

# Capítulo 5

## Diseño

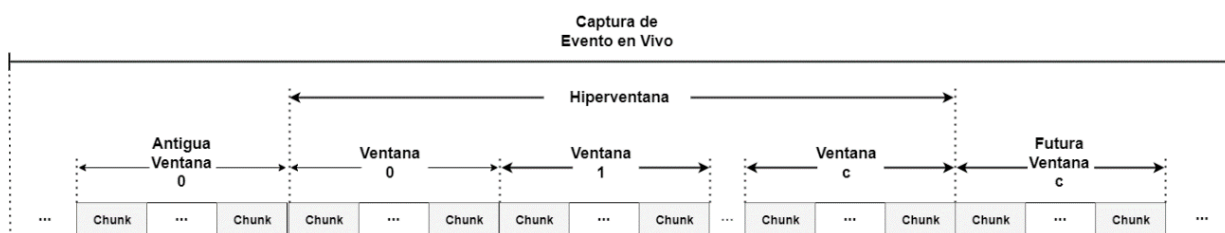
En este capítulo se describe el diseño y estructura de la cadena de Markov que representa a los servicios de video en vivo, así como el diseño del esquema de asignación de recursos que se ha propuesto para este tipo de servicios.

### 5.1 Diseño de la cadena de Markov para servicios de video en vivo

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 que puede tomar la cadena dado un evento que modifique su estado actual.

En este 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.

Los *peers* conectados al sistema descargan el archivo de video *chunk* a *chunk*, sin embargo, con el objeto de no tener una gran cantidad de poblaciones de *peers* descargando el archivo de video, se agrupan *N chunks* en segmentos de video más grandes llamados ventanas. El tamaño de las ventanas influye en la manera en que se distribuyen los recursos entre *peers*, pues al haber ventanas más grandes, la distribución se realiza entre un menor número de grupos de *peers* (poblaciones). Se espera que cada uno de estos grupos sea de un tamaño considerable para una distribución eficiente. Lo anterior se representa en la Figura 19.



### Figura 1. Estructura de una hiperventana

En el caso de video bajo demanda (VoD) se conoce la duración total del archivo de video y por lo tanto se divide en  $N$  ventanas de igual tamaño ( $n$  *chunks*) al momento de distribuirlo entre los usuarios. Sin embargo, para el caso de video en vivo no se conoce la duración total del archivo de video puesto que no se sabe con exactitud el tiempo que durará la captura del evento en tiempo real.

Considerando lo anterior, se optó por definir un contenedor llamado hiperventana, este contenedor es de longitud  $c$ , es decir, la hiperventana contiene de la ventana 0 a la ventana  $c$  y almacena las ventanas en tiempo real. La ventana 0, es la última ventana que se considera para que el usuario visualice el evento en tiempo real, por otro lado, la ventana  $c$  representa el fragmento de video correspondiente al evento capturado en tiempo real.

La hiperventana siempre es de longitud  $c$ , sin embargo, no es un archivo estático sino que su contenido cambia de manera simultánea a la producción del archivo de video. Es decir, cuando se captura una nueva ventana de video, la última ventana contenida en la hiperventana (ventana 0) sale de este contenedor, las demás ventanas se recorren y son reenumeradas para satisfacer la condición de que la hiperventana contiene las ventanas 0 a  $c$ .

En la Figura 19 se puede observar la estructura de la hiperventana compuesta por  $c$  ventanas. Las ventanas son identificadas con un subíndice  $i$ , para  $0 \leq i \leq c$ . A su vez dichas ventanas están conformadas por  $N$  *chunks*.

$c$ : Representa el índice de la ventana de video que se produce de forma simultánea a la captura del evento en tiempo real.

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

Como se mencionó anteriormente, los *peers* conectados al sistema de transmisión de video en vivo, son agrupados en poblaciones y dichas poblaciones se clasifican de acuerdo con la ventana que se encuentran descargando. En el modelo desarrollado para este proyecto se representa el comportamiento (variación en el tamaño) de cada una de las poblaciones en las ventanas del video en vez de representar el comportamiento individual de cada uno de los *peers*, con la finalidad de simplificar el modelo. La clasificación mencionada es la siguiente:

$X_i$ : Representa la población de *peers* que se encuentra descargando y reproduciendo la ventana  $i$ . Para  $i \in [1, 2, 3, \dots, c - 1, c]$ .

