3 Mejoras propuestas para un cliente de VoIP con soporte para QoS.

Tras el análisis realizado para la elección de nuestro cliente de VoIP, se va a describir a continuación las mejoras que se tienen que añadir al código fuente del programa elegido para proporcionarle el soporte de calidad de servicio, objetivo fundamental en el desarrollo del proyecto fin de carrera.

3.1 Soporte de empaquetado.

La codificación de las tramas de voz forma parte de la reducción de ancho de banda en la transmisión de los paquetes por la red, es por ello que la elección de un esquema de codificación basado en un bajo régimen binario puede proporcionar cierta calidad de servicio pero no suficiente. La gran mayoría de los clientes de VoIP disponen de varios códec para que el usuario pueda elegir libremente.

El sistema de empaquetado sí proporciona una mejora visible en la calidad de la comunicación, este sistema se fundamenta en la eliminación de la sobrecarga de cabeceras que añaden a la red los paquetes IP cada vez que son transmitidos, es por ello que para reducir esta sobrecarga sea necesario la transmisión de varias tramas de voz en cada paquete, conociéndose el *Nfpp* como el número de tramas por paquete.

Pero este número de tramas por paquete está limitado por el eco y otros factores que influyen en una conversación entre dos extremos distantes. Mientras menor sea el tamaño de la trama del códec, más tramas puede haber en un paquete sin que se tenga influencia en el retardo, es decir utilización de códec de bajo régimen binario.

Así, una elección aceptable es enviar hasta 120 milisegundos de voz codificada en cada paquete IP (esto equivale a 4 tramas por paquete en el caso del G.723.1).

La cabecera de cada paquete tiene un tamaño mínimo de 40 octetos, correspondiendo a las cabeceras IP, UDP y RTP:

- Tamaño de la cabecera IPv4: **20 octetos.**
- Tamaño de la cabecera UDP: **8 octetos.**
- Tamaño de la cabecera RTP: **12 octetos**.

La suma de todas estas cabeceras, forma la cabecera del paquete IP, que son **40 octetos**. El número de tramas codificadas por paquetes tiene influencia en el ancho de banda requerido para la transmisión.

Es por esto que la utilización de un sistema de empaquetado supondrá una importante reducción en la utilización del ancho de banda que repercutirá inmediatamente en una mejor calidad de servicio, puesto que se dispondrá de mayores recursos para transmitir el mismo contenido de información.

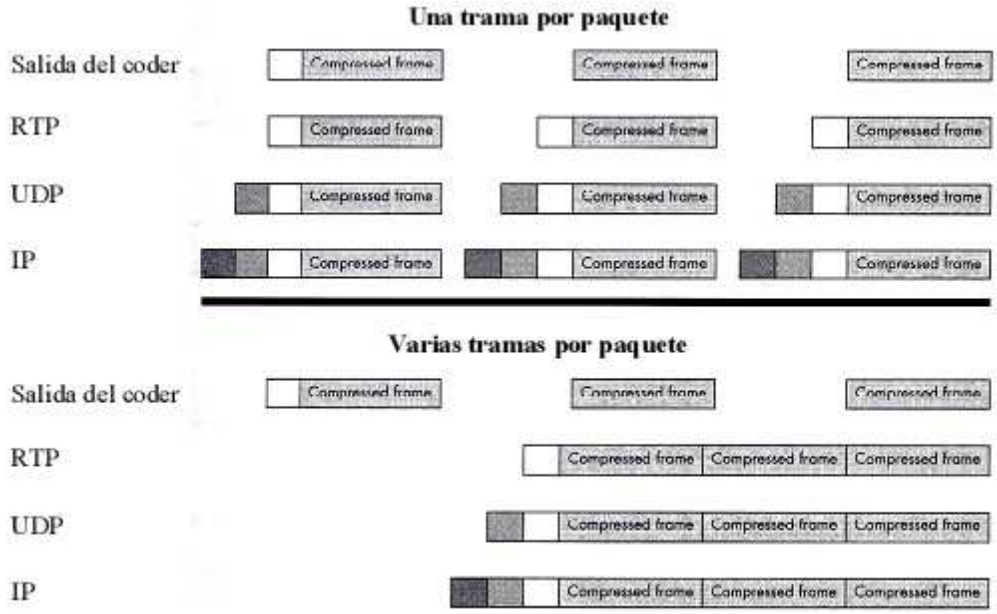

Ilustración 9: Influencia de las cabeceras en el empaquetado.

