ritmo tecnológico actual, este servicio debe ser flexible y adaptativo frente a las nuevas generaciones de dispositivos y el creciente número de usuarios que desean ver contenido en vivo. En el tercer trimestre del año 2019 se observó un incremento del 91.8% del uso de las plataformas de live Streaming. En el siguiente grafico podemos observar el aumento en miles de personas referente al uso de la plataforma de twitch en la categoría de música en vivo.



Gráfica 1. Aumento de personas en la plataforma twitch

Hoy en día existe un vasto número de plataformas que ofrecen este tipo de servicio, por ejemplo, las redes sociales que permiten llevar a cabo la transmisión de video en vivo, en la tabla X se mencionan algunas plataformas de live Streaming; en ese sentido también se pueden encontrar diferentes arquitecturas de red que operan bajo las necesidades del *live streaming*.

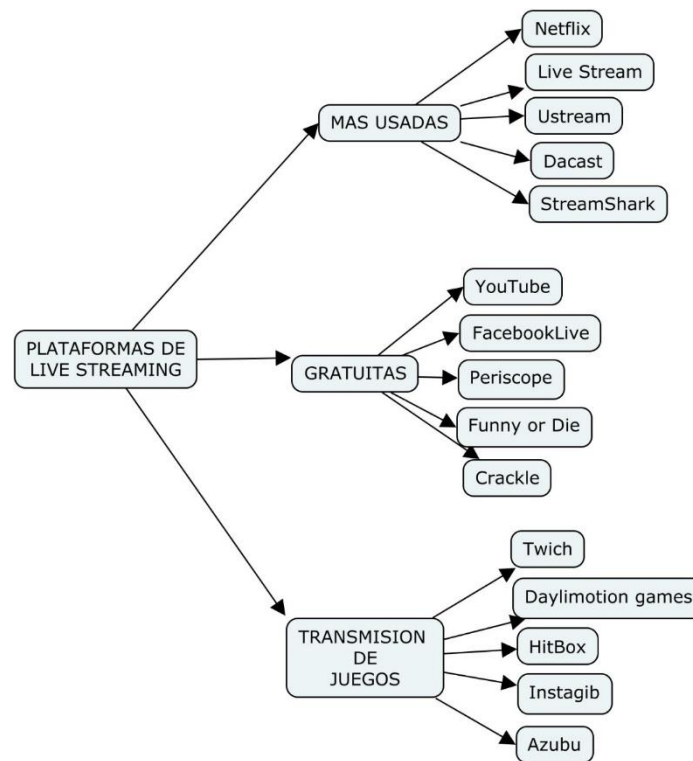


Figura 2. Clasificación de plataformas Live Streaming

Los servicios de live streaming pueden ser proveídos por redes CDN. La arquitectura *CDN* (*Content Delivery Network*) cuenta con un servidor principal donde se encuentra alojado el contenido, posteriormente este servidor distribuye el contenido a diversos servidores que se encuentran establecidos en otras partes geográficas, para que los usuarios acceden al contenido desde los diferentes servidores, evitando con esto la congestión en el servidor principal, por lo tanto, la *CDN* brinda estabilidad en la transmisión. Sin embargo, este tipo de red implica un costo elevado por el número de servidores involucrados.

Por otro lado, existe una red llamada *P2P* (*Peer to Peer*) en la cual los usuarios involucrados pueden operar como servidores y consumidores al mismo tiempo. En las redes *P2P* los usuarios son conocidos como *peers* ya que no solo descargan el contenido del archivo, sino que también están habilitados para compartirlo; este tipo de redes generan un costo menor debido a que no se emplean servidores fijos, sino que se aprovecha a cada uno de los *peers* por ende se logra una mayor escalabilidad.

La arquitectura híbrida *CDN-P2P* es una tecnología empleada para el servicio de transmisión de video debido a que ofrece la escalabilidad de las redes *P2P* y la estabilidad de una red *CDN*. Algunos trabajos de investigación han tomado a bien implementar y evaluar esta arquitectura con el fin de probar que son eficientes en la distribución de video a gran escala sobre Internet [2]. En [3] se hace hincapié que la arquitectura *CDN-P2P* ha sido propuesta para sistemas de transmisión, con la finalidad de alcanzar la escalabilidad de una red *P2P*, así como los pequeños retardos y alto rendimiento de una red *CDN*. Por ejemplo, la plataforma *LiveSky* emplea este tipo de arquitectura.

En consideración con lo descrito anteriormente, el desarrollo del proyecto se enfoca en emplear un sistema híbrido *CDN-P2P* para la transmisión de video en vivo, el cual brinda la estabilidad de una red *CDN* y al mismo tiempo la escalabilidad de una *P2P*.

El resto del documento se organiza de la siguiente forma: la sección 2 está orientada a exponer el problema que se va a abordar en el proyecto. En la sección 3, se da una propuesta de solución en conjunto con los alcances. Después en la sección 4 se plantean los objetivos tanto el general como los específicos. La sección 5 proporciona información acerca de los trabajos que se han desarrollado en torno al tema central. Enseguida la sección 6 está dedicada a proporcionar el marco teórico. La sección 7 establece el escenario bajo el cual se probará el proyecto. Posteriormente la sección 8 refleja las actividades a desarrollar. Por último, la sección 9 muestra las referencias utilizadas para desarrollar el proyecto.

2.- Planteamiento del Problema

Actualmente las tendencias apuntan al servicio de transmisión de video en vivo desde diversos dispositivos como un sector de gran importancia en la vida diaria. Derivado de la situación en la cual se encuentra la sociedad en los últimos días por la pandemia Sars-Cov2 (covid 19), el aforo a eventos masivos es limitado para evitar el riesgo de contagio, por lo cual la transmisión de contenido en vivo resulta favorable. Tal es el caso de premiaciones, partidos de fútbol, conciertos, concursos por mencionar algunos. Estos servicios hacen uso de la transmisión en vivo para que el contenido llegue a todos los usuarios que están interesados en este. De igual manera algunas aplicaciones como *Facebook Live*, *Instagram Stories*, *Periscope*, *WhatsApp*, *Snapchat*, etc. [4], resultan innovadoras y han crecido rápidamente gracias a que transmiten contenido de este estilo.

Sin embargo, cuando se presenta una transmisión de video en vivo desde una plataforma *streaming*, debido a la alta demanda del video por parte de los espectadores, se presenta sobrecarga de peticiones a los servidores donde se aloja temporalmente el contenido, lo que ocasiona el efecto cuello de botella en esta parte de la red. Esto afecta la velocidad de flujo y en consecuencia un retardo en la distribución del contenido, ya que no llega al mismo tiempo a todos los usuarios interesados en él. De igual forma, se afecta la experiencia de visualizar el contenido en tiempo real; para lo cual el uso de una red básica cliente servidor resulta insuficiente [5] ya que no proporciona un alcance adecuado para proveer el servicio a todos los clientes que consumen el video en vivo.

La variedad de aplicaciones multimedia en conjunto con el desarrollo acelerado de las comunicaciones móviles y la accesibilidad a múltiples dispositivos finales permiten emplear el ancho de banda del tráfico de datos presente en las redes subyacentes. [1]

En la literatura existen trabajos que han modelado servicio de video por medio de cadenas de *Markov*. Estos modelos han probado ser muy flexibles y al compararse con otros métodos de modelado han mostrado ser muy precisos. Desafortunadamente, estos desarrollos se han concentrado en servicios de video bajo demanda [5] [6] [7]. Hasta este punto de la investigación se conoce que el modelado con cadenas de *Markov* aún no ha sido aplicado a servicios de video en vivo.

Cabe mencionar que estos modelos utilizados para *VoD* no pueden ser aplicables directamente a *live streaming* debido a que:

En *VoD* los usuarios tienen poca sincronización entre ellos, puesto que el contenido ya se ha generado con anterioridad y los usuarios no esperan una experiencia en vivo. En cambio, en *live streaming*, los usuarios están altamente sincronizados. Como consecuencia, los procesos de arribos en cada caso son significativamente diferentes.

Otra consecuencia de la sincronización en *live streaming* es que los usuarios están interesados únicamente en fragmentos del video de lo que está ocurriendo en tiempo real, mientras que en *VoD* se pueden tener usuarios interesados en intervalos muy diversos del video. Por lo tanto, las poblaciones que pueden compartir recursos son muy diferentes en cada caso.

En *VoD* las tasas de descarga pueden ser significativamente más altas que las tasas de reproducción, porque el contenido ya ha sido almacenado. Lo anterior permite al usuario de *VoD* pre-almacenar gran parte del contenido. Por otro lado, en *live streaming*, este pre-almacenamiento intensivo no es factible, lo que, de nueva cuenta, afecta al modelo e incluso a los esquemas de compartición de contenido.

Debido a que, en la transmisión de video en vivo, el contenido se genera de manera simultánea a su visualización se pueden presentar diversos congelamientos en la reproducción en consecuencia se pierde la QoE de visualizar el contenido en tiempo real. Esto se atribuye a múltiples factores, por ejemplo, que no existan los *peers* suficientes para distribuir el contenido a los *peers* que se conecten posteriormente al sistema.

En este caso el servidor principal se ve obligado a proporcionar el contenido a todos los *peers* que arriben al sistema, sin embargo, este hecho puede incrementar el costo de la transmisión de video afectando inherentemente al usuario final. Por otro lado, si el video se pausa de forma continua, los usuarios pierden el interés en visualizarlo y por lo tanto dejan de seguirlo, esto genera pérdidas a las plataformas que distribuyen el contenido.

Ante este escenario se plantea la siguiente pregunta: ¿Cómo desarrollar un modelo, basado en cadenas de Markov, que represente el comportamiento de una población de *peers* con servicios de video en vivo y que además permita el diseño de nuevos esquemas de asignación de recursos?

3.- Propuesta de solución

En esta sección se describirán las características del servicio y del sistema que se van a modelar. Posteriormente se describe el comportamiento que presentan los *peers* dentro del sistema y su clasificación de acuerdo con la ventana que se encuentran descargando. Finalmente se describe una primera propuesta de la cadena de *Markov* que representa la cantidad de *peers* por ventana de video y se hace mención al diseño de un nuevo esquema de asignación de recursos.

Con este proyecto lo que se busca es modelar el servicio de video en vivo a través de una red híbrida *CDN-P2P* para conocer el comportamiento de los *peers*, una vez conectados al sistema y visualizando el contenido de una transmisión en vivo. El video que se trabajará en este modelo será el conjunto de las señales de audio y video empaquetadas en un solo archivo. Debido a que se desea que el usuario tenga la mayor calidad en cuanto al servicio y la experiencia en la visualización del contenido. Se propone dividir el archivo de video en fragmentos llamados ventanas, con el objeto de que el usuario reciba el video de manera paulatina, pueda crear un poco de *buffer* y reproducir el contenido de manera continua. A su vez cada ventana obtenida se fragmentará en elementos más pequeños denominados *chunks*.

Las ventanas están conformadas por N *chunks*, como se muestra en la figura 3, que son fragmentos aún más pequeños de video. Esta división se realiza debido a que

se plantea que un *peer* pueda descargar una ventana de múltiples *peers* habilitados para compartirle ese contenido, es decir, cuando un *peer* desee descargar una ventana, este podrá descargar fragmentos de esta desde diferentes *peers*, con esto se logra reducir el tiempo de descarga y posibles congelamientos en el video. En este trabajo se propone que la ventana c-d, que es la ventana con d retardos, es la última ventana permisible que se tiene para visualizar el video aún en tiempo real.

