

**INSTITUTO POLITÉCNICO NACIONAL**

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS

***Proyecto Terminal***

**“Modelado matemático, basado en cadenas de Markov, para servicios de video en vivo soportados por redes híbridas P2P-CDN”**

Que para obtener el título de

**“Ingeniero en Telemática”**

P r e s e n t a:

**Ulises Muñoz Ruiz**

**José Manuel Ortiz Islas**

Asesor:

**Dr. Noé Torres Cruz**

**Ciudad de México. Junio, 2022**

UPIITA

UPIITA

**INSTITUTO POLITÉCNICO NACIONAL**

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS

***Proyecto Terminal***

**“Modelado matemático, basado en cadenas de Markov, para servicios de video en vivo soportados por redes híbridas P2P-CDN”**

Que para obtener el título de:

**“Ingeniero en Telemática”**

P r e s e n t a:

Ulises Muñoz Ruiz

José Manuel Ortiz Islas

Asesor:

Dr. Noé Torres Cruz

|  |  |  |
| --- | --- | --- |
|  | Asesor con título |  |

|  |  |  |
| --- | --- | --- |
| Presidente del Jurado: |  | Secretario del Jurado: |
| Dr. Víctor Barrera Figueroa |  | Dra. Iclia Villordo Jimenez |
| Presidente con título |  | Profesor con título |

# 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

[Resumen 3](#_Toc105690499)

[Capítulo 1. Introducción 5](#_Toc105690500)

[1.1 Introducción 5](#_Toc105690501)

[1.2 Planteamiento del Problema 9](#_Toc105690502)

[1.3 Propuesta de Solución 11](#_Toc105690503)

[1.4 Alcances 14](#_Toc105690504)

[1.5 Justificación 15](#_Toc105690505)

[1.6 Metodología 16](#_Toc105690506)

[1.7 Objetivos 19](#_Toc105690507)

[1.7.1 Objetivo General 19](#_Toc105690508)

[1.7.2 Objetivos Específicos 19](#_Toc105690509)

[Capítulo 2. Estado del Arte 19](#_Toc105690510)

[2.1 Estado del arte del modelo de servicios de video con cadenas de Makov 20](#_Toc105690511)

[2.2 Estado del arte de esquemas de asignación de recursos para servicios de video 22](#_Toc105690512)

[Capítulo 3. Marco Teórico 24](#_Toc105690513)

[3.1 Video en Vivo 24](#_Toc105690514)

[3.1.1 Formatos de codificación 24](#_Toc105690515)

[3.1.2 Formatos de empaquetamiento 26](#_Toc105690516)

[3.1.3 Servicios de video en vivo 27](#_Toc105690517)

[3.1.4 Protocolos de arquitectura y distribución de video en vivo por internet 29](#_Toc105690518)

[3.2 Redes *CDN-P2P* 31](#_Toc105690519)

[3.3 Cadenas de Markov 34](#_Toc105690520)

[3.4 Servicios de video sobre redes híbridas *CDN-P2P* 35](#_Toc105690521)

[Capítulo 4. Análisis 37](#_Toc105690522)

[Análisis Cadena de Markov 37](#_Toc105690523)

[Análisis Esquemas de Asignación de Recursos 41](#_Toc105690524)

[Capítulo 5. Diseño 47](#_Toc105690525)

[Diseño de la cadena de Markov 47](#_Toc105690526)

[Diseño del esquema de asignación de recursos 57](#_Toc105690527)

[Conclusiones 60](#_Toc105690528)

[Referencias 60](#_Toc105690529)

[Figura 1. Capas de OTT 6](#_Toc105958024)

[Figura 2. Aumento de personas en la plataforma twich 7](#_Toc105958025)

[Figura 3. Clasificación de plataformas Live Streaming 8](#_Toc105958026)

[Figura 4. Descarga de una ventana de video 12](#_Toc105958027)

[Figura 5. Diagrama a bloques del códec H264 [17] 26](#_Toc105958028)

[Figura 6. Diagrama a bloques del formato de codificación mpeg [19] 27](#_Toc105958029)

[Figura 7. Escenario de video Edge-C3 en redes inalámbricas [20] 31](#_Toc105958030)

[Figura 8. Clasificación de peers en un escenario básico [6] 33](#_Toc105958031)

[Figura 9. Cadena de Markov Unidimensional Erlang-B 38](#_Toc105958032)

[Figura 10. Transición en el estado de la cadena de Markov Erlang-B 38](#_Toc105958033)

[Figura 11. Diagrama de Flujo de Solución Matemática para Erlang-B 40](#_Toc105958034)

[Figura 12. Gráfica de Solución Matemática para Erlang-B 41](#_Toc105958035)

[Figura 13. Estructura del archivo de video en vivo 48](#_Toc105958036)

[Figura 14. Cadena de Markov de un sistema de transmisión de video en vivo 50](#_Toc105958037)

[Figura 15. Diagrama de flujo de solución matemática para la cadena de Markov de un sistema de video en vivo 54](#_Toc105958038)

[Figura 16. Diagrama para mín=ENTran ó EnTRep 55](#_Toc105958039)

[Figura 17. Diagrama para mín=ENTAb 56](#_Toc105958040)

[Figura 18. Obtención de recursos para la población en la ventana 59](#_Toc105958041)

[Tabla 1. Atributos de los videos [20] 28](#_Toc105958042)

[Tabla 2. Características del video en vivo [21] [22] 28](#_Toc105958043)

[Tabla 3. Características principales de esquemas de asignación de recursos investigados 44](#_Toc105958044)

[Tabla 4. Casos de desconexión de un peer 50](#_Toc105958045)

# Capítulo 1. Introducción

## 1.1 Introducción

En la actualidad existen dos formas de transmitir video mediante 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].

Diagrama

Descripción generada automáticamente

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 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 en tiempo real. Esto permite a los usuarios visualizar diversos eventos masivos a distancia (por ejemplo, conciertos, premiaciones, etc.), cuando su ubicación geográfica no les permite estar presentes 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% en el uso de las plataformas de *live streaming*. En el siguiente gráfico podemos observar el aumento en miles de personas referente al uso de la plataforma de twich en la categoría de música en vivo.

Figura 2. Aumento de personas en la plataforma twich

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 figura 2 se pueden observar algunas de las plataformas que brindan servicio de video en vivo.

Diagrama

Descripción generada automáticamente

Figura 3. Clasificación de plataformas Live Streaming

Los servicios de *live streaming* pueden ser distribuidos 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 esta razón, 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* y 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 de este 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: el capítulo 1 está orientada a exponer el problema que se va a abordar en el proyecto, así como las limitaciones, justificaciones y objetivos de este. En el capítulo 2 se abordan los trabajos relacionados a este proyecto. En el capítulo 3, se proporciona la teoría necesaria para comprender la terminología empleada. Después en el capítulo 4 se realiza el análisis de problemáticas similares a la abordada en este proyecto. El capítulo 5 proporciona el diseño de la solución a implementar para la cadena de Markov y el diseño del nuevo esquema de asignación de recursos bajo una distribución uniforme. Posteriormente se establecen las conclusiones generales de este trabajo de investigación. Por último, se muestran las referencias utilizadas para desarrollar el presente proyecto.

## 1.2 Planteamiento del Problema

Actualmente las tendencias de distribución de contenido apuntan al servicio de transmisión de video en vivo 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, y se 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 este. 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 servicios de video basados en 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 tiempo real. 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.

En *VoD* el video es estático y por lo tanto las tasas de descarga y reproducción se mantienen de cierto modo constantes, sin embargo, en *live streaming* la tasa de reproducción está altamente ligada con la velocidad a la que se genera el video (ocurre el evento) y por lo tanto el usuario debe reproducir y descargar a esa velocidad para mantenerse en tiempo real.

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* que consume el servicio de video en vivo y que además permita el diseño de un nuevo esquema de asignación de recursos?

## 1.3 Propuesta de Solución

Con este proyecto lo que se busca es modelar el servicio de video en vivo a través de una red híbrida *P2P-CDN* 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 señales de audio y video empaquetados en un solo archivo, debido a que se desea que el usuario tenga la mayor calidad de servicio y experiencia en la visualización del contenido y tomando en cuenta que este es generado en fragmentos llamados *chunks* se propone agrupar estos fragmentos en segmentos más grandes a los cuales denominaremos ventanas.

Esta agrupación 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 de diferentes *peers* como se muestra en la figura 4.

Diagrama

Descripción generada automáticamente

Figura 4. Descarga de una ventana de video

La descarga de las ventanas que componen al video se realizará de acuerdo con el esquema de asignación de recursos que se propone en el presente trabajo ya que con esto se logra reducir el tiempo de descarga y posibles congelamientos en el video.

Una vez que se tiene la segmentación de los fragmentos del video, los usuarios se agruparan de acuerdo con la ventana que se encuentren descargando en ese instante, de esta manera se procede a formar poblaciones de *peers* por ventana de video. Una población de *peers* son los usuarios que están descargando la misma ventana del video en un mismo instante, la población recibe el nombre de acuerdo con la ventana de descarga, es decir, si se encuentran descargando la ventana cero, la población de *peers* en esta ventana se llamara población .