Cuantos más recursos se tienen para la transmisión de los paquetes de voz, menor es el retraso que sufrirán los paquetes repercutiendo directamente en la calidad de la conversación y siendo por tanto un factor fundamental en la implementación de un cliente de VoIP con soporte QoS.

El protocolo de transporte en tiempo real (RTP) usado sobre el nivel de transporte, identifica el tipo de carga que transporta el paquete. Además, inserta una marca de tiempo en la cabecera RTP lo que permite recuperar el instante de generación de la primera trama del códec que contiene el paquete [7]. Las siguientes tramas que contiene son consecutivas en el tiempo, así que pueden ser reproducidas en el instante apropiado sin necesitar ninguna información adicional.

Para la formación de cada paquete, hay que esperar por tanto un cierto tiempo (el tiempo de formación del conjunto de tramas que lo forman), lo que hace que se produzca un cierto retraso.

Este retraso va a variar dependiendo de si se usan codificadores que tengan activado el VAD y generen distintos tipos de tramas.

La detección de silencios (VAD) es una de las mejores formas de reducir el ancho de banda utilizado por un cliente de VoIP, puesto que la mayoría del tiempo se esta callado esperando a que el otro interlocutor termine de hablar, parece razonable no transmitir nada hasta que se desee hablar, de manera que durante los silencios no se transmite nada y conseguiremos reducir el consumo de ancho de banda en aproximadamente un 50%. El uso de esta técnica puede parecer en principio un poco desconcertante para el interlocutor del otro extremo, pues no oirá absolutamente nada, y puede parecer en principio que la comunicación se ha terminado, puesto que oír el ruido ambiente, propio de los modelos de telefonía por RTB, tenemos la sensación que la conversación continua activa y la ausencia del mismo nos llevaría a confusión.

Al generarse distintos tipos de tramas, se ha visto que las tramas SID que representan los cambios del ruido de fondo de los periodos de silencio son más pequeñas que el resto, y van a tener gran influencia ya que durante una conversación se habla una media del 35 por ciento del tiempo. Por eso, hay que tener en cuenta la existencia de estas tramas y su influencia para determinar el número de tramas por paquete y poder minimizar el efecto negativo que estas tramas tienen sobre la tasa de bits transmitidos.

Tradicionalmente se ha utilizado el empaquetado definido en la RFC 3551 [8]. Hoy día se presenta un nuevo empaquetado que intenta mejorar el ancho de banda ocupado El nuevo esquema pretende insertar varias tramas SID en un mismo paquete permitiendo por tanto reducir la sobrecarga introducida por la cabecera., al igual que se hace en los periodos de actividad de voz, en los que se transmiten varias tramas por paquete.

Vamos por tanto a explicar el esquema de empaquetado basado en la RFC 3551, puesto que es la que se ha utilizado en la realización de este proyecto.

3.1.2 Esquema de empaquetado según la RFC 3551.

En esta RFC se describe cómo los datos de audio y vídeo son transportados sobre RTP.

Debido a las características del medio de transmisión, en el que se producen retrasos variables para cada paquete, lo que hace que los paquetes puedan llegar desordenados, o incluso que se pierdan, es necesario incluir ciertos campos en la cabecera de los paquetes. Además, la capacidad que presenta el tráfico de voz de suprimir los silencios influye de forma considerable en estos aspectos.

Para poder ordenar de forma correcta los paquetes, la cabecera RTP lleva un número de secuencia y una marca de tiempo. De esta forma, el receptor puede distinguir entre paquetes perdidos y periodos de tiempo en los que no hay actividad de voz y no se están transmitiendo datos.

En los paquetes formados se tienen en cuenta el formato de la carga que llevan. Como se ha visto en el apartado de los codificadores, hay algunos códecs que definen un descriptor de inserción de silencio (SID) o trama de ruido confortable, que especifican parámetros para generar un ruido artificial durante un periodo de silencio y de esta forma aproximar el ruido de fondo de la fuente para que parezca que no se ha cortado la comunicación.

Tanto para este tipo de códecs que generan tramas SID durante los periodos de silencio como para los códecs que dejan de enviar paquetes, se debe poder distinguir el inicio de una ráfaga de voz. Esto se consigue poniendo el bit de marca a uno en la cabecera RTP del primer paquete de la ráfaga, es decir, del primer paquete que se transmite después de un periodo de silencio, en el que no ha habido una transmisión continua de paquetes. Este bit, en todos los demás paquetes tiene valor 0. De esta forma, el principio de una ráfaga, puede ser usado para ajustar el retraso de playout debido a los cambios producidos en la red.