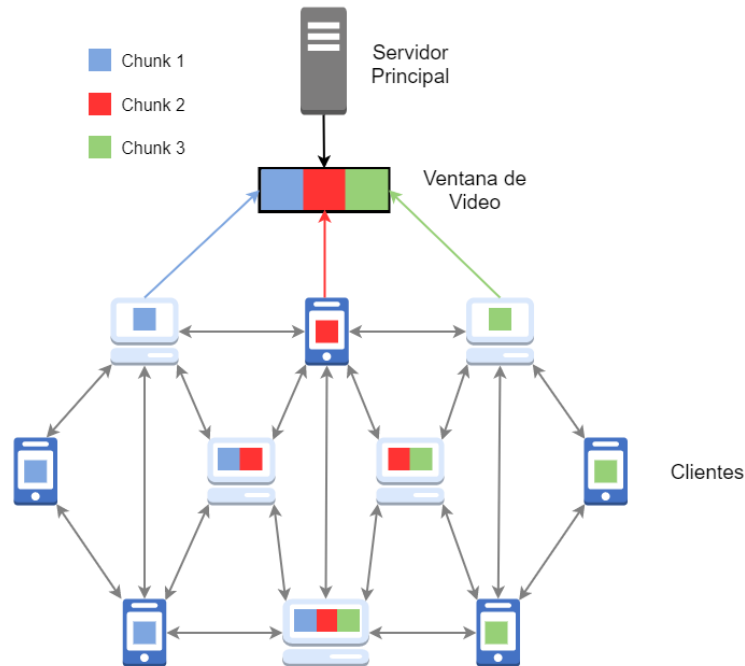


Figura 3. Fragmentación y reparto entre *peers* de Video

En el modelo que se propone para el desarrollo de este proyecto todos los usuarios recibirán el nombre de *peers*. Además, se supone que el sistema cuenta con un servidor central (*server*) y un *tracker* encargado de gestionar la señalización, es decir, cada uno de los *peers* le notifica la ventana que está descargando, para que este actualice la información referente a los estados de cada *peer*.

Aprovechando que la red *CDN-P2P* evita que se sature al servidor principal con peticiones de transmisión, el modelo que se propone para la simulación de eventos de los *peers* dentro del sistema dicta que una vez que un *peer* esté conectado, el *tracker* le notificará si existen *peers* habilitados para compartirle el contenido o si debe conectarse directamente al servidor. El modelo propuesto se muestra en la figura 4.

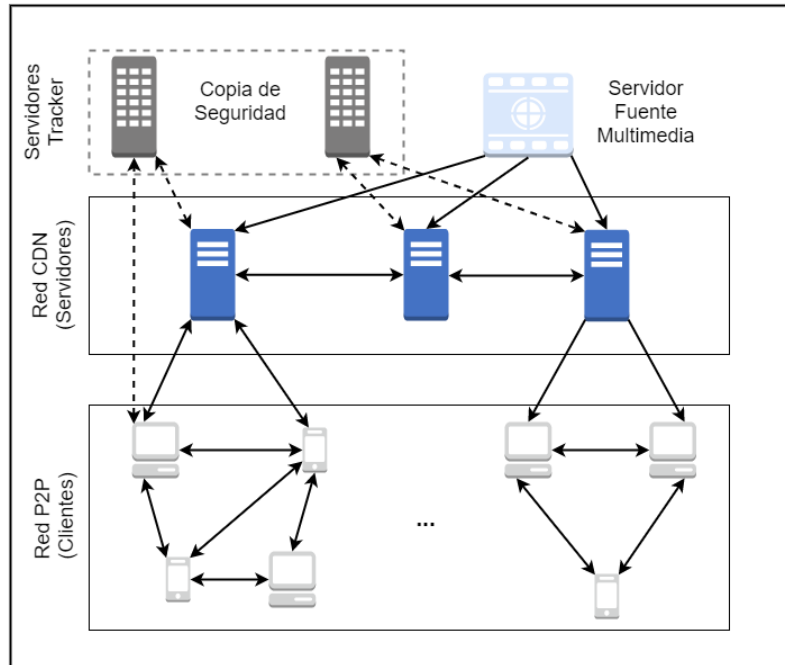


Figura 4. Modelo *CDN-P2P*

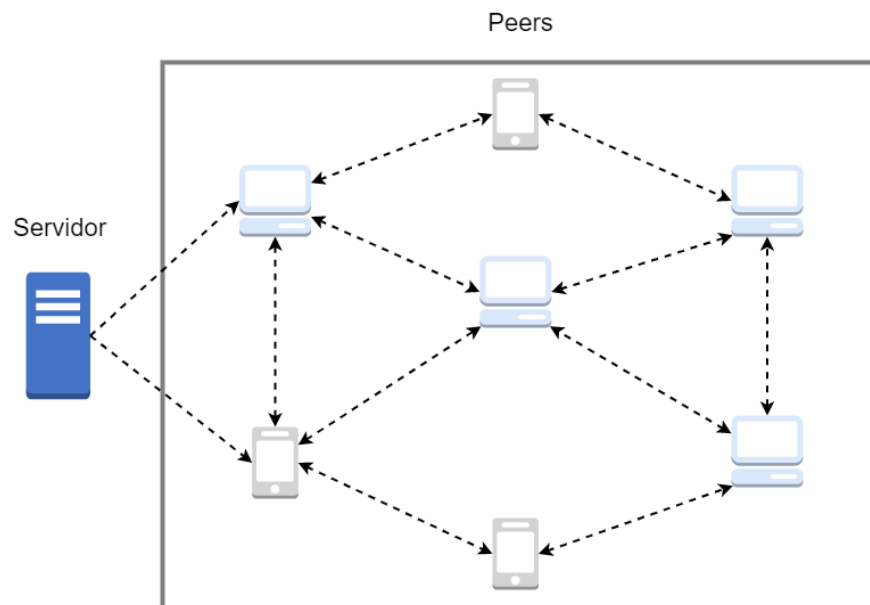


Figura 5. Arquitectura de Red *P2P*

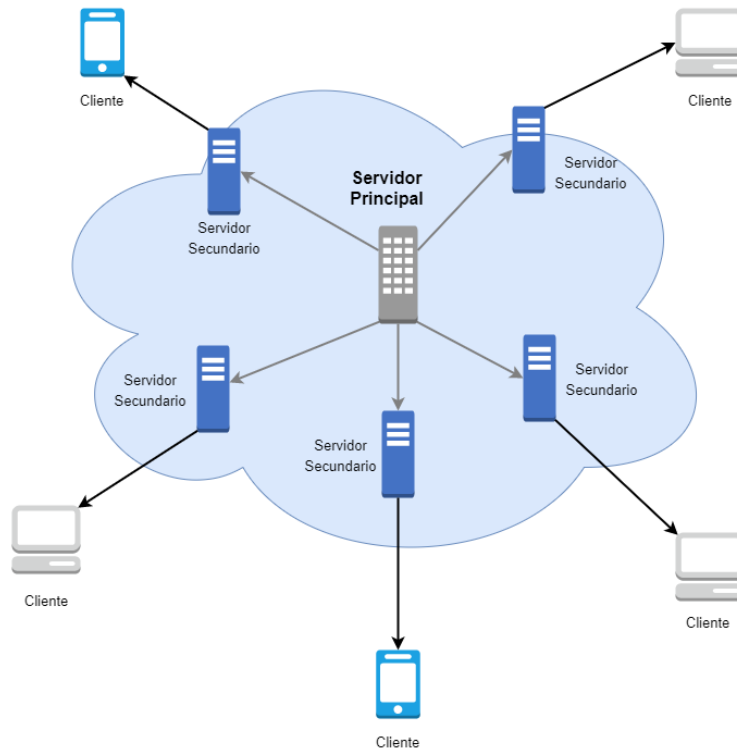


Figura 6. Arquitectura de Red *CDN*

Posteriormente cuando el *peer* esté haciendo uso de los recursos que le fueron asignados debe notificarle al *tracker* que ventana está descargando, para que de esta forma el *peer* sea usado como punto de referencia para que los *peers* en ventanas posteriores puedan obtener el contenido a través de él. Bajo esta estructura todos los *peers* que se encuentren conectados a la transmisión estarán habilitados para compartir el contenido que están consumiendo en ese momento. En consecuencia, se mantiene al servidor central más libre y el flujo de datos dentro de la red aumenta para todos los *peers* que necesitan el servicio de video en vivo.

El servidor distribuirá el contenido principalmente a los usuarios que estén en la ventana más reciente (c), y en caso de que los *peers* en esta ventana no sean los suficientes para proporcionar el contenido a *peers* en ventanas con n retardos (c-n), el servidor deberá proporcionarles las ventanas necesarias.

En el momento que inicie la transmisión de video en vivo n número de usuarios se conectarán de manera directa al servidor, este número de usuarios conformarán la población de *peers* que se encuentra en la ventana actual, denotada con X_c .

En este mismo sentido a la población de *peers* que se encuentre en la ventana i se denota como X_i , los cuales representan a los usuarios que se conectaron un lapso después del inicio de la transmisión del video en vivo. Esta población puede

proporcionar el contenido a usuarios en ventanas $i-n$ y lo obtienen de las poblaciones en ventanas $i+n$.

La población de *peers* que se encuentre en la ventana $c-d$ se denota como X_{c-d} , representa a los usuarios en la ventana con el mayor retardo permitido para conservar la experiencia de tiempo real. De esta forma todos los *peers* se clasifican de acuerdo con la ventana del video que se encuentran descargando.

La conexión de nuevos *peers* al sistema se realiza con la tasa λ . Los *peers* pueden permanecer dentro del sistema el tiempo que ellos deseen, el cual será medido por los tiempos de reproducción de una ventana denotado como t_p . Por distintas razones un usuario puede abandonar el sistema, y lo hará a tasa θ_i . Por otro lado, si las condiciones son favorables y el usuario está interesado en continuar visualizando el video, este se transferirá a las ventanas superiores a tasa τ_i , la cual depende directamente de la tasa de subida μ_i , proporcionada por los *peers* en ventanas superiores y por el servidor *CDN*. El tiempo que tarda un *peer* en descargar una ventana depende directamente de la tasa de subida y de su tasa de bajada en particular, denotado como C_i .

Con la finalidad de evitar congelamientos en la reproducción del video y ofrecer una mayor calidad en la experiencia (QoE) un *peer* solo puede reproducir una ventana del video hasta que haya terminado de descargarla por completo. Por ejemplo, si un *peer* se encuentra reproduciendo la ventana 0, de manera simultánea, deberá estar descargando la ventana 1, para así poder reproducir esta ventana al finalizar la visualización de la ventana 0.

El modelo se desarrollará con base en una cadena de *Markov* y un nuevo esquema de asignación de recursos propuesto en el presente proyecto. En la cadena de *Markov* cada uno de sus estados representa la población de *peers* en una ventana a un momento determinado. Con lo cual se puede observar cómo cambian las poblaciones de *peers*, denominadas estados, en cada una de las ventanas del video, dependiendo de los eventos que demanden los *peers*, en la duración de la transmisión. Los eventos más significativos dentro del sistema son: la llegada de un nuevo usuario al sistema una vez iniciada la transmisión en vivo, la llegada de un usuario en el transcurso de la transmisión, la desconexión de un usuario antes de finalizar la transmisión y la transferencia de un usuario a la ventana siguiente.