Dentro del sistema a desarrollar existen cuatro sucesos principales que tienen ocurrencia dependiente del comportamiento de los *peers* y los cuales afectan directamente al estado de la cadena de Markov. Los sucesos o eventos son los siguientes:

1. La llegada o conexión de un nuevo usuario al sistema, este suceso será cuantificado por una tasa de arribo llamada lambda .
2. Transferencia de un peer a la venta subsecuente, la cual ocurre cuando un *peer* en la población termina de descargar el contenido de la ventana y pasa a la ventana . Este suceso ocurre a una tasa tao
3. Transferencia de un peer a la venta inferior inmediata, la cual ocurre cuando un *peer* no descarga el contenido de video a la misma tasa que realiza la visualización de este. La captura del evento en tiempo real sigue su curso, por lo tanto, el usuario se atrasa y es transferido a la ventana inferior. Si está en la ventana cambiará a la ventana , este evento es cuantificado por la tasa
4. Abandono del sistema, este suceso ocurre cuando por alguna circunstancia un usuario sale del evento antes de que esta finalice. Este evento será estimado por la tasa de abandono .

Por otro lado, el esquema de asignación de recursos definido en este proyecto tiene como principal objetivo aprovechar al máximo los recursos provenientes de la red *P2P*, para lo cual se establece que los *peers* podrán obtener recursos para descargar el video de *peers* que tengan un mayor progreso en la descarga del contenido.

El diseño del esquema de asignación de recursos será el resultado del análisis de esquemas reportados en la literatura. Del cumulo de esquemas investigados se elegirán dos esquemas de asignación de recursos adicionales al esquema de asignación de recursos de distribución uniforme.

En primera instancia el esquema a desarrollar se realizará siguiendo el eje de funcionamiento del esquema de asignación de recursos de distribución uniforme, el cual indica que un usuario puede obtener recursos de usuarios ubicados en ventanas superiores a la que este está descargando.

En ese mismo sentido, se obtendrá una expresión de acuerdo con el esquema de distribución uniforme siguiendo los parámetros y limitaciones propios de un servicio de video en vivo. De forma general, este se expresa abiertamente que el esquema de distribución uniforme no presenta una asignación de recursos óptimas, debido a que los usuarios reciben los recursos en función del tamaño de su población.

Lo descrito anteriormente puede generar que algunas poblaciones de *peers* tengan recursos disponibles de sobra mientras que otras poblaciones se encuentran bajo condiciones de penuria en cuanto a recursos se refiere. A pesar de esto, el análisis basado en la distribución uniforme resulta de gran ayuda para comprender como se puede delimitar el aceeso a los recursos de tal forma que el desempeño del sistema sea óptimo.

Además, este análisis facilita la comprensión de la asignación de recursos. Es decir, al estudiar el esquema de asignación de recursos de distribución uniforme y aplicarlo a servicios de video en vivo se comprende la forma en que los recursos son distribuidos dentro de la red y la razón de esta distribución.

De manera preliminar, el diseño del esquema de asignación de recursos para servicios de video en vivo también se realizará tomando como base los esquemas de asignación de recursos: Esquema Q ventanas hacia atrás y DVPG.

A lo largo del desarrollo del presente proyecto y una vez analizados los esquemas de asignación de recursos se establecerá concretamente las bases que se tomaron para el desarrollo de las expresiones descritas en este documento y las bases que se tomarán para el desarrollo de nuevas expresiones como actividades de Proyecto Terminal II.

Debido a que la simulación de este tipo de sistemas es realmente compleja y se requiere de diversos equipos tecnológicos para validar los resultados el presente proyecto se limita a evaluar matemáticamente la cadena de Markov que representa al sistema y a evaluar el esquema de asignación de recursos propuesto para servicios de video en vivo. Dichas evaluaciones se realizarán tomando parámetros de operación utilizados en sistemas de distribución de video reales.

## 1.4 Alcances

Este proyecto se limita a definir una cadena de Markov que represente el comportamiento de los usuarios a lo largo de una transmisión de video en vivo sobre a través de una red hibrida *P2P-*CDN. En dicha cadena se observa de igual forma la relación que tienen los usuarios con el archivo de video y la composición de este. Finalmente, en la cadena se muestran los eventos que generan un cambio en su estado al momento de ocurrir, así como la tasa a la cual ocurre alguno de estos eventos.

El desarrollo de la cadena se realiza de forma cuasi paralela al diseño de un nuevo esquema de asignación de recursos. En primera instancia se considera al esquema de asignación de recursos tradicional uniforme para plasmar el comportamiento de los usuarios y la forma en la cual estos consumen recursos para poder descargar y visualizar el video en vivo.

Posteriormente se plantea trabajar la cadena definida bajos el esquema uniforme bajo un nuevo esquema que se propondrá en el presente trabajo. Dicho esquema se diseña de acuerdo con esquemas de asignación de recursos que se han investigado hasta el momento, pero con la diferencia de abstraer el método que emplean para asignar los recursos y adaptarlo al concepto de *live streaming*.

El estado de la cadena de Markov definida para servicios de video en vivo considera que un instante de tiempo ocurre un único evento por parte de los usuarios (conexión, desconexión o transferencia).

Las implementaciones de la cadena de Markov y del nuevo esquema de asignación de recursos que se propondrá en este proyecto se realizarán empleando parámetros (tasa de subida, tasa de descarga, tasa de desconexión, etc.) que se han reportado en trabajos de investigación relacionados a los servicios de video y los esquemas de asignación de recursos. Dichas evaluaciones se espera sean correctas debido a que las condiciones de implementación han sido utilizadas en sistemas reales para este tipo de servicios.

La solución matemática de la cadena de Markov y del esquema de asignación de recursos se implementará en el ambiente de desarrollo Matlab, con la finalidad de obtener la cantidad de usuarios promedio situada en cada una de las ventanas que componen al video.

Estos resultados serán traducidos a tasas de subida y descarga requeridos por el sistema para la distribución de contenido en vivo y de esta forma conocer el desempeño general y las condiciones que presenta el sistema.

Debido a que un archivo de video se produce por pequeños *frames,* a los cuales se les denomina *chunk* y estos se agrupan en ventanas, un usuario conectado al sistema, conocido a partir de su conexión como *peer*, podrá obtener *chunks* de múltiples *peers* que tengas la capacidad de compartir el contenido.

El sistema que se desarrolla considera que, al momento de la conexión, un *peer* visualizará el contenido en baja calidad hasta que se obtenga parte significativa del archivo de video, con el propósito de que se cree *buffer* y evitar congelamientos en la reproducción o estancamiento en la descarga del video.

En el supuesto de que los resultados no sean los esperados se debe modificar el esquema propuesto con el objeto de obtener una mejor calidad de servicio en comparación con los esquemas utilizados en ambientes reales. De igual manera, la cadena de Markov que modele el comportamiento del sistema, deberá ser replanteada bajo las condiciones que arroje el nuevo esquema de asignación de recursos.

Este proyecto no considera la implementación de un simulador para este modelo, sin embargo, se presentarán las estadísticas de consumo y desempeño del sistema como ya se ha mencionado en función de parámetros en sistemas reales.

## 1.5 Justificación

A raíz de la pandemia por SARS-CoV-2 el consumo de servicios de video en vivo presento un incremento respecto al consumo de *live streaming* en años anteriores. Actualmente múltiples plataformas de distribución de video y redes sociales disponen de este servicio para compartir eventos del día a día en tiempo real. Este servicio permite tener acceso a contenido generado de forma simultánea a su distribución por la red, desde cualquier ubicación geográfica del planeta. *Live streaming* es un servicio que se distribuye en redes del tipo tradicional (cliente-servidor) y redes híbridas (*P2P-CDN*).

Existen diversos modelos para servicios de video bajo demanda (*VoD*), sin embargo, no se tiene conocimiento acerca de un modelo que represente los servicios de video en vivo. Por esta razón es que se ha decido realizar el modelado de este tipo de servicio mediante cadenas de Markov. El análisis de los eventos, sucesos y comportamiento por parte de los usuarios en el consumo de *live streaming* es análogo a lo que sucede en el ambiente del video bajo demanda, pero con la diferencia de que *live streaming* es un archivo dinámico y no estático como en el caso de *VoD*.

Por otra parte, también existen múltiples esquemas de asignación de recursos aplicados a *VoD* y no se conoce algún esquema propio para video en vivo. En este proyecto se han resaltado los esquemas con un óptimo desempeño para servicios de video bajo demanda con el fin de conocer los parámetros que emplean para la distribución de recursos entre los usuarios y analizar el desempeño que presentan. Este análisis permite desarrollar un esquema de asignación de recursos funcional con un óptimo desempeño para servicios de video en vivo.

Por lo descrito anteriormente es que se decide desarrollar en este proyecto el modelado de servicios de video en vivo mediante cadenas de Markov y desarrollar un esquema de asignación de recursos propio para los servicios de video en vivo sobre una red híbrida *P2P-CDN*. El modelado mediante cadenas de Markov permitirá conocer el comportamiento de los usuarios al momento de consumir un servicio de video y su repercusión en la dinámica del sistema que lo transmite.

A su vez este modelo permite conocer las tasas de operación del sistema y su relación con el desempeño general del mismo. Así mismo, se podrán obtener estadísticas de consumo y desempeño del sistema con la finalidad de buscar que estas sean óptimas.

Finalmente se busca que el esquema de asignación de recursos realice la distribución de estos de forma equitativa entre los usuarios que consumen el video en vivo, para que estos puedan visualizar el contenido con buenas *QoE* y *QoS*. Por otro lado, este esquema también busca obtener mayor proporción de recursos de la red *P2P* y disminuir el consumo de recursos provenientes de la red *CDN*.

## 1.6 Metodología

La metodología que se optó implementar para el desarrollo del presente proyecto es *SCRUM*, debido a que esta se caracteriza por ser una metodología basada en una estructura de desarrollo incremental, lo cual quiere decir, que cualquier ciclo de desarrollo del producto y/o servicio se descompone en pequeños proyectos divididos en distintas etapas: análisis, desarrollo, implementación y evaluación. En la etapa de desarrollo se realizan interacciones del proceso o *sprint*, es decir, entregas regulares y parciales del producto final.