El intervalo de empaquetado determina el mínimo retraso extremo a extremo. Paquetes con un mayor número de tramas, van a presentar menos sobrecarga de cabecera pero van a tener un retraso mayor y hacen la pérdida de tramas más notable, ya que si se pierde un paquete, se pierden todas las tramas que transporta.

La duración de un paquete viene determinada por el número de muestras en el paquete. La marca de tiempo del paquete indica el instante en el que la primera trama se muestreó, es decir, la información más antigua que lleva el paquete.

Por tanto, los codificadores basados en trama codifican un bloque de audio de longitud fija a otro bloque de datos comprimidos, normalmente también de longitud fija. Así, se van generando tramas de forma continua, y el emisor puede decidir combinar varias tramas en un único paquete RTP. Todos estos códecs, deberían poder codificar y decodificar varias tramas consecutivas dentro de un mismo paquete.

El receptor puede conocer el número de tramas que contiene un paquete RTP si todas las tramas tienen la misma longitud, dividiendo la longitud de la carga RTP por el tamaño de las tramas de audio, que viene indicado como parte de la codificación. Si no es así, las tramas deben indicar su tamaño.

En los periodos de silencio en los que se van generando tramas SID no consecutivas, hay bastante sobrecarga de cabeceras, debido a que se fuerza a la transmisión de una única trama SID por paquete, ya que la cabecera RTP sólo indica un instante de tiempo, y no hay manera de

averiguar el instante de reproducción para tramas SID que se encuentran entre tramas no transmitidas.

Hay que tener en cuenta que las características de codificación de tramas son distintas para cada códec, como se ha visto en el apartado anterior. Esto lleva a que los procedimientos de empaquetado sean distintos según el tipo de códec.

Para la realización del proyecto y las pruebas se ha acordado añadir el códec G.723.1, es por ello que nos vamos a limitar a describir el empaquetado para dicho códec, y no vamos a entrar en detalles de la realización de empaquetado para otros que no vayan a ser utilizados por el sistema de empaquetado que ha sido implementado y que en apartados posteriores se analizarán.

Las tramas de audio del códec G.723.1 tienen una duración de 30 ms, y pueden ser de tres tipos distintos. Los dos bits menos significativos del primer octeto de la trama determinan el tipo de trama y el tipo de codificación, es decir, si la trama generada es activa o SID, y la tasa de transmisión de estas tramas.

La forma de empaquetar el flujo de bits codificado en octetos y el orden de transmisión de cada octeto viene especificada en Recomendación G723.1.

En este codificador se indica el tipo de trama generado mediante dos bits. Por tanto, en un momento dado se puede saber el tipo de tramas que contiene el paquete simplemente comprobando el valor de estos bits.

De esta forma, el empaquetado de las tramas no impone ninguna restricción: las tramas G.723.1 SID pueden viajar sin problemas en distintos paquetes ya que su existencia viene indicada por los bits de cabecera HDR según el valor “1 0”. Únicamente al comprobar que a continuación va un periodo de silencio, se produce la transmisión del paquete.

El ejemplo de empaquetado se ve en la siguiente figura, en la que se utiliza una tasa de empaquetado igual a 3:

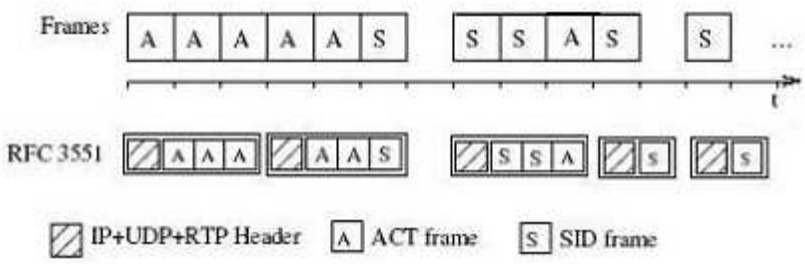

Ilustración 10: Empaquetado según RFC 3551. Codificador G.723. Nfpp = 3

3.2 Soporte de algoritmos de buffer de Playout.

La calidad de servicio en VoIP depende de que los paquetes sean reproducidos en orden y justo en el instante que les corresponde. En Internet los paquetes se ven afectados por retardos distintos, ya que no existe una reserva de recursos extremo a extremo, y eso es lo que hace difícil manejar un tráfico en tiempo real.

Por tanto no va a bastar con tener un control sobre la utilización de los recursos, pues a pesar de disponer de una buena cantidad de recursos la red Internet puede presentar retrasos variables independientemente de los que se dispongan.