La evolución de la cadena se propone de manera preliminar como se muestra en la figura 7.

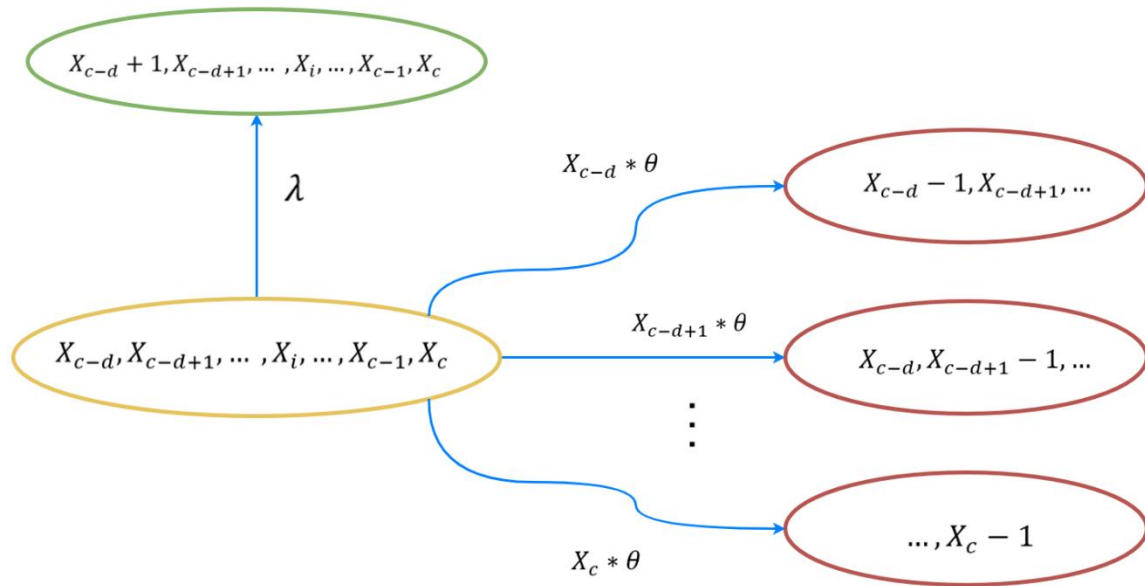


Figura 7. Evolución propuesta de la cadena de *Markov*

Donde

X_i es la población de peers en la ventana i

λ es la tasa de arribo de un peer al sistema

$X_i * \theta$ es la tasa a la que un peer abandona el sistema

Una vez que se obtenga la cadena de *Markov* que simule el comportamiento de las poblaciones de peers en las diversas ventanas que componen al video, se especificarán los métodos numéricos a emplear para llegar a su solución, en el sentido de obtener las poblaciones de peers promedio en una ventana dentro de un lapso determinado.

Finalmente, una vez obtenida la solución de la cadena de *Markov*, se desarrollarán expresiones que permitan obtener el ancho de banda de descarga (consumido por los peers durante el video), así como expresiones que nos permitan determinar el ancho de banda de subida proporcionado por los peers en ventanas superiores, es decir, el ancho de banda de subida proporcionado por la red *P2P*. En caso de ser insuficiente se deberá tomar el que estos proporcionan y solicitar el resto a la red *CDN*, este estado es llamado penuria. Por el contrario, si es suficiente no se solicita ancho de banda a la red *CDN* y entramos a un estado de abundancia, en el cual se establece al ancho de banda proporcionado por los servidores *CDN* como un ancho de banda adicional que garantiza la demanda de descarga que existe para el video.

De manera simultánea a las tareas descritas anteriormente se analizará la forma en la que se realiza la asignación de recursos a los diversos *peers* conectados al servicio en un esquema de asignación de recursos uniforme, además de investigar esquemas de asignación de recursos orientados a servicios de video en vivo sobre redes híbridas *CDN-P2P*.

Una vez investigados los diferentes esquemas de asignación de recursos existentes en redes *CDN-P2P*, se seleccionarán al menos dos de ellos adicionales al esquema de asignación de recursos tradicional uniforme para analizar su funcionalidad. Estos esquemas serán elegidos de acuerdo con la mayor similitud posible con el comportamiento del modelo a desarrollar en este proyecto. Partiendo de dicho análisis se diseñará un nuevo esquema que nos permita asignar los recursos existentes en una red *CDN-P2P* durante una transmisión de video en vivo de manera óptima para todos los *peers* participantes. Tanto la cadena de *Markov* como el nuevo esquema de asignación de recursos generan estadísticas de consumo y cooperación entre los diversos *peers* dentro del sistema. Dependiendo del nivel de funcionalidad que muestre el esquema propuesto se procederá a ajustarlo para que de esta forma pueda ser comparado con otros esquemas puestos ya en práctica. En el mismo sentido la cadena de *Markov* deberá ser replanteada para obtener mejores estadísticas de consumo y cooperación en el modelo.

La implementación y evaluación de este tipo de modelos en sistemas reales es muy compleja, debido a que estos requieren de una población significativa de usuarios para que los resultados obtenidos a partir de su análisis sean válidos. Por ende, esta fuera del alcance de un proyecto a realizar por dos estudiantes de nivel licenciatura a lo largo de un año. A pesar de lo descrito anteriormente, se plantea que los resultados del modelo sean válidos, debido a que estarán basados en parámetros (tasas de arribo, tasas de codificación, tasas de abandono, tasas de subida y de descarga, etc.) de sistemas reales reportados en la literatura.

3.1 Alcances

El proyecto estará limitado a la definición de la cadena de *Markov* bajo el esquema de asignación de recursos tradicional uniforme y posteriormente bajo el esquema de asignación de recursos propuesto en el presente proyecto.

El diseño del nuevo esquema de asignación de recursos se realizará considerando los parámetros obtenidos del análisis de esquemas de asignación de recursos empleados en la operación de las redes actuales.

La cadena de *Markov* a desarrollar supone que dentro del sistema ocurre solo un evento a la vez, desconexión o conexión (de un usuario).

La solución matemática de la cadena de *Markov* se implementará en MatLab, con el fin de obtener las poblaciones promedio de *peers* por ventana del video. Los resultados de la cadena de *Markov* serán traducidos a tasa de subida (ancho de banda de subida proporcionado por los *peers* y ancho de subida requerido de la *CDN*) y tasa de descarga (ancho de banda demandado para la descarga) que son requeridas por el modelo para la distribución de contenido en vivo.

Al momento de que un *peer* se conecte al sistema, el *tracker* notificará cuales son los *peers* capaces de compartirle los *chunks*. Una vez que el *peer* se encuentre dentro del sistema, este podrá obtener *chunks* de múltiples *peers* habilitados para compartir el contenido.

En caso de que los resultados no sean los esperados se debe modificar el esquema propuesto con el objeto de obtener una mejor calidad de servicio (QoS) en comparación con los esquemas utilizados en sistemas reales. En el mismo sentido la cadena de *Markov* que modele el comportamiento del sistema, deberá ser replanteada en caso de que el nuevo esquema de asignación de recursos no muestre los resultados esperados.

El proyecto no considera la implementación del esquema propuesto, sin embargo, si llevara a cabo la evaluación con el modelo desarrollado y considerando parámetros reales de operación de las redes actuales.

4.- Objetivos

4.1.- Objetivo General

Desarrollar un modelo matemático basado en una cadena de Markov, con la finalidad de diseñar un esquema eficiente de asignación de recursos para un sistema de video en vivo, soportado por redes híbridas CDN-P2P

4.2 Objetivos Específicos

Plantear una cadena de Markov basada en las conexiones y desconexiones de los *peers* de cada una de las ventanas del video, para modelar el comportamiento de los múltiples usuarios en el sistema.

Evaluar la cadena de Markov dentro del modelo híbrido CDN-P2P mediante la implementación de métodos numéricos, con la finalidad de estimar la demanda de descarga y la cantidad de recursos de subida disponibles en el sistema.

Diseñar y evaluar un nuevo esquema de asignación de recursos, basado en la solución de la cadena de Markov, con el propósito de mejorar el desempeño del esquema de asignación uniforme tradicional.

5.- Estado del Arte

En este apartado se presentan los trabajos investigados a nivel institucional, nacional e internacional que están relacionados al proyecto que se plantea. Cabe mencionar que a nivel institucional y nacional se lograron identificar muy pocos trabajos como [7], [8] y [9], por lo que la revisión presentada está enfocada principalmente a trabajos publicado en la literatura internacional.

En el presente proyecto el estado del arte se divide en dos áreas:

1. Servicios de video modelados mediante cadenas de Markov
2. Comportamiento y arquitectura de los servicios de video en vivo.

5.1 Trabajos en donde los servicios de video se han modelado mediante cadenas de Markov

En [5] se describe el funcionamiento de una red *Peer to Peer* que soporta videos bajo demanda. En este trabajo se emplea un archivo de video compartido entre diversos *peers*. Estos *peers* son clasificados de acuerdo con la ventana del video que se encuentran descargando. El principal objetivo de este proyecto es establecer un mecanismo eficiente para compartir videos bajo demanda en una red *P2P*. Por otro lado, se menciona que si el ancho de banda es insuficiente se debe recurrir a un ancho de banda proporcionado por algún servidor fijo, sin embargo, esto elevaría los costos de implementación. Finalmente hacen referencia a que este mecanismo permite visualizar los parámetros óptimos para una buena QoE.

En [6] se tiene como principal objetivo optimizar la asignación de recursos en una red para servicio de video bajo demanda. Para lograr dicha asignación de recursos los autores llevan a cabo una evaluación en diversos escenarios basándose en la calidad del servicio (QoS) que ofrece la red. Establecen que a pesar del gran número de factores que analizan, su modelo es preciso y manejable.

En [7] se propone trabajar el servicio de video bajo demanda en una red *P2P* heterogénea de quinta generación, esto con la finalidad de obtener mejores parámetros que reflejen el desempeño del sistema. El objetivo principal de este trabajo es conocer cómo se comporta la red en un servicio de video bajo demanda,

es decir, el ancho de banda que se consume, la calidad del servicio, etc. Para que de esta forma se establezcan estrategias que ayuden a reducir la cantidad de los recursos consumidos, por ejemplo, reducir el ancho de banda que se consume al acceder al video, procurando mantener el desempeño del sistema. Por otro lado, este trabajo se enfoca de igual forma en reducir lo más que se pueda el costo de los servicios de video bajo demanda sobre una red *Peer to Peer*. Para obtener el comportamiento que genera el sistema se emplea un modelo fluido y cadenas de *Markov* que reflejan la evolución de los *peers* dentro del sistema, desde que arriban al sistema hasta el momento de abandonar el mismo. De manera particular este trabajo analiza el comportamiento en periodos de tiempo relativamente grandes con el fin de obtener el comportamiento más cercano al que se suscita en un sistema real.

En [8] y [9] se representó a una red *P2P* pura mediante un modelo fluido (sistemas de ecuaciones diferenciales en estado estable). Estos trabajos fueron los primeros a nivel institucional en modelar un sistema *P2P* donde el video es dividido en ventanas. El resultado de solucionar el modelo fluido provee la cantidad de usuarios en cada ventana. En estos trabajos se definieron las condiciones de abundancia del sistema en función de la tasa de partida de *seeders*.