De acuerdo con la complejidad de este proyecto y el número de desarrollados, esta metodología se adapta de forma adecuada ya que permite abordar proyectos complejos que exigen una flexibilidad y una rapidez esencial al momento de ejecutar los resultados.

*SCRUM* está orientada a gestionar y normalizar los errores que se puedan producir en procesos de desarrollo de larga duración, a través de, reuniones frecuentes para asegurar el cumplimiento de los objetivos establecidos. Para este proyecto esta característica ha sido fundamental al momento de programar reuniones para debatir conceptos, establecer técnicas de trabajo, definir objetivos e intercambiar experiencias personales y de esta forma fomentar la proactividad.

Finalmente, *SCRUM* aporta al proyecto: innovación, flexibilidad, competitividad y productividad.

Siguiendo la metodología antes menciona se han trazado las siguientes fases para el desarrollo de este proyecto:

**Análisis**

En esta fase del proyecto se realizaron las siguientes actividades:

* Investigación sobre los conceptos clave de archivos de video
* Investigación sobre las cadenas de Markov
* Investigación sobre conceptos de arquitecturas de red
* Investigación acerca del modelado de servicios de video mediante cadenas de Markov
* Investigación de esquemas de asignación de recursos
* Análisis de modelos para servicios de video bajo demanda (*VoD*)
* Análisis de esquemas de asignación de recursos
* Análisis de una cadena de Markov unidimensional

Con base en las actividades antes mencionadas se redactaron las siguientes secciones del presente documento: Introducción, planteamiento del problema, propuesta de solución, alcances, objetivos y marco teórico.

**Diseño**

En esta etapa del proyecto se realizaron las siguientes actividades:

* Diseño de la cadena de Markov
* Diseño de la solución matemática de la cadena
* Bosquejo del nuevo esquema de asignación de recursos

Para el diseño de la cadena de Markov que representa los servicios de video en vivo distribuidos sobre una red híbrida *P2P-CDN* se definieron en primer lugar los eventos (comportamiento de los *peers*)que afectan directamente al estado de la cadena y producen un cambio en este.

Posteriormente se definió la estructura que debe tener el archivo de video y la clasificación de los *peers* una vez que están conectados al sistema. Además, se estableció la nomenclatura que representa las tasas de ocurrencia de un evento por parte de los usuarios.

Finalmente se identificaron los cambios que debe tener el estado de la cadena de Markov una vez ocurrido un evento.

En el diagramado de la solución matemática se describió en forma de diagrama de flujo el procedimiento que se debe seguir para implementar la cadena de Markov, es decir, se estableció el algoritmo que se deberá implementar para simular la ocurrencia de eventos por parte de los usuarios dentro de un sistema de video en vivo para poder obtener estadísticas de consumo y desempeño de este basado en las transiciones del estado de la cadena.

Para el bosquejo del nuevo esquema de asignación de recursos se analizaron distintos esquemas de asignación de recursos enfocados al servicio de video bajo demanda y se identificaron las principales diferencias entre este servicio y el de *live streaming* para de esta forma poder establecer un esquema a fin para el servicio de video en vivo.

**Implementación**

En esta etapa del proyecto se desarrollarán las siguientes actividades:

* Implementación del diagrama de flujo de la solución matemática de la cadena de Markov: esta actividad se refiere a implementar los eventos por parte de los usuarios (conexión, desconexión o transferencia) en un IDE que permita obtener estadísticas de consumo y desempeño significativas para la solución de la cadena y el diseño de un nuevo esquema de asignación de recursos.
* Implementación del nuevo esquema de asignación de recursos
* Evaluación del esquema propuesto en este proyecto frente a esquemas desarrollados en otros trabajos de investigación.
* Corrección de errores de implementación como lógica de programación y sintaxis.

**Evaluación**

Finalmente, en esta etapa, se realizarán pruebas de las implementaciones de la cadena de Markov y el nuevo esquema de asignación de recursos para obtener estadísticas del funcionamiento del sistema de servicio en vivo. De acuerdo con los resultados obtenidos en esta etapa se realizarán modificaciones, en caso de ser necesarias, para obtener resultados que permitan al usuario final visualizar el contenido en vivo con *QoE* y *QoS* optimas.

En el presente documento únicamente se han desarrollado actividades correspondientes a las etapas de análisis y diseño antes mencionadas, debido a que son actividades propias de Proyecto Terminal I.

En cuanto a las actividades de las etapas de implementación y evaluación se realizarán en el proceso de Proyecto Terminal II.

## 1.7 Objetivos

### 1.7.1 Objetivo General

Desarrollar un modelo matemático, basado en cadenas de Markov que describa el comportamiento de un sistema de video en vivo, soportado por redes híbridas *CDN-P2P,* con la finalidad de diseñar un esquema eficiente de asignación de recursos para este tipo de sistemas.

### 1.7.2 Objetivos Específicos

Plantear cadenas de Markov basadas 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 las cadenas 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.

# Capítulo 2. Estado del Arte

En este capítulo se presentan diversas investigaciones relacionadas al análisis de servicios de video, de manera particular servicios de video bajo demanda *(VoD)*, y al análisis de esquemas de asignación de recursos dentro de un sistema para este tipo de servicios.

## 2.1 Estado del arte del modelo de servicios de video con cadenas de Makov

En esta sección se introducen investigaciones que muestran gran relación con el análisis de servicios de video bajo *demanda (VoD)* *mediante* cadenas de Markov y que son tomados como base para el desarrollo del análisis de servicios de video en vivo mediante cadenas de Markov en el presente proyecto. Esto debido a que hasta este momento no se han encontrado en la literatura modelos basados en cadenas de Markovpara servicios de video en vivo.

En [8] los autores rescatan las principales características de las redes *CDN* y *P2P* de manera individual y de forma híbrida. En este trabajo se realiza un modelo híbrido *CDN-P2P* sobre la web con la finalidad de disminuir las peticiones realizadas a los servidores *CDN*; se menciona que el *streaming* de video en vivo es una tecnología que permite a un usuario denominado emisor generar un video que se transmite a otros usuarios en tiempo real.

En este trabajo se mencionan algunas ventajas de las redes *CDN* y *P2P*. De la *CDN* explican que tiene una excelente calidad en el servicio y costos de implementación y mantenimiento relativamente altos. Por otro lado, *P2P* tiene una mejor escalabilidad y menores costos de implementación. El modelo propuesto en este trabajo tiene 2 capas fundamentales, la capa *CDN* y la capa *P2P*. En ese modelo el usuario final se sitúa directamente en la capa *P2P* y en caso de no encontrar los recursos necesarios en su nivel los solicitará al nivel de arriba (*CDN*).

En [9] se compara y evalúa el rendimiento de la red *CDN* frente a la red *P2P*; el punto clave de la *CDN* es replicar el contenido del servidor original en un cache local y de ahí distribuir el contenido a los clientes. Mientras que en la arquitectura *P2P* la distribución se aplica del lado de la red del cliente, ya que estos se convierten en asociados activos al proporcionar el contenido que están recibiendo a otros clientes.

Los trabajos relacionados a *CDN, P2P y CDN hybrid P2P* basados en video en vivo se enfocan en problemas específicos como: reducir el retraso de inicio, distorsión de la transmisión, perdida de paquetes y costo de ancho de banda.

En [10] los autores desarrollan un modelo de transmisión de video sobre una red híbrida *CDN-P2P* para lograr estabilidad y escalabilidad en la difusión de contenido. Aunado a esto mencionan que el modelo cuenta también con otros componentes (*SDN, NSP, VCP*) que permiten tener mejores parámetros en la visualización, la experiencia y el servicio. El modelo al ser basado en *SDN* permite que todos los usuarios puedan aprovechar los recursos disponibles en la *CDN-P2P* de forma inteligente para tener mejor calidad de experiencia y de servicio. Por otro lado, con *NSP* y *VCP* se logra tener mayores recursos en la periferia de un usuario y con ello se logra un mejor rendimiento en la red. De igual forma en este trabajo los autores mencionan los algoritmos empleados para clasificar a los *peers* en diversos grupos y con esto verificar que los recursos sean asignados equitativamente.

En [11] se proporciona un sistema de transmisión hibrido (Video en vivo y *VoD*), basado en codificación de video escalable *SVC* (*Scalable Video Coding*). El sistema consiste en un transcodificador en línea *SVC* diseñado para soportar otros formatos de video y transcodificarlo a *SVC*. En este trabajo se propone un servidor de *streaming* basado en *SVC* para hacer paquetes tipo *RTP* y distribuirlos dentro de la red. El servidor *P2P* recibe los paquetes desde el servidor *streaming* y se encarga de distribuirlos a los *peers* (empleado en el sistema en vivo). Además, existe un *tracker* con el fin de mantener la información y el estado de todos los *peers*, para clasificarlos en *seed* *peer* y common *peer* dependiendo de la carga y descarga que estén realizando. Finalmente se menciona que el reproductor *SVC* únicamente está disponible para PC’s desarrolladas como un complemento en ActiveX y teléfonos inteligentes. Con este modelo se logra reducir la probabilidad de perdida y retraso de paquetes.