$X_0$ : Representa la población de *peers* en la ventana 0, únicamente pueden descargar el archivo de video para generar *buffer* y puedan visualizar la transmisión de video sin problemas de estancamiento en la descarga.

Con base en el análisis y revisión de los modelos para el consumo de servicio de VoD reportados en la literatura y al análisis de la cadena unidimensional (Erlang-B) se pudo obtener el comportamiento que tienen los usuarios en la visualización de un video y así identificar los sucesos que pueden ocurrir en la transmisión del video en vivo para comprender como estos influyen en la redimensión de las poblaciones de cada una de las  $N$  ventanas que componen al archivo de video.

Los principales sucesos 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 descarga/reproducción del video. El *peer* deberá recibir, con mayor prioridad, recursos de un servidor *CDN* para poder descargar el video para posteriormente él compartir recursos con *peers* que inicien la descarga del video tiempo después. 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 para comenzar la descarga/reproducción del video. En este caso el *peer* recibirá recursos de acuerdo con el esquema de asignación de recursos, con la finalidad de recibir porciones de recursos que necesita desde diferentes *peers* habilitados para compartir recursos.
- **Transferencia de un *peer* a la ventana superior:** Ocurre cuando un *peer* que estaba descargando la ventana  $i$  comienza a descargar la ventana  $i + 1$ . Por lo tanto, ya no forma parte de la población de la ventana  $i$  ahora es parte de la población que descarga la ventana  $i + 1$ .
- **Desconexión de un *peer*:** Por diversas razones un *peer* puede interrumpir la visualización de un evento (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 se encontraba descargando la ventana  $i$ , se desconecta, se decrementará en 1 la población de esa ventana.

De forma general, cualquier población de una ventana contenida en  $[0, N]$  se altera por la transferencia o desconexión de un *peer*. Sin embargo, el arribo de un *peer* únicamente ocurre en la ventana 0, que representa a la primera ventana del archivo de video.

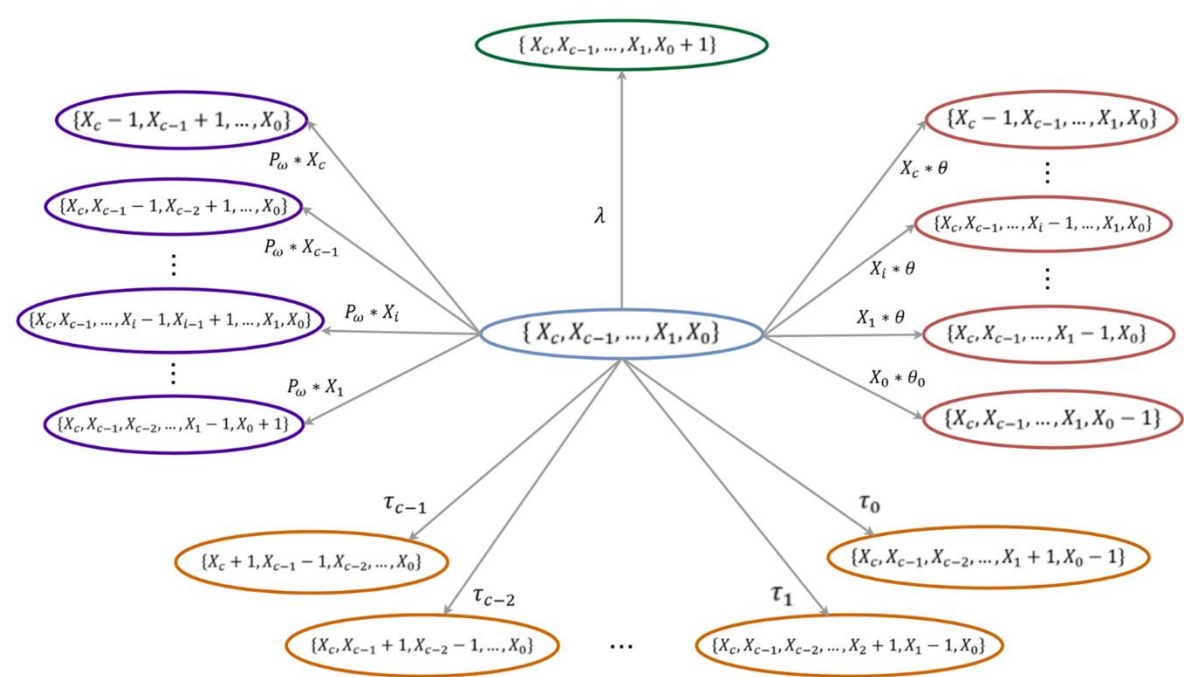
La desconexión de un *peer* contempla los casos reportados en la Tabla 4 con el fin de comprender la estructura de la distribución del video.