Referencia	Tipo de servicio	Tipo de arquitectura	Parámetros que definen el estado de la cadena	Parámetros de QoS evaluados	Tipo de análisis	Aportación
[5]	Video bajo Demanda	Red <i>P2P</i>	<ul style="list-style-type: none"> - Tasa de arribo al sistema - Tasa de transición - Tasa de abandono del sistema - Tiempo de descarga 	<ul style="list-style-type: none"> - Ancho de banda de subida - Tasa de descarga - Cantidad de <i>peers</i> en el sistema 	Matemático	Definición de un mecanismo que permita compartir <i>VoD</i> en redes 5G
[6]	Video bajo Demanda	Red híbrida <i>CDN-P2P</i>	<ul style="list-style-type: none"> - Tamaño del video - Tasa de codificación - Popularidad del video - Tasa de subida de datos 	<ul style="list-style-type: none"> - Retardo inicial del servicio - Condiciones de abundancia - Eventos de congelamiento 	Matemático y simulación	<i>Framework</i> que permite evaluar un sistema en diversos escenarios de coste de red y QoS y permite

			- Capacidad de la red <i>CDN</i> y <i>P2P</i> - Comportamiento de los usuarios			modelar diferentes esquemas de asignación de recursos
Este proyecto	Video en Vivo	Red híbrida <i>CDN-P2P</i>	-Tasa de arribo - Tasa de descarga - Tasa de subida de usuarios - Tiempo de permanencia de usuarios en el sistema	- Ancho de banda demandado - Ancho de banda de la red <i>P2P</i> - Ancho de banda de la <i>CDN</i>	Matemático y simulación	Diseño de modelo de servicio de video en vivo y esquema de asignación de recursos propuesto

Tabla 1.1. Comparación de trabajos relacionados y propuesta de este proyecto

Hasta este punto de la investigación los trabajos relacionados que han realizado modelos, basados en cadenas de *Markov*, para servicios de video se han enfocado en video bajo demanda y considerando un esquema de asignación de recursos específico, además de considerar un enfoque en redes 5G o a fin para la aplicación futura de estos modelos. Por otro lado, en el presente proyecto se plantea modelar un servicio de video en vivo empleando las cadenas de *Markov*; además de analizar los diversos esquemas de asignación de recursos empleados actualmente y así poder generar un nuevo esquema de asignación de recursos que nos permita dividir los recursos de manera eficiente entre los usuarios para obtener una mejor experiencia en los servicios de video en vivo.

5.2 Trabajos en donde se menciona el comportamiento y arquitectura de los servicios de video en vivo

En [2] los autores remarcan que el uso de una red híbrida *CDN-P2P* resulta factible para transmitir un archivo de video, sin embargo, ellos se enfocan en analizar el desempeño que tienen las redes híbridas en la transmisión de contenido, en particular analizan:

- *CDN-P2P unconnected mesh*, las cuales establecen una red independiente por cada uno de los nodos involucrados en la red *CDN*.

- *CDN-P2P connected mesh*, en la cual tanto los nodos de la red *CDN* como los *peers* de la red *P2P* se combinan para generar una red general.

La evaluación se lleva a cabo tomando como punto focal la transmisión de video en redes *P2P* básicas y tomando en cuenta diversos parámetros de calidad de servicio (QoS).

En [3] los autores mencionan que la transmisión de video en vivo está creciendo en volumen e importancia especialmente en eventos relevantes que se transmiten a través de Internet. El objetivo de la *CDN* es mejorar el rendimiento del usuario en términos de retraso. Esto se logra mediante la implementación de múltiples nodos generalmente llamados servidores de borde, distribuidos geográficamente en múltiples *ISPs*. Sobre el trabajo relacionado se menciona que un sistema híbrido *P2P-CDN* podría reducir significativamente los costos de ancho de banda del servidor

En [10] se tiene como principal objetivo disminuir el tiempo que tarda un nodo en cambiar de punto de acceso, para ello proponen el uso de un servidor específico llamado “*Zapping Server*” que es el encargado de hacer la conmutación entre los nodos, con el fin de preservar la calidad en tiempo real, es decir, que el usuario no se pierda ninguna escena del video de su interés. Por otro lado, los autores proponen 3 métodos de reservación para preasignar un *peer* como distribuidor del contenido. El primero es basado en el tiempo, seguido del método basado en eventos y el último es un método híbrido.

En [11] los autores establecen un modelo analítico para comprender cómo es que funcionan los modelos de transmisión de video en tiempo real, de manera particular, su modelo proporciona información clave para solucionar problemas en la transmisión de contenido en directo, basándose en *BitTorrent*. A su vez los autores hacen hincapié en que, un buen diseño de una red *P2P* en conjunto con sistemas de transmisión basados en servidores existentes permiten tasas de transmisión más altas en comparación con sistemas basados en redes *P2P* simples. Considerando que la eficiencia de una solución *peer-to-peer* tipo *BitTorrent* depende del tamaño del grupo de *peers* y del número de fragmentos disponibles para compartir en un momento dado, los autores sugieren que la eficiencia de la solución *peer-to-peer* no es sensible al tamaño del grupo, para conexiones de más de 15 usuarios.

En [12] los autores proponen, una estructura híbrida *CDN-P2P* para la transmisión de vídeo en tiempo real a través de la Web, con el fin de reducir el número de solicitudes a los servidores *CDN*, y por ende reducir el costo de la transmisión. Por otro lado, los autores también establecen un tiempo de espera corto para la solicitud al servidor y dividen el archivo de video en segmentos pequeños llamados *chunks*,

en comparación con el modelo *Unconnected Mesh* establecido en [2] el sistema que proponen tiene un menor retraso de extremo a extremo. Finalmente, en este trabajo los autores muestran que considerar el número de peticiones al servidor *CDN* y el retraso de extremo a extremo en un enfoque híbrido *CDN-P2P* resulta eficaz para proporcionar un servicio de transmisión en directo a los diversos usuarios.

En [13] se menciona que hoy en día hay dos enfoques principales para la transmisión de video a gran escala a través de internet. Redes de distribución de contenido (*CDN*) y arquitecturas *P2P*. En la arquitectura *CDN* la secuencia de video se entrega mediante multidifusión de *IP* o multidifusión de aplicaciones *ALM* a una gran cantidad de servidores conectados en diferentes partes de Internet.

Por lo tanto, los autores plantean que el uso de una arquitectura híbrida basada en redes *CDN* y *P2P* puede ser un enfoque eficiente para la distribución de video a gran escala a través de Internet.

El documento afirma que la arquitectura propuesta es adecuada principalmente en aplicaciones sensibles al retardo y en tiempo real, por ejemplo, transmisión de video en vivo.

En [14] se menciona que, para mejorar la escalabilidad del servicio, se han implementado con éxito varias tecnologías de transmisión de video en vivo basadas en *P2P* como lo son *CoolStreaming*, *PPLive*, *PPStream* y *AnySee*. Establecen que los servicios de transmisión *P2P* antes mencionados se basan en relaciones vecinas estructuradas en malla aleatorias y basadas en extracción que puedan proporcionar de manera flexible el servicio de transmisión de video con la dinámica de unión y salida de nodos, así como la suscripción concurrente, respectivamente también se denota la rotación de nodos y la multitud *flash*. Apuntan que el intercambio de datos asistido por *peers* puede reducir el costo de la red al reducir el ancho de banda de descarga de los servidores de contenido hasta aproximadamente un 96%. Por otro lado, la latencia de reproducción es una de las razones de la degradación del rendimiento de los servicios de transmisión multimedia.

En este trabajo, el procedimiento de suscripción del servicio en el sistema híbrido *CDN P2P* es el siguiente: un usuario examina la lista de servicios de transmisión en vivo desde un portal web y envía una suscripción al servicio. Este paquete de solicitud se reenvía al servidor perimetral *CDN* más cercano para reducir la latencia del servicio.

Referencia	Tipo de servicio	Tipo de arquitectura	Parámetros de QoS evaluados	Tipo de análisis	Aportación

[2]	Video en Vivo	Red híbrida <i>CDN-P2P</i>	<ul style="list-style-type: none"> - Tasa de codificación de video - Tamaño de paquete - Número de vecinos - Tiempo de simulación - Número de servidores <i>CDN</i> 	Simulación	Demostró que la arquitectura <i>CDN-P2P connected mesh</i> resuelve problemas de retardo y distorsión en la transmisión de video.
[3]	Video en Vivo	Red híbrida <i>CDN-P2P</i>	<ul style="list-style-type: none"> - Número de peticiones al servidor proxy - Número de <i>leechers</i> - Número <i>seeders</i> 	Matemático y Simulación	Modelo y análisis de un sistema híbrido <i>CDN-P2P</i> para video en vivo que demuestran servir a más usuarios con tasas de codificación más altas.
[10]	Video en Vivo	Red <i>P2P</i>	<ul style="list-style-type: none"> - Número de usuarios en el sistema - Número de <i>Zapping Servers</i> - Número de vistas 	Simulación	<i>Framework</i> para cambiar de punto de acceso en un servicio de video en vivo
[11]	Video en Vivo	Red <i>P2P</i>	<ul style="list-style-type: none"> - Tamaño del grupo de <i>peers</i> - Número de fragmentos del video disponibles para compartir 	Matemático	Modelo analítico del diseño para una solución de video en vivo con BitTorrent
[12]	Video en Vivo	Red híbrida <i>CDN-P2P</i>	<ul style="list-style-type: none"> - Retardo end-to-end - Número de peticiones a los servidores <i>CDN</i> 	Simulación	Demostró que una red híbrida <i>CDN-P2P</i> es más eficaz para proporcionar un servicio de video en vivo al público

[13]	Video en Vivo	Red híbrida <i>CDN-P2P</i>	<ul style="list-style-type: none"> - Tamaño del paquete - Tasa de codificación de video - Número de vecinos - Duración de la simulación 	Matemático y simulación	<p>Demostró que al comparar <i>CDN-P2P</i> <i>unconnected mesh</i> y <i>CDN-P2P connected mesh</i> se obtuvo un mejor rendimiento de la arquitectura <i>connected mesh</i> en términos de retardo end-to-end y la distorsión del servicio.</p>
[14]	Video en Vivo	Red híbrida <i>CDN-P2P</i>	<ul style="list-style-type: none"> - Ancho de banda de subida - Latencia de reproducción - Radio de la red P2P - Distancia del servidor <i>CDN</i> 	Simulación	Método de reducción de latencia en la reproducción al limitar el radio de la red P2P en un sistema híbrido.
Este proyecto	Video en Vivo	Red híbrida <i>CDN-P2P</i>	<ul style="list-style-type: none"> - Ancho de banda demandado - Ancho de banda de la red <i>P2P</i> - Ancho de banda de la <i>CDN</i> 	Matemático y simulación	Diseño de modelo de servicio de video en vivo y esquema de asignación de recursos propuesto

Tabla 1.2. Comparación de trabajos relacionados y propuesta de este proyecto

En los trabajos relacionados descritos anteriormente se ha encontrado que la mayoría de los proyectos que han trabajado videos en vivo han realizado una implementación y simulación del funcionamiento de los modelos propuestos en cada uno de ellos para distribuir el contenido en vivo ya sea a través de una red *P2P* pura, una red híbrida *CDN- P2P*, o variaciones de estas. Existen también trabajos que se han enfocado en comparar 2 tipos de redes o 2 métodos de distribución del contenido en vivo, con el fin de establecer cual es la opción más viable para brindar un mejor servicio al público. En el presente proyecto se busca

realizar un modelo basado en cadenas de *Markov* permita conocer el comportamiento de los usuarios conectados a servicio de video en vivo a través de una red híbrida *CDN-P2P*, además de generar un nuevo esquema de asignación de recursos que nos permita alcanzar una mejor calidad en el servicio y en la experiencia al establecer parámetros óptimos para la distribución de contenido en vivo.