En [7] se establece un modelo para la distribución de video bajo demanda en redes heterogéneas 5G. En este trabajo se aborda un modelo fluido (sistema de ecuaciones) que representa el comportamiento de los *peers* conectados al servicio. Para analizar el comportamiento de los *peers* se tomaron en cuenta parámetros como duración del video, ancho de banda de los servidores, tiempo de permanencia, entre otros. Estos datos permitieron evaluar el esquema para observar si la asignación de recursos se realizaba de forma uniforme y equitativa para todos los usuarios. Los resultados fueron comparados con resultados obtenidos en otras investigaciones que utilizan cadenas de Markov para analizar el compartimiento de los *peers* en esos sistemas. De igual forma en este trabajo se establece que sí el modelo es exitoso se puede generar un modelo que reduzca el número de recursos utilizados para su funcionamiento. Los resultados que se obtuvieron se consideran veraces por la similitud que tienen con los resultados basados en cadenas de Markov de otras investigaciones basadas en el funcionamiento real que muestra la plataforma de distribución de contenido YouTube.

En [6] los autores presentan un *framework* analítico para modelar el servicio de video bajo demanda (*VoD*) a través de una red híbrida *CDN-P2P*, su trabajo está basado principalmente en un modelo fluido y en cadenas de Markov. El modelo fluido permite a los autores comprender el comportamiento del sistema en términos de retrasos por parte de los usuarios, congelamiento en la transmisión del contenido, desfase en la reproducción de una ventana y la descarga de la ventana posterior, la clasificación de los *peers*, etc. Por su parte la cadena de Markov se emplea para conocer la tasa de subida y bajada que presenta el sistema, es decir, la cantidad de *peers* que se encuentran en cada grupo (j, k), lo que quiere decir que dichos *peers*, están reproduciendo la ventana j y descargando la ventana k. Al final del trabajo los autores muestran los resultados numéricos obtenidos basándose en diferentes parámetros *QoS*.

En [12] el autor propone un nuevo esquema de asignación de recursos llamado Distribución por Ventanas Priorizadas (DVP). El esquema propuesto en este trabajo demostró ser más eficiente comparado con otros esquemas de asignación de recursos, esto al reducir el ancho de banda solicitado a los servidores. Este trabajo se enfoca en analizar un archivo de video segmentado en ventanas de igual tamaño (red heterogénea) mediante un modelo fluido y complementar con un análisis mediante cadenas de Markov. Para realizar dichos análisis el autor se basa en diversos parámetros de *QoS* y *QoE*. Cabe mencionar que el video analizado es un video bajo demanda (*VoD*) a través de una red *P2P*. Finalmente el autor reporta los resultados obtenidos a partir de diferentes parámetros *QoS* y *QoE* de entrada.

En [5] se menciona que las redes *P2P* son una tecnología clave para la distribución de contenido de video dentro de las próximas generaciones de comunicaciones inalámbricas, incluyendo la quinta generación de sistemas móviles denominada 5G. En este trabajo se fragmenta el contenido de video en ventanas que a su vez están compuestas por partes más pequeñas llamadas *chunks*, además se dice que el tamaño de la primer ventana puede ser diferente a la de las demás que componen el archivo de video, con la finalidad de obtener un equilibrio entre el retraso inicial y la duración de las pausas.

## 2.2 Estado del arte de esquemas de asignación de recursos para servicios de video

En esta sección se presentan trabajos de investigación en los que se ha abordado la asignación de recursos para servicios de video bajo demanda *(VoD)* con la finalidad de conocer el desempeño y funcionamiento de dichos esquemas.

En [7] el autor menciona que los recursos disponibles en una red deben ser asignados correctamente para obtener un buen desempeño del sistema para la descarga y visualización de un video. En este trabajo se implementa una red heterogénea, lo cual indica que todos los usuarios conectados a la red tendrán la misma oportunidad de acceder a los recursos disponibles. Sin embargo, esto implica que los recursos se distribuyen de acuerdo con el esquema de asignación de recursos uniforme.

Por esa razón en este trabajo el autor propone un esquema de asignación de recursos diferente, al cual llamó esquema de Q ventanas hacia atrás, este esquema establece que todos los *peers* que deseen descargar y reproducir un video tendrán la misma oportunidad de consumir recursos de *peers* en Q ventanas hacia adelante y del servidor principal. Con esto se garantiza una correcta distribución de recursos y se evita que algunos usuarios se encuentren en sobre abundancia mientras que otros están en penuria. El parámetro llamado Q, es definido como la cantidad de ventanas superiores desde donde un usuario podrá consumir recursos.

En [13] el autor menciona las ventajas que ofrece una red híbrida *CDN-P2P* para la transmisión, descarga y reproducción de un video bajo demanda *(VoD)*. En este tipo de redes los usuarios, llamados *peer* toman el papel de servidor y consumidor al mismo tiempo, con lo cual se incrementa la cantidad de recursos disponibles para descargar y reproducir un video. En este trabajo se realizan diversas comparaciones de esquemas de asignación de recursos con el fin de obtener puntos de mejora y así proponer un nuevo esquema de asignación de recursos. Como resultado de esta investigación se tiene un esquema de asignación de recursos que tiene como parámetro a Q, que es la cantidad de ventanas hacia arriba que se tomarán en cuenta para obtener recursos, es decir, si un *peer* desea descargar la ventana i, los usuarios que le podrán proporcionar recursos son los usuarios que estén descargando la ventana j siempre y cuando i<j<i+Q. Por otro lado, este esquema es iterativo, es decir, si un video está constituido por N ventanas, el parámetro Q deberá cambiar en función de i+Q<N-1.

En [6] el autor hace una evaluación de diferentes esquemas de asignación de recursos presentes en la distribución de contenido sobre redes *CDN, P2P* y redes híbridas *CDN-P2P* con el fin de obtener el desempeño que estos esquemas muestran frente a la distribución de video bajo demanda y así proponer un nuevo esquema de asignación de recursos con base en el modelo fluido que describe el autor, en el cual se explica el comportamiento de los *peers* y las reglas que se deben seguir para una correcta descarga y reproducción del video.

El autor menciona que en diversos esquemas sólo basta con que un *peer* desee descargar el contenido de una ventana i y el grupo de *peers* en ventanas superiores, es decir, en ventanas j>i le proporcionarán recursos, sin embargo, también menciona que esta práctica inevitablemente genera una disparidad y provoca que algunos *peers* tengan más recursos disponibles para consumir que otros. En el esquema que propone, los recursos provenientes de los servidores *CDN* y de los demás *peers* conectados al sistema se asignan de manera equitativa. En este esquema se establece un parámetro de prioridad denotado por , el cual controla la cantidad de recursos asignada a un grupo de *peers* perteneciente a una ventana i. Finalmente el autor establece que este esquema es mejor en comparación con el esquema uniforme y el esquema PWA.

En [12] el autor propone un nuevo esquema de asignación de recursos llamado Distribución por Ventanas Priorizadas (DVP). La idea principal del esquema propuesto en este trabajo es asignar recursos con mayor prioridad para la descarga de las partes del video que se encuentran en penuria, específicamente a las ventanas altas.

Dentro de la asignación de los recursos por priorización de ventana existen dos tipos de recursos: los provenientes de los servidores y los provenientes de los *seed*s. El uso de DPV para asignar los recursos de los *downloaders* provee un esquema más flexible y eficiente ya que propicia que los *peers* en una ventana en específico aprovechen los recursos de *downloaders* en ventanas inmediatas superiores.

En este esquema los recursos de los *downloaders* en la ventana i son inútiles para *peers* con progresos de descarga mayores, es decir, situados en ventanas j, j>i. Finalmente, el autor realiza una generalización del esquema DPV y lo llama DVPG, esto con el fin de analizar de forma conjunta los recursos provenientes tanto de *seed*s como de *downloaders (peers)*.

En [14] el autor analiza un archivo de video bajo demanda que es dividido en ventanas y consumido por diversos usuarios conectados al sistema. Estos usuarios reciben el nombre de *peers* y requieren consumir recursos para descargar y reproducir el video. En este artículo el autor propone un esquema de asignación de recursos llamado *INUA* (*Immediate-neighbors Uniform Allocation*).

El esquema *INUA* establece que un *peer* podrá obtener recursos de *peers* con un progreso de descarga mayor, es decir, un *peer* que se encuentra descargando la ventana j, podrá obtener recursos de usuarios en ventanas i, i>j. Al conjunto de ventanas con mayor progreso de descarga se le llama Uj y es de longitud Mj. Por lo tanto, las ventanas i que pueden proporcionar recursos están acotadas por los limites j+1< i <j+Mj.

# Capítulo 3. 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.

## 3.1 Video en Vivo

### 3.1.1 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*.

Diagrama

Descripción generada automáticamente

Figura 5. Diagrama a bloques del códec H264 [17]

### 3.1.2 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

Diagrama

Descripción generada automáticamente

Figura 6. Diagrama a bloques del formato de codificación mpeg [19]

### 3.1.3 Servicios de video en vivo

Un *stream* de contenido multimedia, generalmente llamado video, está compuesto por varios *stream*s simples que poseen un tipo de contenido especifico (audio, video o subtítulos). Cada *stream* simple es codificado por un *códec* en un formato especifico 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 1 se enlistan las características principales de los archivos de video y en la tabla 2 se enlistan las características principales de un video en vivo.

|  |  |
| --- | --- |
| 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 1. Atributos de los videos [20]

|  |  |
| --- | --- |
| Característica | Descripción |
| Tiempo real | El evento es capturado a través de un dispositivo y transmitido por la red al mismo tiempo. |
| Interacción en tiempo real | En caso de que los usuarios puedan realizar preguntas y/u observaciones, estas ocurren a lo largo de la transmisión en vivo. |
| Sentido de pertenencia | La audiencia espectadora experimenta sensación de exclusividad o pertenencia al grupo que tiene acceso al contenido. |
| Base para *VoD* | El video capturado en tiempo real se convierte en *VoD* si se almacena y los usuarios acceden a este tiempo después de su grabación*.* |
| Edición | El video capturado no tiene ediciones gráficas o de sonido, es decir, se transmite tal cual se graba. |
| Manejo | Los usuarios no pueden manipular el video en cuanto pausarlo, adelantarlo o retrasarlo se refiere. |
| Fases | Consta de 3 fases principales: grabación, procesamiento y distribución. |

Tabla 2. Características del video en vivo [21] [22]

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* [23] son las encargadas de transmitir contenido en vivo.