Para compensar la variabilidad de estos retardos, se coloca en el extremo receptor un buffer, denominado **buffer de playout**. Conforme los paquetes van llegando, se van almacenando en este buffer. Tras esperar un cierto tiempo, comienza la decodificación, de forma que durante este tiempo, se ha permitido la llegada de paquetes que hayan sufrido un retraso mayor. A la salida los paquetes son ordenados y reproducidos en su instante correspondiente, con lo que se ha conseguido eliminar la variación de los retrasos.

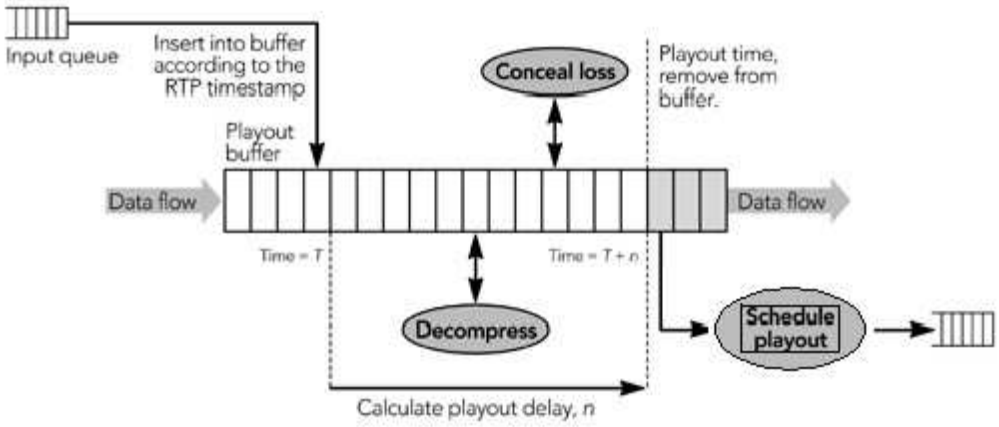

Ilustración 11: Buffer de playout

Así se define el *retraso de playout* como la cantidad de tiempo que transcurre desde que el paquete i es generado por la fuente hasta que se reproduce en el host de destino.

Seleccionar el r*etraso de playout* es importante porque directamente afecta a la calidad de la comunicación de la aplicación:

- Si el *retraso de playout* se define muy pequeño, los paquetes que hayan sufrido un retraso mayor se van a tratar como perdidos, aunque finalmente lleguen, ya que habrá pasado su instante de reproducción.
- Si el *retraso de playout* es demasiado grande (de forma que asegura que ningún paquete de la ráfaga tenga un retardo mayor que este valor y por tanto todos los paquetes lleguen a tiempo para ser reproducidos), se estaría introduciendo un retraso inaceptable que los usuarios no pueden tolerar.

La elección adecuada para el valor del *retraso de playout* no es una tarea sencilla, es más, dicho valor va a depender de los retrasos introducidos por las condiciones en la que se encuentre Internet en el momento de realizar la comunicación, es por ello que generalmente, retrasos entre la generación y reproducción de paquetes menores que 400ms y unas pérdidas de hasta el 5% son aceptables en una conversación.

Llegados a este punto podemos observar que combinar el sistema de empaquetado con una elección adecuada del retraso de playout nos permite ser capaces de adaptarnos a la variabilidad de la red Internet.

Precisamente el objetivo que se persigue en el proyecto fin de carrera es adaptar estas mejoras a un cliente de VoIP con el fin de obtener un programa capaz de adaptarse a las condiciones de la red y soportar adecuadamente las fluctuaciones de la misma, manteniendo en todo momento un nivel de calidad de servicio que nos permita una comunicación fluida y estable entre dos extremos de la red.

Los retrasos de playout podrían ser fijos en toda la duración de la sesión de audio o podrían ser ajustados entre ráfagas, aprovechando los periodos de silencio existentes, de forma que para elegir el nuevo valor se estén teniendo en cuenta las variaciones de retraso que se están produciendo justo en esos instantes. Aún siendo ajustados entre ráfagas, sigue habiendo un problema. Cuando un 'spike' (aumento brusco del retardo en un instante) está contenido en una ráfaga, la próxima oportunidad de cambiar el retraso de playout es al principio de la siguiente ráfaga, por lo que no se reacciona a tiempo. En casos en los que el 'spike' ocupa más de una ráfaga, se puede reaccionar.

Para el control del *retraso de playout*, se han propuesto distintos algoritmos que se conocen como algoritmos de buffer de playout (PBA). Algunos algoritmos tienen en cuenta los retrasos medidos, otros además consideran el porcentaje de pérdida de paquetes, etc.