6.- Marco Teórico

En esta sección se describe al video en vivo, así como los diversos formatos empleados para la codificación del contenido. De igual forma, se muestran los formatos utilizados para el empaquetamiento de este. Enseguida se muestran los protocolos para la distribución de video en vivo por Internet. Y finalmente se enuncia la manera en la cual operan las redes híbridas *CDN-P2P* en los servicios de video en vivo.

6.1. Video en Vivo

6.1.1. Formatos de codificación

En la transmisión de contenido en vivo se usa como principal herramienta a un *códec*, que es el acrónimo de codificador-decodificador. Esto se debe a que la señal de video necesita ser codificada en un formato de *streaming* que permita tener las menores pérdidas de calidad y sonido posibles. Los *códecs* realizan la codificación *streaming* para comprimir el contenido a emitir y descifrarlo una vez recibido en los diferentes dispositivos móviles para la reproducción.

A continuación, se describen algunos de los formatos de codificación que permiten la transmisión de audio y video en un solo paquete a través de una red. Información obtenida a partir de [15] [16].

Mjpeg: Un video no es más que una sucesión de imágenes en movimiento. Si se comprimen todas las imágenes en formato JPEG se obtendría el formato MJPEG, o Motion JPG. Con este formato se logra una buena compresión con respecto al original.

H264: es un formato de codificación de video para grabar y distribuir señales de video FullHD y audio. Fue desarrollado y mantenido por el ITU-T Video Coding Experts Group (VCEG) con el ISO/IEC JTC1 Moving Picture Experts Group (MPEG). Empleado normalmente para grabación, compresión y distribución de contenidos de

vídeo, el formato H.264 es un método de transmisión de video compatible con redes de datos, que suministra imágenes de alta calidad sin consumir demasiado ancho de banda.

Mpeg4: Es una evolución de MPEG. Es más avanzado y ofrece numerosas opciones de configuración y muy alta calidad. Sin embargo, su edición exige gran potencia en hardware.

VP8 (Formato de Compresión de Video o Especificación de Compresión de Video): Es una especificación para codificar y decodificar video de alta definición tanto en archivo como en flujo de bits para visualizar.

VP9: Es de código abierto y libre de regalías. Fue desarrollado por *Google* como sucesor del VP8, la alternativa moderadamente exitosa al H.264. Durante su desarrollo VP9 fue apodado “NGOV” (*Next Gen Open Video*) y *Google* ya ha integrado el soporte en el navegador *Chrome* y en *YouTube*.

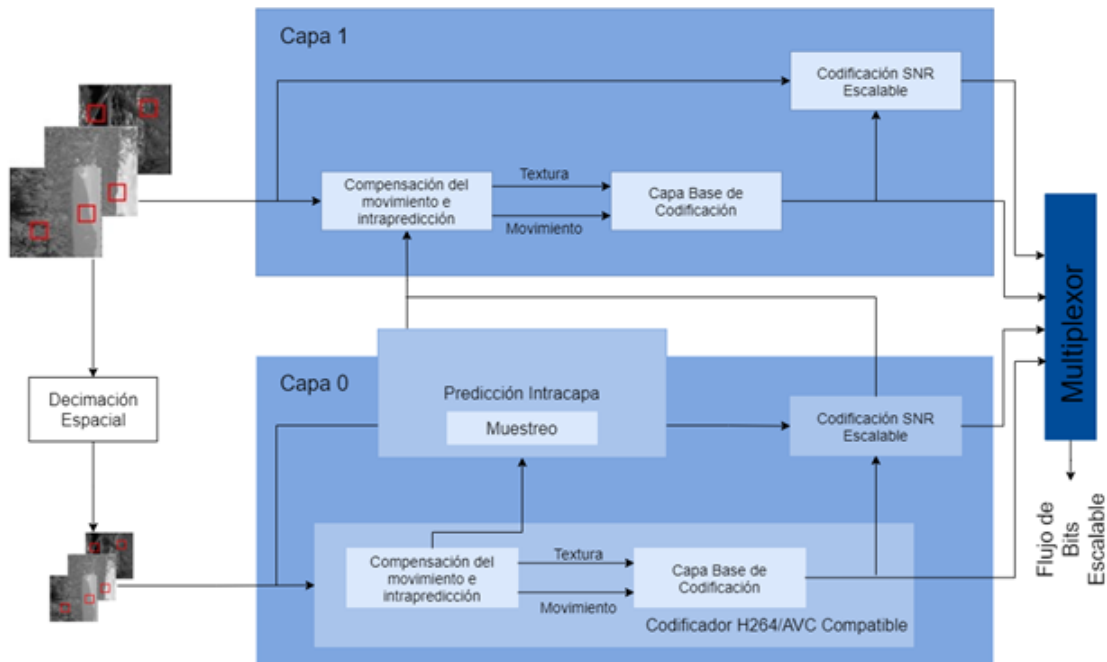


Figura 8. Diagrama a bloques del códec H264 [17]

6.1.2. Formatos de empaquetamiento

A continuación, se mencionan los formatos de empaquetado más usados dentro de la distribución de video y audio en vivo [18]

.avi: Es el empaquetado multimedia con mayor utilidad como archivo de vídeo. Significa audio video interleave, es decir video y audio entremezclados.

.mov: Empaquetado por defecto de *QuickTime* y productos *Apple*. Otras extensiones recientes compatibles con .mov son: .mp4,.m4a,.3gp,.mj2.

.mpeg: Empaquetado ampliamente utilizado, por ejemplo, en la emisión TDT. Otras extensiones compatibles son .MOD y .mpg.

.vob: Formato mpeg2 con opción de encriptado para su uso en el DVD.

.flv: Formato propio de las animaciones *flash*.

.ogg: Empaquetado tradicionalmente usado para el audio vorbis.

.mkv, .matroska: Empaquetado especial para alta densidad de datos en HD.

.webm, .WebM: Formato de empaquetado libre de última generación

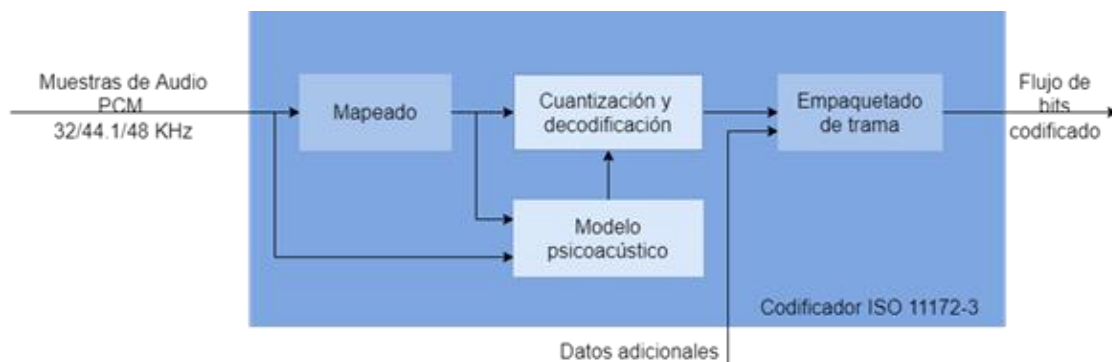


Figura 9. Diagrama a bloques del formato de codificación .mpeg [19]

6.1.3 Servicios de video en vivo

Un *stream* de contenido multimedia, generalmente llamado video, está compuesto por varios *streams* simples que poseen un tipo de contenido específico (audio, video o subtítulos). Cada *stream* simple es codificado por un códec en un formato específico como h264, mp4 o mjpeg para el caso de la señal de video. En el caso de la señal de audio existen formatos como Mp3, aac, ogg, pm_s16le.

Los contenidos de un *stream* de multimedia son empaquetados en un nuevo formato llamado *container*, mov, avi, mpeg, flv, vob son algunos de estos. El objeto de emplear un *container* es lograr sincronía entre los contenidos. Una vez realizado el empaquetado el *stream* de multimedia se puede almacenar como archivo en un servidor o se puede transmitir en directo.

Se estima que el tráfico global en Internet crecerá 8 veces para el año 2022, donde el servicio de video ocupará 80% del tráfico, hecho que no es de extrañarse, ya que desde el año 2018 alrededor del 60% de la población a lo largo del mundo visualizaba videos en sus dispositivos móviles. [20]

Los archivos de video comúnmente son producidos y distribuidos por una amplia gama de equipos de usuario (UE), tales como teléfonos celulares, dispositivos inteligentes, o dispositivos conectados al Internet de las cosas (*IoT*). Aunado a esto, durante el proceso de producción de videos, se pueden encontrar diferentes tipos de contenido de video, como películas, anuncios, aplicaciones de realidad aumentada, videos de seguridad, etc. En la tabla 2 se enlistan las características principales de los archivos de video.

Atributo	Descripción
Espacio de Color	Modelo matemático que asigna una representación del color a su equivalente perceptivo.
Resolución	El número de píxeles en una ventana de video, expresada como NxM píxeles, donde N y M representan el número de columnas y filas respectivamente, en la matriz de píxeles.
Proporcionalidad	Relación entre el ancho y alto de la ventana de video.
Tasa de Ventana	El número de Ventanas por segundo(fps) de un video.
Tasa de bit	El número de bits necesario para representar un segundo de un video codificado.
Calidad de Video	Fidelidad de un video codificado respecto a su versión original, la cual es medida subjetivamente o mediante una métrica objetiva (PSNR, VQM y VMAF)
Esquema de Codificación	Formato de codificación de video y parámetros de codificación.

Tabla 2. Atributos de los videos [20]

Los servicios de video distribuidos a través de la Internet son agrupados en dos principales categorías, los servicios bajo demanda y los servicios de *live streaming*.

La distribución de contenido multimedia hace referencia al proceso por el cual se distribuyen los servicios, como películas, clips de video y/o transmisiones en vivo sobre una red, ya sea en tiempo real o no. Existen 2 métodos principales de dispersión de contenido multimedia sobre una red, *downloading* y *streaming*. Por objetivos del proyecto el *streaming* y una parte de *downloading* marcarán uno de los ejes principales de investigación. [4]

Downloading (progressive downloading): es el procedimiento donde el usuario puede visualizar el contenido multimedia mientras lo está recibiendo de otro dispositivo.

Streaming: a diferencia de *downloading*, en este método de distribución se cuenta con un servidor *streaming* especializado, el cual distribuye el contenido multimedia a los diversos usuarios conectados. La distribución se realiza considerando los recursos específicos por asignar a cada uno de los usuarios en función de su solicitud. En este método no es necesario que el archivo de video sea descargado en el dispositivo final del usuario.

El presente proyecto está enfocado en el servicio de *live streaming*. Plataformas como *Sling TV*, *DirecTV Now*, *YouTube TV*, y *Live TV de Hulu* [21] son las encargadas de transmitir contenido en vivo.