### 3.1.4 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. [24]

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 [24]. 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 7, 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.

Diagrama

Descripción generada automáticamente

Figura 7. Escenario de video Edge-C3 en redes inalámbricas [20]

## 3.2 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 8 se puede observar la clasificación de los *peers* en un plano cartesiano.

Diagrama

Descripción generada automáticamente

Figura 8. 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).

## 3.3 Cadenas de Markov

Generalmente el comportamiento de un sistema, físico o matemático se puede representar describiendo los diferentes estados que puede ocupar e indicando cómo es el cambio entre estos estados. Suponiendo que el sistema que se modela ocupa uno y sólo un estado en cualquier momento del tiempo y su evolución se representa mediante transiciones de estado a estado, se puede decir, que estas transiciones se producen de forma instantánea, es decir, que el paso de un estado a otro no consume tiempo. Si la evolución futura del sistema depende únicamente de su estado actual y no de su historia pasada, el sistema puede representarse mediante un proceso deMarkov. Las aplicaciones de los procesos de Markov se pueden encontrar ampliamente en las ciencias biológicas, físicas y sociales, así como en los negocios y la ingeniería (Libro probabilidad).

Un proceso de Markov es un tipo especial de proceso estocástico. Un proceso estocástico se define como una familia de variables aleatorias . En otras palabras, cada es una variable aleatoria y se define en un espacio de probabilidad. El parámetro suele representar el tiempo, por lo que denota el valor asumido por la variable aleatoria en el momento . se denomina conjunto de índices o espacio de parámetros y es un subconjunto de . Si el conjunto de índices es discreto, por ejemplo, , entonces tenemos un proceso estocástico de parámetros discretos; de lo contrario, si es continuo, por ejemplo, , llamamos al proceso un proceso estocástico de parámetros continuos.

Los valores que asumen las variables aleatorias se denominan estados. El conjunto de todos los estados posibles forma el espacio de estados del proceso y éste puede ser discreto o continuo. Si el espacio de estados es discreto, el proceso se denomina cadena y los estados suelen identificarse con el conjunto de números naturales , o un subconjunto de este. Un ejemplo de espacio de estado discreto es el número de clientes de un servicio, mientras que un ejemplo de espacio de estado continuo es el tiempo de espera de un cliente.

Para una cadena de Markov de tiempo discreto, su estado se encuentra definido en un conjunto discreto, pero infinito, de tiempos. Las transiciones de un estado a otro sólo pueden ocurrir, o no ocurrir, en estos instantes de tiempo. Por tanto, se puede representar, sin pérdida de generalidad, el conjunto de índices discretos del proceso estocástico subyacente con el conjunto de números naturales . Las observaciones sucesivas definen las variables aleatorias en los pasos de tiempo , respectivamente. Formalmente, una cadena de Markov de tiempo discreto es un proceso estocástico que satisface la siguiente relación, denominada propiedad de Markov [25]:

Para todos los números naturales y todos los estados

Así, el hecho de que el sistema esté en el estado en el paso de tiempo 0, en el estado en el paso de tiempo 1, y así sucesivamente, hasta el hecho de que esté en el estado en el paso de tiempo n-1 es completamente irrelevante. El estado en el que se encuentra el sistema en el paso de tiempo n+1 sólo depende de dónde se encuentre en el paso de tiempo n. El hecho de que la cadena de Markov se encuentre en el estado en el paso de tiempo n es la suma total de toda la información relativa a la historia de la cadena que es relevante para su evolución futura.

En la cadena de Markov, la abundancia (o penuria) depende del estado del sistema en el tiempo de simulación .

Puesto que este estado es dinámico, no se puede asegurar que en cualquier instante de tiempo la cadena esté en abundancia o penuria, sino que una u otra condición tienen cierta probabilidad de ocurrir. Si se supone un punto de operación, tal que, según el modelo fluido, garantice abundancia, pero que está relativamente cerca del punto crítico; entonces, durante la evaluación de la cadena de Markov, existe una probabilidad no despreciable de que el sistema caiga en penuria durante algunos intervalos de tiempo, ya que las condiciones de abundancia en el modelo fluido sólo están dadas en términos de los valores promedio . Lo anterior provoca que se reduzca la tasa de transición hacia ventanas superiores durante estos intervalos.

## 3.4 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 [10]. 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 [10] consta de dos fases:

1. 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.
2. 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*.

# Capítulo 4. Análisis

En este capítulo se analizó un caso particular y elemental de una cadena de Markov que representa un sistema de servicio, donde los usuarios consumen recursos de N servidores y transitan entre los diferentes estados de la cadena. Por otro lado, en este capítulo se aborda el análisis del funcionamiento de esquemas de asignación de recursos reportados en trabajos relacionados.

## Análisis Cadena de Markov

En esta sección se realizó el análisis de una cadena de Markov unidimensional conocida como Erlang-B, con la finalidad de saber cómo evoluciona el estado de una cadena de Markov y la solución matemática que se le puede dar. Posteriormente se realizó un diagrama de flujo para dar solución matemática a la cadena Erlang-B en el IDE Matlab y finalmente se implementó el análisis de esta cadena ya que permite obtener estadísticas de consumo y desempeño de un sistema.

La siguiente cadena de Markov de la figura x representa un sistema dónde se consumen recursos para la distribución de contenido, la cadena que se analizó es una cadena unidimensional (su estado transita de 0 a S y los usuarios se clasifican por una única variable). Así mismo, en la figura x se representan las tasas que provocan un cambio en el estado de la cadena.

En la figura 9 se puede observar que los estados de la cadena se encuentran definidos de 0 a S, donde S es el número de servidores que proporcionan recursos a los usuarios.

Imagen en blanco y negro de un reloj

Descripción generada automáticamente con confianza media

Figura 9. Cadena de Markov Unidimensional Erlang-B

Imagen en blanco y negro de un reloj

Descripción generada automáticamente con confianza media

Figura 10. Transición en el estado de la cadena de Markov Erlang-B

En la figura 10 se representan las posibles transiciones que puede presentar un usuario en la cadena Erlang-B asociadas a su correspondiente tasa de ocurrencia, es decir, un usuario transita al estado superior a tasa o transita al estado inferior a tasa .

Las transiciones de estado son eventos discretos, es decir, en un instante de tiempo sólo puede ocurrir una transición al estado superior o una transición al estado inferior. Estos eventos se implementan generando una variable exponencial negativa con su respectiva tasa y se hace un acumulado del tiempo de estancia por estado para de esta forma obtener estadísticas de consumo.

Una vez analizado el comportamiento de los estados de la cadena Erlang-B se realizó el diagrama de flujo de la solución matemática de esta cadena y su implementación en el IDE Matlab. El código inicia definiendo los parámetros de entrada, es decir, las tasas que modificarán la posición del usuario en el estado de la cadena. Las variables más notables son: tasa de conexión , tasa de desconexión y número de servidores .

Posteriormente, se definen los siguientes vectores de longitud : tiempos de estancia por estado () y arribos por estado ().

En el caso de que la cadena se encuentre en estado “cero” únicamente se genera una variable exponencial negativa a partir de la tasa de conexión y se acumula su valor en tiempo de estancia en la posición y se incrementa en 1 al contador de arribos del estado 0.

En caso de que la cadena se encuentre en otro estado, contenido en , se generan 2 variables exponenciales negativas A y B a partir de la tasa de conexión y la tasa de desconexión respectivamente y se elegirá la de menor valor para determinar que evento ocurrió (conexión o desconexión) y se tienen los siguientes 2 casos:

\* Conexión: Se almacena el valor de A en tiempo de estancia por estado en la posición y el contador se incrementa en 1 en la posición .

\* Desconexión: Se almacena el valor de B en tiempo de estancia por estado en la posición .

La implementación de la solución matemática se realizó conforme al siguiente diagrama de flujo.

Diagrama, Esquemático

Descripción generada automáticamente

Figura 11. Diagrama de Flujo de Solución Matemática para Erlang-B

Los resultados de la implementación se compararon con las probabilidades obtenidas mediante la expresión x y se aproximaron empleando dos técnicas: la primera es la de tiempo promedio de estancia por estado y la segunda mediante la cantidad de arribos por estado.

Figura 12. Gráfica de Solución Matemática para Erlang-B

## Análisis Esquemas de Asignación de Recursos

En esta sección se realizó el análisis de los esquemas de asignación de recursos reportados en los trabajos de investigación relacionados a este proyecto, se realizó una tabla informativa con las principales características de cada uno estos para poder seleccionar aquellos esquemas que son adecuados para los servicios de video en vivo y que serán tomados como base para realizar el bosquejo del esquema de asignación de recursos a proponer en el en el presente proyecto.