Para describir estos algoritmos es útil la notación de la ilustración 11. Muestra los tiempos variables asociados con el envío y recepción del paquete i dentro de una llamada de audio.

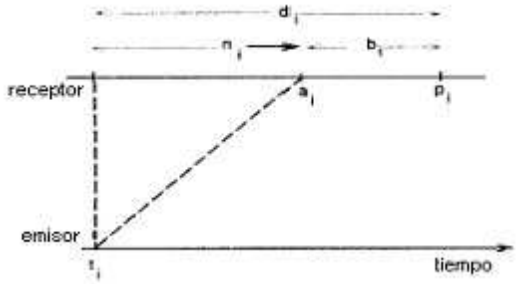

Ilustración 12: Retrasos sufridos por un paquete.

- t_i: instante de tiempo en el que el paquete i se genera en el emisor.
- a_i: instante de tiempo en el que el paquete i se recibe en el extremo distante.
- p_i: instante de tiempo en el que el paquete i ser reproduce en el extremo distante.
- b_i: la cantidad de tiempo que el paquete i pasa en el buffer del receptor esperando su tiempo programado de salida:

  $$b_i = p_i - a_i$$ Ecuación 3.2.1

- d_i: cantidad de tiempo desde que el paquete i es generado por la fuente hasta que es reproducido en el destino. Esto se conoce como *retraso de playout* del paquete i:

  $$d_i = p_i - t_i$$ Ecuación 3.2.2

- n_i: retraso total introducido por la red.

  $$n_i = a_i - t_i$$ Ecuación 3.2.3

En la siguiente ilustración, se puede ver la evolución de la transmisión de distintos paquetes, la variación de retrasos para cada uno, y la reproducción en los instantes correspondientes a la salida, habiendo realizado un ajuste en el periodo de silencio.

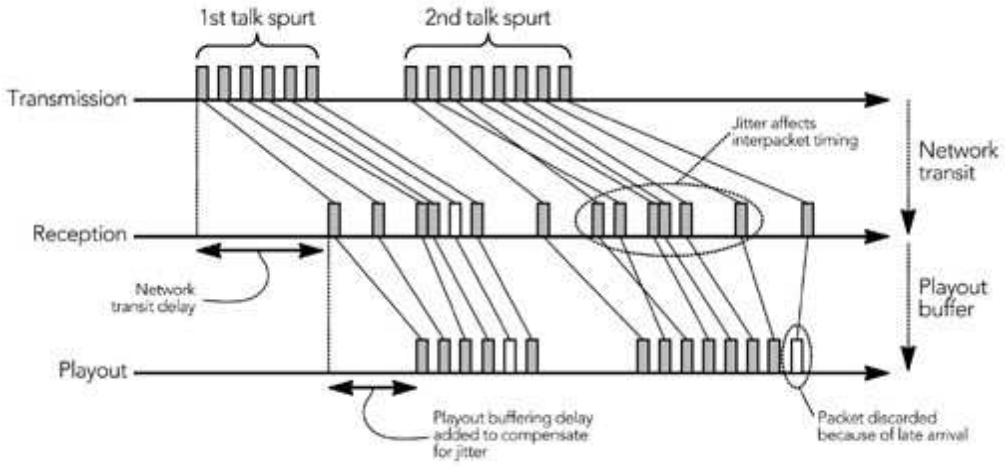

Ilustración 13: Efectos Jitter corregidos por el Buffer de playout.

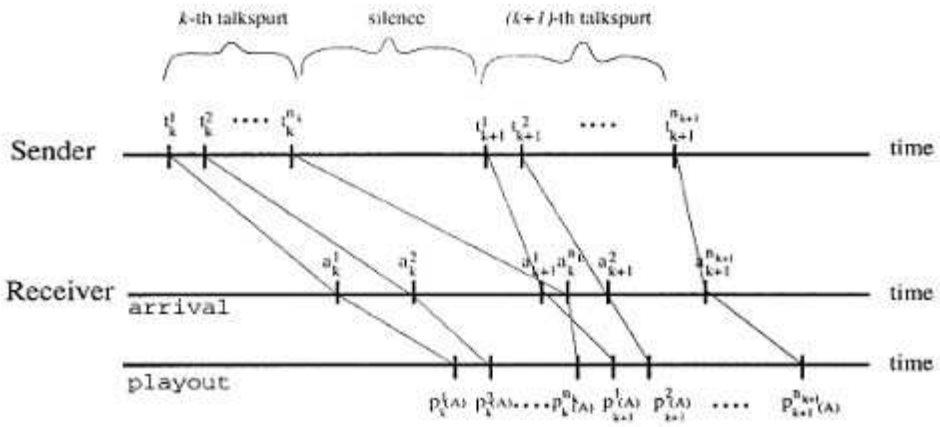

Ilustración 14: Evolución de los retrasos de playout

Para determinar el instante de reproducción del paquete i, hay que distinguir dos casos, dependiendo de si el paquete es el primero en la ráfaga o no:

- Si el paquete es el primero de la ráfaga, su instante de reproducción se calcula como:

$$p_i = t_i + \hat{d}_i + 4 * \hat{v}_i \quad \text{Ecuación 3.2.4}$$

  donde d_i y v_i son respectivamente la media y la variación del retardo extremo a extremo durante la ráfaga.

- El instante de reproducción para cualquier paquete que venga a continuación dentro de la misma ráfaga, se calcula como un offset respecto al instante de tiempo en el que el primer paquete en la ráfaga fue reproducido. Si i es el primer paquete en la ráfaga y el paquete j pertenece a la misma ráfaga, el instante de reproducción se calcula como:

$$p_j = p_i + t_j - t_i \quad \text{Ecuación 3.2.5}$$

Para cada paquete recibido se actualizaran los valores d_i y v_i de forma que continuamente se va calculando el valor de p_i, para tener en cuenta la evolución del retraso de todos los paquetes recibidos. Pero este valor p_i sólo se va a aplicar para determinar el instante de reproducción del primer paquete de la ráfaga.

Los distintos algoritmos de buffer de playout difieren en la forma en la que se calcula d_i , y por tanto se obtienen distintos valores de p_i y v_i, que dependen de valores de d_i, siendo d_i el mismo para todos los algoritmos (excepto para control de pérdidas y E-MOS) y se define como:

$$\hat{v}_i = \alpha \cdot \hat{v}_{i-1} + (1-\alpha) \cdot \left| \hat{d}_i - n_i \right|$$

Ecuación 3.2.6

En la ecuación 3.2.4 se usa el término $4 \cdot v_i$ para que el instante de playout sea suficientemente mayor que el retardo estimado, de forma que sólo una pequeña fracción de los paquetes que llegan después de su instante de reproducción se pierdan.

Al llegar cada paquete, se comprueba su tiempo de playout y se ve si ha llegado a tiempo o no para su reproducción. A continuación se explican distintos algoritmos para el cálculo de este valor, que van a ser utilizados posteriormente en el código fuente del programa.

3.2.2 Algoritmo de buffer de tamaño fijo.

Éste no es realmente un algoritmo. Se selecciona el tiempo de estancia en el buffer que va a tener siempre el primer paquete de cada ráfaga. Es tiempo lo esperan todos los paquetes que contienen la primera trama de una ráfaga de voz, de forma que ajustamos los retrasos aprovechando los instantes de silencio. Este algoritmo viene por defecto implementado en nuestro cliente de comunicación twinkle.

3.2.3 Algoritmo Exponencial-Media.

En este algoritmo [10], el retraso de playout del i-ésimo paquete que ha llegado es determinado a partir de los valores aproximados de la media d_i y la varianza v_i de los retrasos en un único sentido, que vienen dados por:

$$\hat{d}_i = \alpha \cdot \hat{d}_{i-1} + (1-\alpha) \cdot n_i$$

Ecuación 3.2.7

$$\hat{v}_i = \alpha \cdot \hat{v}_{i-1} + (1-\alpha) \cdot \left| \hat{d}_i - n_i \right|$$

donde:

- n_i es el retraso en un único sentido del paquete i-ésimo.
- α toma el valor 0.998002 de acuerdo con [11].

La ecuación para el cálculo del retraso de playout, tal y como se ha comentado sería:

$$\hat{p}_i = \hat{d}_i + 4 \cdot \hat{v}_i$$

Ecuación 3.2.8

De esta forma, sustituyendo p_i en la ecuación 3.2.6, el instante de tiempo de playout p_i se determina como la suma del retraso de playout $\hat{p}_i$ y el instante de tiempo de envío del paquete t_i:

$$p_i = \hat{p}_i + t_i$$

Ecuación 3.2.9

Este algoritmo estima el tiempo de playout a partir de las medias y las varianzas, y no considera la distribución de los retrasos.

3.2.4 Algoritmo para control de Spike.

Cuando varios paquetes se retrasan de manera brusca y repentina, provocan el fenómeno conocido como spike, de forma que al receptor llegará una ráfaga en forma de pico con las tramas de voz que han sido bruscamente retrasadas.