6.1.4. Protocolos de arquitectura y distribución de video en vivo por internet

En la transmisión de video en vivo diversas tecnologías trabajan colectivamente para lograr la distribución del contenido sobre una red. Esto se debe a que el *streaming* implica diversas etapas, en primer lugar, se capta el contenido desde el dispositivo donde se produce, posteriormente este contenido entra en la etapa de codificación para así poder ser almacenado y transmitido a los diversos espectadores de manera simultánea.

Un protocolo de transmisión es la tecnología empleada para transportar archivos de video a través de Internet. [22]

Anteriormente el video disponible en la red era entregado por el protocolo RTMP, que es el protocolo de mensajería en tiempo real. Este protocolo es considerado como un estándar basado en *flash* para *live streaming*. Actualmente aún es utilizado para el envío de video desde el codificador RTMP a una plataforma de video en línea.

Sin embargo, el protocolo RTMP basado en *flash* ya no es apropiado para entregar video a los usuarios. Esto gracias a que el complemento *flash* ha perdido importancia debido a que los dispositivos que admiten este protocolo son cada vez menos. En los últimos diez años el protocolo RTMP ha sido reemplazado paulatinamente por el protocolo HLS [22]. De forma paralela el protocolo MPEG-DASH ha tomado un lugar importante en *live streaming*.

MPEG-DASH es el protocolo de transmisión utilizado con mayor frecuencia en los últimos días. Este protocolo fue creado como una opción frente a la fragmentación que sufrió el mercado de transmisión de video causada por la competencia de HLS de *Apple* con diversos protocolos de transmisión de video. Esto obligo a las organizaciones de estándares a desarrollar MPEG-DASH como un protocolo de transmisión estándar de código abierto, al igual que el protocolo de transmisión HLS.

MPEG-DASH es un método de tasa de bits adaptativo para la transmisión de video. Por otro lado, ha mostrado ser compatible con la publicidad y la tecnología, razón por la cual debe estar a la vanguardia. MPEG-DASH es compatible con DRM, entrega HTTP, transmisión de baja latencia y otras características de *live streaming*.

Dispositivos *Android*, *iOS*, *Windows*, *Mac*, *Linux*, *Chrome OS* por mencionar algunos, trabajan con el protocolo HLS y MPEG-DASH. Sin embargo, este último no es compatible con el navegador *Safari* móvil, por lo tanto, no es admitido por la mayoría de los usuarios *iPhone*, *iPad* y *AppleTV* ya que utilizan el navegador *Safari* como buscador predeterminado.

HLS y MPEG-DASH son protocolos de velocidad de bits adaptables. Lo cual se traduce, en que los usuarios reciben automáticamente el video con mejor calidad a razón de la calidad de conexión a Internet que poseen. En un escenario ideal este hecho proporciona una experiencia de visualización estable y de alta calidad a los espectadores mientras que se reduce el almacenamiento en búfer y el retraso.

A medida que el tráfico de video en las redes celulares crece exponencialmente, los operadores de redes móviles (ORM) aplican tecnologías de quinta generación (5G) de redes de comunicaciones. Esto con la finalidad de satisfacer los requisitos de calidad de servicio y de experiencia en las aplicaciones multimedia. El objetivo principal de la implementación de estas tecnologías es ofrecer servicios multimedia de alta velocidad de datos, baja latencia y fiables en comunicaciones móviles de banda ancha y de baja latencia

Lo descrito anteriormente se logra introduciendo la computación de borde multiacceso (MEC), que integra tecnologías de computación en la nube y de redes inalámbricas. La idea principal de la MEC es asignar recursos informáticos a los usuarios finales dentro de la red de acceso de radio (RAN). El creciente interés por aprovechar los recursos de borde para ofrecer mejores experiencias multimedia ha provocado que surjan múltiples plataformas MEC comerciales. [20]

Las capacidades de almacenamiento de contenido en caché de las redes centradas en la información (ICN) se han combinado con MEC. Esto con el propósito de proporcionar capacidades integradas de almacenamiento en caché, computación y comunicación (edge-C3). En las aplicaciones multimedia, el edge-C3 es capaz de procesar y almacenar simultáneamente contenido de video en caché para proporcionar a los usuarios servicios de baja latencia y gran ancho de banda. Este mecanismo se muestra en la figura 6, donde se puede observar que los UE cuentan con capacidades de computación y almacenamiento de mayor potencia. Esto les permite participar también en el edge-C3. Además, el crowdsourcing móvil y la comunicación de dispositivo a dispositivo (*D2D*) permiten que los equipos de los

usuarios adyacentes puedan compartir recursos entre sí, lo que reduce la congestión de la red y los recursos que deben utilizarse en los servidores de borde.



Figura 10. Escenario de video edge-C3 en redes inalámbricas [20]

6.2. Redes CDN-P2P

Las redes *P2P* (*Peer to Peer*) son utilizadas para incrementar la capacidad de los sistemas. En comparación con los sistemas convencionales cliente-servidor, las redes *P2P* reducen el tráfico de peticiones a los servidores gracias al trabajo colaborativo realizado por los nodos. En este sentido las redes *P2P* resultan ser más escalables, es decir, su capacidad aumenta conforme aumenta la cantidad de usuarios conectados al sistema.

En las redes de distribución de contenidos (*CDN*), los servidores son proporcionados por diversos proveedores de servicios de Internet (*ISP*). Estos servidores permiten a los usuarios finales acceder a los diferentes contenidos disponibles en la red, reduciendo el retraso y mejorando la calidad del servicio (*QoS*); desgraciadamente la cantidad de tráfico de video cada vez es mayor, por lo

tanto, las *CDN* se enfrentan con problemas de escalabilidad debido al costo de su implementación.

Por otro lado, como se ha mencionado las redes *P2P* han demostrado ser más escalables pero su rendimiento se degrada por factores asociados al comportamiento de los usuarios, como *freeriding* (*peers* no cooperativos) y *churning* (desconexión inesperada de *peers* habilitados para distribuir el contenido). [5]

En este contexto, los sistemas híbridos *CDN-P2P* muestran ser eficaces por la capacidad de escalabilidad de las redes *P2P* y la estabilidad proporcionada por la infraestructura fija de las *CDN*. Esto se debe a que las redes *CDN-P2P* consideran las características de los archivos de vídeo (la popularidad, la tasa de codificación y el tamaño), las propiedades de las *CDN* (capacidad de carga), las propiedades de las *P2P* (tasas de datos de carga y descarga, el tiempo de permanencia), y los esquemas implementados para asignar los recursos disponibles. [6]

En un inicio las redes *P2P* surgieron como un sistema de distribución de archivos, sin embargo, en años recientes ha incrementado el análisis de este tipo de redes para la distribución de video. En este tipo de servicio es importante resaltar que la reproducción de un video es iniciada a pesar de que su descarga esté en curso.

Las redes *P2P* representan una tecnología estrechamente relacionada con la distribución de video en la quinta generación de sistemas móviles (5G). Las redes *P2P* tienen como base principal al protocolo *BitTorrent* [5], el cual tiene como principal objetivo dividir un archivo de video en diversos segmentos denominados *chunks*. Estos pequeños fragmentos de video permiten que los diversos *peers* en el sistema descarguen un archivo de video intercambiando *chunks* entre sí.

En las redes *P2P*, el protocolo *BitTorrent* separa a los *peers* dentro del sistema en dos grupos: *leechers*, *peers* que contienen una porción del archivo de video. Y *seeders*, *peers* que poseen el archivo de video por completo y permanecen en el sistema con el fin de compartir sus recursos con otros *peers*. Tanto los *seeders* como los *leechers* comparten sus recursos con otros *leechers*.

Cuando un *peer* se une al sistema para descargar el video, se debe contactar con un nodo llamado *tracker*, cuya función principal es crear una lista de los *peers* que poseen una porción o el video en su totalidad. Posteriormente el *tracker* muestra una lista de los *peers* habilitados para compartir sus recursos con el usuario que llegó. Finalmente, el usuario se pone en contacto con los *peers* en la lista y establece que fragmentos de video descargará de cada *peer* con el que esté conectado.

En trabajos como [5] y [6] agrupan un número determinado de *chunks* en segmentos de video relativamente más grandes llamados ventanas. En consecuencia,

clasifican a los *peers* de acuerdo con la ventana de video que se encuentran descargando. En la figura 11 se puede observar la clasificación de los *peers* en un plano cartesiano.

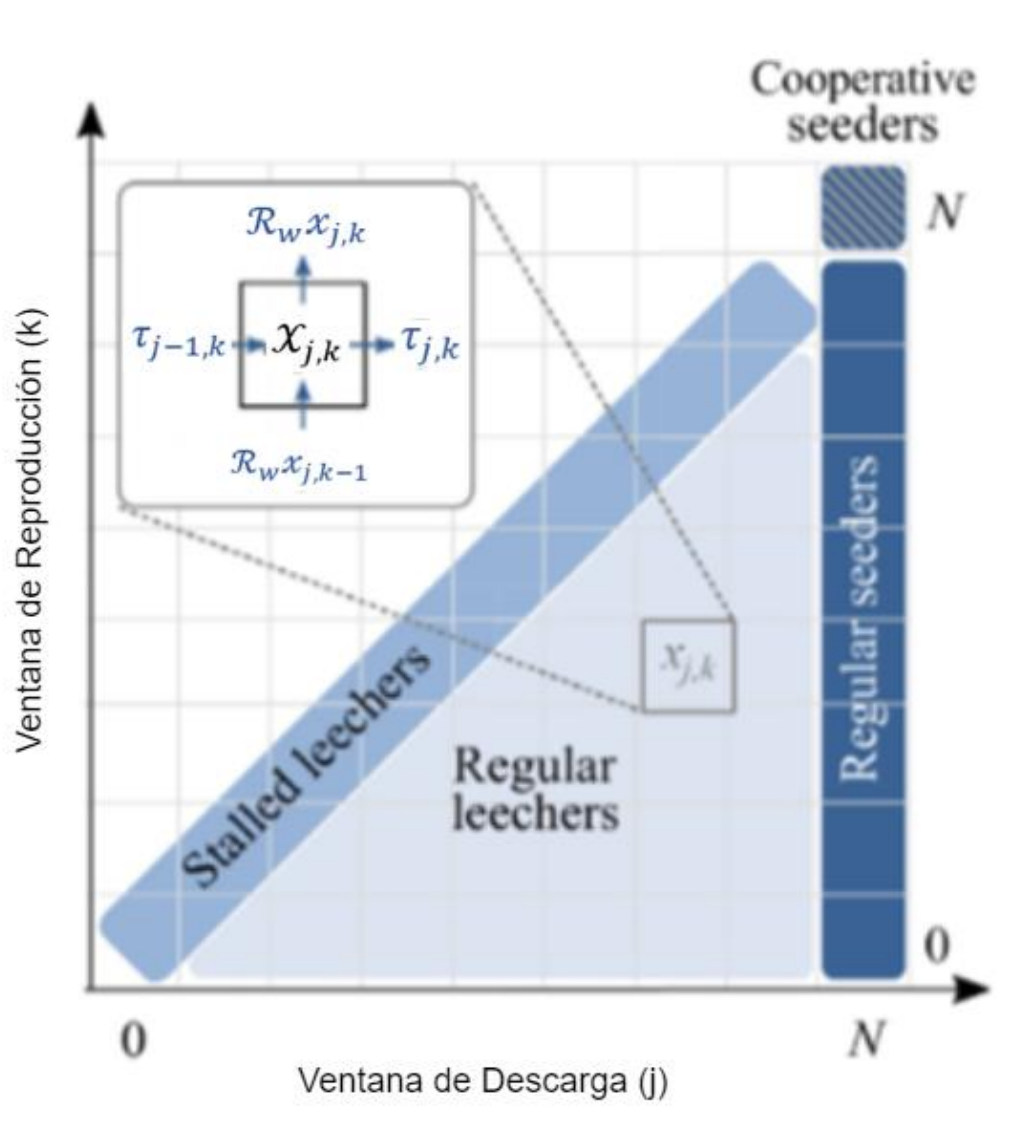


Figura 11. Clasificación de *peers* en un escenario básico [6]

En los sistemas híbridos *CDN-P2P* dado un esquema de asignación de recursos, se puede obtener la proporción de tasa de subida que proporciona la *CDN* y la proporción de la *P2P* para satisfacer la demanda de un servicio de video. Estas tasas de datos están directamente relacionadas con el costo operativo de la *CDN* y con la cantidad de recursos que deben proceder de la red *P2P* (energía y el ancho de banda de subida).