En la tabla 3 se describen los esquemas de asignación de recursos analizados previamente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Esquema** | **Descripción** | **Asignación de Recursos** | **Origen de los Recursos** | **Criterio de Asignación de Recursos** |
| **Distribución Uniforme** | Consiste en distribuir los recursos de forma uniforme entre todos los usuarios que presentan un menor progreso en su descarga respecto a los usuarios desde donde se están obteniendo dichos recursos. | Los *downloaders* en la ventana pueden obtener recursos de los *peers* en ventanas siempre y cuando . | \* *Seeds*  \* *Downloaders*  \*Servidores *CDN* | Ventaja . |
| **Q Ventanas hacia atrás** | Consiste en distribuir los recursos variando el número de ventanas. Esta variación influye en la cooperación de todos los *peers* (*seeds* y *downloaders*) y se establece que los recursos provenientes de los servidores se dividen únicamente entre los *downloaders*. | Los *downloaders* ubicados en una ventana en específico podrán tomar recursos de los *peers* ubicados en ventanas inmediatas superiores. | \* *Seeds*  \* *Downloaders*  \*Servidores *CDN* | Q (número de ventanas de restricción) |
| **GPWA** | Establece un parámetro de prioridad para que una ventana reciba recursos de ventanas posteriores. El parámetro establece el grupo de *peers* que recibirá recursos de una ventana en específico. | Los *leeches* y los *seeds* comparten sus recursos con *leeches* con menor progreso en su descarga y los recursos de los servidores CDN se asignarán cuando no existan suficientes recursos en el sistema. | \* *Seeds*  \* *Leeches*  \*Servidores *CDN* | (parámetro para controlar el nivel de prioridad) |
| **DVPG** | Consiste en asignar los recursos de los *downloaders* a *downloaders* con menor progreso en su descarga, sin embargo, tienen mayor prioridad los *downloaders* ubicados en la ventana inmediata inferior para evitar penuria en algunas ventanas del sistema. Los recursos de los *seeds* y los servidores son aginados a *downloaders* en ventanas superiores. | Los recursos son repartidos de forma tal que los *downloaders* en ventanas superiores tengan un mayor ancho de banda asignado y que los recursos de las ventanas inferiores sean aprovechados por *downloders* en ventanas inferiores. | \* *Seeds*  \* *Leeches*  \*Servidores CDN | (parámetro de control de prioridad) |
| **INUA** | Establece que un usuario puede seleccionar aleatoriamente los recursos de *peers* con mayor progreso en su descarga, es decir, de *peers* en ventanas superiores. Sin embargo, lo deberán realizar bajo una condición que limita el rango de ventanas hacia adelante que puede solicitar como servidores. | Los *peers* de una ventana en específico (j) poseen el mismo avance en su descarga y por ende tienen la misma probabilidad de obtener recursos del grupo de ventanas habilitados para asignarles recursos. | \* *Seeds*  \* *Leeches*  \*Servidores *CDN* | M =QX (denota a un peer con mayor progreso en su descarga) |
| **PWD** | Consiste en asignar recursos provenientes de *seeders* y servidores CDN con mayor prioridad a los *leeches* que se encuentran descargando ventanas superiores, esto con la finalidad de que los recursos se distribuyan de una manera más equitativa y el sistema en general se encuentre en condiciones de abundancia. | La cantidad de recursos que se le asignará a los *leeches* será proporcional tanto al número de *leeches* en la ventana i como a la prioridad de ventanas más altas. | \* *Seeds*  \*Servidores *CDN* | (parámetro de control de prioridad) |

Tabla 3. Características principales de esquemas de asignación de recursos investigados

Como resultado de analizar cada uno de los esquemas de asignación de recursos reportados en la tabla 3 se decidió tomar como referencia a los esquemas:

* Esquema de asignación de recursos de distribución uniforme
* Esquema Q ventanas hacia atrás
* Esquema DPVG

Esto gracias a que, a pesar de que son esquemas para servicios de video bajo demanda (*VoD*), tienen características similares que se pueden adaptar y mejorar para servicios de video en vivo.

Con base en el análisis previo de los esquemas de asignación de recursos seleccionados se realizó un seudocódigo que permite comprender el funcionamiento de cada uno de estos.

De forma general se identificó que todos los esquemas presentan el mismo comportamiento una vez que se ha determinado cuales son los *peers* de donde se obtendrán los recursos que utiliza un *downloader* para descargar el archivo de video y por lo tanto son similares a partir de esta condición.

**Esquema de Asignación de Recursos Uniforme**

Inicio

1.- Llega un usuario nuevo al sistema a la ventana

2.- Se define

3.- Repetición de Recolección

Se recolectan todos los *seeds* y *downloaders* en ventanas

Fin de Repetición

4.- Listado de *peers* con recursos disponibles

5.- ¿La tasa real de descarga es igual a la ideal de descarga?

Si: Almacenar los *peers* que asignan recursos y la cantidad de recursos

No: Se asignan recursos, ¿aún hay *peers* en la lista?

Si: Retornar a paso 4

No: Buscar recursos en el *CDN* y repetir condición Si del paso 5

6.- Generar estadísticas

Fin

**Esquema Q ventanas hacia atrás**

Inicio

1.- Llega un usuario nuevo al sistema a la ventana

2.- Se define que es el límite de ventanas

3.- Repetición de Recolección

Se recolectan todos los *peers* en ventanas

Fin de Repetición

4.- Listado de *peers* con recursos disponibles

5.- ¿La tasa real de descarga es igual a la ideal de descarga?

Si: Almacenar los *peers* que asignan recursos y la cantidad de recursos

No: Se asignan recursos, ¿aún hay *peers* en la lista?

Si: Retornar a paso 4

No: Buscar recursos en el *CDN* y repetir condición Si del paso 5

6.- Generar estadísticas

Fin

**Esquema DVPG**

Inicio

1.- Llega un usuario nuevo al sistema a la ventana

2.- Se define que es el parámetro de control

3.- Repetición de Recolección

Se recolectan todos *peers* que presenten con un índice alto

Fin de Repetición

4.- Listado de *peers* con recursos disponibles

5.- ¿La tasa real de descarga es igual a la ideal de descarga?

Si: Almacenar los *peers* que asignan recursos y la cantidad de recursos

No: Se asignan recursos, ¿aún hay *peers* en la lista?

Si: Retornar a paso 4

No: Buscar recursos en el *CDN* y repetir condición Si del paso 5

6.- Generar estadísticas

Fin

Aunado a esto se analizaron las expresiones para asignar recursos de los esquemas Q ventanas hacia atrás y DVPG para encontrar las diferencias para servicios de video bajo demanda y servicios de video en vivo.

***Esquema Q ventanas hacia atrás***

La siguiente expresión determina la tasa promedio necesaria para descargar una ventana del video:

En primer lugar, el esquema de asignación de recursos a proponer en el presente proyecto no considera 2 regiones o zonas donde están distribuidos los *peers* dentro del sistema, por lo tanto, no se tomará a como variables diferentes.

Debido a que no existen usuarios del tipo *seed* el esquema prescindirá del término , que representa los recursos provenientes de *peers* conectados a la ventana .

El término para el desarrollo de este proyecto va a representar los recursos que proporciona la red *CDN.*

Finalmente, considerando que el número de ventanas queda restringido por el parámetro , los límites en la suma de los recursos cambiarán de forma significativa y el parámetro Q (número de ventanas hacia atrás) deberá ser .

**Esquema DPVG**

La siguiente expresión determina la tasa promedio necesaria para descargar una ventana del video:

La expresión anterior se puede adecuar de manera correcta a los servicios de video en vivo debido a que considera tanto recursos de la red *P2P* como de la red *CDN*, además de que considera a todos los *peers* dentro del sistema para que reciban recursos de manera equitativa o correspondiente a su progreso de descarga.

Sin embargo, se debe limitar la expresión anterior al caso de los videos en vivo. Debido a que en una transmisión de video en directo no existen usuarios del tipo *seed* los recursos dentro la red provienen exclusivamente de *peers* conectados a ventanas y de los servidores *CDN.*

Por lo mencionado anteriormente el límite de la suma no llegará hasta N sino que se cambiará esta variable por una , que identifica a la ventana actual *(currently)*. A su vez en el presente proyecto se busca que los usuarios más cercanos a la ventana sean los que reciban con mayor prioridad los recursos provenientes de los servidores *CDN.* Por lo tanto, el término , seguramente será restringido a poblaciones cercanas a la ventana .

El valor de la variable siempre debe corresponder al índice de la ventana a donde pertenece la población que va a recibir los recursos y por lo tanto se debe limitar a .

Finalmente, el parámetro de priorización para el caso de este proyecto puede ser cualquier número real e indicará la prioridad con que una población de *peers* puede acceder a los recursos ofrecidos por las poblaciones en ventanas superiores.

Como se había mencionado anteriormente el esquema de asignación de recursos que se propondrá en este proyecto puede ser la fusión de los 2 esquemas mencionados anteriormente, con los cual quizá vuelvan a cambiar los límites de la suma.

# Capítulo 5. Diseño

En este capítulo se describe el diseño de la cadena de Markov que representa a los servicios de video en vivo y el diseño del esquema de asignación de recursos apropiado para este tipo de servicio.

## Diseño de la cadena de Markov

En esta sección se describe la cadena de Markov que representa el comportamiento de los usuarios una vez que se conectan a un sistema de servicio de video en vivo, de igual forma se explican los posibles estados en los que puede estar la cadena y los eventos que ocasionan una transición en el estado de la cadena de Markov.

En el presente proyecto se supone la distribución de un archivo de video en vivo sobre una red híbrida *P2P-CDN*. Como se ha mencionado anteriormente, los videos son generados por segmentos pequeños llamados *frames*, en este ámbito los denominaremos *chunks*, que es la unidad indivisible de un video.