Este fenómeno puede provocar que los algoritmos que calculan el retraso de playout tomen un valor mucho mayor de lo que se requiere.

Provocando pérdidas de paquetes validos, para evitar esto el algoritmo debe ser capaz de detectar los picos en la red, y ser capaz de calcular un valor de playout adecuado a las condiciones del pico recibido.

Detectar el inicio de un pico (spike) es relativamente sencillo, consiste en detectar: si el retraso entre dos paquetes consecutivos es muy elevado es que ha ocurrido un retraso brusco o spike. Por lo tanto una vez que se ha detectado el pico (spike) debe de suspenderse el cálculo normal de retraso de playout para pasar a calcular el retraso de playout en el modo spike, y esto debe mantenerse hasta que se detecte el final del pico.

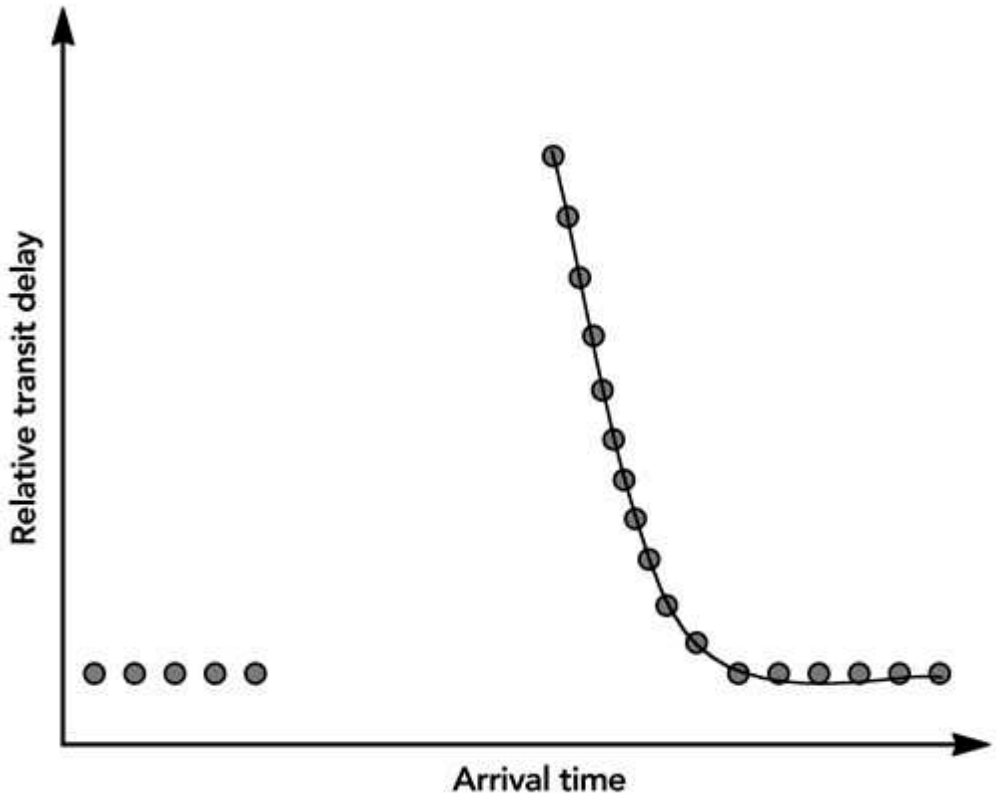

Ilustración 15: Tráfico de red durante un retraso por spike.

En el anexo de este documento se puede consultar el código fuente para los algoritmos de buffer de playout que han sido analizados y posteriormente utilizados en el código fuente del programa.

Para comprender el concepto e implementación del buffer de playout, se puede observar en el siguiente esquema como debe ser la interpretación de los valores y como implementarlos posteriormente en el código fuente.