6.3. Servicios de video sobre redes híbridas *CDN-P2P*

El video en vivo a través de internet, incluidos los servicios *OTT* e *IPTV*, han incrementado su polaridad en los últimos años [23]. Estos servicios se basan en el modelo cliente-servidor que utiliza la transmisión adaptativa del protocolo de transmisión de hipertexto (*HTTP*).

Los servicios de video *OTT* comúnmente emplean redes de distribución *CDN* que almacenan el contenido en caché en múltiples ubicaciones geográficas, conocidas como puntos de presencia (*PoP*). *CDN* no solo facilita la carga en el servidor de origen, sino que también reduce la latencia. Por otro lado, los servicios de *IPTV* emplean la transmisión de multidifusión del protocolo de internet *IP* a través de redes privadas habilitadas para multidifusión, donde una única transmisión es originada en el servidor y se entrega a varios clientes. Sin embargo, el protocolo de multidifusión de *IP* no se admite en la internet abierta, por lo tanto, no está disponible para los servicios de video.

Se han propuesto sistemas híbridos *CDN-P2P* para ofrecer un servicio *CDN* escalable al mismo tiempo que se reduce su ancho de banda y los costos del servidor. Los sistemas más recientes se centran en servicios *CDN-P2P* entregados a navegadores a través de canales de datos *WebRTC* sin depender de complementos de terceros o software de propietarios.

Las principales aplicaciones *CDN-P2P* basadas en *WebRTC* incluyen *SwarmCDN*, *PeerCDN* y *Sharefest* que a su vez condujeron a soluciones comerciales, como *Swarmify*, *Peer5* y *Streamroot*. En estos sistemas, los clientes que solicitan contenido son redirigidos a un servidor centralizado (rastreador) que devuelve una lista de *peers* con el contenido deseado. Sin embargo, el intercambio de contenido entre *peers* no es administrado por el servidor central y estos sistemas pueden no ser compatibles con *NSP*. Un componente importante de los sistemas *P2P* es la formación de grupos.

Existe una serie de diferentes métodos de agrupamiento en la literatura, que se pueden aplicar a la formación de grupos *P2P* basados en mallas.

La aplicación *CDN* es un servicio que se ejecuta en el nodo de cómputo del centro de datos de *NSP Edge*. Extrae contenido de video en vivo desde el servidor de origen al centro de datos *NSP Edge*. Solo un pequeño subconjunto de pares recibe contenido en vivo desde el nodo *CDN*. Todas las demás solicitudes de contenido se redirigen a la aplicación *P2P*.

El control del servicio *P2P-CDN* que se propone en [23] consta de dos fases:

- i) La fase de inicio, la aplicación *P2P* envía una serie de fragmentos a los *peers* a través de un árbol de multidifusión para garantizar que cada usuario almacena en búfer un número predeterminado de fragmentos antes de comenzar la reproducción. Se prefiere la multidifusión de *IP* administrada por el controlador *SDN* en la fase de inicio, ya que se sabe que la multidifusión de *IP* tiene un retardo bajo y no se espera un abandono de *peers* durante el corto período de inicio.
- ii) En la fase posterior a la reproducción, la aplicación *P2P* realiza una programación de fragmentos pseudocentral basada en malla para todos los *peers* en cada grupo *P2P*. En esta etapa se usa la transmisión *P2P* basada en malla porque es más resistente a la rotación de *peers*. En los sistemas *P2P* de última generación, los *peers* realizan la programación de forma distribuida, lo que conduce a un uso ineficiente de los recursos de la red a medida que los *peers* se favorecen a sí mismos. Se propone una programación pseudocentralizada de fragmentos, realizados en los centros de datos *NSP-edge* para todos los *peers* dentro de cada grupo *P2P*, que sea justa para todos y utilice los recursos de la red de manera más eficiente. El procedimiento es escalable, ya que el número de *peers* en un grupo es limitado.

La fase de inicio continúa hasta que el servidor *CDN* inserta un número predeterminado de fragmentos a lo largo del árbol de multidifusión.

Los *peers* inician la reproducción y entran en la fase posterior a la reproducción tan pronto como almacenan estos fragmentos en búfer. La demora máxima de inicio se calcula cuando todos los *peers* del grupo inician la reproducción. La programación *P2P* basada en malla, posterior a la reproducción, se lleva a cabo en intervalos de tiempo fijos para minimizar la sobrecarga de señalización entre la aplicación *P2P* y los *peers*.

7.- Escenario de Pruebas

En las secciones anteriores se estableció que el enfoque del proyecto es la definición y solución numérica de la cadena de *Markov* que modela a los servicios de video en vivo. La solución de esta cadena permitirá conocer la población promedio de *peers* en cada una de las ventanas del video y, con base en este resultado, se obtendrán los siguientes parámetros de operación de la red:

- Población de *peers*.

- El ancho de banda demandado para la descarga.
- El ancho de banda de subida proporcionado por los *peers*.
- El ancho de subida requerido de la *CDN*.

Dichos parámetros se mostrarán gráficamente en función de:

- La tasa de arribo (λ).
- La tasa de abandono (θ).
- El número de ventanas (N).

Además, se calculará el requerimiento por ventana para cada uno de los parámetros de operación antes mencionados, lo que permitirá también graficarlos en función de los identificadores de ventanas. Estos últimos resultados serán la base para el cálculo de los siguientes parámetros de QoS:

- Tiempo de descarga de cada ventana.
- Probabilidad de pausa en la reproducción.

Las evaluaciones previamente mencionadas se realizarán considerando valores para los parámetros de entrada obtenidos de sistemas reales (reportados en estudios experimentales de otros autores). En la Tabla 3, se resumen algunos de ellos.

Parámetro	Valor
Tasa de arribo (λ)	0.04
Tasa de abandono (θ)	0.006
Tasa de subida de usuarios (μ)	0.00254
Tasa de descarga (c)	0.00407

Tabla 3. Parámetros de entrada en un Sistema Real [6]

El análisis por ventana, permitirá el diseño de un nuevo esquema de asignación de recursos, como se mencionó en secciones previas. Por lo tanto, los parámetros de desempeño serán evaluados, tanto para el esquema tradicional de asignación de recursos uniforme como para el nuevo esquema que se diseñará en este proyecto. Se espera que el nuevo esquema propuesto proporcione un mejor desempeño frente al esquema uniforme, particularmente en términos de los parámetros de QoS.

Debido a que la solución numérica de la cadena de *Markov* en diversos escenarios puede ser computacionalmente compleja, se realizará en una computadora con sistema operativo Windows 10 Home Single de 64 bits, marca Dell Inc, con una

BIOS de 2.5.1, un procesador Intel(R) core(TM) i5-10400 de 2.9 GH y una memoria RAM de 8 GB, además de tener disponible entorno de desarrollo MatLab. Las evaluaciones mencionadas anteriormente se realizarán considerando parámetros de entrada (tasa de arribo, tasa de descarga, tasa de subida de usuarios, tiempo de permanencia de usuarios en el sistema) obtenidos de sistemas reales.

Como producto de este proyecto se tendrá, entonces:

- Un modelo del servicio de video en vivo, mediante cadenas de *Markov*.
- Un nuevo esquema de asignación de recursos
- Gráficas que permitan analizar el desempeño del sistema en función de parámetros de QoS, dado un conjunto de valores para los parámetros de entrada.

Todo esto se realiza con la finalidad de mejorar el desempeño en términos de QoS de los sistemas analizados.

8.- Cronograma de Actividades

En la siguiente tabla se describen las actividades a desarrollar durante Proyecto Terminal I. Cada una de las actividades es asignada a cada integrante del equipo. Debido a las actividades inmersas dentro del proyecto y los tiempos en que se debe desarrollar, se tomó la decisión de plantear la siguiente metodología para lograr los objetivos.

Alumno 1: Muñoz Ruiz Ulises

Alumno 2: Ortiz Islas José Manuel

Número	Responsable	Nombre de la Actividad	Objetivo	Resultados Esperados
1	Alumno 1 Alumno 2	Delimitación del proyecto	Obtener los tópicos de investigación base para el proyecto.	Escrito donde se plasman las ideas fundamentales para establecer la estructura

				general del proyecto.
2	Alumno 1	Investigación	Investigar sobre los temas a tratar.	Estado actual de nuestro esquema de investigación.
3	Alumno 1	Investigación	Investigar esquemas de asignación de recursos.	Conocer diversos esquemas de asignación de recursos para servicios de video en redes <i>P2P-CDN</i> .
4	Alumno 2	Análisis de los protocolos para video en vivo	Analizar el efecto que tienen los protocolos IP, UDP y MEGA-DASH en el desempeño de sistemas de video en vivo.	Efectos de los protocolos en el video en vivo.
5	Alumno 2	Investigación	Analizar los procesos de codificación y empaquetado de señales de audio y video utilizadas para servicio de video en vivo.	Conocer el tipo de codificación y empaquetado de señales de video vivo.

6	Alumno 2	Análisis del funcionamiento de asignación de recursos	Analizar la asignación de recursos en un esquema uniforme.	Comprender el funcionamiento de asignación de recursos uniforme.
7	Alumno 2	Esquemas	Seleccionar esquemas de asignación de recursos.	Esquemas de asignación de recursos adicionales al uniforme.
8	Alumno 2	Esquemas seleccionados	Analizar el funcionamiento de los esquemas seleccionados.	Esquemas por utilizar.
9	Alumno 1	Agregar características del proceso de codificación y empaquetado	Incorporar las características de codificación y empaquetado de video en vivo al modelo y al esquema de asignación de recursos a desarrollar.	Parámetros para obtener un mejor diseño en el modelo y el esquema de asignación de recursos.
10	Alumno 2	Añadir los efectos de los protocolos	Incorporar los efectos de los protocolos en el modelado y en el diseño del nuevo esquema.	Impacto de los protocolos en el modelado de video en vivo.

11	Alumno 1	Modelado del comportamiento de los <i>peers</i>	Obtener los distintos estados que puede presentar el sistema.	Escenarios en donde se pueden desenvolver los <i>peers</i> .
12	Alumno 1	Análisis de escenarios por ventana	Analizar los escenarios en una ventana.	Tamaño de la población de <i>peers</i> en el transcurso de una ventana.
13	Alumno 1	Análisis de la conexión de <i>peers</i>	Analizar la conexión de <i>peers</i> antes de iniciar la transmisión.	Número de <i>peers</i> que se conectarán directamente al servidor. Encargados de distribuir el contenido a los <i>peers</i> que se conecten posteriormente.
14	Alumno 1	Análisis de la llegada de nuevos <i>peers</i>	Analizar el arribo de <i>peers</i> después de iniciada la transmisión.	Número de <i>peers</i> en la ventana con d retardos.
15	Alumno 1	Análisis del abandono del sistema por parte de un <i>peer</i>	Analizar la desconexión de <i>peers</i> antes de que finalice la transmisión.	Cantidad de <i>peers</i> que obtenían el contenido del <i>peer</i> que se desconectó.