Con el objeto de que no se tenga una cantidad muy grande de poblaciones de *peers* descargando el contenido, los *chunks* se agruparán en ventanas. Lo anterior se representa en la figura 13.

Diagrama

Descripción generada automáticamente

Figura 13. Estructura del archivo de video en vivo

En la figura 13 se puede observar que el archivo de video está compuesto por ventanas, (*current*). Las ventanas son identificas con un subíndice , para . A su vez dichas ventanas están por N *chunks*.

: Representa la ventana que se genera de forma simultánea a la captura en video del evento en tiempo real.

: Representa la ventana con el máximo retardo respecto a la ventana , es decir, es la última ventana que aún se considera como visualización en tiempo real.

Los *peers* conectados al sistema de transmisión de video en vivo son clasificados de acuerdo con la ventana que se encuentran descargando y además son agrupados en poblaciones por ventana y no se les considera como usuarios individuales dentro del sistema. La clasificación mencionada es la siguiente:

Representa la población de *peers* que se encuentra descargando y reproduciendo la ventana . Para .

Representa la población de *peers* en la ventana y son aquellosque se conectan al sistema una vez que la transmisión de video en vivo ha empezado, por lo tanto, solo pueden descargar el archivo de video para que generen *buffer* y más adelante puedan visualizar la transmisión sin problemas de estancamiento en la descarga o reproducción.

Debido a la naturaleza de los videos en vivo, un *peer* que está descargando la ventana , realiza esta acción a la misma tasa con que es generado el archivo de video . De manera análoga esta tasa indica la velocidad a la cual el sistema demanda que los *peers* realicen la reproducción del video con el fin de no perder sincronía con la ventana .

Con base en el análisis realizado a la cadena Erlang-B se puede obtener el comportamiento que tienen los usuarios en la visualización un video y así identificar los sucesos que pueden ocurrir en la transmisión de un video en vivo y comprender como influyen estos en las poblaciones de cada una de las ventas que componen al archivo de video.

Los sucesos principales que se identificaron a lo largo de la descarga y reproducción de un video son:

* Conexión de un *peer*: Sucede cuando un *peer* se conecta al iniciar la transmisión del video en vivo. El *peer* deberá recibir con mayor prioridad recursos de un servidor *CDN* para poder descargar el video y posteriormente él deberá compartir recursos con *peers* que se conecten tiempo después de iniciada la transmisión. El *peer* no recibirá necesariamente el 100% de los recursos que necesita directamente de un servidor *CDN*, sino que puede recibir algún porcentaje de este y otro tanto de algunos otros *peers*.
* Arribo de un *peer*: Sucede cuando un *peer* se conecta al sistema tiempo después de que inicio la transmisión en vivo. En este caso el *peer* recibirá recursos de acuerdo con algún esquema de asignación de recursos con la finalidad de recibir porciones de recursos que necesita desde diferentes *peers* habilitados para compartir sus recursos.
* Transferencia de un *peer* a la ventana superior: Ocurre cuando un *peer* que estaba descargando la ventana comienza a descargar la ventana . Por lo tanto, ya no forma parte de la población de la ventana y ahora es parte de la población .
* Desconexión de un *peer*: Por diversas razones un *peer* puede interrumpir la visualización de un evento en vivo (mala conexión a internet, pérdida de interés en el contenido, fallas en su dispositivo, etc.) antes de que termine la transmisión. Por lo tanto, si un usuario que se encontraba descargando la ventana al ocurrir este suceso, se decrementará en 1 la población de esa ventana.

De forma general la población de cualquier ventana contenida en se altera por la transferencia o desconexión de un *peer*. Sin embargo, el arribo de un *peer* únicamente ocurre en la ventana , que representa a la ventana con retardos respecto a la ventana actual . Por otro lado, la conexión de un *peer* ocurre únicamente en la ventana puesto que es la única ventana que se ha generado al iniciar la transmisión en vivo.

La desconexión de un *peer* contempla los casos reportados en la tabla x con el fin de comprender la estructura de la distribución del video.

|  |  |
| --- | --- |
| **Caso** | **Descripción** |
| Primero | El *peer* que se desconectó estaba conectado directamente al servidor *CDN* y que no era punto de acceso para un *peer* que llego después de iniciada la transmisión. |
| Segundo | El *peer* que abandona la conexión estaba conectado al servidor *CDN* y a su vez tenia a otro *peer* conectado a él. |
| Tercer | El *peer* que abandona la transmisión estaba conectado a otro *peer* y no tenía ningún *peer* conectado a él. |
| Cuarto | El *peer* que abandona el sistema estaba conectado a otro *peer* y a su tenía otro *peer* conectado a él. |

Tabla 4. Casos de desconexión de un peer

Tomando en cuenta lo descrito anteriormente se realizó el diseño de una cadena de Markov para servicios de video en vivo y las transiciones que presenta en sus estados al ocurrir un suceso por parte de los usuarios conectados a la transmisión del video en vivo.

El sistema de transmisión de contenido en tiempo real se representa con la siguiente cadena de Markov:

Diagrama

Descripción generada automáticamente

Figura 14. Cadena de Markov de un sistema de transmisión de video en vivo

En la figura 14, se observan los sucesos que generan una transición en el estado de la cadena y el cambio que presenta el estado de esta.

En el centro de la figura 14, se observa al vector , que representa al estado de la cadena de Markov. Este vector contiene las poblaciones de *peers* en cada una de las ventanas que componen al archivo de video (desde la ventana hasta la ventana ).

Este vector representa un estado general de la cadena de Markov, es decir, la cadena se encuentra en este estado en cualquier instante de la transmisión de video en vivo. Cada una de las poblaciones , contenidas en este vector estado, tiene un valor aleatorio.

En el sistema a desarrollar en este proyecto se considera que los sucesos son discretos, es decir, ocurre un único evento a la vez (conexión de un usuario, transferencia de un peer a la ventana superior inmediata, transferencia de un peer a la ventana inferior inmediata o desconexión de un *peer* antes de terminar la transmisión).

A continuación, se describen los sucesos que afectan el estado de la cadena de Markov y generan modificaciones en las poblaciones de *peers* en cada una de las ventanas que conforman al archivo de video.

Conexión de un nuevo usuario al sistema: Una vez que inicia la transmisión de video en vivo, diversos usuarios se pueden conectar a la transmisión para visualizar la transmisión en tiempo real. Dicha conexión se realiza a tasa , que representa la tasa de conexión de un usuario en general del sistema.

El estado general cambia cuando se conecta un usuario a tasa y la cadena de Markov transita del estado al estado . Este evento es plasmado en la parte superior de la figura x.

Esta transición se genera por la naturaleza de los videos en vivo, la cual implica que un usuario se debe conectar a la ventana (ventana con máximo retardo) para comenzar a descargar el video, crear *buffer* y con esto evitar congelamientos en la descarga y/o reproducción del contenido.

Transferencia de un *peer* a la ventana inmediata superior: Cualquier población de *peers* correspondiente a una ventana contenida en evoluciona cuando se genera la transferencia de un *peer* a la ventana superior inmediata a tasa , es decir, una vez que un usuario termina de descargar la ventana y comienza a descargar la ventana , abandona la población y se agrega a la población en la ventana .

Este evento provoca una transición en el estado de la cadena de Markov como se describe a continuación:

El evento descrito anteriormente está plasmado en la parte inferior de la figura 14.

: Representa la tasa promedio de transferencia de la ventana . Y se define con la siguiente expresión:

Es la tasa de descarga de un usuario en general del sistema.

Es la tasa de reproducción a la cual debe visualizar usuario general del sistema el video en vivo.

La diferencia establece la tasa efectiva a la cual un usuario se transfiere a la ventana inmediata superior, esta tasa es escalada por con la finalidad de obtener la tasa efectiva a la cuál un usuario es transferido de la ventana a la ventana

Representa a los recursos de descarga efectivos en penuria para la ventana .

Debido a que los *peers* conectados a la ventana únicamente pueden descargar el video (sin reproducirlo), su tasa de transferencia se representa con la siguiente expresión:

Transferencia de un *peer* a la ventana inmediata inferior:Cualquier población de *peers* correspondiente a una ventana contenida en evoluciona cuando se genera la transferencia de un *peer* a la ventana inferior inmediata a tasa , es decir, cuando un usuario por diversas razones (fallas en sus servicios, fallas en sus dispositivos, etc.) deja de descargar el video en vivo a la misma tasa que se está generando el evento en tiempo real, por lo tanto se atrasa en la descarga del video y pasa de la ventana (abandona la población ) y a la ventana (se adiere a la población ).

Este evento provoca una transición en el estado de la cadena de Markov como se describe a continuación:

El evento descrito anteriormente está plasmado en la parte izquierda de la figura 14.

Desconexión de un *peer* antes de terminar la transmisión: Por causas diversas (fallas de conexión, fallas en sus servicios, fallas en sus dispositivos, desinterés en el contenido, etc.) un *peer* que está visualizando la transmisión de video en vivo puede desconectarse.

Entonces, cualquier población de *peers* de la cadena evoluciona cuando se genera la desconexión de un *peer*, a tasa , antes de que finalice la transmisión en vivo. representa la tasa promedio de desconexión de la ventana , es el resultado de multiplicar la población de la ventana por (tasa de desconexión de un *peer* conectado al sistema en general).

Este evento provoca una transición en el estado de la cadena de Markov como se describe a continuación:

Lo anterior quiere decir un *peer* que formaba parte de la población por alguna razón abandono el sistema antes de finalizar la transmisión en vivo, por lo tanto, la población de la ventana se decrementa en 1. Este evento es representado en la sección derecha de la figura 14.