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

Tabla 1. 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 se estableció la transición que presenta la cadena en su estado, 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:



### *Figura 2. Cadena de Markov de un sistema de transmisión de video en vivo*

En la Figura 20, se observan los sucesos que generan una transición en el estado de la cadena y el resultado de esta.

En el centro de la Figura 20, se observa al vector  $\{X_c, X_{c-1}, \dots, X_1, X_0\}$ , que representa al estado de la cadena de Markov para servicios de video en vivo. Este vector contiene las poblaciones de *peers* en cada una de las ventanas contenidas en la hiperventana de un archivo de video (desde la ventana 0 hasta la ventana  $c$ ).

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  $X_i$ , contenidas en este vector estado, tiene un valor aleatorio.

En el sistema a modelar 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 producen un cambio en alguna(s) población de *peers* perteneciente a una ventana del archivo de video y por ende el estado de la cadena de Markov transita.

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 distribución de contenido en tiempo real. Dicha conexión se realiza a tasa  $\lambda$ , que representa la tasa de conexión de un usuario en general del sistema.

En este proyecto se supone que el usuario se debe conectar a la ventana 0 de la hiperventana para comenzar a descargar el video, crear *buffer* y así evitar congelamientos en la descarga del contenido.

Al conectarse un usuario al sistema, el estado general de la cadena de Markov transita del estado  $\{X_c, X_{c-1}, \dots, X_1, X_0\}$  al estado  $\{X_c, X_{c-1}, \dots, X_1, X_0 + 1\}$ . Este suceso se plasma gráficamente en la parte superior de la Figura 20.

Transferencia de un *peer* a la ventana inmediata superior: Cualquier población de *peers* correspondiente a una ventana contenida en  $[0, c - 1]$  se modifica al generarse la transferencia de un *peer* a la ventana superior inmediata, esta transferencia ocurre a tasa  $\tau_i$ , es decir, una vez que un usuario termina de descargar la ventana  $i$  y comienza a descargar la ventana  $i + 1$ , abandona la población  $X_i$  a tasa  $\tau_i$  y se agrega a la población en la ventana  $i + 1$  ( $X_{i+1}$ ).

Debido a la naturaleza de los videos en vivo y la definici3n de hiperventana que se introdujo para el desarrollo de este proyecto, un *peer* que est1 descargando la ventana  $c$ , al finalizar la descarga de esta, no puede ser transferido a una ventana superior inmediata; porque no hay disponible otra ventana para descargar. Es decir, la poblaci3n de *peers* que se encuentra descargando la ventana actual deben esperar a que se produzca otra ventana del video y esta entre a la hiperventana para poder descargarla.

Una vez que la nueva ventana es producida e ingresa a la hiperventana, de manera autom1tica la poblaci3n  $X_c$  ahora es la poblaci3n  $X_{c-1}$ . Y por lo tanto, pueden ser trasferidos a la ventana superior inmediata y comenzar la descarga de la ventana actual nuevamente.

El suceso de transferencia a la ventana superior inmediata, provoca una transici3n en el estado de la cadena de Markov como se describe a continuaci3n:

$$\{X_c, X_{c-1}, \dots, X_{i+1}, X_i, \dots, X_0\} \rightarrow \{X_c, X_{c-1}, \dots, X_{i+1} + 1, X_i - 1, \dots, X_0\}$$

El suceso descrito anteriormente est1 plasmado gr1ficamente en la parte inferior de la Figura 20.

$\tau_i$  : Representa la tasa promedio de transferencia de la ventana  $i$ . Y se define con la siguiente expresi3n:

$$\tau_i = \min\{C_\omega X_i, r_i\} \text{ para } i \in [0: c - 1]$$

Donde:

$C_\omega$ : Es la tasa de descarga general de un usuario dentro del sistema.