16	Alumno 1	Análisis de transferencia de un <i>peer</i> a la ventana superior	Analizar el proceso de descarga de una ventana.	Determinar la ventana siguiente que debe descargar.
17	Alumno 2	Diseño del nuevo esquema de asignación	Diseñar un nuevo esquema de asignación de recursos entre <i>peers</i> .	Esquema de asignación de recursos a utilizar.
18	Alumno 1 Alumno 2	Proposición de cadena de <i>Markov</i>	Obtener una cadena de <i>Markov</i> que ilustre el cambio en los estados del sistema.	Cadena de <i>Markov</i> que simule el comportamiento del sistema considerando el esquema propuesto.
19	Alumno 1 Alumno 2	Evaluación de la cadena de <i>Markov</i>	Verificar la cadena de <i>Markov</i> durante la transmisión.	Población de <i>peers</i> promedio por ventana.
			Determinar el ancho de	

20	Alumno 1	Obtención del ancho de banda consumido	banda demandado por los <i>peers</i> para descargar una ventana.	Cantidad de ancho de banda consumido al descargar una ventana.
21	Alumno 2	Obtención del ancho de banda de los <i>peers</i>	Determinar la cantidad de ancho de banda obtenido a partir de los <i>peers</i> .	Proporción del ancho de banda consumido, proporcionado por los <i>peers</i> .
22	Alumno 2	Obtención del ancho de banda del servidor <i>CDN</i>	Determinar la cantidad de ancho de banda obtenido del servidor <i>CDN</i> .	Proporción del ancho de banda consumido a través del servidor <i>CDN</i> .
23	Alumno 1	Análisis de las expresiones del modelo	Determinar las expresiones de abundancia y penuria considerando un esquema de distribución uniforme.	Expresiones de abundancia y penuria del sistema.
24	Alumno 1 Alumno 2	Redacción de estructura PT1	Generar el documento que contenga el desarrollo de PT1.	Documento que reporte las especificaciones y procedimientos de PT1.

25	Alumno 1 Alumno 2	Evaluación de esquema	Ajustar el esquema propuesto en función del desempeño que se obtenga.	Funcionamiento del esquema.
26	Alumno 1 Alumno 2	Replantear la cadena de <i>Markov</i>	Establecer la cadena de <i>Markov</i> con base en el funcionamiento del esquema propuesto.	Cadena de <i>Markov</i> modificada.
27	Alumno 1 Alumno 2	Evaluar la cadena de <i>Markov</i> modificada	Comparar los resultados de la cadena de <i>Markov</i> empleando un esquema de asignación uniforme y los resultados de la cadena de <i>Markov</i> basada en el esquema propuesto.	Tasas de subida y descarga basados en los diferentes esquemas de asignación de recursos.
28	Alumno 1 Alumno 2	Cotejo del modelo	Evaluar el sistema utilizando la asignación uniforme con el esquema propuesto.	Funcionamiento del modelo.

29	Alumno 1 Alumno 2	Implementación de la cadena de <i>Markov</i>	Implementar la cadena de <i>Markov</i> en un entorno de programación.	Verificación de la cadena de <i>Markov</i> .
30	Alumno 1 Alumno 2	Evaluación esquema final	Comparar el esquema final con otros esquemas.	Esquema final.

En el siguiente diagrama se muestran las actividades a realizar a lo largo de Proyecto Terminal I.

[illegible]

21																	
22																	
23																	
24																	

En el siguiente diagrama se muestran las actividades posibles a realizar a lo largo de Proyecto Terminal II.

Actividad	Semana															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
25																
26																
27																
28																
29																
30																

9.- Referencias

- [1] T. Cooper, «BROADBANDNOW,» 28 Octubre 2020. [En línea]. Available: <https://broadbandnow.com/report/cable-vs-satellite-vs-iptv-vs-ott-streaming/#:~:text=Essentially%2C%20IPTV%20is%20a%20formally,available%20to%20consumers%20from%20ISPs.&text=Rather%20than%20streaming%20content%20directly,to%20display%20on%20your%20TV..> [Último acceso: 21 Mayo 2021].
- [2] S. M. Y. Seyyedi y B. Akbari, «Hybrid CDN-P2P Architectures for Live Video,» de *International Symposium on Computer Networks and Distributed Systems (CNDs)*, Tehran, 2011.
- [3] A. Mansy y M. Ammar, «Analysis of Adaptive Streaming for HybridCDN/P2P Live Video Systems,» IEEE, Atlanta, 2011.
- [4] R. Trestian, I.-S. Comsa y M. Fatih, «Seamless Multimedia Delivery Within aHeterogeneous Wireless NetworksEnvironment: Are We There Yet?,» *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. 20, nº 2, pp. 945-977, 2018.

- [5] N. Torres Cruz, M. E. Rivero Angeles, G. Rubino, R. Menchaca Mendez y R. Menchaca Mendez, «A Window-Based, Server-Assisted P2P Network forVoD Services with QoE Guarantees,» *Hindawi*, vol. 2017, nº 2084684, pp. 1-18, 2017.
- [6] N. Torres Cruz, M. E. Rivero-Angeles, G. Rubino, R. Menchaca Mendez, R. Mechaca Mendez y D. Ramirez, «A comprehensive analytical framework VoD Services in hybrid CDN-P2P,» *Elsevier*, vol. 102643, nº 161, pp. 1-17, 2020.
- [7] E. D. Terrones Celis, Artist, *Modelado y evaluación de servicios de video sobre redes P2P en ambientes móviles 5G*. [Art]. Centro de Investigación en Computación, 2020.
- [8] E. E. Báez Esqueive, Artist, *Estudio del efecto de la movilidad en redes de radio cognoscitivo con arquitectura P2P*. [Art]. ESCOM-IPN, 2015.
- [9] E. E. Báez Esquivel, Artist, *Análisis del desempeño de una red P2P basada en BitTorrent con prioridades para la transmisión de video en demanda almacenado basado en ventanas*. [Art]. UPIITA-IPN, 2012.
- [10] D. Kondo, Y. Hirota, A. Fujimoto, H. Tode y K. Murakami, «P2P Live Streaming System for Multi-view Video with Fast Switching,» de *16th International Telecommunications Network Strategy and Planning Symposium (Networks)*, Funchal, 2014.
- [11] S. Tewari y L. Kleinrock, «Analytical Model for BitTorrent-Based Live Video Streaming,» de *4th IEEE Consumer Communications and Networking Conference*, Las Vegas, 2007.
- [12] T. T. Thu Ha, J. Kim y J. Nam, «Design and Deployment of Low-Delay Hybrid CDN–P2PArchitecture for Live Video Streaming Over the Web,» *Springer Science+Business Media*, pp. 513-525, 2017.
- [13] S. Seyyedi y B. Akbari, «Hybrid CDN-P2P Architectures for Live Video Streaming: Comparative Study of Connected and Unconnected Meshes,» de *Intenational Symposium on Computer Networks and Distributed Systems(CNDS)*, México, 2011.
- [14] S. Cho, J. Cho y S.-J. Shin, «Playback Latency Reduction for Internet Live Video Services in CDN-P2P Hybrid Architecture,» de *Media Solution Center, Samsung Electronics Co., Ltd.*, Suwon, 2010.
- [15] BlackBox, «BlackBox,» 20 Julio 2018. [En línea]. Available: <https://www.blackbox.com.mx/mx-mx/page/40830/Recursos/Technical/black-box-explica/Multimedia/Compresion-de-video-H264>. [Último acceso: 22 Mayo 2021].

- [16] R. Catro, «WIKIVERSUS,» 25 Julio 2020. [En línea]. Available: <https://www.wikiversus.com/fotografia-y-video/codecs-h-265-vs-vp9/>. [Último acceso: 22 Mayo 2021].
- [17] C. Cano y M. Raffo, «Arquitectura de alta frecuencia de un filtro de escalabilidad para sobremuestreo de imágenes en factor 2 sobre una FPGA,» 2013.
- [18] J. Alvaro, «Formatos Multimedia: Codificación y Empaquetado,» 2014. [En línea]. Available: <https://edu.fauno.org/R.multimedia.formatos.html>. [Último acceso: 22 Mayo 2021].
- [19] UMA, «Herramientas web para la enseñanza de protocolos de comunicación,» evirtual, [En línea]. Available: <https://neo.lcc.uma.es/evirtual/cdd/tutorial/presentacion/mpeg.html>. [Último acceso: 5 Junio 2021].
- [20] B. Jedari, G. Premsankar, G. Illahi, M. DiFrancesco, A. Mehrabi y A. Ylä-Jääski, «Video Caching, Analytics, and Delivery at the,» *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. 23, nº 1, pp. 431-471, 2021.
- [21] D. Matus, «digitaltrends,» 18 Noviembre 2020. [En línea]. Available: <https://es.digitaltrends.com/entretenimiento/tv-streaming-comparativa-servicios/>. [Último acceso: 22 Mayo 2021].
- [22] M. Wilbert, «HLS vs. MPEG-DASH: A Live Streaming Protocol Comparison for 2021,» datacast, 3 Mayo 2021. [En línea]. Available: <https://www.dacast.com/blog/mpeg-dash-vs-hls-what-you-should-know/>. [Último acceso: 5 Mayo 2021].
- [23] S. Nacakli y A. Tekalp, «Controlling P2P-CDN Live Streaming Services at SDN-enabled Multi-Access Edge Datacenters,» *Fellow,IEEE*, vol. 10, nº Y, pp. 1-12, 2020.

Glosario de Términos

Chunk: elemento más pequeño en el que es dividida una ventana de video.

Ventana: elemento en el que se fragmenta un archivo de video, para que se distribuya de mejor manera.

Peer: usuario conectado al servicio de video en vivo.

Seeder: usuario que ya ha terminado de descargar el archivo de video en su totalidad o gran parte de este. Comparte los recursos que el tiene disponibles con los demás usuarios conectados al servicio.

Leecher: usuario que está consumiendo recursos de sus vecinos, este usuario no cuenta con recursos disponibles para compartir con sus vecinos.

Vecino: son los usuarios más cercanos que tiene un *peer*.

Red P2P: arquitectura de red *peer to peer*, en este tipo de red los usuarios se comportan tanto como consumidores como servidores, es decir, consumen y comparten recursos.

Red CDN: arquitectura de red de distribución de contenido, en este tipo de red el servidor principal distribuye su contenido a otros servidores ubicados en diferentes ubicaciones geográficas.

Red Híbrida CDN-P2P: arquitectura de red resultante de fusionar la red *P2P* y la red *CDN*.

Server: servidor donde se encuentra almacenado el contenido que se va a distribuir sobre la red.

Tracker: es el servidor encargado de la señalización, es decir, este informa a los *peers* de cuál de sus vecinos debe consumir los recursos.

Live Streaming: transmisión de contenido en vivo. El contenido es generado de manera simultánea a su transmisión.