Los *peers* conectados a la ventana son más susceptibles a desconectarse del sistema, en caso de que no logren una buena conexión a internet u óptimas *QoE* o *QoS* por lo tanto, se considera a como la tasa de desconexión de los *peers* pertenecientes a la población .

Una vez diseñada la cadena de Markov y definidos los sucesos que generan un cambio en su estado, se diagramo la solución matemática a implementar para simular la ocurrencia de los sucesos y la transición de estado en la cadena.

*Diagrama

Descripción generada automáticamente*

Figura 15. Diagrama de flujo de solución matemática para la cadena de Markov de un sistema de video en vivo

Figura 16Diagrama

Descripción generada automáticamente. Diagrama para mín=ENTran ó EnTRep

Diagrama

Descripción generada automáticamente

Figura 17. Diagrama para mín=ENTAb

El diagrama de la figura x no contempla en este momento el bloque de esquemas de asignación de recursos, puesto que es una actividad por desarrollar en PT2. Las estadísticas que se obtendrán de la implementación del diagrama se graficarán en función de distintos valores para los parámetros de entrada

## Diseño del esquema de asignación de recursos

En esta sección se desarrolla el bosquejo del esquema de asignación de recursos para servicios de video en vivo. En este diseño se explican las condiciones que debe cumplir el sistema para realizar la asignación de recursos. En esta etapa se ha desarrollado un esquema de asignación uniforme y considerando condición de abundancia en el sistema.

Tomando como base el análisis del comportamiento de los *peers* se plantea desarrollar un esquema de asignación de recursos conveniente para los servicios de video en vivo. Los esquemas presentados en [12] y [7] muestran un óptimo desempeño para la asignación de recursos para servicios de video bajo demanda (*VoD*).

Por lo tanto, el nuevo esquema de asignación de recursos será una variante o mezcla de los dos esquemas mencionados anteriormente. Debido a que se identificaron puntos de mejora para asignar los recursos disponibles dentro de la red y aunado a esto, cabe mencionar que la principal diferencia entre los trabajos que emplean estos esquemas y el presente trabajo es el enfoque que se le dará al esquema.

En el presente trabajo se les dará un enfoque dirigido a los servicios de video en vivo, por lo cual dichos esquemas serán modificados y adaptados para que tengan un buen funcionamiento en este tipo de servicios.

En primer lugar, se analizó la cantidad de recursos necesarios para que la población correspondiente a una ventana pueda descargar de manera satisfactoria el video en vivo y así poder establecer una expresión que permite cuantificar esta cantidad de recursos al cual se le denomino ancho de banda consumido.

De manera general dentro del sistema se tiene la tasa de descarga global que regula el ancho de banda consumido en el proceso de descarga del contenido en vivo, dicha tasa es representada con . Esta tasa permite conocer a qué velocidad promedio un *peer* en general conectado al sistema realiza la descarga del contenido.

Sin embargo, esta tasa no específica la cantidad de recursos necesarios para que la población de una ventana descargue de forma continua el contenido de la ventana posterior.

Por lo tanto, se establece a como la cantidad de recursos (ancho de banda de descarga) requerida por la población en la ventana para descargar y visualizar el video en vivo. La cantidad es el resultado de multiplicar la tasa global , que es la tasa individual promedio de descarga, por la población de *peers* en la ventana .

De igual forma dentro del sistema existe la tasa de reproducción, la cual indica la velocidad a la que se genera el video, de forma inherente, esta tasa también indica la velocidad con la que los usuarios deben reproducir el video para no perder sincronía respecto a la ventana actual Esta tasa es denotada por .

En el presente proyecto se supone que la tasa de reproducción debe ser menor que la tasa de descarga para evitar que un usuario consuma el contenido almacenado en su *buffer*, se estanque en la descarga del video y presente congelamiento en la visualización del contenido.

Una vez que se analizó la cantidad de recursos necesarios para que la población correspondiente a una ventana en específico pudiera descargar el video en vivo se analizó el tamaño de la población de las ventanas superiores para conocer si la cantidad de recursos requeridos puede ser cubierta únicamente por los recursos procedentes de los *peers*.

Dentro del sistema que se desarrolla en el presente proyecto los *peers* conectados a la transmisión en directo reciben también el nombre de *downloaders*, debido a que descargan el archivo de video mientras lo reproducen. Debido a la naturaleza de la red *P2P* los *downloaders*, son parte medular del ancho de banda de subida que requiere el sistema, ya que los *peers* tienen la capacidad de compartir el contenido a otros *peers* que se encuentren en ventanas inferiores dentro del sistema.

Este hecho permite que el servidor no se sature y no se le demande un ancho de banda mayor, por lo tanto, el servidor únicamente se centra en abastecer a los *peers* que están conectados directamente a él y con esto se crea una cadena de transferencia con una buena transmisión en el video en vivo. Sin embargo, el sistema debe estar preparado y tener reservado un espacio del ancho de banda proveniente de la red *CDN* para los posibles eventos que puedan ocurrir con los *peers*. Ya sea que los recursos provenientes de los *peers* sean insuficientes o un *peer* pierda la conexión a la transmisión de video.

El sistema tiene una tasa de subida global que establece el ancho de banda de subida promedio que puede proporcionar un *peer* en el proceso de la transmisión de contenido en vivo, dicha tasa es representada con . Esta tasa permite conocer la velocidad promedio a la que un *peer* conectado al sistema sube la porción del archivo de video que tiene almacenada en su *buffer*.

Sin embargo, no específica la cantidad de recursos que puede proporcionar la población de la ventana a los *peers* en ventanas , para que puedan comenzar el proceso de descarga del archivo de video.

Por lo tanto, se establece a como la cantidad de recursos (ancho de banda de subida) proporcionada por la población en la ventana . Es decir, los *chunks* contenidos en su *buffer* para compartir con otros *peers* en ventanas inferiores. La cantidad es el resultado de multiplicar la tasa global , que es la tasa individual promedio de subida, por la población de *peers* en la ventana .

El objetivo principal de un esquema de asignación de recursos es procurar condiciones de abundancia dentro del sistema, es decir, que los recursos disponibles en la red (ancho de banda de subida) sea mayor o igual al demandado por el sistema (ancho de banda de descarga) y que este a su vez sea proporcionado equitativamente a todos los *peers* conectados al sistema.

Con base en lo descrito hasta este punto de la investigación se decide desarrollar un bosquejo del esquema de asignación de recursos para servicios de video en vivo basado en el esquema [7].

En la figura 18, se observa la forma en que son distribuidos los recursos dentro del sistema de servicio de video en vivo y las expresiones que permiten realizar dicha asignación.

Imagen que contiene objeto

Descripción generada automáticamente

Figura 18. Obtención de recursos para la población en la ventana

En la figura 18 se observa que la población de una ventana cual esta sea puede obtener recursos únicamente de poblaciones situadas en ventanas superiores. Esto debido a que los usuarios en ventanas superiores tienen un mayor progreso en la descarga del contenido en vivo.

Debido a que el esquema de asignación de recursos en esta etapa del proyecto es un esquema uniforme, el número de ventanas de donde provienen los recursos no está limitado, es decir, la población en la ventana pueden obtener recursos de ventanas tal que , como se muestra en la siguiente expresión:

Realizando análisis matemático se observa que y son factores en común de todas las expresiones sumadas, por lo tanto, esa suma se puede resumir y generalizar para cualquier población como se muestra en la siguiente expresión:

# Conclusiones

# 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 comprenhensive 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] | 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,* vol. 10, nº 94, p. 13, 2015. |
| [9] | M. Hassan, C. K. Neng y L. C. Suan, «Performance Analysis of Video Streaming on different HybridCDN & P2P Infrastructure,» MIMOS Berhad, Malaysia, 2021. |
| [10] | 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. |
| [11] | J. Sun, Y. Zhou, Y. Duan y Z. Guo, «A Low-latency Peer-to-Peer Live and VODStreaming System Based on Scalable Video Coding,» Beijing, IEEE, 2014, p. 319. |
| [12] | N. Torres Cruz, *Esquemas de asignación de recursos para servicios de video en redes heterogéneas 5G,* México: IPN, 2019. |
| [13] | V. Pichardo Herrera, *Evaluación de esquemas de asignación de recursos,* México: IPN, 2021. |
| [14] | C. Zhao, J. Zhao, X. Lin y C. Wu, «Capacity of P2P on-demand streaming with simple, robust and decentralized control,» IEEE, Turin, 2013. |
| [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, «Arqeuitectura 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] | V. Rodríguez, «Scenikus blog,» 21 Abril 2021. [En línea]. Available: https://blog.scenikus.com/diferencia-entre-live-streaming-y-vod/. [Último acceso: 23 Mayo 2022]. |
| [22] | M. Willbert, «Live Streaming vs. VOD: Comparing the 10 Best Video Platforms That Do Both,» datacast, 20 Enero 2022. [En línea]. Available: https://www.dacast.com/es/blog/live-streaming-vs-vod/. [Último acceso: 23 Mayo 2022]. |
| [23] | D. Matus, «digitaltrends,» 18 Noviembre 2020. [En línea]. Available: https://es.digitaltrends.com/entretenimiento/tv-streaming-comparativa-servicios/. [Último acceso: 22 Mayo 2021]. |
| [24] | 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]. |
| [25] | W. J. Stewart, Probability, Markov Chains, Queues, and Simulation, New Jersey: Princeton University Press, 2009. |
|  |  |