$r_i$ : Representa a los recursos de descarga efectivos en penuria para la ventana  $i$ , es decir, cuando la tasa de descarga es mayor que la tasa de subida el sistema entra en penuria y debe obtener recursos de la red *CDN*. Esta expresi3n depende del esquema de asignaci3n de recursos, por ejemplo, el esquema de distribuci3n uniforme que se retomara con m1s detalle en la siguiente secci3n.

Transferencia de un *peer* a la ventana inmediata inferior: Cualquier poblaci3n de *peers* correspondiente a una ventana contenida en  $[1, c]$  se modifica al generarse la transferencia de un *peer* a la ventana inferior inmediata a tasa  $P_\omega X_i$ , 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 est1 capturando el evento en tiempo real y produciendo una nueva ventana del video, por lo tanto el *peer* se atrasa en la descarga del video y pasa de la ventana  $i$  (abandona la poblaci3n  $X_i$ ) a la ventana  $i - 1$  (se adiere a la poblaci3n  $X_{i-1}$ ).

Por la naturaleza de los videos en vivo y la definici3n de hiperventana, un *peer* que est1 descargando la ventana 0 interrumpe su proceso de descarga, no puede ser transferido a una ventana inferior inmediata; porque esto provoca que salga de la hiperventana y por lo tanto abandone el sistema. Este caso en particular representa un factor para generar la desconexi3n de un usuario en la ventana, por ello, su efecto en el estado de la cadena de Markov se hace en conjunto a la desconexi3n de un usuario en la ventana 0, que se describir1 m1s adelante.

El suceso de transferencia a la ventana inferior inmediata provoca una transici3n en el estado de la cadena de Markov como se describe a continuaci3n:

$$\{X_c, X_{c-1}, \dots, X_i, X_{i-1}, \dots, X_0\} \rightarrow \{X_c, X_{c-1}, \dots, X_i - 1, X_{i-1} + 1, \dots, X_0\}$$

El suceso descrito anteriormente est1 plasmado gr1ficamente en la parte izquierda de la Figura 20.

Desconexi3n de un *peer* antes de terminar la transmisi3n: Por causas diversas (fallas de conexi3n, fallas en sus servicios, fallas en sus dispositivos, desinter3s en el contenido, etc.) un *peer* que est1 visualizando la transmisi3n de video en vivo puede desconectarse.

Entonces, cualquier poblaci3n de *peers* de la cadena cambia cuando se genera la desconexi3n de un *peer* a tasa  $X_i\theta$ , antes de que finalice la transmisi3n en vivo.  $X_i\theta$  representa la tasa promedio de desconexi3n de la poblaci3n que se encuentra descargando ventana  $i$ . Es el resultado de multiplicar la poblaci3n de la ventana  $i$  ( $X_i$ ) por  $\theta$  (tasa de desconexi3n de un *peer* conectado al sistema en general).

Este suceso provoca una transici3n en el estado de la cadena de Markov como se describe a continuaci3n:

$$\{X_c, X_{c-1}, \dots, X_i, \dots, X_0\} \rightarrow \{X_c, X_{c-1}, \dots, X_i - 1, \dots, X_0\}$$

Lo anterior quiere decir que un *peer* que formaba parte de la poblaci3n  $X_i$  por alguna raz3n abandono el sistema antes de finalizar la transmisi3n en vivo, por lo tanto, la poblaci3n de la ventana  $i$  se decrementa en 1. Este evento es representado en la secci3n derecha de la Figura 20.

Los *peers* conectados a la ventana 0 son m1s susceptibles a desconectarse del sistema, en caso de interrumpir su proceso de descarga, por una mala conexi3n a internet, problemas con el hardware, no tener 3ptimas QoE 3 QoS. Por lo tanto, se considera a  $\theta_0$  como la tasa de desconexi3n de los *peers* pertenecientes a la poblaci3n  $X_0$  y se define como:

$$\theta_0 = \theta + P_\omega$$

Donde:

$\theta$ : Representa la tasa de abandono general de un usuario dentro del sistema

$P_{\omega}$ : Representa la tasa a la cuál es producido el archivo de video.

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