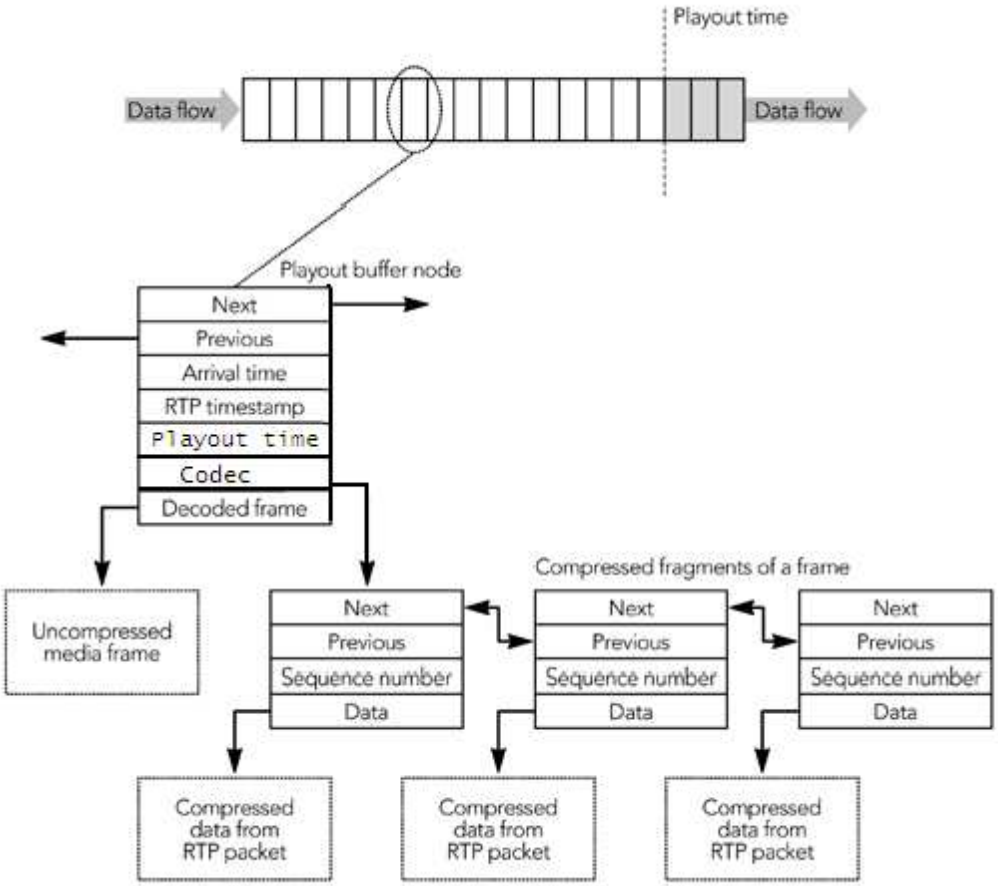

Ilustración 16: Estructura de datos de un buffer de playout y su implementación.

3.3 Cliente con parámetros de QoS configurables.

Todas las mejoras por si solas no implican absolutamente nada, necesitan un cliente de VoIP en el cuál puedan aplicarse, en el desarrollo de este proyecto de fin de carrera se ha perseguido precisamente ese objetivo, conseguir un cliente de comunicaciones por VoIP con unos parámetros de calidad de servicio configurables por el usuario.

De hecho, al cliente de software libre escogido, llamado twinkle, se le ha implementado las mejoras de QoS que se han sido mencionadas en el apartado anterior, es decir se ha añadido:

- Sistema de empaquetado y desempaquetado.
- Cálculo dinámico del retraso de playout.
- Códec G.723.1

Aprovechando que dicho cliente de VoIP disponía de una interfaz gráfica, se ha adaptado dicha interfaz permitiendo al usuario elegir:

- Nffp (el número de tramas por paquete)
- Códec a utilizar durante la comunicación, permitiendo seleccionar:
  - GSM
  - G.711
  - G.723.1
    - ✔ Régimen binario: 5,3 kbps ó 6,3 kbps
    - ✔ Detección de silencio (VAD) : On / Off
- Algoritmo de buffer de playout a utilizar:
  - spike
  - exponencial
  - fijo

Por tanto, en nuestro cliente de parámetros de QoS configurable, tenemos como variables:

- Nffp
- Algoritmo de PBA
- códec

En nuestro cliente de comunicaciones VoIP con QoS se debe conseguir que tanto el sistema de empaquetado como los algoritmos de buffer de playout sean capaces de integrarse, y de trabajar conjuntamente para alcanzar el objetivo fundamental de este proyecto, crear un cliente de VoIP

adaptativo a las condiciones caprichosas de la red de redes, Internet.

Conseguida la integración de la calidad de servicio (QoS) en este cliente, el siguiente paso sería automatizar el proceso de selección de los parámetros de forma que una vez configurada unas condiciones iniciales el programa fuera capaz de detectar los cambios en las condiciones de la red y posteriormente ir adaptándose progresivamente cambiando automáticamente los parámetros de configuración de forma transparente al usuario, intentando mantener en todo momento la calidad de servicio configurada por el mismo.

Posteriormente en este documento, se trataran las lineas de avance en la implementación de un cliente de VoIP autómata, capaz de adaptarse a las condiciones de la red sin la intervención del usuario.